Dissertation

Improving Prediction Accuracy Using Dynamic Information

by
Björn Böken

A thesis submitted in fulfillment of the requirements for the degree of a doctor rerum naturalium (Dr. rer. nat.)

at the

Lehrstuhl für Wirtschaftsinformatik, Prozesse und Systeme
Institut für Informatik und Computational Science
Mathematisch-Naturwissenschaftliche Fakultät

17 April 2023

Supervisor: Prof. Dr.-Ing. Norbert Gronau
Universität Potsdam

Second reviewer: Prof. Dr. Hanna Theuer
Hochschule für Wirtschaft und Recht Berlin

Third reviewer: Prof. Dr. Hanno Gottschalk
Bergische Universität Wuppertal

Date of defense: 13 February 2023

Published online on the Publication Server of the University of Potsdam:
https://doi.org/10.25932/publishup-58512
https://nbn-resolving.org/urn:nbn:de:kobv:517-opus4-585125

Abstract

Accurately solving classification problems nowadays is likely to be the most relevant machine learning task. Binary classification separating two classes only is algorithmically simpler but has fewer potential applications as many real-world problems are multi-class. On the reverse, separating only a subset of classes simplifies the classification task. Even though existing multi-class machine learning algorithms are very flexible regarding the number of classes, they assume that the target set $\mathcal{Y}$ is fixed and cannot be restricted once the training is finished. On the other hand, existing state-of-the-art production environments are becoming increasingly interconnected with the advance of Industry 4.0 and related technologies such that additional information can simplify the respective classification problems. In light of this, the main aim of this thesis is to introduce dynamic classification that generalizes multi-class classification such that the target class set can be restricted arbitrarily to a non-empty class subset $\mathcal{M}$ of $\mathcal{Y}$ at any time between two consecutive predictions.

This task is solved by a combination of two algorithmic approaches. First, classifier calibration, which transforms predictions into posterior probability estimates that are intended to be well calibrated. The analysis provided focuses on monotonic calibration and in particular corrects wrong statements that appeared in the literature. It also reveals that bin-based evaluation metrics, which became popular in recent years, are unjustified and should not be used at all. Next, the validity of Platt scaling, which is the most relevant parametric calibration approach, is analyzed in depth. In particular, its optimality for classifier predictions distributed according to four different families of probability distributions as well its equivalence with Beta calibration up to a sigmoidal preprocessing are proven. For non-monotonic calibration, extended variants on kernel density estimation and the ensemble method EKDE are introduced. Finally, the calibration techniques are evaluated using a simulation study with complete information as well as on a selection of 46 real-world data sets.

Building on this, classifier calibration is applied as part of decomposition-based classification that aims to reduce multi-class problems to simpler (usually binary) prediction tasks. For the involved fusing step performed at prediction time, a new approach based on evidence theory is presented that uses classifier calibration to model mass functions. This allows the analysis of decomposition-based classification against a strictly formal background and to prove closed-form equations for the overall combinations. Furthermore, the same formalism leads to a consistent integration of dynamic class information, yielding a theoretically justified and computationally tractable dynamic classification model. The insights gained from this modeling are combined with pairwise coupling, which is one of the most relevant reduction-based classification approaches, such that all individual predictions are combined with a weight. This not only generalizes existing works on pairwise coupling but also enables the integration of dynamic class information.

Lastly, a thorough empirical study is performed that compares all newly introduced approaches to existing state-of-the-art techniques. For this, evaluation metrics for dynamic classification are introduced that depend on corresponding sampling

strategies. Thereafter, these are applied during a three-part evaluation. First, support vector machines and random forests are applied on 26 data sets from the UCI Machine Learning Repository. Second, two state-of-the-art deep neural networks are evaluated on five benchmark data sets from a relatively recent reference work. Here, computationally feasible strategies to apply the presented algorithms in combination with large-scale models are particularly relevant because a naive application is computationally intractable. Finally, reference data from a real-world process allowing the inclusion of dynamic class information are collected and evaluated. The results show that in combination with support vector machines and random forests, pairwise coupling approaches yield the best results, while in combination with deep neural networks, differences between the different approaches are mostly small to negligible. Most importantly, all results empirically confirm that dynamic classification succeeds in improving the respective prediction accuracies. Therefore, it is crucial to pass dynamic class information in respective applications, which requires an appropriate digital infrastructure.

Acknowledgements

I would like to thank the following people for supporting this work during the last years without whom it would have not been possible to complete it: First and most importantly, I thank my supervisor Prof. Dr.-Ing. Norbert Gronau, University of Potsdam, Germany, for supervising and supporting my research, for the lot of advice, the guidance and the support whenever any form of help was required.

Next, I like to thank Prof. Dr. Hanna Theuer, Berlin School of Economics and Law, Germany, as well as Prof. Dr. Hanno Gottschalk, University of Wuppertal, Germany, for supporting this research project as second and third reviewer, respectively. Similarly, I would like to thank the remaining supervising team from the University of Potsdam: Prof. Dr. Christoph Kreitz as chairman of the committee as well as Prof. Dr.-Ing. Miloš Krstić and Prof. Dr. Torsten Schaub for their availability as committee members.

Besides this, I would like to thank Dr. Andreas Kasel, CSB-System SE, Geilenkirchen, Germany, for his support and all the many useful discussions we had during the last years, Dr. Peter Schimitzek, CEO of CSB-System SE, for his confidence and patience in funding most parts of this research project and the Fleischhof Rasting GmbH, Meckenheim, Germany, for supporting the prototype development as well as all my colleagues who supported my work throughout the last years. All the useful discussions we had were a great help in finalizing this work.

Finally, I thank my family and friends for all the support they gave and all the patience they had.

Contents

Abstract **V**

Acknowledgements **VII**

1 Introduction **1**

2 State of the Research **5**
2.1 Introduction . . . 5
2.1.1 Decision Rules on Data . . . 5
2.1.2 Loss and Risk . . . 5
2.1.3 Predictor Training . . . 6
2.1.4 Probabilistic Predictions . . . 7
2.1.5 Simplifying Assumptions . . . 8
2.2 Classifier Calibration . . . 10
2.2.1 Problem Definition . . . 10
2.2.2 Existing Techniques . . . 12
2.2.3 Existing Results and Open Issues . . . 18
2.3 Decomposition-based Classification . . . 22
2.3.1 Standard Decompositions . . . 23
2.3.2 Extended Decompositions . . . 25
2.3.3 Existing Results . . . 27
2.4 Comparison of Methods . . . 34

3 Classifier Calibration **37**
3.1 Theoretic Results . . . 37
3.1.1 Monotonicity . . . 37
3.1.2 Platt Scaling's Parametric Assumptions . . . 40
3.1.3 Platt Scaling and Beta Calibration . . . 43
3.1.4 Parameter Estimation . . . 44
3.2 New Calibration Techniques . . . 45
3.2.1 Kernel Density Estimation . . . 46
3.2.2 Ensemble of Kernel Density Estimation . . . 48
3.3 Evaluation Metrics . . . 49
3.3.1 Classification Metrics and Proper Scoring Rules . . . 49
3.3.2 Bin-based Evaluation Metrics . . . 50
3.3.3 Analyzing Evaluation Metrics . . . 50
3.4 Comparison of Calibration Techniques . . . 53
3.4.1 Simulation Studies . . . 53
3.4.2 Real-World Data . . . 62
3.5 Summary . . . 67

4 Evidence Theory **69**
4.1 Introduction to Evidence Theory 69
4.1.1 Mass Functions 70
4.1.2 Dempster's Rule of Combination 72
4.2 Application to Decomposition-based Classification 73
4.2.1 One-vs-All Decomposition 74
4.2.2 One-vs-One Decomposition 79
4.2.3 New Decompositions 85
4.3 Dynamic Classification using Evidence Theory 90
4.4 Summary 93

5 Generalized Pairwise Coupling **95**
5.1 Constant Pairwise Coupling 95
5.2 Bayesian Interpretation 97
5.3 Non-Uniform Generalization 98
5.3.1 Generalized Pairwise Coupling 99
5.4 Dynamic Classification using Pairwise Coupling 104
5.5 Computational Aspects 105
5.5.1 Extended Decompositions with Large-Scale Models 105
5.5.2 Weight Estimation 108
5.6 Summary 111

6 Evaluation **113**
6.1 Evaluation Metrics 113
6.1.1 Dynamic Risk 113
6.1.2 Sampling Strategies 115
6.2 Empirical Comparison 116
6.2.1 Overview of Methods 117
6.2.2 Reference Data 118
6.2.3 Deep Learning Data 125
6.3 Real-World Application 132
6.4 Summary 138

7 Conclusion **141**

Bibliography **147**

List of Figures

3.1 Illustration of the simulated score distributions used in the first two calibration experiments. 54
3.2 Nemenyi test results for the log-loss, the Brier score, the Kullback-Leibler divergence and the L^2 error in the simulation study. 56
3.3 Nemenyi test results for the bin-based error metrics. 57
3.4 Nemenyi test results for ECE and MCE using all samples. 60
3.5 Illustration of the simulated score distributions and the resulting posterior probabilities used in the the third calibration experiment. 61
3.6 Nemenyi test results for the log-loss and the Brier score in the fitted calibration setting. 65
3.7 Nemenyi test results for the log-loss and the Brier score in the predicted calibration setting. 65

4.1 Illustration of the belief $Bel(A)$, the plausibility $Pl(A)$ and the pignistic probabilities $BetP(\omega)$ based on (4.7). 71

5.1 Resultung Posterior probability $p^{\mathsf{WLW}}(\Delta)$ and their extrema for the system induced by $\Phi(\Delta)$ as given in (5.17). 102
5.2 Illustration of the differences between a one-vs-all softmax model (top) and a one-vs-one neural network with an optional dynamic classification fusing step (bottom). 106

6.1 Average ranks of the base classification rates for both classifiers in combination with a Nemenyi test. 123
6.2 Average ranks of the empirical dynamic risks according to sampling q_1 for both classifiers in combination with a Nemenyi test. 124
6.3 Average ranks of the empirical dynamic risks according to sampling q_{all} for both classifiers in combination with a Nemenyi test. 124
6.4 Training and test data losses of the deep neural networks (pretrained) on each of the five reference data sets. 129
6.5 Training and test data accuracies of the deep neural networks (pretrained) on each of the five reference data sets. 130
6.6 Training and test data losses as well as accuracies, respectively, of the deep neural networks on the real-world data set. 133

List of Tables

2.1 Summarized properties of the most relevant existing decomposition-based classification methods. 36

3.1 Estimated standard deviations per calibration technique for the predicted probability p, the log-loss φ^{LL}, the Brier score φ^{BS}, the Kullback-Leibler divergence KL and the L^2 error, each averaged over the respective data set. 62
3.2 Individual data sets and their most important properties. 63
3.3 Summary of the data set properties. 63
3.4 Number of data sets where predicted calibration reduces the calibration error. Additionally, the p-value of a one-sided sign test is given. 67

6.1 Individual data sets and their most important properties. 118
6.2 Average ranks of the three weight estimation techniques for each classifier and respective fusing method. 119
6.3 Number of data sets on which criterion (6.13) reports an improvement for support vector machines and random forests, respectively. Additionally, a Bonferroni-corrected p-value is given. 120
6.4 Improvements from evaluating (6.15) on all data sets. The given values are the average value and the standard deviation computed over all data sets. 121
6.5 Overview of the deep learning data sets. 125
6.6 Average ranks of the two weight estimation techniques for each network structure and respective fusing method. 127
6.7 Improvements from evaluating (6.15) on all deep learning data sets. The given values are the average value and the standard deviation computed over the five data sets. 128
6.8 Average classification accuracy per data set and fusing method based on the pretrained network architectures. 131
6.9 Average classification accuracy with dynamic class information according to sampling q_1 per data set and fusing method based on the pretrained network architectures. 131
6.10 Average accuracy and standard deviation computed over the five iterations of the two weight estimation techniques for each network structure and respective fusing method on the real-world data set. 134
6.11 Improvements from evaluating (6.15) on the real-world data set. Mean value and standard deviation are computed over the five iterations. . . 135
6.12 Dynamic classification accuracies on the different article groups. Mean value and standard deviation are computed over the five iterations. . . 136
6.13 Classification accuracies $\mathsf{Acc}(\mathsf{f})$ on data sets induced by $\mathcal{M} \subseteq \mathcal{Y}$. The given values are computed using the same data and models as in table 6.12 such that the difference yields the improvement of forwarding the dynamic class information to the fusing step. 137

Chapter 1

Introduction

The advancing progress in digital manufacturing and processing environments offers many benefits, but nevertheless simultaneously includes new requirements for both, human as well as non-human entities involved. Often, this involves a trade-off between predominant standards and new recommendations based on research. For example, classical production planning and control is *centrally* organized, while decentralized organization of cyber-physical production systems is part of ongoing research and development on Industry 4.0 for more than a decade [Leitão & Restivo 2006; Qin & Lu 2021]. Similarly, a production environment where all systems are directly interconnected rarely exists in practice.

Generally, cyber-physical systems in real-world processing environments are particularly relevant for automatization. On the one hand for obvious reasons as efficiency increases, but, on the other hand, different reasons as for example increasing skill shortage might necessitate automatization solutions in existing processes. Here, artificial intelligence and in particular machine learning became increasingly relevant in recent years. This is particularly noticeable by the presence of terms like artificial intelligence and deep learning even in non-scientific literature and media. Therefore, the main aim of this thesis is to transfer classification algorithms into contexts where the production processes allow simplifications of the decision problems based on additionally or externally supplied *dynamic* information.

Existing Machine Learning Approaches

In fact, nowadays there is a large variety of possible applications of data mining and machine learning algorithms in real-world problems. Existing examples throughout different domains cover decision analysis [Naeini et al. 2014], decision making systems [Guo et al. 2017], resource planning [Witt et al. 2019] and customer expenditure prediction [Bella et al. 2014]. Further applications of business-related predictive analytics tasks cover credit scoring [Cruz et al. 2018; Fonseca & Lopes 2017; Lessmann et al. 2015; Xiao et al. 2016], credit risk analysis [Bella et al. 2009a; Bequé et al. 2017] or investment management [Naeini et al. 2015a; Naeini & Cooper 2015, 2016, 2018; Naeini et al. 2015b] as well as failure prediction in business processes [Borkowski et al. 2019] and fraud or phishing detection [Arruti et al. 2014; Sun et al. 2018]. Possible use-cases also include automatic classification of internet contents [Joachims 1998; Montañés et al. 2013; Morales-Ramirez et al. 2019], traffic management [Zhao et al. 2016] or different kinds of security-related tasks like fingerprint detection [Hong et al. 2008] or face recognition [Jafri & Arabnia 2009; Yang et al. 2013]. Especially the latter also is an example from the large field of computer and machine vision tasks. Further interesting applications of cost-sensitive learning exist in medical contexts [Connolly et al. 2017; Jiang et al. 2012; Kim & Simon 2011].

Most of aforementioned applications can jointly be described as a prediction task between a given n-dimensional input domain or feature space $\mathcal{X}$ and a target space $\mathcal{Y}$ that seeks a function $\mathsf{f} : \mathcal{X} \to \mathcal{Y}$ mapping input instances $\mathsf{x} \in \mathcal{X}$ to their respective target values $\mathsf{y} \in \mathcal{Y}$. Here, $\mathcal{X} \subseteq \mathbb{R}^n$ holds in most cases and is always assumed as a standard Borel space, while the characteristics of the target space are more important. The scope of this thesis particularly lies at classification problems, i.e. the target set consists of discrete, mutual-exclusive labels that can be assumed without loss of generality as $\mathcal{Y} = \{1, 2, \dots k\}$. In the binary case, $\mathcal{Y} = \{0, 1\}$ and $\mathcal{Y} = \{1, -1\}$ are also often used for convenience instead. The target labels are also widely specified as *classes* and the prediction mapping f as a *classifier*. Still, the task-specific meaning of a class can vary arbitrarily: success or failure, category, state or event.

Machine Learning in Existing Processes

With respect to real-time processing environments, rare incorrect automatic predictions can result in substantial costs and, consequently, significantly decrease the efficiency of an automatization solution. For example, mistaking a lower worth product A as a higher worth article B can result in a reclamation of A such that at least additional costs equal to the difference between the respective worths result from the incorrect decision. What is more are follow-up costs where wrong products are supplied to further production processes such that a correction can be arbitrarily complicated or complex. As a matter of fact, incorrect decisions should be reduced as much as possible such that the accuracy both in manual and automatic decision processes should be as high as possible. This means that each option to increase the latter can significantly improve the profitability of the whole system.

On the other hand as previously discussed, production processes became increasingly interconnected in recent years and therefore have access to additional knowledge or at least are expected to have in the near future. This information can influence the characteristic of the classification task underlying the respective automatization solution by supplying additional information about ordered / produced units or goods such that certain ones are impossible to observe. For example, additionally supplying the information to the automatization solution that product A is currently not produced can be used to avoid mistaking of any product B as A: a detection of A is impossible, therefore any mistake with corresponding costs can be avoided.

A particular relevant and more concrete example application is the identification and classification of raw meat products in dissection factories. First, because the identification and classification of organic and therefore non-standardized material is more challenging than the one of artificial or more standardized goods. Currently, this is performed manually by human experts that identify the product inside a transportation unit (crate, box, etc.) such that this can be matched to an identification tag (barcode, QR code, RFID transponder, etc.). By joining this information and supplying it through digital process management solutions, identifying the product thereafter is possible by scanning the identification tag only.

Furthermore, this is also a good example of additional knowledge that allows a simplification of the decision problem. The process is often organized into different lines on which the products are put into the transportation units. Because the identification usually is performed *centrally* for multiple lines, while the preceding production processes are more specialized, backtracking the transportation unit by its identifier allows a restriction of the identification task: Only articles or products produced on the production lines on which the transportation unit was before are possible. Still, this information is fixed inside a factory because its production line

architecture does not change. This information is *static*, and it is relatively straightforward to explicitly address each possible combination as an independent decision problem.

Most importantly, this production process is often controlled by dissection lists that define the set of possible products that are currently produced. This information can easily change over time but interconnecting a classification solution with it, for example using an ERP system that has online information about the possible articles, allows a restriction of the target set while identifying the products.

Similar applications are possible wherever order or production lists are available by a planning or controlling instance (i.e. an ERP system) that defines which articles or goods are to be identified in a production process. In the general case, this additional *dynamic class information* can be modeled as a restriction of the target set $\mathcal{Y}$ to a subset $\mathcal{M} \subseteq \mathcal{Y}$ resulting from excluding all classes in $\mathcal{M}^{\mathsf{C}} = \mathcal{Y} \setminus \mathcal{M}$ such that the underlying prediction function $\mathsf{f}_{\mathcal{M}}$ has to restrict its target set to $\mathcal{M}$, i.e. $\mathsf{f}_{\mathcal{M}}(\mathsf{x}) \in \mathcal{M}$ has to hold.

Even though machine learning and in particular deep learning led to many interesting to groundbreaking results in recent years, it is interesting to observe that most state-of-the-art data mining and machine learning algorithms can only use this dynamic information in a relatively restricted manner. This is caused from the respective training processes. Once these procedures are finished, the resulting statistical models are *by design* relatively inflexible as the target set $\mathcal{Y}$ is assumed to be fixed. As a result, any intended modification can easily require a retraining of the model and thus is prohibitive once the additional information is supplied during a production process due to its large effort. Similarly, an explicit training for each possible target set is also prohibitive as it causes an exponential complexity: If there are k possible classes, there are $2^k - (k+1)$ subsets that contain at least two elements for which a training would be required.

Another strategy requires the prediction function f to compute an estimate of the posterior probability $\mathsf{f}(\mathsf{x}) \approx P(\mathsf{y} \mid \mathsf{x})$ such that the class prediction is induced by selection the class with maximum probability: $\arg\max_i \mathsf{f}_i(\mathsf{x})$. In this case, the additional information allows the conclusion that $\mathsf{f} \equiv 0$ holds on $\mathcal{M}^{\mathsf{C}}$, i.e. the estimated probabilities can be post-processed but simultaneously require a renormalization. Still, the posterior probability estimate in general will still depend on data from *all* classes, even though the additional knowledge obtained by the dynamic class information allows the conclusion that some are impossible to observe.

Dynamic Classification

In light of this, the main aim of this thesis is to introduce *dynamic classification*, which at first is a generalization of multi-class classification into a context supplying dynamic information such that the prediction function adapts to a given dynamic set $\mathcal{M} \subseteq \mathcal{Y}$ of possible classes that can change at any time between two consecutive predictions. Since $\mathcal{M}$ can also equal the set of all classes $\mathcal{Y}$, it is a strict generalization of multi-class classification. For discussed reasoning, it is similarly interesting from both, the process management as well as the algorithmic-theoretical point of view, but not directly discussed in the literature to date.

As there are no direct reference works aiming at this particular setting, the proposed strategy of this work is a combination of two related techniques. The first one focuses on post-processing of classifier outputs into well-calibrated, probabilistic predictions. The relevance of accurate posterior probability estimates and dynamic

classification on the one hand is relatively straightforward, as the latter enforces posterior probability estimates to become non-zero only on the given dynamic class subset $\mathcal{M} \subseteq \mathcal{Y}$. Here, reliable estimations require an adaption of the estimation process using the dynamic information. For comprehensive models, this is hardly possible at all. Therefore, the second set of techniques reduces a multi-class classification problem to a set of simpler, usually binary problems that are, thereafter, independently solved. During prediction, all individual predictions are computed and combined into an overall solution, such that this allows the extension of the fusing process with supplying additional, dynamic class information.

Thesis Structure

The remaining work is structured as follows. First, the subsequent chapter 2 presents both relevant algorithmic areas – classifier calibration and decomposition-based classification – in full detail, summarizes current research results and presents several open issues in sections 2.2 and 2.3, respectively. Thereafter, chapter 3 focuses on classifier calibration and contributes both, theoretical as well as empirical results that partly criticize and even contradict existing ones.

Based on the results of chapters 2 and 3, the following chapter 4 combines classifier calibration and decomposition-based classification into an evidence-theoretic modeling. Even though this results in a substantial theoretic formalism, it allows the analysis of decomposition-based classification in a formal framework that offers several advantages over existing approaches, which are – as will be presented in full detail in the summary in chapter 2 – often heuristically or empirically motivated but lack theoretical justification. Most importantly, evidence theory allows a consistent integration of the dynamic class information into the fusing process to yield both, a theoretically justified and a computationally tractable approach to dynamic classification.

Using the insights gained from the presented evidence-theoretic approach to dynamic classification, existing *pairwise coupling* algorithms are extended into *generalized* pairwise coupling in chapter 5 that in particular is designed to support the integration of dynamic class information. Additionally, a particular relevant aim are computationally tractable strategies to apply the presented algorithms in combination with large-scale classification models like deep neural networks. Here, traditional applications of the respective algorithms are easily computationally intractable, as they require to train and deploy multiple complex and large models.

Thereafter, a thorough empirical evaluation is performed in chapter 6. In particular, several data sets are evaluated to compare the introduced algorithms with existing reference methods. Especially relevant are the capabilities of the respective techniques to improve from integrating dynamic class information. This also requires to develop respective loss functions and corresponding evaluation metrics. Besides this, applying the algorithms on a real-world task that supplies dynamic context information and analyzing the respective improvements is a second main aim. Finally, all results are summarized in chapter 7, where additionally several open issues are discussed.

Chapter 2

State of the Research

2.1 Introduction

The main aim of realizing dynamic classification in real-world production environments requires a sufficiently interconnected infrastructure that supplies the respective dynamic class information – formally described by the target set $\mathcal{M}$ – to the corresponding automatization solution. Even though there is ongoing research on the right infrastructure management as discussed before – i.e. the right way *how* to solve these tasks – the actual way is not directly relevant for using this information in automatization solutions. Therefore, the focus of this thesis lies on extending existing data mining and machine learning algorithms to optimally adapt to these dynamic contexts.

2.1.1 Decision Rules on Data

Depending on the domain, the used classification function f can either be constructed *explicitly*, which necessarily requires task-specific knowledge about the relationship between the input features and the target classes, or *implicitly* by data. The former *rule-based* approach is advantageous if the relationship can uniquely be described by a feasible set of explicit decision rules. However, this unluckily is impossible in many applications. For example, humans often can solve certain recognition tasks (e.g. computer vision) very well, but they do not precisely know *how* they do it. Thus, the implicit approach is the only possible option in these particular cases. Here, it is assumed that a training data set

$$D \;=\; \{(\mathsf{x}_i, \mathsf{y}_i) \;:\; i = 1, \ldots, r\} \;\subset\; \mathcal{X} \times \mathcal{Y} \tag{2.1}$$

is given, where the x_i form the n-dimensional input vectors whose outputs are the y_i. Since the training data are also assumed to contain the class labels y_i, the task is restricted to be a *supervised* learning problem. Other problems like *unsupervised* or *semi-supervised* ones lie beyond of the scope of this work.

2.1.2 Loss and Risk

Formally, this setting means that the given training data set D forms an independently generated sample from an unknown distribution P over $\mathcal{X} \times \mathcal{Y}$, and the task is to infer the classification function f from D. In practice, there is a straightforward demand for classifiers that perform as good as possible, which requires to compare the results of different ones. Formally, this can be achieved using a *loss function*, which is a mapping $L : \mathcal{Y} \times \mathcal{Y} \to \mathbb{R}_{\geq 0}$ that given an instance (x, y), compares the predicted class $\mathsf{f}(\mathsf{x})$ to the true one y. The most commonly used loss function for classification problems is the 0-1 or binary loss $L_{\text{bin}}(\mathsf{f}(\mathsf{x}), \mathsf{y}) = \mathbb{1}(\mathsf{f}(\mathsf{x}) \neq \mathsf{y})$ that equals 0 for a correct

and 1 for an incorrect prediction. A given predictor's expected loss

$$\mathcal{R}(\mathsf{f}) \;=\; \mathcal{R}_{L,P}(\mathsf{f}) \;=\; \mathbb{E}_P\left[L \mid \mathsf{f}\right] \;=\; \int\limits_{\mathcal{X}\times\mathcal{Y}} L(\mathsf{f}(\mathsf{x}),\mathsf{y})\,\mathrm{d}P(\mathsf{x},\mathsf{y}) \tag{2.2}$$

with respect to L and P defines the *risk* of f and expresses the overall expected loss. Thus, it is a natural quantity to compare different classifiers. The best possible predictor f^* minimizing (2.2) over all measurable predictors[1] is the *Bayes-optimal* one and the corresponding risk the *Bayes risk* $\mathcal{R}^*$. In case of the 0-1 loss, the risk equals the overall error probability and consequently, the Bayes risk is identical to the minimal error probability. In particular, the Bayes-optimal predictor here by definition maps x to the class with maximum *posterior probability*: $\mathsf{f}^*(\mathsf{x}) = \arg\max_{i=1,\dots,k} P(\mathsf{y} = i \mid \mathsf{x})$.

It is important to emphasize that in general, neither the Bayes risk nor the Bayes-optimal predictor can be computed because both depend on the unknown distribution P. Still, these insights show the strong relation between well performing predictors and posterior probability estimation. A possible surrogate for the risk is the *empirical risk*

$$\mathcal{R}_{\mathrm{emp}}(\mathsf{f}) \;=\; \frac{1}{r}\cdot\sum_{i=1}^{r} L(\mathsf{f}(\mathsf{x}_i),\mathsf{y}_i) \tag{2.3}$$

that can be computed using the given data D. Often in real-world applications that use the binary loss, the minimization of the empirical risk is replaced by the equivalent maximization of the *accuracy* or *classification rate*

$$\mathsf{Acc}(\mathsf{f}) \;=\; \frac{1}{r}\cdot\sum_{i=1}^{r} \mathbb{1}(\mathsf{f}(\mathsf{x}) = \mathsf{y}) \tag{2.4}$$

which is why in combination with the binary loss, both terms are used interchangeably. Still from the theoretical point of view, the minimization of the loss is often preferred. By the law of large numbers, the empirical risk converges to the risk for $r \to \infty$ and, as a result, forms a Monte-Carlo approximation of it. This justifies to compare predictors using their empiric risks, which, however, should be estimated on independent data that were not previously used during predictor training.

2.1.3 Predictor Training

It is relatively straightforward to replace the risk with its empirical counterpart, however there is no similar equivalent replacement of the Bayes-optimal predictor. Therefore, one of the main aims of data mining and machine learning algorithms applied to these problems is to compute well performing (with respect to the respective loss function) predictors f.

Here, usually a statistical model is created that depends on a set of parameters. Thereafter, these are explicitly optimized over the training data set D. This is mostly performed using a maximum likelihood approach that in almost all cases requires iterative estimators, as closed-form solutions rarely exist in practice (linear regression forms an exception here). These iterative optimizations require at least partially differentiable dependencies between the problem's objective function and its parameters.

For this reason, a discontinuous loss like the 0-1 loss cannot be used during parameter fitting and is replaced by a continuous surrogate loss function. Popular alternatives are the $L^1(p,q) = \|p-q\|_1$ and $L^2(p,q) = \|p-q\|_2^2$ loss as well as the

[1] Formally, the minimizer is not guaranteed to exist and thus the Bayes risk only is an infimum.

Kullback-Leibler divergence $\mathsf{KL}(p, q) = \sum_{i=1}^{k} p_i \log \frac{p_i}{q_i}$, which is sometimes also called cross-entropy error. The training procedure to adapt a specific algorithm to minimize the respective (eventually also regularized) loss for a concrete problem can be resource and time consuming, but yields a trained predictor that can be used to classify newly observed instances x whose true class value y is unknown. This works surprisingly well throughout different applications. However, even well-performing predictors often have a small but still existing remaining error.

2.1.4 Probabilistic Predictions

Even though direct reference works focusing on dynamic classification are not available, accurate posterior probability estimation techniques are particularly relevant. Besides directly depending on any form of dynamic class information, there is a natural demand for predictors whose accuracies are as good as possible throughout aforementioned applications. Clearly, the best option is to predict the true but unknown posterior probabilities $P(\mathsf{y} \mid \mathsf{x})$, an oracle accurately estimating the latter could replace any other machine learning algorithm. Still, accurate probabilistic predictions are interesting besides purely optimizing the predictor's recognition performance.

First, probabilities can be used to extend predictions by a measure of trust or confidence to point to insecure classifications, for example by sorting multiple ones by their confidences. Besides this, there is also a straightforward demand for accurate probabilities in many aforementioned tasks where the risks of incorrect decisions require a reasonable trade-off against their chances of being correct, for example investment managements or all kinds of cost-sensitive [Elkan 2001b] classification tasks in general. Thus, they are a relevant research aim for many different reasons.

The first challenge in estimating posterior probabilities is that training data sets in practice never directly contain any information about the true posterior probabilities. The most that can be assumed is that all class labels are correct – which is also not guaranteed to hold in every application. Thus, it is explainable that the classical way of estimating the posterior probabilities is based on a *generative* approach to infer the class-conditional likelihoods $p(\mathsf{x} \mid \mathsf{y} = i)$ by applying probability density estimation techniques on the training data, and to convert them into a posterior probability using Bayes' theorem:

$$P(\mathsf{y} = i \mid \mathsf{x}) \;=\; \frac{P(\mathsf{y} = i) \cdot p(\mathsf{x} \mid \mathsf{y} = i)}{\sum_{j=1}^{k} P(\mathsf{y} = j) \cdot p(\mathsf{x} \mid \mathsf{y} = j)} \tag{2.5}$$

The required *prior probabilities* $P(\mathsf{y} = i)$ describe the respective class' probability without observing any data and can be estimated by using the respective fraction from the training data. Alternatively, the priors might be supplied from optional domain-specific knowledge. As a result, any posterior probability estimation problem and thus, also any classification problem, can in theory be solved using n-dimensional density estimation techniques. From a purely theoretical point of view, it additionally yields a direct way to integrate dynamic information. Restricting the target set here can be modeled by defining the prior probabilities $P(\mathsf{y} = i) = 0$ for all $i \notin \mathcal{M}$ and renormalize them on $\mathcal{M}$, therefore the remaining task is the estimation of the class-conditional likelihoods $p(\mathsf{x} \mid \mathsf{y})$.

There are various options to solve density estimation. Among the most popular ones are model-free approaches like nearest neighbor, kernel density estimation or histogram binning and parametric ones, where a certain parametric model is assumed

whose parameters are fitted, usually using a maximum likelihood approach. Common examples for the latter are Gaussian distributions or Gaussian mixture models fitted using the expectation maximization (EM) algorithm, however in general any parametric model can be used in the same way if there is some evidence available why the respective distribution is sufficiently suited for the corresponding task, but this is highly problem-dependent.

From the purely theoretical point of view, there exist solid statistical justifications why the respective approaches are well suited to solve the density estimation problem. It is well known that the empirical distribution function converges almost surely to the respective density function as well as it is proven that histogram and kernel density estimators asymptotically converge to the true distribution under relative mild assumptions [Wasserman 2006], which give rise to using model-free approaches. Similar theoretical approximation guarantees also exist for parametric models. For example, any probability density function can be approximated arbitrarily well by using a Gaussian mixture model with a sufficiently large number of components with properly selected parameters [Plataniotis & Hatzinakos 2000; Rossi 2014].

Even though these properties are positively remarkable, they can easily become less relevant in practice. The model-free density estimation techniques usually tend to work pretty well for low dimensional data, but as soon as there are more than a couple of dimensions, it is well known that they can suffer from different problems related to the input dimension [Liu et al. 2007; Scott & Sain 2005; Walt & Barnard 2017]. Error bounds usually are exponential in the input dimension and, consequently, require an exponential amount of data for accurate results, which is infeasible for large input dimensions. This fact is also known as the *curse of dimensionality*. Further common problems in the practical application of model-free approaches are the selection of the involved parameters, for example bin sizes of histograms or bandwidths of kernel density estimators, which is either performed heuristically or via cross validation techniques.

2.1.5 Simplifying Assumptions

To circumvent these problems, some kind of simplifying assumption is usually made in practice. Pretty common examples are assumptions over the number and form of individual distributions used for mixture models, for example as Gaussian distributions with equal variance and / or zero covariance, which reduces the number of parameters that have to be estimated. A relatively strong assumption even assumes fully independent distributions of the individual dimensions, such that the class-conditional likelihood factorizes into one-dimensional densities:

$$p(\mathsf{x} \mid \mathsf{y} = i) \;=\; p(x_1, ..., x_n \mid \mathsf{y} = i) \;\overset{!}{=}\; \prod_{j=1}^{n} p(x_j \mid \mathsf{y} = i) \tag{2.6}$$

If this holds, the full density estimation (and consequently the posterior estimation as well, even in contexts with dynamic class information) is reduced to one-dimensional density estimation, which often can accurately be solved using any of the aforementioned techniques. This is also the core of the *naive Bayes classifier*. Obviously, in practice these assumptions can be highly doubtful, but interestingly this does not render the approaches as practically useless. For example, data mining competitions were won by naive Bayes classifiers outperforming much more sophisticated techniques including decision trees, support vector machines and neural networks in the past [Elkan 2001a].

However, the aim of this work is not to completely review model-free or parametric density estimation techniques, which can be found in specific literature [Kruppa et al. 2014a,b; Malley et al. 2012; Simon 2014; Xu & Wang 2012]. These results should only demonstrate why generally, density and posterior probability estimation for more than a few dimensions is usually a challenging to very hard problem [Kim & Simon 2011; Simon 2014] and consequently, some kind of simplifying assumption is practically unavoidable.

Interestingly, neither for classification nor for posterior estimation, the class-conditional likelihoods are directly necessary at all, even if an estimate of the posterior distribution is required. Thus, a particularly interesting and much more common alternative in practice [Bella et al. 2014] is to use *discriminative* algorithms. From the theoretical point of view, there exist similar results justifying their application in practice. For example, a sufficiently large two-layer neural network can approximate any continuous function on a compact input set to arbitrary accuracy [Bishop 2009; Gebel 2009]. Similarly, there are predictors that are proven to be *universally consistent* [Devroye et al. 2008], i.e. for sample sizes $|D| \to \infty$, their losses converge to the Bayes loss for any data-generating distribution. However, these asymptotic properties can be less relevant in practice as well.

From the practical point of view, this means that there are different options to estimate posterior probabilities by using discriminative models that are theoretically justified. Interestingly, their estimated posterior probabilities commonly tend to be skewed and biased, i.e. they often differ from the true but unknown ones: the predictions are *uncalibrated*, as discussed in full detail in the next section 2.2. This is a remarkable observation because the algorithms still often succeed in discriminating the classes well. Thus, at least the maximum index in the posterior distribution is reasonably well estimated such that it is an interesting problem whether, and, if so, how the distorted posterior estimation can be post-processed into more accurate *calibrated* estimates.

Besides the direct application of posterior probability estimation in aforementioned contexts, accurately estimated probabilities are useful for model interpretability [Guo et al. 2017] and have important applications at techniques that combine different classifiers. Most importantly, it has a strong impact in classifier systems, where individual predictions from multiple classifiers are combined into an overall prediction [Bella et al. 2013; Bennett 2006; Xu et al. 2016], but is also applied as part of more complex algorithms like casual Bayesian networks [Jabbari et al. 2017] and probability calibration trees [Leathart et al. 2017]. Furthermore, particularly interesting are its applications in reduction- and decomposition-based approaches like multi-class support vector machines [Chang & Lin 2011; Wu et al. 2004] whose aim is to reduce a multi-class classification problem to a larger set of simpler, usually binary individual problems. In most cases, the latter are independently solved, and, thereafter, their solutions are aggregated into an overall prediction. Thus, they allow the application of binary-only classifiers like the support vector machine or AdaBoost in multi-class settings.

The key strategy to realize dynamic classification models will be the combination of both algorithmic techniques, classifier calibration as well as decomposition-based classification, and extending the final decision making aggregation step by using the respective dynamic class information. Therefore, the next two sections 2.2 and 2.3 at first present both research lines in full detail before summarizing the existing methods to formulate the research aims in section 2.4.

2.2 Classifier Calibration

The previous part summarized that even though there is a solid theoretical justification for the generative approach to classification to compute posterior probabilities, it is infeasible for many real-world applications without further simplifying assumptions. On the other hand, applying a discriminative model often tends to yield more accurate results with respect to accuracy or error statistics. However, the respective posterior probability estimates are often uncalibrated, i.e. do not well correlate with the true but unknown ones [Bella et al. 2009b; Cohen & Goldszmidt 2004; Flach 2016; Guo et al. 2017; Kull et al. 2017; Naeini & Cooper 2016, 2018; Naeini et al. 2014]. In particular, this observation is specifically discussed for different models like naive Bayesian classifiers [Bennett 2000; Domingos & Pazzani 1996] and decision trees [Zadrozny & Elkan 2001a,b, 2002], random forests [Dankowski & Ziegler 2016] as well as logistic regression models [Jiang et al. 2012]. Even the introduction of deep neural networks significantly increased the recognition performance, but interestingly also resulted in uncalibrated probabilistic outputs [Guo et al. 2017].

With respect to generative techniques modeling the likelihood distributions, these observations are explainable by the underlying simplifications that distort the results and thus, also the posterior estimates. For the discriminative models, it means that the training procedures succeed at approximating $\arg\max_i P(\mathsf{y} = i \mid \mathsf{x})$ without well approximating $P(\mathsf{y} \mid \mathsf{x})$ itself. To remedy this issue, it may be possible to design other prediction algorithms that even approximate the true posterior probabilities $P(\mathsf{y} \mid \mathsf{x})$ reasonably well. At least from practical requirements, it is well explainable that these do not exist in the same way as there are well-performing classification algorithms. Even if posterior probability estimates are required as well, it might still be more important to discriminant the classes well [Naeini et al. 2014], while the demand for probabilities is mostly an extension, at least historically as the requirement for *accurate* posterior probability estimation has not been that present when these algorithms were actually introduced [Bella et al. 2009b]. As a consequence, today there is no algorithm available that efficiently computes accurate posterior probabilities from sample data of computationally tractable size. Thus, posterior probabilities can only approximately be estimated and directly computing them remains infeasible without any simplifying assumption.

Therefore, this section summarizes classifier calibration that aims at transferring classifier predictions into posterior probabilities by an explicit postprocessing step. First, subsections 2.2.1 and 2.2.2 introduce calibration in full detail and present respective techniques. Thereafter, several open issues are summarized in subsection 2.2.3 that will be analyzed in chapter 3.

2.2.1 Problem Definition

The step of transferring a classifier's output into probability estimates that are intended to be well calibrated is generally known as *classifier calibration*. Besides the demand for accurate probabilistic predictions, this is also an interesting issue from the theoretical point of view. Discriminative algorithms infer the decision boundaries between the classes using all data *simultaneously*, while in the generative approach, each likelihood is only inferred using data from *one* class, ignoring the remaining ones. Thus, inferring (or approximating) the Bayes-optimal decision border using a classification algorithm is also quite interesting from the statistical point of view besides maximizing its predictive performance because the potential approximation of the Bayes-optimal prediction is completely captured in the decision function. Hence,

it is especially interesting to analyze how the classification algorithm's ability to infer the decision boundary might be generalized to infer further details of the posterior distribution while maintaining the underlying classifier's discriminative power.

As a basis for all further analysis, an arbitrary classifier f is needed as a first step. In practice, there is an obvious strong bias towards maximum accuracy while the challenging task is to transfer the corresponding outputs into calibrated probability estimates. It is necessary that the classification mapping is assumed as a vector-valued function $\mathsf{f} : \mathcal{X} \to \mathbb{R}^k$ consisting of components $\mathsf{f}(\mathsf{x}) = (\mathsf{f}_1(\mathsf{x}), \mathsf{f}_2(\mathsf{x}), \ldots, \mathsf{f}_k(\mathsf{x}))^\top$ instead of a class prediction only. The latter is simply induced by $\arg\max_i \mathsf{f}_i(\mathsf{x})$. This is not a restriction as most classifiers compute some kind of scoring function that is used to resolve the class index. The special case of a *probabilistic* classifier, i.e. the function's target space already is restricted as $[0, 1]^k$ and each prediction is normalized to sum one, is explicitly included but not assumed. In the binary case of $k = 2$ classes, this almost always results in a one-dimensional mapping $\mathsf{f} : \mathcal{X} \to \mathbb{R}$ that is thresholded at 0, or a probability estimation function $\mathsf{f} : \mathcal{X} \to [0, 1]$ intended to approximate $P(\mathsf{y} = 1 \mid \mathsf{x})$. The vector-valued, two-dimensional equivalents are recovered by $(\mathsf{f}, -\mathsf{f})$ or $(\mathsf{f}, 1 - \mathsf{f})$, respectively. Notably by applying classifier calibration, also algorithms that initially do not allow probabilistic interpretations like support vector machines become applicable in settings that require them.

For several reasons, the binary case is of primary interest. First, even here accurate posterior probability estimation is already a very hard problem. Second, many of aforementioned applications are binary classification problems, thus accurately estimating them only in two-class scenarios has many potential benefits. Third, the problem is analytically and implementation-wise simpler if numeric values are processed instead of vector-valued ones. Finally and most importantly, decomposition strategies exist to reduce multi-class problems to binary ones. As will be shown in section 2.3, these in theory even allow the complete solving of the general, multi-class posterior estimation problem by using binary calibration techniques only.

From the practical point of view, this approach is advantageous as the calibration step is independent of the underlying classification algorithm. Despite that its main idea seems to be a straightforward generalization of discriminant methods, it is still relatively rarely studied in data mining and machine learning research [Hüllermeier & Vanderlooy 2010; Naeini et al. 2015a; Naeini & Cooper 2015, 2016, 2018; Naeini et al. 2015b], and even a strictly formal definition is not available in the literature. However, there exists the definition of a *well calibrated classifier* [Murphy & Winkler 1977; Zadrozny & Elkan 2002] to formalize the main aim of calibration.

In particular, a probabilistic classifier f is well calibrated if the empirical class distribution of samples $P(\mathsf{y} \mid \mathsf{f}(\mathsf{x}) = p)$ with predicted probabilities $p \in [0, 1]^k$ converges to p if the number of instances go to infinity [DeGroot & Fienberg 1983; Kim & Simon 2011; Zadrozny & Elkan 2002]. Despite being a frequentist concept, this can also be interpreted from the Bayesian point of view [Bennett 2006]. A further restriction into *perfect calibration* [Bella et al. 2013; Guo et al. 2017; Kull et al. 2017] requires $P(\mathsf{y} \mid \mathsf{f}(\mathsf{x})) = \mathsf{f}(\mathsf{x})$ to hold for all predictions $\mathsf{f}(\mathsf{x}) \in [0, 1]^k$.

Even though these definitions reflect the rationale behind a probability, they are not directly helpful in practice because they impose a trade-off between accuracy and calibrateness [DeGroot & Fienberg 1983; Kull & Flach 2015]. Taking an arbitrary binary classification problem with known class priors, for example a balanced one with class priors equal to 0.5 [Bella et al. 2013; Flach 2016], in combination with a constant predictor $\mathsf{f} \equiv P(\mathsf{y} = 1)$ is perfectly calibrated but practically fails in separating the classes. Also the very reverse is possible [Jiang et al. 2012]: Using a well performing classifier together with arbitrary probabilities obviously is badly calibrated but highly

discriminative. Further, perfect calibration can never be achieved using a *finite* data set only [Guo et al. 2017] as well as there is no quantity like a calibrateness degree available. Instead, the definitions are strictly binary and only formalize the main aim but are not directly helpful or useful in achieving it.

Common definitions of classifier calibration are based on slightly informal terms like "transforming the classifier outputs into probabilities" [Azami et al. 2016; Bennett 2006; Connolly et al. 2017; Gebel 2009; Kull et al. 2017; Naeini & Cooper 2015, 2016, 2018; Xu et al. 2016]. Even though this accordance exists, it still allows different interpretations. Sometimes [Bequé et al. 2017] it is defined as the converting to calibrated probabilities. Formally, a well calibrated probability is hardly computable, thus it can at most be interpreted as the conversion into *better calibrated probabilities*. Paradoxically, these are not formally defined as the definition is strictly binary and defining them requires an accordingly selected error measure. Interestingly, these are not straightforward as the ground truth posterior probability is unknown, such that selecting a calibration error metric is also a highly non-trivial task. In this regard, the next part presents existing state-of-the-art classifier calibration techniques.

2.2.2 Existing Techniques

As a basis for any calibration technique, an arbitrary existing classifier f is needed. To fit the calibration function, it is assumed that together with the data set D, also the set of predictions $\{\mathsf{f}_i = \mathsf{f}(\mathsf{x}_i) : i = 1, \dots, r\}$ is given. The aim is to estimate a calibration mapping that can be used to compute probabilities of newly observed instances x that are intended to approximate the unknown posterior probabilities $P(\mathsf{y} \mid \mathsf{x})$. In fact, almost all existing techniques are designed for binary predictors such that in the following part, the prediction mapping can be identified with a real-valued function $\mathsf{f} : \mathcal{X} \to \mathbb{R}$ and hence, the calibration mapping as an approximation of $P(\mathsf{y} = 1 \mid \mathsf{x})$.

The first calibration technique is *Platt scaling* [Platt 1999, 2000] that assumes a parametric relationship between real-valued scores $\mathsf{f}(\mathsf{x})$ and posterior probabilities $P(\mathsf{y} = 1 \mid \mathsf{f})$. To calibrate f, a parametric sigmoid function of the form

$$P(\mathsf{y} = 1 \mid \mathsf{f}(\mathsf{x})) \;=\; \sigma_{a,b}(\mathsf{f}(\mathsf{x})) \;=\; \frac{1}{1 + \exp(a \cdot \mathsf{f}(\mathsf{x}) + b)} \tag{2.7}$$

with $a, b \in \mathbb{R}$, $a < 0$ is fitted using a maximum likelihood approach. Hence, Platt scaling is equivalent to a one-dimensional logistic regression of y by f. Sometimes [Bennett 2006] minor implementation issues are used to differ between Platt scaling and logistic regression, however throughout this work, Platt scaling generally refers to *any* parametric model of the form (2.7). The different ways how the parameters are estimated is a subsequent implementation detail.

Once the two parameters a and b were estimated, the additional time requirement at prediction time is negligible as only a few additional computations are required to obtain the probability, which makes Platt scaling extremely efficient. It is straightforward to see that the transformation is both differentiable and strictly monotonic, i.e. for all predictions z_1 and z_2 with $z_1 < z_2$ holds $\sigma_{a,b}(z_1) < \sigma_{a,b}(z_2)$, and therefore also continuous as well as invertible.

An alternative to Platt scaling is *histogram binning* [Zadrozny & Elkan 2001b], whose application is relatively straightforward. The range of all predictions is partitioned into b bins

$$[\mathsf{f}_{\min},\ \mathsf{f}_{\max}] \;=\; [z_0, z_1] \;\cup\; (z_1, z_2] \;\cup\; (z_2, z_3] \;\cup\; \cdots \;\cup\; (z_{b-1}, z_b] \tag{2.8}$$

and for each bin, the empirical probability distributions are computed using the data set D. In particular, for the predictions of a newly observed instance x, at first the respective bin satisfying $\mathsf{f}(\mathsf{x}) \in (z_{i_0}, z_{i_0+1}]$ is found such that the respective posterior probability is obtained as the bin's empirical one. Thus, after the binning model is computed, it serves as a lookup table for the posterior probability and only requires to project each prediction to the respective bin. Moreover, the estimation is model-free and non-monotonic but also discontinuous.

To circumvent the relatively arbitrary selection of the bin size, the same authors introduced *isotonic regression* [Zadrozny & Elkan 2002] as a calibration technique. Here, the selection of the bin size is replaced with the constraint of a monotonically increasing transformation [Guo et al. 2017; Xu et al. 2016] $\rho : \mathbb{R} \to [0, 1]$ that minimizes the mean squared errors $\sum_{i=1}^{r} (\rho(\mathsf{f}(\mathsf{x}_i)) - \mathsf{y}_i)^2$ between the predicted probability and the respective true class value in $\{0, 1\}$, which can be efficiently computed using the pair-adjacent violators (PAV) algorithm [Gebel 2009; Niculescu-Mizil & Caruana 2005a,b]. The optimal solution is a piecewise constant binning model and thus, can be interpreted as a hybrid method between Platt scaling and binning. Generally, this approach could also be extended to higher order than squared differences, but solutions cannot be computed efficiently anymore [Jiang et al. 2012]. In a similar way, also relatively recent modifications exist that transform the piecewise constant predictions into a smooth monotonic function using cubic Hermite interpolating splines [Jiang et al. 2011] as well as monotonic higher-degree polynomials [Wang et al. 2019].

Following work on classifier calibration applied asymmetric distributions like the asymmetric Gaussian or asymmetric Laplace [Bennett 2006, 2003; Zhang & Yang 2004] to calibrate asymmetrically-shaped scores that empirically were observed at text classification. However besides the introducing works, these techniques were not used in more recent studies.

A different work [Bella et al. 2009b] combined the approach of histogram binning with the K-nearest neighbor algorithm into the calibration technique of *similarity-binning averaging* (SBA). In particular, the authors criticized that existing approaches are based on the classifier outputs $\mathsf{f}(\mathsf{x})$ only and discard the input instance x itself. Therefore, they dynamically compute the K-nearest neighbors of the combined vector $(\mathsf{x}, \mathsf{f}(\mathsf{x}))$ in the training data set and compute the empirical class probabilities inside this local neighborhood. Consequently, SBA suffers from the same drawbacks as the K-nearest neighbor algorithm [Naeini 2016]: The selection of K is arbitrary or at least unclear, each prediction requires a dynamic search in the training data and distance-based comparisons can be problematic in large dimensions. Furthermore, exhaustive neighbor searches require the whole training data to be kept in memory and thus, can be too resource- and time-consuming if they have to be performed at each prediction. Finally, if the input dimension n is large, the neighbor search will be dominated by the input features themselves, such that the whole procedure is likely to become nearest neighbor-based posterior estimation. Instead, the key concept of classifier calibration should be to approximate the posterior probabilities using a classification function that approximates the optimal decision boundary and thus, circumvents the problems that are usually caused by the large input dimension that make a direct estimation infeasible. So criticizing calibration techniques for ignoring the original input features is not really reasonable at all – they are *designed* for exactly this purpose.

A completely different approach called *adaptive calibration of predictions* (ACP) is based on confidence intervals [Jiang et al. 2012]. Here, the core idea differs from any of the previously presented approaches. Instead, it is assumed that besides the estimate of the predictor $\mathsf{f}(\mathsf{x})$ also a 95 % confidence interval is available from which the calibrated probability is computed. This has two critical drawbacks. First, it

is only applicable for classification algorithms that allow the computation of such a confidence interval like for example logistic regression that is used in the introducing work, which conflicts with the modular design of calibration techniques. The second issue is even more problematic. The computation of *valid* confidence intervals requires the classifier outputs to be distributed according to a sufficiently selected distribution. If this assumption is violated, the confidence interval itself is not valid and thus, the calibrated probability can also be biased. On the contrary, if these assumptions are valid, they can directly be used to compute a calibrated probability.

Bin-based Ensemble Methods

All of the presented techniques construct a single calibration model. A more recent research line [Naeini 2016; Naeini et al. 2015a; Naeini & Cooper 2015, 2016, 2018; Naeini et al. 2014, 2015b] extended this by developing several bin-based techniques that are based on ensembles of different, individual ones. Clearly, this requires to define a set of possible models as well as a criterion how to assess their predictive performances. The former is accomplished by constraining an arbitrary binning model such that the bin borders only lie in the set of all predictions $\{\mathsf{f}(\mathsf{x}_i) : i = 1, \ldots, r\}$. Formally, this yields to defining[2] a binning model as a tupel $M = (B, \Theta)$, where $B = \{(z_0, z_1], (z_1, z_2], \ldots, (z_{m-1}, z_m]\}$ is the set of bins and $\Theta = (\theta_1, \theta_2, \ldots, \theta_m)$ their respective parameters such that the posterior probability $P(\mathsf{y} = 1 \mid \mathsf{f}(\mathsf{x}) \in b_i)$ of the i-th bin $b_i = (z_{i-1}, z_i]$ follows a binomial distribution parametrized by θ_i.

Furthermore, a given binning model M is scored by the posterior probability $P(M \mid D)$, which is proportional to $P(D \mid M) \cdot P(M)$. The likelihood $P(D \mid M)$ implicitly depends on the parameterization θ and thus, is in fact a marginal likelihood. Marginalizing out yields

$$P(D \mid M) \;=\; \int_{\Theta} P(D \mid M, \theta) \cdot P(\theta \mid M) \,\mathrm{d}\theta, \tag{2.9}$$

which has a closed-form solution if all data are independently sampled from the same distribution, the bin-wise class posterior distributions are binomially distributed and pairwise independent to each other as well as that the data-independent prior distributions $P(\theta \mid M)$ follow a beta distribution parameterized by α and β [Heckerman et al. 1995; Naeini et al. 2015a, 2014, 2015b].

If the parameters α and β as well as the prior distribution $P(M)$ are chosen, a binning model can be respectively scored. The currently introduced techniques differ in how these degrees of freedom are defined. In the first works [Naeini et al. 2014, 2015b], α and β are both set to one (i.e. $P(\theta \mid M) \equiv 1$) such that the closed-form of the likelihood takes the form

$$P(D \mid M) \;=\; \prod_{b \in B} \frac{r_0(b)! \cdot r_1(b)!}{(r(b) + 1)!} \tag{2.10}$$

where $r_0(b)$ and $r_1(b)$ denote the number of samples inside bin b with class 0 or 1, respectively, as well as $r(b) = r_0(b) + r_1(b)$ their sum. The prior distribution $P(M)$ is constructed such that the probability of creating a bin border at z is modeled using a Poisson distribution. In particular, without loss of generality the samples are increasingly sorted, i.e. $\mathsf{f}(\mathsf{x}_i) \leq \mathsf{f}(\mathsf{x}_j)$ holds for all i and j with $i < j$, such that the probability $q(i)$ of creating a bin border at $\mathsf{f}(\mathsf{x}_i)$ follows a Poisson distribution

[2] For completeness it should be added that the definitions in the original works differed slightly from this definition by using three and four elements, respectively.

parameterized by λ

$$q(i) \;=\; 1 - \exp\left(-\lambda \cdot \frac{\mathsf{f}(\mathsf{x}_{i+1}) - \mathsf{f}(\mathsf{x}_i)}{\mathsf{f}(\mathsf{x}_1) - \mathsf{f}(\mathsf{x}_r)}\right) \tag{2.11}$$

for all inner indices $i = 2, \ldots, r-1$, as well as $q(1) = q(r) = 1$ forcing a bin boundary at the extreme points. Further, let $\ell(b)$ and $u(b)$ denote the lower and upper bound indices, respectively, i.e. $b = (\mathsf{f}(\mathsf{x}_{\ell(b)}),\ \mathsf{f}(\mathsf{x}_{u(b)})]$. Under the assumption of independence between the different possible boundaries, the binning model's prior probability takes the form of

$$P(M) \;=\; \prod_{b \in B} q(u(b)) \cdot \prod_{j=\ell(b)}^{u(b)-1} (1 - q(j)), \tag{2.12}$$

which yields the overall binning score as:

$$P(D \mid M) \cdot P(M) \;=\; \prod_{b \in B} \left(\frac{r_0(b)! \cdot r_1(b)!}{(r(b)+1)!} \cdot q(u(b)) \cdot \prod_{j=\ell(b)}^{u(b)-1} (1 - q(j)) \right) \tag{2.13}$$

Equation (2.13) can be used to select the optimal binning model M_0 to estimate the posterior probability $P(\mathsf{y} = 1 \mid \mathsf{x}) \approx p(\mathsf{f}(\mathsf{x}); M_0)$ in the same way as the previously presented binning model. Consequently, this calibration technique is called *selection over Bayesian binnings* (SBB) [Naeini et al. 2014, 2015b]. Since the number of different possible models is exponential in r, a dynamic programming procedure has been presented by the authors in advance. Still, it has complexity $\mathcal{O}(r^2)$.

In a slightly modified way, the same approach can also be used to compute a weighted average over all t different binning models, where the weighting is performed by the respective score $P(D \mid M) \cdot P(M)$:

$$P(\mathsf{y} = 1 \mid \mathsf{f}(\mathsf{x})) \;=\; \sum_{i=1}^{t} \frac{P(D \mid M_i) \cdot P(M_i)}{\sum_{j=1}^{t} P(D \mid M_j) \cdot P(M_j)} \cdot P(\mathsf{y} = 1 \mid \mathsf{f}(\mathsf{x}), M_i) \tag{2.14}$$

Here, $P(\mathsf{y} = 1 \mid \mathsf{f}(\mathsf{x}), M_i)$ simply refers to the i-th binning model's posterior probability and yields the calibration technique *averaging over Bayesian binnings* (ABB). The result can also be computed using dynamic programming techniques that require a runtime of $\mathcal{O}(r^2)$. As the dynamic programming technique depends on the instance x whose posterior class probabilities are to be estimated, this runtime is required during each prediction in both SBB and ABB. The authors discuss possibilities to alleviate this problem by binning the unit interval into a fixed number of bins and to compute the ABB predictions only for these, which exposes the same problem as binning itself and thus, the problems why SBB and ABB were introduced at all.

Instead of fixing α and β to one, essentially the same approach has also been applied with bin-specific parameter values α_b and β_b as well as a uniform prior $P(M)$ over the models [Naeini et al. 2015a]. In particular, the b-th bin's parameters are set to $\alpha_b = \frac{2}{m} \cdot c_b$ and $\beta_b = \frac{2}{m} \cdot (1 - c_b)$, respectively, where c_b is the bin's center point. Under this assumptions, the likelihood and thus the binning score itself takes the form of

$$P(D \mid M) \cdot P(M) \;=\; \prod_{b \in B} \frac{\Gamma(\frac{2}{m})}{\Gamma(r(b) + \frac{2}{m})} \cdot \frac{\Gamma(r_1(b) + \alpha_b)}{\Gamma(\alpha_b)} \cdot \frac{\Gamma(r_0(b) + \beta_b)}{\Gamma(\beta_b)} \tag{2.15}$$

where Γ refers to the Gamma function. The overall posterior probabilities $p(\mathsf{f}(x))$ are obtained in the same way as in case of ABB in (2.14), the only difference lies in

the way how the scoring is performed, yielding the calibration technique *Bayesian binning into quantiles* (BBQ).

Similarly to the single histogram's bin size in regular histogram binning, the number of models and their bin sizes have to be selected. The authors present some heuristic strategies for doing so, but especially introduced a further improvement of BBQ such that the bin boundaries as well as the number of models can be computed from an optimization problem [Naeini & Cooper 2015, 2018]. In fact, isotonic regression is recovered for $\lambda \to \infty$ in the following, generalized optimization problem

$$\rho_\lambda^* = \underset{\rho \in \mathbb{R}^r}{\arg\min} \sum_{i=1}^{r} (\rho_i - \mathsf{y}_i)^2 + \lambda \sum_{i=1}^{r-1} (\rho_i - \rho_{i+1}) \cdot \mathbb{1}(\rho_i > \rho_{i+1}) \tag{2.16}$$

if the respective solution is interpreted as a piecewise constant prediction mapping $\rho : \mathbb{R} \to [0, 1]$ that maps $\mathsf{f}(\mathsf{x})$ to the respective bin's i_0 probability ρ_{i_0}. Interestingly, even the whole solution path can efficiently be computed using the modified pair-adjacent violators algorithm, such that different binning models $M_1, M_2, \dots M_t$ together with the respective values $\lambda_1, \lambda_2, \dots, \lambda_t$ are estimated. These are similarly combined using the *Bayesian information criterion* (BIC) as before to yield the overall posterior probabilities as

$$P(\mathsf{y} = 1 \mid \mathsf{f}(\mathsf{x})) \;=\; \sum_{i=1}^{t} \frac{\mathsf{BIC}(M_i)}{\sum_{j=1}^{t} \mathsf{BIC}(M_j)} \cdot P(\mathsf{y} = 1 \mid \mathsf{f}(\mathsf{x}), M_i) \tag{2.17}$$

which is exactly the same as the combination in BBQ and ABB in (2.14), respectively, the only difference is that the BIC scoring is used instead. As the posterior probabilities are computed using an *ensemble of near isotonic regression* models, this calibration technique is called ENIR.

The previously presented technique ENIR and its predecessors BBQ, ABB and SBB as well as binning and isotonic regression all compute piecewise constant and thus discontinuous transformation mappings. This drawback is improved in a further extension called *ensemble of linear trend estimation* (ELiTE) [Naeini & Cooper 2016, 2018]. Similar to the optimization problem involved in ENIR to compute an ensemble of piecewise constant calibration functions, ELiTE solves the following optimization problem

$$\rho_\lambda^* = \underset{\rho \in \mathbb{R}^r}{\arg\min} \; \frac{1}{2} \cdot \sum_{i=1}^{r} (\rho_i - \mathsf{y}_i)^2 + \lambda \cdot \sum_{i=1}^{r-2} \left| \frac{\rho_{i+2} - \rho_{i+1}}{\mathsf{y}_{i+2} - \mathsf{y}_{i+1}} - \frac{\rho_{i+1} - \rho_i}{\mathsf{y}_{i+1} - \mathsf{y}_i} \right| \tag{2.18}$$

where $\mathsf{f}(\mathsf{x})$ is constrained to lie in the unit interval $[0, 1]$, and otherwise requires an according preprocessing. It can be shown that for each value of λ, the resulting solution defines a continuous, piecewise linear function whose non-differentiable points (i.e. these where the slope changes) lie into the set of training data points $\{\mathsf{f}(\mathsf{x}_1), \dots, \mathsf{f}(\mathsf{x}_r)\}$.

Similarly to ENIR, the optimization problem is solved simultaneously for different values of the regularization parameter λ such that different calibration models $M_1, \dots, M_t$ are obtained and combined at prediction time. However, instead of the BIC scoring function used at ENIR, the *corrected Akaike information criterion* (AICc) is applied

$$P(\mathsf{y} = 1 \mid \mathsf{f}(\mathsf{x})) \;=\; \sum_{i=1}^{t} \frac{\mathsf{AICc}(M_i)}{\sum_{j=1}^{t} \mathsf{AICc}(M_j)} \cdot P(\mathsf{y} = 1 \mid \mathsf{f}(\mathsf{x}), M_i) \tag{2.19}$$

to combine the individual models. Finally, it is important to emphasize that the sorting of the instances by their predictions is crucial for the construction of the optimization problems (2.16) and (2.18) since they do not depend on the predictions despite their ordering.

Extended Parametric Approaches

All of the presented, more recently introduced techniques are model-free, as they do not assume a certain parametric relationship between predictions f and posterior probabilities $P(\mathsf{y} \mid \mathsf{f})$. The only exception is Platt scaling and, at least to some degree, isotonic regression because it assumes a monotonic relationship. However besides these, also two other parametric calibration techniques exist.

Platt scaling assumes a real-valued prediction function $\mathsf{f} : \mathcal{X} \to \mathbb{R}$ that is analytically transformed into a closed-form posterior probability. Despite that any binary classifier can be interpreted as a real-valued decision function and thus, this setting can be assumed as the most general one in binary classification, it still remains unreasonable to apply calibration techniques that are designed for unbounded, real-valued predictions on probabilistic classifiers.

A relatively recently introduced technique for probabilistic classifiers is beta calibration [Kull et al. 2017]. The authors show that Platt scaling is provably optimal for Gaussian-distributed class-conditional likelihoods $p(\mathsf{f} \mid \mathsf{y})$ with equal variance, which is an inappropriate model for probabilistic classifiers. Instead, they motivate to model the likelihoods using beta distributions

$$p(\mathsf{f}(\mathsf{x}) \mid \mathsf{y} = i) \;=\; \frac{(\mathsf{f}(\mathsf{x}))^{u_i - 1} \cdot (1 - \mathsf{f}(\mathsf{x}))^{v_i - 1}}{B(u_i, v_i)}, \tag{2.20}$$

parameterized by real-valued parameters $u_i, v_i > 0$ for each class $i \in \mathcal{Y}$ and the normalizing beta function B. Substituting the likelihoods from (2.20) into Bayes' theorem yields the posterior probabilities as:

$$P(\mathsf{y} = 1 \mid \mathsf{f}(\mathsf{x})) \;=\; \tau_{a,b}(\mathsf{f}(\mathsf{x})) \;=\; \left(1 + \frac{1}{\exp(b) \cdot \frac{(\mathsf{f}(\mathsf{x}))^a}{(1 - \mathsf{f}(\mathsf{x}))^a}}\right)^{-1} \tag{2.21}$$

Thus, the two parameters a and b are estimated and the posterior estimation at prediction time is straightforward.

Since the multinomial distribution generalizes the binomial one analogous to the way how the Dirichlet distribution generalizes the beta distribution for $k > 2$ outcome states, also the Dirichlet distribution has been applied to introduce Dirichlet calibration [Gebel 2009]. Consequently, Dirichlet calibration is designed for multi-class classifiers predicting posterior class distributions, i.e. non-negative vectors that sum to one, over $k > 2$ classes. To apply Dirichlet calibration for general, non-probabilistic multi-class classifiers $\mathsf{f} : \mathcal{X} \to \mathbb{R}^k$, a preprocessing with the softmax function

$$\sigma_{\text{softmax}} \;:\; \mathbb{R}^k \to (0,1)^k,\; z \mapsto \left(\frac{\exp(z_1)}{\sum_{i=1}^k \exp(z_i)},\; \ldots,\; \frac{\exp(z_k)}{\sum_{i=1}^k \exp(z_i)}\right)^\top \tag{2.22}$$

can be applied. Dirichlet calibration transforms Dirichlet-distributed, k-dimensional predictions f into a new Dirichlet-distributed random variable whose expected value equals the prior probabilities $P(\mathsf{y})$ estimated on the training data. In particular, a series a of different transformations is applied to move the distribution's expected

value accordingly. However, since the binary case has much more relevance for the remaining work, Dirichlet calibration is mainly mentioned for completeness. Still, it is interesting to note that Dirichlet calibration has been introduced roughly a decade before beta calibration.

2.2.3 Existing Results and Open Issues

The aim of this thesis is to apply classifier calibration in an extended variant of decomposition-based classification to realize a computationally feasible approach to dynamic classification. Therefore, evaluating calibration as part of decomposition-based strategies is particularly relevant. However, corresponding reference results do not directly focus on the actual calibration but only do so implicitly because the overall evaluation metrics depend on the calibrated probabilities of the involved base classifiers. Further details will be presented in subsection 2.3.3 after presenting the corresponding decomposition-based approaches in subsections 2.3.1 and 2.3.2, respectively. Therefore, at first the focus lies on the results of the calibration techniques themselves, which are relevant as they are the basis of all approaches that will be developed in chapters 4 and 5.

Here, it is interesting to observe that classifier calibration is still relatively rarely studied in the data mining and machine learning literature. The three first techniques Platt scaling, binning and isotonic regression are at least mentioned in almost any work on classifier calibration, while the more recently introduced approaches in most cases are only applied in the introducing or follow-up works of the same authors. Consequently, there are more results available for the three standard techniques than for the remaining, more recent ones.

Binning

First, histogram binning is model-free and non-monotonic in general. Thus, it does not assume a fixed parametric model, but this also includes certain drawbacks. The number of bins has to be selected, which remains arbitrary as the optimal number of bins is unclear [Connolly et al. 2017; Jiang et al. 2012; Naeini & Cooper 2015, 2016, 2018; Naeini et al. 2015b]. This can be alleviated by applying cross validation techniques to estimate the bin count [Zadrozny & Elkan 2002], however this requires to score a binning model [Naeini et al. 2015a, 2014, 2015b], which by itself is complicated since evaluating a calibration model is.

Even if the number of bins has been estimated, the size of each as well as their positions and thus, the breaks between them remain arbitrary and fixed [Zadrozny & Elkan 2002]. As a consequence, even instances might be forced to the same value whose probability should better be different [Zadrozny & Elkan 2002], i.e. nothing besides the empirical fractions of samples remains from the respective distributions [Jiang et al. 2012]. Even after the binning model has been computed, it defines a piecewise constant, discontinuous transformation, which can be counterintuitive as often continuous modelings are preferred [Eck et al. 2017]. Furthermore, the number of different calibrated probabilities is bounded by the number of bins [Zadrozny & Elkan 2001a], which itself should not be too high to control the variance [Bella et al. 2009a] and is often arbitrarily fixed to ten.

Besides these relatively strong disadvantages, binning is still efficient [Bella et al. 2013], easy to implement as well as applied without handling its drawbacks in other works [Bella et al. 2009b; Drish 2001; Guo et al. 2017; Kruppa et al. 2014a]. Furthermore, it can be applied for multi-class predictors $\mathsf{f} : \mathcal{X} \to \mathbb{R}^k$ as well, however

as soon as k is not relatively small, this is prone to the same problems as general density estimation.

Isotonic Regression

In this regard, there exists accordance that the construction of the bins themselves is the major drawback of binning. This also justifies to replace the arbitrary bin selection with a monotonicity constraint yielding isotonic regression and giving rise to the question whether it is reasonable to construct binary calibration techniques monotonically.

Besides this, the isotonic regression transformation mapping can still be efficiently computed but remains discontinuous and piecewise constant. Evaluations report good or at least comparable results [Bella et al. 2009b; Fonseca & Lopes 2017; Jiang et al. 2012; Naeini et al. 2015a, 2014; Niculescu-Mizil & Caruana 2005a,b; Zadrozny & Elkan 2002], while other works also mention (highly) varying [Gebel 2009; Naeini & Cooper 2015, 2016, 2018] to worse results [Bequé et al. 2017; Likhomanenko et al. 2016] in comparison with other calibration techniques.

Especially overfitting is a present problem in isotonic regression (or more precisely in the PAV algorithm used to compute the prediction model) [Bella et al. 2009a, 2013; Xu et al. 2016], but if many data are available, it is still possible to obtain better results with it [Bennett 2006]. Some of the authors of the aforementioned, recent results even state that isotonic regression is the "most commonly used non-parametric classifier calibration method" in the same way as they criticize the problematic monotonic assumption [Naeini & Cooper 2015, 2016, 2018]. However, this at least emphasizes the relevance of isotonic regression to date at classifier calibration in practice.

Platt Scaling

It is an interesting fact that the monotonic assumption is also one of the biggest criticisms brought against Platt scaling, which is even more controversially discussed in the data mining and machine learning literature. Usually good results are reported [Connolly et al. 2017; Platt 1999; Xu et al. 2016], while other authors even present it as a standard technique for probabilistic support vector predictions [Cai et al. 2016; Flach 2016] and "as the most highly approved approach" [Gebel 2009].

Clearly, these results directly depend on the underlying model assumptions. Since Platt scaling assumes a sigmoid-shaped relationship between predictions and probabilities, its predictive performance depends on the validity of this assumption [Bella et al. 2009a; Niculescu-Mizil & Caruana 2005a,b], which does not always hold. Interestingly, it has been empirically observed to hold for boosted decision trees and for support vector machines, while other classifiers like the naive Bayes showed non-sigmoid-shaped relations [Kull et al. 2017; Niculescu-Mizil & Caruana 2005a,b; Xu et al. 2016; Zadrozny & Elkan 2002].

Yet, there are also different opinions. First, the monotonic assumption can be generally criticized, as previously mentioned. But since Platt scaling is even strictly monotonic, this discussion is slightly more focused on Platt scaling than on other approaches. Furthermore, the sigmoid-shaped assumption of the transformation can be criticized as too restrictive in general [Grandvalet et al. 2005; Jiang et al. 2012; Kruppa et al. 2014a; Naeini 2016; Naeini et al. 2015a; Naeini & Cooper 2015, 2016, 2018; Naeini et al. 2014, 2015b], which especially can bias probability estimates whose predictions are close to the decision boundary [Wang et al. 2019]. In other works [Franc et al. 2011; Wang et al. 2008] the good empirical results are emphasized in the same way as the lack of statistical evidence explaining or justifying it.

While some of these claims are sometimes neither proven nor empirically validated, a more recent work [Kull et al. 2017] at least tries to explain why Platt scaling can fail. A quantity used in the respective analysis is the likelihood ratio $\frac{p(\mathsf{f}(x)|\mathsf{y}=1)}{p(\mathsf{f}(x)|\mathsf{y}=-1)}$, which in the binary case is, together with the class priors, sufficient to express the posterior probabilities. The authors show that if Gaussian distributions with equal variance are assumed for the two likelihoods $p(\mathsf{f}(\mathsf{x}) \mid \mathsf{y} = i)$, their ratio simplifies to $\exp(a \cdot \mathsf{f}(\mathsf{x}) + b)$, yielding a closed-form posterior equation of the form (2.7). On the reverse, they show that for a given posterior distribution in form of (2.7), it is straightforward to construct corresponding equal-variance Gaussian-distributed likelihoods. Moreover, it is also important to emphasize that equal-variance Gaussian-distributed likelihoods are not an assumption of Platt scaling. Even Platt himself observed explicitly non-Gaussian distributions in his introductory work [Platt 1999]. Furthermore, he showed that Platt scaling is not valid for general Gaussian-distributed likelihoods with non-equal variances. However, there are no results available that clarify its true parametric assumptions.

Because any family of likelihood distributions that is valid for Platt scaling especially is valid for a monotonic calibration mapping, these two questions are strongly related. Here, different relatively recent works even base their criticisms on wrong statements [Bella et al. 2013; Naeini 2016; Naeini & Cooper 2015, 2016, 2018] that will be discussed and corrected in subsection 3.1.2. Some of these works explicitly exclude Platt scaling from their evaluations because it should perform inferior to isotonic regression and BBQ [Naeini & Cooper 2015, 2018].

Comparative Studies

Presumably the most important question related to classifier calibration, at least from the practical point of view, is which one out of the set of different techniques should be used or preferred in practice. This point is strongly related to the question which one is actually the *best* possible option, but not necessarily equivalent as different factors like computational effort and efficiency might also be relevant in practice and might be traded-off against the bare quality of the results.

Even though there are various comparisons of the three standard techniques in the aforementioned works, unluckily there are much less results for the more recent ones. Usually, the respective introductory works contain evaluations and discussions about the results, but general comparative studies are rare. Techniques like ACP, SBA or asymmetric distribution-based ones are not applied in more recent works, SBB and ABB are mentioned but not applied in the follow-up works introducing BBQ, ENIR and ELiTE even by the same authors due to their impractical runtime requirements[3]. Possible reasons might be the lack of confidence intervals that make the application of ACP relatively restricted, nearest-neighbor searches can be too unreliable in general to apply SBA in practice or asymmetric distributions might be too task-specific.

On the other hand, there are a few promising techniques like beta calibration, ENIR and ELiTE that simply might be too new to the communities to already gain much further interest, especially since classifier calibration is rarely studied in general. BBQ showed improved accuracy in comparison with the three standard techniques that were both improved by ENIR and ELiTE on mostly the same benchmark data sets [Naeini et al. 2015a; Naeini & Cooper 2015, 2016, 2018]. The differences between ENIR and ELiTE are relatively small, while an approximately eight times larger computational time of ELiTE is reported.

[3]It should be added that the authors at least state that ABB performs comparable to BBQ on smaller data sets [Naeini & Cooper 2015, 2016, 2018].

Furthermore, there are a few comparative works that apply one of the more recently introduced techniques. In one particular example, BBQ is applied in a medical context [Connolly et al. 2017] as well as a different work [Wang et al. 2019] applies the three standard techniques as well as ACP, BBQ, ENIR and ELiTE. The latter work additionally analyzes their presented extension to isotonic regression as well as the existing one based on spline interpolation [Jiang et al. 2011] besides single models from the ENIR and ELiTE ensembles. Their evaluations were performed on one artificial as well as two real-world data sets and showed that their introduced extension to isotonic regression outperformed all other approaches, while BBQ performed superior to ENIR and ELiTE. However, it should be emphasized that the authors used a relatively unusual evaluation strategy. Instead of cross-validating the whole data sets, they randomly generated 100 training instances per class in case of the artificially data set and randomly selected 200 or 500 samples from 45000 overall ones in case of the real-world data sets. Thereafter, they generated $2 \cdot 200000$ test instances or used all remaining ones as test data, respectively. Using only about one percent of the overall data for training is hardly reasonable at all. A potential explanation might be that they iterated this procedure 50 times on the real-world data sets and kept the training data sizes small to maintain a feasible runtime of the training procedures.

However, none of these studies applied beta calibration such that the only existing comparative study is the one found in the introductory work of if. In particular, the authors showed that beta calibration outperformed Platt scaling and isotonic regression in a comparison based on 41 data sets, while the other, non-parametric techniques were not applied. Furthermore, they used probabilistic classifiers, which strongly conflicts with Platt scaling's assumption of a real-valued prediction function. Thus, it is questionable if this comparison is really representative for real-world applications at all.

A general observation in all previously mentioned, existing studies is that they either focus on carefully analyzing a few, potentially large data sets only [Guo et al. 2017; Naeini et al. 2014, 2015b; Wang et al. 2019; Zadrozny & Elkan 2001b, 2002] or if they analyze 10 to 20 or more data sets, these are often relatively small. In respective reference works [Bella et al. 2013, 2009b; Gebel 2009; Kull et al. 2017; Naeini et al. 2015a; Naeini & Cooper 2015, 2016, 2018], most data sets consist only of a few one hundred instances, while only some of them contain more than 10000 instances. A comparative study that analyzes many large-scale data sets unluckily is not available at all and the main reason for the comprehensive empirical part in the following chapter 3.

Still, there are also a few more interesting issues that motivate a representative study on large data sets. The first one refers to possible evaluation metrics. It is interesting to observe that there is no well-accepted standard for classifier calibration error metrics. Earlier works as the ones introducing binning, Platt scaling and isotonic regression evaluated the probabilistic predictions using the Brier score and log-loss, while in the recent studies, bin-based evaluation metrics gained popularity instead. Since the true probabilities cannot be used as a reference, this is a highly non-trivial question. In full detail this will be reviewed in section 3.3.

Fitted or Predicted Calibration?

A further, less obvious issue refers to the way how classifier calibration should correctly be applied. The calibration function has to be optimized based on a data set $\{(\mathsf{f}_1, \mathsf{y}_1), (\mathsf{f}_2, \mathsf{y}_2), \ldots, (\mathsf{f}_r, \mathsf{y}_r)\}$ consisting of predictions $\mathsf{f}_i = \mathsf{f}(\mathsf{x}_i)$ and true class values y_i. To generate the predictions f_i, the prediction function has to be trained at first.

Thereafter, it can be used to predict the instances x_i. However, if the *same* samples x_i were already used to generate f, the predictions form a biased sample of the score distribution on independent data. Reusing the training data's fitted values directly is referred to as *fitted calibration*, while alternatively using an independent hold-out set (for example by applying cross validation) is referred to as *predicted calibration* throughout this work.

Both approaches are commonly used in reference studies. One existing work states that generally for binning, predicted calibration should be preferred, but for naive Bayes classifiers it is not required [Zadrozny & Elkan 2001b] and thus not applied. In a different work [Zadrozny & Elkan 2002] the same authors apply fitted calibration as well since this should be valid whenever the classifier does not overfit its training data. For support vector machines, predicted calibration should be preferred, at least for non-linear kernels [Platt 1999]. Following these reasoning, other empirical evaluations [Bella et al. 2013; Kull et al. 2017; Naeini et al. 2014; Niculescu-Mizil & Caruana 2005a,b] apply predicted calibration where a single work [Naeini et al. 2014] states that based on the performed experiments, this should not be necessary at all. Similarly, in different relatively recent publications predicted calibration is applied [Bequé et al. 2017; Fonseca & Lopes 2017] and sometimes even formulated as an assumption [Connolly et al. 2017; Flach 2016; Guo et al. 2017; Leathart et al. 2017]. Moreover, it is emphasized that the additional data set can be reused for parameter optimization [Guo et al. 2017], which was also noted earlier by different authors [Niculescu-Mizil & Caruana 2005b].

In contrast to this, there are also particularly recent works of different authors that prefer fitted calibration. The whole series of works introducing BBQ, ENIR and ELiTE only apply fitted calibration, the same holds for the recently presented extension to isotonic regression [Wang et al. 2019]. However, there are some additional works that at least explicitly mention both strategies as possible options [Bella et al. 2009a], but only two also apply both and compare them. The first one [Bella et al. 2009b] applies four different calibration techniques on 20 relatively small data sets with at maximum 8124 instances, while 15 of them consist of less than 1000. Additionally, a second work [Drish 2001] compares both for support vector machines using binning calibration on a single but challenging classification task.

In summary, a comprehensive in-depth comparison of both approaches is a mainly unanswered question and also has been explicitly formulated as an open research question [Naeini 2016]. Therefore, it will be analyzed as well in the following chapter 3. In fact, this is a special case of the lack of general guidelines on how to ideally reuse data for different stages of learning, validation and model selection [Duin & Pekalska 2005].

2.3 Decomposition-based Classification

The previous section 2.2 presented different classifier calibration methods and summarized the respective current research results. As the main intention in applying calibration is its application in combination with decomposition-based classification, the next section summarizes the respective research of decomposition- or reduction-based classification.

Here, the key idea is to reduce the task of solving a k-class classification problem by computing a classifier $\mathsf{f} : \mathcal{X} \to \mathcal{Y}$ to a set of multiple, but simpler (usually binary) classification problems that can be solved independently. In case of the reduction to

binary problems only, the respective approaches are also known as *class binarization techniques*.

Historically, these have a strong relation to the demand to apply binary-only classification algrorithms like the support vector machine or AdaBoost on multi-class problems, where a direct application of them – in contrast to alternatives like decision trees, random forests or neural networks – is impossible. Still throughout this work, these algorithmic techniques are explicitly discussed per se without aiming or restricting to particular classification algorithms. In the following chapters 4 and 5, the existing decomposition-based strategies will be combined with calibration techniques to integrate the dynamic class information. As a basis for any reduction, the training data are assumed as $D = \{(\mathsf{x}_i, \mathsf{y}_i) \ : \ i = 1, \ldots, r\} \subset \mathcal{X} \times \mathcal{Y}$ where $\mathcal{Y} = \{1, 2, \ldots k\}$ refers to the set of k classes.

2.3.1 Standard Decompositions

The existing decompositions differ in the way how they, on the one hand, create a set of binary classification problems from the given task, and, on the other hand, how the respective individual predictions are combined. Consequently, they offer different advantages in the same way as they expose different drawbacks.

One-vs-All Decomposition

A straightforward way to decompose the problem is to construct k independent binary classification problems, where each class i (as positive class) is separated from the set of remaining classes $\{1, ..., i-1, i+1, ...k\}$ (as negative class). This approach is known as *one-vs-all* multi-class decomposition [Crammer & Singer 2001; Doğan et al. 2016; Lee et al. 2004; Maass 2000; Rifkin & Klautau 2004; Rifkin 2002] and probably the first technique that has been used to construct multi-class support vector machines [Lei & Govindaraju 2005]. Alternative names are *one-of-k* or *one-hot encoding* as well as *winner-takes-all* scheme.

After training, there are k classifiers f_i available, and to predict a newly observed input sample x, most commonly the overall prediction $\mathsf{f}(\mathsf{x})$ is computed as $\mathsf{f}(\mathsf{x}) \coloneqq \arg\max_i \mathsf{f}_i(\mathsf{x})$. The rationale behind this prediction approach is relatively straightforward. Ideally, only the individual classifier corresponding to the true but unknown class y outputs a positive, while all others output a negative prediction. Still, the maximum rule also enables to resolve conflicts. This could be tackled by using more sophisticated approaches than to simply find the maximum value [Galar et al. 2011], however is still the accepted or predominant standard.

To obtain a probabilistic interpretation, the one-vs-all decision functions can be combined with a softmax transformation (2.22), which can also be used to train the decision functions simultaneously as commonly done at neural network training.

One-vs-One Decomposition

The second popular class binarization technique besides the one-vs-all decomposition is the *one-vs-one* reduction. Here, for each pair of classes the task of separating i and j is formulated, resulting in $\binom{k}{2} = \frac{k \cdot (k-1)}{2}$ individual classifiers $\mathsf{f}_{i,j}$, $1 \leq i < j \leq k$. Consequently, to predict a newly observed instance, there are as many individual predictions $\mathsf{f}_{i,j}(\mathsf{x})$. Combining them into an overall prediction is a less obvious task as in the one-vs-all case. A default option that does not require anything else besides the pairwise binary predictions is to perform a voting, where each individual prediction is interpreted as a vote for the respective class [Alam et al. 2003; Angulo & Català

2000; Fernandez et al. 2015; Friedman 1996; Fürnkranz 2002a,b, 2003; Jelonek & Stefanowski 1998; Moreira & Mayoraz 1998; Ou & Murphey 2007; Sáez et al. 2014; Weston & Watkins 1998]. Consequently, the overall prediction is obtained as the class receiving the maximum number of votes. Normalizing the votes vector additionally allows the interpretation as a posterior probability estimate, even though this is highly likely to be miscalibrated. However, it is possible that a prediction cannot be uniquely resolved if an instance receives an equal number of votes for different classes. The respective parts of the feature space are known as *unclassifiable regions* and require extended tie-breaking techniques [Liu et al. 2008, 2006; Qin et al. 2017; Wu et al. 2014]. Another alternative is to arrange the binary predictors in a tree-based structure, whose main advantage besides avoiding of ties is that not always all binary predictions have to be computed [Platt et al. 1999; Rahman & Fairhurst 1997].

More sophisticated approaches besides binary voting exist, however in the vast majority these require a probabilistic output $\phi_{i,j}(\mathsf{x}) \in [0,1]$ of $\mathsf{f}_{i,j}$ that is interpreted as an estimator of the posterior probability $\phi_{i,j}(\mathsf{x}) \approx P(\mathsf{y} = i \mid \mathsf{x},\ \mathsf{y} \in \{i,j\})$. Estimating these is a challenging problem, as discussed before, and their demand in decomposition-based classification approaches shows the strong connection between classifier calibration and decomposition-based classification. Using any of the methods from section 2.2 or chapter 3, all pairwise predictions are assumed as calibrated probabilities satisfying $\phi_{i,j}(\mathsf{x}) + \phi_{j,i}(\mathsf{x}) = 1$ for all $1 \leq i, j \leq k$ with $i \neq j$. Consequently, they can be arranged into a pairwise probabilities matrix

$$\Phi = \begin{pmatrix} \bullet & \phi_{1,2} & \phi_{1,3} & \cdots & \phi_{1,k-1} & \phi_{1,k} \\ \phi_{2,1} & \bullet & \phi_{2,3} & \cdots & \phi_{2,k-1} & \phi_{2,k} \\ \phi_{3,1} & \phi_{3,2} & \bullet & \cdots & \phi_{3,k-1} & \phi_{3,k} \\ \vdots & \vdots & \vdots & \ddots & \vdots & \vdots \\ \phi_{k-1,1} & \phi_{k-1,2} & \phi_{k-1,3} & \cdots & \bullet & \phi_{k-1,k} \\ \phi_{k,1} & \phi_{k,2} & \phi_{k,3} & \cdots & \phi_{k,k-1} & \bullet \end{pmatrix} \tag{2.23}$$

where the diagonal is completely irrelevant and two entries at transposed positions (i,j) and (j,i) always sum to one. Clearly, the matrix Φ depends on the input instance $\mathsf{x} \in \mathcal{X}$ and therefore can be interpreted as a function $\Phi \equiv \Phi(\mathsf{x})$. However, this dependency is not directly important for any algorithm processing it and usually omitted for increased readability. Still, it is important to emphasize that all techniques have to be applied at prediction time, i.e. *after* observing instance x such that real-time performance is usually required. The following *fusing*, *aggregation* or *combination* step refers to the task that transforms the set of pairwise probabilities into a k-class vector $p = p(\mathsf{x})$, which approximates the unknown posterior probabilities $P(\mathsf{y} \mid \mathsf{x})$. Even though there is no general agreement about this terminus in the literature, all respective approaches will be referred to as *pairwise coupling techniques* throughout this work.

Existing Pairwise Coupling Techniques

Generally, different techniques have appeared for this task besides the ones that will be presented in full detail in the following [Duan et al. 2003; Hastie & Tibshirani 1998; Krzyśko & Wołyński 2009; Moreira & Mayoraz 1998; Price et al. 1995]. All of them finally predict the class with maximum posterior probability, thus their differences only lie in the way how the binary probabilities are combined. Following existing

works [Galar et al. 2010, 2011, 2013, 2017, 2014, 2015; García-Pedrajas & Ortiz-Boyer 2011; Krzyśko & Wołyński 2009] that summarize and compare the different approaches with each other, the three ones presented in the following are particularly relevant [Galar et al. 2017].

The first option is a relatively straightforward generalization of the binary to a probabilistic voting (Vote) [Fürnkranz 2002a,b, 2003; Ou & Murphey 2007; Park & Fürnkranz 2007] such that each classifier simultaneously votes for its corresponding classes with the respective probabilities. Consequently, the accumulated predictions are obtained as

$$p_i^{\mathsf{Vote}}(\mathsf{x}) \;=\; \frac{2}{k \cdot (k-1)} \cdot \sum_{\substack{j=1 \\ j \neq i}}^{k} \phi_{i,j}(\mathsf{x}) \tag{2.24}$$

for each class $i = 1, \ldots, k$. Probabilistic voting can also be interpreted as computing the row sums in (2.23), where normalization is applied to allow a probabilistic interpretation of them. Furthermore, also variants motivated by extending weighting voting to ranking were presented [Hüllermeier & Brinker 2008; Hüllermeier & Vanderlooy 2010].

The second particularly relevant pairwise coupling approach is based on a *non-dominance criterion* (ND) [Fernández et al. 2010; Galar et al. 2010] in fuzzy preference relations. Using the pairwise probabilities, a strict preference relation with elements $\phi'_{i,j}(\mathsf{x}) = \max(\phi_{i,j}(\mathsf{x}) - \phi_{j,i}(\mathsf{x}), 0)$ for all $1 \leq i, j \leq k$ is computed that expresses a non-negative, pairwise preference. Thereafter, a non-dominance vector ND as well as the respective posterior probabilities estimate p^{ND} can be computed as

$$\mathsf{ND}_i(\mathsf{x}) \;=\; 1 - \max_{j \neq i} \phi'_{j,i}(\mathsf{x}) \quad \text{and} \quad p_i^{\mathsf{ND}}(\mathsf{x}) \;=\; \frac{\mathsf{ND}_i(\mathsf{x})}{\sum_{j=1}^{n} \mathsf{ND}_j(\mathsf{x})}, \tag{2.25}$$

respectively. It should be mentioned that the original works only compute the non-dominance vector, still the normalization is a straightforward extension for consistency with the other approaches. A slightly related idea to interpret a voting *against* the respective class already appeared earlier [Cutzu 2003] and can be interpreted as a connection between voting and the non-dominance approach.

Finally, the third important pairwise coupling technique (WLW) [Wu et al. 2004] computes the posterior probabilities by solving the following quadratic optimization problem:

$$\min_{p} \sum_{i=1}^{k} \sum_{\substack{j=1 \\ j \neq i}}^{k} (\phi_{j,i} \cdot p_i \;-\; \phi_{i,j} \cdot p_j)^2 \quad \text{s.t.} \; \sum_{i=1}^{k} p_i = 1 \tag{2.26}$$

The authors proved that problem (2.26) has a unique, non-negative solution $p^{\mathsf{WLW}}(\mathsf{x})$ that can be computed by solving an equivalent linear equation system if for all pairwise probabilities hold $\phi_{i,j}(\mathsf{x}) > 0$. Furthermore, they presented an iterative algorithm that globally converges to the optimal solution. Therefore, it can efficiently be applied in practice despite formulating a relatively complex optimization problem that has to be solved for each prediction. Additionally, the authors related their approach to equivalent reformulations of other techniques to emphasize differences and similarities between them.

2.3.2 Extended Decompositions

Besides the one-vs-all and one-vs-one decomposition, there is also a third family. Even though many works describe it as an alternative to the former two techniques

[Arruti et al. 2014; Bagheri et al. 2012; Chmielnicki 2015; Chmielnicki & Stąpor 2016; Galar et al. 2010; Garcia-Pedrajas & Ortiz-Boyer 2006; García-Pedrajas & Ortiz-Boyer 2008; Lorena et al. 2008; Mendialdua et al. 2015; Montañés et al. 2013; Rifkin & Klautau 2004; Rifkin 2002; Rocha & Goldenstein 2014; Wu et al. 2014], in fact it can be interpreted as a generalization [Escalera et al. 2010; Quost & Destercke 2018; Wang & Xue 2014] that is based on *encoding* and *decoding* the classes in the *error correcting output code* (ECOC) framework [Dietterich & Bakiri 1995; Kong & Dietterich 1995]. In particular, a decomposition consisting of ℓ binary reductions is represented using a ternary code word matrix W:

$$W = (W_{i,j})_{\substack{1 \leq i \leq k \\ 1 \leq j \leq \ell}} = \begin{cases} 1 & \text{class } i \text{ is mapped to 1 in problem } j \\ -1 & \text{class } i \text{ is mapped to } -1 \text{ in problem } j \\ 0 & \text{class } i \text{ is excluded from problem } j \end{cases} \tag{2.27}$$

Here, each column j defines an element of the decomposition that maps instances with class label i to the (i,j)-th entry in W. Therefore, the columns $W_{(\cdot,j)}$ can be interpreted as the j-th subproblem's mapping required at training time, while the i-th row $W_{(i,\cdot)}$ represents the *codeword* that *encodes* class i. Especially the introducing works restricted the codeword matrix to binary codewords in $\{-1,1\}^{k \times \ell}$ only that cannot represent all reductions. Without this restriction, any reduction to binary problems can be represented using an accordingly selected codeword matrix in the form of (2.27). For example, the one-vs-all decomposition corresponds to the codeword matrix

$$W_{\mathsf{1vA}} = \begin{pmatrix} 1 & -1 & -1 & \cdots & -1 & -1 \\ -1 & 1 & -1 & \cdots & -1 & -1 \\ -1 & -1 & 1 & \cdots & -1 & -1 \\ \vdots & \vdots & \vdots & \ddots & \vdots & \vdots \\ -1 & -1 & -1 & \cdots & 1 & -1 \\ -1 & -1 & -1 & \cdots & -1 & 1 \end{pmatrix} \tag{2.28}$$

while the one-vs-one reduction is defined by

$$W_{\mathsf{1v1}} = \left(\begin{array}{cccccc|cccc|c|c} 1 & 1 & 1 & \cdots & 1 & 1 & 0 & 0 & \cdots & 0 & \cdots & 0 \\ -1 & 0 & 0 & \cdots & 0 & 0 & 1 & 1 & \cdots & 1 & \cdots & 0 \\ 0 & -1 & 0 & \cdots & 0 & 0 & -1 & 0 & \cdots & 0 & \cdots & 0 \\ 0 & 0 & -1 & \cdots & 0 & 0 & 0 & -1 & \cdots & 0 & \cdots & 0 \\ \vdots & \vdots & \vdots & \ddots & \vdots & \vdots & \vdots & \vdots & \ddots & \vdots & \ddots & \vdots \\ 0 & 0 & 0 & \cdots & -1 & 0 & 0 & 0 & \cdots & 0 & \cdots & 1 \\ 0 & 0 & 0 & \cdots & 0 & -1 & 0 & 0 & \cdots & -1 & \cdots & -1 \\ \underbrace{}_{k-1} & & & & & & \underbrace{}_{k-2} & & & & & \underbrace{}_{1} \end{array}\right) \tag{2.29}$$

where each of the $\binom{k}{2}$ columns contains exactly a single 1 and -1, respectively. The one-vs-one codeword matrix consists of $k-1$ consecutive column blocks (indicated by vertical bars) that define the pairwise classifiers to separate class i from the other classes $j > i$. It could be added that reductions to non-binary problems can be represented by extending the codeword elements to arbitrary values in $\{0, 1, \ldots, k\}$. However, in practice the main aim is to reduce multi-class problems to binary ones such that these are of minor practical relevance.

To predict an instance x in the ECOC framework, all individual predictions $\mathsf{f}_j(\mathsf{x})$,

$j = 1, \ldots, \ell$, are computed and have to be combined into an overall decision. Classically, all classifier predictions are concatenated to form a predicted codeword[4] $w \in \{-1, 1\}^\ell$ that is compared to each of the k codeword rows in W to *decode* it into a class prediction. If there is a class that is encoded with w, this class is predicted. Still, the ECOC framework requires a similarity metric to resolve the general case that the predicted word w is not contained in the cordword matrix and has to be mapped to the most similar one. Traditionally, the *Hamming distance* $\mathsf{d}_{\mathrm{Ham}}(v, w) = |\{i : v_i \neq w_i\}|$ counting the number of different digits between its arguments is used. The rationale behind this approach is that the Hamming decoding can *correct* individual misclassifications as long as the codewords enconding the classes are sufficiently different.

As long as there is exactly a single positive prediction in a one-vs-all ensemble, the Hamming decoding coincides with the maximum's predictions index. Consequently, the one-vs-all reduction is a special case of the default binary ECOC approach if there are no conflicting individual predictions. Similarly, as long as the binary voting in a one-vs-one ensemble results in a single class receiving the maximum number of votes, it is equivalent to Hamming decoding in the ECOC framework [Fürnkranz 2002b]. However, resolving the class using Hamming decoding can result in unclassifiable regions and consequently requires tie-breaking techniques [Rätsch et al. 2002].

Other works deal with generalizing the ECOC framework's predicting step besides the Hamming decoding. Straightforward choices are other and particularly non-discrete loss functions that are used to compare the codewords besides the Hamming distance [James 1998; Kong & Dietterich 1995; Passerini et al. 2004], summarized as *loss-based decoding* [Allwein et al. 2000]. Still, interesting alternatives also cover approaches to estimate posterior probabilities by using an overconstrained linear equation system [Kong & Dietterich 1997] as well as ideas to generalize pairwise coupling for arbitrary codeword matrices [Zadrozny 2002]. Complex approaches even formulate the decoding step using regression algorithms [Chen et al. 2010].

All of these techniques are directly designed to target the ECOC framework. However, different approaches that are not directly introduced in this context can also be interpreted as a respective decoding of a predicted word. Particular examples are aforementioned tree-based approaches based on the one-vs-one reduction and extensions of these into recursive merging of classes [Lei & Govindaraju 2005; Lorena et al. 2008; Madzarov et al. 2009; Montañés et al. 2013; Yang et al. 2013; Zhao et al. 2016]. The same holds for any decomposition-based approach given both, an accordingly selected codeword matrix and a prediction function, respectively.

2.3.3 Existing Results

With respect to this thesis' main aim, the first important observation is that all techniques assume a fixed class set such that any approach realizing dynamic classification requires to extend existing algorithms. Besides this, the question arises which combination of decomposition and fusing technique should be preferred to maximize the classification accuracy or minimize the corresponding loss function, respectively.

Codeword Design

As the ECOC framework comprises all reductions, this question can be reformulated as to ask for good codeword design and fusing strategies. Here, an important property is a large minimum distance between the codewords, i.e. rows in W. If the minimum

[4] Here, a zero prediction is impossible because always all classifiers are used.

distance between rows (codewords) is $\mathsf{d}_{\min}$, then the Hamming decoding allows the correction of up to $\lfloor \frac{\mathsf{d}_{\min}-1}{2} \rfloor$ individual misclassifications [Kong & Dietterich 1995]. Here, a first possibility is to construct a *complete* binary codeword matrix [Allwein et al. 2000], containing all $2^{k-1}-1$ possible partitions covering all classes. In an extended way also all ternary codewords can be constructed [Gebel 2009], which yields $0.5 \cdot (3^k - 2^{k+1} + 1)$ columns in total. Both are exponential in k and thus are intractable for more than a few classes.

A computationally feasible alternative is to use randomization during codeword generation [Allwein et al. 2000]. In particular, either *dense random codes* randomly select each digit as ± 1 with probability 0.5. In contrast to this, *sparse random codes* select each digit as 0 with probability 0.5 and as ± 1 with equal probability of 0.25. Still, the number of digits (or equivalently, number of columns in the codeword matrix) remains as a degree of freedom. The introducing work constructed the dense and sparse random codes with $\lceil 10 \cdot \log_2(k) \rceil$ as well as $\lceil 15 \cdot \log_2(k) \rceil$ independently generated columns, respectively, and selected the best codeword matrix with respect to the largest minimum distance over 10000 iterations for aforementioned reasons.

However, selecting the best codeword length is a highly non-trivial problem. On the one hand, increasing the code length ℓ (i.e. number of classifiers) easily increases the minimum distance and consequently yields more robust codes. On the other hand, longer codes also result in more and potentially harder classification problems. Here, it has been proven in advance that finding a codeword matrix that minimizes the empirical error is NP-complete [Crammer & Singer 2002]. Still, well designed codes can improve the learning ability and the classification accuracy [Kong & Dietterich 1995, 1997].

Consequently, the loss-based ECOC framework requires to use some heuristics for the codeword generation. Besides aforementioned complete and random codes, different data-dependent strategies were introduced that use the given training data to create or improve codeword matrices. Particular examples cover *discriminant ECOC* [Pujol et al. 2006], *data-driven ECOC* (DECOC) [Zhou et al. 2008], *topology-preserving output codes* [Zhang et al. 2013] as well as explicit improving of random codes [Chmielnicki 2015].

Generally, the usefulness of ECOC decoding massively depends on the independence of the classifiers [Garcia-Pedrajas & Ortiz-Boyer 2006; Kong & Dietterich 1994]. In particular, different codewords enforce to learn parts of the same decision boundary multiple times [Kong & Dietterich 1995], but also are influenced by other classes that are differently labeled in the respective codewords. This induces redundancy, which at least partially explains why ECOC succeeds in improving the recognition accuracy. Besides the aforementioned intractability of complete codewords, empirical studies [García-Pedrajas & Ortiz-Boyer 2008, 2011] report similar results of dense and sparse codeword design strategies, where the former are more robust, while the latter are faster to train. Most interestingly, additionally no significant differences between deterministic and random codeword generation strategies were reported and thus, the latter is generally recommended.

However, the existing random codeword generation strategies generating 10000 codes can fail: If the number of classes is small, they might not even yield a single valid codeword matrix. On the reverse, by an increasing number of classes the generation of many random matrices takes much time, while in the same way the codewords might become invalid due to equivalent rows encoding different classes [Gebel 2009].

Besides these drawbacks, other works compare the results of ECOC approaches to the ones of the one-vs-all and one-vs-one decomposition. In particular, it is stated that one-vs-all support vector machines yield comparable results to those of more

complex ECOC-based approaches [Kikuchi & Abe 2003; Liu et al. 2008], especially if the respective classifiers are carefully optimized [Rifkin & Klautau 2004; Rifkin 2002]. Later works reported similar [Galar et al. 2015] to superior [Escalera et al. 2010] recognition performance of the one-vs-one decomposition in comparison with other codeword design strategies. Since the one-vs-one and one-vs-all decompositions are additionally more common [Bain et al. 2019; Chmielnicki & Stąpor 2016; Fernandez et al. 2015; Galar et al. 2010, 2011, 2013, 2017, 2014; Hsu et al. 2019; Khalifa et al. 2019; Krzyśko & Wołyński 2009; Mendialdua et al. 2015; Morán-Fernández et al. 2016; Ribeiro et al. 2018] than other approaches, they are the most studied [Sáez et al. 2012] de-facto standards, most likely due to their clarity [Arruti et al. 2014; Bagheri et al. 2012].

One-vs-All or One-vs-One Decomposition?

The previously summarized results lead to the question which one of the two standard decompositions should be preferred. Still at first, a few general properties can be remarked. The underlying concepts imply that the one-vs-all decomposition constructs k binary classification problems consisting of all r training instances. On the contrary, the one-vs-one decomposition creates a quadratic number of binary problems. The first conclusion might be that the one-vs-all decomposition is thus faster to train. Interestingly, the reverse is true in most cases since the number of samples r_i+r_j is an order of magnitude smaller than r. Hence, the optimization problems are also smaller and faster or easier to solve, resulting in reduced overall training time requirements as stated by many authors [Bagheri et al. 2012; Escalera et al. 2010; Fürnkranz 2001, 2002a,b, 2003; Gonzalez-Abril et al. 2010; Hsu & Lin 2002; Hüllermeier & Brinker 2008; Hüllermeier & Vanderlooy 2010; Park & Fürnkranz 2007; Platt et al. 1999; Saez et al. 2019; Sáez et al. 2012; Wu et al. 2014].

Nevertheless, training the individual models with much less data can increase their variances [Lee et al. 2004], and the overall classification should have a higher tendency to overfit, at least if the individual classifiers are not sufficiently regularized [Platt et al. 1999]. Furthermore, the quadratic complexity of the one-vs-one scheme might be too high for large numbers of classes [Chmielnicki & Stąpor 2016], while a one-vs-all decomposition might still be applicable. Similarly, the vast majority of one-vs-one decomposition-based algorithms are less efficient at prediction time. Not only a quadratic instead of a linear number of individual predictions needs to be computed, also the fusion step based on pairwise coupling is required. However, for many practical problems it is still sufficiently efficient such that this issue is of minor practical relevance.

Another important aspect is that the one-vs-one scheme keeps the original class ratios. In contrast to this, the one-vs-all decomposition's training data sets are unavoidably imbalanced as there are much more samples from one class than from the other one, actually consisting of $k-1$ merged classes [Chmielnicki & Stąpor 2016; Galar et al. 2011; Saez et al. 2019; Sáez et al. 2012, 2014; Zhang et al. 2016]. As a consequence, the individual one-vs-all classifiers might tend to always predict the negative class [Arruti et al. 2014; Bagheri et al. 2012; Chmielnicki & Stąpor 2016].

Besides these primarily theoretical properties, many authors report superior accuracy of the one-vs-one over the one-vs-all approach [Allwein et al. 2000; Angulo & Català 2000; Bagheri et al. 2012; Fürnkranz 2001, 2002b, 2003; Hsu & Lin 2002; Hüllermeier & Brinker 2008; Hüllermeier & Vanderlooy 2010; Krzyśko & Wołyński 2009; Liu et al. 2006; Saez et al. 2019; Sáez et al. 2012; Tsujinishi et al. 2004; Wu et al. 2014], which even is amplified with an increasing number of classes [Fürnkranz

2003]. Especially relevant are available comprehensive studies [Galar et al. 2011; García-Pedrajas & Ortiz-Boyer 2008, 2011; Krzyśko & Wołyński 2009] concluding that one-vs-one in combination with pairwise coupling is often and sometimes even significantly superior to the one-vs-all approach. Still, the best decomposition strategy might also depend on the respective classifier and its parameterization.

Presumably, the one-vs-one decomposition is often superior to the one-vs-all scheme because the decision boundaries between the classes are simpler [Fürnkranz 2001, 2002b; Mendialdua et al. 2015; Saez et al. 2019; Sáez et al. 2012, 2014] and discriminant features are easier to learn in the pairwise problems [Lin & Davis 2008]. Other authors even state that the one-vs-all decomposition cannot be a good method for complex class distributions [Park et al. 2009]. A family of different works emphasize the good results obtained with the one-vs-one decomposition [Chmielnicki & Stąpor 2016; Fernandez et al. 2015; Moreira & Mayoraz 1998] and its widespread application [Galar et al. 2013, 2017, 2014, 2015; Li et al. 2005a,b; Sáez et al. 2014], however without explicitly contrasting it to possible alternatives.

Comparisons between the two decompositions that motivate to use the one-vs-all approach instead exist [Rifkin & Klautau 2004; Rifkin 2002], however the authors only conclude that fine-tuned one-vs-all support vector machines can compete with one-vs-one-based ones if binary voting is used for the latter. Neither different classification algorithms nor more elaborated pairwise coupling strategies were used in the respective experiments. Other works also note comparable performance between both decompositions in combination with support vector machines [Platt et al. 1999], still only during an experiment on three data sets applying the one-vs-one approach with binary voting. Additionally, the one-vs-all decomposition is more robust towards class noise [García-Pedrajas & Ortiz-Boyer 2008, 2011], which is easily explainable since incorrect labels in groups of merged classes are irrelevant. In particular, this was even generally observed for dense ECOC codes in comparison with sparse ones.

Further Applications of Decomposition-based Classification

Many of aforementioned research was conducted from the demand to apply binary classification algorithms and in particular support vector machines for multi-class problems. Still, one-vs-all classification is also prevalent in other machine learning algorithms like neural networks or different native multi-class variants of support vector machines [Doğan et al. 2016]. Despite supporting more than two classes by construction, here the important difference is that the classifier's components are *jointly* optimized – for example using a softmax combination in case of neural networks – while in the classical reduction approach, the trainings are performed *independently*. Nevertheless, once the classifier is fully trained, this difference does not really matter anymore. In the same way, all one-vs-one or arbitrary ECOC classifiers could be trained simultaneously in a large comprehensive optimization problem, at least in theory.

In this regard, aforementioned results are particularly relevant to apply binary classification algorithms for multi-class problems. Interestingly, there are also many different works that apply [Fürnkranz 2002b, 2003; Galar et al. 2011; García-Pedrajas & Ortiz-Boyer 2008, 2011; Kong & Dietterich 1995, 1997; Ou & Murphey 2007; Reid 2010; Sáez et al. 2012, 2014] or at least motivate [Chmielnicki 2015; Elkano et al. 2015; Fernandez et al. 2015; Galar et al. 2013, 2017, 2014, 2015; Morán-Fernández et al. 2016; Quost & Destercke 2018] ECOC strategies and in particular the one-vs-one decomposition in combination with native multi-class machine learning algorithms like for example decision trees to increase their classification accuracy or robustness

towards noise. In a similar way, the one-vs-one reduction was applied to improve the results in the *overlapping class problem* [Saez et al. 2019], which both emphasizes the advantages of the one-vs-one decomposition strategy besides its straightforward application in combination with binary-only classifiers.

Non-Competence Problem

Besides these general recommendations regarding the one-vs-one decomposition, there is still a mostly unsolved problem related to it - the *non-competence problem*: In total there are $\binom{k}{2}$ different individual classifiers, but while predicting a newly observed instance x with *unknown* class only a fraction out of them was actually trained using data from the respective class. All other classifiers are *incompetent* to predict x and their predictions influence the ensemble's overall prediction. Ideally, the coupling of the individual classifiers is restricted to the competent ones only, but it is unknown in practice which are the competent ones, otherwise the classification problem would be solved.

Clearly, in case of the one-vs-all decomposition there are no incompetent classifiers, and the same holds for any ECOC-based approach whose codeword matrix does not contain at least one zero. On the reverse, there are incompetent classifiers as soon as there are zero entries in the codeword matrix. Since the one-vs-one scheme is the most popular respective approach, the existing literature about the non-competence problem focuses on it. Still, it is generally relevant for other but mainly uncommon approaches.

Some works dealing with the one-vs-one decomposition mention it as an error source, but assuming that all competent classifiers vote for the correct class will overrule all incompetent votes in combination with binary voting [Fürnkranz 2001, 2002b; Sáez et al. 2012]. Additionally, the incompetent votes are likely to be independent, which might also alleviate their influences. Different works mention it in the contexts of *ordinal classification* and *multipartite ranking* [Fürnkranz et al. 2009] as well as *label ranking* [Hüllermeier & Vanderlooy 2010].

Because there are only $k-1$ competent classifiers, the number of incompetent ones increases quadratically with respect to k. Therefore, the ratio of competent ones even approaches zero for $k \to \infty$. Even though this is only an asymptotic result, it shows that the influence of the incompetent votes increases with the number of classes [García-Pedrajas & Ortiz-Boyer 2008]. Other authors emphasize that there will be more incompetent than competent classifiers as soon as $k \geq 5$ [Quost & Destercke 2018]. Thus, works dealing with the non-competence problem should explicitly aim at problems with more than a couple of classes. Besides these results, different works explicitly formulated the non-competence problem as an open issue [Elkano et al. 2015; Fernández et al. 2013; Galar et al. 2011, 2014].

Extended Fusing Methods

The first work addressing the non-competence problem trained $\binom{k}{2}$ additional *correcting classifiers* (CC) [Moreira & Mayoraz 1998] that separate each pair of classes $\{i,j\}$ from the set of all other ones $\{1,2,\ldots,k\}\backslash\{i,j\}$. In particular, the authors extended the probabilistic voting by multiplying the pairwise probabilities $\phi_{i,j}$ with the respective probabilistic output $\mathsf{w}_{i,j}$ of the corresponding correcting classifier and reported a significant improvement. However, the respective study only evaluated five data sets. In a second experiment they additionally presented a modified approach to improve the efficiency. Here, the correction classifiers for the pair $\{i,j\}$ were replaced

by the corresponding one-vs-all classifiers, however the reported results are inferior and most times even worse than an uncorrected probabilistic voting.

The same approach using correcting classifiers was independently reintroduced a decade later [Reid 2010] with similar reported accuracy increases, however with a deeper formal reasoning: The unknown posterior probability $P(\mathsf{y} \mid \mathsf{x})$ can be expressed as

$$P(\mathsf{y} = i \mid \mathsf{x}) \;=\; P(\mathsf{y} = i \mid \mathsf{x},\ \mathsf{y} \in \{i, j\}) \;\cdot\; P(\mathsf{y} \in \{i, j\} \mid \mathsf{x}) \tag{2.30}$$

for any other class $j \neq i$, where the former term can be estimated from the pairwise classifier $\mathsf{f}_{i,j}$, while the latter represents the weight. However, estimating the pairwise posterior $P(\mathsf{y} \in \{i, j\} \mid \mathsf{x})$ is similarly complex as estimating $P(\mathsf{y} = i \mid \mathsf{x})$, but (2.30) allows the averaging of the posterior probability for class i over all pairs $\{i, j\}$ with $i \neq j$. Here, $P(\mathsf{y} \in \{i, j\} \mid \mathsf{x})$ is replaced by the pairwise probability of the correcting classifier (which is called *pair-vs-rest* in the respective work). Therefore, the two approaches describe in fact the same technique.

Even though this formal interpretation is an interesting insight, unluckily it is flawed: The correcting classifiers do not produce estimates of $P(\mathsf{y} \in \{i, j\} \mid \mathsf{x})$ in the same way as independent one-vs-all classifiers do not produce estimates of $P(\mathsf{y} = i \mid \mathsf{x})$ since the latter do not sum to one. This can be alleviated by normalizing the one-vs-all probabilities, however most likely this will destroy the previously performed calibration, and an equivalent normalization is impossible for the correcting classifiers.

Another drawback of this approach is that the corrected pairwise probabilities do not sum to one anymore, therefore the application of all pairwise coupling techniques is not possible. Here, they can be explicitly back-transformed into pairwise probabilities [Li et al. 2005a,b], but in combination with the preceding weighting, this results in a simple nonlinear transformation of the pairwise probabilities. This is highly unjustified as it conflicts with the foregoing calibration step. Similarly to the one-vs-all reduction, the correcting classifiers might also suffer from imbalanced data sets [Chmielnicki & Stąpor 2016].

An alternative approach to tackle the non-competence problem constructs k modified one-vs-one ensembles in parallel, where the i-th one only contains the $k-1$ pairwise classifiers that are trained using instances from class i [Li & Tang 2002; Xu et al. 2005]. The presented, modified pairwise coupling approach for incomplete ensembles introduced by the authors allows an estimation of the posterior probability distribution by each of the k ensembles. Overall, the class is predicted that yields the smallest error with respect to its corresponding ensemble under either the Brier score or the log-loss. However, this approach only addresses the non-competence problem in the respective modified incomplete pairwise coupling, but does not address its influence in complete one-vs-one ensembles.

Two other works independently introduced the same technique that avoids the non-competence problem by first selecting the two most confident one-vs-all predictions and thereafter applying the respective single one-vs-one classifier for the overall classification [Garcia-Pedrajas & Ortiz-Boyer 2006; Ko & Byun 2003] to yield a significant improvement. In a related way, also the combination of all one-vs-all and one-vs-one classifiers can be used in a simple majority voting scheme [Arruti et al. 2014]. The same work additionally introduces NOV@ that adaptively combines both approaches: If there is exactly one positive response from the one-vs-all classifiers, it is accepted. Otherwise, if there are multiple positive one-vs-all responses, the pairwise coupling is restricted to the corresponding set of classes. Finally, if there are no positive responses from the one-vs-all classifiers, pairwise coupling is applied on the whole class set. Hence, the one-vs-all scheme controls the set of classes on which

the pairwise coupling is applied. The results obtained with NOV@ were superior in most of the performed experiments. Even before this, the one-vs-one ensemble was restricted to the selection of most confident one-vs-all classifiers in previous work [Böken 2014] to achieve similar advantages. However, the confidence threshold or the number of classes have to be arbitrarily selected such that NOV@ can be interpreted as an extended variant that avoids this disadvantage, even though it was independently introduced later.

Besides these approaches, different strategies were presented that are based on nearest neighbor searches. Hence, they use the instance x in question at prediction time and compute its K nearest neighbors in the training data set. The existing approaches differ in the way how they use these in combination with pairwise coupling techniques, but these ideas are similar to those of aforementioned hybrid strategies.

A first option uses the two most frequent classes of the neighborhood in combination with the corresponding classifier or to restrict the pairwise coupling to neighbored classes only [Bagheri et al. 2012]. In particular, the authors used the five nearest neighbors in their experiments and reported accuracy increases over all other alternative decomposition strategies and their corresponding prediction approaches. The latter approach also was additionally independently introduced [Fernandez et al. 2015; Galar et al. 2013], however with an additional adaptive nearest neighbor search. In particular, the number of neighbors was selected as three times the class count, $K = 3 \cdot k$, however this count is adaptively increased up to $6 \cdot k$ if the neighborhood contains only a single class. If this still also holds for $K = 6 \cdot k$, the whole set of classes is used. In the performed empirical studies, also an increased prediction performance is reported. In a follow-up work, the same idea was generalized to a nearest neighbor classification based on the pairwise probabilities matrix [Galar et al. 2017]. Other authors combine these techniques [Goienetxea et al. 2021] with dynamic classifier selection [Mendialdua et al. 2015].

In contrast to these strategies performing a binary selection of classes, also nearest neighbor-based weighting was introduced as an alternative [Galar et al. 2015]. Here, at first the minimum distances $\mathsf{d}_i(\mathsf{x})$ of x to a training data instance of class i are computed and used to construct weights

$$\mathsf{w}_{i,j}(\mathsf{x}) = \frac{\mathsf{d}_j^2(\mathsf{x})}{\mathsf{d}_i^2(\mathsf{x}) + \mathsf{d}_j^2(\mathsf{x})},\ 1 \leq i, j \leq k \tag{2.31}$$

from the distances. Thereafter, the pairwise probabilities are multiplied with the weights $\phi_{i,j}(\mathsf{x}) \cdot \mathsf{w}_{i,j}(\mathsf{x})$ and coupled with a probabilistic voting. Hence, the approach is analogous to the correcting classifiers and they only differ in the weight computation. Furthermore, in both cases the rescaled probabilities do not sum to one anymore. Still, the presented approach outperformed the alternative approaches in the performed experiments.

Independent of its current approach (i.e. either one-vs-all classifiers or nearest neighbor searches), any adaptive class selection technique alleviates the influence of the non-competent classifiers but does not solve it. As long as there are at least three remaining classes, there are still non-competent classifiers, only their number and consequently the overall influence is decreased. Additionally, restricting the set of classes can be combined with any other technique directly targeting the non-competence in the remaining class set. Therefore, only the approaches multiplying the pairwise probabilities by weights directly and systematically address the non-competence problem. However, these suffer from two drawbacks. First, they make general pairwise coupling techniques inapplicable because the pairwise probabilities do not sum to

one anymore. Furthermore, using nearest neighbor searches for both, selection and weighting approaches, can be problematic in practice. For large data sets they are expensive at prediction time, require to save the whole training data set and can be unreliable in large feature dimensions. Besides this, if nearest neighbor searches yield particularly reliable results in the current application, it is at least questionable why they are not directly used to classify the instances. Therefore, it is easily explainable that the respective technique's scalability is already formulated as an open research question [Galar et al. 2013].

Another important property of all decomposition-based approaches is the primarily heuristic motivation and the lack of theoretical interpretation. The only exception are jointly optimized probabilistic one-vs-all classifiers, however they are most likely to be uncalibrated because even the estimation of calibrated binary posterior probabilities is a very challenging problem. On the other hand, even if all individual classifiers were *independently* predicting the true respective binary posterior probabilities, fusing them into an overall estimation has no clear Bayesian interpretation. The best possible one is interpreting them as marginals of the unknown posterior distribution $P(\mathsf{y} \mid \mathsf{x})$, however how to combine them remains unclear. Here, adaptive weighting of one-vs-one predictions in pairwise coupling empirically tends to perform well in some studies but similarly lacks theoretical justification. Even though the respective discussion aims at the correcting classifiers, the situation of (2.30) does not depend on them, is equally valid for any weighting approach that adjusts the pairwise probabilities and is similarly extendable for other decompositions.

2.4 Comparison of Methods

Based on the previous summary of existing methods, the following approaches, which are also summarized in table 2.1, are used as references for an extended algorithmic approach that generalizes multi-class classification into dynamic contexts:

- The one-vs-all decomposition using the softmax transformation (Softmax) allows a probabilisitic interpretation of the predictions. It consists only of k subproblems, thus is efficient after training, but the individual problems are relatively complex and hence, hard to solve. As long as the prediction functions are simultaneously estimated using a proper scoring function (as usually the case during neural network training), it has a particular theoretic background from posterior probability estimation. This has a strong relation to classifier calibration and will be discussed in chapter 3. Additionally, there are no incompetent classifiers, but this also means that there is almost no real possibility to integrate dynamic class information: The only option is to restrict the softmax function to the dynamic set $\mathcal{M}$. Still, each prediction depends on data from *all* classes such that only a minor adaption to dynamic contexts is possible.

- The one-vs-one decomposition using probabilistic voting (Vote) as given by (2.24) is less efficient at prediction time since a quadratic number of predictions has to be computed. On the other hand, the individual problems are simpler and thus easier to solve. Summing of probabilities with different interpretations has no real theoretic justification such that it is a mainly heuristically motivated approach. The non-competence problem is predominant, but its negative influences in practice remain unclear. The approach is not designed to integrate dynamic information, still restricting the fusing process to class subsets only is possible.

- The one-vs-one decomposition using the non-dominance criterion (ND) as given by (2.25) shares the same advantages and disadvantages of probabilistic voting.

- The one-vs-one decomposition in combination with solving (2.26) at prediction time (WLW) similarly shares the same advantages and disadvantages of the previous two approaches, however offers an improved theoretic background. The squared errors $(\phi_{j,i} \cdot p_i - \phi_{i,j} \cdot p_j)^2$ compensate individual deviations between $\phi_{i,j}$ and $p_i \approx P(\mathsf{y} = i \mid \mathsf{x},\ \mathsf{y} \in \{i, j\})$ into an overall estimation that minimizes the sum of squared errors. Still, it becomes problematic to condition on conflicting events as all pairs are used simultaneously. This major issue will be discussed in chapter 5.

- The correcting classifiers approach (CC) is presumably the most elaborated technique to tackle the non-competence problem, however this results in a reduced efficiency. At prediction time, $k \cdot (k + 1)$ individual predictions are required, while at training time the complex pair-vs-rest classifiers have to be computed in addition to the pairwise predictors. A theoretic foundation is also not available and the support of restricting to a dynamic target set is limited because the pair-vs-rest classifiers always depend on all data.

- Extending the combination by voting on the one-vs-all and one-vs-one predictions (CombVote) is comparable efficient to the one-vs-one decomposition alone at prediction time because only k additional predictions have to be computed here. However, the real benefit remains unclear, performing voting with different kinds of classifiers is completely heuristically motivated and lacks any theoretical justification. Similarly, it is unclear how useful this voting scheme is with respect to the non-competence problem. There remain as many incompetent votes as there are incompetent classifiers in the one-vs-one reduction.

- Finally, combining both decompositions with NOV@ is presumably the most elaborated technique to select the subset of classes on which probabilistic voting is performed. It requires the training of all classifiers from both decompositions, but at prediction time only the fraction of relevant one-vs-one predictions have to be computed. Even though it is problem-dependent how many one-vs-one predictions are required, it is still reasonable to expect a significantly smaller fraction on average. Therefore, also the influence of the non-competent predictions is expected to be alleviated in many cases. Besides this, dynamic class information can be integrated, but the one-vs-all classifiers still implicitly depend on all data.

It is important to emphasize that all methods only refer to the respective decomposition and fusing approaches. To apply them in practice, they always depend on an arbitrary machine learning algorithm to solve the decomposition's base problems. Here, classifier calibration is important in supplying the respective probabilistic predictions that are required for most fusing techniques, e.g. (2.23). In light of this, the following chapter 3 presents both, theoretic and empirical results on classifier calibration.

With particular focus on dynamic classification, the summarized methods share two remarkable properties: First, they assume the target set to be fixed such that no direct reference results are available on dynamic classification. Besides this, the approaches often have no real theoretical justification and are mostly heuristically motivated. Therefore, evidence theory is used in chapter 4 to develop a theoretically

Method	Number of Classifiers	Avg. Problem Complexity	Prediction Efficiency	Theoretic Justification	Non-Competence Problem	Support of Dynamic Class Information
Softmax (1vA)	k	−	+	∘	+	−
Vote (1v1)	$\frac{1}{2} \cdot k \cdot (k-1)$	+	∘	−	−	∘
ND (1v1)	$\frac{1}{2} \cdot k \cdot (k-1)$	+	∘	−	−	∘
WLW (1v1)	$\frac{1}{2} \cdot k \cdot (k-1)$	+	∘	∘	−	∘
CC (1v1 + CC)	$k \cdot (k-1)$	∘	∘	−	+	−
CombVote (1vA + 1v1)	$\frac{1}{2} \cdot k \cdot (k+1)$	∘	∘	−	−	−
NOV@ (1vA + 1v1)	$\frac{1}{2} \cdot k \cdot (k+1)$	∘	∘	−	∘	−

TABLE 2.1: Summarized properties of the most relevant existing decomposition-based classification methods.

justified approach to decomposition-based classification that *by design* supports the extension for a dynamic class set.

Here, the characteristics of the respective learning algorithm are particularly relevant. Throughout many domains, deep neural networks gained much popularity in recent years. In combination with multi-class classification, they are usually trained using a softmax prediction layer, whose corresponding one-vs-all decision functions are simultaneously optimized. Training other decompositions in combination with large and complex models poses a challenging task: Even though from the theoretical point of view, other reductions like the one-vs-one decomposition could be applied and the respective optimizations be independently solved, this requires to train $\Omega(k^2)$ models. These depend on sufficiently large training sets, require time to optimize and computing independent predictions massively increases the required computational resources at prediction time. Still in combination with simpler models, there is a consensus in the literature – as presented in full detail in section 2.3 – that the one-vs-one decomposition often outperforms the one-vs-all reduction. Consequently, it is an interesting and relevant question whether large-scale models similarly improve from feasible approaches to apply other decomposition-based classification methods besides the one-vs-all reduction. These can result in potential improvements of the respective models in actual applications.

Chapter 3

Classifier Calibration

The preceding summary of the existing results showed that probabilistic predictions are either required for most decomposition-based classification techniques or at least are useful to preserve comparability between different individual predictors. Additionally, there are many other particularly relevant applications as presented in full detail before. Still, the focus of this thesis lies on the former kind of applications.

In this regard, this chapter presents both theoretical and empirical results that are relevant to select the appropriate calibration methods for the extended algorithms presented in the following chapters 4 and 5. Here, section 3.1 at first performs a theoretical analysis where the monotonic assumption is one of the main aims. Next, section 3.2 presents two powerful model-free calibration techniques, which are especially useful for non-monotonic settings. A different main aim is to review existing calibration evaluation metrics in section 3.3 and to show that bin-based ones, which gained popularity in recent publications, are invalid as well as unreasonable and should not be used at all. Finally, the theoretical findings are empirically supported using two empirical studies in section 3.4. First, a simulation study is performed under perfect information and thereafter, the results of different state-of-the-art calibration techniques are analyzed on a collection of 46 large-scale real-world data. Excerpts of the following results were already published [Böken 2021], in particular a summarized version of the current research results previously presented in subsection 2.2.2, most of the theoretical results as well as a slightly modified simulation study.

3.1 Theoretic Results

Accurate posterior probability estimation in general and classifier calibration in particular face the challenging problem that in practice, true reference probabilities are unknown. Thus even for practical applications, theoretical results supporting certain techniques are important in selecting appropriate calibration methods.

3.1.1 Monotonicity

Presumably one of the most controversially discussed issues related to classifier calibration is the question whether binary calibration techniques should be designed monotonic or not, as presented in chapter 2. Most likely, this assumption originates from the fact that the decision function's magnitude can be interpreted as some measure of confidence, giving rise to a monotonic assumption. This explains why it is often made in practice [Bennett 2006], even though it is strong but usually reasonable [Wu et al. 2014; Xu et al. 2016] and can help to avoid overfitting. Another more recent work even formulates non-decreasingness as a requirement [Wang et al. 2019].

Model-free calibration techniques like binning are by construction non-monotonic in general, thus the monotonic discussion is strongly related to the parametric

approaches as well as isotonic regression. Before proving theoretic properties of Platt scaling and beta calibration, at first two incorrect statements related to monotonicity that appeared in the literature are corrected.

Correction of Wrong Statements

The first particular example [Bella et al. 2013] motivates non-monotonic calibration from general multi-class settings, where the authors assume a 1-of-k encoding scheme. After independently calibrating the k one-dimensional decision functions, the probabilities will not sum to one and thus, normalizing destroys monotonicity even if all binary calibrations are monotonic. However, using this argumentation to criticize monotonicity in the binary case is simply wrong. The correct way to embed the binary case in the multi-class one is to interpret the prediction function f as a two-dimensional equivalent one $(\mathsf{f} - \mathsf{f}_0, \mathsf{f}_0 - \mathsf{f})^\top$ relative to the decision threshold f_0. Generally, this two-dimensional function can also be independently calibrated, but the sum-to-one constraint of a probability reduces it to a one-dimensional estimation, which has a natural ordering. However in the multi-class case, there still remain $k-1 > 1$ degrees of freedom in the calibration map and thus, monotonicity is simply undefined since there is no natural ordering on $\mathbb{R}^k$ for $k > 1$. Consequently, there is no monotonicity in multi-class calibration and thus, it can neither criticize nor support monotonicity in the binary case.

Another incorrect statement is made in the series of works that introduced the state-of-the-art ensemble-based calibration techniques ENIR and ELiTE [Naeini 2016; Naeini & Cooper 2015, 2016, 2018]. Here, the authors state that assuming a monotonic relationship between classifier scores and posterior probabilities is equivalent to assuming that the classifier has $\mathsf{AUC} = 1$, at least asymptotically. Even though this seems to be valid, in fact it is wrong and the opposite is true:

Proposition 3.1. *For any AUC in* $(0.5, 1)$ *there exist distributions of class-conditional likelihoods* $p(\mathsf{f} \mid \mathsf{y} = \pm 1)$ *such that the posterior probabilities take the form of* $P(\mathsf{y} = 1 \mid \mathsf{f}) = (1 + \exp(a \cdot \mathsf{f} + b))^{-1}$ *for some real-valued parameters* a *and* b*. Thus, the transformation from* f *into* $P(\mathsf{y} = 1 \mid \mathsf{f})$ *is strictly monotonic.*

Proof. Select a uniform prior of 0.5 and unit-variance, Gaussian-distributed likelihoods $p(\mathsf{f} \mid \mathsf{y} = \pm 1)$ with means $\pm\mu$. By substituting these into Bayes' theorem it is straightforward to see that the posterior distribution takes the claimed form, for any selected value of μ (in particular, even $a = -2\mu$ and $b = 0$ hold). Clearly, for $\mu \to \infty$ the AUC becomes arbitrarily close to one, while for $\mu \to 0$, the AUC becomes arbitrarily close to 0.5. Since the AUC obviously depends continuously on μ, the claim is proven. □

It is important to emphasize that the sigmoidal relationship in the extreme cases of $\mathsf{AUC} = 1$ and $\mathsf{AUC} = 0.5$ is not valid because the posterior probability $P(\mathsf{y} = 1 \mid \mathsf{f})$ is either binary or constant (but the transformation still is trivially monotonic).

However, neither do these results yield a justification to always enforce monotonicity in calibration nor is a similar general reasoning likely to exist at all – there always can be settings in which the transformation function is non-monotonic. Still, it is an interesting question under which circumstances a monotonic transformation can reasonably be defended and how it can be well approximated given finite training data only. The theoretic results following in subsection 3.1.2 focusing on Platt scaling enlighten that a monotonic transformation indeed is valid for different families of score distributions.

Analyzing Isotonic Regression

Besides results that will focus on parametric estimations of monotonic calibration functions, it is also relevant to analyze problem settings in which none of these parametric models are valid, but still a monotonic transformation function should be estimated. Isotonic regression is a straightforward selection here, but with focus on classifier calibration, further results like convergence rate or approximation guarantees were not discussed in the corresponding literature, and research in this particular direction is still relatively rare. In light of this, a few interesting results should also be mentioned.

First of all, it is important to note that isotonic regression is not primarily designed for classifier calibration, but instead as a non-parametric regression technique to fit independent and identically distributed data points $z_i \in \mathbb{R}$, $i = 1, \ldots, r$, with given continuous function values t_i by non-decreasing estimates $\hat{t}_i$ that minimize the sum of squared distances $\sum_{i=1}^{r}(t_i - \hat{t}_i)^2$. In this setting, the empirical ℓ^p risk is bounded by $r^{-\frac{1}{3}}$

$$\left(\frac{1}{r} \cdot \sum_{i=1}^{r} \mathbb{E}\left|t_i - \hat{t}_i\right|^p\right)^{\frac{1}{p}} \in \mathcal{O}\left(r^{-\frac{1}{3}}\right) \tag{3.1}$$

for $1 \leq p < 3$ if the true relationship is monotonic but the observations are influenced by zero-mean equal-variance Gaussian-distributed noise [Zhang 2002], similar to the assumptions in linear regression. Thus, the estimator's error converges to 0 in ℓ_p for $r \to \infty$, and consequently this convergence also holds in probability. Different extended properties about the convergence of isotonic regression are discussed in the respective literature [Robertson et al. 1988; Yang & Barber 2019].

A particular important result of these two works with focus on classifier calibration shows that the mapping computed using isotonic regression – or more precisely the constrained maximum likelihood estimator computed using the PAV algorithm – also equals the optimal solution for *binary* observations $z_i \equiv \mathsf{f}(\mathsf{x}_i)$ with Bernoulli-distributed labels y_i according to $p(\mathsf{f}(\mathsf{x}_i)) = P(\mathsf{y} = 1 \mid \mathsf{f}(\mathsf{x}_i))$. Thus, the simple PAV algorithm enables to approximate the true function with high probability without assuming anything else besides monotonicty of the transformation function, and without restricting the interpolation between the given data points z_i.

The next important observation is that isotonic regression does not necessarily define piecewise constant and thus discontinuous transformations. Instead, the PAV algorithm returns points from its training data set – with respect to calibration from the set of predictions $\{\mathsf{f}(\mathsf{x}_1), \mathsf{f}(\mathsf{x}_2), \ldots, \mathsf{f}(\mathsf{x}_r)\}$ – and returns associated posterior probability estimates. These can be converted into a binning model in the same way as being linearly interpolated to form a piecewise linear, continuous and strictly monotonic transformation with non-differentiable points where the slopes of the line segments change. Even any monotonic piecewise continuous function is a valid option [Álvarez & Yohai 2011].

In a similar category fall the two existing strategies that interpolate the points using splines or monotonic polynomials [Jiang et al. 2011; Wang et al. 2019]. From a strictly formal point of view, these extensions can be advantageous over a discontinuous, piecewise constant transformation. However, higher order polynomials were already observed to overfit [Naeini & Cooper 2016, 2018] and given sufficient data such that overfitting should be less problematic, especially a simple piecewise constant (or piecewise linear to satisfy continuity and strict monotonicity) function will probably accurately approximate the true one and simply by Occam's razor will be the better choice. If the results are still not satisfying, it should be interpreted as an

evidence against monotonic calibration in the respective application instead, but not as a demand for more complex, but still monotonic transformation functions. Additionally, it is particular challenging to objectively detect the invalidity of a monotonic transformation, which is a consequence of the discussion following in section 3.3.

Still, a parametric model can be superior if its assumptions are met. In light of this, the following part proves different sufficient conditions for the validity of Platt scaling and an equivalence to beta calibration.

3.1.2 Platt Scaling's Parametric Assumptions

Even though some works [Flach 2016] incorrectly state that Platt scaling assumes Gaussian-distributed likelihoods $p(\mathsf{f} \mid \mathsf{y})$, the summary of the existing results related to Platt scaling already showed that it is also valid for shifted exponential distributions [Platt 1999]. Consequently, a transformation function in the form of (2.7) is an optimal choice for at least two different families of probability distributions. This additionally shows that it is still relatively flexible, in contrast to aforementioned criticism. Still, there are no results available clarifying its parametric assumptions. In light of this, the main contribution of this section is to prove under which circumstances it is an optimal choice.

The key idea of the following analysis is to decompose the likelihoods into different parts that cancel out in the likelihood estimation and thus, do not influence the posterior probabilities.

Theorem 3.2. *Let $f\colon \mathcal{X} \to \mathbb{R}$ be a given binary classifier on the input domain $\mathcal{X}$. If the class-conditional likelihoods are strictly positive and can be expressed as*

$$p(f(x) \mid y = i) \;=\; g_i(f(x); \lambda_i, \theta) \tag{3.2}$$

for $f(x) \in I \subseteq \mathbb{R}$ such that the g_i can be factorized as

$$g_i(z; \lambda_i, \theta) \;=\; \gamma(z; \theta) \cdot \beta_i(\lambda_i, \theta) \cdot \exp(\alpha_i(\lambda_i, \theta) \cdot z) \tag{3.3}$$

where

- *the index $i = \pm 1$ refers to the respective class*
- *λ_i defines arbitrary class-specific parameters*
- *θ are the jointly used parameters*
- *$\alpha_i(\lambda_i, \theta)$ and $\beta_i(\lambda_i, \theta)$ are arbitrary functions that are constant with respect to z*
- *γ is an arbitrary function that is independent of the class-specific parameters λ_i*

then the posterior probabilities on I are distributed such that

$$P(y = 1 \mid f(x)) \;=\; \sigma_{a,b}(f(x)) \;=\; \frac{1}{1 + \exp(a \cdot f(x) + b)} \tag{3.4}$$

holds with real-valued parameters given by $a = \alpha_{-1}(\lambda_{-1}, \theta) - \alpha_1(\lambda_1, \theta)$ as well as $b = \log \frac{P(y=-1)}{P(y=1)} - \log \frac{\beta_1(\lambda_1, \theta)}{\beta_{-1}(\lambda_{-1}, \theta)}$. In particular, the posterior probabilities are non-constant (i.e. $a \neq 0$) if and only if $\alpha_1(\theta, \lambda_1) \neq \alpha_{-1}(\theta, \lambda_{-1})$ and further, the sigmoid assumption holds.

Proof. Without loss of generality it holds $\gamma(z;\theta) > 0$ and $\beta_i(\lambda_i,\theta) > 0$ as their product is positive. Thus, the likelihood ratio is well defined and simplifies to

$$\begin{aligned}
\tfrac{p(\mathsf{f}(\mathsf{x})|\mathsf{y}=1)}{p(\mathsf{f}(\mathsf{x})|\mathsf{y}=-1)} &= \tfrac{g_1(\mathsf{f}(\mathsf{x});\lambda_1,\theta)}{g_{-1}(\mathsf{f}(\mathsf{x});\lambda_{-1},\theta)} \\
&= \tfrac{\gamma(\mathsf{f}(\mathsf{x});\theta)\ \cdot\ \beta_1(\lambda_1,\theta)\ \cdot\ \exp(\alpha_1(\lambda_1,\theta)\ \cdot\ \mathsf{f}(\mathsf{x}))}{\gamma(\mathsf{f}(\mathsf{x});\theta)\ \cdot\ \beta_{-1}(\lambda_{-1},\theta)\ \cdot\ \exp(\alpha_{-1}(\lambda_{-1},\theta)\ \cdot\ \mathsf{f}(\mathsf{x}))} \\
&= \tfrac{\beta_1(\lambda_1,\theta)}{\beta_{-1}(\lambda_{-1},\theta)}\ \cdot\ \exp(\alpha_1(\lambda_1,\theta)\cdot\mathsf{f}(\mathsf{x}))\ \cdot\ \exp(-\alpha_{-1}(\lambda_{-1},\theta)\cdot\mathsf{f}(\mathsf{x})) \\
&= \tfrac{\beta_1(\lambda_1,\theta)}{\beta_{-1}(\lambda_{-1},\theta)}\ \cdot\ \exp((\alpha_1(\lambda_1,\theta)-\alpha_{-1}(\lambda_{-1},\theta))\cdot\mathsf{f}(\mathsf{x})) \\
&= \exp\left((\alpha_1(\lambda_1,\theta)-\alpha_{-1}(\lambda_{-1},\theta))\cdot\mathsf{f}(\mathsf{x}) + \log\left(\tfrac{\beta_1(\lambda_1,\theta)}{\beta_{-1}(\lambda_{-1},\theta)}\right)\right).
\end{aligned}$$

In the last line, taking the logarithm is valid since $\beta_1(\lambda_1,\theta)/\beta_{-1}(\lambda_{-1},\theta) > 0$ holds on I because otherwise, one of the densities would be non-positive. Substituting the likelihood ratio and a as well as b into Bayes' theorem yields the posterior probabilities as

$$\begin{aligned}
P(\mathsf{y}=1 \mid \mathsf{f}(\mathsf{x})) &= \tfrac{1}{1+\frac{P(\mathsf{y}=-1)}{P(\mathsf{y}=1)}\cdot\left(\frac{p(\mathsf{f}(\mathsf{x})|\mathsf{y}=1)}{p(\mathsf{f}(\mathsf{x})|\mathsf{y}=-1)}\right)^{-1}} \\
&= \left(1+\tfrac{P(\mathsf{y}=-1)}{P(\mathsf{y}=1)}\cdot\exp\left(a\cdot\mathsf{f}(\mathsf{x})-\log\tfrac{\beta_1(\lambda_1,\theta)}{\beta_{-1}(\lambda_{-1},\theta)}\right)\right)^{-1} \\
&= \left(1+\exp\left(a\cdot\mathsf{f}(\mathsf{x})-\log\tfrac{\beta_1(\lambda_1,\theta)}{\beta_{-1}(\lambda_{-1},\theta)}+\log\tfrac{P(\mathsf{y}=-1)}{P(\mathsf{y}=1)}\right)\right)^{-1} \\
&= (1+\exp(a\cdot\mathsf{f}(\mathsf{x})+b))^{-1}
\end{aligned}$$

which proves the claim. □

Clearly, the respective decomposition is not unique as constants can be arbitrarily shifted between the factors. But for practical applications of the last result, there is no requirement for unique decompositions. Next, also the reverse direction can be proven.

Proposition 3.3. *Let $f\colon \mathcal{X} \to \mathbb{R}$ be a given binary classifier on the input domain $\mathcal{X}$. If the class-conditional densities exist and the posterior distributions can be expressed as a sigmoid function*

$$P(y=1 \mid f(x)) \;=\; \frac{1}{1+\exp(a\cdot f(x)+b)} \tag{3.5}$$

for $f(x) \in I \subseteq \mathbb{R}$, then there exist a function $\gamma : I \to (0,\infty)$ and constants $\alpha_i, \beta_i \in \mathbb{R}$ such that the class-conditional likelihoods can be expressed as

$$p(f(x) \mid y=i) \;=\; \gamma(f(x))\cdot\beta_i\cdot\exp(\alpha_i\cdot f(x)) \tag{3.6}$$

where $i = \pm 1$.

Proof. First, the sigmoid-shaped posterior implies that $p(\mathsf{f}(\mathsf{x}) \mid \mathsf{y}=i) > 0$ holds for $\mathsf{y} = \pm 1$. Substituting the given posterior into Bayes' theorem yields

$$\frac{1}{1+\exp(a\cdot\mathsf{f}(\mathsf{x})+b)} \;=\; \frac{1}{1+\frac{P(\mathsf{y}=-1)}{P(\mathsf{y}=1)}\cdot\frac{p(\mathsf{f}(\mathsf{x})|\mathsf{y}=-1)}{p(\mathsf{f}(\mathsf{x})|\mathsf{y}=1)}}$$

which is equivalent to:

$$p(\mathsf{f}(\mathsf{x}) \mid \mathsf{y} = -1) \;=\; p(\mathsf{f}(\mathsf{x}) \mid \mathsf{y} = 1) \cdot \frac{P(\mathsf{y} = 1)}{P(\mathsf{y} = -1)} \cdot \exp(a \cdot \mathsf{f}(\mathsf{x}) + b)$$

Thus, selecting $\alpha_1 \coloneqq 0$, $\beta_1 \coloneqq 1$ and $\gamma(\mathsf{f}(\mathsf{x})) \coloneqq p(\mathsf{f}(\mathsf{x}) \mid \mathsf{y} = 1)$ as well as $\alpha_{-1} \coloneqq a$ and $\beta_{-1} \coloneqq \frac{P(\mathsf{y}=1)}{P(\mathsf{y}=-1)} \cdot e^b$ yields

$$\begin{aligned} p(\mathsf{f}(\mathsf{x}) \mid \mathsf{y} = 1) \quad &= \quad \gamma(\mathsf{f}(\mathsf{x})) \;=\; \gamma(\mathsf{f}(\mathsf{x})) \cdot \beta_1 \cdot \exp(\alpha_1 \cdot \mathsf{f}(\mathsf{x})) \quad \text{and} \\ p(\mathsf{f}(\mathsf{x}) \mid \mathsf{y} = -1) \quad &= \quad \gamma(\mathsf{f}(\mathsf{x})) \cdot \beta_{-1} \cdot \exp(\alpha_{-1} \cdot \mathsf{f}(\mathsf{x})), \end{aligned}$$

respectively, which finalizes the proof. □

An important consequence of this result is that it enables to prove that Platt scaling is especially optimal for likelihoods distributed according to the following four different families of probability distributions.

Corollary 3.4. *Let $f\colon \mathcal{X} \to \mathbb{R}$ be a given binary classifier on the input domain $\mathcal{X}$. The posterior probability distribution $P(y = 1 \mid f(x))$ has the sigmoid form of* (3.4) *on $I \subseteq \mathbb{R}$ if there exist scalings $s_i \in \{-1, 1\}$ such that the class-conditional likelihoods for both classes $i = \pm 1$ are strictly positive and can be expressed as one of following distributions over I:*

1. *As* Gaussian distributions *with mean λ_i and standard deviation θ:*

$$p(f(x) \mid y = i) \;=\; \frac{1}{\sqrt{2\pi}\theta} \cdot \exp\left(-\frac{(f(x) - \lambda_i)^2}{2 \cdot \theta^2}\right) \tag{3.7}$$

2. *As shifted* exponential distributions *controlled by a parameter $\lambda_i > 0$ and a translation $t_i \in \mathbb{R}$:*

$$p(f(x) \mid y = i) \;=\; \lambda_i \exp(-\lambda_i \cdot (s_i \cdot f(x) + t_i)) \tag{3.8}$$

3. *As* gamma distributions *parameterized by $\lambda_i > 0$, $\theta > 0$ and the* gamma function $\Gamma(z) = \int_0^\infty u^{z-1} e^{-u}\,\mathrm{d}u$*:*

$$p(f(x) \mid y = i) \;=\; \frac{\lambda_i^\theta}{\Gamma(\theta)} \cdot (s_i \cdot f(x))^{\theta-1} \cdot \exp(-\lambda_i \cdot s_i \cdot f(x)) \tag{3.9}$$

4. *As* χ^2-distributions *of $\theta \in \mathbb{N}$ degrees of freedom:*

$$p(f(x) \mid y = i) \;=\; \frac{1}{2\Gamma(\frac{\theta}{2})} \cdot \left(\frac{s_i \cdot f(x)}{2}\right)^{\frac{\theta}{2}-1} \cdot \exp\left(-\frac{s_i \cdot f(x)}{2}\right) \tag{3.10}$$

Proof. The Gaussian and the exponential case are known to be valid and are only mentioned again for completeness. Still, it is straightforward to apply theorem 3.2 for them as well. For the gamma distribution, the likelihoods can be factorized as

$$p(\mathsf{f}(\mathsf{x}) \mid \mathsf{y} = i) \;=\; \frac{\lambda_i^\theta}{\Gamma(\theta)} s_i^{\theta-1} \;\cdot\; (\mathsf{f}(\mathsf{x}))^{\theta-1} \;\cdot\; \exp(-\lambda_i s_i \;\cdot\; \mathsf{f}(\mathsf{x})) \tag{3.11}$$

while for the χ^2-distributions a possible factorization is

$$p(\mathsf{f}(\mathsf{x}) \mid \mathsf{y} = i) \;=\; \frac{1}{2\Gamma(\frac{\theta}{2})} \left(\frac{s_i}{2}\right)^{\frac{\theta}{2}-1} \;\cdot\; (\mathsf{f}(\mathsf{x}))^{\frac{\theta}{2}-1} \;\cdot\; \exp\left(-\frac{s_i}{2} \;\cdot\; \mathsf{f}(\mathsf{x})\right) \tag{3.12}$$

such that the application of theorem 3.2 implies the claim. □

The scalings s_i as well as the translations t_i in case of the exponential distribution are only used to flip and translate the distribution's argument accordingly, without changing the integral to ensure that the function remains a valid probability density. They are not needed in the Gaussian case as here, the distribution is symmetric and the mean value parameter already controls the shift. It might also be useful to preprocess f before applying any of the previous results, for example by adding a constant or by scaling it. Only as soon as densities are estimated, these transformations have to be properly handled (i.e. back substituted) because they might result in the function not integrating to one anymore. Further, as the Erlang distribution is a special case of the gamma distribution, the last result holds for it as well.

A different important observation is that even if the sigmoid function has a symmetric shape, the class-conditional likelihoods do not have to be symmetric. For example in the exponential case, the parameters λ_1 and λ_{-1} are not constrained to be equal.

3.1.3 Platt Scaling and Beta Calibration

Beta calibration is based on the observation that Platt scaling aims at calibrating unbounded classifier scores, i.e. a reasonable probability can even be computed for $|\mathsf{f}| \to \infty$. This does not make much sense if the classifier is bounded, for example if it allows a probabilistic interpretation satisfying $\mathsf{f}(\mathsf{x}) \in [0, 1]$. Here, the authors proposed to fit beta distributions to the likelihoods in form of (2.20) yielding posterior probabilities given by (2.21).

A comparison between beta calibration and Platt scaling showed that the former outperformed the latter in an experiment on 41 data sets. However, using a probabilistic classifier in the respective experiments, which returns predictions in $(0, 1)$, the parametric assumptions of Platt scaling are violated. But a comparison in the inverse setting where Platt scaling's parametric assumptions hold, while the ones of beta calibration are violated remains open. Doing so requires to extend the beta distribution by zero to $\mathbb{R}$, while the parametric form of (2.21) is not valid anymore – the only reasonable posterior probability would be the constant class prior, simply because nothing is known from the parametric model. It is straightforward to see that applying this in practice unavoidably results in arbitrarily bad calibrations.

Hence, the correct conclusion from these experiments is that beta calibration is not superior to Platt scaling on probabilistic classifiers but applying the former on unbounded, real-valued classifiers is just as unreasonable as applying the latter on probabilistic ones. Instead, the question remains how to map both prediction functions to each other such that the application of both techniques is possible in any case, yielding a fair comparison. Here, the following result holds.

Proposition 3.5. *For the Platt scaling's calibration function $\sigma_{a,b}$, as given by* (2.7)*, and the beta calibration's one $\tau_{a,b}$, as given by* (2.21)*, hold the following identity:*

$$\sigma_{a,b}(z) \;=\; \tau_{-a,-b}\left((1 + \exp(-z))^{-1}\right) \tag{3.13}$$

Proof. Straightforward computation yields:

$$\begin{aligned}
\sigma_{a,b}(z) &= (1+\exp(a\cdot z+b))^{-1} \;=\; (1+\exp(b)\cdot(\exp(-z))^{-a})^{-1} \\
&= \left(1+\exp(b)\cdot\left(\tfrac{1+\exp(-z)-1}{1}\right)^{-a}\right)^{-1} \\
&= \left(1+\exp(b)\cdot\left(\tfrac{1-(1+\exp(-z))^{-1}}{(1+\exp(-z))^{-1}}\right)^{-a}\right)^{-1} \\
&= \left(1+\left(\exp(-b)\cdot\left(\tfrac{(1+\exp(-z))^{-1}}{1-(1+\exp(-z))^{-1}}\right)^{-a}\right)^{-1}\right)^{-1} \\
&= \tau_{-a,-b}\left((1+\exp(-z))^{-1}\right)
\end{aligned}$$

□

Thus, beta calibration is actually equivalent to Platt scaling. The only difference is a sigmoidal preprocessing $z \mapsto (1+\exp(-z))^{-1}$. Similarly, beta-calibrating a probabilistic classifier is equivalent to transforming the prediction to whole $\mathbb{R}$ using the sigmoid mapping's inverse $z \mapsto \log(z/(1-z))$ – also known as logit or log-odds – and applying Platt scaling thereafter. Interestingly, the latter fact was already observed in the work introducing beta calibration [Kull et al. 2017] to easily compute the parameters using existing algorithms to solve for Platt scaling or logistic regression. Similarly, the same transformation is used during the constructing of probability calibration trees [Leathart et al. 2017] to apply Platt scaling on the leaf nodes of logistic regression trees. Based on the provided insights, the respective approach can equivalently be described as applying beta calibration to the leaf nodes of logistic model trees.

Finally, comparing Platt scaling to beta calibration hardly makes any sense at all. Actually, this just means that a transformation function of the form (3.4) is fitted twice – once with a sigmoidal preprocessing and once again without it. So comparing them actually means to analyze the influence of the preprocessing, but not between different techniques. This question might be particularly interesting for model-free calibration approaches where the sigmoid transformation serves as a form of *precalibration* to approximate the calibration function on the compact unit interval only. Even though some works assume the latter setting by using a sigmoid [Naeini 2016; Naeini & Cooper 2015, 2016, 2018] or a linear [Zadrozny & Elkan 2002] transformation, the provided insights show that the situation here actually is more complex. In light of this, the empirical evaluations in section 3.4 will also compare binning performed on unbounded scores to sigmoidally precalibrated ones. It should also be noted that the precalibration function is a non-optimized variant of Platt scaling obtained by simply selecting $a = -1$ and $b = 0$.

3.1.4 Parameter Estimation

Finally, applying parametric calibration techniques in practice should also be handled carefully, especially if its results are to be criticized. Based on the proven equivalence between Platt scaling and beta calibration as well as the fact that there is no other well-accepted parametric calibration approach, the following discussion is strongly focused on Platt scaling. Still, the main statements are valid for any parametric approach in similar reformulations.

To find the optimal values for a and b in Platt scaling's calibration function (3.4), an optimization problem is formulated whose solution yields the two parameters. This

optimization problem contains different degrees of freedom that influence the optimal solution. In particular, Platt himself fitted the parameters by a maximum likelihood approach, iteratively minimizing the following cross-entropy error

$$L(a,b) \;=\; -\sum_{i=1}^{r} \left[t_i \cdot \log(\sigma_{a,b}(\mathsf{f}(\mathsf{x}_i))) \;+\; (1-t_i) \cdot \log(1-\sigma_{a,b}(\mathsf{f}(\mathsf{x}_i)))\right] \tag{3.14}$$

where the target values t_i are defined as $t_+ = (r_+ + 1)/(r_+ + 2)$ for all positive instances, while the target values for the negative instances are $t_- = 1/(r_- + 2)$. Here, r_+ and r_- refer to the overall number of positive and negative instances. Finding a and b in this way is generally known as Platt scaling. However, differently parameterizing the optimization problem – for example by using the squared error between t_i and $\sigma_{a,b}(\mathsf{f}(\mathsf{x}_i))$ – will result in a different solution and thus a different calibration mapping. In light of this, Platt scaling should generally refer to *any* calibration mapping using a transformation function of the form (3.4). The question how to estimate the parameters is a subsequent implementation detail but does not directly deal with the general validity of a transformation whose parameters are properly estimated.

Next, fitting the parameters a and b indirectly corresponds to estimating the likelihoods $p(\mathsf{f} \mid \mathsf{y} = \pm 1)$. This is advantageous in comparison with a *direct* likelihood estimation since, as shown by theorem 3.2, the sigmoidal posterior is valid for different likelihood models and thus fitting the posterior directly avoids to specify a probability distribution whose parameters are fitted. Even though some well-known issues exist [Lin et al. 2007] in this domain that have been handled in available implementations [Chang & Lin 2011], directly fitting the likelihood distribution however might involve selecting a real subset $I \subset \mathbb{R}$ (usually an interval) for each class where the respective parametric model is expected to be valid. For example even in Platt's introductory work [Platt 1999], the exponential distributions are observed for all data points x_i satisfying $\mathsf{y}_i \cdot \mathsf{f}(\mathsf{x}_i) \leq 1$. Consequently, the resulting sigmoid-shaped posterior probability generally will only be valid where each likelihood is distributed according to the respective distribution. In this particular example, this is exactly where $|\mathsf{f}(\mathsf{x})| \leq 1$ holds. For all predictions $\mathsf{f}(\mathsf{x}) \notin I$, the likelihood model is invalid.

Furthermore, such proper handling would even mean that while fitting the sigmoid parameters, only data points $\mathsf{f}(\mathsf{x})$ whose predictions lie in a valid range of both likelihoods are used to estimate the parameters. If this does not hold but the respective data points are still used to estimate the sigmoid parameters, it is implicitly assumed that the sigmoid shape is valid *even if the respective likelihoods are not*. This can be reasonable, for example at large predictions that correspond to extreme probabilities of ≈ 0 or ≈ 1 since this is consistent with the sigmoid's limits for $\mathsf{f}(\mathsf{x}) \to \pm\infty$. In practice it is hard to always check which instances should be used, which might explain why this issue is often ignored. Even as long as all points are used and the results are reasonably well, the practitioner does not really need to care about it. However, a general justification to always include all data points in the parameter fitting simply does not exist and thus at least should be respected as soon as suboptimal results are obtained.

3.2 New Calibration Techniques

The previous results focusing on monotonic calibration showed that Platt scaling and isotonic regression both have a solid theoretical background explaining good results in many applications where a monotonic assumption can be reasonably defended. In particular with respect to Platt scaling, detailed theoretical insights about its

validity are given. Still, there might be applications where a calibration function should be created *non-monotonically*. In this regard, this section aims explicitly at non-monotonic calibration options.

Here as a first observation, the large majority of existing calibration techniques are based on direct transformations of the scores $\mathsf{f}(\mathsf{x})$ to posterior estimates $p(\mathsf{f}(\mathsf{x}))$ without estimating likelihood distributions $p(\mathsf{f} \mid \mathsf{y})$. However, the discussion in subsection 3.1.4 showed that estimating parametric calibration functions also indirectly estimates the likelihoods. This can be advantageous for several previously discussed reasons. Still by using Bayes' theorem,

$$P(\mathsf{y} = 1 \mid \mathsf{f}) \;=\; \frac{p(\mathsf{f} \mid \mathsf{y} = 1) \cdot P(\mathsf{y} = 1)}{p(\mathsf{f} \mid \mathsf{y} = 1) \cdot P(\mathsf{y} = 1) \;+\; p(\mathsf{f} \mid \mathsf{y} = -1) \cdot P(\mathsf{y} = -1)} \tag{3.15}$$

any density estimation technique can be converted into an *indirect* calibration algorithm. The prior probabilities are estimated by the respective frequencies on the training data and the likelihoods using the respective density estimation technique. It should be emphasized that this approach can also be used for more than two classes, however the estimation of the densities $p(\mathsf{f} \mid \mathsf{y})$ becomes multi-dimensional. This can suffer from the same problems as posterior class estimation based on Bayes' theorem itself, depending on the dimension of f.

Even though this is a straightforward application of Bayes' theorem, this option is relatively rarely discussed in the literature. Some works explicitly mention Bayes' theorem as an explicit option for calibration [Gebel 2009], however focusing on parametric density estimation. Clearly, arbitrary parametric models can be applied, which of course is highly problem-dependent. Thus, particularly interesting are model-free density estimation techniques that allow a larger flexibility as they do not restrict to a fixed parametric model. As a result, they will also be non-monotonic in general. Typical problems caused by the input dimension are not present in binary calibration since the densities $p(\mathsf{f} \mid \mathsf{y})$ remain one-dimensional. Besides aforementioned works on binning – where direct posterior estimation based on histograms is equivalent to estimating the likelihoods by binning and applying Bayes' theorem – only a few other works apply Bayes' theorem for calibration based on non-parametric approaches. Existing works cover density estimation based on Polya trees [Connolly et al. 2017], Dirichlet process modeling [Naeini 2016] and kernel density estimation [Naeini 2016], however the latter only using a constant kernel and Silverman's [Silverman 1986] potentially suboptimal bandwidth estimation.

3.2.1 Kernel Density Estimation

The key idea of kernel density estimation (KDE) [Parzen 1962; Wasserman 2006] is to transform a *discrete* unlabeled sample set $D = \{\mathsf{x}_i \;:\; i = 1, \ldots, r\} \subset \mathbb{R}^n$ into an estimate of the *continuous* density function. With respect to binary classifier calibration, this will consist of the one-dimensional predictions $\mathsf{f}(\mathsf{x})$ from each of the two classes, respectively. Additionally, let $K(z)$ be a kernel function, i.e. $K(z) \geq 0$ and $\int_{-\infty}^{\infty} K(z)\,\mathrm{d}z = 1$ hold. The kernel density estimator is the function

$$\hat{f}(z) \;=\; \frac{1}{r \cdot h} \sum_{i=1}^{r} K\left(\frac{z - z_i}{h}\right). \tag{3.16}$$

where the real-valued bandwidth $h > 0$ is a free parameter that has to be selected accordingly. There are a few remarkable properties that make kernel density estimation a powerful alternative for classifier calibration. It can be shown [Wasserman

2006] that given the optimal bandwidth h^*, the $\mathcal{L}^2$ risk is bounded by $\mathcal{O}(r^{-\frac{4}{5}})$

$$\int \mathbb{E}\left(f(z) - \hat{f}(z)\right)^2 \mathrm{d}z \;\in\; \mathcal{O}(r^{-\frac{4}{5}}) \tag{3.17}$$

under relative mild assumptions, where f denotes the unknown, true density function. Consequently, the estimator converges in $\mathcal{L}^2$ and thus also in probability to the true density function. The bandwidth can be interpreted as an equivalent counterpart of the bin size that controls the amount of local smoothing performed by the kernel function. The concrete selection of the latter usually is less important than the bandwidth selection, still the kernel defines the analytic properties of the transformation function.

It can be shown that the optimal bandwidth scales with $\mathcal{O}(r^{-\frac{1}{5}})$, but the constant depends on the unknown distribution [Venables & Ripley 2002]. However, in contrast to relatively arbitrary selecting of bin sizes, there exist useful heuristics that can be applied instead to estimate h. The same authors also motivate to apply the Sheather-Jones bandwidth estimation [Sheather & Jones 1991] as being "close to optimal" and thus, preferable to the aforementioned Silverman heuristic.

The next degree of freedom is the selection of the kernel. Using a constant one results in a discontinuous calibration map, which can be counterintuitive. This could be resolved by a Gaussian kernel (which would even yield an infinitely often differentiable calibration function) but instead the Epanechnikov[1] kernel [Epanechnikov & Seckler 1969]

$$K_{\mathrm{Ep}}(z) \;=\; \begin{cases} \frac{3}{4}(1 - z^2) & |z^2| \leq 1 \\ 0 & \text{otherwise} \end{cases} \tag{3.18}$$

minimizes the mean integrated squared error over all kernels [Zambom & Dias 2012], however only using a bandwidth that depends on the unknown density [Cybakov 2009]. But it still justifies to select it over a constant kernel since this yields a continuous calibration mapping. Furthermore, the unknown, optimal bandwidth can be approximated in practice using the aforementioned heuristics.

Finally, another improvement aims at a general problem prevalent in any density-based calibration technique. Similar to the proof of theorem 3.2, the posterior distribution can completely be expressed

$$P(\mathsf{y} = 1 \mid \mathsf{f}(\mathsf{x})) \;=\; \left(1 + \frac{P(\mathsf{y} = -1)}{P(\mathsf{y} = 1)} \cdot \left(\frac{p(\mathsf{f}(\mathsf{x}) \mid \mathsf{y} = 1)}{p(\mathsf{f}(\mathsf{x}) \mid \mathsf{y} = -1)}\right)^{-1}\right)^{-1} \tag{3.19}$$

using the likelihood ratio. Even if the priors are assumed as correctly estimated, the likelihood ratio preserves no information about the magnitude of the individual likelihoods. If both are reasonably large, the ratio is a good estimate. However, if at least one of them approaches zero, unavoidable minor inaccuracies can have large influences. Especially if both are very small, the ratio can take arbitrary large values caused by random influences, which are completely unjustified based on the previously observed training data. To resolve this issue, the class-conditional likelihoods are not only estimated based on the training data, but dynamically extended with the instance in question, x_0, with predicted score $\mathsf{f}_0 = \mathsf{f}(\mathsf{x}_0)$. This is feasible as its class value is not required while estimating the densities. Thus, the proposed approach can be summarized as follows:

[1] It should be noted that the kernel function is sometimes differently scaled. This is not directly important from the practical point of view as software implementations (for example these in R) scale the kernels such that their standard deviations equal the bandwidth.

1. Compute the training data sets $F_i = \{\mathsf{f}(\mathsf{x}) : (\mathsf{x}, i) \in D\}$ for both classes $i = \pm 1$ at training time.
2. At prediction time, extend both sets by $\mathsf{f}_0 = \mathsf{f}(\mathsf{x}_0)$ to form $\overline{F_i} = F_i \cup \{\mathsf{f}_0\}$.
3. Estimate both likelihoods using kernel density estimation (3.16).
4. Estimate the posterior probabilities for x_0 using Bayes' theorem by substituting the obtained likelihoods into (3.19).

If the training data reasonably covers f_0, there will unlikely be a large influence by dynamically extending the training data. On the reverse, if f_0 is not well covered by the training data, the proposed approach lower bounds the estimated density. In a Bayesian sense, this approach can be interpreted as a uniform prior over f.

3.2.2 Ensemble of Kernel Density Estimation

The last section motivated to use extended variants of kernel density estimation as a classifier calibration technique, which mainly generalizes the idea to transform density estimation techniques into classifier calibration ones, but to be able to circumvent the intrinsic drawbacks of binning, namely the choice of the bin size and the discontinuous calibration function. The rationale behind calibration techniques like ENIR and ELiTE as well as their predecessors can be summarized by instead of using only *one* calibration binning model, multiple ones are combined. In case of ELiTE, instead of binning models an ensemble of piecewise linear mappings is computed. The prediction rule can jointly be described as

$$P(\mathsf{y} = 1 \mid \mathsf{x}) \;=\; \sum_{i=1}^{t} \frac{\Psi(M_i)}{\sum_{j=1}^{t} \Psi(M_j)} \cdot P(\mathsf{y} = 1 \mid \mathsf{x}, M_i) \tag{3.20}$$

where Ψ is an accordingly selected scoring function and $P(\mathsf{y} = 1 \mid \mathsf{x}, M_i)$ refers to the i-th model's posterior probability. In case of SBB, the scoring is binary since actually only a single model is selected, still it can be expressed in form of (3.20). The respective evaluations showed that this resulted in superior recognition results in comparison with only a single model.

Here, it is important to note that this strategy to combine different techniques is not constrained to be applied to binning models only and, because kernel density estimation can also be interpreted as a continuously generalized variant of binning, it is interesting to analyze whether ensemble strategies can also improve the results here. This raises two questions: First, how to create such an ensemble or equivalently, how to choose a reasonable set of parameterizations and second, how to select the scoring function Ψ, i.e. how to combine these predictions?

Possible options to create an ensemble are the kernel function and the bandwidth parameter. Because based on aforementioned results it is known that the bandwidth has more relevance and is proportional to $r^{-\frac{1}{5}}$, a starting point can be estimated using the Sheather-Jones heuristic that thereafter can be scaled using a predefined set of factors $\{s_1, s_2, \ldots, s_m\}$. Clearly, the selection of the latter remains arbitrary in general, but it is still reasonable to neither select them too large nor too small as doing so is likely to over- or under-smooth the estimation function. Thus, a small grid $\{0.5, 0.75, 1, 1.5, 2\}$ is used where each scaling factor is used for both, the Gaussian

and the Epanechnikov kernel. Hence, an ensemble of $t = 10$ kernel density estimation predictors is obtained[2].

After estimating a posterior probability with each individual model, the second step is to combine the models, i.e. to select the scoring function Ψ. Because each kernel density estimation model depends on all data, there are no varying degrees of freedom. As the log-likelihood is obtained by using the AIC or BIC scorings and removing the degrees of freedom, here a kernel density estimation model M_i is scored using the model's log-likelihood $\Psi(M_i) = \log p(D \mid M_i)$. Finally, this yields the overall posterior probability estimation by using (3.20).

3.3 Evaluation Metrics

The previous sections analyzed classifier calibration from a theoretical point of view and presented different results for both, monotonic and non-monotonic approaches. With respect to practical applications, there is a straightforward demand to evaluate a calibration technique's predictive performance, for example to choose the best out of a set of available ones. For this purpose, an evaluation metric is required to compare the estimated probability to the true one. Since the true posterior probabilities are unknown, a direct error cannot be computed. Thus, some surrogate error functions have to be used instead that are only based on a probability estimate $p(\mathsf{x})$ of a given test instance x and its true class value y.

3.3.1 Classification Metrics and Proper Scoring Rules

It is interesting to observe that even in the three works that introduced the most frequently mentioned calibration techniques Platt scaling [Platt 1999], histogram binning [Zadrozny & Elkan 2001b] and isotonic regression [Zadrozny & Elkan 2002], there is no well-accepted standard how calibration techniques should be evaluated and compared. The first option is to interpret the posterior estimate as a classification algorithm, i.e. to use well-accepted metrics like the classification rate or the receiver-operator characteristic, summarized in the AUC statistic. Accuracy or equivalently error statistics are included in all three aforementioned works. Besides these, either the log-loss [Platt 1999; Zadrozny & Elkan 2001b] or the Brier score [Zadrozny & Elkan 2001b, 2002] is used to evaluate the posterior probability estimates. Both metrics are instances of the infinitely large family of proper scoring rules [Merkle & Steyvers 2013] but the most commonly used ones [Kull & Flach 2015]. Both metrics refer to the general case of k different classes and require a posterior probability estimate $p = p(\mathsf{x}) \approx P(\mathsf{y} \mid \mathsf{x})$ and a binary class vector y (i.e. the respective class is encoded with a one). This enables to compute the log-loss φ^{LL} and the Brier score φ^{BS} as

$$\varphi^{\mathsf{LL}}(p, \mathsf{y}) = -\log p_{\mathsf{y}} \quad \text{and} \quad \varphi^{\mathsf{BS}}(p, \mathsf{y}) = \sum_{i=1}^{k} (p_i - \mathsf{y}_i)^2, \tag{3.21}$$

respectively. Brier score is equivalent to the mean squared error (MSE), which is sometimes used instead[3]. Both metrics are minimized by the posterior probabilities, however this requires to integrate over the whole distribution, i.e. only holds asymptotically. Still, it justifies to use them as an evaluation metric that has to be minimized

[2] It should be added that additionally each KDE calibration model consitis of two density estimators, one for each class.

[3] It should be added that some authors prefer to use the *root* mean squared error (RMSE).

over a test data set. However, the absolute error value is hardly interpretable and non-zero, even if all instances would be predicted correctly.

An important difference between log-loss and Brier score in the general, multi-class case is that the former only depends on the probability of the true class, while the latter is also influenced by varying probabilities besides y. This property is known as *locality* [Bickel 2010] of the log-loss, but obviously irrelevant in the binary case. Next, the Brier score is bounded, while the log-loss has a singularity at 0. This is problematic as a single outlier can cause an infinitely large overall loss. However, the Brier score also has different limitations for very rare or very frequent events [Benedetti 2010]. The same work shows that the Brier score in fact is a second-order approximation to the log-loss.

3.3.2 Bin-based Evaluation Metrics

Both aforementioned scoring functions have a long history in forecast evaluation [DeGroot & Fienberg 1983] and have been used in subsequent work on classifier calibration [Bella et al. 2009a; Bennett 2006; Niculescu-Mizil & Caruana 2005b], but interestingly some authors began to additionally use bin-based metrics [Bella et al. 2009b; Niculescu-Mizil & Caruana 2005b]. Here, the core idea is to discretize the unit interval into m bins $B_1, B_2, \ldots, B_m$ such that for each bin B_i, the average predicted probability can be compared to the fraction of instances from the respective class. If the bin is sufficiently small, e.g. $[p_0 - \varepsilon, p_0 + \varepsilon]$ for a certain fixed probability p_0, then the prediction is well calibrated if the fraction of class-1 occurrences in the bin equals p_0, at least asymptotically and sufficiently small ε. So the predicted average probability $e(B_i)$ of a bin can be computed and compared to the fraction of observed class-1 instances $o(B_i)$.

This idea is used to define two evaluation metrics, *expected calibration error* (ECE) and *maximum calibration error* (MCE) [Naeini et al. 2014]. Given a data set D, these are defined as

$$\mathsf{ECE}(D) = \sum_{i=1}^{m} \frac{r_i}{r} \cdot |o(B_i) - e(B_i)| \quad \text{and} \quad \mathsf{MCE}(D) = \max_{i=1,\ldots,m} |o(B_i) - e(B_i)|, \quad (3.22)$$

respectively, where r_i refers to the number of data points in the i-th bin B_i and r to the overall number (i.e. $\sum_{i=1}^{m} r_i = r$). These metrics are regularly used in recent works on classifier calibration, by the introducing authors [Jabbari et al. 2017; Naeini 2016; Naeini et al. 2015a; Naeini & Cooper 2015, 2016, 2018; Naeini et al. 2015b] as well as other works [Guo et al. 2017; Seo et al. 2019; Tran et al. 2018; Wang et al. 2019]. In many of these works, bin-based metrics are used as main evaluation criteria, while others like AUC and sometimes the MSE or RMSE are given, too.

Bin-based evaluation metrics can also be related to *reliability diagrams* [Jiang et al. 2012]. These are used to visualize calibration where the observed fractions in the bins are plotted to illustrate deviations from perfect calibration, i.e. the $x = y$ line.

3.3.3 Analyzing Evaluation Metrics

In summary, these three families of evaluation metrics are commonly used. Classification-based ones like the AUC are useful to control the calibration to avoid pathological solutions (i.e. constant predictors), but definitely are not useful in comparing calibration techniques as the computed probabilities are mainly ignored. Only the ranking or the location to decision threshold are relevant. Further, there is no need to apply classifier calibration at all if the aim is to maximize the accuracy or the AUC.

Thus to directly evaluate the probabilities, there are two common families of metrics, proper scoring rules and bin-based ones.

The justification for introducing bin-based metrics in probabilistic predictions over proper scoring rules has deeper reasons that boil down to the fact that proper scoring rules can be decomposed into *calibration* and *refinement loss*. In practice, it is common to evaluate both parts independently [Parmigiani & Inoue 2009], which explains and justifies the introduction of the bin-based metrics. In particular, they are used to measure the calibration loss, while the refinement loss is evaluated using statistics like the AUC. However, it should be emphasized that both losses cannot be computed directly. In fact, they are expected values of random variables over the unknown distribution. The same holds for other decompositions of proper scoring rules, which lies beyond the scope of this work and is discussed in the respective literature [Kull & Flach 2015].

Even though this justifies to not use proper scoring rules as general probabilistic classification evaluation metrics, the situation of comparing different calibration techniques for a given, fixed classifier is inherently different. First, the refinement loss is mainly optimized during the training of the underlying classifier that is completely independent of the calibration optimizations. Even if the calibrated probabilities' refinement loss does not have to coincide with the classifier's one, it is still reasonable to assume that the calibration only slightly influences the AUC, a strictly monotonic transformation mapping is even guaranteed to preserve it. Thus, assuming only minor influences on the refinement loss, comparing error values of proper scoring rules after applying calibration mainly means to compare the calibration losses. Therefore, they are an appropriate choice as classifier calibration evaluation metrics.

Inappropriateness of Bin-based Metrics

On the other hand, there are three major omnipresent problems related to the bin-based metrics. First, the selection of the number of bins (or equivalently, the bin size) remains arbitrary. Some consequences of this issue and their influence to the error values have already been observed and discussed in relatively recent work [Nixon et al. 2019]. Second, the concrete probability values inside a bin are discarded. It is only a metric measuring the closeness of averaged but not individual probabilities. Symmetrically miscalibrating all predictions inside a bin will obviously change the calibration, but not the ECE or MCE as the means remain unchanged. Thus, even an error of zero does not mean that all probabilities are perfectly calibrated.

The third and major problem lies in the construction of the reference binning model and the corresponding error function $|o(B_i) - e(B_i)|$ itself. Computing the error function values in this way consists of a two-step procedure. The first step constructs the bins B_i and their reference probabilities $o(B_i) =: p_{\text{ref}}(\mathsf{x}; B_i)$ (to make the dependency on x explicit, even though they are constant for a given B_i) as the fraction of class-1 instances from the test data inside the bin B_i, which is the same as applying binning calibration to the test data probabilities $p(\mathsf{x}_j),\ j = 1, \ldots, r$, obtained from the model and the calibration function in question. Using the identity

$$\sum_{j:p(\mathsf{x}_j)\in B_i} \mathbb{1}(\mathsf{y}_j = 1) \;=\; \sum_{j:p(\mathsf{x}_j)\in B_i} p_{\text{ref}}(\mathsf{x}_j; B_i) \tag{3.23}$$

for the binning reference model (on the left-hand side, binary values are accumulated, and their average is summed on right-hand side), in the second step the error is simply computed as the discrepancy between the probabilities obtained from the reference

model and the original model in question, respectively:

$$\begin{aligned} |o(B_i) - e(B_i)| &= \left| \frac{1}{r_i} \cdot \sum_{j:p(\mathsf{x}_j)\in B_i} \mathbb{1}(\mathsf{y}_j = 1) \;-\; \frac{1}{r_i} \cdot \sum_{j:p(\mathsf{x}_j)\in B_i} p(\mathsf{x}_j) \right| \\ &= \left| \frac{1}{r_i} \cdot \sum_{j:p(\mathsf{x}_j)\in B_i} p_{\text{ref}}(\mathsf{x}_j; B_i) \;-\; \frac{1}{r_i} \cdot \sum_{j:p(\mathsf{x}_j)\in B_i} p(\mathsf{x}_j) \right| \\ &= \frac{1}{r_i} \cdot \left| \sum_{j:p(\mathsf{x}_j)\in B_i} (p_{\text{ref}}(\mathsf{x}_j; B_i) \;-\; p(\mathsf{x}_j)) \right| \end{aligned} \tag{3.24}$$

The remaining difference between ECE and MCE only is how the bin-wise errors are accumulated. For MCE, only the maximum average absolute error is computed, while substituting (3.24) into the definition (3.22) yields:

$$\mathsf{ECE}(D) \;=\; \frac{1}{r} \cdot \sum_{i=1}^{m} \left| \sum_{j:p(\mathsf{x}_j)\in B_i} (p_{\text{ref}}(\mathsf{x}_j; B_i) \;-\; p(\mathsf{x}_j)) \right| \tag{3.25}$$

Consequently, actually only the signed differences per instance are accumulated inside each reference model's bin. Similarly, the same approach can easily be generalized for every calibration technique by using it to compute the reference probabilities $p_{\text{ref}}(\mathsf{x})$ and substitute them into equation (3.24). Here, especially the sample grouping is no longer required and thus, also emphasizes the problematic and arbitrary sample grouping involved in binning. Additionally, there is no justification to use the same sample grouping for both steps, reference model estimation *and* error computation. In this sense, the error computation overfits to the reference model.

Besides this, it is questionable to perform the reference model estimation on the probabilities $p(\mathsf{x})$ instead of the underlying classifier's predictions $\mathsf{f}(\mathsf{x})$. The former adapts the reference model based on the predicted probabilities and thus directly depends on the previously estimated calibration model that is to be evaluated. This means that using bin-based evaluation metrics to compare *different* calibration models based on the *same* classifier outputs yields error values that are computed using *different* reference models. Thus, the overall errors are not as comparable as they would be if the reference model was the same and independent of the probabilities in question. To remedy these issues, it could be an option to omit the sample grouping in the ECE and MCE computation and accumulate the pointwise absolute differences instead, as well as replace the reference probability with the true class vector to be independent of the reference binnig model. This results in an error metric similar to the Brier score except that the absolute instead of squared errors are computed. Even though this seems to be a valid alternative, it results in an improper socring rule and thus is also unjustified.

In light of the analysis provided, bin-based evaluation metrics do nothing but apply calibration by binning on the test data's probabilities $p(\mathsf{x}_i),\ i = 1, \ldots, r$, to evaluate the previous calibration on the training/validation data and, therefore, simply lose any justification to be used as a calibration error function. They can still be useful to evaluate the calibrateness of a classifier that has not been optimized to produce calibrated probabilities – just like other calibration techniques. But as soon as these techniques were already used to explicitly postprocess the outputs into calibrated probabilities, it is unreasonable to apply one of these techniques again to evaluate itself.

Furthermore, these insights also may explain that bin-based metrics tend to prefer bin-based calibration techniques because there is an interesting coincidence between the introduction of bin-based state-of-the-art calibration techniques like ENIR and bin-based evaluation metrics, as discussed in full detail in chapter 2.

On the contrary, the problems described above are not present with the proper scoring rules. However, a prevalent problem for both, proper scoring rules and bin-based metrics, is the fact that they assume a representative test set. Proper scoring rules are at least provably valid if an integration over the whole data-generating distribution is possible, but in practice only the test data can be used instead. Thus, predicting the test data's empirical posterior probabilities $P_{\text{emp}}(\mathsf{y} \mid \mathsf{x})$ or $P_{\text{emp}}(\mathsf{y} \mid \mathsf{f}(\mathsf{x}))$, respectively, will minimize the proper scoring rules on the test data. There simply is no known error function that can identify the true posterior probabilities using finite data that do not provide information about the actual probabilities. Thus, it is only partly possible to compare calibration techniques at all. But if doing so, currently the only reasonable error metrics are proper scoring rules.

This statement can further be supported if the perfectly calibrated probabilities $P(\mathsf{y} \mid \mathsf{f})$ were given at testing time. It would be straightforward to compare the estimated probabilities to the true ones using a regression error function $L(p, q)$. Using the squared error $L^2(p, q) = \|p - q\|_2^2$ recovers the Brier score, while using the Kullback-Leibler divergence[4] $\mathsf{KL}(p, q) = \sum_{i=1}^{k} q_i \log \frac{q_i}{p_i}$ recovers the log-loss, respectively, using a binary vector q encoding the given class value. Thus, proper scoring rules are equivalent to using well-known error metrics, only restricted to binary reference vectors.

3.4 Comparison of Calibration Techniques

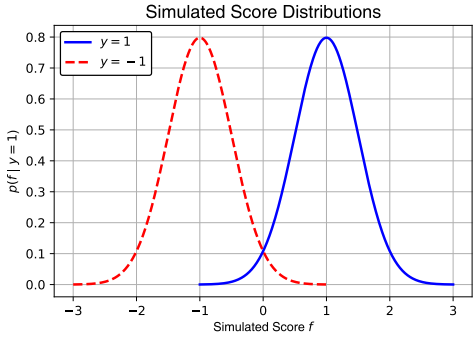
The preceding sections presented several theoretic results where for practical applications, the provided insights are particularly relevant as recent reference studies rely on unjustified and invalid evaluation metrics. As presented in the summary of existing results in chapter 2, there are different empirical comparisons of classifier calibration techniques in the respective literature available, however all of them expose certain drawbacks. In this regard, this section performs different experiments to empirically compare the state-of-the-art techniques and to analyze several open issues related to classifier calibration applied in practice.

As on real-world data true reference probabilities are not available, the first part of the empirical comparison uses distributions to sample the data such that after comparing the different calibration techniques, the true posterior probabilities computed from the distribution can be used as a reference. Thereafter, different real-world data sets are evaluated.

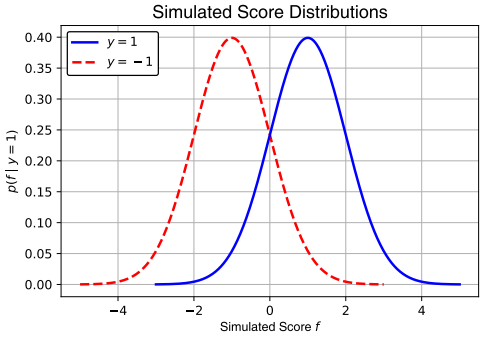
3.4.1 Simulation Studies

The first empirical evaluations are performed using artificially generated data. Interestingly, these are rather uncommonly applied in the classifier calibration literature, but are at least in some even relatively recent works [Naeini 2016; Naeini et al. 2015a; Naeini & Cooper 2015, 2018; Naeini et al. 2014, 2015b]. In particular, the authors simulated different two-dimensional data sets that are correctly predictable using a sufficiently parameterized support vector machine (a spherical and an XOR example).

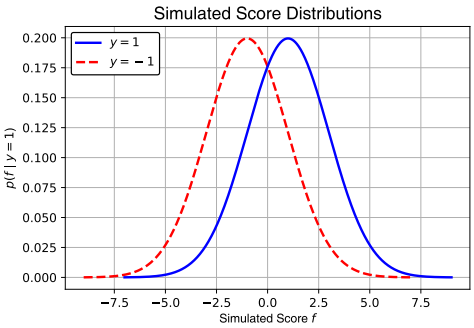
[4] It should be emphasized that in this definition, the role of p and q is swapped for consistency.



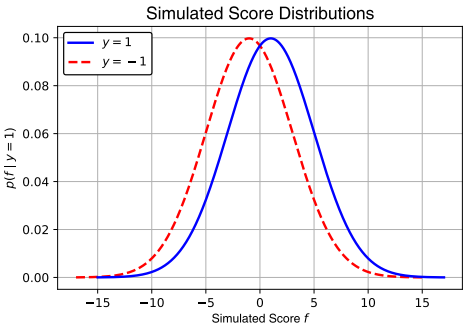
(a) Simulated scores with standard deviation $\sigma = 0.5$.



(b) Simulated scores with standard deviation $\sigma = 1$.



(c) Simulated scores with standard deviation $\sigma = 2$.



(d) Simulated scores with standard deviation $\sigma = 4$.

FIGURE 3.1: Illustration of the simulated score distributions used in the first two calibration experiments.

A similar study is performed by other authors [Jiang et al. 2011] where Gaussian-distributed data were generated and evaluated by training a logistic regression model on it. The biggest advantage of simulated data sets is that the true data-generation distribution is known and thus, the predicted probabilities directly can be compared to the true ones. Still, in the following part the *score* instead of *data* distributions will be selected and sampled. Interestingly, respective reference results are not available.

First Experiment: Calibration Accuracy

Based on aforementioned reasoning, this section applies different state-of-the-art classifier calibration techniques under artificially generated, optimal conditions to exclude any influence of classifier training algorithms or sampling issues. For this, the calibration data $(\mathsf{f}_i, \mathsf{y}_i)$ were generated, for each of the two classes $\mathsf{y} = \pm 1$ by sampling from Gaussian distributions with mean $\mu = \mathsf{y}$, standard deviation σ and 100, 1000, 10000 and 100000 samples per class. The respective distributions are illustrated in figure 3.1. The standard deviation was selected as 0.5, 1, 2 and 4, yielding 16 artificial data sets in total. This setting has two important properties: First, the true posterior probabilities can be computed as $P(\mathsf{y} = 1 \mid \mathsf{f}) = \left(1 + \exp\left(-2 \cdot \mathsf{f}/\sigma^2\right)\right)^{-1}$ and thus second, a parametric model in the form of (3.4) is provably optimal. For $\sigma = 1$, this sigmoid function is illustrated in figure 3.5b. All techniques are evaluated using a 10-fold random, stratified cross validation on each data set, where the same partitions are used for all techniques to preserve comparability.

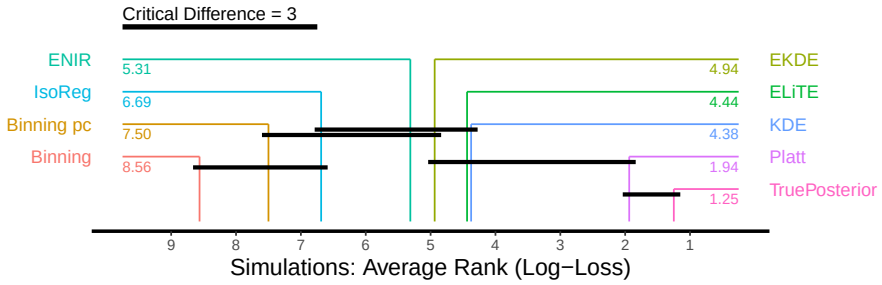
On these generated data sets the following calibration techniques were compared: ENIR and ELiTE as the most recent state-of-the-art approaches from the literature, isotonic regression and Platt scaling as monotonic and strictly monotonic, respectively, techniques. Additionally, two binning variants were applied: One directly bins the real-valued decision function into bins of size 0.1, another performs a sigmoidal *precalibration* (pc) using $z \mapsto (1 + \exp(-z))^{-1}$. Furthermore, the presented variant of KDE and the newly introduced technique EKDE were applied as techniques that are based on continuous density estimation, both also with a sigmoidal precalibration to better focus on the decision boundary during bandwidth estimation. Additionally, also the "calibration oracle" has been used that directly maps the score f to the true posterior probability. It should be emphasized that the difference between Platt scaling and directly predicting the true posteriors only lies in computing the parameters from data, or analytically from the known distributions. Thus, the predictions between these two are close, whereas it is still especially interesting to analyze their differences under the evaluation metrics.

Different evaluation metrics have been used to evaluate the respective probability estimates. Based on the insights of section 3.3, the proper scoring rules log-loss and Brier score are used as the main evaluation criteria for classifier calibration. Additionally, the bin-based metrics ECE and MCE were computed as well. As they depend on an arbitrarily selected bin number, both have been computed for varying bin numbers of 10, 20, 30, 50 and 100, respectively. It was also tried to construct a more robust bin-based error metric by averaging the former ten individual ones. Finally, the Kullback-Leibler divergence and the squared error L^2 between the true posterior probabilities and the actually predicted ones are computed. Both can be interpreted as the continuous equivalents of the two proper scoring rules as discussed in section 3.3, while explicitly evaluating them is only possible in simulation studies where the true posterior probabilities are known. Here, it is very interesting to analyze the correlations between these direct and the other calibration error metrics as the latter are the only feasible choices as soon as real-world data are evaluated. Consequently, the calibration oracle necessarily always yields zero error under the direct ones.

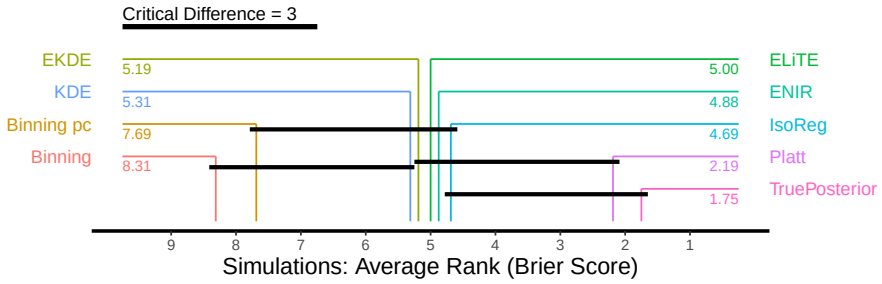
To compare the algorithms using any of the evaluation metrics over all data sets, an established standard procedure was applied [Demšar 2006]. In particular, the methods are ranked on each data set according to their errors, where the best performing technique is ranked first and ties are broken by assigning average ranks. Next, average ranks over all data sets are computed and combined with a non-parametric Friedman test to test whether there are significant differences in the average ranks, using the default $\alpha = 0.05$ significance level. After rejecting the null hypotheses, the post hoc Nemenyi test has been used, too.

The respective results are illustrated in critical difference diagrams in figures 3.2 and 3.3. In the former, the average ranks together with the critical difference are illustrated for the log-loss and the Brier score as well as the Kullback-Leibler divergence and the L^2 error to the true probabilities, while in the latter, the bin-based averaged ranks are presented for ten bins and the overall average over all ten individual bin-based ranks.

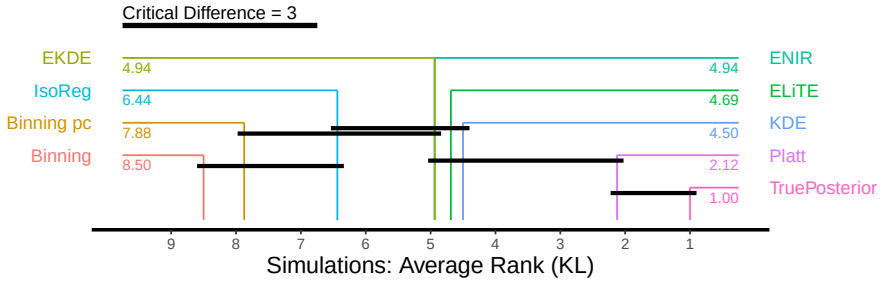
It is important to observe in figure 3.2 that both proper scoring rules succeed at identifying that Platt scaling and predicting the true posteriors are optimal, without significant differences, which indeed is true. The two binning variants are ranked last, while the group of five different techniques yields roughly similar results under all four metrics. It is especially interesting to observe that the two proper scoring rules yield reasonably similar rankings as well as that the results obtained from both, log-loss



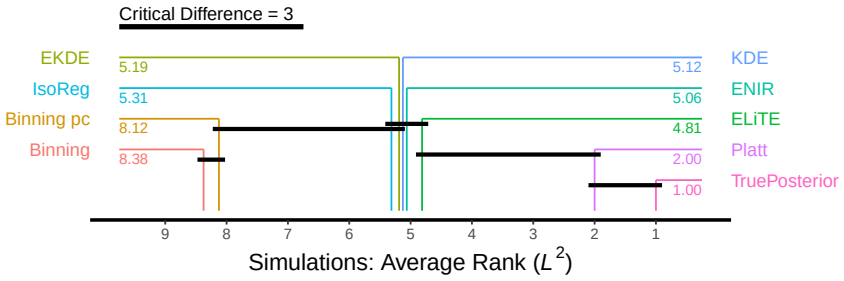
(a) Nemenyi test results for the log-loss.



(b) Nemenyi test results for the Brier score.

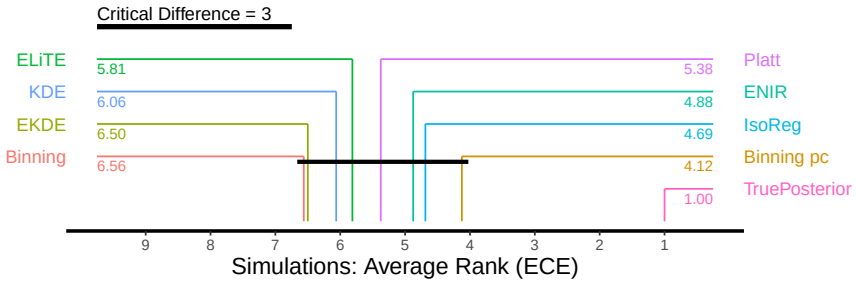


(c) Nemenyi test results for the Kullback-Leibler divergence.

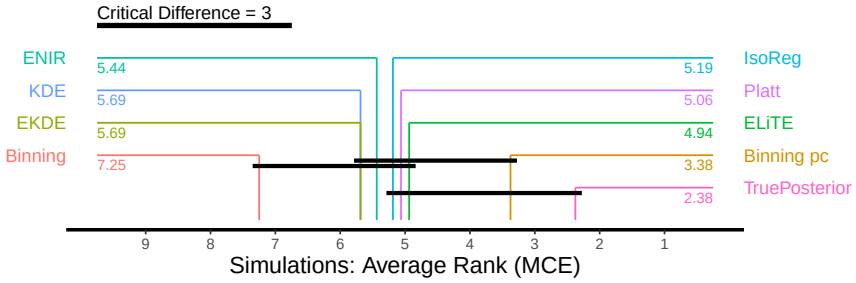


(d) Nemenyi test results for the L^2 error.

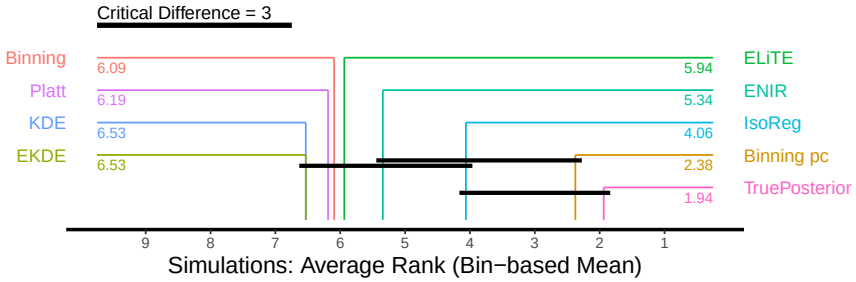
FIGURE 3.2: Nemenyi test results for the log-loss, the Brier score, the Kullback-Leibler divergence and the L^2 error in the simulation study.



(a) Nemenyi test results for the expected calibration error (ECE) with 10 bins.



(b) Nemenyi test results for the maximum calibration error (MCE) with 10 bins.



(c) Nemenyi test results for the averaged rank over all ten individual parameterizations.

FIGURE 3.3: Nemenyi test results for the bin-based error metrics.

and Brier score, are similar to the ones of their respective continuous counterpart. This is very interesting as the former two are computed using the known class values (which are also available on real-world data), while the latter were directly computed using the true posterior probabilities (which are unknown in general). It should be noted that the Brier score and the L^2 error ranks are slightly different to those of the log-loss and the Kullback-Leibler divergence. However, this is easily explainable since the latter have a singularity at 0.

On the contrary, figure 3.3 shows that the bin-based error metrics mostly succeeded at identifying[5] the true posteriors as the best calibration technique, while failing at identifying Platt scaling as equivalent. Furthermore, all techniques besides predicting the true posteriors and the two variants of binning are roughly comparable, while the binning variant without precalibration seems to be the worst technique. On

[5]The respective omitted figures are available in the supplementary material.

the other hand, the precalibrated binning variant gave second-best results with respect to ECE variants and those of MCE with 10 and 20 bins, while the MCE variants with at least 30 bins even detect it as superior to the true posteriors, however without significant differences. Still, most remarkably is the fact that the provably optimal technique under this conditions is never detected as such, the best average rank 5.06 is obtained with MCE using 10 bins.

Besides interpreting this as a strong empirical evidence against the validity of bin-based metrics and supporting their problematic aspects, as discussed in section 3.3, these results also show that at least under artificial conditions, a sigmoidal preprocessing can massively improve the calibration results with respect to all metrics. This empirically supports the application of sigmoidal transformations in classifier calibration and thus also those of monotonic ones.

Additionally, it is interesting to analyze the difference between Platt scaling and directly predicting the true posterior probabilities in more detail. As previously mentioned, they differ only in computing the sigmoid's parameters by using the given data or the data-generating distribution, respectively. Even though these differences should be negligible, they resulted in a large gap with respect to the bin-based metrics (which will be addressed in the following experiment) and small differences under the proper scoring rules. Still, the deviations between them are not symmetric under the proper scoring rules either, because in both cases the true posteriors yield smaller average ranks, even though these are not detected as significant in the (rank-based) Nemenyi test. Thus, the proper scoring rules seem to be sensitive enough to even detect these small differences, which also empirically supports their validity as calibration error functions.

The existence of these differences can be explained from multiple sources of errors that influence the parameter estimation and thus the predicted probabilities. First, since there is no closed-form solution, the parameters are estimated iteratively, and the estimation is terminated as soon as a certain target accuracy is reached. Thus, there will always be a remaining error. Second, if a distribution is sampled, there will always be errors resulting from the sampling. The sample simply contains less information than the data-generating distribution. This is additionally amplified from the random cross validation sampling that is used to estimate the parameters. Third, the sigmoid parameters are estimated using a maximum likelihood approach and thus can slightly overfit its training data, which is obviously strongly related to the sampling issues. Finally, there are numerical inaccuracies in the involved floating-point operations.

All these issues accumulate into small errors $\varepsilon_i = P(\mathsf{y} \mid \mathsf{f}(\mathsf{x}_i)) - \sigma_{a,b}(\mathsf{f}(\mathsf{x}_i))$ between the true posterior probability and the one computed with Platt scaling, even if all parametric assumptions hold. In particular, the signed mean error per data set is positive in six and negative in ten cases with mean $-5.5 \cdot 10^{-4}$ and standard deviation $4.8 \cdot 10^{-3}$, respectively, and thus reasonably small but tends to be symmetric. On the contrary, the scoring rules report a larger overall error from the symmetric individual ones in almost all cases. This observation can be explained from the fact that Brier score and log-loss can be interpreted as discrete versions of the squared error and the Kullback-Leibler divergence, respectively. A continuous loss function's expectation $\mathbb{E}[L]$ over the whole distribution by definition has to show a larger error for any individual difference $L(p, p \pm \varepsilon)$. Assuming that a sample data set including its posterior probabilities is given, by the law of large numbers the empirical error of L will show the same behavior if the sample is sufficiently large. A very interesting observation here is the fact that the proper scoring rules seem to also inherit this property of their continuous counterparts, at least in this particular setting. If this

generally holds, it will be a further justification for the viability of proper scoring rules as classifier calibration evaluation metrics. This is an interesting open question, but without the true posterior probabilities, unfortunately it can hardly be analyzed under real-world conditions.

Second Experiment: Extended Bin-based Metrics

Evaluating the bin-based error metrics in the previous experiment yielded unexpectedly bad results with respect to the provably optimal technique Platt scaling. In the same way, it is interesting to observe that the difference between an analytical and a data-based estimation of the parameters resulted in a relatively small but detectable difference with respect to the proper scoring rules, the Kullback-Leibler divergence and the L^2 error, while it caused a remarkably large performance gap with respect to the bin-based metrics.

This is most likely caused by the inaccurateness of the binning calibration underlying the computation of the bin-based error values, resulting from the issues discussed in section 3.3. To support this hypothesis, an additional experiment reevaluating the ECE and MCE values was performed. While the proper scoring rules are instance-based error metrics (i.e. the error is computed *per instance* and averaged over the whole data set), bin-based metrics are computed using a data set to fit the reference calibration model yielding a single error value on it. Based on the equivalence shown in equation (3.24), at least the ECE value can be interpreted for each instance, however only if the reference model and thus indirectly the reference data set is given.

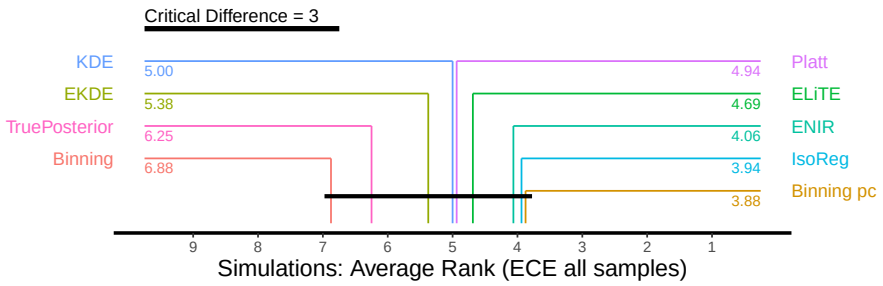
This is an important consequence as during the performed cross validation iterations, the reference data consist only of the respective cross validation fold's test data, while the remaining data were used to fit the calibration function and thus were excluded during evaluation. Thus, after finishing *all* cross validation iterations, there is a predicted probability available for each instance and, as a result, enables to recompute the bin-based error values using *all* predicted probabilities simultaneously as a reference instead of accumulating the fold-wise errors. It should be noted that with respect to the proper socring rules, both approaches are equivalent, as these only depend on the predicted probabilities (per instance) and the corresponding class values, but not indirectly on a reference data set. Thus, ECE and MCE (using 10 bins) were recomputed in a follow-up step using all predicted probabilities per data set. The respective results are illustrated in figure 3.4.

With respect to the surprisingly large difference between Platt scaling and the true posteriors, the first observation here is that the differences in the average ranks between these two now are more comparable to the respective ones under the proper scoring rules. However, Platt scaling is detected as the superior one out of these two approaches here. In particular, the true posteriors even yield a poor average rank in both cases despite being optimal. Still, most differences were not detected as significant using the Nemenyi test even though the Friedman test rejected the null hypothesis.

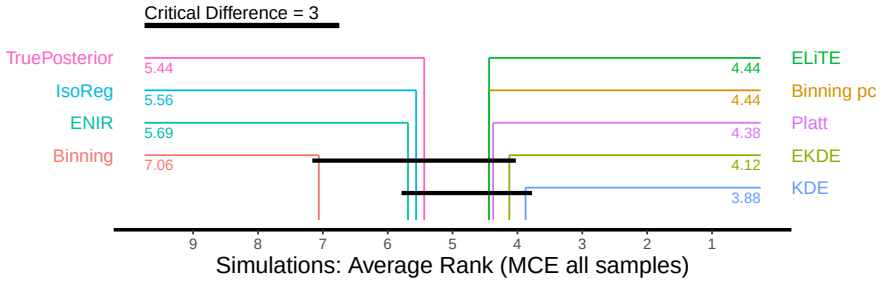
These results can be explained by the inaccuracies of the binning models underlying the ECE and MCE computations, which simply might be too coarse to yield reliable error metrics. An inferior model is simply unable to assess a superior or even provably optimal one.

Third Experiment: Random Influences

Besides comparing the bare accuracies of the different techniques, another important but actually not addressed issue in available studies are the predicted probabilities'



(a) Nemenyi test results for the expected calibration error (ECE) with 10 bins.



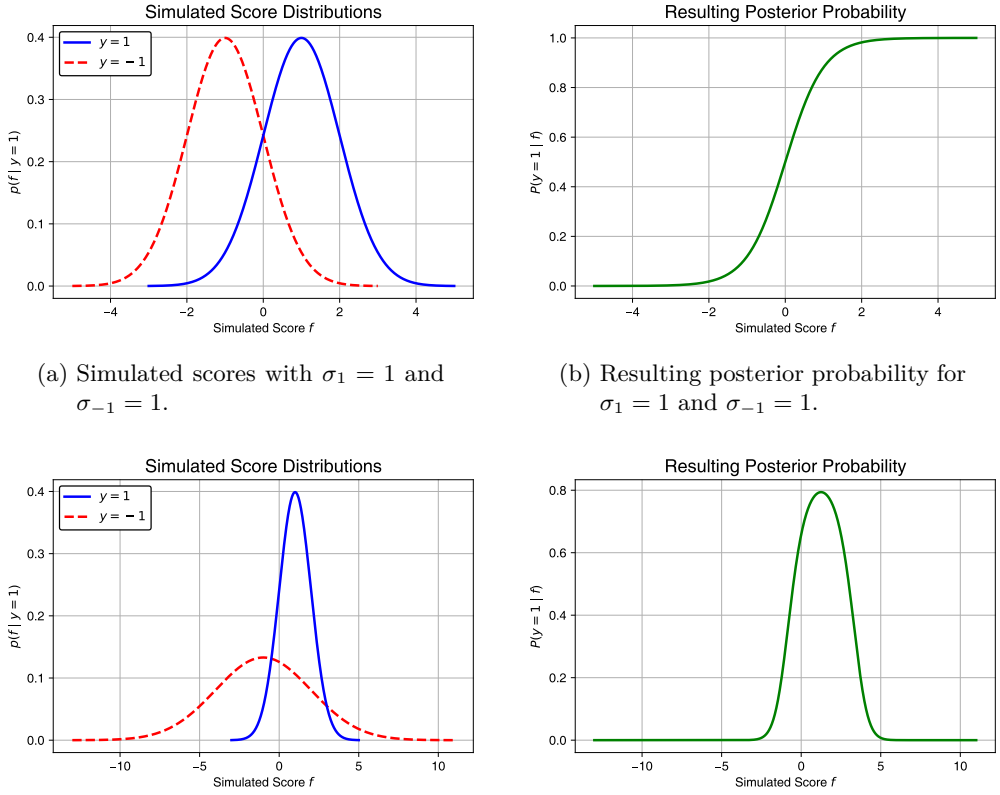
(b) Nemenyi test results for the maximum calibration error (MCE) with 10 bins.

FIGURE 3.4: Nemenyi test results for ECE and MCE using all samples.

variances. In particular, the minimum error is unknown and thus, the absolute error function values are hardly interpretable. Thus, it is rather interesting to observe that this question is not discussed in existing state-of-the-art works on classifier calibration. A possible explanation is that it is hardly answerable at all. An analytical solving requires a closed-form expression for the derivation of an optimization problem's solution (the training result) with respect to variations in the input data, which is usually impossible as even a closed-form solution for the optimization problem itself does not exist. However, a feasible alternative is to estimate the variance empirically by analyzing the dependence of the overall results on random variations.

In a comprehensive study, this involves to use a data set, at least one classification algorithm and a set of calibration techniques as well as to evaluate the predicted probabilities' variation caused by random changes in the input data. However, iterated classifier trainings will bias these results because usually even the training algorithm itself might be non-deterministic and thus, the analysis cannot distinguish between controlled random influences in the input data and the remaining other ones.

To address these problems, a simulation study was performed to fully control all random influences. Here it is important that, given a fixed sample of classifier predictions and their corresponding class labels, the estimation of the calibration functions as well as the error metrics are deterministic, i.e. do not depend on random influences. This enables to repeat the experiments under carefully selected random influences. In particular, two data sets were artificially generated, each consisting of 10000 instances per class by sampling Gaussian distributions with means $\mu = \mathsf{y} = \pm 1$. The first data set is generated with equal standard deviations $\sigma_1 = \sigma_{-1} = 1$, while in case of the second asymmetric data set, the standard deviations were selected as $\sigma_1 = 1$ and $\sigma_{-1} = 3$. Both distributions and their resulting posterior probabilities



(a) Simulated scores with $\sigma_1 = 1$ and $\sigma_{-1} = 1$.

(b) Resulting posterior probability for $\sigma_1 = 1$ and $\sigma_{-1} = 1$.

(c) Simulated scores with $\sigma_1 = 1$ and $\sigma_{-1} = 3$.

(d) Resulting posterior probability for $\sigma_1 = 1$ and $\sigma_{-1} = 3$.

FIGURE 3.5: Illustration of the simulated score distributions and the resulting posterior probabilities used in the the third calibration experiment.

are illustrated in figure 3.5.

In contrast to the previous experiments, the aim of this study is not to evaluate the calibration accuracies, but instead to focus on the variances of the predicted results. For this, small but controlled random influences are necessary. It might be possible to recreate the sample multiple times. Still, it remains at least unclear in how far the results remain comparable as the sample data are not the same between different iterations. For this reason, the score sample data are only generated once and to introduce a minor randomization, only the cross validation was iterated 100 times with independently generated partitions that were used to fit the calibration models. Consequently, each predicted score was calibrated 100 times.

The results are presented in table 3.1 where each row contains the estimated standard deviation *per instance*, averaged over all samples in the respective data set. In particular, the standard deviations are given for the predicted probability p as well as the four evaluation metrics log-loss, Brier score, Kullback-Leibler divergence and L^2 error. It should be emphasized that the error metric variances per instance cannot be computed using bin-based metrics because they are not instance-based and require a respective reference set. Still based on the insights of the second experiment, the variances of the bin-based metrics are expected to be an order of magnitude larger than those of the proper scoring rules.

The results show that in any case, the average standard deviation of the probability and the two proper scoring rules is lower bounded by approximately 10^{-3}. Furthermore, the averaged standard deviations of the Kullback-Leibler divergence

		Platt	Binning	Binning pc	IsoReg	ENIR	ELiTE	KDE	EKDE
$\sigma_{-1} = 1$	p	0.0009205 (1)	0.0049598 (6)	0.0030788 (4)	0.0050708 (8)	0.0050676 (7)	0.0045733 (5)	0.0025937 (3)	0.0025218 (2)
	φ^{LL}	0.0018451 (1)	0.0107470 (8)	0.0062052 (4)	0.0105946 (7)	0.0103446 (5)	0.0103804 (6)	0.0051511 (3)	0.0049900 (2)
	φ^{BS}	0.0010953 (1)	0.0056975 (6)	0.0037670 (4)	0.0063818 (8)	0.0063687 (7)	0.0053586 (5)	0.0031538 (3)	0.0030693 (2)
	KL	0.0000228 (1)	0.0024396 (7)	0.0005205 (4)	0.0010718 (6)	0.0007349 (5)	0.0030416 (8)	0.0002243 (3)	0.0002131 (2)
	L^2	0.0000062 (1)	0.0001810 (7)	0.0001147 (4)	0.0001297 (6)	0.0001294 (5)	0.0003094 (8)	0.0000395 (3)	0.0000376 (2)
$\sigma_{-1} = 3$	p	0.0010478 (1)	0.0071715 (7)	0.0037350 (3)	0.0027193 (2)	0.0076770 (8)	0.0043480 (4)	0.0049298 (6)	0.0048488 (5)
	φ^{LL}	0.0022614 (1)	0.0156614 (8)	0.0074887 (3)	0.0056306 (2)	0.0146863 (7)	0.0084944 (4)	0.0098044 (6)	0.0096305 (5)
	φ^{BS}	0.0016643 (1)	0.0107569 (8)	0.0057044 (3)	0.0043269 (2)	0.0105483 (7)	0.0061493 (4)	0.0076498 (6)	0.0075377 (5)
	KL	0.0012919 (6)	0.0019517 (8)	0.0006284 (4)	0.0008486 (5)	0.0019137 (7)	0.0005332 (3)	0.0004522 (2)	0.0004400 (1)
	L^2	0.0004036 (6)	0.0004520 (7)	0.0002154 (5)	0.0002131 (4)	0.0004711 (8)	0.0001251 (1)	0.0001646 (3)	0.0001619 (2)

TABLE 3.1: Estimated standard deviations per calibration technique for the predicted probability p, the log-loss φ^{LL}, the Brier score φ^{BS}, the Kullback-Leibler divergence KL and the L^2 error, each averaged over the respective data set.

and the L^2 error are smaller (in most cases even an order of magnitude), but this has less practical relevance as these cannot be computed in practice. Thus, it is unlikely that probabilities can be calibrated with errors smaller than those obtained under relative mild random influences.

Finally, the most important conclusion from the performed simulation studies is a strong empirical evidence against using bin-based error metrics while comparing classifier calibration techniques, as well as an empirically justification to use proper scoring rules for this purpose, supporting the theoretical insights of section 3.3.

3.4.2 Real-World Data

The summary of existing results in chapter 2 showed that there are different empirical comparisons of classifier calibration techniques in the respective literature available, however all of them expose certain drawbacks. Often the data sets are either large and the experiments focus on a relative restricted number of different data sets (sometimes even only a single one), or multiple data sets are used but many are relatively small. There simply is no study available that evaluates calibration on 20 different data sets where each has 10000 or more training instances. Since no direct evaluation metric exists, it is hardly reasonable at all to evaluate calibration on data sets consisting of about one hundred instances – the sample is highly likely to be not representative enough. On the other hand, evaluations should also cover high-dimensional data sets because here probability estimation becomes particularly challenging, as previously discussed. However, simultaneously analyzing large-scale high-dimensional data sets can easily become infeasible if multiple training iterations are necessary, as each one easily takes too much time.

Thus in contrast to these existing studies, the experiments were preformed using a set of 46 reference real-world data sets covering a large variety of domains. All are publicly available, mostly in either the UCI Machine Learning Repository [Dua & Graff 2019] or by LIBSVM [Chang & Lin 2011]. The most important properties are presented in table 3.2, while further details including the necessary steps to recompute the used data files from their respective sources are given in the supplementary material, too. Summarized aspects are illustrated in table 3.3 showing that the data sets used here are on order of magnitude larger than in any comprehensive comparative study, as summarized in subsection 2.2.3.

First Experiment: Calibration Accuracy

Besides comparing the pure accuracies of the respective techniques only, the performed study especially addresses the open question whether the data set should be

	Name	Samples (r)	Class 1	Class -1	Features (n)	Size [MB]
1	Adult	48842	7841	41001	41	4.57
2	Arcene	200	88	112	9961	5.48
3	Arrhythmia	452	207	245	257	0.35
4	Bank Marketing	41188	4640	36548	62	6.27
5	Code-RNA	488565	162855	325710	8	31.02
6	Connect-4 (With Draw)	67557	44473	23084	42	6.17
7	Connect-4 (Without Draw)	61108	44473	16635	42	5.58
8	Covertype	495141	211840	283301	49	59.92
9	Default of credit card clients	30000	6636	23364	23	2.75
10	Detect Malacious Executable (AntiVirus)	373	72	301	503	0.38
11	Dota2 Games Results	102944	54284	48660	114	24.55
12	First-order theorem proving	6118	2554	3564	51	2.69
13	Gas Sensor Array Drift	5935	2926	3009	128	7.43
14	Gesture Phase Segmentation	5691	2741	2950	32	2.01
15	Gisette	7000	3500	3500	4971	78.75
16	Give Me Some Credit	150000	10026	139974	8	5.76
17	Grammatical Facial Expressions	27936	9877	18059	300	57.78
18	Hill-Valley (No Noise)	1212	612	600	100	1.45
19	Hill-Valley (With Noise)	1212	606	606	100	0.84
20	HTRU2	17898	1639	16259	8	1.76
21	Human Activity Recognition Using Smartphones	10299	4672	5627	561	67.29
22	Ijcnn1	191681	18418	173263	22	26.04
23	Insurance Company Benchmark (COIL 2000)	9822	586	9236	85	1.72
24	KDD Cup 1998 Data	191779	9716	182063	313	158.90
25	Letter Recognition	1536	753	783	16	0.06
26	Localization Data for Person Activity	87190	32710	54480	32	9.90
27	Madelon	2600	1300	1300	500	5.21
28	MAGIC Gamma Telescope	19020	12332	6688	10	1.50
29	MicroMass	931	357	574	1139	3.34
30	MiniBooNE particle identification	130064	36499	93565	50	55.33
31	Multiple Features	400	200	200	649	1.36
32	Musk 1	476	207	269	167	0.32
33	Musk 2	6598	1017	5581	167	4.46
34	Nomao	34465	24621	9844	89	6.91
35	Occupancy Detection	20560	4750	15810	5	0.96
36	Online News Popularity	39644	19562	20082	59	16.95
37	p53 Mutants	31159	151	31008	5408	1115.00
38	Polish companies bankruptcy	10503	495	10008	20	1.70
39	PUC-Rio	165633	43390	122243	21	12.72
40	Quality Assessment of Digital Colposcopies	287	216	71	62	0.24
41	Skin Segmentation	245057	50859	194198	3	3.60
42	Spambase	4601	1813	2788	57	0.71
43	Statlog (Shuttle)	58000	45586	12414	9	1.61
44	Tamilnadu Electricity Board Hourly Readings	5811	2906	2905	2	0.23
45	UJIIndoorLoc	21048	9760	11288	520	43.86
46	Weight Lifting Exercises	39242	11159	28083	51	9.37

TABLE 3.2: Individual data sets and their most important properties.

Number of Instances	Number of Features
39 / 46 with at least 1000	39 / 46 with at least 10
34 / 46 with at least 5000	27 / 46 with at least 50
27 / 46 with at least 10000	18 / 46 with at least 100
13 / 46 with at least 50000	9 / 46 with at least 500
9 / 46 with at least 100000	

TABLE 3.3: Summary of the data set properties.

split into separate parts for training the classifier and the calibration function, respectively. Each data set was partitioned using a 10-fold stratified cross validation into train and test data, where each feature column was standardized to mean 0 and standard deviation 1. The test data were only used in the final evaluation and excluded from anything else.

Here in case of fitted calibration, the train data were used for both, classifier training and calibration estimation. To obtain comparable results with predicted calibration, an additional inner 10-fold stratified cross validation was applied to use $\frac{9}{10}$ of the training data to train the classifier and to predict the remaining fraction. In total this results in $10 \cdot (10 + 1) = 110$ classifier trainings per data set to apply both approaches with maximum comparability.

For aforementioned reasons, the data sets are relatively large. Thus, combining large data sets with 110 training/test iterations on each requires highly efficient classification algorithms, otherwise a result cannot be computed in acceptable runtime. For this reason, support vector machines based on LIBLINEAR [Fan et al. 2008] were applied. Even though this means that only a single classification algorithm is used, the analysis explicitly focuses on the influence of the learning algorithm's parameterization. In particular, linear support vector machines solve the following optimization problem at training time

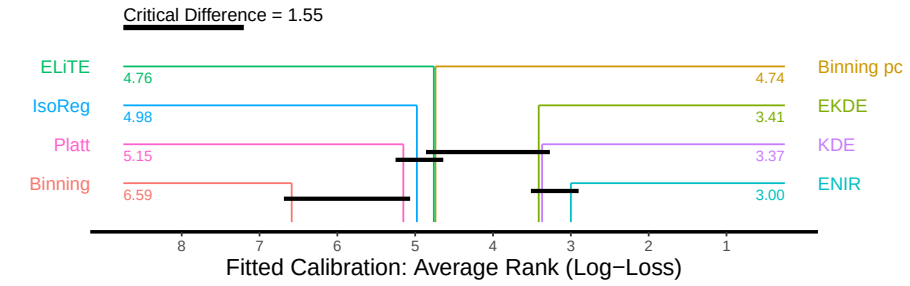
$$\arg\min_{w,b} \left\{ \frac{1}{2}\|w\|_2^2 \;+\; C \cdot \sum_{i=1}^{r} \xi_i \;\middle|\; \mathsf{y}_i \cdot (w^\top \mathsf{x}_i + b) \geq 1 - \xi_i,\ \forall i = 1, \ldots, r \right\} \tag{3.26}$$

where the only free parameter C, which controls the amount of regularization between margin width and margin violation, has to be selected accordingly. Thereafter, newly observed instances can be predicted using $\mathsf{f}(\mathsf{x}) = w^\top \mathsf{x} + b$ and assigned to one of the two classes using the sign of f.

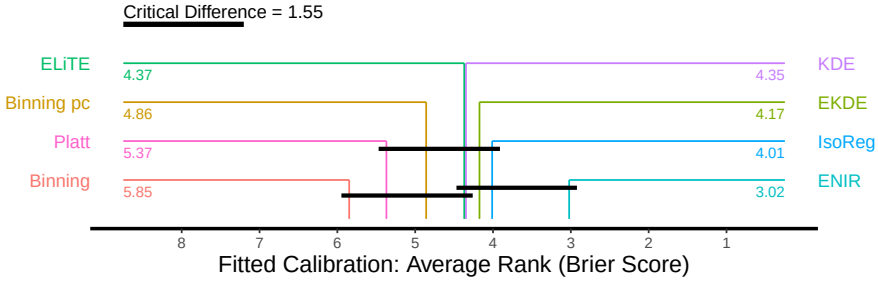
It should be added that this formulation does not cover the case of kernelized (or non-linear) support vector machines. Even though these are interesting alternatives, currently their application requires to solve the dual version of (3.26) using the SMO algorithm [Fan et al. 2005], which has a runtime of $\Omega(r^2)$ and thus, is prohibitively slow on large data sets such that a single training iteration takes multiple days [Alvarsson et al. 2016]. Further details about kernelized variants and the corresponding solvers are discussed in the respective literature [Chang & Lin 2011; Hsu et al. 2016], and performing a similar large-scale study using them remains as an interesting open issue.

Consequently, the only remaining parameter is the one that controls the amount of regularization. A default choice of implementations [Chang & Lin 2011; Fan et al. 2008] is $C = 1$, while most studies on calibration do not discuss the respective classifiers' parameterizations. Explicit optimization of the regularization is usually performed using iterated cross validations over an exponential grid of predefined values [Hsu et al. 2016]. Since this ideally can be combined with the necessary inner cross validation iterations for predicted calibration [Guo et al. 2017; Niculescu-Mizil & Caruana 2005b], an explicit hyperparameter optimization over the exponential grid $\{2^{-10}, 2^{-8}, \ldots, 2^{8}, 2^{10}\}$ was performed. The respective optimal value was selected as the one with maximum cross-validated (referring to the inner cross validation) accuracy on the training data, where the actual value was scaled reciprocally to the class priors to balance both classes during training. Consequently, the hyperparameter optimization increases the number of overall training iterations by 11, which in total results in $10 \cdot (11 \cdot 10 + 1) = 1110$ optimizations per data set.

Following the discussion of section 3.3 and empirically supported by the insights

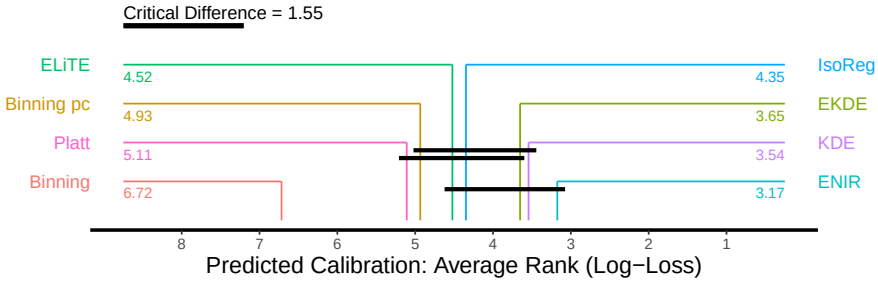


(a) Nemenyi test results for the log-loss.

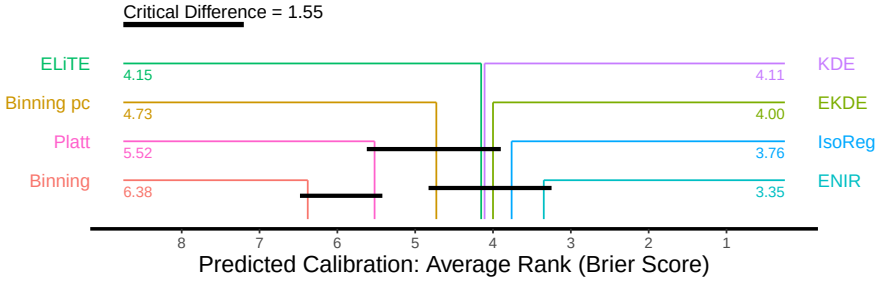


(b) Nemenyi test results for the Brier score.

FIGURE 3.6: Nemenyi test results for the log-loss and the Brier score in the fitted calibration setting.



(a) Nemenyi test results for the log-loss.



(b) Nemenyi test results for the Brier score.

FIGURE 3.7: Nemenyi test results for the log-loss and the Brier score in the predicted calibration setting.

obtained under perfect information, the two proper scoring rules log-loss and Brier score were used to evaluate the calibrated probabilities. Similar to the previous simulation study, the techniques are ranked by their respective scores on each data set such that overall average ranks can be computed. Analogously, the Friedman and Nemenyi tests were applied using the 0.05 significance level, figures 3.6 and 3.7 illustrate the results in the fitted and predicted calibration setting, respectively.

The best results are obtained in any case using ENIR, however the difference towards KDE and EKDE are never be detected as significant. Besides the log-loss in case of the fitted calibration setting, also the results obtained with ELiTE and isotonic regression are not detected to be significantly different. Besides this, there are two other remarkable observations. First, the results obtained with binning are much improved if a sigmoidal preprocessing is applied (similar to the results observed in the previous simulation study). Second, isotonic regression's results are much better with respect to the Brier score than based on the log-loss. This can be explained by the fact that isotonic regression optimizes the Brier score subject to a monotonicity constraint. Thus on larger data sets, it is explainable that the computed probabilities yield relatively small Brier scores.

Second Experiment: Inner Cross Validation?

In the previous results, the techniques' predictive performances between the fitted and predicted calibration setting were roughly comparable. In case of the Brier score, the overall rankings of the techniques are even the same in both settings, while in case of the log-loss, the respective relative performances slightly change. Nevertheless, it is important to emphasize that this only means that their rankings not seem to be much influenced by using the fitted or predicted sores as training data, but it does not say anything about their *absolute* calibration accuracies. The demand for an additional validation step to compute unbiased predictions means that the absolute errors are expected to be smaller. Hence, a subsequent experiment was performed in which the absolute error values in both settings were compared.

In particular, for each calibration technique and each evaluation metric (log-loss and Brier score), each data set's error in the predicted calibration setting is compared to the fitted setting's counterpart. The test hypothesis formulates that the predicted calibration's error is smaller than the one of the fitted calibration, i.e. that applying the additional cross validation step reduces the calibration error. The number of respective data sets where this holds is given in table 3.4. Additionally, this hypothesis was combined with a one-sided sign test.

Here, it is interesting to observe that the maximum number of data sets where the additional cross validation reduces the calibration error is 22, which is less than the half of the overall number of data sets. Thus, in every comparison the error is increased in more cases than it is decreased. Consequently, also the sign test's p-values are far away from rejecting the null hypothesis. In fact, even formulating the inverse hypothesis that fitted calibration yields superior results would be significant at $\alpha = 0.05$ in six cases (the binomial distribution's critical value is 16 in this particular example). It should be emphasized that properly applying the latter would also require to perform a Bonferroni correction. Still, the aim is not to derive too many conclusions based on significance tests because the usefulness of statistical tests in data mining can generally be criticized [Demšar 2008] in the same way as incorrect conclusions about them are omnipresent [Goodman 2008].

	Platt	Binning	Binning pc	IsoReg	ENIR	ELiTE	KDE	EKDE
Log-loss	14 (0.9977)	20 (0.8490)	17 (0.9730)	22 (0.6706)	17 (0.9730)	21 (0.7693)	19 (0.9080)	18 (0.9481)
Brier Score	14 (0.9977)	18 (0.9481)	16 (0.9871)	15 (0.9943)	13 (0.9992)	15 (0.9943)	18 (0.9481)	18 (0.9481)

TABLE 3.4: Number of data sets where predicted calibration reduces the calibration error. Additionally, the p-value of a one-sided sign test is given.

In conclusion of this experiment, there is no general indication available that applying the additional cross validation step should be preferred. This is an interesting result as it means a substantial speedup in practical applications if the data set is large such that the respective additional cross-validated training iterations would significantly increase the training time requirements.

However, it is important to emphasize that this result is only based on linear support vector machines, which are usually sparse classifiers, i.e. the output function does only depend on a fraction of the training samples. Consequently, there might be sufficient data to unbiasedly estimate the score distribution even without cross validation or a hold-out set in the predicted calibration setting, respectively. Thus, it is an interesting open question to repeat a similar experiment with kernelized support vector machines or even different classification algorithms and to analyze if the same observations can be confirmed.

3.5 Summary

In summary there were many different results presented. With respect to monotonic calibration, some incorrect statements that appeared in the literature were corrected. Besides showing its equivalence to beta calibration up to a sigmoidal preprocessing, most importantly proving Platt scaling's optimality for different families of score distributions gives a theoretical justification for its application and at least partially provides a so far lacking explanation for its good results. Besides this, theorem 3.2 even enables to extend these properties for different families of likelihood distributions as long as they fit the respective decomposition. Therefore despite of its simplicity, Platt scaling is still a powerful calibration technique and simultaneously very versatile due to its high efficiency.

More generally with respect to monotonic calibration, the combination of isotonic regression's convergence properties and its relation to observations of Bernoulli-distributed random variables turn it into a straightforward selection in any case where a monotonic transformation is assumed but Platt scaling's parametric assumptions do not hold. Artificial possibilities are distributions with non-differentiable density functions, whose practical relevance is at least unclear. Thus, especially interesting variants are likelihood distributions with differentiable density functions $p(\mathsf{f} \mid \mathsf{y})$ such that the transformation into posterior probabilities is monotonic and, as a combination of differentiable functions, differentiable but not a valid instance of Platt scaling. Here, it is an explicit interesting open question in which cases these distributions are actually observed for sufficiently selected data and classification algorithms, respectively.

Other important aspects of possible classifier calibration error metrics were analyzed. Here, despite attracting much interest in recent years, bin-based evaluation metrics are useful to evaluate the calibration of a classifier that has not been explicitly optimized to produce calibrated probabilities, but they are inappropriate to evaluate and compare calibration techniques. This is caused by the fact they rely on computing

a reference calibration binning model and evaluate the previous calibration by comparing the different models' probabilities. These insights were empirically supported using a simulation study that showed a well correlation between the proper scoring rules and their continuous counterparts, while the bin-based metrics yield biased and wrong results. In light of this, conclusions based on bin-based evaluation metrics as those that are available in reference results can be misleading.

With respect to the evaluated real-world data sets, the first observations are that the presented extension to KDE calibration as well as EKDE are two powerful and easy to apply non-monotonic calibration techniques. They successfully compete with state-of-the-art techniques while being easier to apply as practically any statistical software toolkit allows the application of kernel density estimation, whereas applying comparable state-of-the-art techniques like ENIR or ELiTE requires to solve the respective optimization problems.

Additionally, there is no indication of any relevant accuracy difference between KDE and EKDE. This can be explained by the fact that KDE (especially with the extensions presented) already most probably well approximates the true likelihood distributions and, consequently, an extension to ensembles might only have a minor impact. Thus, improvements using density-based calibration are less likely to reduce the error by using ensembles that share the same constant priors, but instead maybe by constructing ensembles sharing the same likelihoods learned on the data in combination with varying prior distributions. Here, the class priors $P(\mathsf{y} = \pm 1)$ could be modeled alternatively by using Gaussian distributions whose means are selected as the fractions estimated on the data. Sampling this distribution to construct an ensemble with slightly different class priors presumably better improves the calibration than varying the density estimations. This also enables to include the respective class priors in the scoring of the different models in (3.20).

Still, the question remains whether well-performing density estimation-based models are already generally sufficient for any binary classifier calibration task. Their theoretical properties often guarantee that the densities are reasonably approximated, while verifying the posterior estimation using error metrics is biased by the lack of true posterior probabilities. Thus, it is impossible to differentiate between *real* improvements and those caused by the lack of true reference probabilities, i.e. effects caused from an insufficient sampling of the test data.

With respect to practical applications, it is especially interesting that a direct necessity for a separate hold-out calibration data set was not observed, at least in this particular setting. Hence, applying calibration in practice without iterated trainings is very efficient. It is an interesting open issue to repeat a similar experiment with kernelized support vector machines or other machine learning models. However, even a quadratic training runtime complexity will most likely be too inefficient here. Additionally, the sigmoidal precalibration massively improved the calibration results of binning, which also empirically supports its application before applying density estimation techniques, if not even its general application in the context of classifier calibration. This is remarkable because also state-of-the-art techniques restrict their application to probabilistic classifiers using sigmoidal preprocessings.

In the following chapters, classifier calibration will be applied in combination with decomposition-based classification. In particular, classifiers will be used and calibrated for different subproblems that are created according to the respective decomposition. For this, at first a theoretical framework will be applied that offers several advantages over existing, Bayesian-motivated or heuristic approaches, in particular with focus on dynamic classification.

Chapter 4

Evidence Theory

An important part of decomposition-based classification is the estimation of comparable individual predictions, thus calibration techniques are particularly relevant for it. In combination with the one-vs-one decomposition, calibration is an important basis for all most relevant pairwise coupling techniques, as presented in full detail in chapter 2.

According to recent results, one of the main issues in decomposition-based classification and in particular the one-vs-one decomposition is the non-competence problem. Not only to deal with the related open issues but especially to integrate dynamic information given by a varying class set $\mathcal{M} \subseteq \mathcal{Y}$, this chapter introduces a new approach to decomposition-based classification using *evidence theory*, which is a powerful framework that generalizes Bayesian probability theory.

The motivation for this approach is three-fold: First, evidence theory allows the structured modeling of *partial* information. For example, independently estimated probabilistic one-vs-all predictions $(\mathsf{f}_1(\mathsf{x}), \mathsf{f}_2(\mathsf{x}), \ldots, \mathsf{f}_k(\mathsf{x}))$ have no real probabilistic interpretation as they do not sum to one. Still, evidence theory allows a reasonable interpretation. Next, the framework's theoretical constraints yield a new understanding of the non-competence problem. Most importantly, it allows a structured way to combine partial into joint information such that decomposition-based classification not only can be modeled, but also dynamic class information can be integrated in the same way to obtain a consistent, theoretically justified evidence-theoretic approach to dynamic classification.

First, section 4.1 shortly presents evidence theory and its concepts that are relevant for the remaining work. Thereafter, section 4.2 applies the one-vs-all and one-vs-one reductions using the presented evidence-theoretic modeling in subsections 4.2.1 as well as 4.2.2, respectively, and demonstrates in subsection 4.2.3 how evidence theory can be used to construct new decomposition-based approaches. Finally, section 4.3 presents how dynamic classification can be solved using evidence theory.

4.1 Introduction to Evidence Theory

Evidence theory [Shafer 1976], also known as *Dempster-Shafer theory of evidence* or *theory of belief functions*, is a generalization of finite Bayesian probability theory. Therefore, $\Omega = \{1, 2, \ldots, k\}$ refers to a finite set of possible outcome events. In any actual application, Ω will always equal the set of classes on which different predictions are to be combined.

4.1.1 Mass Functions

In a classical stochastic context, probabilities of events $\omega \in \Omega$ are modeled using a probability function

$$f : \Omega \to [0,1] \quad \text{satisfying} \quad \sum_{\omega \in \Omega} f(\omega) = 1 \tag{4.1}$$

such that probabilities of arbitrary events $A \subseteq \Omega$ can be computed by the induced probability measure P on the power set $\mathcal{P}(\Omega)$ as $P(A) = \sum_{\omega \in A} f(\omega)$. This has several intuitive consequences, for example $P(\emptyset) = 0$ and probabilities of disjoint unions $C = A \cup B$ can be computed by summing the respective sets' probabilities $P(C) = P(A \cup B) = P(A) + P(B)$ and in particular $P(A) + P(A^{\mathsf{C}}) = 1$.

Evidence theory models uncertainty differently using a *mass function* or *basic probability assignment*

$$m : \mathcal{P}(\Omega) \to [0,1] \quad \text{satisfying} \quad m(\emptyset) = 0 \quad \text{and} \quad \sum_{A \subseteq \Omega} m(A) = 1 \tag{4.2}$$

that assigns a *basic probability number* to each $A \subseteq \Omega$. All subsets such that $m(A) > 0$ holds are called *focal sets* or *focal elements*. Extending a probability function with zero to all subsets containing more than a single element always induces a *Bayesian* mass function, therefore the latter can be interpreted as a strict generalization of the former.

In the context of evidence theory, Ω is often referred to as the *frame of discernment* where the key concept is to model uncertainty using a two-dimensional measure consisting of *belief* $Bel(A)$ and *plausibility* $Pl(A)$ of $A \subseteq \Omega$ that both are constructed using m. In particular, the belief of A

$$Bel(A) \coloneqq \sum_{B : B \subseteq A} m(B) \tag{4.3}$$

is the sum of all masses assigned to A or any of its subsets, while the plausibility is one minus the belief of A^{C}

$$Pl(A) \coloneqq 1 - Bel(A^{\mathsf{C}}) = 1 - \sum_{B : B \cap A = \emptyset} m(B) = \sum_{B : B \cap A \neq \emptyset} m(B) \tag{4.4}$$

where the two equivalences are simple reformulations. Using (4.3) and (4.4), it is straightforward to see that $Bel(A) \leq Pl(A)$ always holds. Therefore, the belief $Bel(A)$ can be interpreted as a degree of strict support for A, while the plausibility $Pl(A)$ as a measure of non-conflict, both according to the evidence of m.

In the Bayesian case where the mass function m is equivalent to a probability function f, the belief coincides with the plausibility and both are equal to the induced probability measure P. Even though arbitrary mass functions do not represent probability distributions, it is still possible to create the *pignistic probability distribution* [Smets & Kennes 1994]

$$BetP(\omega) \coloneqq \sum_{A \subseteq \Omega \,:\, \omega \in A} \frac{m(A)}{|A|}, \quad \forall \omega \in \Omega \tag{4.5}$$

from an arbitrary mass function m. Here, each set's mass is uniformly distributed among its elements and each element's masses are accumulated over all sets containing it.

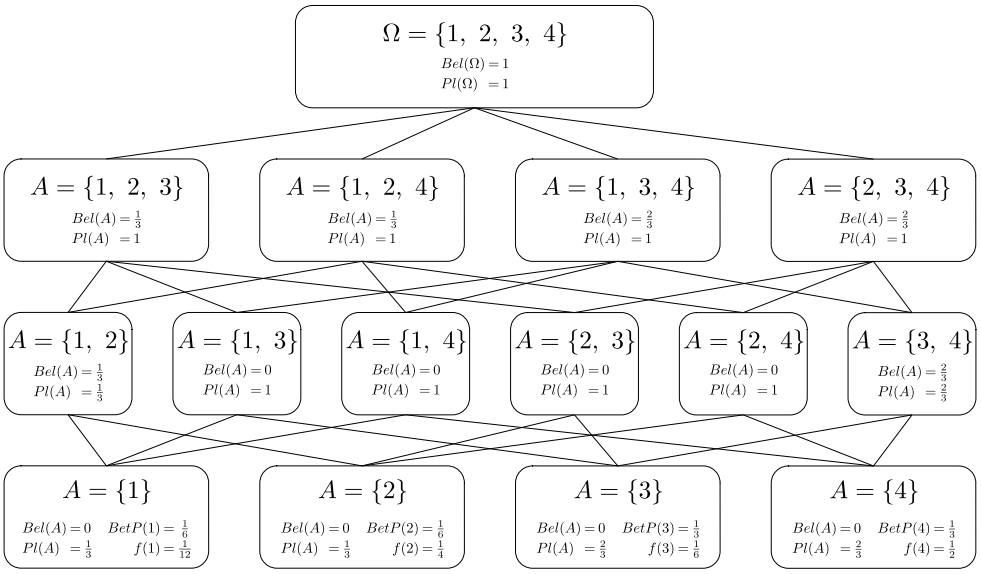


FIGURE 4.1: Illustration of the belief $Bel(A)$, the plausibility $Pl(A)$ and the pignistic probabilities $BetP(\omega)$ based on (4.7).

It is straightforward to verify that (4.5) indeed defines a valid probability distribution over Ω. Moreover, the pignistic probability distribution coincides with the respective probability function $BetP(\omega) = P(\{\omega\}) = f(\omega)$ for all $\omega \in \Omega$ in case of a Bayesian mass function m and probability function f, respectively.

The previous definitions can be demonstrated by using the following non-uniform probabilities

$$f(\omega) = \begin{cases} \frac{1}{12} & \text{if } \omega = 1 \\ \frac{1}{4} & \text{if } \omega = 2 \\ \frac{1}{6} & \text{if } \omega = 3 \\ \frac{1}{2} & \text{if } \omega = 4 \end{cases} \tag{4.6}$$

to model the rolling of a tetrahedron (i.e. a four-face die). This leads to the probabilities $P(\{1,2\}) = \frac{1}{3}$ and $P(\{3,4\}) = \frac{2}{3}$. Assuming that the probability function f itself is unknown, these two properties alone can be expressed by a mass function

$$m_1(A) = \begin{cases} \frac{1}{3} & \text{if } A = \{1,2\} \\ \frac{2}{3} & \text{if } A = \{3,4\} \\ 0 & \text{otherwise} \end{cases} \tag{4.7}$$

that models the partial evidence. Even though there are only two focal elements, still belief and plausibility can be computed for any subset of Ω. The example is illustrated in figure 4.1 that shows all non-empty subsets (for the empty set always holds $Bel(\emptyset) = Pl(\emptyset) = 0$) as well as their corresponding beliefs and plausibilities in a tree structure. Additionally, the leaves contain the pignistic and true probabilities. It should be noted that these two distributions do not coincide because the mass function simply does not capture enough information. In this regard, the pignistic probabilities are an approximation of an unknown distribution that is computed only from partial or incomplete information about it.

4.1.2 Dempster's Rule of Combination

Intuitive properties of probabilities generally do neither hold for beliefs nor plausibilites. In the last example, neither belief nor plausibility of a set and its complement in general sum to one. Even though these properties can be counterintuitive, the evidence-theoretic modeling allows the combination of multiple mass functions. For this, at first the *conflict*

$$\kappa \;=\; \kappa(m_1, m_2) \;\coloneqq \sum_{\substack{B,C\subseteq\Omega \\ B\cap C=\emptyset}} m_1(B)\cdot m_2(C) \tag{4.8}$$

between two mass functions m_1 and m_2 has to be defined. Here, $0 \leq \kappa \leq 1$ always holds and measures the amount of contradicting information between m_1 and m_2, as shown below. In case of full conflict $\kappa = 1$, there is no hypothesis on which they agree and evidence theory does not allow to combine them. Otherwise, they can be combined into the mass function

$$(m_1 \oplus m_2)(A) \;\coloneqq\; \frac{1}{1-\kappa} \cdot \sum_{\substack{B,C\subseteq\Omega \\ B\cap C=A}} m_1(B)\cdot m_2(C) \tag{4.9}$$

which is known as *Dempster's rule of combination*. The sum in (4.9) accumulates all possibilities such that m_1 and m_2 agree on A, which in general does not yield a valid mass function. Therefore, the result is normalized using the non-conflict $1-\kappa$. Here, the *closed-world assumption* assumes that each possible hypothesis is in accordance with both individual mass functions. It should be noted that (4.8) can equivalently be expressed as

$$\kappa(m_1, m_2) \;=\; 1 \;-\; \sum_{\substack{B,C\subseteq\Omega \\ B\cap C\neq\emptyset}} m_1(B)\cdot m_2(C) \tag{4.10}$$

which shows that the combination using (4.9) is possible if and only if there are at least two sets $B, C \subseteq \Omega$ with $B \cap C \neq 0$ and $m_1(B) > 0$ as well as $m_2(C) > 0$.

In the previous four-face die example, m_1 could be combined with

$$m_2(A) = \begin{cases} \frac{1}{4} & \text{if } A = \{1,3\} \\ \frac{3}{4} & \text{if } A = \{2,4\} \\ 0 & \text{otherwise} \end{cases} \tag{4.11}$$

which is also consistent with the respective probabilities in the same way as m_1 is. Analogously to figure 4.1, the beliefs, plausibilities and pignistic probabilities obtained with m_2 can be computed. Still, the latter will also not equal the true ones. But computing κ using (4.8) shows zero conflict and combining both using (4.9) recovers the true probability function.

It should be emphasized that in this particular example, the pignistic probabilities of both mass functions can alternatively be interpreted as two independent marginal distributions such that the joint distribution factorizes into these two. Therefore, their combination simplifies to a simple multiplication, which coincides with the true distribution. This is recovered in the evidence-theoretic context by zero conflict and exact information after combining them.

As (4.8) and (4.9) define a valid combination of two mass functions, iteratively applying them enables to combine an arbitrary, finite number of mass functions. Here, it is important that the operation is commutative and associative, thus the

general combination $\bigoplus_{i=1}^{t} m_i$ is well defined as long as no combination has full conflict. Furthermore, it should be emphasized that the combination is not idempotent, i.e. $m \oplus m \neq m$ in general. Instead, $m \oplus m$ should be interpreted as the overall evidence obtained by again observing the already existing one.

In general, an arbitrary mass function – in particular those obtained by combinations – can be used to compute the pignistic probabilities, which can be used for decision making. However, involved computations are exponential in the number of elements in Ω. Thus, any application of evidence theory either has to be restricted to small possible outcome event sets or requires further simplifications that circumvent the exponential complexity.

Besides these practical relevant issues, further different theoretical aspects can be discussed as well. Still, these are beyond the scope of this work and available in the respective literature [Dezert & Tchamova 2014; Shafer 1976, 2016; Tchamova & Dezert 2012; Voorbraak 1991; Wang 1994; Wilson 1993]. Here, particular interesting aspects cover algorithmic applications [Reineking 2014] and counterintuitive combination results [Zadeh 1979, 1984, 1986], especially in situations with high conflict.

4.2 Application to Decomposition-based Classification

Evidence theory is not only used in fusion strategies [Xu et al. 2014; Zhong et al. 2008] but offers two particular interesting properties with respect to decomposition-based classification: First, a mass function allows a better modeling of partial knowledge only than a probability function and second, it differentiates between two different degrees of probabilities in form of belief and plausibility. It is well known that a simple probability is incapable to differentiate between competent but insecure predictions and incompetent decisions. This was also reported in form of *conflict* and *ignorance* [Hüllermeier & Brinker 2008] as well as *ambiguity* and *imprecision* [Lachaize et al. 2016; Yang et al. 2017]. Both, conflict and ambiguity, refer to insecure decisions caused by overlapping class distributions, i.e. multiple labels are possible with respect to the data, but the decision remains competent. On the other hand, ignorance and imprecision refer to situations with low or no data where the predictions are also insecure, but due to missing competence from the data. Therefore, evidence theory is well suitable to model decomposition-based classification with explicit focus on the non-competence problem.

Besides works that discuss decomposition-based classification with *imprecise probabilities* [Destercke & Quost 2011; Quost & Destercke 2018; Yang et al. 2017] that aim at providing lower and upper bounds for the probabilities similar to interval-based approaches [Elkano et al. 2017], the most related existing work applies the one-vs-all and one-vs-one decomposition with an evidence-theoretic modeling [Quost et al. 2007]. However, the latter work exposes various drawbacks. Even though the authors additionally present a modification for probabilistic classifiers, the original approach requires to model the individual predictions with mass functions on the respective subsets of Ω. The formal background on which they are combined requires to define *subnormal mass functions* for which $m(\emptyset) > 0$ is allowed to hold. Consequently, the closed world assumption is relaxed such that the true state does not have to agree with all observed evidences given by mass functions. This is required because the true, but unknown class can be outside of the respective subset. Therefore, also the combination rule has to generalized by omitting the normalization factor in (4.9) such that masses of conflicting events are added to the empty set. Even though the authors note that the respective plausibilities can be computed using one-vs-all or

correcting classifiers as discussed in chapter 2, they alternatively motivate to compute them using one-class classifiers that are trained by only using data from one class. Finally, an overall mass function over $\{1, \dots, k\}$ cannot be computed directly and requires to solve a constrained optimization problem that scales exponentially in k during each prediction. Therefore, it is only applicable for small number of classes, but especially computing plausibilities from one-class classifiers is most likely to be unreliable. As discussed in chapters 2 and 3, even estimating accurate probabilities in supervised settings is extremely challenging. Thus, interpreting outputs from novelty detection and thus necessarily unsupervised learning algorithms as plausibilities is highly unjustified. Using *evidential calibration* [Xu et al. 2016], the approach still had already been applied in combination with support vector machines as pairwise classifiers besides the one-vs-all decomposition and a hybrid strategy [Lachaize et al. 2016].

In light of this, evidence theory offers an interesting option to model decomposition-based classification, but avoiding the exponential complexity is a challenging task. Therefore, the following part differently applies evidence theory such that the overall complexity is not increased but instead is completely given by the underlying reduction. Additionally, the presented techniques will be designed to only require probabilistic classifiers to preserve a general applicability.

4.2.1 One-vs-All Decomposition

The one-vs-all decomposition consists of only k classifiers whose predictions are slightly better interpretable than the ones of the one-vs-one decomposition. Additionally, there are no incompetent classifiers. Hence, it is preferable to start applying evidence-theoretic modelings with it. In the following, each one-vs-all classifier f_i, $i = 1, \dots, k$, is assumed to compute probabilistic predictions $(p_i(\mathsf{x}), 1 - p_i(\mathsf{x}))$. As in chapter 2, the explicit dependency on x can be omitted.

Whenever the binary one-vs-all classifiers are computed *independently*, the probability vector $(p_1, \dots, p_k)$ does not sum to one, therefore it does not define a posterior probability estimate. From the Bayesian point of view, the probabilities can only be interpreted as independently computed estimates of the i-th marginal distribution of the unknown posterior distribution $P(\mathsf{y} \mid \mathsf{x})$. Still, an estimate of the latter requires further processing, e.g. a normalization or a softmax transformation. In contrast to this, evidence theory allows the alternative interpretation of the independent estimates as respective mass functions that can be combined.

Therefore, the given one-vs-all predictions are used to define the following k mass functions

$$m_i(A) \coloneqq \begin{cases} p_i & \text{if } A = \{i\} \\ 1 - p_i & \text{if } A = \{i\}^\mathsf{C} \\ 0 & \text{otherwise} \end{cases} \tag{4.12}$$

for each class $1 \leq i \leq k$. Using this modeling, the combination $\bigoplus_{i=1}^{k} m_i$ is of interest. In general, this could be computed by $k - 1$ applications of (4.9) as long as there is no full conflict. Still, for practical applications a naive, iterative application is extremely inefficient as it is exponential in k. Therefore, a closed-form expression for the combination will be derived that can be computed efficiently. As will be shown in advance, this recovers a decision rule equivalent to the maximum rule $\arg\max_i p_i$, but extends this with a posterior probability estimate that differs from a simple normalization.

Here, the following result formulates one of the most important properties that enables to derive a closed-form expression for the combination $\bigoplus_{i=1}^{\ell} m_i$ for an arbitrary index $1 \leq \ell \leq k$. It is formulated more general than needed for the one-vs-all decomposition only as it will be applied for the modelings presented in subsections 4.2.2 and 4.2.3 as well.

Lemma 4.1. *Let m_1 and m_2 be two mass functions on $\Omega = \{1, \ldots, k\}$ and $I_1, I_2 \subseteq \Omega$ be two subsets such that*

$$m_1(A) \;=\; \begin{cases} u_i & \text{if } A = \{i\} \text{ for } i \in I_1 \\ 1 - \sum_{i \in I_1} u_i & \text{if } A = I_1^{\mathsf{C}} \\ 0 & \text{otherwise} \end{cases} \tag{4.13}$$

and

$$m_2(A) \;=\; \begin{cases} v_i & \text{if } A = \{i\} \text{ for } i \in I_2 \\ 1 - \sum_{i \in I_2} v_i & \text{if } A = I_2^{\mathsf{C}} \\ 0 & \text{otherwise} \end{cases} \tag{4.14}$$

hold. Define $\lambda : \mathcal{P}(\Omega) \to [0,1]$,

$$\lambda(A) \;\coloneqq\; \begin{cases} m_1(\{i\}) \cdot m_2(\{i\}) & \text{if } A = \{i\} \subseteq I_1 \cap I_2 \\ m_1(\{i\}) \cdot m_2(I_2^{\mathsf{C}}) & \text{if } A = \{i\} \subseteq I_1 \setminus I_2 \\ m_1(I_1^{\mathsf{C}}) \cdot m_2(\{i\}) & \text{if } A = \{i\} \subseteq I_2 \setminus I_1 \\ m_1(I_1^{\mathsf{C}}) \cdot m_2(I_2^{\mathsf{C}}) & \text{if } A = (I_1 \cup I_2)^{\mathsf{C}} \neq \emptyset \\ 0 & \text{otherwise} \end{cases} \tag{4.15}$$

as an unnormalized mass function on Ω. For the conflict of m_1 and m_2 holds

$$\kappa(m_1, m_2) \;=\; 1 \;-\; \sum_{i \in I_1 \cup I_2} \lambda\left(\{i\}\right) \;-\; \lambda\left((I_1 \cup I_2)^{\mathsf{C}}\right) \tag{4.16}$$

such that the combination $m_1 \oplus m_2$ using (4.9) is well defined if and only if $\lambda \not\equiv 0$ (i.e. there is an $A \subseteq \Omega$ with $\lambda(A) > 0$) and satisfies:

$$(m_1 \oplus m_2)(A) \;=\; \frac{1}{1 - \kappa(m_1, m_2)} \cdot \lambda(A) \tag{4.17}$$

Proof. To prove the statement, in combination with (4.10) it suffices to show that

$$\lambda(A) \;=\; \sum_{\substack{B, C \subseteq \Omega \\ B \cap C = A}} m_1(B) \cdot m_2(C) \tag{4.18}$$

holds for all $A \subseteq \Omega$. First, assume $\lambda(A) = 0$. For any $A = B \cap C \subseteq \Omega$ with $\lambda(A) = 0$ also $m_1(B) \cdot m_2(C) = 0$ has to hold, thus the equality is straightforward to verify.

Thus, assume that $\lambda(A) > 0$ and in particular $A \neq \emptyset$ holds. This implies that there are subsets $B, C \subseteq \Omega$ such that $A = B \cap C$ with $m_1(B) \cdot m_2(C) > 0$, i.e. $m_1(B) > 0$ and $m_2(C) > 0$ hold. By definition this is equivalent to either $B = \{i\}$, with $i \in I_1$ sufficiently selected, or $B = I_1^{\mathsf{C}} \neq \emptyset$.

1. If $B = \{i\}$ holds:
   From $\emptyset \neq A = B \cap C \subseteq B = \{i\}$ follows $A = \{i\}$. In particular, only a unique selection for C such that $m_2(C) > 0$ holds is possible: either $C = \{i\}$, if

$i \in I_1 \cap I_2$, or $C = I_2^{\complement}$, if $i \notin I_2$. Therefore,

$$\lambda(A) \;=\; m_1(B) \cdot m_2(C) \;=\; \begin{cases} m_1(\{i\}) \cdot m_2(\{i\}) & \text{if } A = \{i\} \subseteq I_1 \cap I_2 \\ m_1(\{i\}) \cdot m_2(I_2^{\complement}) & \text{if } A = \{i\} \subseteq I_1 \setminus I_2 \end{cases} \tag{4.19}$$

holds.

2. If $B = I_1^{\complement} \neq \emptyset$ holds:
From $m_2(C) > 0$ follows $C = \{i\}$, with $i \in I_2 \setminus I_1 = I_2 \cap I_1^{\complement}$ sufficiently selected, or $C = I_2^{\complement} \neq \emptyset$ and in particular B uniquely defines C. Since $A = I_1^{\complement} \cap C$, either $A = I_1^{\complement} \cap \{i\} = \{i\}$ or $A = I_1^{\complement} \cap I_2^{\complement} = (I_1 \cup I_2)^{\complement}$ follows. Therefore,

$$\lambda(A) \;=\; m_1(B) \cdot m_2(C) \;=\; \begin{cases} m_1(I_1^{\complement}) \cdot m_2(\{i\}) & \text{if } A = \{i\} \subseteq I_2 \setminus I_1 \\ m_1(I_1^{\complement}) \cdot m_2(I_2^{\complement}) & \text{if } A = (I_1 \cup I_2)^{\complement} \neq \emptyset \end{cases} \tag{4.20}$$

holds.

Combining the derived equations together with the definitions of m_1 and m_2 yields the overall form of λ as claimed. □

It should be emphasized that the combination $m_1 \oplus m_2$ in lemma 4.1 in particular allows inductive applications to combine multiple mass functions. Furthermore, it is not assumed that the weights u_i and v_i are non-zero. Similarly, they might even sum to one such that $I_1^{\complement}$ or $I_2^{\complement}$ receives zero mass. Thus, these situations cause the combination to place zero mass in some cases. The only constraint for the last result to hold is that the combination exists at all. Here, the following equivalence is obtained as a direct consequence:

Corollary 4.2. *Let m_1 and m_2 be mass functions as in lemma 4.1. The combination $m_1 \oplus m_2$ is well defined if and only if at least one of the following conditions holds:*

1. *There is an $i \in I_1 \cap I_2$ with $u_i \cdot v_i > 0$.*
2. *There is an $i \in I_1 \setminus I_2$ with $u_i > 0$ and $\sum_{j \in I_2} v_j < 1$ holds.*
3. *There is an $i \in I_2 \setminus I_1$ with $v_i > 0$ and $\sum_{j \in I_1} u_j < 1$ holds.*
4. *$\sum_{i \in I_1} u_i < 1$, $\sum_{i \in I_2} v_i < 1$ and $I_1^{\complement} \cap I_2^{\complement} = (I_1 \cup I_2)^{\complement} \neq \emptyset$ hold.*

With respect to the one-vs-all decomposition's modeling using (4.12), the inductive application of lemma 4.1 enables to prove the following closed-form expression:

Theorem 4.3. *Let m_i be the one-vs-all decomposition's mass functions as given by* (4.12) *such that $0 < p_i < 1$ holds for all $i = 1, \ldots, k$. Then, their overall combination is well defined and can be expressed as*

$$\left(\bigoplus_{i=1}^{k} m_i \right)(A) \;=\; \begin{cases} \frac{1}{1-\kappa} \cdot p_i \cdot \prod\limits_{\substack{j=1 \\ j \neq i}}^{k} (1 - p_j) & \text{if } A = \{i\} \text{ for } 1 \leq i \leq k \\ 0 & \text{otherwise} \end{cases} \tag{4.21}$$

while for the conflict

$$\kappa \;=\; 1 - \sum_{i=1}^{k} p_i \cdot \prod_{\substack{j=1 \\ j \neq i}}^{k} (1 - p_j) \tag{4.22}$$

holds.

Proof. First, define $M_\ell \coloneqq \bigoplus_{i=1}^{\ell} m_i$ for $1 \leq \ell \leq k$ and prove that M_ℓ is well defined as well as satisfies

$$M_\ell(A) \;=\; \begin{cases} \frac{1}{1-\kappa_\ell} \cdot p_i \cdot \prod\limits_{\substack{j=1 \\ j\neq i}}^{\ell} (1-p_j) & \text{if } A = \{i\} \text{ for } 1 \leq i \leq \ell \\ \frac{1}{1-\kappa_\ell} \cdot \prod\limits_{i=1}^{\ell} (1-p_i) & \text{if } A = \{1,\ldots,\ell\}^{\mathsf{C}} \neq \emptyset \\ 0 & \text{otherwise} \end{cases} \tag{4.23}$$

where for the conflict $\kappa_0 \coloneqq 0$ and

$$\kappa_\ell \;=\; 1 - \frac{1}{1-\kappa_{\ell-1}} \cdot \left(\sum_{i=1}^{\ell} p_i \cdot \prod_{\substack{j=1 \\ j\neq i}}^{\ell} (1-p_j) \;+\; \mathbb{1}(\ell \neq k) \cdot \prod_{i=1}^{\ell} (1-p_i) \right) \tag{4.24}$$

hold[1]. For $\ell = 1$, it holds $M_1 = m_1$ by definition and the claimed form is straightforward to verify.

If $\ell \geq 2$ and the claim is true for $\ell - 1$, it holds $M_\ell = M_{\ell-1} \oplus m_\ell$ by definition. Thereafter, lemma 4.1 with $I_1 = \{1,\ldots,\ell-1\}$ and $I_2 = \{\ell\}$ as well as $0 < p_\ell < 1$ implies that M_ℓ is well defined. Furthermore, also $I_1 \cap I_2 = \emptyset$ holds such that lemma 4.1 yields three remaining relevant cases for the unnormalized combination $\lambda(A)$:

1. $A = \{i\}$ with $i \in I_1 \setminus I_2 = I_1$, i.e. $1 \leq i \leq \ell - 1$, yields:

$$\lambda(A) \;=\; M_{\ell-1}(\{i\}) \cdot m_\ell(\{\ell\}^{\mathsf{C}}) \;=\; \frac{1}{1-\kappa_{\ell-1}} \cdot p_i \cdot \prod_{\substack{j=1 \\ j\neq i}}^{\ell} (1-p_j) \tag{4.25}$$

2. $A = \{\ell\} = I_2 \setminus I_1 = I_2$ yields:

$$\lambda(A) \;=\; M_{\ell-1}(I_1^{\mathsf{C}}) \cdot m_\ell(\{\ell\}) \;=\; \frac{1}{1-\kappa_{\ell-1}} \cdot \prod_{j=1}^{\ell-1} (1-p_j) \cdot p_\ell \tag{4.26}$$

3. $A = (I_1 \cup I_2)^{\mathsf{C}} = \{1,\ldots,\ell\}^{\mathsf{C}}$ with $\ell < k$ yields:

$$\lambda(A) \;=\; M_{\ell-1}(I_1^{\mathsf{C}}) \cdot m_\ell(\{\ell\}^{\mathsf{C}}) \;=\; \frac{1}{1-\kappa_{\ell-1}} \cdot \prod_{i=1}^{\ell} (1-p_i) \tag{4.27}$$

Besides noting that the second case extends the first to $i = \ell$, the remaining step is the normalization. Here, lemma 4.1 leads to:

$$\begin{aligned} \kappa_\ell \;&=\; 1 \;-\; \sum_{i \in I_1 \cup I_2} \lambda\left(\{i\}\right) \;-\; \lambda\left((I_1 \cup I_2)^{\mathsf{C}}\right) \\ &=\; 1 \;-\; \frac{1}{1-\kappa_{\ell-1}} \cdot \left(\sum_{i=1}^{\ell} p_i \cdot \prod_{\substack{j=1 \\ j\neq i}}^{\ell} (1-p_j) \;+\; \mathbb{1}(\ell \neq k) \cdot \prod_{i=1}^{\ell} (1-p_i) \right) \end{aligned} \tag{4.28}$$

[1]Here, the special case of $\ell = k$ is necessary because $\{1,\ldots,k\}^{\mathsf{C}} = \emptyset$.

In particular for $\ell = k$, the normalization constant becomes

$$1 - \kappa_k \;=\; \frac{1}{1-\kappa_{k-1}} \cdot \left(\sum_{i=1}^{k} p_i \cdot \prod_{\substack{j=1 \\ j \neq i}}^{k} (1 - p_j) \right) \tag{4.29}$$

and thus, normalizing $(1 - \kappa_{k-1}) \cdot \lambda$ in (4.21) requires the normalization constant

$$1 - \kappa \;=\; (1 - \kappa_k) \cdot (1 - \kappa_{k-1}) \;=\; \sum_{i=1}^{k} p_i \cdot \prod_{\substack{j=1 \\ j \neq i}}^{k} (1 - p_j) \tag{4.30}$$

which finalizes the proof. □

The last result yields a closed-form expression for the overall combination of all k one-vs-all mass functions. In particular, it can efficiently be computed and it is numerically more stable to compute the logarithm of the resulting probabilities to transform the product into a sum. Thereafter, the latter can be back-transformed into a probability using the exponential function. Furthermore, the result $\bigoplus_{i=1}^{k} m_i$ is a Bayesian mass function, thus it directly allows the interpretation as a posterior probability estimation. Consequently, the presented evidence-theoretic modeling also yields a multi-class calibration approach because it transforms predictions into an overall posterior probability estimate. The latter even holds if the original probabilities $p_1, \ldots, p_k$ sum to one, i.e. the presented modeling can be used to calibrate multi-class probabilistic classifiers as well.

Besides this observation, the connection between this evidence-theoretic modeling and the classical one-vs-all maximum rule to resolve the class deserves particular interest. In fact, the class with maximum probability *after* applying (4.21) is the same as the one with maximum base probability.

Proposition 4.4. *Under the same assumptions of theorem 4.3, let* $M = \bigoplus_{i=1}^{k} m_i$ *be the overall combination* (4.21) *of the one-vs-all mass functions. Then*

$$\operatorname*{arg\,max}_{1 \leq i \leq k} M(\{i\}) \;=\; \operatorname*{arg\,max}_{1 \leq i \leq k} p_i \tag{4.31}$$

holds such that the resulting class prediction remains unchanged.

Proof. Using

$$q_i \;\coloneqq\; p_i \cdot \prod_{\substack{j=1 \\ j \neq i}}^{k} (1 - p_j) \tag{4.32}$$

for $i = 1, \ldots, k$, it holds $q_i \;=\; \frac{1}{1-\kappa} \cdot M(\{i\})$ such that the maximizers are the same. Here, for two arbitrary indices $1 \leq i, j \leq k$ holds

$$\begin{aligned} q_i - q_j \;&=\; p_i \cdot \prod_{\substack{\ell=1 \\ \ell \neq i}}^{k} (1 - p_\ell) \;-\; p_j \cdot \prod_{\substack{\ell=1 \\ \ell \neq j}}^{k} (1 - p_\ell) \\ &=\; (p_i \cdot (1 - p_j) - p_j \cdot (1 - p_i)) \;\cdot\; \prod_{\substack{\ell=1 \\ \ell \neq i,j}}^{k} (1 - p_\ell) \;=\; (p_i - p_j) \;\cdot\; \underbrace{\prod_{\substack{\ell=1 \\ \ell \neq i,j}}^{k} (1 - p_\ell)}_{>0} \end{aligned}$$

such that $q_i \geq q_j$ holds if and only if $p_i \geq p_j$ as well as $q_i > q_j$ holds if and only if $p_i > p_j$, respectively. Therefore, the claim is proven. □

In summary, the evidence-theoretic modeling based on the one-vs-all decomposition recovers the same decision rule, but also computes a posterior probability estimation. Based on these insights, the next part performs a similar modeling based on extended decompositions and in particular the one-vs-one reduction before focusing on dynamic classification based on evidence theory.

4.2.2 One-vs-One Decomposition

As presented in chapter 2, there are many reference works that report superior results of the one-vs-one decomposition in comparison with the one-vs-all reduction. Consequently, it is especially interesting to also perform an evidence-theoretic modeling of it. In full analogy to chapter 2, a probability matrix $\Phi = \Phi(\mathsf{x})$ as in (2.23) is assumed to be given that contains the individual pairwise probabilities $\phi_{i,j}(\mathsf{x}) \in [0, 1]$ satisfying $\phi_{i,j}(\mathsf{x}) + \phi_{j,i}(\mathsf{x}) = 1$. As before, the explicit dependency on the respectively predicted instance x is omitted in the following part.

Based on the previous results, a feasible strategy to perform an evidence-theoretic modeling of the one-vs-one decomposition that avoids the drawbacks, most importantly the exponential complexity, from existing works [Quost et al. 2007] is to transform the pairwise probabilities into mass functions and thereafter prove a closed-form expression for their combination. Ideally, this will also result in a Bayesian mass function such that no further steps are required to obtain a probabilistic interpretation. Unluckily, this is not directly possible as it is in case of the one-vs-all decomposition. More precisely, analogously modeling mass functions using the pairwise probabilities would yield

$$m_{i,j}(A) \coloneqq \begin{cases} \phi_{i,j} & \text{if } A = \{i\} \\ \phi_{j,i} & \text{if } A = \{j\} \\ 0 & \text{otherwise} \end{cases} \tag{4.33}$$

for all pairs $1 \leq i < j \leq k$. However, the combination $m_{1,2} \oplus m_{1,3} \oplus m_{2,3}$ even for $k = 3$ classes does not exist as there is no event on which all mass functions induce positive belief. Straightforward computation also yields full conflict.

Alternatively, for $1 \leq i < j \leq k$ the pairwise probabilities could be split into two mass functions

$$m_{i,j}(A) \coloneqq \begin{cases} \phi_{i,j} & \text{if } A = \{i\} \\ \phi_{j,i} & \text{if } A = \{i\}^{\mathsf{C}} \\ 0 & \text{otherwise} \end{cases} \quad \text{and} \quad m_{j,i}(A) \coloneqq \begin{cases} \phi_{j,i} & \text{if } A = \{j\} \\ \phi_{i,j} & \text{if } A = \{j\}^{\mathsf{C}} \\ 0 & \text{otherwise} \end{cases} \tag{4.34}$$

respectively. Here, a combination $\bigoplus_{i=1}^{k-1} \bigoplus_{j=i+1}^{k} (m_{i,j} \oplus m_{j,i})$ could be computed under relatively mild assumptions regarding the pairwise probabilities. However, the modeling is unjustified as it places positive mass on classes in the complements that are not consistent with the underlying classifiers.

Furthermore, the mass function (4.33) places zero mass on all classes where the underlying classifier used to compute the probabilities is incompetent. On the other hand, the evidence theory's closed world assumption requires at least a single set on which all mass functions induce positive belief. Therefore, a reasonable strategy to develop a similar evidence-theoretic modeling based on the one-vs-one decomposition also requires to address the non-competence problem.

As presented in full detail in section 2.3, one of the most promising existing techniques to tackle the non-competence problem multiplies the pairwise probabilities $\phi_{i,j}$ with a weight $\mathsf{w}_{i,j}$. In the respective works, the latter is computed by training additional correcting classifiers to separate the pair of classes $\{i, j\}$ from the remaining set of all other ones. Still, there are possible generalizations on how to estimate the weights that will be discussed in full detail in subsection 5.5.2. Here, it is only assumed that $\mathsf{w}_{i,j} \in [0, 1]$ as well as $\mathsf{w}_{i,j} = \mathsf{w}_{j,i}$ holds (i.e. the weights are symmetric) for all pairs $\{i, j\}$.

Even though there also is a formal reasoning [Reid 2010] for the multiplicative combination of pairwise probabilities and weights given by (2.30), this is not valid in a Bayesian sense because the weights cannot be interpreted as estimates of the posterior probabilities $P(\mathsf{y} \in \{i, j\} \mid \mathsf{x})$, as discussed in full detail in chapter 2. However in an evidence-theoretic context, this is unproblematic and emphasizes the advantages of evidence theory for the current application. Therefore, the following mass functions

$$m_{i,j}(A) \;\coloneqq\; \begin{cases} \phi_{i,j} \cdot \mathsf{w}_{i,j} & \text{if } A = \{i\} \\ \phi_{j,i} \cdot \mathsf{w}_{i,j} & \text{if } A = \{j\} \\ 1 - \mathsf{w}_{i,j} & \text{if } A = \{i, j\}^{\mathsf{C}} \\ 0 & \text{otherwise} \end{cases} \tag{4.35}$$

are defined for all pairs $1 \leq i, j \leq k$ with $i \neq j$. These do not only enable to systematically address the non-competence problem but also to prove a closed-form expression for the combination of all pairwise mass functions.

As $\mathsf{w}_{i,j} = \mathsf{w}_{j,i}$ holds, this definition consists of two equivalent subsets containing all mass functions $m_{i,j}$ with index pairs (i, j) where either $i < j$ or $j < i$ holds. These can be interpreted as an upper or a lower triangle, respectively. Here, the overall combination over each triangular set

$$\bigoplus_{i=1}^{k-1} \bigoplus_{j=i+1}^{k} m_{i,j} \;=\; \bigoplus_{i=2}^{k} \bigoplus_{j=1}^{i-1} m_{i,j} \tag{4.36}$$

is of interest. Still, combining the mass functions in the right-hand side manner is advantageous for the presented proof of the closed-form expression.

Clearly, computing the combination is more complex than in the previous case based on the one-vs-all decomposition in theorem 4.3. As a first step, computing a closed-form expression for the row-wise combinations $\bigoplus_{j=1}^{i-1} m_{i,j}$ is possible using lemma 4.1. Thereafter, all of them are combined into a single mass function that equals the overall combination for which a closed-form expression will be proven.

Lemma 4.5. *Let $m_{i,j}$ be the one-vs-one decomposition's mass functions as given by (4.35) such that $0 < \phi_{i,j} < 1$ and $0 < \mathsf{w}_{i,j} < 1$ hold for all $i, j = 1, \ldots, k$ with $i \neq j$. For $1 \leq \ell < i \leq k$ define $\lambda_{i,\ell} : \mathcal{P}(\{1, \ldots, k\}) \to [0, 1]$,*

$$\lambda_{i,\ell}(A) \;\coloneqq\; \begin{cases} \displaystyle\prod_{j=1}^{\ell} m_{i,j}(\{i\}) & \text{if } A = \{i\} \\ m_{i,j}(\{j\}) \cdot \displaystyle\prod_{\substack{s=1 \\ s \neq j}}^{\ell} m_{i,s}(\{i, s\}^{\mathsf{C}}) & \text{if } A = \{j\} \text{ for } 1 \leq j \leq \ell \\ \displaystyle\prod_{j=1}^{\ell} m_{i,j}(\{i, j\}^{\mathsf{C}}) & \text{if } A = \{1, \ldots, \ell, i\}^{\mathsf{C}} \neq \emptyset \\ 0 & \text{otherwise} \end{cases} \tag{4.37}$$

as an unnormalized mass function. Then, for given $1 < i \leq k$ *the combination of the first* $1 \leq \ell < i$ *mass functions* $m_{i,j}$ *is well defined and can be expressed as*

$$R_{i,\ell}(A) \;\coloneqq\; \left(\bigoplus_{j=1}^{\ell} m_{i,j}\right)(A) \;=\; \frac{1}{1-\kappa_{i,\ell}} \cdot \lambda_{i,\ell}(A) \tag{4.38}$$

while for the conflict

$$\kappa_{i,\ell} \;=\; 1 \;-\; \sum_{j=1}^{\ell} \lambda_{i,\ell}(\{j\}) \;-\; \lambda_{i,\ell}(\{i\}) \;-\; \lambda_{i,\ell}(\{1,\ldots,\ell,i\}^{\mathsf{C}}) \tag{4.39}$$

holds.

Proof. For $1 = \ell < i$, $\lambda_{i,1} = m_{i,1}$ and $\kappa_{i,1} = 0$ hold from the normalization of $m_{i,1}$ such that the claim is true. Therefore, let the statement be true for $R_{i,\ell-1}$. By definition $R_{i,\ell} = R_{i,\ell-1} \oplus m_{i,\ell}$ holds. By the induction hypothesis and the definition of $m_{i,\ell}$, lemma 4.1 is applicable with index sets $I_1 = \{1,\ldots,\ell-1,i\}$ and $I_2 = \{\ell,i\}$, respectively. Furthermore, it implies that $R_{i,\ell}$ is well defined and using

$$\lambda(A) \;=\; \begin{cases} R_{i,\ell-1}(\{i\}) \cdot m_{i,\ell}(\{i\}) & \text{if } A = \{i\} = I_1 \cap I_2 \\ R_{i,\ell-1}(\{j\}) \cdot m_{i,\ell}(I_2^{\mathsf{C}}) & \text{if } A = \{j\} \subseteq \{1,\ldots,\ell-1\} = I_1 \setminus I_2 \\ R_{i,\ell-1}(I_1^{\mathsf{C}}) \cdot m_{i,\ell}(\{\ell\}) & \text{if } A = \{\ell\} = I_2 \setminus I_1 \\ R_{i,\ell-1}(I_1^{\mathsf{C}}) \cdot m_{i,\ell}(I_2^{\mathsf{C}}) & \text{if } A = \{1,\ldots,\ell,i\}^{\mathsf{C}} = (I_1 \cup I_2)^{\mathsf{C}} \neq \emptyset \\ 0 & \text{otherwise} \end{cases} \tag{4.40}$$

as well as

$$\begin{aligned} \kappa \;&=\; 1 \;-\; \sum_{j \in I_1 \cup I_2} \lambda(\{j\}) \;-\; \lambda\left((I_1 \cup I_2)^{\mathsf{C}}\right) \\ &=\; 1 \;-\; \sum_{j=1}^{\ell} \lambda(\{j\}) \;-\; \lambda(\{i\}) \;-\; \lambda(\{1,\ldots,\ell,i\}^{\mathsf{C}}) \end{aligned} \tag{4.41}$$

yields:

$$R_{i,\ell}(A) \;=\; \frac{1}{1-\kappa} \cdot \lambda(A) \tag{4.42}$$

Here, substituting (4.40) and (4.37) into $R_{i,\ell-1}(A) = \frac{1}{1-\kappa_{i,\ell-1}} \cdot \lambda_{i,\ell-1}(A)$ enables to integrate $m_{i,\ell}$ into the products in $R_{i,\ell-1}$ by extending their ranges from $\ell-1$ to ℓ. In particular,

$$\lambda(A) \cdot (1-\kappa_{i,\ell-1}) \;=\; \begin{cases} \prod_{j=1}^{\ell} m_{i,j}(\{i\}) & \text{if } A = \{i\} \\ m_{i,j}(\{j\}) \cdot \prod_{\substack{s=1 \\ s \neq j}}^{\ell} m_{i,s}(\{i,s\}^{\mathsf{C}}) & \text{if } A = \{j\} \subseteq \{1,\ldots,\ell-1\} \\ \prod_{j=1}^{\ell-1} m_{i,j}(\{i,j\}^{\mathsf{C}}) \cdot m_{i,\ell}(\{\ell\}) & \text{if } A = \{\ell\} \\ \prod_{j=1}^{\ell} m_{i,j}(\{i,j\}^{\mathsf{C}}) & \text{if } A = \{1,\ldots,\ell,i\}^{\mathsf{C}} \neq \emptyset \\ 0 & \text{otherwise} \end{cases}$$

is obtained and by noting that the third case is equivalent to the second one extended

to ℓ, $\lambda(A) \cdot (1 - \kappa_{i,\ell-1}) = \lambda_{i,\ell}(A)$ holds. In combination with (4.38) as well as (4.42), this implies $1 - \kappa_{i,\ell} = (1 - \kappa_{i,\ell-1}) \cdot (1 - \kappa)$ and thus

$$R_{i,\ell-1}(A) \;=\; \frac{1}{1-\kappa} \cdot \lambda(A) \;=\; \frac{1}{(1-\kappa)\cdot(1-\kappa_{i,\ell-1})} \cdot \lambda_{i,\ell}(A) \;=\; \frac{1}{(1-\kappa_{i,\ell})} \cdot \lambda_{i,\ell}(A)$$

finalizes the proof. □

Using the last result, the overall combination of all one-vs-one mass functions (4.35) can be simplified to

$$\bigoplus_{i=2}^{k} \bigoplus_{j=1}^{i-1} m_{i,j} \;=\; \bigoplus_{i=2}^{k} R_{i,i-1} \tag{4.43}$$

such that the remaining task is to combine all $R_{i,i-1}$ where for the latter already a closed-form expression exists. Arranging the combination in a lower triangular shape, the $R_{i,i-1}$ can be interpreted as the i-th row's mass function. Therefore, the remaining task can be interpreted as combining the row mass functions. For this, a closed-form expression is given by the following result:

Theorem 4.6. *Let $m_{i,j}$ be the one-vs-one decomposition's mass functions as given by (4.35) such that $0 < \phi_{i,j} < 1$ and $0 < \mathsf{w}_{i,j} < 1$ hold for all $i, j = 1, \ldots, k$ with $i \neq j$ and $R_{i,i-1}$ be their combination as in lemma 4.5. Furthermore, for $2 \leq i \leq k$ define $\lambda_i : \mathcal{P}(\{1, \ldots, k\}) \to [0, 1]$,*

$$\lambda_i(A) \coloneqq \begin{cases} \displaystyle\prod_{s=1}^{j-1} m_{j,s}(\{j\}) \prod_{t=j+1}^{i} m_{t,j}(\{j\}) \prod_{\substack{s=1 \\ s\neq j}}^{i-1} \prod_{\substack{t=s+1 \\ t\neq j}}^{i} m_{t,s}(\{s,t\}^{\mathsf{C}}) & \text{if } A = \{j\},\ j \leq i \\ \displaystyle\prod_{s=1}^{i-1} \prod_{t=s+1}^{i} m_{t,s}(\{s,t\}^{\mathsf{C}}) & \text{if } A = \{1,\ldots,i\}^{\mathsf{C}} \neq \emptyset \\ 0 & \text{otherwise} \end{cases} \tag{4.44}$$

as an unnormalized mass function. Then, the combination $\bigoplus_{j=2}^{i} R_{j,j-1}$ is well defined for each $2 \leq i \leq k$ and satisfies

$$\left(\bigoplus_{j=2}^{i} R_{j,j-1} \right)(A) \;=\; \frac{1}{1-\kappa_i} \cdot \lambda_i(A) \tag{4.45}$$

where for the conflict

$$\kappa_i \;=\; 1 \;-\; \sum_{j=1}^{i} \lambda_i(\{j\}) \;-\; \lambda_i(\{1,\ldots,i\}^{\mathsf{C}}) \tag{4.46}$$

holds.

Proof. By applying lemma 4.5 for $i = 2$, the base case of $m_{2,1} = \lambda_{2,1} = \lambda_2$ is straightforward to verify. Thus,

$$\bigoplus_{j=2}^{2} R_{j,j-1} \;=\; R_{2,1} \;=\; m_{2,1} \;=\; \frac{1}{1-\kappa_{2,1}} \cdot \lambda_{2,1} \;=\; \frac{1}{1-\kappa_2} \cdot \lambda_2 \tag{4.47}$$

holds by definition (4.38) with $\kappa_{2,1} = \kappa_2 = 0$ as claimed. Therefore, let the statement be true for $2 < i \leq k - 1$.

By combining (4.45) from the induction hypothesis for $i-1$ and (4.38) with $\ell = i-1$ from lemma 4.5

$$\bigoplus_{j=2}^{i} R_{j,j-1} \;=\; \left(\bigoplus_{j=2}^{i-1} R_{j,j-1}\right) \oplus R_{i,i-1} \;=\; \frac{1}{1-\kappa_{i-1}} \cdot \lambda_{i-1} \oplus \frac{1}{1-\kappa_{i,i-1}} \cdot \lambda_{i,i-1} \tag{4.48}$$

is obtained. Here, lemma 4.1 is applicable with $\frac{1}{1-\kappa_{i-1}} \cdot \lambda_{i-1}$ on $I_1 = \{1,\dots,i-1\}$ by the induction hypothesis and $\frac{1}{1-\kappa_{i,i-1}} \cdot \lambda_{i,i-1}$ on $I_2 = \{1,\dots,i\} = I_1 \cup \{i\}$ by lemma 4.5, respectively. In particular, it implies that the combination is well defined such that with

$$\lambda(A) \;=\; \begin{cases} \lambda_{i-1}(\{j\}) \cdot \lambda_{i,i-1}(\{j\}) & \text{if } A = \{j\} \subseteq I_1 \cap I_2 = I_1 \\ \lambda_{i-1}(I_1^{\mathsf{C}}) \cdot \lambda_{i,i-1}(\{i\}) & \text{if } A = \{i\} = I_2 \setminus I_1 \\ \lambda_{i-1}(I_1^{\mathsf{C}}) \cdot \lambda_{i,i-1}(I_2^{\mathsf{C}}) & \text{if } A = (I_1 \cup I_2)^{\mathsf{C}} = I_2^{\mathsf{C}} \neq \emptyset \\ 0 & \text{otherwise} \end{cases} \tag{4.49}$$

and

$$\kappa \;=\; 1 - \sum_{j \in I_1 \cup I_2} \lambda(\{j\}) - \lambda\left((I_1 \cup I_2)^{\mathsf{C}}\right) \;=\; 1 - \sum_{j=1}^{i} \lambda(\{j\}) - \lambda(\{1,\dots,i\}^{\mathsf{C}}) \tag{4.50}$$

it holds:

$$\bigoplus_{j=2}^{i} R_{j,j-1} = \frac{1}{1-\kappa} \cdot \lambda \tag{4.51}$$

It should be emphasized that the multiplication by $\frac{1}{1-\kappa_{i-1}} \cdot \frac{1}{1-\kappa_{i,i-1}} > 0$ can be ignored because it cancels out during the normalization of λ. Consequently, only the following cases in (4.49) are relevant:

1. $A = \{j\}$ with $j \leq i-1$:

$$\begin{aligned}
\lambda(A) \;&=\; \lambda_{i-1}(A) \cdot \lambda_{i,i-1}(A) \;=\; \underbrace{\lambda_{i-1}(\{j\})}_{i-1 \text{ in (4.44)}} \cdot \underbrace{\lambda_{i,i-1}(\{j\})}_{\ell = i-1 \text{ in (4.37)}} \\
&= \prod_{s=1}^{j-1} m_{j,s}(\{j\}) \prod_{t=j+1}^{i-1} m_{t,j}(\{j\}) \prod_{\substack{s=1 \\ s\neq j}}^{i-2} \prod_{\substack{t=s+1 \\ t\neq j}}^{i-1} m_{t,s}(\{s,t\}^{\mathsf{C}}) \cdot m_{i,j}(\{j\}) \prod_{\substack{s=1 \\ s\neq j}}^{i-1} m_{i,s}(\{i,s\}^{\mathsf{C}}) \\
&= \prod_{s=1}^{j-1} m_{j,s}(\{j\}) \underbrace{\prod_{t=j+1}^{i} m_{t,j}(\{j\})}_{\text{extended to } t=i} \prod_{\substack{s=1 \\ s\neq j}}^{i-2} \prod_{\substack{t=s+1 \\ t\neq j}}^{i-1} m_{t,s}(\{s,t\}^{\mathsf{C}}) \prod_{\substack{s=1 \\ s\neq j}}^{i-1} m_{i,s}(\{i,s\}^{\mathsf{C}}) \\
&= \prod_{s=1}^{j-1} m_{j,s}(\{j\}) \prod_{t=j+1}^{i} m_{t,j}(\{j\}) \prod_{\substack{s=1 \\ s\neq j}}^{i-2} \underbrace{\prod_{\substack{t=s+1 \\ t\neq j}}^{i} m_{t,s}(\{s,t\}^{\mathsf{C}})}_{\text{extended to } t=i} \; m_{i,i-1}(\{i-1,i\}^{\mathsf{C}}) \\
&= \prod_{s=1}^{j-1} m_{j,s}(\{j\}) \prod_{t=j+1}^{i} m_{t,j}(\{j\}) \underbrace{\prod_{\substack{s=1 \\ s\neq j}}^{i-1} \prod_{\substack{t=s+1 \\ t\neq j}}^{i} m_{t,s}(\{s,t\}^{\mathsf{C}})}_{\text{extended to } s=i-1} \overset{(4.44)}{=} \lambda_i(\{j\}) \;=\; \lambda_i(A)
\end{aligned}$$

2. $A = \{i\}$:

$$\begin{aligned}
\lambda(A) &= \lambda_{i-1}(I_1^{\complement}) \cdot \lambda_{i,i-1}(\{i\}) = \underbrace{\lambda_{i-1}(\{1,\dots,i-1\}^{\complement})}_{i-1 \text{ in (4.44)}} \cdot \underbrace{\lambda_{i,i-1}(\{i\})}_{\ell=i-1 \text{ in (4.37)}} \\
&= \prod_{s=1}^{i-2} \prod_{t=s+1}^{i-1} m_{t,s}(\{s,t\}^{\complement}) \cdot \prod_{j=1}^{i-1} m_{i,j}(\{i\}) \\
&= \prod_{s=1}^{i-1} m_{i,s}(\{i\}) \underbrace{\prod_{t=i+1}^{i} m_{t,i}(\{i\})}_{=1} \cdot \prod_{\substack{s=1 \\ s\neq i}}^{i-1} \prod_{\substack{t=s+1 \\ t\neq i}}^{i} m_{t,s}(\{s,t\}^{\complement}) \overset{(4.44)}{=} \lambda_i(\{i\}) = \lambda_i(A)
\end{aligned}$$

3. $A = (I_1 \cup I_2)^{\complement} = I_2^{\complement} = \{1,\dots,i\}^{\complement} \neq \emptyset$:

$$\begin{aligned}
\lambda(A) &= \lambda_{i-1}(I_1^{\complement}) \cdot \lambda_{i,i-1}(I_2^{\complement}) = \underbrace{\lambda_{i-1}(\{1,\dots,i-1\}^{\complement})}_{i-1 \text{ in (4.44)}} \cdot \underbrace{\lambda_{i,i-1}(\{1,\dots,i\}^{\complement})}_{\ell=i-1 \text{ in (4.37)}} \\
&= \prod_{s=1}^{i-2} \prod_{t=s+1}^{i-1} m_{t,s}(\{s,t\}^{\complement}) \cdot \prod_{j=1}^{i-1} m_{i,j}(\{i,j\}^{\complement}) \\
&= \prod_{s=1}^{i-2} \prod_{t=s+1}^{i} m_{t,s}(\{s,t\}^{\complement}) \cdot m_{i,i-1}(\{i-1,i\}^{\complement}) \\
&= \prod_{s=1}^{i-1} \prod_{t=s+1}^{i} m_{t,s}(\{s,t\}^{\complement}) = \lambda_i(\{1,\dots,i\}^{\complement}) = \lambda_i(A)
\end{aligned}$$

Thus in total, $\lambda(A) = \lambda_i(A)$ is proven. This implies $1-\kappa_i = 1-\kappa$ and in combination with (4.51), the claim is proven. □

Similar to the one-vs-all decomposition, the last result yields a closed-form expression for the combination of all one-vs-one mass functions. In particular, substituting the definition (4.35) of $m_{i,j}$ into the extreme case λ_k in the last result (4.44) where $\{1,\dots,k\}^{\complement} = \emptyset$ holds, a Bayesian mass function is obtained.

Corollary 4.7. *Let $m_{i,j}$ be the one-vs-one decomposition's mass functions as given by (4.35) such that $0 < \phi_{i,j} < 1$ and $0 < w_{i,j} < 1$ hold for all $i,j = 1,\dots,k$ with $i \neq j$. Their overall combination is a well defined Bayesian mass function and satisfies*

$$\left(\bigoplus_{i=2}^{k} \bigoplus_{j=1}^{i-1} m_{i,j}\right)(A) = \begin{cases} \frac{p_i}{\sum_{j=1}^{k} p_j} & \text{if } A = \{i\} \text{ for } 1 \leq i \leq k \\ 0 & \text{otherwise} \end{cases} \tag{4.52}$$

such that for the unnormalized probabilities

$$p_i = \prod_{\substack{j=1 \\ j\neq i}}^{k} (\phi_{i,j} \cdot w_{i,j}) \cdot \prod_{\substack{s=1 \\ s\neq i}}^{k-1} \prod_{\substack{t=s+1 \\ t\neq i}}^{k} (1 - w_{s,t}) \tag{4.53}$$

holds for all $i = 1,\dots,k$.

In full analogy to the one-vs-all decomposition, the overall combination allows the interpretation as a posterior probability estimation and it is numerically more stable to transform the product into a sum by computing the logarithm. Using (4.53) to

compute the probabilities requires $\mathcal{O}(k^2)$ operations, therefore applying it for each class $i = 1, \ldots, k$ would require $\mathcal{O}(k^3)$ calculations. However, by substituting

$$\prod_{\substack{s=1 \\ s \neq i}}^{k-1} \prod_{\substack{t=s+1 \\ t \neq i}}^{k} (1 - \mathsf{w}_{s,t}) \;=\; \frac{\prod_{s=1}^{k-1} \prod_{t=s+1}^{k} (1 - \mathsf{w}_{s,t})}{\prod_{\substack{j=1 \\ j \neq i}}^{k} (1 - \mathsf{w}_{i,j})} \tag{4.54}$$

into (4.53) yields

$$p_i \;\propto\; \prod_{\substack{j=1 \\ j \neq i}}^{k} (\phi_{i,j} \cdot \mathsf{w}_{i,j}) \cdot \left(\prod_{\substack{j=1 \\ j \neq i}}^{k} \cdot (1 - \mathsf{w}_{i,j}) \right)^{-1} \;=\; \prod_{\substack{j=1 \\ j \neq i}}^{k} \frac{\phi_{i,j} \cdot \mathsf{w}_{i,j}}{1 - \mathsf{w}_{i,j}} \tag{4.55}$$

for $i = 1, \ldots, k$ such that the remaining multiplicative constant is irrelevant after the normalization in (4.52) and, as a result, can simply be ignored while computing the unnormalized probabilities.

4.2.3 New Decompositions

Besides applying the one-vs-all and one-vs-one decomposition while especially aiming at the non-competence problem in the latter, the presented approach can additionally be used to create new approaches to decomposition-based classification. Usually, any decomposition is designed such that varying (depending on the respective decomposition) sets of classes are separated from each other, but at least one of them only contains a single class.

Here, evidence theory even allows the application of decompositions that do not restrict to this assumption. As an particular example, a *two-vs-all decomposition* is presented that is constructed similarly to aforementioned correcting classifiers to separate each pair of classes from all other $k-2$ classes, but it does not combine them with other classifiers. Thus, there are $\binom{k}{2}$ individual ones in total, which coincides with the one-vs-one decomposition, but each is trained on the whole training data such that there is no non-competence problem.

At prediction time, each classifier computes a pairwise membership probability $\mathsf{w}_{i,j}$, which – as previously discussed – is not an estimate of $P(\mathsf{y} \in \{i,j\} \mid \mathsf{x})$. Still, this is unproblematic in an evidence-theoretic modeling. Here, it is consistent to define the mass functions

$$m_{i,j}(A) \;\coloneqq\; \begin{cases} \mathsf{w}_{i,j} & \text{if } A = \{i,j\} \\ 1 - \mathsf{w}_{i,j} & \text{if } A = \{i,j\}^{\mathsf{C}} \\ 0 & \text{otherwise} \end{cases} \tag{4.56}$$

for all pairs $1 \leq i, j \leq k$ with $i \neq j$. Similar to the previous modeling in subsection 4.2.2, this definition consists of two equivalent subsets containing the same mass functions that are arranged in either an upper or a lower triangle, respectively. Similarly, the task is to compute the overall combination over each triangular-shaped set

$$\bigoplus_{i=1}^{k-1} \bigoplus_{j=i+1}^{k} m_{i,j} \;=\; \bigoplus_{i=2}^{k} \bigoplus_{j=1}^{i-1} m_{i,j} \tag{4.57}$$

where the right-hand side is advantageous for the presented derivation of a closed-form expression for the combination.

Here, it is important to emphasize that even though the statement resembles the expressions obtained with lemma 4.5, its assumptions are different. In case of the one-vs-one decomposition, the respective individual mass function $m_{i,j}$ puts positive mass on both sets, $\{i\}$ and $\{j\}$, while in case of the presented modeling of two-vs-all reduction, there are no focal sets containing only a single element. Therefore, lemma 4.1 is not applicable and thus, the following result is neither an implication nor a generalization of lemma 4.5.

Lemma 4.8. *Let $m_{i,j}$ be the two-vs-all decomposition's mass functions as given by (4.56) such that $0 < \mathsf{w}_{i,j} < 1$ hold for all $i, j = 1, \ldots, k$ with $i \neq j$. Furthermore, for $2 \leq \ell < i \leq k$ let $\lambda_{i,\ell} : \mathcal{P}(\{1, \ldots, k\}) \to [0, 1]$,*

$$\lambda_{i,\ell}(A) \;=\; \begin{cases} \mathsf{w}_{i,j} \cdot \prod\limits_{\substack{s=1 \\ s \neq j}}^{\ell} (1 - \mathsf{w}_{i,s}) & \text{if } A = \{j\} \text{ for } 1 \leq j \leq \ell \\ \prod\limits_{j=1}^{\ell} \mathsf{w}_{i,j} & \text{if } A = \{i\} \\ \prod\limits_{j=1}^{\ell} (1 - \mathsf{w}_{i,j}) & \text{if } A = \{1, \ldots, \ell, i\}^{\mathsf{C}} \neq \emptyset \\ 0 & \text{otherwise} \end{cases} \tag{4.58}$$

be an unnormalized mass function. Then, for given $3 \leq i \leq k$ the combination of the first $2 \leq \ell < i$ mass functions is well defined and can be expressed as

$$R_{i,\ell}(A) \;\coloneqq\; \left(\bigoplus_{j=1}^{\ell} m_{i,j} \right)(A) \;=\; \frac{1}{1 - \kappa_{i,\ell}} \cdot \lambda_{i,\ell}(A) \tag{4.59}$$

while for the conflict

$$\kappa_{i,\ell} \;=\; 1 \;-\; \sum_{j=1}^{\ell} \lambda_{i,\ell}(\{j\}) \;-\; \lambda_{i,\ell}(\{i\}) \;-\; \lambda_{i,\ell}(\{1, \ldots, \ell, i\}^{\mathsf{C}}) \tag{4.60}$$

holds.

Proof. To prove the statement, at first the unnormalized combination of $m_{i,1}$ and $m_{i,2}$ is defined

$$\lambda_{i,2}(A) \;\coloneqq\; \sum_{\substack{B, C \subseteq \Omega \\ B \cap C = A}} m_{i,1}(B) \cdot m_{i,2}(C) \tag{4.61}$$

such that the claim is true for $\ell = 2 < i$ if (4.58) is recovered and there is at least one set A such that $\lambda_{i,2}(A) > 0$ holds, i.e. the normalization is possible. Therefore, assume that $A = B \cap C$ is given with $\lambda_{i,2}(A) > 0$.

1. If $i \in A$ holds:
   From $i \in A = B \cap C$ follows $i \in B$ and $i \in C$ such that the properties of $m_{i,1}$ and $m_{i,2}$ imply $B = \{i, 1\}$ and $C = \{i, 2\}$, respectively. In particular, both sets are uniquely defined such that $A = \{i\} = \{i, 1\} \cap \{i, 2\}$, which yields $\lambda_{i,2}(A) = \mathsf{w}_{i,1} \cdot \mathsf{w}_{i,2}$.

2. If $1 \in A$ holds:
   Analogously to the previous case, it follows $B = \{i, 1\}$, $C = \{i, 2\}^{\mathsf{C}}$ and $A = \{1\}$ with $\lambda_{i,2}(A) = \mathsf{w}_{i,1} \cdot (1 - \mathsf{w}_{i,2})$ because $i > 2$.

3. If $2 \in A$ holds:
   Similarly, $B = \{i,1\}^{\mathsf{C}}$, $C = \{i,2\}$ and $A = \{2\}$ with $\lambda(A) = (1 - \mathsf{w}_{i,1}) \cdot \mathsf{w}_{i,2}$ is obtained because $i > 2$.

4. If $\{1,2,i\} \cap A = \emptyset$ holds:
   Finally, $B = \{i,1\}^{\mathsf{C}}$, $C = \{i,2\}^{\mathsf{C}}$ has to hold, which implies $A = \{1,2,i\}^{\mathsf{C}}$ with $\lambda(A) = (1 - \mathsf{w}_{i,1}) \cdot (1 - \mathsf{w}_{i,2})$ for $i < k$ because $i > 2$.

In total, no other options for A, B and C are possible such that exactly the claimed form is recovered. Therefore, let the claim be true for $\ell - 1 < i \leq k$ and similarly define the unnormalized combination

$$\lambda_{i,\ell}(A) \;\coloneqq\; (1 - \kappa_{\ell-1}) \;\cdot \sum_{\substack{B,C \subseteq \Omega \\ B \cap C = A}} R_{i,\ell-1}(B) \cdot m_{i,\ell}(C) \tag{4.62}$$

where the claim is proven if there is at least one $A \subseteq \Omega$ with $\lambda_{i,\ell}(A) > 0$ such that the normalized combination exists and the functional form (4.58) is recovered for it as well. Therefore, let $A = B \cap C$ be given with $\lambda_{i,2}(A) > 0$.

1. If $i \in A$ holds:
   From $i \in A = B \cap C$ with $R_{i,\ell-1}(B) > 0$ and $m_{i,\ell}(C) > 0$ follows $B = \{i\}$ from the induction hypothesis and $C = \{i,\ell\}$ from the definition of $m_{i,\ell}$. In particular, both B and C are uniquely defined implying $A = \{i\}$ as well as

   $$\lambda_{i,\ell}(A) \;=\; \prod_{j=1}^{\ell-1} \mathsf{w}_{i,j} \cdot \mathsf{w}_{i,\ell} \;=\; \prod_{j=1}^{\ell} \mathsf{w}_{i,j} \tag{4.63}$$

   which is positive as all $\mathsf{w}_{i,j}$ are. Hence, the existence of the combination is already proven.

2. If $\{1,\ldots,\ell-1\} \cap A \neq \emptyset$ holds:
   Here, in particular $\{1,\ldots,\ell-1\} \cap B \neq \emptyset$ has to hold from $A \subseteq B$. This implies $B = \{j\}$ for $1 \leq j \leq \ell - 1$ sufficiently selected by the induction hypothesis. Thus, also $A = \{j\}$ holds such that $C = \{i,\ell\}^{\mathsf{C}}$ is the only possible selection. In total

   $$\lambda_{i,\ell}(A) \;=\; \mathsf{w}_{i,j} \cdot \prod_{\substack{s=1 \\ s \neq j}}^{\ell-1} (1 - \mathsf{w}_{i,s}) \cdot (1 - \mathsf{w}_{i,\ell}) \;=\; \mathsf{w}_{i,j} \cdot \prod_{\substack{s=1 \\ s \neq j}}^{\ell} (1 - \mathsf{w}_{i,s}) \tag{4.64}$$

   is obtained.

3. If $\ell \in A$ holds:
   Now, $\ell \in A = B \cap C$ implies $B = \{1,\ldots,\ell-1,i\}^{\mathsf{C}}$ and $C = \{i,\ell\}$ such that B and C are uniquely defined as well as

   $$\lambda_{i,\ell}(A) \;=\; \prod_{j=1}^{\ell-1} (1 - \mathsf{w}_{i,j}) \cdot \mathsf{w}_{i,\ell} \;=\; \mathsf{w}_{i,\ell} \cdot \prod_{\substack{s=1 \\ s \neq \ell}}^{\ell} (1 - \mathsf{w}_{i,s}) \tag{4.65}$$

   holds.

4. If $\{1,\ldots,\ell,i\} \cap A = \emptyset$ holds:
   In the last case, $B = \{1,\ldots,\ell-1,i\}^{\mathsf{C}}$ as well as $C = \{i,\ell\}^{\mathsf{C}}$ have to hold such

that B and C are uniquely defined and

$$\lambda_{i,\ell}(A) \;=\; \prod_{j=1}^{\ell-1}(1-\mathsf{w}_{i,j})\cdot(1-\mathsf{w}_{i,\ell}) \;=\; \prod_{j=1}^{\ell}(1-\mathsf{w}_{i,j}) \tag{4.66}$$

is obtained.

In total, by noting that the third case actually extends the second one from $\ell-1$ to ℓ, the claimed form is recovered where the normalization constant is $\frac{1}{1-\kappa_{i,\ell}}$ with

$$\kappa_{i,\ell} \;=\; 1 \;-\; \sum_{j=1}^{\ell}\lambda_{i,\ell}(\{j\}) \;-\; \lambda_{i,\ell}(\{i\}) \;-\; \lambda_{i,\ell}(\{1,\ldots,\ell,i\}^{\mathsf{C}}) \tag{4.67}$$

which finalizes the proof. □

The last results yields a closed-form expression for all combinations $R_{i,i-1}$. Thereafter, their combination

$$\bigoplus_{i=2}^{k}\bigoplus_{j=1}^{i-1} m_{i,j} \;=\; \bigoplus_{i=2}^{k} R_{i,i-1} \tag{4.68}$$

is required to compute the resulting overall mass function. Even though the functional form of the individual $m_{i,j}$ does not allow the application of lemma 4.1, the combination $R_{i,i-1}$ is a valid instance for the selection. This enables to elegantly prove the following result as an implication of the proof of theorem 4.6.

Theorem 4.9. *Let $m_{i,j}$ be the two-vs-all decomposition's mass functions as given by (4.56) such that $0 < \mathsf{w}_{i,j} < 1$ hold for all $i,j = 1,\ldots,k$ with $i \neq j$ and $R_{i,i-1}$ be their combination as in lemma 4.8. Further, let*

$$\lambda_i(A) = \begin{cases} \prod\limits_{\substack{s=1\\ s\neq j}}^{i-1}\prod\limits_{\substack{t=s+1\\ t\neq j}}^{i}(1-\mathsf{w}_{s,t})\prod\limits_{\substack{s=1\\ s\neq j}}^{i}\mathsf{w}_{i,s} & \text{if } A=\{j\},\ j\leq i \\ \prod\limits_{s=1}^{i-1}\prod\limits_{t=s+1}^{i}(1-\mathsf{w}_{s,t}) & \text{if } A=\{1,\ldots,i\}^{\mathsf{C}}\neq\emptyset \\ 0 & \text{otherwise} \end{cases}$$

be an unnormalized mass function. Then, defining $R_{2,1} \coloneqq m_{2,1}$, the combination $\bigoplus_{j=2}^{i} R_{j,j-1}$ is well defined for each $3 \leq i \leq k$ and satisfies

$$\left(\bigoplus_{i=2}^{k}\bigoplus_{j=1}^{i-1} m_{i,j}\right)(A) \;=\; \left(\bigoplus_{j=2}^{i} R_{j,j-1}\right)(A) \;=\; \frac{1}{1-\kappa_i}\cdot\lambda_i(A) \tag{4.69}$$

where for the conflict

$$\kappa_i \;=\; 1 \;-\; \sum_{j=1}^{i}\lambda_i(\{j\}) \;-\; \lambda_i(\{1,\ldots,i\}^{\mathsf{C}}) \tag{4.70}$$

holds.

Proof. For $i = 3$, it holds $R_{3,2} \oplus R_{2,1} = R_{3,2} \oplus m_{2,1}$. Here, assuming that $B \subseteq \Omega$ with $R_{3,2}(B) > 0$ and $C \subseteq \Omega$ with $m_{1,2}(C) > 0$ are given implies that $B = \{j\}$ with $1 \leq j \leq 3$ or $B = \{1,2,3\}^{\mathsf{C}}$ as well as $C = \{1,2\}$ or $C = \{1,2\}^{\mathsf{C}}$ has to hold. The claimed form of the resulting combination is obtained from similar combinatorial

reasoning as in the proof of lemma 4.8. Thereafter, combining $R_{i,i-1}$ with $\bigoplus_{j=1}^{i-1} R_{j,j-1}$ can be performed using lemma 4.1, but additionally is completely analogous to the proof of theorem 4.6. □

By simply using the case of $i = k$ in the last result, the following closed-form expression for the overall combination is obtained. Again, it defines a Bayesian mass function that is equivalent to the one obtained in corollary (4.7) by removing all pairwise probabilities $\phi_{i,j}$. Thus applying the same simplifications as in (4.55), the combination can computed using $\mathcal{O}(k^2)$ operations in practical applications.

Corollary 4.10. *Let $m_{i,j}$ be the two-vs-all decomposition's mass functions as given by (4.56) such that $0 < \mathsf{w}_{i,j} < 1$ hold for all $i, j = 1, \ldots, k$ with $i \neq j$. Their overall combination is a well defined Bayesian mass function and satisfies*

$$\left(\bigoplus_{i=1}^{k-1} \bigoplus_{j=i+1}^{k} m_{i,j}\right)(A) = \begin{cases} \frac{p_i}{\sum_{j=1}^{k} p_j} & \text{if } A = \{i\},\ 1 \leq i \leq k \\ 0 & \text{otherwise} \end{cases} \tag{4.71}$$

such that for the unnormalized probabilities

$$p_i \;=\; \prod_{\substack{j=1 \\ j \neq i}}^{k} \mathsf{w}_{i,j} \;\cdot\; \prod_{\substack{s=1 \\ s \neq i}}^{k-1} \prod_{\substack{t=s+1 \\ t \neq i}}^{k} (1 - \mathsf{w}_{s,t}) \tag{4.72}$$

holds.

Besides creating a two-vs-all decomposition, evidence theory allows the creation of other reduction approaches. For example, a three-vs-all decomposition can be created by separating each triple of classes from the set of all other ones and in general, for each $\ell \leq \frac{k}{2}$ the ℓ-vs-all decomposition can be created by separating each possible subset of ℓ classes from all other $k - \ell$ ones. However, from

$$\binom{k}{\ell} \;=\; \frac{k!}{\ell! \cdot (k-\ell)!} \;=\; \frac{1}{\ell!} \cdot \prod_{i=1}^{\ell} (k - i + 1) \;\in\; \Theta(k^\ell) \tag{4.73}$$

follows that there are $\Theta(k^\ell)$ individual classifiers in the respective ℓ-vs-all decomposition. This can quickly become too large even for $\ell = 3$ or $\ell = 4$ but still, it is an interesting question to analyze the respective combination. Therefore, let $\tau(i)$ be the corresponding i-th subset containing exactly ℓ out of k classes, i.e. $1 \leq i \leq \binom{k}{\ell}$. Using a classifier for the obtained classification problem that each predicts a calibrated binary membership probability $q_i \in (0, 1)$, a consistent evidence-theoretic modeling is to create the mass functions

$$m_i(A) \;\coloneqq\; \begin{cases} q_i & \text{if } A = \tau(i) \\ 1 - q_i & \text{if } A = \tau(i)^{\mathsf{C}} \\ 0 & \text{otherwise} \end{cases} \tag{4.74}$$

for all sets $1 \leq i \leq \binom{k}{\ell}$. Thereafter, the remaining task is to compute the overall combination and derive the necessary assumptions such that there is no full conflict. Here, all previous combination results give rise to the following, unproven statement:

Conjecture 4.11. *Let m_i be the ℓ-vs-all decomposition's mass functions as in* (4.74)*. Then, their combination exists and forms a Bayesian mass function*

$$\left(\bigoplus_{i=1}^{\binom{k}{\ell}} m_i\right)(A) = \begin{cases} \frac{p_i}{\sum_{j=1}^{k} p_j} & \text{if } A = \{i\},\ 1 \leq i \leq k \\ 0 & \text{otherwise} \end{cases} \tag{4.75}$$

such that for the unnormalized class probabilities, $i = 1, \ldots, k$, holds:

$$p_i \ = \ \prod_{j=1}^{\binom{k}{\ell}} Pl_{m_j}(\{i\}) \tag{4.76}$$

A proof is not obvious but still, a reasonable generalization of the results proven in the previous sections. It should be emphasized that the plausibilities simplify to

$$Pl_{m_j}(\{i\}) \ = \ \begin{cases} q_j & \text{if } i \in \tau(j) \\ 1 - q_j & \text{if } i \notin \tau(j) \end{cases} \tag{4.77}$$

and a general proof is mainly interesting from the theoretical point of view, but presumably even for $\ell = 3$ or $\ell = 4$ of less practical relevance.

Besides this, the evidence-theoretic modelings generally allow the creation and combination of *incomplete* ensembles. On the one hand, this is advantageous to keep the computational complexity feasible, but, on the other hand, it remains highly unclear how to decide which classifiers should be kept in the ensemble and which not (and might even not be trained at all). Still, this general possibility of evidence theory should be emphasized.

4.3 Dynamic Classification using Evidence Theory

The previous results share the common property of combining decomposition-based classification with classifier calibration such that multiple probabilistic predictions were used to model mass functions, which thereafter were combined by iteratively applying Dempster's rule of combination (4.9). Due to the selected modelings, the respective overall mass function turned out to be Bayesian, therefore justifies the interpretation as an estimate of the posterior probabilities $P(\mathsf{y} \mid \mathsf{x})$.

The only involved assumption under which the closed-form expressions for the resulting posterior probability estimates were derived are non-binary probabilities, i.e. not degenerated to zero or one, such that no involved product becomes zero. Even though it is at least arguable whether a data-based model is sufficiently competent to predict a probability of zero, it might still be reasonable in certain applications.

Besides this, another important aspect is to extend the prediction from a classical into a *dynamic* context. This means that there is a dynamically changing set of classes $\emptyset \neq \mathcal{M} \subseteq \{1, \ldots, k\}$ such that the prediction is constrained to $\mathcal{M}$. A particular strategy is to compute a posterior probability estimate that is zero outside of $\mathcal{M}$. This information can be consistently represented using a mass function

$$m_{\mathcal{M}}(A) \ \coloneqq \ \begin{cases} 1 & \text{if } A = \mathcal{M} \\ 0 & \text{if } A \neq \mathcal{M} \end{cases} \tag{4.78}$$

such that in an evidence-theoretic context, the remaining task is to combine the overall mass function m, which could be obtained with any of aforementioned approaches,

with $m_{\mathcal{M}}$. Computing the combination is relatively straightforward using the following properties. Thereafter, the case of degenerated probabilities will be recovered as a special case such that the presented results allow both, an extension to a dynamic classification context as well as an integration of degenerated probabilistic predictions.

Proposition 4.12. *Let m be an arbitrary mass function on $\Omega = \{1, \dots, k\}$. Further, let $\emptyset \neq \mathcal{M} \subseteq \Omega$ and $m_{\mathcal{M}}$ as given by (4.78). Then, the following properties hold:*

1. *The combination $m \oplus m_{\mathcal{M}}$ is well defined if and only if there is an $A \subseteq \mathcal{M}$ such that $m(A) > 0$.*

2. *If the combination is well defined, it holds:*

$$(m \oplus m_{\mathcal{M}})(A) \;=\; \frac{1}{\sum\limits_{B \subseteq \mathcal{M}} m(B)} \cdot \sum_{B : B \cap \mathcal{M} = A} m(B) \tag{4.79}$$

3. *If two different focal sets of m are always disjoint, i.e. for $A, B \subseteq \Omega$ with $A \neq B$ and $m(A) > 0$ as well as $m(B) > 0$ always holds $A \cap B = \emptyset$, the combination simplifies to:*

$$(m \oplus m_{\mathcal{M}})(A) \;=\; \frac{m(A \cap \mathcal{M})}{\sum\limits_{B \subseteq \mathcal{M}} m(B)} \tag{4.80}$$

Proof. The combination is well defined if and only if $\kappa(m, m_{\mathcal{M}}) < 1$ holds. Using definition (4.8) yields

$$\kappa(m, m_{\mathcal{M}}) \;=\; \sum_{\substack{B, C \subseteq \Omega \\ B \cap C = \emptyset}} m(B) \cdot m_{\mathcal{M}}(C) \;=\; \sum_{B \subseteq \mathcal{M}^{\mathsf{C}}} m(B) \;=\; 1 - \sum_{B \subseteq \mathcal{M}} m(B) \tag{4.81}$$

which is < 1 if and only if there exists an $A \subseteq \mathcal{M}$ with $m(A) > 0$ as claimed. Next, simplifying the resulting combination using (4.8) yields

$$\begin{aligned} (m \oplus m_{\mathcal{M}})(A) \;&=\; \frac{1}{1 - \kappa(m, m_{\mathcal{M}})} \cdot \sum_{\substack{B, C \subseteq \Omega \\ B \cap C = A}} m(B) \cdot m_{\mathcal{M}}(C) \\ &=\; \frac{1}{\sum\limits_{B \subseteq \mathcal{M}} m(B)} \cdot \sum_{B : B \cap \mathcal{M} = A} m(B) \end{aligned} \tag{4.82}$$

as claimed. Furthermore for given A, the existence of $B \neq A$ with $A = B \cap \mathcal{M}$ and $m(B) > 0$ directly implies the existence of two non-disjoint focal sets. □

With respect to a Bayesian mass function with focal elements $\{i\}$ and respective probabilities p_i – for example those computed using the methods from section 4.2 – the last result yields an integration of dynamic class information: A class probability p_i for class $i \in \mathcal{M}$ will be maintained because $\{i\} \cap \mathcal{M} = \{i\}$. For $i \notin \mathcal{M}$, the class probability becomes zero from $\{i\} \cap \mathcal{M} = \emptyset$. Finally, all remaining class probabilities are normalized as the normalization constant simplifies to the sum of all remaining probabilities. Even though the approach recovers a straightforward selection and renormalization, it should be emphasized that it is still additionally theoretically justified by evidence theory.

Besides integrating dynamic classification against a formal background, a similar approach allows the extension of the presented results for extreme probabilities of zero whose exclusion is the only assumption of any previous result. As soon as probabilities

become zero in the presented modelings, they induce sets with mass and plausibility both equal to zero. Thus, this situation is in fact a special case of a mass function that yields zero plausibility for non-empty sets. Combining such a mass function with an arbitrary selection of other mass functions always yields zero plausibility:

Proposition 4.13. *Let $m_1, m_2, \ldots, m_\ell$ be mass functions such that their combination $m = \bigoplus_{i=1}^{\ell} m_i$ is well defined, i.e. the overall conflict satisfies $\kappa < 1$. If there is an index $i \in \{1, \ldots, \ell\}$ and a set $M \subseteq \Omega$ with $Pl_{m_i}(M) = 0$, also $Pl_m(M) = 0$ holds.*

Proof. Since the combination rule is commutative, assume without loss of generality that $i = 1$ holds. By definition of the plausibility,

$$Pl_{m_1}(M) \;=\; \sum_{\substack{B \subseteq \Omega \\ B \cap M \neq \emptyset}} m_1(B) \;=\; 0 \tag{4.83}$$

implies $m_1(B) = 0$ for all $B \subseteq \Omega$ satisfying $B \cap M \neq \emptyset$. Thus, the plausibility $Pl_m(M)$ satisfies

$$Pl_m(M) \;=\; \sum_{\substack{B \subseteq \Omega \\ B \cap M \neq \emptyset}} m(B) \;=\; \sum_{\substack{B \subseteq \Omega \\ B \cap M \neq \emptyset}} \sum_{\substack{C, D \subseteq \Omega \\ C \cap D = B}} \frac{1}{1-\kappa} \cdot m_1(C) \cdot \left(\bigoplus_{i=2}^{\ell} m_i \right)(D) = 0 \tag{4.84}$$

as $m_1(C) = 0$ holds because $B = C \cap D \subseteq C$ implies $C \cap M \supseteq B \cap M \neq \emptyset$. □

In light of this, extreme probabilities simply yield mass functions with zero plausibilities for certain non-empty sets. Clearly, as long as there are no conflicting extreme probabilities, the combination still exists and also yields zero plausibility for any set where a respective individual mass function does. Here, the combination even yields zero plausibility for any union of the respective sets.

Proposition 4.14. *Let m be a mass function such that for $M_1, \ldots, M_\ell \subseteq \Omega$ holds $Pl(M_i) = 0$ for all $i = 1, \ldots, \ell$. Then, also $M \coloneqq \bigcup_{i=1}^{\ell} M_i$ satisfies $Pl(M) = 0$.*

Proof. It holds

$$Pl(M) \;=\; \sum_{\substack{B \subseteq \Omega \\ B \cap M \neq \emptyset}} m(B) \;\leq\; \sum_{i=1}^{\ell} \sum_{\substack{B \subseteq \Omega \\ B \cap M_i \neq \emptyset}} m(B) \;=\; \sum_{i=1}^{\ell} Pl(M_i) \;=\; 0 \tag{4.85}$$

which implies $Pl(M) = 0$ from the fact that mass functions are non-negative. □

Based on the last two results, extreme probabilities occurring at arbitrary individual classifiers in total only yield a single set with plausibility zero. Thus with application to classification, extreme probabilities only restrict the prediction into a dynamic target set $\mathcal{M}$, which can be integrated in the combination using (4.80) as long as $\mathcal{M} \neq \emptyset$ holds.

In particular, the dynamic class set $\mathcal{M}$ becomes empty if and only if each class yields plausibility zero under at least one individual mass function. This occurs if and only if the mass functions cannot be combined. Hence, all previous results are generally valid for arbitrary probabilities, the only remaining assumption is that the combination still exists. This can be verified by simply computing the unnormalized combination. As long as a normalization is possible, i.e. it is positive for at least a single set of classes, the combination exists.

4.4 Summary

This chapter applied evidence theory for decomposition-based classification, particular focusing on the one-vs-all and one-vs-one decompositions. Here, two alternative combination strategies are obtained that both yield a Bayesian mass function for each of the reductions, thus each overall combination result allows a classic probabilistic interpretation.

With respect to the one-vs-all reduction, a class prediction equivalent to the classical maximum probability is recovered. Particular relevant are the presented results with respect to the non-competence problem as in case of the one-vs-one decomposition, an evidence-theoretic modeling was only possible if each prediction is combined with a weighting of the classifier that models its competence.

Thereafter, evidence theory led a systematic approach to dynamic classification that yields zero plausibility (or probability in a Bayesian context) for all currently impossible classes. This is an intuitive consequence, but interestingly obtained from a complex formalism, i.e. modeling the calibrated predictions as basic mass functions and thereafter applying Dempster's rule of combination.

In light of this, the next chapter integrates the obtained results into existing approaches to decomposition-based classification, yielding a generalization of classical pairwise coupling to tackle the non-competence problem and to integrate dynamic class information as well.

Chapter 5

Generalized Pairwise Coupling

The last chapter was focused on evidence theory and decomposition-based classification in general. Still, it is particularly interesting to analyze the results with respect to the one-vs-one decomposition as comparing them to existing pairwise coupling techniques will allow the derivation of a systematic approach to generalize them with respect to dynamic classification and the non-competence problem.

In particular, an evidence-theoretic modeling was only possible by combining the pairwise classifier differentiating between the classes i and j with a weighting $\mathsf{w}_{i,j}$ such that a prediction different to i or j is possible, but has estimated probability $1 - \mathsf{w}_{i,j}$ in a Bayesian context. One the reverse only restricting to the pairwise probabilities $\phi_{i,j}(\mathsf{x})$, the predictions could be reasonably represented as mass functions in the form of (4.33), however a combination is impossible. Therefore, the influence of a weight that controls the competence deserves particular interest to systematically address the non-competence problem in arbitrary pairwise coupling techniques.

Here, section 5.1 will interpret the existing variants presented in full detail in chapter 2 as *constant* pairwise coupling approaches as each individual classifier receives the same weight during the coupling. This also leads towards a Bayesian interpretation of the non-competence problem in section 5.2, recovering the Bayesian counterpart of the full-conflict situation using evidence theory observed in the previous chapter. Thereafter, non-constant generalizations are presented in section 5.3 that extend constant pairwise coupling using arbitrary weights. As will be presented in advance in section 5.4, dynamic class information can simply be modeled by weighting a selection of classifiers with zero. In sections 5.3 and 5.4, the focus lies on the most commonly used approaches that were presented in subsection 2.3.1 in chapter 2. Still, similar extensions most likely are possible for other pairwise coupling techniques as well. Finally, section 5.5 presents new methods to compute the required weights $\mathsf{w}_{i,j}$ as well as discusses how these extended algorithms can be applied in combination with large-scale models like deep neural networks where a classical application of reductions requires to train and deploy an impractically large number of individual models.

5.1 Constant Pairwise Coupling

Formally, combining the one-vs-one reduction's predictions using evidence theory as in subsection 4.2.2 is not a pairwise coupling technique, as it depends not only on the predictions $\phi_{i,j}(\mathsf{x})$ but also on the weights $\mathsf{w}_{i,j}(\mathsf{x})$. Still, the overall combination yields the Bayesian mass function and therefore posterior probabilities estimates given by (4.52). Here, a remarkable result is obtained if the pairwise weights are assumed as constant $\mathsf{w}_{i,j} \equiv \mathsf{w}_0$ with the only restriction that $\mathsf{w}_0 > 0$ holds. Substituting this into

(4.52) yields the posterior probability estimates

$$p_i(\mathsf{x}) \;\propto\; \prod_{\substack{j=1\\ j\neq i}}^{k} \phi_{i,j}(\mathsf{x}) \tag{5.1}$$

for each class $i = 1, \ldots, k$ because all other factors are independent of i (i.e. are contained in the proportionality factor) and therefore cancel out in the remaining normalization.

This has two important consequences: First, evidence theory using constant weights can be used to construct a new pairwise coupling technique. Second, analogously to combining the evidence-theoretic modeling with both, constant and non-constant weights, it is reasonable to interpret pairwise coupling as a special case of a generalized approach that combines the pairwise predictions with a weighting. The latter enables to systematically address the non-competence problem in any pairwise coupling approach as long as it can be extended to integrate non-constant weights.

Here, combining each individual pairwise prediction $\phi_{i,j}(\mathsf{x})$ with a constant weight $\mathsf{w}_0 \equiv \mathsf{w}_{i,j}(\mathsf{x})$ is a reasonable first step. Motivated by (2.30) and the respective analysis, it is reasonable to multiplicatively combine the pairwise prediction $\phi_{i,j}(\mathsf{x})$ and the weight. With respect to probabilistic voting (2.24), the weighted formulation is obtained as

$$p_i^{\mathsf{Vote}}(\mathsf{x}) \;=\; \frac{2}{k\cdot(k-1)\cdot \mathsf{w}_0} \cdot \sum_{\substack{j=1\\ j\neq i}}^{k} \phi_{i,j}(\mathsf{x}) \cdot \mathsf{w}_0 \tag{5.2}$$

where the weighting reduces the pairwise sum from 1 to $\mathsf{w}_0 = \mathsf{w}_0 \cdot (\phi_{i,j}(\mathsf{x}) + \phi_{j,i}(\mathsf{x}))$ such that the sum of all probabilities changes from $\binom{k}{2}$ to $\binom{k}{2} \cdot \mathsf{w}_0$. Hence, the multiplication by w_0 simply cancels out during the normalization.

Integrating the constant weight into the non-dominance approach (2.25) requires to equivalently reformulate the non-dominance vector. Originally, it is constructed without weights for each component $1 \leq i \leq k$ as

$$\begin{aligned} 1 \;-\; \max_{j\neq i} \phi'_{j,i}(\mathsf{x}) \;&=\; 1 \;-\; \max_{j\neq i} \max(\phi_{j,i}(\mathsf{x}) - \phi_{i,j}(\mathsf{x}), 0) \\ &=\; \phi_{i,j}(\mathsf{x}) \;+\; \phi_{j,i}(\mathsf{x}) \;+\; \min_{j\neq i} \min(\phi_{i,j}(\mathsf{x}) - \phi_{j,i}(\mathsf{x}), 0) \\ &=\; \min(\phi_{i,j}(\mathsf{x}) + \phi_{j,i}(\mathsf{x}) + \phi_{i,j}(\mathsf{x}) - \phi_{j,i}(\mathsf{x}),\; \phi_{i,j}(\mathsf{x}) + \phi_{j,i}(\mathsf{x})) \\ &=\; \min(2\cdot\phi_{i,j}(\mathsf{x}),\; \phi_{i,j}(\mathsf{x}) + \phi_{j,i}(\mathsf{x})) \end{aligned} \tag{5.3}$$

such that for the non-dominance vector equivalently

$$\mathsf{ND}_i(\mathsf{x}) \;=\; 1 - \max_{j\neq i} \phi'_{j,i}(\mathsf{x}) \;=\; \min_{j\neq i} \min(2\cdot\phi_{i,j}(\mathsf{x}),\; \phi_{i,j}(\mathsf{x}) + \phi_{j,i}(\mathsf{x})) \tag{5.4}$$

holds. Since the probabilities $p^{\mathsf{ND}}(\mathsf{x})$ are obtained by normalizing the non-dominance vector, expressing the latter using the last equality shows that a multiplication $\phi_{i,j}\cdot\mathsf{w}_0$ simply results in a non-dominance vector $\mathsf{w}_0 \cdot \mathsf{ND}(\mathsf{x})$. Therefore, the weighting cancels out in the following normalization step, similar to the probabilistic voting.

The remaining pairwise coupling approach used to compute the probabilities $p^{\mathsf{WLW}}(\mathsf{x})$ solved the optimization problem (2.26). For a constant scaling, the objective

function can be rewritten as

$$\begin{aligned} &\min_p \sum_{i=1}^{k} \sum_{\substack{j=1 \\ j\neq i}}^{k} (\mathsf{w}_0 \cdot \phi_{j,i}(\mathsf{x}) \cdot p_i \;-\; \mathsf{w}_0 \cdot \phi_{i,j}(\mathsf{x}) \cdot p_j)^2 \\ = \mathsf{w}_0^2 \cdot &\min_p \sum_{i=1}^{k} \sum_{\substack{j=1 \\ j\neq i}}^{k} (\phi_{j,i}(\mathsf{x}) \cdot p_i \;-\; \phi_{i,j}(\mathsf{x}) \cdot p_j)^2 \end{aligned} \tag{5.5}$$

such that the constant w_0 is independent of the optimization and shows that for each respective value, the solution exists under the same assumptions.

Thus, each of the most relevant pairwise coupling techniques can be interpreted as a special case of an algorithm using both, the pairwise probabilities and the weights. Similar extensions most likely exist for other but less commonly used pairwise coupling techniques as well, presumably the derivations are analogous to the presented ones.

5.2 Bayesian Interpretation

Before generalizing arbitrary pairwise coupling techniques for non-constant weights $\mathsf{w}_{i,j}(\mathsf{x})$, it is interesting to analyze pairwise coupling from a classical Bayesian probabilistic point of view. Even though there are theoretical justifications for the existing techniques, these are based on relationships between the true but unknown probabilities. As a particular example,

$$P(\mathsf{y} = i \mid \mathsf{x}) \cdot P(\mathsf{y} = j \mid \mathsf{x}, \mathsf{y} \in \{i,j\}) \;=\; P(\mathsf{y} = j \mid \mathsf{x}) \cdot P(\mathsf{y} = i \mid \mathsf{x}, \mathsf{y} \in \{i,j\}) \tag{5.6}$$

holds for all $1 \leq i, j \leq k$ satisfying $P(\mathsf{y} \in \{i,j\} \mid \mathsf{x}) > 0$.

Even though in practice none of the involved quantities is known, it still justifies to replace $P(\mathsf{y} = i \mid \mathsf{x}, \mathsf{y} \in \{i,j\})$ with $\phi_{i,j}(\mathsf{x})$ such that an overconstrained system of $\binom{k}{2}$ equations with k unknown posterior class estimates is obtained. Finding a solution that minimizes the squared differences is an alternative to derive problem (2.26). Similar justifications exist for other approaches as well, which are discussed in full detail in the respective aforementioned works. Still, all of them are based on replacing the true but unknown pairwise probabilities $P(\mathsf{y} = i \mid \mathsf{x}, \mathsf{y} \in \{i,j\})$ with $\phi_{i,j}(\mathsf{x})$.

However from a strictly formal point of view, this justification becomes slightly problematic. For an arbitrary pair of classes $\{i,j\}$ such that $P(\mathsf{y} \in \{i,j\} \mid \mathsf{x}) > 0$ holds, the conditional probability $P(\mathsf{y} = i \mid \mathsf{x}, \mathsf{y} \in \{i,j\})$ formally exists. However as soon as *multiple* conditional probabilities are *combined*, they have to be multiplied with the probability of the event under which they exist. With respect to pairwise coupling, this means that $P(\mathsf{y} = i \mid \mathsf{x}, \mathsf{y} \in \{i,j\})$ needs to be combined with $P(\mathsf{y} \in \{i,j\} \mid \mathsf{x})$. Omitting the latter at first can be interpreted as replacing it with probability one, i.e. assume that $\mathsf{y} \in \{i,j\}$ holds. Simultaneously doing so for all pairs results in the paradox situation that contradicting events are assumed as given. This is the Bayesian counterpart of evidence-theoretic modeling using the mass functions (4.35), which resulted in full conflict.

Based on the scaling invariance discussed in section 5.1, omitting probabilities can also be interpreted as *constant* replacement of $P(\mathsf{y} \in \{i,j\} \mid \mathsf{x})$. This can similarly be interpreted as a uniform prior over all classes, analogously to replacing the weights $\mathsf{w}_{i,j}$ with constants in (4.53). For this reason, the non-competence problem becomes

in fact a *constant* competence problem instead as in practice, the pairwise posterior probabilities are highly unlikely to be constant. This is empirically supported by aforementioned significant improvements that are obtained by the application of correcting classifiers in probabilistic voting. Therefore, the following part generalizes pairwise coupling using non-constant weights.

5.3 Non-Uniform Generalization

Based on the previous insights, the competence of the individual one-vs-one classifiers can systematically be addressed by extending pairwise coupling with a corresponding weight. This means that during the actual coupling process, the pairwise predictions $\phi_{i,j}(\mathsf{x})$ are always *combined* with the respective weight $\mathsf{w}_{i,j}(\mathsf{x})$. Thus, there is a straightforward demand for reasonable weight estimators.

From the theoretical point of view, the ideal weight would be the pairwise posterior probability $P(\mathsf{y} \in \{i,j\} \mid \mathsf{x})$. Still, this does not help in practice because estimating all pairwise posterior probabilities $P(\mathsf{y} \in \{i,j\} \mid \mathsf{x})$ is equivalently complex as directly estimating $P(\mathsf{y} = i \mid \mathsf{x})$, similar to (2.30) and the related discussion. Therefore, it is an interesting and relevant question to do both, compute non-constant weights and integrate them into arbitrary pairwise coupling techniques. Here, theoretical justifications are particularly relevant.

With respect to one-vs-all decomposition-based classifiers, there are k individual discriminant functions f_i, $i = 1, \ldots, k$, such that an observed instance x is assigned to the class whose associated discriminant function returns the largest value. Presumably the most commonly used method to transform them into a posterior probability estimation is to use the softmax function (2.22). The latter assumes that the sign is used in the binary decisions, therefore analyzing its application with probabilistic classifiers $\mathsf{f}_i : \mathcal{X} \to [0,1]$ (e.g. obtained from an explicit calibration step in each binary classification problem) allows the derivation of an interesting insight. First, doing so should be combined with a subtraction of 0.5 to shift the decision threshold to zero accordingly. Applying the softmax transformation thereafter means to first apply the exponential function on each component and to normalize the resulting vector. Because the inputs are probabilities, a first-order taylor series of $\exp(z-0.5)$ in $z_0 = 0.5$ simplifies to

$$\exp(z-0.5) \;=\; \exp(0) \;+\exp(0)\cdot(z-1) \;+\; \varepsilon \;=\; z \;+\; \varepsilon \tag{5.7}$$

such that on $(0,1)$, it holds $\exp(z-0.5) \approx z$ with an approximation error $\varepsilon \in \mathcal{O}(z^2)$. Thus, applying the softmax function this way on probabilistic functions is similar to a simple normalization. As a matter of fact, reasonably approximating the posterior probabilities this way assumes an approximate proportional relationship between membership and unknown posterior probabilities, $\mathsf{f}_i(\mathsf{x}) = \alpha \cdot P(\mathsf{y} = i \mid \mathsf{x})$ for all $i = 1, \ldots, k$ where $\alpha > 0$ is the respective proportionality constant. Equivalently,

$$\frac{\mathsf{f}_i(\mathsf{x})}{\mathsf{f}_j(\mathsf{x})} \;=\; \frac{P(\mathsf{y} = i \mid \mathsf{x})}{P(\mathsf{y} = j \mid \mathsf{x})} \tag{5.8}$$

holds approximately for all functions i and j as long as the denominators are non-zero. It should be emphasized that the latter only holds because the proportionality constant α does not depend on i. Thus, additionally

$$P(\mathsf{y} \in \{i,j\} \mid \mathsf{x}) \;=\; P(\mathsf{y} = i \mid \mathsf{x}) + P(\mathsf{y} = j \mid \mathsf{x}) \;=\; \alpha \cdot (\mathsf{f}_i(\mathsf{x}) + \mathsf{f}_j(\mathsf{x})) \tag{5.9}$$

holds approximately such that a reasonable modification is to replace the individual predictions $\mathsf{f}_i(\mathsf{x})$ and $\mathsf{f}_j(\mathsf{x})$ with a *combined* estimate while approximately preserving the proportional relationship to the unknown posterior probabilities, yielding

$$P(\mathsf{y} \in \{i,j\} \mid \mathsf{x}) \;=\; \beta \cdot \mathsf{w}_{i,j}(\mathsf{x}) \tag{5.10}$$

for each pair $1 \leq i,j \leq k$ of classes as an assumption that holds at least approximately.

5.3.1 Generalized Pairwise Coupling

Based on the previous analysis, the product $P(\mathsf{y}=i \mid \mathsf{y} \in \{i,j\},\mathsf{x}) \cdot P(\mathsf{y} \in \{i,j\} \mid \mathsf{x})$ of the unknown probabilities is approximately proportional to multiplying the pairwise prediction $\phi_{i,j}(\mathsf{x})$ by the weights $\mathsf{w}_{i,j}(\mathsf{x})$

$$P(\mathsf{y}=i \mid \mathsf{y} \in \{i,j\},\mathsf{x}) \cdot P(\mathsf{y} \in \{i,j\} \mid \mathsf{x}) \;=\; \beta \cdot \phi_{i,j}(\mathsf{x}) \cdot \mathsf{w}_{i,j}(\mathsf{x}) \tag{5.11}$$

such that *generalized pairwise coupling* can be formulated as the following task: Given the inputs

- instance $\mathsf{x} \in \mathcal{X}$ and an integer k such that $\mathcal{Y} = \{1,\ldots,k\}$ holds
- pairwise probabilities matrix $\phi_{i,j}(\mathsf{x})$ and $\phi_{j,i}(\mathsf{x}) = 1-\phi_{i,j}(\mathsf{x})$ for all $1 \leq i < j \leq k$
- pairwise weight matrix $\mathsf{w}_{i,j}(\mathsf{x}) = \mathsf{w}_{j,i}(\mathsf{x})$ for all $1 \leq i,j \leq k$ with $i \neq j$

compute a posterior probability estimation $p(\mathsf{x}) = (p_1(\mathsf{x}),\ldots,p_k(\mathsf{x}))$. Based on the insights of section 5.1, the existing case of pairwise coupling is recovered by using a constant weight matrix $\mathsf{w}_{i,j} = \mathsf{w}_0 > 0$ for all pairs (i,j). As rescaling the weight matrix does not change the solution, $\mathsf{w}_{i,j} \leq 1$ can be assumed without loss of generality for all pairs (i,j).

Algorithmic Solutions

Generalized pairwise coupling at first can be solved using evidence theory as presented in section 4.2, the result will be given by (4.52). Still, it is particularly interesting to also extend the existing pairwise coupling techniques for this task. With respect to probabilistic voting and the non-dominance criterion, this is relatively straightforward based on the insights of section 5.1. In particular, only the constant weight w_0 has to be replaced by $\mathsf{w}_{i,j}(\mathsf{x})$ in (5.2) and (5.3), yielding *generalized probabilistic voting*

$$p_i^{\mathsf{GVote}}(\mathsf{x}) \;=\; \frac{1}{\sum\limits_{\ell=1}^{k-1} \sum\limits_{j=\ell+1}^{k} \mathsf{w}_{i,j}(\mathsf{x})} \cdot \sum_{\substack{j=1 \\ j \neq i}}^{k} \phi_{i,j}(\mathsf{x}) \cdot \mathsf{w}_{i,j}(\mathsf{x}) \tag{5.12}$$

as well as the *generalized non-dominance criterion*

$$\begin{aligned} \mathsf{GND}_i(\mathsf{x}) \;&=\; \min_{j \neq i} \; \min(2 \cdot \mathsf{w}_{i,j}(\mathsf{x}) \cdot \phi_{i,j}(\mathsf{x}),\; \mathsf{w}_{i,j}(\mathsf{x}) \cdot (\phi_{i,j}(\mathsf{x}) + \phi_{j,i}(\mathsf{x}))) \\ &=\; \min_{j \neq i} \; \mathsf{w}_{i,j}(\mathsf{x}) \cdot \min(2 \cdot \phi_{i,j}(\mathsf{x}),\; \phi_{i,j}(\mathsf{x}) + \phi_{j,i}(\mathsf{x})) \\ &=\; \min_{j \neq i} \; \mathsf{w}_{i,j}(\mathsf{x}) \cdot \min(2 \cdot \phi_{i,j}(\mathsf{x}),\; 1) \end{aligned} \tag{5.13}$$

with probabilities $p_i^{\mathsf{GND}}(\mathsf{x})$ obtained by normalizing $\mathsf{GND}(\mathsf{x})$, respectively. Here, the former coincides with the proposed approach first introducing the correcting classifiers

[Moreira & Mayoraz 1998] if the weights are computed using the latter. Still besides an extended theoretical justification, other options to compute the weights are possible as well – these will be discussed in section 5.5. Generalizing the third pairwise coupling approach given by (2.26) yields

$$\min_{p} \sum_{i=1}^{k} \sum_{\substack{j=1 \\ j \neq i}}^{k} [\mathsf{w}_{i,j}(\mathsf{x}) \cdot (\phi_{j,i}(\mathsf{x}) \cdot p_i \; - \; \phi_{i,j}(\mathsf{x}) \cdot p_j)]^2 \quad \text{s.t.} \; \sum_{i=1}^{k} p_i = 1 \tag{5.14}$$

whose solution is used as an estimate of $P(\mathsf{y} \mid \mathsf{x})$. However, there are some important aspects to consider. Most importantly, the existence of a solution for arbitrary weights $\mathsf{w}_{i,j}$ has to be proven. If this holds, it is important to analyze if the solution is still uniquely defined and satisfies $p_i \geq 0$ for all $i = 1, \ldots, k$ such that the interpretation as a posterior probability estimate remains valid. For this, the prediction instance x can be assumed as fixed such that the explicit dependency of all computed probabilities on it is omitted in the following part, similar to chapter 4. Here at first, the following generalized result holds:

Lemma 5.1. *If $\phi_{i,j} > 0$ as well as $\mathsf{w}_{i,j} > 0$ hold for all $1 \leq i, j \leq k$ with $i \neq j$, problem* (5.14) *has a uniquely defined solution $p^{\mathsf{GWLW}}(\mathsf{x})$ satisfying $p_i^{\mathsf{GWLW}}(\mathsf{x}) \geq 0$ for all $i = 1, \ldots, k$.*

Proof. The proofs of these properties are generalizations of the constant-scaling counterparts, in particular theorems 2 and 3 in the original work [Wu et al. 2004]. Therefore using the property that $\mathsf{w}_{i,j} = \mathsf{w}_{j,i}$ holds for all pairs (i, j), the most elegant way is to define pairwise probability estimates $\psi_{i,j} \coloneqq \mathsf{w}_{i,j} \cdot \phi_{i,j}$ and replace all occurrences of $\phi_{i,j}$ with $\psi_{i,j}$ in the original proofs[1] for all pairs (i, j). From this, the only difference is that $\psi_{i,j} + \psi_{j,i} = \mathsf{w}_{i,j} \leq 1 = \phi_{i,j} + \phi_{j,i}$. However, both proofs in the original work do not depend on the restriction that $\phi_{i,j} + \phi_{j,i} = 1$ has to hold. As a matter of fact, the proofs remain valid without this restriction such that the claim is proven. □

The last result guarantees that generalizing (2.26) into (5.14) does not influence the existence of a solution satisfying all required properties. Still, it is important to efficiently compute it with respect to practical applications. The authors of the original work presented a direct algebraic way solving (2.26) as well as an efficient numeric iterative alternative. Consequently, extending these is relevant for solving (5.14). Generalizing the algebraic method, the matrix $Q \in \mathbb{R}^{k \times k}$ with entries

$$Q_{i,j} = \begin{cases} \sum\limits_{\substack{\ell=1 \\ \ell \neq i}}^{k} \mathsf{w}_{\ell,i}^2 \cdot \phi_{\ell,i}^2 & \text{if } i = j \\ -\mathsf{w}_{i,j}^2 \cdot \phi_{i,j} \cdot \phi_{j,i} & \text{if } i \neq j \end{cases} \tag{5.15}$$

has to be constructed. Now, p is optimal for (5.14) if and only if

$$\begin{bmatrix} Q & \mathbf{1} \\ \mathbf{1} & 0 \end{bmatrix} \cdot \begin{bmatrix} p \\ b \end{bmatrix} \; = \; \begin{bmatrix} 0 \\ 1 \end{bmatrix} \tag{5.16}$$

holds for a real-valued parameter b. In the original work, the matrix Q does only depend on the pairwise probabilities $\phi_{i,j}$. Still, the given derivation does not depend on the restriction that $\phi_{i,j} + \phi_{j,i} = 1$ holds for all pairs (i, j), analogously to the last

[1] It should be emphasized that the authors used the nomenclature of $r_{i,j}$ instead of $\phi_{i,j}$ for all pairwise probability estimates.

proof. Consequently, p is optimal for (5.14) if and only if it is a solution of (5.16). The latter equation system sized $(k+1)\times(k+1)$ can be solved with $\mathcal{O}(k^3)$ operations using direct algebraic methods.

As the optimal solution has to be computed during each prediction, the iterative numeric is usually preferred over the explicit algebraic solving based on the linear equation system. In particular using an arbitrary initial solution, the iterative computations converge to the optimal one. As neither the algorithm itself nor its convergence proof depends on the fact that the pairwise probabilities sum to one, the generalized variant given in algorithm 5.1 can be used to efficiently compute the solution of (5.14).

Algorithm 5.1: Iteratively compute p^{GWLW} in (5.14)

Result: Unique solution p of (5.16)
initialize $\ell = 1$ and p (e.g. $p_i = \frac{1}{k}$ for all $i = 1, \dots, k$)
while (5.16) *does not hold for* p *with error* $\leq \varepsilon$ **do**
  $p_\ell \coloneqq \frac{1}{Q_{\ell,\ell}} \cdot \left(p^\top Q p - \sum_{\substack{j=1 \\ j\neq\ell}}^{k} Q_{\ell,j} p_j \right)$
  normalize p
  $\ell \coloneqq (\ell \mod k) + 1$
end
return p

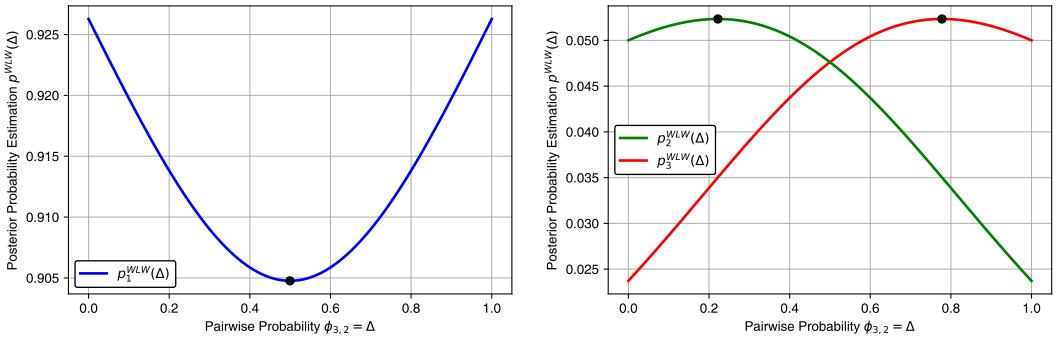
Here, several important properties from the original work are maintained. The optimal solution satisfies $(Qp)_i = -b$ for all components $i = 1, \dots, k$ from (5.16), such that $b = -p^\top Q p$ holds as well. The latter can be used to efficiently check for the termination criterion. Even though the involved computations require $\mathcal{O}(k^2)$ operations per iteration, they can be reduced to $\mathcal{O}(k)$ by caching Qp and performing sequential updates as in the original algorithm. Still, the authors suggest to perform a full update after k iterations to avoid accumulated numerical inaccuracies such that the average complexity per iteration remains in $\mathcal{O}(k)$.

Monotonicity Property in Pairwise Coupling

The previous analysis showed that the three most relevant pairwise coupling approaches can be extended for non-constant weights. Still, similar extensions most likely exist for any pairwise coupling technique as well. Next, there is another important, theoretic difference between the presented techniques.

Both, probabilistic voting and the non-dominance criterion, satisfy the following monotonicity property: For any fixed pair (i, j), the probabilities $\phi_{i,j}$ and $\phi_{j,i}$ can be replaced by $\phi_{i,j}+\Delta$ and $\phi_{i,j}-\Delta$, respectively, such that the confidence in the pairwise prediction moves a part $\Delta \geq 0$ from $\phi_{j,i}$ to $\phi_{i,j}$. For a given pairwise probabilities matrix Φ and an index pair (i, j) – such that all other probabilities remain fixed – the overall posterior probability estimate can be interpreted as a function of Δ. In case of probabilistic voting and the non-dominance criterion, this will always result in an overall increased probability p_i for class i and a decreased one p_j for class j. Consequently, both coupling approaches are *monotonic* in this sense. Analogously, the same holds for their generalized formulations assuming that the weight matrix remains unchanged.

In contrast to these properties, computing p^{WLW} and p^{GWLW} by solving (2.26) and (5.14), respectively, does not satisfy this monotonicity property, counterexamples



(a) $p_1^{\mathsf{WLW}}(\Delta)$ and its minimum value.

(b) $p_2^{\mathsf{WLW}}(\Delta)$ as well as $p_3^{\mathsf{WLW}}(\Delta)$ and their respective maximum values.

FIGURE 5.1: Resultung Posterior probability $p^{\mathsf{WLW}}(\Delta)$ and their extrema for the system induced by $\Phi(\Delta)$ as given in (5.17).

can already be constructed using three classes. For example, the following pairwise probabilities matrix depending on $\Delta \in [0, 1]$

$$\Phi(\Delta) \;=\; \begin{pmatrix} \bullet & 0.95 & 0.95 \\ 0.05 & \bullet & 1-\Delta \\ 0.05 & \Delta & \bullet \end{pmatrix} \tag{5.17}$$

can be defined such that the solution $p^{\mathsf{WLW}}(\Delta)$ can be interpreted as a function of Δ. Here, selecting $\Delta = 0.8$ and $\Delta = 0.9$ *increases* $\phi_{3,2}$ by 0.1 but simultaneously *decreases* the corresponding posterior probability estimation $p_3^{\mathsf{WLW}}(\Delta)$. This first can be seen by solving both systems, yielding the solutions $p_3^{\mathsf{WLW}}(0.8) = 0.05231$ as well as $p_3^{\mathsf{WLW}}(0.9) = 0.05157$, respectively, each rounded to five digits. Even though p_3^{WLW} decreases from increasing $\phi_{3,2}$, the overall deviation might still be caused from numerical inaccuracies as the absolute values are relatively small.

To exclude this possibility, an algebraic verification was performed. In particular using computer-algebraic methods, the optimal solution $p_3^{\mathsf{WLW}}(\Delta)$ is computed in direct functional dependency on Δ. Differentiating allows the verification that it has a uniquely defined maximum on $[0, 1]$ in $\Delta_{\max} \approx 0.77787$ yielding the probability $p_3^{\mathsf{WLW}}(\Delta_{\max}) \approx 0.05234$. The whole graph of $p_3^{\mathsf{WLW}}(\Delta_{\max})$ with its maximum is illustrated in figure 5.1.

As a result, the respective pairwise coupling approach as well as the presented generalization using non-constant pairwise weights $\mathsf{w}_{i,j}$ are capable of discarding local increased probabilities in favor of a globally more consistent estimation. However, the additional cost for this is the requirement to solve an optimization problem at each prediction. Still, this is an important advantage over pairwise coupling approaches for which this property does not hold. As most of the pairwise classifiers are not competent during each prediction, they can easily produce unreliable large probabilities for one of the two classes. Consequently, discarding *local* information in favor of a more consistent, global one is a remarkable property.

With respect to this difference between the existing pairwise coupling techniques, it should be emphasized that the evidence-theoretic approach (5.1) is also monotonic in this sense, i.e. any increase in pairwise probabilities increases the overall probability for the respective class. Still, this only holds if all pairwise probabilities are strictly positive. For degenerated probabilities of zero, the whole product remains zero even if one factor increases.

To prove this statement, assume that there is a counterexample, i.e. for any number of classes k and pairwise probability matrix Φ satisfying $0 < \phi_{i,j} < 1$ for all $1 \leq i, j \leq k$ such that moving $\Delta > 0$ from $k-1$ to k does not increase[2] p_k if the latter is computed according to (5.1). In particular, the dependency on Δ is represented by

$$\phi_{i,j}(\Delta) \coloneqq \begin{cases} \phi_{i,j} & \text{if } i \leq k-1 \ \vee \ j \leq k-1 \\ \phi_{k-1,k} - \Delta & \text{if } i = k-1 \ \wedge \ j = k \\ \phi_{k,k-1} + \Delta & \text{if } i = k \ \wedge \ j = k-1 \end{cases} \tag{5.18}$$

for all $1 \leq i, j \leq k$ and sufficiently selected Δ (i.e. $\phi_{k-1,k} - \Delta > 0$ and $\phi_{k,k-1} + \Delta < 1$) such that the posterior probability estimates are computed as

$$p_i(\Delta) \ = \ \frac{1}{Z(\Delta)} \cdot \prod_{\substack{j=1 \\ j \neq i}}^{k} \phi_{i,j}(\Delta) \tag{5.19}$$

with normalization constant

$$Z(\Delta) \ = \ \sum_{i=1}^{k} \prod_{\substack{j=1 \\ j \neq i}}^{k} \phi_{i,j}(\Delta) \tag{5.20}$$

such that p sums to one. Now, by assumption holds $p_k(\Delta) \leq p_k(0)$ for sufficiently small $\Delta > 0$, i.e.

$$\frac{1}{Z(\Delta)} \cdot \prod_{j=1}^{k-1} \phi_{k,j}(\Delta) \ \leq \ \frac{1}{Z(0)} \cdot \prod_{j=1}^{k-1} \phi_{k,j}(0) \tag{5.21}$$

holds, which implies $Z(\Delta) > Z(0)$ because $\phi_{k,k-1}(\Delta) > \phi_{k,k-1}(0)$. From this,

$$p_i(\Delta) \ = \ \frac{1}{Z(\Delta)} \cdot \prod_{\substack{j=1 \\ j \neq i}}^{k} \phi_{i,j}(\Delta) \ < \ \frac{1}{Z(0)} \cdot \prod_{\substack{j=1 \\ j \neq i}}^{k} \phi_{i,j}(\Delta) \ < \ \frac{1}{Z(0)} \cdot \prod_{\substack{j=1 \\ j \neq i}}^{k} \phi_{i,j}(0) < p_i(0) \tag{5.22}$$

is obtained for all $1 \leq i \leq k-1$. Thus, $\sum_{i=1}^{k} p_i(\Delta) < \sum_{i=1}^{k} p_i(0) = 1$ yields a contradiction such that a corresponding pairwise probabilities matrix cannot exist.

Summary

In summary, the evidence-theoretic approach to pairwise coupling has a strong theoretical background that revealed interesting insights with respect to the existing alternatives. Particular relevant are their extended versions to allow a non-uniform weight for each pairwise prediction, which is presumably the most systematic approach to deal with the non-competence problem.

With respect to influences in practice like noise, it still might be preferred to solve the generalized problem (5.14) to combine both, robustness with systematic addressing the non-competence problem. Thus, the insights from analyzing pairwise coupling in the context of evidence theory were used to systematically improve the existing pairwise coupling strategies. However from a practical point of view, it still remains at least unclear whether non-constant weights are reasonably proportional

[2] It is assumed without loss of generality that Δ is moved from $k-1$ to k, simply from permuting the indices accordingly.

to the unknown pairwise posterior probability $P(\mathsf{y} \in \{i,j\} \mid \mathsf{x})$ in general. Nevertheless, the next section aims at recovering dynamic classification as a special case of generalized pairwise coupling.

5.4 Dynamic Classification using Pairwise Coupling

One of the main aims of this work is the generalization of multi-class classification into a dynamic context. Here, the strategies of the last section that integrated arbitrary, non-constant weightings into pairwise coupling can also be used to systematically integrate a dynamic target set. The respective results are similar generalizations to the ones provided by using evidence theory as presented in section 4.3. Still, they lack a similar theoretic foundation as pairwise coupling in general does.

If $\emptyset \neq \mathcal{M} \subseteq \mathcal{Y}$ refers to the dynamic target class set such that $|\mathcal{M}| \geq 2$ is assumed without loss of generality (otherwise the classification problem is trivial), a possible strategy to integrate the dynamic class information is to modify the weights to

$$\hat{\mathsf{w}}_{i,j}(\mathsf{x}) := \begin{cases} \mathsf{w}_{i,j}(\mathsf{x}) & \text{if } \{i,j\} \subseteq \mathcal{M} \\ 0 & \text{otherwise} \end{cases} \tag{5.23}$$

which is reasonable for all generalized pairwise coupling approaches that multiplicatively combine the predictions $\phi_{i,j}(\mathsf{x})$ with the respective weights $\mathsf{w}_{i,j}(\mathsf{x})$. This in particular holds for the previously presented ones. Consequently, only classifiers that differentiate between still recognizable classes inside the target set $\mathcal{M}$ receive non-zero weights in the remaining coupling process. In particular, the similarity to modeling (4.78) should be noted.

Besides this, an explicit additional constraint might be required in general that the posterior probability estimation vector $p(\mathsf{x})$ resulting from the pairwise coupling is non-zero only on $\mathcal{M}$, i.e. $\sum_{i \in \mathcal{M}} p_i(\mathsf{x}) = 1$ or equivalently $\sum_{i \notin \mathcal{M}} p_i(\mathsf{x}) = 0$ holds. However for the three pairwise coupling techniques of main interest, this constraint is redundant, as will be shown in advance.

Integrating (5.23) into generalized probabilistic voting (5.12) yields

$$\begin{aligned} p_i^{\mathsf{GVote}}(\mathsf{x}) &= \frac{1}{\sum_{\ell=1}^{k-1} \sum_{j=\ell+1}^{k} \hat{\mathsf{w}}_{i,j}(\mathsf{x})} \cdot \sum_{\substack{j=1 \\ j \neq i}}^{k} \phi_{i,j}(\mathsf{x}) \cdot \hat{\mathsf{w}}_{i,j}(\mathsf{x}) \\ &= \frac{1}{\sum_{\ell \in \mathcal{M}} \sum_{j \in \mathcal{M}: j > \ell} \mathsf{w}_{i,j}(\mathsf{x})} \cdot \sum_{j \in \mathcal{M} \setminus \{i\}} \phi_{i,j}(\mathsf{x}) \cdot \mathsf{w}_{i,j}(\mathsf{x}) \end{aligned} \tag{5.24}$$

as the respective coupling result. For the generalized non-dominance criterion the extension is similarly straightforward, however requires to restrict the minimization over $\mathcal{M}$ only such that

$$\begin{aligned} \mathsf{GND}_i(\mathsf{x}) &= \min_{j \in \mathcal{M} \setminus \{i\}} \min(2 \cdot \mathsf{w}_{i,j}(\mathsf{x}) \cdot \phi_{i,j}(\mathsf{x}),\ \mathsf{w}_{i,j}(\mathsf{x}) \cdot (\phi_{i,j}(\mathsf{x}) + \phi_{j,i}(\mathsf{x}))) \\ &= \min_{j \in \mathcal{M} \setminus \{i\}} \mathsf{w}_{i,j}(\mathsf{x}) \cdot \min(2 \cdot \phi_{i,j}(\mathsf{x}),\ \phi_{i,j}(\mathsf{x}) + \phi_{j,i}(\mathsf{x})) \\ &= \min_{j \in \mathcal{M} \setminus \{i\}} \mathsf{w}_{i,j}(\mathsf{x}) \cdot \min(2 \cdot \phi_{i,j}(\mathsf{x}),\ 1) \end{aligned} \tag{5.25}$$

yields the non-dominance vector with respective posterior probability estimation by normalization. Finally, the generalized problem (5.14) results in the modified version

$$\min_{p} \sum_{i \in \mathcal{M}} \sum_{\substack{j \in \mathcal{M} \\ j \neq i}} [\mathsf{w}_{i,j}(\mathsf{x}) \cdot (\phi_{j,i}(\mathsf{x}) \cdot p_i - \phi_{i,j}(\mathsf{x}) \cdot p_j)]^2 \text{ s.t. } \sum_{i \in \mathcal{M}} p_i = 1,\ p_i = 0\ \forall i \notin \mathcal{M} \quad (5.26)$$

which thereafter has to be solved analogously. In each of the three cases, the extension is equivalent to applying the respective generalized version, but to restrict the pairwise probabilities $\phi_{i,j}(\mathsf{x})$ as well as weights $\mathsf{w}_{i,j}(\mathsf{x})$ to only those class indices inside $\mathcal{M}$.

It should be emphasized that even though this modeling aims at generalized pairwise coupling using arbitrary weights $\mathsf{w}_{i,j}(\mathsf{x})$ for the individual classifiers, it still remains applicable for default pairwise coupling approaches as the latter are simply special cases of the former with constant weights $\mathsf{w}_{i,j}(\mathsf{x}) \equiv \mathsf{w}_0 > 0$, as discussed in full detail in section 5.1. Here at least for the pairwise coupling techniques of main interest, $\mathsf{w}_0 = 1$ can be assumed without loss of generality as the respective solutions are already observed to be independent of constant multiplications. As a result, extending the respective pairwise coupling techniques with weights given by (5.23) is equivalent to perform pairwise coupling only with classifiers separating between classes inside $\mathcal{M}$. In particular, all theoretical properties of (5.14) remain valid for (5.26), which is important for practical applications.

Even though the presented approaches to dynamic classification focus on three pairwise coupling techniques, similar extensions are expected to be possible for most if not all existing variants as well. The generalizations are mostly straightforward, however they might require the additional constraint that each coupling method produces non-zero probabilities only on $\mathcal{M}$. In most cases where the combination between pairwise probabilities and weights remains multiplicative, the constraint is most likely redundant, but still remains important in general.

5.5 Computational Aspects

The previous sections of this chapter generalized the classical pairwise coupling approach such that each individual prediction $\phi_{i,j}$ always has to be combined with a weight $\mathsf{w}_{i,j}$, which in particular yields a structured way to integrate dynamic classification into the fusing process. Ideally, the weighting equals the pair's unknown posterior probability $\mathsf{w}_{i,j}(\mathsf{x}) = P(\mathsf{y} \in \{i,j\} \mid \mathsf{x})$, which however is unknown in practice. Existing works introduced the pair-vs-rest correction classifiers for this task and, as shown in section 5.3 by (5.10), this remains a valid surrogate if there is a proportional relationship between posterior probabilities and weights.

Not only even this proportional relationship might be unreasonable to assume, it additionally poses a challenging learning problem. The pair-vs-rest approach shares the same disadvantages of the one-vs-all reduction as imbalanced learning problems that are harder to solve than the ones of the one-vs-one decomposition – eventually even harder than these of the one-vs-all reduction – which can result in unreliable predictions. Using the latter during generalized pairwise coupling can skew the estimation such that a constant weight might still be superior in practice, depending on the respective task.

5.5.1 Extended Decompositions with Large-Scale Models

A related open issue was already presented in section 2.4. State-of-the-art models like large-scale deep neural networks trained on millions of training examples are usually

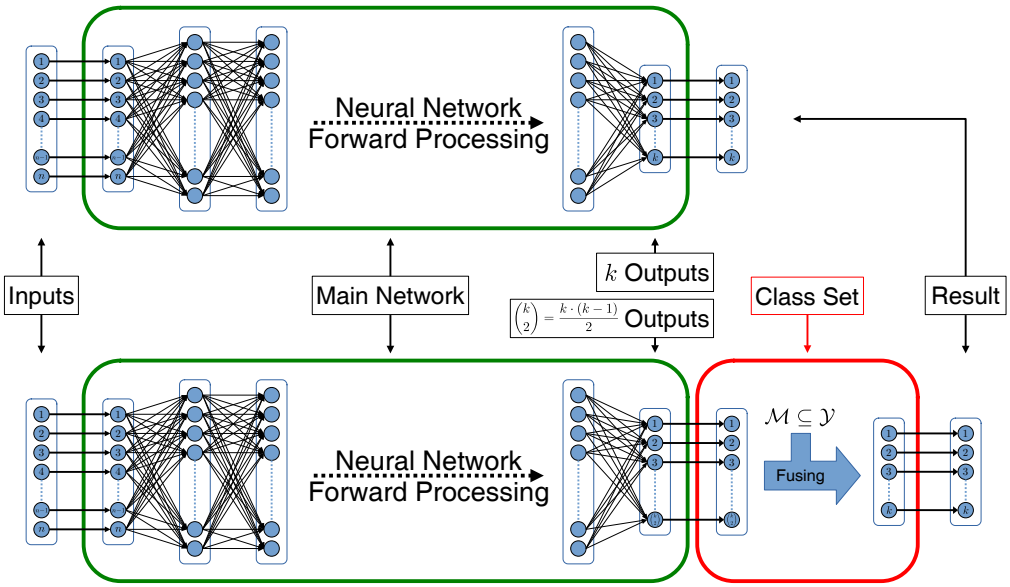


FIGURE 5.2: Illustration of the differences between a one-vs-all softmax model (top) and a one-vs-one neural network with an optional dynamic classification fusing step (bottom).

equipped with a softmax layer that performs the final prediction. The corresponding parameters are jointly estimated because independent trainings of the involved one-vs-all functions have no clear advantage and require to train and deploy k models instead of a single one. This might explain why combining decomposition-based classification besides the one-vs-all reduction in combination with large-scale neural networks received less attraction in the literature.

Two recent works try to extend these capabilities, however these are designed such that the whole data are encoded and decoded, respectively, into different representations besides the one-hot encoding used for the one-vs-all reduction: First, using binary ECOC codewords [Klimo et al. 2021] and applying techniques as presented in full detail in section 2.3 achieving comparable results and second, into a one-vs-one encoding as given by (2.29), where each class label is encoded by $\binom{k}{2}$ ternary outputs from $\{-1, 0, 1\}$ [Pawara et al. 2020]. Here, the last layer consists of $\binom{k}{2}$ hyperbolic tangent functions (tanh) to encode the input accordingly. Even though the authors reported comparable results, their approach in fact does not train one-vs-one classifiers. Instead of separating only classes i and j, both are simultaneously separated from all other classes. This increases the training complexity and also introduces redundancies because each one-vs-all decision boundary in fact is learned $k-1$ times.

One-vs-One Neural Networks

The correct way of training one-vs-one neural networks requires to train the corresponding pairwise prediction functions instead by only using data from the respective two classes. This is computationally complicated as it cannot be integrated into usual forward and backward passes that are performed during neural network training as well as requires to simultaneously process *all* data forwards and backwards.

As a trade-off solution for this problem, the following approach is introduced as a feasible solution to combine large-scale models with the one-vs-one decomposition. First, an arbitrary neural network is trained containing a default softmax prediction

layer based on the one-vs-all decomposition using the given training data. Thereafter, the whole last layer is removed and the remaining network is extended by a one-vs-one layer containing $\binom{k}{2}$ neurons. Each of the latter will be trained using a sigmoidal activation function, which is commonly used for binary classification tasks. Here, two parts are important. First, the sigmoid functions are trained *only* using data from one pair of classes, supplied as a binary-only problem. This training can be performed following default optimizations, e.g. forward/backward passes on the respective model, which has to be iterated $\binom{k}{2}$ times on a network that only contains a single output neuron. Second during these trainings, only the last layer's parameters are optimized, while the remaining network remains fixed. This is important to ensure that all $\binom{k}{2}$ individual networks remain the same besides except only the last layer. Thus after finishing all trainings, there are $\binom{k}{2}$ binary neural networks that form the one-vs-one ensemble. Since they are the same except their last layer, they can be combined into a single model whose last layer consists of all $\binom{k}{2}$ one-vs-one neurons. The resulting network and its differences from a standard one-vs-all softmax model are illustrated in figure 5.2.

It should be emphasized that these steps are only performed in this way to keep the approach feasible for large-scale models. Otherwise, the trainings can either be performed completely independently (as usual the case for simpler models) or after finishing the one-vs-all training, the individual one-vs-one models can be optimized as a whole and not only their last layer. Still, this also requires to deploy $\binom{k}{2}$ independent models that share the same structure (i.e. layers and shapes), but use different coefficients. Even though the approaches focus on neural networks, they can analogously be applied for other large-scale models as long as the prediction characteristics are similar and allow the respective modifications.

Extensions for Generalized Pairwise Coupling

In any case, the one-vs-one networks output a probability matrix Φ in form of (2.23). Depending on the actual implementation, this can be represented by a 1D vector with $\binom{k}{2}$ entries and, consequently, requires pairwise coupling techniques to transfer the individual predictions into an overall posterior probability estimation. This directly points to extending the presented strategy for *generalized* pairwise coupling, i.e. how to compute the weights using large-scale models, too. Here, combining the presented methods with existing approaches to compute the weights yields three possible alternatives:

1. The first alternative replaces the softmax layer in the initial training by a joint optimization of all pair-vs-rest classifiers. In particular, the last layer consists of $\binom{k}{2}$ independent sigmoidal activation units in which each instance's class value y is encoded at training time with a binary vector that contains exactly $k-1$ ones and $\binom{k}{2} - (k-1) = \binom{k-1}{2}$ zeros, respectively. Thereafter, the model can be similarly extended to train the one-vs-one prediction functions. Finally, all pair-vs-rest and one-vs-one predictions can be merged into a layer containing $k \cdot (k-1)$ neurons.

2. Learning the pair-vs-rest classifiers can be a difficult task and there is a quadratic number in k of them. Therefore, it can be significantly faster to train the network first using a softmax layer only (i.e. the one-vs-all decomposition) and, thereafter, learn the pair-vs-rest classifiers by replacing the softmax layer on an existing model. Here, the whole network can either be optimized during

the pair-vs-rest optimization because all classifiers can be jointly optimized or alternatively be fixed as during the one-vs-one training.

3. Both previous approaches directly train the pair-vs-rest classifiers using neural networks. Alternatively, their predictions can also be replaced by the sum of the respective one-vs-all probabilities, $\mathsf{w}_{i,j}(\mathsf{x}) = \mathsf{f}_i(\mathsf{x}) + \mathsf{f}_j(\mathsf{x})$. This approach was introduced as a surrogate for the correction classifiers in their introductory work [Moreira & Mayoraz 1998] to avoid a quadratic number of additional models, however for the presented algorithms it is particularly relevant.

   As discussed in section 2.3, the authors reported decreased classification accuracy if the correcting classifiers were replaced this way. With respect to neural networks, the one-vs-all classifiers are trained anyway to initialize the model such that reusing them comes at practically no additional cost. Furthermore, this training is performed *simultaneously* such that adding the probabilities might produce more reliable estimates of the posterior probability $P(\mathsf{y} \in \{i, j\} \mid \mathsf{x})$ than summing *independent* one-vs-all predictions. Especially since neural networks themselves often perform superior to simpler models like decision trees, it is at least questionable if the respective negative results remain valid here. In total, this approach constructs a neural network whose last layer contains $\binom{k}{2} + k = \binom{k+1}{2}$ neurons, which at least turns it into the most efficient alternative that computes both kinds of predictions.

All of these techniques can also be interpreted as a combination of transfer learning – i.e. transferring models trained on one data set to another one – and classifier calibration. The former since an initial training is performed, the latter because estimating the binary predictors using either the log-loss or the mean squared error is actually the same as performing a calibration step. The only difference is the fact that the model was trained on not exactly the same problem, but a related one.

It might be advantageous – in particular for implementations – to arrange the individual prediction functions into a compact representation given by the following $k \times k$ matrix

$$\begin{pmatrix} \mathsf{f}_1(\mathsf{x}) & \phi_{1,2}(\mathsf{x}) & \phi_{1,3}(\mathsf{x}) & \cdots & \phi_{1,k-1}(\mathsf{x}) & \phi_{1,k}(\mathsf{x}) \\ \mathsf{w}_{2,1}(\mathsf{x}) & \mathsf{f}_2(\mathsf{x}) & \phi_{2,3}(\mathsf{x}) & \cdots & \phi_{2,k-1}(\mathsf{x}) & \phi_{2,k}(\mathsf{x}) \\ \mathsf{w}_{3,1}(\mathsf{x}) & \mathsf{w}_{3,2}(\mathsf{x}) & \mathsf{f}_3(\mathsf{x}) & \cdots & \phi_{3,k-1}(\mathsf{x}) & \phi_{3,k}(\mathsf{x}) \\ \vdots & \vdots & \vdots & \ddots & \vdots & \vdots \\ \mathsf{w}_{k-1,1}(\mathsf{x}) & \mathsf{w}_{k-1,2}(\mathsf{x}) & \mathsf{w}_{k-1,3}(\mathsf{x}) & \cdots & \mathsf{f}_{k-1}(\mathsf{x}) & \phi_{k-1,k}(\mathsf{x}) \\ \mathsf{w}_{k,1}(\mathsf{x}) & \mathsf{w}_{k,2}(\mathsf{x}) & \mathsf{w}_{k,3}(\mathsf{x}) & \cdots & \mathsf{w}_{k,k-1}(\mathsf{x}) & \mathsf{f}_k(\mathsf{x}) \end{pmatrix} \tag{5.27}$$

where the diagonal entries refer to the one-vs-all predictions, the upper triangle refers to the one-vs-one pairwise probabilities and the lower triangle to the weights. This yields a neural network with a square-shaped output layer whose k diagonal entries are the default softmax predictions, i.e. existing models with a vector-valued output layer are generalized and their predictions respectively recovered.

5.5.2 Weight Estimation

Aforementioned approaches generalize pairwise coupling algorithms with particular focus on dynamic classification. Besides strategies that aim at keeping the algorithms computationally tractable in combination with large-scale classification models (e.g. deep neural networks), the computation of the required weights $\mathsf{w}_{i,j}(\mathsf{x})$ is especially relevant in contexts using dynamic class information.

Following existing approaches, the weights can be computed using correction classifiers (WeightCC) or by pairwise sums of one-vs-all predictions (Weight1vA). In most cases, both kinds of learning problems are presumably harder than those of the one-vs-one reduction, where always two actual classes are separated instead of class sets that are merged into new artificial classes.

Most importantly, these strategies expose an additional drawback with respect to dynamic contexts. Applying pairwise coupling *without* one-vs-all or correcting classifiers maximally adapts to the dynamic context since no involved classifier depends in any way on data observed from classes that are excluded from the dynamic target set $\mathcal{M}$. On the reverse, applying generalized pairwise coupling in combination with weights computed in one of the aforementioned ways still implicitly depends on data from all classes.

Weights by Pairwise Coupling

This situation might lead to a potential trade-off problem in practice: On the one hand, applying pairwise coupling while simultaneously supplying dynamic class information becomes completely independent of classes that cannot be observed and is likely to benefit better from the dynamic information than the generalized counterparts that implicitly depend on all classes. On the other hand, the generalized variants can yield improved results, at least according to the reported improvements in respective reference works, as presented in full detail in chapter 2.

Here, the following alternative approach combines both advantages by computing the *weights by pairwise coupling* (Weight1v1). For this, the respective pairwise coupling procedure is applied without supplying any weights to compute an initial posterior probability estimation $q(\mathsf{x})$. Using this, the weights are computed as $\mathsf{w}_{i,j}(\mathsf{x}) = q_i(\mathsf{x}) + q_j(\mathsf{x})$. Thereafter, the generalized pairwise coupling approach is used to compute the final posterior probability estimation $p(\mathsf{x})$. In this way, the weight computation does not depend on data from all classes in dynamic classification contexts. This presumably yields superior results in real-time applications with class sets $\mathcal{Y}$ from which only a significantly smaller portion $\mathcal{M}$ should be predicted.

Generally, this approach can even be iterated: First, no weights are used to compute an initial posterior probability estimation $q(\mathsf{x})$. Thereafter, the weights are initialized by pairwise sums and are used to compute a new posterior probability estimation. After this, the weights are recomputed by summing. This approach can be iterated arbitrarily often, for example until convergence. However, this can be computationally complicated as long as there is no proven convergence property under mild additional assumptions. Alternatively, it requires a different termination criterion that remains arbitrary as well.

With respect to the evidence-theoretic approach to pairwise coupling (5.1), the following result guarantees that these iterations are not required at all because they will not change the predicted class.

Proposition 5.2. *Let the unnormalized probabilities p_i be as in corollary* (4.7)*, i.e.*

$$p_i \;=\; \prod_{\substack{j=1\\ j\neq i}}^{k} (\phi_{i,j} \cdot \mathsf{w}_{i,j}) \;\cdot\; \prod_{\substack{s=1\\ s\neq i}}^{k-1} \prod_{\substack{t=s+1\\ t\neq i}}^{k} (1 - \mathsf{w}_{s,t}) \tag{5.28}$$

for all $i = 1, \ldots, k$. If the weights additionally are computed as

$$\mathsf{w}_{i,j} \;=\; \frac{q_i + q_j}{\sum\limits_{\ell=1}^{k} q_\ell} \quad \textit{with} \quad q_i \;=\; \prod_{\substack{j=1 \\ j \neq i}}^{k} \phi_{i,j} \tag{5.29}$$

it holds

$$\operatorname*{arg\,max}_{1 \leq i \leq k} p_i \;=\; \operatorname*{arg\,max}_{1 \leq i \leq k} q_i \tag{5.30}$$

such that the induced class predictions are equivalent.

Proof. For each $i = 1, \ldots, k$ let $j \neq i$ be arbitrarily selected. First, $q_i \geq q_j$ implies for each $\ell \neq i, j$

$$1 - q_i - q_\ell \;\leq\; 1 - q_j - q_\ell \quad \Rightarrow \quad (1 - q_i - q_\ell)^{-1} \;\geq\; (1 - q_j - q_\ell)^{-1} \tag{5.31}$$

such that multiplying all these inequalities yields

$$\prod_{\substack{\ell=1 \\ \ell \neq i}}^{k} (1 - q_i - q_\ell)^{-1} \;\geq\; \prod_{\substack{\ell=1 \\ \ell \neq j}}^{k} (1 - q_j - q_\ell)^{-1} \tag{5.32}$$

where strict inequality is maintained. By rescaling according to (4.55), it holds

$$p_i \;\propto\; \prod_{\substack{\ell=1 \\ \ell \neq i}}^{k} \frac{\phi_{i,\ell} \cdot \mathsf{w}_{i,\ell}}{1 - \mathsf{w}_{i,\ell}} \;=\; \prod_{\substack{\ell=1 \\ \ell \neq i}}^{k} \phi_{i,\ell} \;\cdot\; \prod_{\substack{\ell=1 \\ \ell \neq i}}^{k} \mathsf{w}_{i,\ell} \;\cdot\; \prod_{\substack{\ell=1 \\ \ell \neq i}}^{k} (1 - \mathsf{w}_{i,\ell})^{-1} \tag{5.33}$$

$$= \; q_i \;\cdot\; \prod_{\substack{\ell=1 \\ \ell \neq i}}^{k} (q_i + q_\ell) \;\cdot\; \prod_{\substack{\ell=1 \\ \ell \neq i}}^{k} (1 - q_i - q_\ell) \tag{5.34}$$

$$= \; q_i \cdot (q_i + q_j) \cdot (1 - q_i - q_j)^{-1} \;\cdot\; \prod_{\substack{\ell=1 \\ \ell \neq i,j}}^{k} (q_i + q_\ell) \;\cdot\; \prod_{\substack{\ell=1 \\ \ell \neq i,j}}^{k} (1 - q_i - q_\ell) \tag{5.35}$$

as well as analogously

$$p_j \;\propto\; q_j \cdot (q_i + q_j) \cdot (1 - q_i - q_j)^{-1} \;\cdot\; \prod_{\substack{\ell=1 \\ \ell \neq i,j}}^{k} (q_j + q_\ell) \;\cdot\; \prod_{\substack{\ell=1 \\ \ell \neq i,j}}^{k} (1 - q_j - q_\ell) \tag{5.36}$$

such that $q_i \geq q_j$ implies $p_i \geq p_j$ as well as $q_i > q_j$ implies $p_i > p_j$, respectively. Therefore, the maximizing indices coincide as claimed. □

The last result not only proves that the weighted and unweighted evidence-theoretic predictions are equivalent, but additionally the proof shows that a multiplicative voting with probabilities

$$p_i \;\propto\; \prod_{\substack{j=1 \\ j \neq i}}^{k} (\phi_{i,j} \cdot \mathsf{w}_{i,j}) \tag{5.37}$$

would result in an equivalent class prediction under the same assumptions.

The selection of the best weight estimation technique presumably remains task-specific in practice. Still, the presented approach at least allows the computation of the weights with a maximum adaption to dynamic class information.

5.6 Summary

Based on the provided insights of evidence theory in chapter 4, the further contributions are three-fold: First, a new interpretation of the non-competence problem in pairwise coupling is presented as an *assumed constant* weighting during the pairwise coupling process, which results in both, an evidence-theoretic approach to pairwise coupling using multiplicative instead of additive voting and a corresponding Bayesian counterpart of a full-conflict evidence-theoretic modeling.

Thereafter, the second main contribution generalizes existing pairwise coupling techniques such that the individual one-vs-one predictions $\phi_{i,j}(\mathsf{x})$ are combined with non-constants weights $\mathsf{w}_{i,j}(\mathsf{x})$. In particular, existing approaches are recovered as special cases with constant weights, similar to existing works that introduced the correcting classifiers for probabilistic voting. Here, the corresponding coupling result can be computed using extended variants of existing algorithms, especially focusing on the three most relevant ones. With focus on dynamic classification, the evidence-theoretic approach was integrated into pairwise coupling by setting a corresponding selection of weights to zero. This yields two different approaches to dynamic classification in general: either based on standard or generalized pairwise coupling.

Finally, solutions for two different computational aspects are presented that are particularly relevant for practical applications. First, transferring decomposition-based classification approaches to large-scale models like deep learning neural networks, where for computational constraints a single final model is required and second, computing the weights in generalized pairwise coupling such that no predictors are used that depend on data from all classes.

In combination with the results of chapter 4, two different strategies for dynamic classification are developed. For practical applications, it is interesting to compare standard and generalized pairwise coupling as well as to analyze how supplying dynamic class information can improve the prediction accuracy. Thus, the next chapter presents evaluation strategies and empirically compares the different algorithms.

Chapter 6

Evaluation

The last two chapters 4 and 5 presented different strategies to integrate dynamic class information into the fusing process of pairwise coupling, therefore a feasible integration into real-world applications is possible. All presented algorithms are based on decomposition-based classification approaches, thus they require probabilistic predictions from the individual classifiers. These can be computed using calibration algorithms as presented in chapters 2 and 3, respectively, which emphasizes their relevance in the following evaluation.

With respect to integrating dynamic information into real-world applications, the most important aspects are how large improvements yielded from integrating dynamic information are and which techniques mostly improve the prediction accuracy from this integration. Consequently, at first evaluation metrics for dynamic classification are required that depend not only on data, but additionally on the respective target set $\mathcal{M}$.

Therefore, in section 6.1 at first evaluation metrics for dynamic classification are derived. Thereafter, the introduced new algorithms are compared to state-of-the-art reference techniques in section 6.2 in a thorough empirical evaluation comprising several experiments. Finally, section 6.3 applies the algorithms in an actual real-world application where the dynamic class information successfully improves the recognition accuracy.

6.1 Evaluation Metrics

There is a natural demand to compare the different approaches with respect to both, their main accuracy results obtained as classification algorithms in general as well as their improvements gained from integrating dynamic class information that simplifies the respective task in particular. For the former, standard evaluation metrics like the classification or error rate, respectively, can be used, while for the latter, a direct evaluation metric is not available. Here, every possible choice has to depend on the corresponding dynamic target set $\mathcal{M}$, which directly points to a related problem: Existing reference data do not contain information about possible target sets, thus they have to be approximated, which requires a respective strategy. Both issues are addressed by the following part.

6.1.1 Dynamic Risk

Computing evaluation metrics for a given prediction algorithm f as an average error $\mathcal{R}_{\text{emp}}(\mathsf{f})$ on validation or test data using a loss function L – in case of classification problems mostly the binary loss counting the incorrect classifications – means that a Monte-Carlo approximation of the risk $\mathcal{R}(\mathsf{f})$ is computed, cf. equations (2.2) and (2.3). Even though the latter cannot be computed in general because it depends

on the unknown data-generating distribution P, it still allows a theoretical way to integrate a dynamic target set into the risk calculation.

Integrating a set $\mathcal{M} \subseteq \mathcal{Y}$ such that at prediction time $P(\mathsf{y} = i \mid \mathsf{x}) = 0$ holds for all $i \notin \mathcal{M}$ means that the classification function as well as the data-generating distribution change, i.e. $\mathsf{f} = \mathsf{f}_\mathcal{M}$ and $P = P_\mathcal{M}$ depend on $\mathcal{M}$. For the former, it is assumed that f deterministically depends an $\mathcal{M}$ such that in particular $\mathsf{f} = \mathsf{f}_\mathcal{Y}$ holds. Assuming that there is a density function $p(\mathsf{x}, \mathsf{y}) = p_\mathcal{M}(\mathsf{x}, \mathsf{y})$, the risk of $\mathsf{f}_\mathcal{M}$ with respect to $P_\mathcal{M}$ can be equivalently computed as

$$\mathcal{R}(\mathsf{f}_\mathcal{M} \mid P_\mathcal{M}) \;=\; \int\limits_{\mathcal{X}\times\mathcal{Y}} L(\mathsf{f}_\mathcal{M}(\mathsf{x}), \mathsf{y}) \cdot p_\mathcal{M}(\mathsf{x}, \mathsf{y}) \,\mathrm{d}(\mathsf{x}, \mathsf{y}) \tag{6.1}$$

such that the risk's dependency on $\mathcal{M}$ is expressed by the joint density $p_\mathcal{M}(\mathsf{x}, \mathsf{y})$. For each $\mathsf{y} \notin \mathcal{M}$, without loss of generality[1] $p_\mathcal{M}(\mathsf{x}, \mathsf{y}) = 0$ can be assumed for each $\mathsf{x} \in \mathcal{X}$. Otherwise, the joint distribution can be factorized as

$$p_\mathcal{M}(\mathsf{x}, \mathsf{y}) \;=\; p_\mathcal{M}(\mathsf{x} \mid \mathsf{y}) \cdot P_\mathcal{M}(\mathsf{y}) \tag{6.2}$$

for each $\mathsf{y} \in \mathcal{M}$. Here, it is reasonable to assume that the class-conditional likelihoods do not depend on $\mathcal{M}$ such that $p_\mathcal{M}(\mathsf{x} \mid \mathsf{y}) \equiv p(\mathsf{x} \mid \mathsf{y})$ holds in (6.2). Therefore, the dynamic class information cause a change in the prior probabilities $P_\mathcal{M}(\mathsf{y})$. As they are usually estimated as the fraction of respective training data, estimating them on data with classes inside $\mathcal{M}$ is equivalent to assuming a simple rescaling of the prior probabilities, which is the same as assuming a proportional relationship between the non-zero prior probabilities over $\mathcal{Y}$ and $\mathcal{M}$, respectively

$$P_\mathcal{M}(\mathsf{y}) \;=\; \frac{1}{\sum\limits_{i \in \mathcal{M}} P(i)} \cdot P(\mathsf{y}) \tag{6.3}$$

for each $\mathsf{y} \in \mathcal{Y}$. Using the last two equations allows the computation of a Monte-Carlo approximation of (6.1), however this only holds for a fixed target set $\mathcal{M}$.

A desired property of the target class set $\mathcal{M} \subseteq \mathcal{Y}$ is that it can change over time. This means that there is a distribution $Q(\mathcal{M})$ over the sets of possible class sets that defines how likely each set $\mathcal{M}$ is to be observed at prediction time. Using this, the expected value of (6.1) with respect to $Q(\mathcal{M})$

$$\begin{aligned} \mathbb{E}_\mathcal{M}\left[\mathcal{R}(\mathsf{f}_\mathcal{M} \mid P_\mathcal{M})\right] \;&=\; \sum_{\emptyset \neq \mathcal{M} \subseteq \mathcal{Y}} Q(\mathcal{M}) \cdot \mathcal{R}(\mathsf{f}_\mathcal{M} \mid P_\mathcal{M}) \\ &=\; \sum_{\emptyset \neq \mathcal{M} \subseteq \mathcal{Y}} Q(\mathcal{M}) \cdot \int\limits_{\mathcal{X}\times\mathcal{Y}} L(\mathsf{f}_\mathcal{M}(\mathsf{x}), \mathsf{y}) \cdot p(\mathsf{x} \mid \mathsf{y}) \cdot P_\mathcal{M}(\mathsf{y}) \,\mathrm{d}(\mathsf{x}, \mathsf{y}) \end{aligned} \tag{6.4}$$

can be defined as the *dynamic risk*. It is a strict generalization of the risk because the distribution $Q(\mathcal{M})$ can degenerate into $Q(\mathcal{Y}) = 1$ such that the dynamic risk coincides with the risk. Thus, it is a natural generalization for arbitrary distributions $Q(\mathcal{M})$.

However, the dynamic risk cannot be computed directly for two different reasons: Not only the data-generating distribution $P = P_\mathcal{M}$ is unknown, the same also holds for the dynamic target set distribution Q. If $D = \{(\mathsf{x}_i, \mathsf{y}_i) \;:\; i = 1, \ldots, r\}$ refers to a

[1] Formally, $p_\mathcal{M}(\mathsf{x}, \mathsf{y}) = 0$ only holds almost everywhere. Still, the remaining zero set does not influence the integral.

given test data set, for a given $\mathcal{M} \subseteq \mathcal{Y}$ the induced subset

$$D(\mathcal{M}) \coloneqq \{(\mathsf{x}_i, \mathsf{y}_i) \ : \ \mathsf{y}_i \in \mathcal{M}\} \tag{6.5}$$

allows the computation of

$$\mathcal{R}_{\text{emp}}(\mathsf{f}_{\mathcal{M}} \mid D(\mathcal{M})) \coloneqq \frac{1}{|D(\mathcal{M})|} \cdot \sum_{(\mathsf{x}_i, \mathsf{y}_i) \in D(\mathcal{M})} L(\mathsf{f}_{\mathcal{M}}(\mathsf{x}_i), \mathsf{y}_i) \tag{6.6}$$

as an approximation of (6.1), but extending this into an approximation of (6.4) is not possible if Q is unknown. There might be applications where this distribution is known, however to evaluate reference data sets, approximation strategies are necessary. The latter yield probabilities $q(\mathcal{M})$, thus the dynamic risk can be approximated by

$$\mathcal{R}_{\text{emp}}(\mathsf{f}_{\mathcal{M}} \mid q) \coloneqq \sum_{\emptyset \neq \mathcal{M} \subseteq \mathcal{Y}} q(\mathcal{M}) \cdot \frac{1}{|D(\mathcal{M})|} \cdot \sum_{(\mathsf{x}_i, \mathsf{y}_i) \in D(\mathcal{M})} L(\mathsf{f}_{\mathcal{M}}(\mathsf{x}_i), \mathsf{y}_i) \tag{6.7}$$

such that defining the probabilities $q(\mathcal{M})$ using respective sampling strategies remains.

6.1.2 Sampling Strategies

The previous analysis shows that the computation of the dynamic risk depends on sampling the class set distribution Q by defining probabilities q. The first reasonable restriction is to assume $|\mathcal{M}| \geq 2$ because otherwise, the problem is already solved. Consequently, there are $2^k - (k+1)$ different remaining sets. This leads to the first strategy to exhaustively use all possible data sets at prediction time. Since this involves an exponential complexity, it is only feasible for data sets with up to 10-15 classes. Still, the probabilities $q(\mathcal{M})$ have to be defined.

Here, the first modeling assumes that all sets containing the same number of classes receive in sum the same probabilities, i.e. on average observing a target set $\mathcal{M}$ containing $|\mathcal{M}| = \ell \leq k$ classes is equally probable, independent of ℓ. As there are $\binom{k}{\ell}$ possible choices for $\mathcal{M}$ containing ℓ of k classes, this results in probabilities

$$q_{\text{all}}(\mathcal{M}) = \begin{cases} \left((k-1) \cdot \binom{k}{\ell}\right)^{-1} & \text{if } \ell = |\mathcal{M}| \geq 2 \\ 0 & \text{otherwise} \end{cases} \tag{6.8}$$

where target sets containing only a single class are excluded for previously discussed reasoning. Alternatively, also a uniform distribution over all sets $\mathcal{M}$ with at least two remaining classes is possible. However, this would introduce a relatively strong bias towards medium-sized sets as there are $\binom{k}{\ell} \in \Theta(k^\ell)$ subsets for each $\ell \leq \frac{k}{2}$.

In comparison with the default case where always all classes are possible (i.e. $\mathcal{M} = \mathcal{Y}$), sampling according to (6.8) might still introduce a relatively strong skew as only a $\frac{1}{k-1}$ fraction of the probability mass is placed on $\mathcal{Y}$. To mitigate this issue, the following alternative modeling uses the probabilities

$$q_t(\mathcal{M}) = \begin{cases} 1-t & \text{if } \mathcal{M} = \mathcal{Y} \\ \frac{t}{k} & \text{if } |\mathcal{M}| = k-1 \\ 0 & \text{otherwise} \end{cases} \tag{6.9}$$

such that an arbitrary probability mass $t \in [0,1]$ is equally distributed on all target sets from which only a single class is removed (i.e. $|\mathcal{M}| = k-1$), while the remaining probability remains on $\mathcal{Y}$. Thus, it forms a reasonable lower bound for each setting where at least sometimes any form of dynamic class information restricting the target set is available.

Even though (6.9) depends on the actual value of t, it yields a reasonable lower bound to any distribution observed under real-world conditions because t can be selected sufficiently small. Now, an expected improvement means that the inequality

$$\mathcal{R}_{\text{emp}}(\mathsf{f}_{\mathcal{M}} \mid q_t) \overset{!}{<} \mathcal{R}_{\text{emp}}(\mathsf{f}_{\mathcal{M}} \mid q_0) = \mathcal{R}_{\text{emp}}(\mathsf{f} \mid D) \tag{6.10}$$

holds for the respective value $t > 0$. Therefore, a reasonable lower-bound check is obtained by criterion (6.10). Still for applying it as part of evaluations, it should not be depend on a parameter. Here, the definitions of $\mathcal{R}_{\text{emp}}(\mathsf{f}_{\mathcal{M}})$ in (6.6) and $q_t(\mathcal{M})$ in (6.9), respectively, yield

$$\mathcal{R}_{\text{emp}}(\mathsf{f}_{\mathcal{M}} \mid q_t) = (1-t) \cdot \mathcal{R}_{\text{emp}}(\mathsf{f}_{\mathcal{M}} \mid q_0) + t \cdot \mathcal{R}_{\text{emp}}(\mathsf{f}_{\mathcal{M}} \mid q_1) \tag{6.11}$$

such that condition (6.10) is equivalent to

$$(1-t) \cdot \mathcal{R}_{\text{emp}}(\mathsf{f}_{\mathcal{M}} \mid q_0) + t \cdot \mathcal{R}_{\text{emp}}(\mathsf{f}_{\mathcal{M}} \mid q_1) \overset{!}{<} \mathcal{R}_{\text{emp}}(\mathsf{f}_{\mathcal{M}} \mid q_0) \tag{6.12}$$

which simplifies into

$$\mathcal{R}_{\text{emp}}(\mathsf{f}_{\mathcal{M}} \mid q_1) \overset{!}{<} \mathcal{R}_{\text{emp}}(\mathsf{f}_{\mathcal{M}} \mid q_0) = \mathcal{R}_{\text{emp}}(\mathsf{f} \mid D) \tag{6.13}$$

such that (6.13) is an easily verifiable criterion for the existence of an improvement.

For this reason, by sampling according to (6.8) and (6.9), respectively, two possible evaluation metrics for dynamic classification are obtained. Even though the latter depends on the free parameter t, selecting $t = 1$ yields a reasonable criterion to check for an improvement.

In the same way as the empirical risk in combination with the binary loss can be interpreted as an error rate – whose complement yields the accuracy – also the *dynamic classification accuracy*

$$\mathsf{Acc}(\mathsf{f}_{\mathcal{M}} \mid q) \coloneqq \sum_{\emptyset \neq \mathcal{M} \subseteq \mathcal{Y}} q(\mathcal{M}) \cdot \frac{1}{|D(\mathcal{M})|} \cdot \sum_{(\mathsf{x}_i, \mathsf{y}_i) \in D(\mathcal{M})} \mathbb{1}(\mathsf{f}_{\mathcal{M}}(\mathsf{x}_i), \mathsf{y}_i) \tag{6.14}$$

can be defined for a given sampling q. Because classically it is often preferred to present accuracies than error rates, in the following part also dynamic classification accuracies instead of error rates are given. For $\mathcal{M} = \mathcal{Y}$, this recovers the standard accuracy (2.4). Still, in the same way as the risk can be defined for each loss function, the same holds for the dynamic risk.

6.2 Empirical Comparison

After presenting evaluation metrics, an empirical study is performed to compare the different methods with respect to both, their accuracies as well as their improvements yielded from integrating dynamic class information. Here, at first a short overview of the methods that are compared in the following part is given.

6.2.1 Overview of Methods

As a reference to existing approaches, all techniques presented in section 2.4 are used. In addition to these, the following newly introduced methods are applied:

- The one-vs-one decomposition using the evidence-theoretic modeling (ET) given by (4.52) that uses constant weights $\mathsf{w}_{i,j}(\mathsf{x}) \equiv \mathsf{w}_0 > 0$, which can be interpreted as an evidence-theoretic pairwise coupling approach as presented in section 5.1 and (5.1).

- The one-vs-one decomposition in combination with non-constant weights using the evidence-theoretic modeling (GET) as part of generalized pairwise coupling. In combination with weight estimation by pairwise coupling (Weight1v1), this yields an equivalent approach to the previous one (cf. proposition 5.2).

- The one-vs-one decomposition in combination with non-constant weights using generalized voting (GVote) as part of generalized pairwise coupling. It should be emphasized that this is equivalent to the existing correcting classifiers approach (1v1 + CC) discussed in full detail in section 2.3 as long as the weights are computed using the latter. Still, other methods discussed in section 5.5 can be used alternatively to supply the weights.

- The one-vs-one decomposition in combination with non-constant weights using the generalized non-dominance criterion (GND) as part of generalized pairwise coupling.

- The one-vs-one decomposition in combination with non-constant weights by solving (5.14) as part of generalized pairwise coupling (GWLW).

It should be emphasized that the evidence-theoretic prediction based on the one-vs-all decomposition given by (4.21) is excluded because its class prediction rule is equivalent to the one of the softmax prediction (cf. proposition 4.4) such that the classification accuracies coincide. As a consequence, there are no ambiguities regarding the underlying reduction in combination with evidence theory as it always refers to the one-vs-one decomposition.

Similarly to the methods described in section 2.4, each approach only refers to the decomposing and fusing process, but still needs a learning algorithm to solve the subproblems obtained from applying the respective decomposition. An important difference to each of the existing methods is that here, each method is *designed* to support the integration of dynamic context information. In case of the default pairwise coupling approaches, this extension recovers the restriction to pairwise coupling on the remaining classes only.

Two particular algorithms are used to solve the base classification problems: First, support vector machines (SVM) based on LIBLINEAR [Fan et al. 2008], which were already applied during the calibration study in section 3.4. Besides this, random forests (RF) consisting of 50 randomized decision trees using the implementation from OpenCV [Bradski 2000] were used as a second classification model.

In combination with the respective fusing approaches, probabilistic predictions are required from each binary classifier to transform them into an overall posterior probability estimation. Based on the results of chapter 3, Platt scaling is applied for the individual support vector machines. The sigmoidal parameters are computed using the implementation from LIBSVM [Chang & Lin 2011], without creating a hold-out data set. In case of the random forests, each tree is trained using a bootstrap,

	Name	Samples (r)	Features (n)	Classes (k)	Size [MB]
1	Annealing	898	8	5	0.02
2	Arrhythmia	452	257	13	0.35
3	Car Evaluation	1728	21	4	0.08
4	Covertype	581012	54	7	75.75
5	Crop mapping using fused optical-radar	325834	174	7	428.7
6	Dermatology	366	33	6	0.03
7	Ecoli	336	7	8	0.01
8	Gas Sensor Array Drift	13910	128	6	17.29
9	Gesture Phase Segmentation	9873	32	5	3.5
10	Glass Identification	214	9	6	0.01
11	Human Activity Recognition Using Smartphones	10299	561	6	67.29
12	Localization Data for Person Activity	164860	32	11	18.65
13	MoCap Hand Postures	78095	10	5	12.51
14	Multiple Features	2000	649	10	6.75
15	Nursery	12960	26	5	0.71
16	Optical Recognition of Handwritten Digits	5620	62	10	0.81
17	Page Blocks	5473	10	5	0.24
18	PAMAP2 Physical Activity Monitoring	175498	52	12	78.43
19	Pen-Based Recognition of Handwritten Digits	10992	16	10	0.55
20	PUC-Rio	165633	21	5	12.6
21	Sensorless Drive Diagnosis	58509	48	11	25.68
22	Statlog (Shuttle)	58000	9	7	1.6
23	UJIIndoorLoc	21048	520	3	43.85
24	Weight Lifting Exercises	39242	51	5	9.34
25	Yeast	1484	8	10	0.06
26	Zoo	101	16	7	0.004

TABLE 6.1: Individual data sets and their most important properties.

i.e. a randomized subselection of the original training data. Thereafter, each tree is calibrated by traversing it with the whole data also including the out-of-bag data that was not used to train the tree, and computing a probability associated to each leaf node. During prediction, this yields a posterior probability from each tree such that computing the respective average over the whole forest yields a posterior probability estimation for the binary subproblem.

6.2.2 Reference Data

After summarizing the different methods, a selection of 26 real-world data sets covering various applications from the UCI Machine Learning Repository [Dua & Graff 2019] were used in a thorough empirical study. Similar to the data sets used in section 3.4, table 6.1 summarizes their most important properties, while all further information including the necessary steps to compute the used data from publicly available sources are presented in the supplementary material. Most importantly, the data sets were selected such that the number of classes is sufficiently moderate such that an exhaustive iteration over all 2^k class sets remains challenging but feasible. Consequently, evaluation metric (6.9) can be computed as well. The main aim of the study is to answer the following questions:

- Is there a substantial or significant improvement from integrating dynamic class information in the fusing process?
- If there is an improvement, how large is the benefit?
- Which decomposition and fusing method yields the best results, both with respect to the base accuracy where all classes are possible as well as the improvement from integrating dynamic class information?

It is important to emphasize that all data sets contain no information about possible dynamic target sets, still sampling according to the strategies presented in

Algorithm	Method	WeightCC	Weight1vA	Weight1v1
SVM	GVote	2.115	2.596	1.288
	GND	2.673	2.096	1.231
	GWLW	2.596	1.942	1.462
	GET	2.346	2.404	1.250
RF	GVote	2.308	2.019	1.673
	GND	2.731	1.885	1.385
	GWLW	2.231	1.885	1.885
	GET	2.385	2.096	1.519

TABLE 6.2: Average ranks of the three weight estimation techniques for each classifier and respective fusing method.

section 6.1 is possible. To compare the methods, a 10-fold stratified cross validation was used to partition each data set into train and test data, where each feature column was standardized to mean 0 and standard deviation 1 based on the respective training data's values.

Thereafter, all reductions were trained in combination with both classification algorithms (support vector machines and random forests). To achieve a maximum comparability, the same partitions are used for the individual trainings as well as the respective classifiers are only trained once and shared between the different decompositions. Thus on each data set and each classification algorithm, the one-vs-all, one-vs-one and correction classifiers are trained once per fold. The methods differ in the final fusing step that transforms the individual predictions into an overall posterior estimation and in the adaption capabilities of integrating dynamic context information into these fusing approaches.

First Experiment: Weight Estimation

Before targeting the study's main aims, the first evaluation focuses on the weight estimation techniques required for generalized pairwise coupling used in the following evaluations. As discussed in section 5.5, this is a particularly relevant issue because computing weights by using pairwise coupling does not require to train the one-vs-all or correction classifiers at all. Consequently, it substantially increases the training efficiency as the induced problems of both reductions are harder learning tasks than those of the one-vs-one decomposition. Furthermore, it allows the most flexibility with respect to the integration of dynamic class information.

To compare the different weight estimation techniques, the weights were estimated using the three different alternatives from section 5.5, i.e. either using the correction classifiers (WeightCC), the sum of one-vs-all predictions (Weight1vA) or using iterated pairwise coupling (Weight1v1). As only the four generalized pairwise coupling techniques depend on the weights, this comparison restricts to just these approaches.

The methods are ranked by their respective classification accuracy per data set using the same standard procedure [Demšar 2006] that was already used in section 3.4. Computing the average ranks yields an overall score for each classifier and weight estimation technique, respectively, which are presented in table 6.2.

Generally, it is possible to combine each of the evaluations with the Friedman test and the post hoc Nemenyi test. This results in eight critical difference diagrams that are available in the supplementary material. The respective critical difference is 0.65 such that groups of not significantly different techniques can generally be identified in table 6.2. However, it is important to emphasize that the tests are not independent

	SVM	RF
Softmax	26 (0.0000003)	25 (0.0000080)
Vote	26 (0.0000003)	26 (0.0000003)
ND	26 (0.0000003)	26 (0.0000003)
WLW	24 (0.0001049)	26 (0.0000003)
CombVote	26 (0.0000003)	26 (0.0000003)
NOV@	26 (0.0000003)	26 (0.0000003)
GVote	26 (0.0000003)	26 (0.0000003)
GND	26 (0.0000003)	26 (0.0000003)
GWLW	26 (0.0000003)	26 (0.0000003)
ET	26 (0.0000003)	26 (0.0000003)

TABLE 6.3: Number of data sets on which criterion (6.13) reports an improvement for support vector machines and random forests, respectively. Additionally, a Bonferroni-corrected p-value is given.

such that an analysis focusing on the statistical significance additionally requires a Bonferroni correction. Still, for the main conclusion this is not even necessary at all. Most importantly, the weight estimation by iterated pairwise coupling (Weight1v1) yields the best results (lowest average rank) in any case such that, even if the differences were not statistically significant, they still remain slightly superior to at least comparable. The selection of the best technique might still remain task specific in general, still these results clearly favor the weight estimation by iterated pairwise coupling.

This is a remarkable result as this weight estimation does not need to train one-vs-all or pair-vs-rest classifiers such that it avoids to solve hard unbalanced training problems. Similarly, this might also be the explanation for this observation: The complex learning tasks required to estimate the weights using either the one-vs-all or pair-vs-rest classifiers yield less reliable results. Not only the learning task itself is challenging, also the following calibration step might compute too unreliable posterior probability estimations, which are used as weights.

Besides these empirically observed advantages, weight estimation by iterated pairwise coupling at least in theory allows an improved adaption to dynamic context information. Therefore in the following evaluations focusing on the respective improvements, the weights were estimated this way. Proposition (5.2) shows that ET and GET predictions and thus the classification accuracies coincide here, such that the latter was skipped in the respective evaluations. Still, the same experiments can be repeated with different weight estimation techniques such that both approaches will not be equivalent.

Second Experiment: Improvement from Dynamic Information?

After comparing the three different weight estimation techniques, the next evaluation particularly focuses on improvements that are obtained from integrating dynamic class information into the coupling process. This evaluation does not target the magnitude of the improvement, but only tries to evaluate whether it exists. As dynamic class information simplifies the decision problem, it is expected to observe an improvement.

With respect to the discussion of section 6.1, a reasonable evaluation criterion is to verify whether (6.13) holds. The respective results are given in table 6.3 where

	q_1		q_{all}	
	SVM	**RF**	**SVM**	**RF**
Softmax	0.019 ± 0.017	0.016 ± 0.016	0.072 ± 0.062	0.054 ± 0.052
Vote	0.019 ± 0.020	0.015 ± 0.016	0.062 ± 0.065	0.051 ± 0.054
ND	0.018 ± 0.019	0.015 ± 0.015	0.059 ± 0.061	0.051 ± 0.053
WLW	0.012 ± 0.028	0.015 ± 0.016	0.053 ± 0.065	0.050 ± 0.055
CombVote	0.018 ± 0.019	0.015 ± 0.016	0.061 ± 0.060	0.051 ± 0.053
NOV@	0.019 ± 0.019	0.015 ± 0.016	0.062 ± 0.061	0.051 ± 0.053
GVote	0.019 ± 0.020	0.015 ± 0.016	0.061 ± 0.064	0.051 ± 0.054
GND	0.018 ± 0.019	0.015 ± 0.015	0.059 ± 0.061	0.051 ± 0.053
GWLW	0.018 ± 0.019	0.015 ± 0.016	0.059 ± 0.061	0.050 ± 0.053
ET	0.018 ± 0.019	0.015 ± 0.016	0.060 ± 0.062	0.051 ± 0.054

TABLE 6.4: Improvements from evaluating (6.15) on all data sets. The given values are the average value and the standard deviation computed over all data sets.

for each classification algorithm, the number of data sets on which the average error decreased is given. Furthermore, each comparison was combined with a one-sided sign test that formulates the null hypothesis that there is no improvement, i.e. criterion (6.13) does not hold. Since the individual tests are not independent, a Bonferroni correction was applied as well. The respective result are also given in table 6.3.

Because without any exception all p-values are smaller than 0.001, integrating dynamic context information significantly reduces the error rate in almost all cases. This confirms the expectation, but still is an important result because it empirically supports the improvements in practical applications.

Third Experiment: Evaluating the Improvement

The previous result showed that dynamic class information significantly improves the prediction accuracy, even though it provides no information about the improvement's magnitude. With respect to the insights of section 6.1, two different sampling strategies generating the target sets are used: According to q_1 in (6.9) as well as using sampling q_{all} in (6.8). Here, the parameter was selected as $t = 1$ in the former case because it forms – together with using no dynamic class information, which is equivalent to sampling according to q_0 – both extreme cases of (6.9). The latter strategy samples the classes such that all sets with $2 \leq \ell \leq k$ elements are in total equally probable with probability $\frac{1}{k-1}$. The respective number of classes k per data set is given in table 6.1, too.

Both samplings were independently used to compute the respective empirical dynamic risks (6.6) as an approximation of (6.4). Still in general, these evaluations can be performed with any sampling strategy, i.e. a distribution q that defines the possible class sets with their respective probabilities. Similar to the deriving of criterion (6.13), it is reasonable to subtract the respective fusing method's base error rate $\mathcal{R}_{\text{emp}}(\mathsf{f} \mid D)$ from the computed evaluation metric to obtain a direct measure of the improvement

$$\mathsf{Imp}(\mathsf{f}_{\mathcal{M}} \mid q) \;\coloneqq\; \mathcal{R}_{\text{emp}}(\mathsf{f} \mid D) \;-\; \mathcal{R}_{\text{emp}}(\mathsf{f}_{\mathcal{M}} \mid q) \tag{6.15}$$

which measures the increase in accuracy obtained from integrating dynamic class information according to sampling by q in comparison with no dynamic context information. In a comprehensive way, these could be presented in multiple tables, e.g.

for each classifier and each evaluation metric reporting the respective value for each data set (row) and each fusing method (column).

Generally, comparing algorithms by averaging their accuracies on different data sets is criticized as unreasonable [Demšar 2006], but still it is not completely uncommon in practice to summarize results. Besides this, here the situation is slightly different because the improvement computed using (6.15) can directly be interpreted as the expected increase in classification accuracy, which is at least partially more comparable than the classification accuracies themselves. Thus, the results are presented in this summarized way for an improved clarity in table 6.4. Here in summary, the average classification accuracy is improved by roughly 2 % (SVM) and 1.5 % (RF) as well as 6 % to 7 % (SVM) and 5 % (RF) in case of sampling according to q_1 and q_{all}, respectively. In most cases, the standard deviations are comparable to the improvements themselves. Furthermore on average, most fusing methods improve comparably from the dynamic context information.

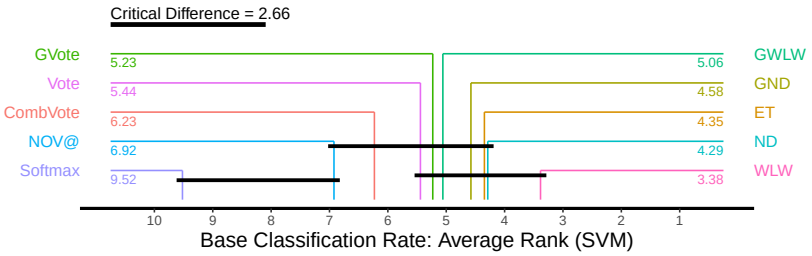
Fourth Experiment: Comparison

The previous two evaluations confirmed a significant improvement from integrating dynamic class information as well as with respect to average improvements, all fusing methods are roughly comparable. Here, the following part focuses on comparing the fusing methods in full detail.

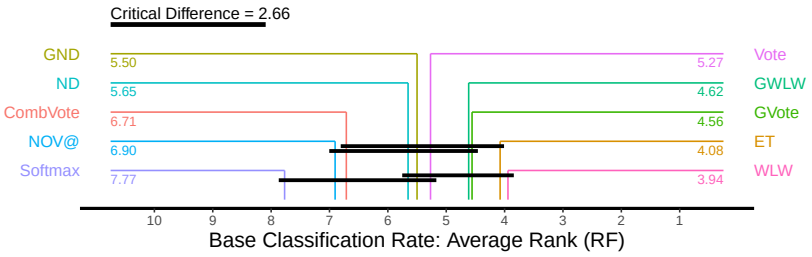
The first evaluation compares all approaches by their respective classification accuracies without using any dynamic class information. Following the same comparison method, the fusing techniques are ranked by their accuracies such that higher values are ranked first. Thus, a rank on each data set is available, which thereafter can be used to compute average ranks. The latter can be combined with the same hypothesis testing done before in chapter 3. First by using the non-parametric Friedman test and if the null hypothesis is rejected, by applying the post hoc Nemenyi test, both using the default $\alpha = 0.05$ significance level. The respective critical difference diagrams are presented in figure 6.1, while the corresponding results are available in full detail in the supplementary material.

Here, two interesting observations are possible. First, fusing based on the one-vs-all reduction (Softmax) yields the worst results, i.e. highest rank. On the one hand, this confirms the respective reference results, as presented in full detail in chapter 2, but, on the other hand, may point to the question whether large-scale state-of-the-art models like neural networks based on the one-vs-all reduction can improve their recognition accuracies by using them in combination with other fusing methods based on different reductions. Here, reference results are rare as summarized in section 5.5. Additionally, the fusing methods combining one-vs-all and one-vs-one predictions (CombVote and NOV@) also lead to relatively large average ranks. This slightly contradicts with their introductory works that report improved results.

The second important observation is that the best results in any case are obtained using fusing methods that only use pairwise one-vs-one predictions. This is remarkable because current reference results identified the non-competence problem as one of the major drawbacks in decomposition-based classification and in particular the one-vs-one reduction. Empirically, this disadvantage is *not* confirmed because even the opposite is observed here instead. In particular, the differences between the pairwise coupling approaches and their generalized counterparts were not detected as significantly different. Still, these results do not directly mean that existing statements emphasizing the non-competence problem's relevance are incorrect. Instead, the correct estimation of the weights that alleviate the influence of the incompetent



(a) Nemenyi test results for support vector machines.



(b) Nemenyi test results for random forests.

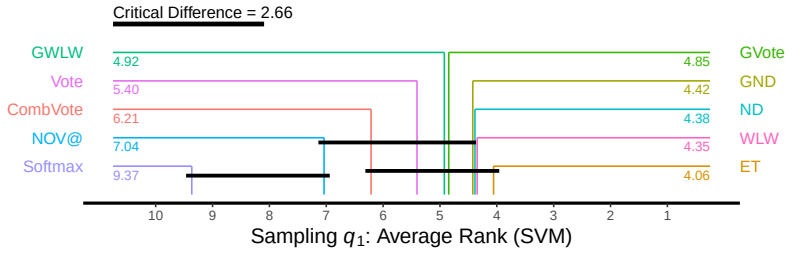
FIGURE 6.1: Average ranks of the base classification rates for both classifiers in combination with a Nemenyi test.

classifiers is the challenging part. The appropriate weights are the pairwise posterior probabilities $\mathsf{w}_{i,j} = P(\mathsf{y} \in \{i, j\} \mid \mathsf{x})$, as discussed in full detail in sections 2.3 and 5.2. Accurately estimating these is a more complex task than solving the classification problem itself. This is also confirmed by the discussion on classifier calibration in section 2.2 and chapter 3, respectively.

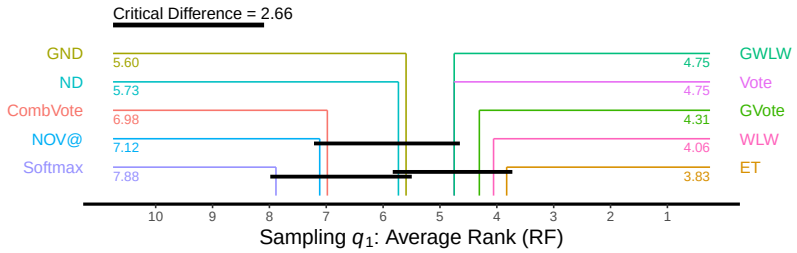
Hence, on the one hand identifying the non-competence problem as a major issue in the one-vs-one decomposition is correct, but, on the other hand, adequately solving it is mostly impossible. Thus, observing empirical results where pairwise coupling approaches outperform their generalized counterparts does at first only mean that the used weights insufficiently approximate the unknown pairwise posterior probabilities.

Besides these evaluations, particular relevant are accuracy results that are obtained in combination *with* supplying dynamic class information. As these are not directly contained in the reference data sets, this evaluation requires sampling strategies. Similar to the previous part, both samplings q_1 according to (6.9) as well as q_{all} in (6.8) were used to compute the empirical dynamic risk (6.6) as an approximation of (6.4).

In full analogy to figure 6.1, the fusing techniques are ranked per data set. Thereafter, average ranks over all data sets are computed and combined with the Friedman and Nemenyi test, respectively, at the same $\alpha = 0.05$ significance level as before. These are illustrated in figures 6.2 and 6.3 where a few interesting observations are possible. First, both fusing methods based on the one-vs-all reduction still yield the worst results. Thus, also in dynamic contexts there are significant performance gaps, which amplifies the previously discussed drawbacks of the one-vs-all decomposition. Furthermore, the best results are obtained using pairwise coupling based on evidence theory. In particular, integrating the weights also does not increase the classification accuracy.

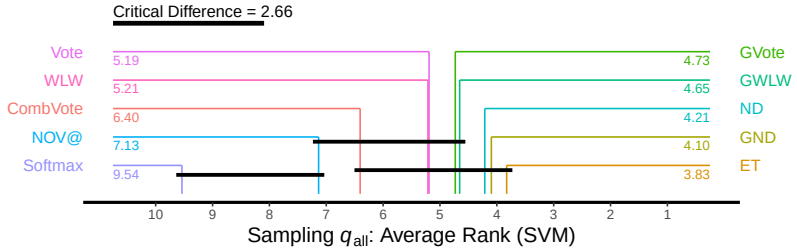


(a) Nemenyi test results for support vector machines.

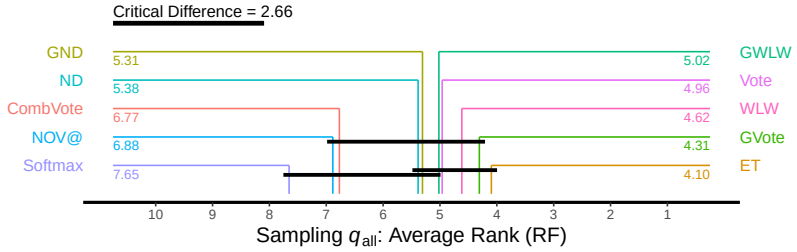


(b) Nemenyi test results for random forests.

FIGURE 6.2: Average ranks of the empirical dynamic risks according to sampling q_1 for both classifiers in combination with a Nemenyi test.



(a) Nemenyi test results for support vector machines.



(b) Nemenyi test results for random forests.

FIGURE 6.3: Average ranks of the empirical dynamic risks according to sampling q_{all} for both classifiers in combination with a Nemenyi test.

	Name	Samples (r)	Classes (k)	Size [MB]
1	AgrilPlant	3000	10	187
2	Animal	26126	10	601
3	Monkey	1370	10	581
4	Swedish Leaf	1125	15	3397
5	Tropic	14376	52	1976

TABLE 6.5: Overview of the deep learning data sets.

These observations mainly coincide between figures 6.1 and 6.2 as well as 6.3. Still, now the best results in any case are obtained using the evidence-theoretic pairwise coupling approach (ET). Besides the one-vs-all decomposition-based techniques, the approaches were mainly not detected to be significantly different. It should be emphasized that the statistical tests are not independent to the preceding ones because the data and the statistical models are the same in both cases. Still, this does not influence the main conclusions.

Besides this, it is a remarkable observation that for both classifiers, the rankings of the fusing techniques are very well comparable if the possible target set changes. This means that integrating dynamic information into the fusing process does not change the rank order of the respective fusing methods, at least according to the observations in the presented study.

6.2.3 Deep Learning Data

The previous study evaluated 26 reference data sets using support vector machines and random forests. For practical applications, deep learning-based approaches are particular relevant. First, because often deep learning neural networks outperform other, more conventional data mining and machine learning algorithms such that improvements from integrating dynamic class information are similarly interesting and relevant. Second, because transferring the presented techniques to large-scale deep learning models is a challenging task that was discussed in full detail in section 5.5. Finally and most importantly, the previous results confirmed that the one-vs-one decomposition often outperforms the one-vs-all reduction. Since deep learning classification models almost always use a softmax function – i.e. a jointly optimized one-vs-all prediction – it is a particular interesting question how one-vs-one deep learning models perform in comparison to those using a softmax prediction. For this reasoning, also five deep learning benchmark data sets are similarly evaluated.

Because there are no direct reference results available, the five selected data sets[2] were obtained from the most related existing work [Pawara et al. 2020] whose most important properties are summarized in table 6.5. The authors train two deep convolutional neural networks using the state-of-the-art network structures Inception-V3 [Szegedy et al. 2016] and ResNet-50 [He et al. 2016], which are used here as well to maintain a good comparability to the respective reference results. For the same reasoning, the neural networks were trained using the same parameterization, even though many of the involved training steps are randomized. Similarly, both network architectures were trained from scratch as well as using pretrained weights based on the ImageNet data[3] [Deng et al. 2009].

In particular, the neural networks were trained for 200 epochs (i.e. complete iterations over the training data) where a randomized 80 % / 20 % split was used to

[2]The data sets are publicly available at: https://www.ai.rug.nl/~p.pawara/dataset.php
[3]https://www.image-net.org/

create training and test data, respectively. As there are no validation steps involved for hyperparameter estimation, further splitting of the data was not required. The running loss and accuracy during training both models with pretrained weights as well as the developments of the test data loss and accuracy are illustrated in figures 6.4 and 6.5, respectively. The remarkable decreases in the losses and increases in the accuracies, respectively, after 50 training epochs are explainable by the learning rate decrease from 0.001 to 0.0001.

During training, a data augmentation using the same parameterization was used that randomly shifts or horizontally flips each training image during each epoch. Data augmentation was disabled on the test images such that all predictions remain deterministic and thus comparable. After finishing the 200 training epochs, the one-vs-one decision functions were trained on each network by fixing all neural network's parameters and iteratively creating the one-vs-one layer, as presented in full detail in section 5.5. Here, each decision function was trained for additional ten epochs on the respective two-class data set, including the same data augmentation steps as before. The correction classifiers were not trained because of the large computational complexity involved.

Using the presented procedure, the test data predictions for all fusing methods were computed. Because there are many randomization steps involved, the whole evaluation was iterated ten times such that for each fusing type's accuracy, mean value and standard deviation can be computed over the ten iterations. Consequently, the methods are compared using their average values. Thereafter, the performed experiments are analogous to those of subsection 6.2.2.

First Experiment: Weight Estimation

First, the weight estimation techniques are compared in combination with the two neural network architectures as well as initialization variants. Because the weight estimation based on the correcting classifiers was omitted due to its large complexity, only the two approaches based one the one-vs-all (Weight1vA) and one-vs-one decomposition (Weight1v1) are compared. The respective average test error accuracies per data set were used to rank the two weight estimation techniques, following the same evaluation procedure as before. The average ranks are presented in table 6.6.

Here, follow-up significance testing is generally possible, however the number of data sets is too low for reasonable significance testing. Besides this, most importantly the weights based on the one-vs-all decomposition yield superior accuracy in most cases. Consequently, the observation here is contrary to the ones in subsection 6.2.2, still this is also explainable because the one-vs-one decision functions are not independent as in the previous study. Instead, they depend on the training of the one-vs-all decision functions. Based on these observations, the weights are estimated using the one-vs-all decision functions in the following experiments.

Second Experiment: Improvement from Dynamic Classification?

The next evaluation focuses on the improvements obtained from integrating dynamic class information. Again, the accuracy increase by sampling the class set according to (6.9) and comparing the respective differences in classification accuracy as given by (6.15) yields the improvements presented in table 6.7. Sampling (6.8) was not applied in this particular evaluation because its exponential complexity makes it computationally intractable on a data set consisting of 52 classes. Still, sampling q_1 is sufficient to reasonably lower bound a possible improvement.

Algorithm	Method	Weight1vA	Weight1v1
Inception-V3 (pretrained)	GVote	1.2	1.8
	GND	1.2	1.8
	GWLW	1.5	1.5
	GET	1.2	1.8
Inception-V3 (not pretrained)	GVote	1.0	2.0
	GND	1.0	2.0
	GWLW	1.8	1.2
	GET	1.0	2.0
ResNet-50 (pretrained)	GVote	1.0	2.0
	GND	1.0	2.0
	GWLW	1.3	1.7
	GET	1.2	1.8
ResNet-50 (not pretrained)	GVote	1.0	2.0
	GND	1.0	2.0
	GWLW	1.7	1.3
	GET	1.0	2.0

TABLE 6.6: Average ranks of the two weight estimation techniques for each network structure and respective fusing method.

Here, two important observations are possible. First, there is an accuracy improvement of 0.1 % to 0.5 % and second, the pretrained models improve less than the non-pretrained ones. Still, this is explainable from the fact that pretrained models yield higher accuracies on each data set, as illustrated in figure 6.5. Similar to comparing the weight estimation techniques, further significance testing was not applied because the number of data sets is too low.

Most importantly, dynamic classification according to sampling q_1 in any case improves the accuracy on all five data sets, all models and all fusing techniques. The only exception is the pretrained Inception-V3 model where the respective classification accuracy is not increased for five fusing methods (Vote, ND, WLW, GWLW and ET).

Comparing these results based on deep learning models to those of support vector machines and random forests as presented before, the absolute improvements in the latter case are an order of magnitude larger. The first possible explanation for this behavior is similar to the reduced improvement of the pretrained models: the better the base classification task is solved, the smaller the possible improvements are expected to be. Still, there is a further, less obvious difference with respect to the fusing techniques based on the one-vs-one decomposition. In contrast to subsection 6.2.2, they still depend on the previous one-vs-all training. Thus, the generally superior flexibility with respect to dynamic classification contexts is limited because even though the individual decision functions are trained on two-class data only, the most parts of the neural network still depend on all data such that the flexibility is lost besides the final layer. For this reason, differences between the results from subsection 6.2.2 and those based on deep neural networks are explainable.

Third Experiment: Comparison

The previous evaluation analyzed the improvement of integrating dynamic class information into the different fusing techniques based on deep neural networks. Still, the performed evaluations did not compare the methods to each other.

	Inception-V3		ResNet-50	
	pretrained	not pretrained	pretrained	not pretrained
Softmax	0.0011 ± 0.0011	0.0024 ± 0.0023	0.0017 ± 0.0015	0.0034 ± 0.0029
Vote	0.0011 ± 0.0011	0.0023 ± 0.0020	0.0018 ± 0.0013	0.0038 ± 0.0033
ND	0.0012 ± 0.0011	0.0025 ± 0.0023	0.0018 ± 0.0016	0.0036 ± 0.0029
WLW	0.0012 ± 0.0011	0.0025 ± 0.0023	0.0017 ± 0.0015	0.0036 ± 0.0030
CombVote	0.0012 ± 0.0011	0.0025 ± 0.0023	0.0017 ± 0.0015	0.0036 ± 0.0030
NOV@	0.0012 ± 0.0011	0.0025 ± 0.0023	0.0016 ± 0.0015	0.0036 ± 0.0031
GVote	0.0011 ± 0.0010	0.0024 ± 0.0023	0.0017 ± 0.0015	0.0035 ± 0.0029
GND	0.0011 ± 0.0011	0.0024 ± 0.0023	0.0017 ± 0.0015	0.0034 ± 0.0029
GWLW	0.0012 ± 0.0011	0.0024 ± 0.0021	0.0018 ± 0.0016	0.0036 ± 0.0029
GET	0.0011 ± 0.0011	0.0024 ± 0.0022	0.0016 ± 0.0014	0.0034 ± 0.0029
ET	0.0013 ± 0.0011	0.0025 ± 0.0022	0.0018 ± 0.0016	0.0038 ± 0.0031

TABLE 6.7: Improvements from evaluating (6.15) on all deep learning data sets. The given values are the average value and the standard deviation computed over the five data sets.

Here, in theory critical differences diagrams could be computed. However as before, the number of data sets remains too low to reasonably apply significance tests based on critical differences of rank-transformed scores. Additionally, there are almost no average ranks above the critical difference of 6.75 in this particular setting (five data sets and eleven methods).

Because critical differences diagrams are not reasonable, instead the accuracy per data set and each fusing type is computed. As the evaluation was iterated ten times to alleviate effects of random influences, the given values are also averages over the ten iterations per data set. The corresponding accuracy statistics for both pretrained network structures are presented in table 6.8. Here, a few interesting observations are possible. With respect to both pretrained network structures, GET, GVote and Softmax yield the best results, while the worst ones are observed by approaches based on the one-vs-one decomposition. In particular for the pretrained Inception-V3 network structure, ET and ND yield the worst results, as well as Vote for ResNet-50.

With respect to the non-pretrained models, the results are slightly different. In full detail they can be summarized similarly to table 6.8, still the respective table is omitted here and available in the supplementary material. Most importantly for the Inception-V3 network structure, Softmax, GET and GND yield the best results, while for ResNet-50 these are observed with GND, GET, and GVote. In both cases GWLW yields the worst results followed by ND (in case of Inception-V3) and Vote (for ResNet-50), respectively.

Furthermore, it is particularly interesting to compare these results to the situation where dynamic class information according to sampling q_1 is integrated as well. The respective accuracy statistics are presented in table 6.8 for both pretrained network structures, while those of the non-pretrained models are analogously available in the supplementary material. Most importantly, comparing the fusing methods based on their average ranks of the different approaches yields similar results, the ordering of the ranks is almost identical. This confirms the previous observations of applying the same algorithms in combination with support vector machines and random forests. In particular, dynamic class information improves the classification accuracy, but at most seems to have a minor influence on the methods' orderings.

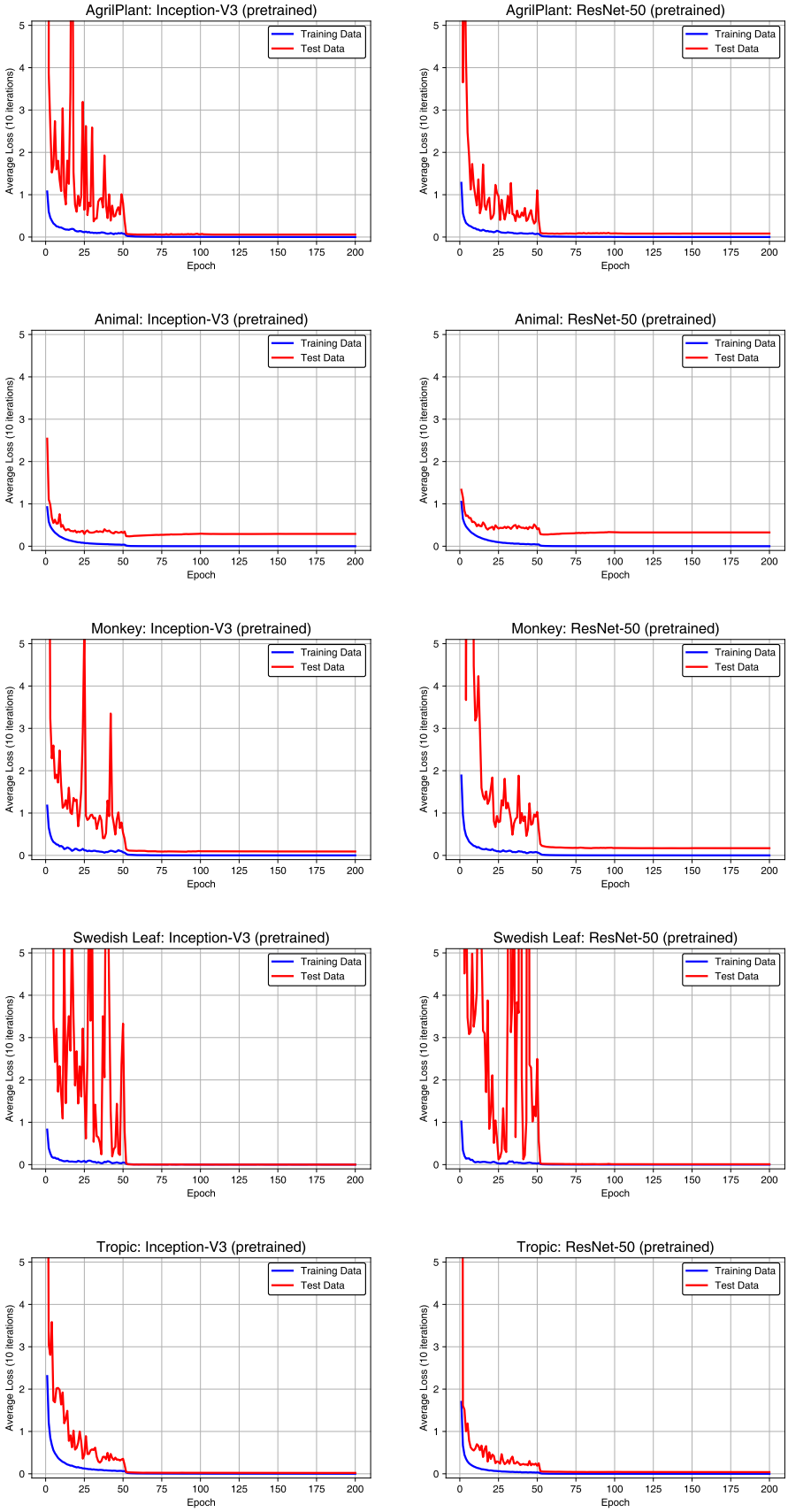


FIGURE 6.4: Training and test data losses of the deep neural networks (pretrained) on each of the five reference data sets.

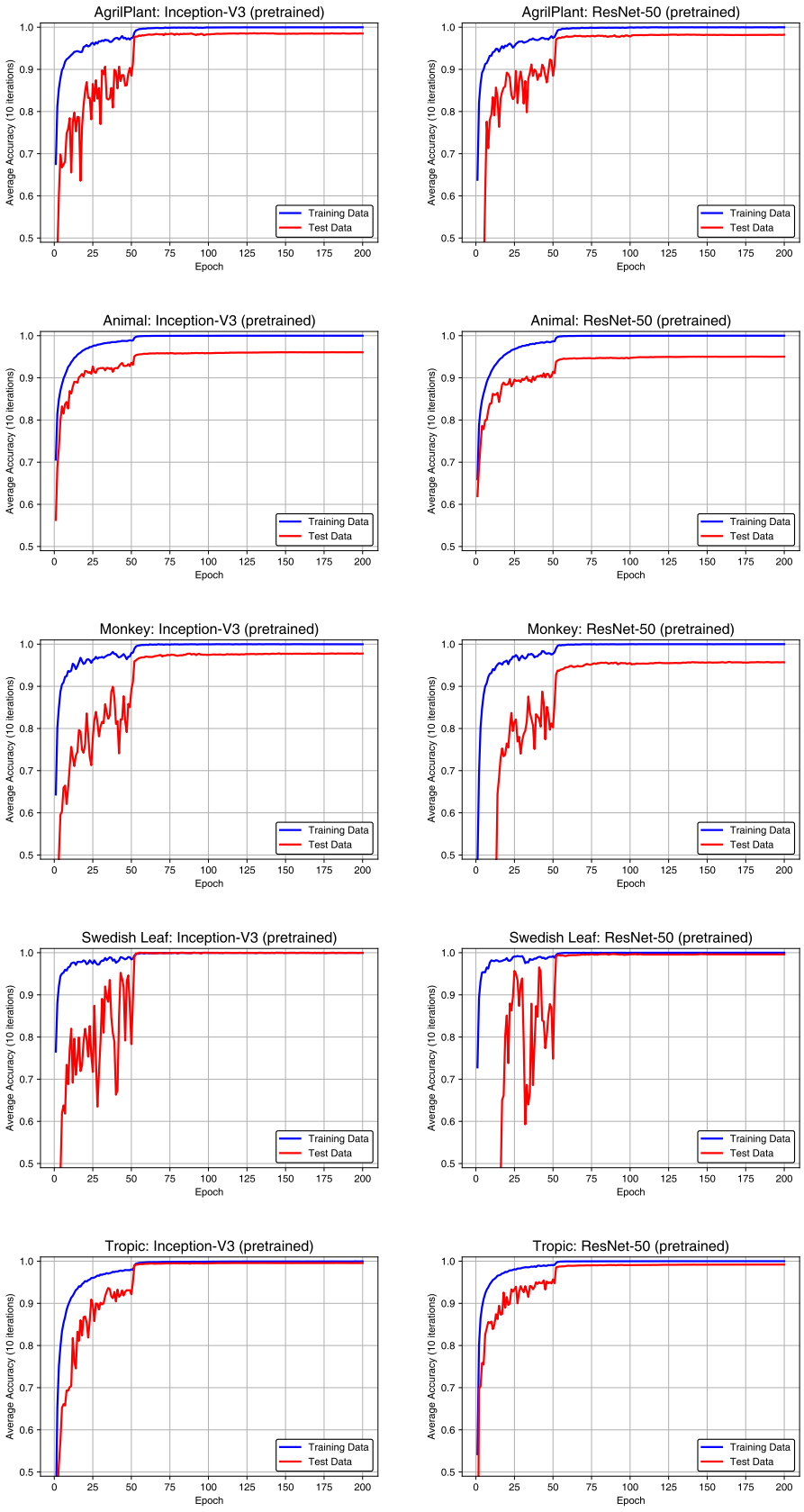


FIGURE 6.5: Training and test data accuracies of the deep neural networks (pretrained) on each of the five reference data sets.

		Softmax	Vote	ND	WLW	CombVote	NOV@	GVote	GND	GWLW	GET	ET
Inception-V3	AgrilPlant	0.9850000 (3)	0.9826667 (7.5)	0.9826667 (7.5)	0.9823333 (11)	0.9848333 (6)	0.9850000 (3)	0.9850000 (3)	0.9850000 (3)	0.9825000 (9)	0.9850000 (3)	0.9823333 (10)
	Animal	0.9606008 (4.5)	0.9605243 (7)	0.9603712 (11)	0.9604286 (8.5)	0.9605817 (6)	0.9606200 (3)	0.9606582 (1.5)	0.9606008 (4.5)	0.9604286 (8.5)	0.9606582 (1.5)	0.9604095 (10)
	Monkey	0.9777372 (1.5)	0.9729927 (7)	0.9715328 (10)	0.9718978 (9)	0.9759124 (6)	0.9770073 (5)	0.9773723 (3.5)	0.9773723 (3.5)	0.9715328 (11)	0.9777372 (1.5)	0.9722628 (8)
	Swedish Leaf	0.9995556 (8.5)	1.0000000 (3)	1.0000000 (3)	1.0000000 (3)	0.9995556 (8.5)	0.9995556 (8.5)	0.9995556 (8.5)	0.9995556 (8.5)	1.0000000 (3)	0.9995556 (8.5)	1.0000000 (3)
	Tropic	0.9955494 (3.5)	0.9935675 (10)	0.9938456 (9)	0.9938804 (7)	0.9953755 (6)	0.9955146 (5)	0.9955841 (1.5)	0.9955494 (3.5)	0.9938804 (8)	0.9955841 (1.5)	0.9935327 (11)
	Avg	0.9836886 (4.2)	0.9819502 (6.9)	0.9816833 (8.1)	0.9817080 (7.7)	0.9832517 (6.5)	0.9835395 (4.9)	0.9836340 (3.6)	0.9836156 (4.6)	0.9816684 (7.9)	0.9837070 (3.2)	0.9817077 (8.4)
	StdDev	0.0155131 (2.64)	0.0158159 (2.51)	0.0161371 (3.13)	0.0160640 (2.99)	0.0156772 (1.12)	0.0155726 (2.25)	0.0155343 (2.88)	0.0155489 (2.25)	0.0161224 (2.97)	0.0154984 (3.03)	0.0159503 (3.21)
ResNet-50	AgrilPlant	0.9821667 (3)	0.9785000 (11)	0.9801667 (7.5)	0.9798333 (9)	0.9820000 (6)	0.9821667 (3)	0.9821667 (3)	0.9821667 (3)	0.9801667 (7.5)	0.9821667 (3)	0.9795000 (10)
	Animal	0.9502488 (3)	0.9490050 (8.5)	0.9490624 (7)	0.9489476 (10)	0.9501148 (6)	0.9501914 (4)	0.9503062 (1)	0.9501722 (5)	0.9488519 (11)	0.9502679 (2)	0.9490050 (8.5)
	Monkey	0.9569343 (3.5)	0.9521898 (7)	0.9492701 (11)	0.9507299 (9)	0.9554745 (6)	0.9572993 (1.5)	0.9569343 (3.5)	0.9562044 (5)	0.9500000 (10)	0.9572993 (1.5)	0.9510949 (8)
	Swedish Leaf	0.9960000 (5.5)	0.9955556 (11)	0.9960000 (9.5)	0.9964444 (1.5)	0.9960000 (5.5)	0.9960000 (5.5)	0.9960000 (5.5)	0.9960000 (5.5)	0.9960000 (9.5)	0.9960000 (5.5)	0.9964444 (1.5)
	Tropic	0.9920028 (1)	0.9825104 (11)	0.9856050 (8)	0.9857789 (7)	0.9912726 (6)	0.9918985 (4.5)	0.9919680 (2.5)	0.9918985 (4.5)	0.9854312 (9)	0.9919680 (2.5)	0.9840056 (10)
	Avg	0.9754705 (3.2)	0.9715521 (9.7)	0.9720208 (8.6)	0.9723468 (7.3)	0.9749724 (5.9)	0.9755112 (3.7)	0.9754750 (3.1)	0.9752883 (4.6)	0.9720899 (9.4)	0.9755404 (2.9)	0.9720100 (7.6)
	StdDev	0.0207325 (1.6)	0.0201730 (1.86)	0.0216251 (1.64)	0.0214008 (3.42)	0.0209482 (0.224)	0.0206482 (1.52)	0.0207081 (1.64)	0.0208997 (0.962)	0.0214644 (1.29)	0.0206386 (1.56)	0.0209983 (3.52)

TABLE 6.8: Average classification accuracy per data set and fusing method based on the pretrained network architectures.

		Softmax	Vote	ND	WLW	CombVote	NOV@	GVote	GND	GWLW	GET	ET
Inception-V3	AgrilPlant	0.9864760 (3)	0.9841976 (9)	0.9843447 (7)	0.9839750 (11)	0.9863093 (6)	0.9864760 (3)	0.9864760 (3)	0.9864760 (3)	0.9842157 (8)	0.9864760 (3)	0.9839754 (10)
	Animal	0.9630731 (4)	0.9628806 (7)	0.9628233 (10)	0.9628753 (8)	0.9630415 (6)	0.9630772 (3)	0.9631176 (1)	0.9630647 (5)	0.9628484 (9)	0.9631111 (2)	0.9628231 (11)
	Monkey	0.9793563 (1)	0.9746580 (7)	0.9733652 (11)	0.9737741 (9)	0.9779835 (6)	0.9789937 (4)	0.9787478 (5)	0.9790317 (3)	0.9734035 (10)	0.9793152 (2)	0.9743757 (8)
	Swedish Leaf	0.9995875 (8.5)	1.0000000 (3)	1.0000000 (3)	1.0000000 (3)	0.9995875 (8.5)	0.9995875 (8.5)	0.9995875 (8.5)	0.9995875 (8.5)	1.0000000 (3)	0.9995875 (8.5)	1.0000000 (3)
	Tropic	0.9956067 (4)	0.9935929 (11)	0.9939179 (9)	0.9939248 (8)	0.9954328 (6)	0.9955699 (5)	0.9956401 (1.5)	0.9956073 (3)	0.9939513 (7)	0.9956401 (1.5)	0.9936270 (10)
	Avg	0.9848199 (4.1)	0.9830658 (7.4)	0.9828902 (8)	0.9829098 (7.8)	0.9844709 (6.5)	0.9847409 (4.7)	0.9847138 (3.8)	0.9847534 (4.5)	0.9828838 (7.4)	0.9848259 (3.4)	0.9829602 (8.4)
	StdDev	0.0144909 (2.75)	0.0148071 (2.97)	0.0150821 (3.16)	0.0149949 (2.95)	0.0146062 (1.12)	0.0145176 (2.28)	0.0145405 (3.05)	0.0145255 (2.4)	0.0150709 (2.7)	0.0144868 (2.9)	0.0148691 (3.21)
ResNet-50	AgrilPlant	0.9838463 (3)	0.9806065 (11)	0.9819953 (8)	0.9816811 (9)	0.9836805 (6)	0.9838278 (5)	0.9841236 (1)	0.9838463 (3)	0.9819958 (7)	0.9838463 (3)	0.9812356 (10)
	Animal	0.9534797 (2)	0.9522598 (9)	0.9523131 (7)	0.9522353 (10)	0.9533450 (6)	0.9534292 (4)	0.9534848 (1)	0.9534206 (5)	0.9521589 (11)	0.9534714 (3)	0.9522904 (8)
	Monkey	0.9599876 (3)	0.9548759 (7)	0.9528908 (11)	0.9537881 (9)	0.9586974 (6)	0.9601966 (1)	0.9599842 (4)	0.9592602 (5)	0.9534224 (10)	0.9601922 (2)	0.9547135 (8)
	Swedish Leaf	0.9962863 (4.5)	0.9959686 (11)	0.9962535 (9.5)	0.9965720 (2)	0.9962538 (7.5)	0.9962538 (7.5)	0.9962863 (4.5)	0.9962863 (4.5)	0.9962535 (9.5)	0.9962863 (4.5)	0.9966351 (1)
	Tropic	0.9920997 (1)	0.9828433 (11)	0.9857757 (8)	0.9859611 (7)	0.9913722 (6)	0.9919981 (5)	0.9920670 (3)	0.9919995 (4)	0.9856005 (9)	0.9920745 (2)	0.9841817 (10)
	Avg	0.9771399 (2.7)	0.9733108 (9.8)	0.9738457 (8.7)	0.9740475 (7.4)	0.9766698 (6.3)	0.9771411 (4.5)	0.9771892 (2.7)	0.9769626 (4.3)	0.9738862 (9.3)	0.9771742 (2.9)	0.9738113 (7.4)
	StdDev	0.0192961 (1.3)	0.0189767 (1.79)	0.0200849 (1.57)	0.0199611 (3.21)	0.0194673 (0.671)	0.0192361 (2.35)	0.0193135 (1.64)	0.0194583 (0.837)	0.0199633 (1.48)	0.0192485 (1.02)	0.0194388 (3.71)

TABLE 6.9: Average classification accuracy with dynamic class information according to sampling q_1 per data set and fusing method based on the pretrained network architectures.

Besides this, it should be noted that in many cases one of the generalized pairwise coupling approaches slightly outperforms the default softmax prediction. This is a remarkable observation as it points to potential further improvements in neural network prediction accuracies. Still, the involved weights are computed from the one-vs-all probabilities that are also used by the softmax prediction itself.

Nevertheless, it is important to conclude that differences between the fusing methods are always small. This in particular holds for the situations in which one of the one-vs-one approaches yields the best results. The maximum overall accuracy difference between two approaches for a fixed model is ≈ 0.022, i.e. approximately 2.2 %. In particular, this was observed between Vote and GVote during one iteration evaluating the non-pretrained ResNet-50 model.

Potential explanations for these observations are two fold. First, the classification accuracies of the models are always very high. As a matter of fact, there is less room for differences between the fusing techniques. Even though the decision problems involved in the one-vs-one decomposition are usually easier to solve than those of the one-vs-all reduction, this issue can be less relevant if both, the characteristic of the problem and the learning capacity of the classification algorithm, allow a highly accurate solving of it even using the one-vs-all decomposition. Besides this, the second explanation is that the model's one-vs-one decision functions depend on the previous training of the one-vs-all decomposition such they are less independent as they would be if a quadratic number of fully independent networks were trained. Still, this is primarily of theoretic interest because training and deploying these requires too many computational resources.

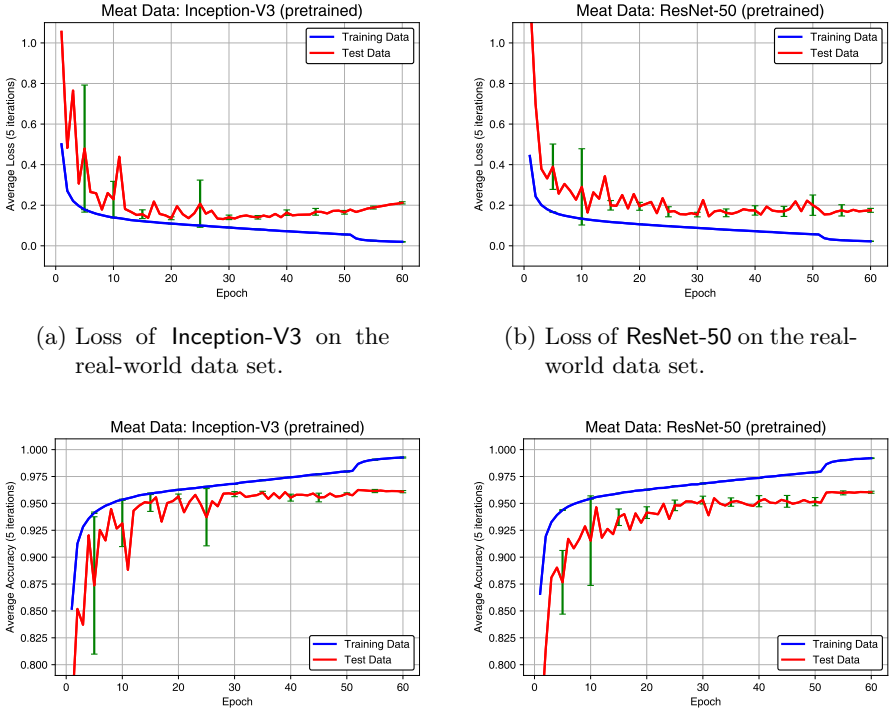
However, it is an interesting question for further research how to ideally train one-vs-one networks and whether these can outperform one-vs-all softmax models. Besides fixing the whole network architecture as presented in section 5.5, an alternative strategy is to similarly train each one-vs-one decision function, but update all coefficients in the model during each training. Clearly, doing so will result in conflicting parameter updates. Training a one-vs-one function modifies the respective values that were optimized during all previous trainings. A straightforward choice to alleviate this problem is an iterative training procedure where each one-vs-one function is trained multiple times. Still, possible termination criteria remain at least unclear.

6.3 Real-World Application

The last section 6.2 evaluated and compared the different approaches that support the integration of dynamic class information into the aggregation or fusing phase of decomposition-based classification. Consequently, it is both relevant and interesting to transfer these results to real processing environments where dynamic class information is available or at least expected to be.

Here, meat processing in dissection factories was identified as a potential use case for two reasons, as already discussed before in more detail. Not only machine learning enables to automatize human classification tasks, the process is also controlled by dissection lists that serve as dynamic class information.

The main problem regarding practical applications in any of these potential factories is that the conditions are not particularly research-friendly. There are many potential customers for a working automatization solution, still they mostly expected a proof-of-concept reference. On the other hand, creating the latter requires a respective processing factory that allows the installation of a prototype device to collect data



(a) Loss of Inception-V3 on the real-world data set.

(b) Loss of ResNet-50 on the real-world data set.

(c) Accuracy of Inception-V3 on the real-world data set.

(d) Accuracy of ResNet-50 on the real-world data set.

FIGURE 6.6: Training and test data losses as well as accuracies, respectively, of the deep neural networks on the real-world data set.

that can be used to train and deploy a machine learning model solving the classification task.

Luckily, a reference factory was found in Meckenheim, Germany, where a prototype installation was possible. Still, the conditions in the respective factory are not optimal for the prototype development because the installation of the device was only possible on a conveyor belt where most times the transported raw meat products were not the high-worth main parts. Additionally, it was not possible to receive online information about the produced articles. Nevertheless, the factory was still chosen for a prototype installation because for explained reasoning there was no better alternative available and it was sufficient for a proof-of-concept development. In this regard, the following task was to collect training data using a digital camera system, create a reference data set, solve the respective classification task with a high accuracy and – with particular focus on dynamic classification – analyze improvements from restricting the target set during the fusing process.

From August 2018 until May 2019, in total 136009 training images were collected. Because no online identification data were available to the system, the images were manually sorted into 57 classes by human experts. Furthermore based on the physical properties of the respective classes, there are three main article groups such that each is a member of one, two or all three groups. In particular, the first group refers to the front part and contains 30 different articles. Similarly, the second group refers to the middle part and consists 23 products, while the rear part comprises 28 classes.

As the sets are obviously not disjoint, the unions of two sets contains fewer classes than summing the individual cardinalities. In particular, front and middle part together consist of 43 possible products, while the two other unions (i.e. front and rear

Network	Method	Weight1vA	Weight1v1
Inception-V3 (pretrained)	GVote	0.9611 ± 0.0011	0.9163 ± 0.0020
	GND	0.9614 ± 0.0011	0.9566 ± 0.0011
	GWLW	0.9557 ± 0.0013	0.9542 ± 0.0013
	GET	0.9610 ± 0.0011	0.9307 ± 0.0012
ResNet-50 (pretrained)	GVote	0.9604 ± 0.0010	0.9286 ± 0.0016
	GND	0.9614 ± 0.0009	0.9583 ± 0.0013
	GWLW	0.9574 ± 0.0013	0.9575 ± 0.0012
	GET	0.9604 ± 0.0010	0.9413 ± 0.0017

TABLE 6.10: Average accuracy and standard deviation computed over the five iterations of the two weight estimation techniques for each network structure and respective fusing method on the real-world data set.

as well as middle and rear part) each contain 44 articles. Clearly, the union of all three groups yields the overall class set comprising all 57 products. As no online information about the possible class sets was available, these sets are a reasonable choice and thus the best available surrogate for dynamic class information. Besides this, sampling q_1 according to (6.9) in the same way as in both previous studies can be applied, while sampling according to (6.8) remains computationally intractable on 57 classes.

Similar to the last evaluation, both neural network architectures Inception-V3 and ResNet-50 were trained here, too. Still based on the previous results, only the pretrained models were used. Because there are even more images available, the number of epochs during one-vs-all training was reduced from 200 to 60. Similarly, the number of epochs used for each one-vs-one training was reduced from ten to three. Finally, also the number of overall iterations to alleviate the effect of random influences was reduced from ten to five. Besides being a reasonable selection based on previous observations, these modifications were made to keep the training runtime requirements feasible.

During each iteration, a randomized 80 % / 20 % split was used to create training and test data, respectively, which was combined with a similar randomized data augmentation as in the previous evaluation of the deep neural networks. The only small extension in the data augmentation was a randomized vertical flip of the images because the respective products could be observed in both orientations. The corresponding running losses and accuracies on the training data as well as the test data loss and accuracy, respectively, are illustrated in figure 6.6. Here, each value represents the average value of the five observations, while the error bars illustrate the respective standard deviations.

First Experiment: Weight Estimation

Following the same evaluation procedure as in section 6.2, first the two weight estimation techniques Weight1vA and Weight1v1 were compared. Due to its high computational effort at training time, the correction classifiers were not trained such that WeightCC was excluded from the comparison, too.

In contrast to both preceding weight comparisons, here the comparison was performed based on the respective accuracies instead of ranks. Applying the rank transformation on a single data set only is equivalent to directly compare the methods

	Inception-V3	ResNet-50
Softmax	0.000560 ± 0.000019	0.000570 ± 0.000010
Vote	0.001156 ± 0.000052	0.001102 ± 0.000068
ND	0.000541 ± 0.000015	0.000541 ± 0.000013
WLW	0.000648 ± 0.000086	0.000589 ± 0.000033
CombVote	0.000556 ± 0.000019	0.000559 ± 0.000008
NOV@	0.000559 ± 0.000022	0.000567 ± 0.000012
GVote	0.000557 ± 0.000025	0.000562 ± 0.000014
GND	0.000551 ± 0.000020	0.000554 ± 0.000009
GWLW	0.000542 ± 0.000018	0.000545 ± 0.000009
GET	0.000559 ± 0.000022	0.000566 ± 0.000015
ET	0.000796 ± 0.000073	0.000742 ± 0.000072

TABLE 6.11: Improvements from evaluating (6.15) on the real-world data set. Mean value and standard deviation are computed over the five iterations.

using the respective metric without applying the rank transformation at all, while the respective metric itself captures more information.

The corresponding results are presented in table 6.10. Similar to table 6.6, the weights based on the one-vs-all decomposition yield superior results here and thus were used in the following evaluations.

Second Experiment: Improvement from Dynamic Classification?

After comparing the different weight estimation techniques, particular relevant are the improvements yielded by integrating dynamic class information in the respective fusing methods. Applying sampling q_1 according to (6.9) is relatively straightforward, analogous to the similar evaluations in section 6.2 and presented in table 6.11.

Here, the absolute accuracy improvement is 0.05 % to 0.1 %, depending on the network structure and fusing method, respectively. Even though this is relatively small, it is still an improvement that remains an order of magnitude larger than the respective standard deviations, i.e. is highly unlikely to be caused by random effects only.

Besides this experiment, analyzing the improvement by integrating dynamic class information according to the article groups is particularly relevant. During each of the five iterations, additionally the dynamic classification accuracies with sampling the test data according to the respective class subset were computed. It should be noted that this the same as computing (6.6), together with the corresponding dynamic class set $\mathcal{M}$. The respective average accuracies and standard deviations are presented in table 6.12, which are computed over the five iterations. In the last column, the overall accuracy without any dynamic class information is given that can be used as a reference to check for an improvement. Alternatively by subtracting the base classification rate from the corresponding one computed using dynamic class information, the improvement could be evaluated directly. Still, here it is more useful to focus on the classification accuracy instead of the improvement because – in contrast to the previous evaluation on benchmark data – the former is more relevant for the actual application than the latter.

First, it is important to observe that on the front and middle group, there is a relatively large improvement between 0.3 % and 1 %, depending on the actual fusing method. On the other hand, the classification accuracy on the rear group is decreased

		Front	Middle	Rear	Front + Middle	Front + Rear	Middle + Rear	All
Inception-V3	Softmax	0.964 ± 0.001	0.970 ± 0.001	0.953 ± 0.001	0.962 ± 0.001	0.962 ± 0.001	0.954 ± 0.001	0.961 ± 0.001
	Vote	0.937 ± 0.002	0.956 ± 0.002	0.925 ± 0.001	0.928 ± 0.002	0.916 ± 0.002	0.914 ± 0.002	0.910 ± 0.003
	ND	0.961 ± 0.001	0.968 ± 0.001	0.951 ± 0.001	0.958 ± 0.001	0.958 ± 0.001	0.951 ± 0.001	0.957 ± 0.001
	WLW	0.956 ± 0.001	0.966 ± 0.001	0.945 ± 0.001	0.953 ± 0.001	0.949 ± 0.001	0.944 ± 0.001	0.947 ± 0.001
	CombVote	0.962 ± 0.001	0.968 ± 0.002	0.952 ± 0.002	0.960 ± 0.001	0.959 ± 0.001	0.952 ± 0.002	0.958 ± 0.001
	NOV@	0.964 ± 0.001	0.970 ± 0.001	0.953 ± 0.001	0.962 ± 0.001	0.962 ± 0.001	0.954 ± 0.001	0.961 ± 0.001
	GVote	0.964 ± 0.001	0.970 ± 0.001	0.953 ± 0.001	0.962 ± 0.001	0.962 ± 0.001	0.955 ± 0.001	0.961 ± 0.001
	GND	0.964 ± 0.001	0.970 ± 0.001	0.954 ± 0.001	0.962 ± 0.001	0.963 ± 0.001	0.955 ± 0.001	0.961 ± 0.001
	GWLW	0.960 ± 0.001	0.968 ± 0.001	0.950 ± 0.001	0.958 ± 0.001	0.957 ± 0.001	0.950 ± 0.002	0.956 ± 0.001
	GET	0.964 ± 0.001	0.970 ± 0.001	0.953 ± 0.001	0.962 ± 0.001	0.962 ± 0.001	0.954 ± 0.001	0.961 ± 0.001
	ET	0.948 ± 0.001	0.962 ± 0.002	0.937 ± 0.001	0.942 ± 0.001	0.935 ± 0.001	0.930 ± 0.001	0.931 ± 0.001
ResNet-50	Softmax	0.963 ± 0.001	0.969 ± 0.001	0.954 ± 0.001	0.961 ± 0.001	0.963 ± 0.001	0.954 ± 0.001	0.960 ± 0.001
	Vote	0.949 ± 0.002	0.964 ± 0.001	0.935 ± 0.002	0.940 ± 0.002	0.929 ± 0.002	0.926 ± 0.002	0.923 ± 0.002
	ND	0.964 ± 0.002	0.970 ± 0.001	0.954 ± 0.002	0.961 ± 0.002	0.960 ± 0.001	0.953 ± 0.002	0.958 ± 0.001
	WLW	0.962 ± 0.002	0.969 ± 0.001	0.952 ± 0.001	0.958 ± 0.002	0.956 ± 0.001	0.950 ± 0.001	0.954 ± 0.001
	CombVote	0.963 ± 0.001	0.969 ± 0.001	0.954 ± 0.001	0.960 ± 0.001	0.961 ± 0.001	0.953 ± 0.001	0.959 ± 0.001
	NOV@	0.964 ± 0.001	0.969 ± 0.001	0.954 ± 0.001	0.961 ± 0.001	0.963 ± 0.001	0.954 ± 0.001	0.961 ± 0.001
	GVote	0.964 ± 0.001	0.969 ± 0.001	0.954 ± 0.001	0.961 ± 0.001	0.963 ± 0.001	0.954 ± 0.001	0.960 ± 0.001
	GND	0.964 ± 0.001	0.970 ± 0.001	0.955 ± 0.001	0.962 ± 0.001	0.964 ± 0.001	0.955 ± 0.001	0.961 ± 0.001
	GWLW	0.963 ± 0.002	0.969 ± 0.001	0.953 ± 0.002	0.960 ± 0.002	0.960 ± 0.001	0.952 ± 0.001	0.957 ± 0.001
	GET	0.964 ± 0.001	0.969 ± 0.001	0.954 ± 0.001	0.961 ± 0.001	0.963 ± 0.001	0.954 ± 0.001	0.960 ± 0.001
	ET	0.956 ± 0.002	0.967 ± 0.001	0.945 ± 0.002	0.951 ± 0.002	0.945 ± 0.002	0.939 ± 0.002	0.941 ± 0.002

TABLE 6.12: Dynamic classification accuracies on the different article groups. Mean value and standard deviation are computed over the five iterations.

in comparison with the overall accuracy. This seems to be counterintuitive, still this is not caused by dynamic class information that harm the fusing process. Instead, the situation is more complex.

For a fixed model and fusing type, the classification accuracy or empirical risk is computed as the fraction of correctly (or incorrectly, respectively) classified instances. Extending this into a dynamic classification accuracy means that the fusing process is constrained according to the dynamic class information, but *also the data are sampled according to this class subset only.*

To analyze this issue in more detail, it is reasonable to group the data D according to the classes

$$D \;=\; \bigcup_{i=1}^{k} D_i \;=\; \bigcup_{i=1}^{k} \{(\mathsf{x}, \mathsf{y}) \in D \;:\; \mathsf{y} = i\} \tag{6.16}$$

and thereafter to compute the accuracies on each subset D_i only

$$\mathsf{Acc}(\mathsf{f}; D_i) \;=\; \frac{1}{|D_i|} \cdot \sum_{(\mathsf{x}_j, \mathsf{y}_j) \in D_i} \mathbb{1}(\mathsf{f}(\mathsf{x}_i) = \mathsf{y}_i) \tag{6.17}$$

to express the overall accuracy as their combination

$$\mathsf{Acc}(\mathsf{f}; D) \;=\; \sum_{i=1}^{k} \frac{|D_i|}{r} \cdot \mathsf{Acc}(\mathsf{f}; D_i) \tag{6.18}$$

weighted by the respective number of instances. Here, the class-specific accuracies $\mathsf{Acc}(\mathsf{f}; D_i)$ – usually called *sensitivity* in binary classification contexts – can differ. Consequently, the accuracy can be changed even if no dynamic class information is forwarded to the fusing step. This means that the accuracy computed on *all* classes is an inadequate estimator for the accuracy on a class subset only. If the computation of the dynamic classification accuracy respects *all* individual classes similarly (as in the case of sampling q_1), this is alleviated by the overall average. However, computing dynamic classification accruacies based on an relatively arbitrary selection of classes can increase the influence of those with low class-specific accuracy. Since this situation

		Front	Middle	Rear	Front + Middle	Front + Rear	Middle + Rear
Inception-V3	Softmax	0.954 ± 0.001	0.962 ± 0.001	0.944 ± 0.001	0.960 ± 0.001	0.955 ± 0.001	0.954 ± 0.001
	Vote	0.922 ± 0.002	0.932 ± 0.003	0.908 ± 0.002	0.920 ± 0.003	0.908 ± 0.002	0.906 ± 0.003
	ND	0.953 ± 0.001	0.960 ± 0.001	0.943 ± 0.001	0.958 ± 0.001	0.952 ± 0.001	0.951 ± 0.001
	WLW	0.946 ± 0.001	0.954 ± 0.002	0.934 ± 0.001	0.950 ± 0.001	0.942 ± 0.001	0.942 ± 0.001
	CombVote	0.953 ± 0.002	0.960 ± 0.002	0.943 ± 0.001	0.958 ± 0.001	0.953 ± 0.001	0.952 ± 0.002
	NOV@	0.955 ± 0.001	0.962 ± 0.001	0.944 ± 0.001	0.961 ± 0.001	0.956 ± 0.001	0.954 ± 0.001
	GVote	0.955 ± 0.001	0.962 ± 0.001	0.945 ± 0.001	0.961 ± 0.001	0.956 ± 0.001	0.954 ± 0.001
	GND	0.955 ± 0.001	0.963 ± 0.002	0.945 ± 0.001	0.961 ± 0.001	0.956 ± 0.001	0.955 ± 0.001
	GWLW	0.952 ± 0.001	0.959 ± 0.001	0.943 ± 0.001	0.957 ± 0.001	0.951 ± 0.001	0.950 ± 0.002
	GET	0.955 ± 0.001	0.962 ± 0.002	0.944 ± 0.001	0.961 ± 0.001	0.956 ± 0.001	0.954 ± 0.001
	ET	0.935 ± 0.001	0.944 ± 0.002	0.923 ± 0.001	0.937 ± 0.001	0.927 ± 0.001	0.926 ± 0.002
ResNet-50	Softmax	0.955 ± 0.002	0.961 ± 0.001	0.945 ± 0.002	0.960 ± 0.001	0.956 ± 0.001	0.954 ± 0.001
	Vote	0.933 ± 0.002	0.941 ± 0.001	0.918 ± 0.002	0.932 ± 0.002	0.921 ± 0.002	0.919 ± 0.002
	ND	0.956 ± 0.002	0.963 ± 0.001	0.946 ± 0.002	0.960 ± 0.002	0.954 ± 0.001	0.953 ± 0.001
	WLW	0.952 ± 0.002	0.960 ± 0.001	0.942 ± 0.001	0.957 ± 0.002	0.949 ± 0.001	0.949 ± 0.001
	CombVote	0.954 ± 0.001	0.961 ± 0.001	0.945 ± 0.002	0.959 ± 0.001	0.955 ± 0.001	0.953 ± 0.001
	NOV@	0.955 ± 0.002	0.961 ± 0.001	0.946 ± 0.002	0.960 ± 0.001	0.956 ± 0.001	0.954 ± 0.001
	GVote	0.955 ± 0.002	0.961 ± 0.001	0.946 ± 0.002	0.960 ± 0.001	0.956 ± 0.001	0.954 ± 0.001
	GND	0.956 ± 0.002	0.963 ± 0.001	0.947 ± 0.001	0.961 ± 0.001	0.957 ± 0.001	0.955 ± 0.001
	GWLW	0.955 ± 0.002	0.962 ± 0.001	0.945 ± 0.002	0.959 ± 0.002	0.953 ± 0.001	0.952 ± 0.001
	GET	0.955 ± 0.002	0.961 ± 0.001	0.946 ± 0.002	0.960 ± 0.001	0.956 ± 0.001	0.954 ± 0.001
	ET	0.945 ± 0.002	0.952 ± 0.001	0.932 ± 0.002	0.947 ± 0.002	0.938 ± 0.002	0.936 ± 0.002

TABLE 6.13: Classification accuracies $\mathsf{Acc}(\mathsf{f})$ on data sets induced by $\mathcal{M} \subseteq \mathcal{Y}$. The given values are computed using the same data and models as in table 6.12 such that the difference yields the improvement of forwarding the dynamic class information to the fusing step.

can occur whenever these sensitivities differ between classes, it is particularly relevant to still evaluate the improvements from integrating dynamic class information.

A possible solution is to analyze the respective class-specific accuracies in more detail in any of these situations. For a given classifier f trained on all classes $\mathcal{Y}$ and an arbitrary subselection $\mathcal{M} \subseteq \mathcal{Y}$, the respective accuracy $\mathsf{Acc}(\mathsf{f}; \mathcal{M})$ on data sampled according to $\mathcal{M}$ only can be computed. It should be noted that this is mainly the same as computing the respective dynamic classification accuracy (6.14). The important difference is the used classifier f and $\mathsf{f}_{\mathcal{M}}$ respectively, i.e. the computation evaluates the accuracy of f on data on $\mathcal{M}$ only *without forwarding this dynamic class information to the fusing step.* Consequently, the difference between $\mathsf{Acc}(\mathsf{f})$ and $\mathsf{Acc}(\mathsf{f}_{\mathcal{M}})$ yields the improvement from integrating the dynamic class information. The respective classification accuracies $\mathsf{Acc}(\mathsf{f})$ are presented in table 6.13. For a maximum comparability, these are computed under the same conditions as the dynamic classification accuracies $\mathsf{Acc}(\mathsf{f}_{\mathcal{M}})$ in table 6.12. It should be noted that the last column is missing because for $\mathcal{M} = \mathcal{Y}$, the classification rates coincide.

By computing the element-wise differences, for both network architectures on each of the three article groups an improvement of about one percent in classification accuracy is observed. For the unions of two article groups, the improvement is still about 0.3 % in classification accuracy. Relative to the remaining error of approximately 4 % to 5 %, this means a relative improvement of roughly 10 % to 20 %.

Besides these drastic improvements, it should be emphasized that in table 6.13 the classification accuracies computed on subsets cannot be averaged into the accuracy of their respective union. Even though the classifier does not depend on the restricted target class set, the different article sets are not disjoint.

Third Experiment: Comparison

After analyzing the improvement from dynamic classification, the last part focuses on comparing the methods on this particular data set. Still similarly to the study performed on deep learning benchmark data, the differences between most methods are small. Furthermore, the approaches based on the one-vs-one decomposition yield lower classification accuracies than the methods using the one-vs-all reduction. Even though this confirms the observations of the previous study evaluating the same network structures on other data sets, it is still worth noting because at least in theory the one-vs-one decomposition can better adapt to dynamic classification contexts.

6.4 Summary

After introducing two approaches realizing dynamic classification based on evidence theory as well as generalized pairwise coupling in chapters 4 and 5, respectively, this chapter performed different empirical evaluations. The first contribution introduces evaluation metrics for dynamic classification. These are based on generalizing the risk into the dynamic risk, still both cannot be computed directly as they depend on unknown probability distributions. Therefore, a similar extension to the empirical risk (equivalent to the error rate if the binary loss is used) is introduced as well. Still, this depends on a distribution over possible target sets $\mathcal{M}$, which can be approximated by respective sampling strategies. Because it yields a reasonable lower bound, particularly relevant is sampling q_1.

Thereafter, several comprehensive empirical experiments were performed. The study evaluated a collection of eleven fusing approaches that are either state-of-the-art reference methods presented in chapter 2 or newly introduced in chapters 4 and 5. It should be emphasized that each method depends on different reduction strategies: one-vs-all, one-vs-one or both.

In the first part of the performed experiments, all fusing methods are applied in combination with support vector machines and random forests as base classifiers on a collection of 26 reference data sets. Here, the one-vs-one reduction outperformed the one-vs-all decomposition in almost all experiments. Similarly, the weight estimation required for generalized pairwise coupling based on the one-vs-one reduction, as introduced in section 5.5, outperformed both competing approaches here. Most importantly, integrating dynamic class information improved the classification accuracy in all cases, applied significance tests were even highly significant.

The second part of the experiments repeated a similar evaluation using the state-of-the-art deep neural network structures Inception-V3 and ResNet-50. To compare the fusing approaches, five reference data sets from a recent publication were used. Because training a quadratic number of similar networks is computationally intractable, the techniques presented in section 5.5 were used that extend a previously trained neural network with a softmax output layer by one-vs-one prediction functions.

Interestingly, here the jointly optimized one-vs-all softmax outputs yield better results than their competing methods in the first study. In particular, they outperformed the one-vs-one decomposition in estimating the weights for generalized pairwise coupling as well as the softmax classification accuracy is either best or only outperformed by approaches that also depend on the one-vs-all class probabilities. However, the observed differences between the methods are always very small.

Besides this, all fusing approaches improve from integrating dynamic class information according to sampling q_1. Still, the absolute improvements are an order of magnitude smaller than in the previous study based on support vector machines and

random forests, most likely because the base accuracy is already very high in any case.

In the last part of the empirical evaluation, a real-world application was analyzed. For this, at first a reference data set consisting of 136009 images from 57 classes was collected. Here, the same deep learning neural network structures were trained and evaluated. Particularly relevant are the results with respect to dynamic classification because besides evaluating sampling strategy q_1, also task-specific target sets were evaluated.

The results show that the integration of dynamic class information successfully improves the classification accuracy. Particularly interesting is the insight that depending on the actual target set, the classification accuracy on *all* classes can be an inappropriate estimator for data sampled from a class subset only. If this is not sufficiently well respected, it can yield to the wrong conclusion that dynamic classification decreases the accuracy. To avoid this problem for a fixed target set $\mathcal{M}$, the accuracies *with* and *without* passing this information to the fusing step can be compared. This yields to an increase in accuracy of up to 1 % absolute, i.e. reduces the remaining error by 20 % to 25 %.

Chapter 7

Conclusion

The main aim of this thesis was to introduce dynamic classification as a generalization of classical multi-class classification such that at each time during two consecutive predictions, the target set can change arbitrarily to a subset $\mathcal{M} \subseteq \mathcal{Y}$ of the overall target set. Since there were no direct reference results available, the introduced approach combines two areas to remain computationally feasible.

Classifier Calibration

The first part thoroughly dealt with classifier calibration whose aim is to transform uncalibrated predictions into probabilities that are intended to be well calibrated. Existing works controversially discuss whether calibration mappings should be created monotonic or not. The major contributions in chapter 3 corrected wrong statements that appeared in the literature and most importantly proved that Platt scaling is optimal for different families of probability distributions, while the independently introduced approach Beta calibration is actually equivalent to Platt scaling up to a sigmoidal preprocessing.

Further theoretical results show that bin-based classifier calibration evaluation metrics are unreasonable because they apply another iteration of calibration on the test data. Therefore, they are simply unjustified and should not be used at all. Instead, the discussion and the empirical results show that proper scoring rules are well suited to compare calibration approaches. With respect to non-monotonic calibration, KDE and EKDE are two powerful model-free approaches that yield state-of-the-art results in the respective comprehensive empirical evaluation performed on 46 data sets.

Still, there are many interesting lines for further research. First, the selection of the appropriate calibration method remains unclear to arbitrary. Presumably, discrete approaches are more useful for discrete classifier prediction functions like decision trees and random forests, while continuous techniques like Platt scaling or KDE calibration are better suited for continuous calibration functions like those of support vector machines or neural networks. Furthermore, the choice between a monotonic or non-monotonic calibration technique remains similarly unclear.

The presented results at least partially explain why Platt scaling often can yield good results in practice. Still, one of the major problems is that probabilistic predictions can only be compared to class labels, but not to true posterior probabilities as they are unknown in practice. In light of this, it can be unreasonable to create increasingly complex and ensemble calibration methods – it is hardly possible at all to really compare them objectively to less complicated methods such that comparisons can easily be biased. Finally, it is unclear whether there are any justified evaluation metrics besides proper scoring rules.

Evidence Theory

Thereafter, chapter 4 presented an evidence-theoretic approach to reduction-based classification. Even though this involved a complex formalism including some potentially counterintuitive properties, evidence theory has several advantages that were useful to model the different decompositions. After first using classifier calibration to model mass functions for the individual classifiers instead of computing probabilities, Dempster's rule of combination allowed the predictions to be iteratively combined under relatively mild assumptions. This resulted in a Bayesian mass function such that the overall combination was obtained from a consistent base modeling and only applying the tools of evidence theory. In particular, this avoided an exponential complexity and therefore kept the approach computationally feasible.

With respect to the one-vs-all decomposition, a prediction rule equivalent to the one of the standard softmax function was recovered. However, the posterior probabilities induced by the respective mass function are different in general. Focusing on the one-vs-one reduction, the modeling required the integration of weights for each one-vs-one prediction function. Thereafter, a combination was possible and a closed-form expression could be proven.

Even though this required a weighting for each individual one-vs-one prediction function, thereafter assuming that the weights were constant but positive, they canceled out in the overall combination that simplified into a multiplicative voting. In this regard, evidence theory did not only allow the analysis of decomposition-based classification against a more formal background, it also yielded a unifying view to the one-vs-all and one-vs-one reduction. Most importantly, evidence theory allowed the integration of dynamic class information by modeling it as a mass function such that this framework led to a feasible approach to dynamic classification and thus yielded the first solution for this thesis' main aim.

Generalized Pairwise Coupling

Based on the insights of the evidence-theoretic modeling, the algorithmic family generalized pairwise coupling was introduced in chapter 5. The core idea was to combine pairwise classifier predictions with a corresponding weight during the coupling process.

Using this extension, the – according to reference results – three most relevant pairwise coupling approaches were generalized to be able to integrate these weightings, too. This not only extended the insights of the evidence-theoretic modeling, but additionally generalized the correcting classifiers approach that was independently introduced twice to the community.

However, integrating dynamic classification into generalized pairwise coupling was not similarly straightforward as in the evidence-theoretic case because there is no equivalent of the combination rule. Still in the context of evidence theory, the modeling obtained by integrating dynamic class information using a respective mass function could equivalently be described by modifying the weights accordingly. In particular, only those referring to pairs of classes inside the dynamic target set $\mathcal{M}$ remain, while all others were set to zero. This step was also possible in combination with generalized pairwise coupling and turned this algorithmic family into a second computationally tractable model for dynamic classification.

Empirical Results

Finally in chapter 6, a comprehensive empirical evaluation was performed to realize two main aims. First, to compare the introduced methods to existing state-of-the-art techniques and second, to analyze the improvement from integrating dynamic class information. In combination with support vector machines and random forests, the evidence-theoretic approaches yielded best or at least comparable results depending on the respective evaluation, still the differences were not detected to be significant in any case.

Applying deep learning networks on five benchmark data sets showed a different situation. Here, the one-vs-all softmax prediction often yielded very high accuracy. Even though there were situations where a more elaborated technique outperformed the softmax prediction in accuracy, all respective methods also depended on the one-vs-all predictions. Still, it should be emphasized that existing differences between the fusing methods were always very small in combination with deep neural networks.

These observations are explainable by the fact that the one-vs-all training was performed as an initialization step and all further trainings were added to the same network to keep the approaches computationally tractable. Here, it is a very interesting question whether and how training of one-vs-one networks can be realized such that significant accuracy gains might be obtained while the approach remains computationally feasible.

Besides this, applications of any of the introduced techniques in general multi-class settings is similarly interesting, both with respect to classical machine learning algorithms as well as large-scale deep neural networks. Due to their increased popularity in recent years, possible improvements of deep neural networks are highly relevant for many practical applications. Therefore, comparing the newly introduced methods on additional real-world data and maybe identifying problem characteristics that are well suited for these new techniques and in particular the evidence-theoretic methods remains an interesting question for further research.

According to reference results on decomposition-based classification, the non-competence problem is one of the major drawbacks of the one-vs-one reduction. In light of this, it is particularly interesting to observe that the pairwise coupling approaches mostly outperformed their generalized counterparts in the performed experiments. Additionally, estimating the weights by iterating pairwise coupling (`Weight1v1`, as discussed in full detail in section 5.5) yielded the best results in the first part of the study evaluating support vector machines and random forests. Even though this seems to contradict existing results, the challenging part is the correct estimation of the weights.

The non-competence problem does exist, but accurately solving it requires to solve a more generalized problem than the classification task itself. Especially the latter fact has also a direct connection to probability estimation and classifier calibration. Here, it remains an interesting question for further research whether there are weight estimation techniques that can be used in combination with generalized pairwise coupling such that the respective approaches successfully outperform their constant counterparts in almost any application.

Besides focusing on regular multi-class settings, the main aim of this thesis was to introduce computationally tractable approaches for dynamic classification. This was realized by two methods, an evidence-theoretic and a generalized pairwise coupling model. Here, the performed evaluations showed an improvement by integrating them into the fusing process. With respect to support vector machines and random forests, the observed accuracy increases were even detected as highly significant while the

best results are observed with fusing techniques based on one-vs-one classifiers only. On the other hand in combination with deep neural networks, the softmax one-vs-all predictions were not only highly accurate, but furthermore even remained among the best approaches when dynamic class information was additionally integrated.

This is remarkable because in general, the one-vs-one classifiers can adapt to dynamic contexts much better than those of the one-vs-all reduction, where all classifiers are trained on data from all classes. Here, the implicit dependency on the whole class set to combine all decision functions into a single model or using independent models to better adapt to dynamic classification contexts using independently trained one-vs-one prediction functions poses an interesting trade-off problem whose best solution remains unclear. Thus, combining extended techniques to train the one-vs-one decision functions in combination with large-scale neural networks has a strong relation to evaluating improvements from integrating dynamic class information. It is an interesting question whether sufficiently independently trained one-vs-one neural networks significantly outperform the one-vs-all softmax prediction as soon as a sufficient amount of dynamic class information is integrated in the fusing process.

It should be emphasized that this issue is an extension of the previously discussed one because it generalizes the question of improved one-vs-one neural networks to dynamic classification contexts. Even if the answer was negative, i.e. even integrating a substantial amount of dynamic class information would not outperform the one-vs-all softmax prediction, at least evidence theory justifies to integrate the dynamic class information in the fusing process by simply renormalizing the probability vector because the respective softmax prediction is equivalent to the one based on the one-vs-all evidence-theoretic modeling.

Real-World Application

With respect to a real-world application where also dynamic class information is available in general, the evaluation in section 6.3 showed that dynamic class information did not only improve the classification accuracy in theoretic benchmarks, but instead also in real-world settings. Sampling the dynamic class information, the observed improvements were mostly small, but still these depend on the amount of dynamic class information that can be supplied.

However, one of the main issues while realizing dynamic classification in the respective application was the lack of digital infrastructure to supply this information to the device. Still, this is highly task-/factory-specific and not a general statement such that analyzing further real-world applications of dynamic classification remains particularly relevant. It might be very interesting to focus on classification tasks that are sufficiently complex such that they only can be solved with an acceptable remaining error by integrating dynamic class information. It is unclear whether these applications exist or not.

Additionally, the evaluation on reasonable dynamic target sets showed that evaluating the improvements is a non-trivial task. Even though reasonable evaluation metrics were introduced in section 6.1, applying them in practice actually means that two effects are combined: The integration of dynamic class information in the respective fusing methods as well as test data sampling according to a subset of classes only. The latter fact alone can modify the classification accuracy, even though no dynamic class information is passed to the fusing step. This can be alleviated by sampling the test data only on the respective subset of classes and compare the respective classification accuracy *with* and *without* supplying it to the fusing methods. However, it can be counterintuitive for a user or a customer that an overall accuracy rating depends

on the current dynamic class information and hence improvements are evaluated to varying base accuracies.

Still, the evaluations showed that this can be reasonable because the overall accuracy might be an inadequate estimator for data sampled to a class subset only. Here, explicitly focusing on classes with a low class-specific sensitivity is particularly interesting as well. These difficult classes might not be noticed at all if their prior probabilities are sufficiently small and the evaluations focus only on accuracy. Here, dynamic classification is likely to significantly improve the respective sensitivities if other classes that cause ambiguities and misclassifications are excluded, even though this is not easily noticeable by accuracy statistics only.

Further Open Issues

The discussed issues are different potential directions for further research that aim directly on multi-class or dynamic classification, respectively. Besides this, it might also be an idea to also apply these techniques with focus on classifier calibration.

In this work, the latter was used as part of the former. Still, even the binary case is a very complicated problem such that multi-class calibration is even more challenging. Here, it is an interesting option to transform respective binary approaches in combination with decomposition-based classification into multi-class calibration techniques.

As reference results focus mostly on the binary case, existing works on multi-class calibration are rare. Still, this is an interesting direction for further research on multi-class calibration, for example compare respective approaches to those that solve it by reducing it to the binary case.

Bibliography

Alam, H., Rahman, F., Tarnikova, Y., & Hartono, R. (2003). "A pair-wise decision fusion framework: recognition of human faces". In: *Sixth International Conference of Information Fusion, 2003. Proceedings of the*. Cairns, Queensland, Australia: IEEE, pp. 1484–1489.

Allwein, E. L., Schapire, R. E., & Singer, Y. (2000). "Reducing Multiclass to Binary: A Unifying Approach for Margin Classifiers". In: *Journal of Machine Learning Research* 1, pp. 113–141.

Álvarez, E. E. & Yohai, V. J. (May 2011). "M-estimators for Isotonic Regression". In: *arXiv:1105.5065 [math, stat]*. arXiv: 1105.5065.

Alvarsson, J., Lampa, S., Schaal, W., Andersson, C., Wikberg, J. E. S., & Spjuth, O. (Dec. 2016). "Large-scale ligand-based predictive modelling using support vector machines". In: *Journal of Cheminformatics* 8.1.

Angulo, C. & Català, A. (2000). "K-SVCR. A Multi-class Support Vector Machine". In: *Machine Learning: ECML 2000*. Ed. by Carbonell, J. G., Siekmann, J., Goos, G., Hartmanis, J., Leeuwen, J. van, López de Mántaras, R., & Plaza, E. Vol. 1810. Berlin, Heidelberg: Springer Berlin Heidelberg, pp. 31–38.

Arruti, A., Mendialdua, I., Sierra, B., Lazkano, E., & Jauregi, E. (Oct. 2014). "New One Versus$^{\text{All}}_{\text{One}}$ method: NOV@". In: *Expert Systems with Applications* 41.14, pp. 6251–6260.

Azami, M. E., Lartizien, C., & Canu, S. (2016). "Converting SVDD Scores into Probability Estimates". In: *Computational Intelligence*, p. 6.

Bagheri, M. a., Gao, Q., & Escalera, S. (2012). "Efficient Pairwise Classification Using Local Cross Off Strategy". In: *Advances in Artificial Intelligence*. Ed. by Hutchison, D. et al. Vol. 7310. Berlin, Heidelberg: Springer Berlin Heidelberg, pp. 25–36.

Bain, T. C., Avila-Herrera, J. F., Subasi, E., & Subasi, M. M. (Sept. 2019). "Logical analysis of multiclass data with relaxed patterns". In: *Annals of Operations Research*.

Bella, A., Ferri, C., Hernández-Orallo, J., & Ramírez-Quintana, M. J. (2009a). "Calibration of Machine Learning Models". In: *Handbook of Research on Machine Learning Applications and Trends: Algorithms, Methods, and Techniques*, p. 18.

Bella, A., Ferri, C., Hernández-Orallo, J., & Ramírez-Quintana, M. J. (June 2013). "On the effect of calibration in classifier combination". In: *Applied Intelligence* 38.4, pp. 566–585.

Bella, A., Ferri, C., Hernández-Orallo, J., & Ramírez-Quintana, M. J. (Mar. 2014). "Aggregative quantification for regression". In: *Data Mining and Knowledge Discovery* 28.2, pp. 475–518.

Bella, A., Ferri, C., Hernández-Orallo, J., & Ramírez-Quintana, M. J. (2009b). "Similarity-Binning Averaging: A Generalisation of Binning Calibration". In: *Intelligent Data Engineering and Automated Learning - IDEAL 2009*. Ed. by Corchado, E. & Yin, H. Vol. 5788. Berlin, Heidelberg: Springer Berlin Heidelberg, pp. 341–349.

Benedetti, R. (Jan. 2010). "Scoring Rules for Forecast Verification". In: *Monthly Weather Review* 138.1, pp. 203–211.

Bennett, P. N. (2006). "Building Reliable Metaclassifiers for Text Learning". PhD thesis. Pittsburgh, PA 15213: Carnegie Mellon University.

Bennett, P. N. (Sept. 2000). *Assessing the Calibration of Naive Bayes' Posterior Estimates*. Tech. rep. CMU-CS-00-155. Computer Science Department, School of Computer Science, Carnegie Mellon University, p. 10.

Bennett, P. N. (2003). "Using Asymmetric Distributions to Improve Text Classifier Probability Estimates". In: *Proceedings of the 26th Annual International ACM SIGIR Conference on Research and Development in Informaion Retrieval*, pp. 111–118.

Bequé, A., Coussement, K., Gayler, R., & Lessmann, S. (Oct. 2017). "Approaches for credit scorecard calibration: An empirical analysis". In: *Knowledge-Based Systems* 134, pp. 213–227.

Bickel, J. E. (Dec. 2010). "Scoring Rules and Decision Analysis Education". In: *Decision Analysis* 7.4, pp. 346–357.

Bishop, C. M. (2009). *Pattern recognition and machine learning*. Corrected at 8th printing 2009. Information science and statistics. OCLC: 845772798. New York, NY: Springer.

Böken, B. (2014). *Algorithms for the classification of organic material in the food industry*.

Böken, B. (Jan. 2021). "On the appropriateness of Platt scaling in classifier calibration". In: *Information Systems* 95, p. 101641.

Borkowski, M., Fdhila, W., Nardelli, M., Rinderle-Ma, S., & Schulte, S. (Mar. 2019). "Event-based Failure Prediction in Distributed Business Processes". In: *Information Systems* 81. arXiv: 1712.08342, pp. 220–235.

Bradski, G. R. (2000). "The OpenCV Library". In: *Dr. Dobb's Journal of Software Tools*.

Cai, S., Zhang, L., Zuo, W., & Feng, X. (June 2016). "A Probabilistic Collaborative Representation Based Approach for Pattern Classification". In: *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. Las Vegas, NV, USA: IEEE, pp. 2950–2959.

Chang, C.-C. & Lin, C.-J. (Apr. 2011). "LIBSVM: A library for support vector machines". In: *ACM Transactions on Intelligent Systems and Technology* 2.3, pp. 1–27.

Chen, P.-C., Lee, K.-Y., Lee, T.-J., Lee, Y.-J., & Huang, S.-Y. (Mar. 2010). "Multiclass support vector classification via coding and regression". In: *Neurocomputing* 73.7-9, pp. 1501–1512.

Chmielnicki, W. (June 2015). "Creating Effective Error Correcting Output Codes for Multiclass Classification". In: *Lecture Notes in Artificial Intelligence* 9121, pp. 502–514.

Chmielnicki, W. & Stąpor, K. (Mar. 2016). "Using the one-versus-rest strategy with samples balancing to improve pairwise coupling classification". In: *International Journal of Applied Mathematics and Computer Science* 26.1, pp. 191–201.

Cohen, I. & Goldszmidt, M. (2004). "Properties and Benefits of Calibrated Classifiers". In: *Knowledge Discovery in Databases: PKDD 2004*. Ed. by Hutchison, D. et al. Vol. 3202. Berlin, Heidelberg: Springer Berlin Heidelberg, pp. 125–136.

Connolly, B., Cohen, K. B., Santel, D., Bayram, U., & Pestian, J. (Dec. 2017). "A nonparametric Bayesian method of translating machine learning scores to probabilities in clinical decision support". In: *BMC Bioinformatics* 18.1.

Crammer, K. & Singer, Y. (2001). "On the Algorithmic Implementation of Multiclass Kernel-based Vector Machines". In: *Journal of Machine Learning Research* 2, pp. 265–292.

Crammer, K. & Singer, Y. (May 2002). "On the Learnability and Design of Output Codes for Multiclass Problems". In: *Machine Learning* 47.2, pp. 201–233.

Cruz, R. M., Sabourin, R., & Cavalcanti, G. D. (May 2018). "Dynamic classifier selection: Recent advances and perspectives". In: *Information Fusion* 41, pp. 195–216.

Cutzu, F. (2003). "Polychotomous Classification with Pairwise Classifiers: A New Voting Principle". In: *Multiple Classifier Systems*. Springer Berlin Heidelberg, pp. 115–124.

Cybakov, A. B. (2009). *Introduction to nonparametric estimation*. Springer series in statistics. OCLC: 844927432. New York NY: Springer.

Dankowski, T. & Ziegler, A. (Sept. 2016). "Calibrating random forests for probability estimation". In: *Statistics in Medicine* 35.22, pp. 3949–3960.

DeGroot, M. H. & Fienberg, S. E. (1983). "The comparison and evaluation of forecasters". In: *The Statistician: Journal of the Institute of Statisticians* 32, pp. 12–22.

Demšar, J. (2006). "Statistical Comparisons of Classifiers over Multiple Data Sets". In: *The Journal of Machine Learning Research* 7, p. 30.

Demšar, J. (2008). *On the Appropriateness of Statistical Tests in Machine Learning*.

Deng, J., Dong, W., Socher, R., Li, L.-J., Kai Li, & Li Fei-Fei (June 2009). "ImageNet: A large-scale hierarchical image database". In: *2009 IEEE Conference on Computer Vision and Pattern Recognition*. Miami, FL: IEEE, pp. 248–255.

Destercke, S. & Quost, B. (2011). "Combining Binary Classifiers with Imprecise Probabilities". In: *Integrated Uncertainty in Knowledge Modelling and Decision Making*. Ed. by Tang, Y., Huynh, V.-N., & Lawry, J. Vol. 7027. Berlin, Heidelberg: Springer Berlin Heidelberg, pp. 219–230.

Devroye, L., Györfi, L., & Lugosi, G. (2008). *A probabilistic theory of pattern recognition*. 3. print. Applications of mathematics 31. OCLC: 553508627. New York, NY: Springer.

Dezert, J. & Tchamova, A. (Mar. 2014). "On the Validity of Dempster's Fusion Rule and its Interpretation as a Generalization of Bayesian Fusion Rule". In: *International Journal of Intelligent Systems* 29.3, pp. 223–252.

Dietterich, T. G. & Bakiri, G. (Jan. 1995). "Solving Multiclass Learning Problems via Error-Correcting Output Codes". In: *Journal of Artificial Intelligence Research* 2, pp. 263–286.

Doğan, Ü., Glasmachers, T., & Igel, C. (2016). "A Unified View on Multi-class Support Vector Classification". In: *Journal of Machine Learning Research* 17.45, pp. 1–32.

Domingos, P. M. & Pazzani, M. J. (1996). "Beyond independence: conditions for the optimality of the simple Bayesian classifier". In: *Proceedings of the Thirteenth International Conference on International Conference on Machine Learning*. Bari, Italy, pp. 105–112.

Drish, J. (2001). *Obtaining Calibrated Probability Estimates from Support Vector Machines*.

Dua, D. & Graff, C. (2019). *UCI Machine Learning Repository*.

Duan, K., Keerthi, S. S., Chu, W., Shevade, S. K., & Poo, A. N. (2003). "Multi-category Classification by Soft-Max Combination of Binary Classifiers". In: *Multiple Classifier Systems: 4th International Workshop, Guildford, UK, June 11 -*

13, 2003; Proceedings. Ed. by Windeatt, T. & Roli, F. Lecture notes in computer science 2709. Berlin: Springer, pp. 125–134.

Duin, R. P. W. & Pekalska, E. (2005). "Open Issues in Pattern Recognition". In: *Computer Recognition Systems*. Ed. by Kurzyński, M., Puchała, E., Woźniak, M., & Żołnierek, A. Vol. 30. Berlin, Heidelberg: Springer Berlin Heidelberg, pp. 27–42.

Eck, C., Garcke, H., & Knabner, P. (2017). *Mathematical modeling*. Springer undergraduate mathematics series. Cham: Springer.

Elkan, C. (2001a). "Magical Thinking in Data Mining: Lessons From CoIL Challenge 2000". In: *Proceedings of the Seventh ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*. San Francisco, California, pp. 426–431.

Elkan, C. (2001b). "The Foundations of Cost-Sensitive Learning". In: *Proceedings of the 17th International Joint Conference on Artificial Intelligence*. Vol. 2. IJCAI'01. Seattle, WA, USA: Morgan Kaufmann Publishers Inc., pp. 973–978.

Elkano, M., Galar, M., Sanz, J., Lucca, G., & Bustince, H. (June 2017). "IVOVO: A new interval-valued one-vs-one approach for multi-class classification problems". In: *2017 Joint 17th World Congress of International Fuzzy Systems Association and 9th International Conference on Soft Computing and Intelligent Systems (IFSA-SCIS)*. Otsu, Japan: IEEE, pp. 1–6.

Elkano, M., Galar, M., Sanz, J. A., Fernandez, A., Barrenechea, E., Herrera, F., & Bustince, H. (Oct. 2015). "Enhancing Multiclass Classification in FARC-HD Fuzzy Classifier: On the Synergy Between n-Dimensional Overlap Functions and Decomposition Strategies". In: *IEEE Transactions on Fuzzy Systems* 23.5, pp. 1562–1580.

Epanechnikov, V. A. & Seckler, B. (1969). "Non-parametric estimation of a multivariate probability density". In: *Theory of Probability and its Applications* 14.1, pp. 153–158.

Escalera, S., Pujol, O., & Radeva, P. (May 2010). "Re-coding ECOCs without retraining". In: *Pattern Recognition Letters* 31.7, pp. 555–562.

Fan, R.-E., Chang, K.-W., Hsieh, C.-J., Wang, X.-R., & Lin, C.-J. (2008). "LIBLINEAR: A Library for Large Linear Classification". In: *Journal of Machine Learning Research* 2008.9, pp. 1871–1874.

Fan, R.-E., Chen, P.-H., & Lin, C.-J. (2005). "Working Set Selection Using Second Order Information for Training Support Vector Machines". In: *Journal of Machine Learning Research* 2005.6, pp. 1889–1918.

Fernandez, A., Galar, M., Sanz, J. A., Bustince, H., & Herrera, F. (2015). "Improving Pairwise Learning Classification in Fuzzy Rule Based Classification Systems Using Dynamic Classifier Selection". In: *Proceedings of the 2015 Conference of the International Fuzzy Systems Association and the European Society for Fuzzy Logic and Technology*. Gijón, Spain.: Atlantis Press.

Fernández, A., Calderón, M., Barrenechea, E., Bustince, H., & Herrera, F. (Dec. 2010). "Solving multi-class problems with linguistic fuzzy rule based classification systems based on pairwise learning and preference relations". In: *Fuzzy Sets and Systems* 161.23, pp. 3064–3080.

Fernández, A., López, V., Galar, M., Jesus, M. J. del, & Herrera, F. (Apr. 2013). "Analysing the classification of imbalanced data-sets with multiple classes: Binarization techniques and ad-hoc approaches". In: *Knowledge-Based Systems* 42, pp. 97–110.

Flach, P. A. (2016). "Classifier Calibration". In: *Encyclopedia of Machine Learning and Data Mining*. Springer US, pp. 210–217.

Fonseca, P. G. & Lopes, H. D. (2017). *Calibration of Machine Learning Classifiers for Probability of Default Modelling*. Tech. rep. James Finance (CrowdProcess Inc.)

Franc, V., Zien, A., & Schölkopf, B. (2011). "Support Vector Machines as Probabilistic Models". In: *Proceedings of the 28th International Conference on Machine Learning*. Bellevue, WA, USA, p. 8.

Friedman, J. H. (Oct. 1996). *Another Approach to Polychotomous Classification*. Tech. rep. Stanford, CA: Department of Statistics, Stanford University.

Fürnkranz, J. (2001). "Round Robin Rule Learning". In: *Proceedings of the Eighteenth International Conference on Machine Learning*. ICML '01. San Francisco, CA, USA: Morgan Kaufmann Publishers Inc., pp. 146–153.

Fürnkranz, J. (2002a). "Pairwise Classification as an Ensemble Technique". In: *Machine Learning: ECML 2002*. Ed. by Goos, G., Hartmanis, J., Leeuwen, J. van, Elomaa, T., Mannila, H., & Toivonen, H. Vol. 2430. Berlin, Heidelberg: Springer Berlin Heidelberg, pp. 97–110.

Fürnkranz, J. (Mar. 2002b). "Round Robin Classification". In: *Journal of Machine Learning Research* 2, pp. 721–747.

Fürnkranz, J. (Oct. 2003). "Round Robin Ensembles". In: *Intell. Data Anal.* 7.5, pp. 385–403.

Fürnkranz, J., Hüllermeier, E., & Vanderlooy, S. (2009). "Binary Decomposition Methods for Multipartite Ranking". In: *Machine Learning and Knowledge Discovery in Databases*, pp. 359–374.

Galar, M., Fernandez, A., Barrenechea, E., Bustince, H., & Herrera, F. (2010). *Aggregation schemes for binarization techniques. Methods' description*.

Galar, M., Fernández, A., Barrenechea, E., Bustince, H., & Herrera, F. (Aug. 2011). "An overview of ensemble methods for binary classifiers in multi-class problems: Experimental study on one-vs-one and one-vs-all schemes". In: *Pattern Recognition* 44.8, pp. 1761–1776.

Galar, M., Fernández, A., Barrenechea, E., Bustince, H., & Herrera, F. (Dec. 2013). "Dynamic classifier selection for One-vs-One strategy: Avoiding non-competent classifiers". In: *Pattern Recognition* 46.12, pp. 3412–3424.

Galar, M., Fernández, A., Barrenechea, E., Bustince, H., & Herrera, F. (July 2017). "NMC: nearest matrix classification – A new combination model for pruning One-vs-One ensembles by transforming the aggregation problem". In: *Information Fusion* 36, pp. 26–51.

Galar, M., Fernández, A., Barrenechea, E., & Herrera, F. (Apr. 2014). "Empowering difficult classes with a similarity-based aggregation in multi-class classification problems". In: *Information Sciences* 264, pp. 135–157.

Galar, M., Fernández, A., Barrenechea, E., & Herrera, F. (Jan. 2015). "DRCW-OVO: Distance-based relative competence weighting combination for One-vs-One strategy in multi-class problems". In: *Pattern Recognition* 48.1, pp. 28–42.

Garcia-Pedrajas, N. & Ortiz-Boyer, D. (June 2006). "Improving multiclass pattern recognition by the combination of two strategies". In: *IEEE Transactions on Pattern Analysis and Machine Intelligence* 28.6, pp. 1001–1006.

García-Pedrajas, N. & Ortiz-Boyer, D. (2008). *A thorough empirical study of output coding methods for multiclass classification*. Tech. rep. Córdoba, Spain: Computational Intelligence and Bioinformatics Research Group, University of Córdoba, p. 34.

García-Pedrajas, N. & Ortiz-Boyer, D. (Apr. 2011). "An empirical study of binary classifier fusion methods for multiclass classification". In: *Information Fusion* 12.2, pp. 111–130.

Gebel, M. (2009). "Multivariate calibration of classifier scores into the probability space". PhD thesis. Technische Universität Dortmund.

Goienetxea, I., Mendialdua, I., Rodríguez, I., & Sierra, B. (Jan. 2021). "Problems selection under dynamic selection of the best base classifier in one versus one: PSEUDOVO". In: *International Journal of Machine Learning and Cybernetics*, p. 15.

Gonzalez-Abril, L., Angulo, C., Velasco, F., & Ortega, J. A. (2010). "A Probabilistic Tri-Class Support Vector Machine". In: *Journal of Pattern Recognition Research* 5.1, pp. 1–9.

Goodman, S. (July 2008). "A Dirty Dozen: Twelve P-Value Misconceptions". In: *Seminars in Hematology* 45.3, pp. 135–140.

Grandvalet, Y., Mariethoz, J., & Bengio, S. (2005). "A Probabilistic Interpretation of SVMs with an Application to Unbalanced Classification". In: *Advances in Neural Information Processing Systems*. NIPS'05, p. 11.

Guo, C., Pleiss, G., Sun, Y., & Weinberger, K. Q. (June 2017). "On Calibration of Modern Neural Networks". In: *arXiv:1706.04599 [cs]*. arXiv: 1706.04599.

Hastie, T. & Tibshirani, R. (Apr. 1998). "Classification by pairwise coupling". In: *The Annals of Statistics* 26.2, pp. 451–471.

He, K., Zhang, X., Ren, S., & Sun, J. (June 2016). "Deep Residual Learning for Image Recognition". In: *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. Las Vegas, NV, USA: IEEE, pp. 770–778.

Heckerman, D., Geiger, D., & Chickering, D. M. (Sept. 1995). "Learning Bayesian Networks: The Combination of Knowledge and Statistical Data". In: *Machine Learning* 20.3, pp. 197–243.

Hong, J.-H., Min, J.-K., Cho, U.-K., & Cho, S.-B. (Feb. 2008). "Fingerprint classification using one-vs-all support vector machines dynamically ordered with naïve Bayes classifiers". In: *Pattern Recognition* 41.2, pp. 662–671.

Hsu, C.-W., Chang, C.-C., & Lin, C.-J. (2016). *A Practical Guide to Support Vector Classification*. Tech. rep. Taiwan: National Taiwan University, Department of Computer Science, p. 16.

Hsu, C.-W. & Lin, C.-J. (Mar. 2002). "A comparison of methods for multiclass support vector machines". In: *IEEE Transactions on Neural Networks* 13.2, pp. 415–425.

Hsu, Y.-C., Lv, Z., Schlosser, J., Odom, P., & Kira, Z. (2019). "Multi-class Classification without Multi-class Labels". In: *CoRR* abs/1901.00544, p. 16.

Hüllermeier, E. & Brinker, K. (Sept. 2008). "Learning valued preference structures for solving classification problems". In: *Fuzzy Sets and Systems* 159.18, pp. 2337–2352.

Hüllermeier, E. & Vanderlooy, S. (Jan. 2010). "Combining predictions in pairwise classification: An optimal adaptive voting strategy and its relation to weighted voting". In: *Pattern Recognition* 43.1, pp. 128–142.

Jabbari, F., Naeini, M. P., & Cooper, G. F. (Dec. 2017). "Obtaining Accurate Probabilistic Causal Inference by Post-Processing Calibration". In: *arXiv:1712.08626 [cs, stat]*. arXiv: 1712.08626.

Jafri, R. & Arabnia, H. R. (June 2009). "A Survey of Face Recognition Techniques". In: *Journal of Information Processing Systems* 5.2, pp. 41–68.

James, G. (1998). "Majority Vote Classifiers: Theory and Applications". PhD thesis. Stanford University.

Jelonek, J. & Stefanowski, J. (1998). "Experiments on solving multiclass learning problems by n^2-classifier". In: *Machine Learning: ECML-98*. Ed. by Carbonell, J. G., Siekmann, J., Goos, G., Hartmanis, J., Leeuwen, J. van, Nédellec, C., & Rouveirol, C. Vol. 1398. Berlin, Heidelberg: Springer Berlin Heidelberg, pp. 172–177.

Jiang, X., Osl, M., Kim, J., & Ohno-Machado, L. (2011). "Smooth Isotonic Regression: A New Method to Calibrate Predictive Models". In: *AMIA Joint Summits on Translational Science*, p. 5.

Jiang, X., Osl, M., Kim, J., & Ohno-Machado, L. (Mar. 2012). "Calibrating predictive model estimates to support personalized medicine". In: *Journal of the American Medical Informatics Association* 19.2, pp. 263–274.

Joachims, T. (1998). "Text categorization with Support Vector Machines: Learning with many relevant features". In: *Machine Learning: ECML-98*. Ed. by Carbonell, J. G., Siekmann, J., Goos, G., Hartmanis, J., Leeuwen, J. van, Nédellec, C., & Rouveirol, C. Vol. 1398. Berlin, Heidelberg: Springer Berlin Heidelberg, pp. 137–142.

Khalifa, Y., Hawks, J., & Sejdic, E. (May 2019). "Single neuron-based neural networks are as efficient as dense deep neural networks in binary and multi-class recognition problems". In: *arXiv:1905.12135 [cs, stat]*. arXiv: 1905.12135.

Kikuchi, T. & Abe, S. (Jan. 2003). *Error Correcting Output Codes vs. Fuzzy Support Vector Machines*.

Kim, K. I. & Simon, R. (July 2011). "Probabilistic classifiers with high-dimensional data". In: *Biostatistics* 12.3, pp. 399–412.

Klimo, M., Lukáč, P., & Tarábek, P. (Apr. 2021). "Deep Neural Networks Classification via Binary Error-Detecting Output Codes". In: *Applied Sciences* 11.8, p. 3563.

Ko, J. & Byun, H. (2003). "Binary Classifier Fusion Based on the Basic Decomposition Methods". In: *Multiple Classifier Systems: 4th International Workshop, Guildford, UK, June 11 - 13, 2003; Proceedings*. Ed. by Windeatt, T. & Roli, F. Lecture notes in computer science 2709. Berlin: Springer, p. 406.

Kong, E. B. & Dietterich, T. G. (Oct. 1994). *Why Error-Correcting Output Coding Works*. Tech. rep. Corvallis, Oregon: Department of Computer Science, Oregon State University, p. 23.

Kong, E. B. & Dietterich, T. G. (1995). "Error-Correcting Output Coding Corrects Bias and Variance". In: *In Proceedings of the Twelfth International Conference on Machine Learning*, pp. 313–321.

Kong, E. B. & Dietterich, T. G. (1997). "Probability Estimation via Error-Correcting Output Coding". In: *In Int. Conf. of Artificial Inteligence and soft computing*. Banff, Canada, p. 6.

Kruppa, J., Liu, Y., Biau, G., Kohler, M., König, I. R., Malley, J. D., & Ziegler, A. (July 2014a). "Probability estimation with machine learning methods for dichotomous and multicategory outcome: Theory". In: *Biometrical Journal* 56.4, pp. 534–563.

Kruppa, J., Liu, Y., Diener, H.-C., Holste, T., Weimar, C., König, I. R., & Ziegler, A. (July 2014b). "Probability estimation with machine learning methods for dichotomous and multicategory outcome: Applications". In: *Biometrical Journal* 56.4, pp. 564–583.

Krzyśko, M. & Wołyński, W. (Dec. 2009). "New variants of pairwise classification". In: *European Journal of Operational Research* 199.2, pp. 512–519.

Kull, M., Filho, T. S., & Flach, P. (2017). "Beta calibration: a well-founded and easily implemented improvement on logistic calibration for binary classifiers". In: *Proceedings of Machine Learning Research* 54, p. 9.

Kull, M. & Flach, P. (2015). "Novel Decompositions of Proper Scoring Rules for Classification: Score Adjustment as Precursor to Calibration". In: *Machine Learning and Knowledge Discovery in Databases*. Ed. by Appice, A., Rodrigues, P. P.,

Santos Costa, V., Soares, C., Gama, J., & Jorge, A. Vol. 9284. Cham: Springer International Publishing, pp. 68–85.

Lachaize, M., Le Hégarat-Mascle, S., Aldea, E., Maitrot, A., & Reynaud, R. (2016). "SVM Classifier Fusion Using Belief Functions: Application to Hyperspectral Data Classification". In: *Belief Functions: Theory and Applications*. Ed. by Vejnarová, J. & Kratochvíl, V. Vol. 9861. Cham: Springer International Publishing, pp. 113–122.

Leathart, T., Frank, E., Holmes, G., & Pfahringer, B. (2017). "Probability Calibration Trees". In: *Proceedings of Machine Learning Research* 77, pp. 145–160.

Lee, Y., Lin, Y., & Wahba, G. (Mar. 2004). "Multicategory Support Vector Machines: Theory and Application to the Classification of Microarray Data and Satellite Radiance Data". In: *Journal of the American Statistical Association* 99.465, pp. 67–81.

Lei, H. & Govindaraju, V. (2005). "Half-Against-Half Multi-class Support Vector Machines". In: *Multiple Classifier Systems*. Ed. by Hutchison, D. et al. Vol. 3541. Berlin, Heidelberg: Springer Berlin Heidelberg, pp. 156–164.

Leitão, P. & Restivo, F. (Feb. 2006). "ADACOR: A holonic architecture for agile and adaptive manufacturing control". In: *Computers in Industry* 57.2, pp. 121–130.

Lessmann, S., Baesens, B., Seow, H.-V., & Thomas, L. C. (Nov. 2015). "Benchmarking state-of-the-art classification algorithms for credit scoring: An update of research". In: *European Journal of Operational Research* 247.1, pp. 124–136.

Li, H., Qi, F., & Wang, S. (2005a). "Face Recognition with Improved Pairwise Coupling Support Vector Machines". In: *Computational Intelligence and Bioinspired Systems*. Ed. by Hutchison, D. et al. Vol. 3512. Berlin, Heidelberg: Springer Berlin Heidelberg, pp. 927–934.

Li, H., Qi, F., & Wang, S. (2005b). "Improved Pairwise Coupling Support Vector Machines with Correcting Classifiers". In: *MICAI 2005: Advances in Artificial Intelligence*. Ed. by Hutchison, D. et al. Vol. 3789. Berlin, Heidelberg: Springer Berlin Heidelberg, pp. 454–461.

Li, Z. & Tang, S. (2002). "Face recognition using improved pairwise coupling support vector machines". In: *Proceedings of the 9th International Conference on Neural Information Processing, 2002. ICONIP '02.* Vol. 2. Singapore: Nanyang Technol. Univ, pp. 876–880.

Likhomanenko, T., Derkach, D., & Rogozhnikov, A. (Oct. 2016). "Inclusive Flavour Tagging Algorithm". In: *Journal of Physics: Conference Series* 762, p. 012045.

Lin, H.-T., Lin, C.-J., & Weng, R. C. (Aug. 2007). "A note on Platt's probabilistic outputs for support vector machines". In: *Machine Learning* 68.3, pp. 267–276.

Lin, Z. & Davis, L. S. (2008). "Learning Pairwise Dissimilarity Profiles for Appearance Recognition in Visual Surveillance". In: *Advances in Visual Computing*. Ed. by Hutchison, D. et al. Vol. 5358. Berlin, Heidelberg: Springer Berlin Heidelberg, pp. 23–34.

Liu, B., Hao, Z., & Tsang, E. C. C. (Dec. 2008). "Nesting One-Against-One Algorithm Based on SVMs for Pattern Classification". In: *IEEE Transactions on Neural Networks* 19.12, pp. 2044–2052.

Liu, B., Hao, Z., & Yang, X. (Nov. 2006). "Nesting Algorithm for Multi-Classification Problems". In: *Soft Computing* 11.4, pp. 383–389.

Liu, H., Lafferty, J. D., & Wasserman, L. (2007). "Sparse Nonparametric Density Estimation in High Dimensions Using the Rodeo". In: *Proceedings of the Eleventh International Conference on Artificial Intelligence and Statistics*, pp. 283–290.

Lorena, A. C., Carvalho, A. C. P. L. F. de, & Gama, J. M. P. (Dec. 2008). "A review on the combination of binary classifiers in multiclass problems". In: *Artificial Intelligence Review* 30.1-4, pp. 19–37.

Maass, W. (2000). *On the Computational Power of Winner-Take-All*. Tech. rep. 32. Graz: Institute for Theoretical Computer Science, Technische Universität Graz, p. 19.

Madzarov, G., Gjorgjevikj, D., & Chorbev, I. (Jan. 2009). "A Multi-class SVM Classifier Utilizing Binary Decision Tree". In: *Informatica (Slovenia)* 33, pp. 225–233.

Malley, J. D., Kruppa, J., Dasgupta, A., Malley, K. G., & Ziegler, A. (2012). "Probability Machines: Consistent Probability Estimation Using Nonparametric Learning Machines". In: *Methods of Information in Medicine* 51.01, pp. 74–81.

Mendialdua, I., Martínez-Otzeta, J. M., Rodriguez, I., Ruiz-Vázquez, T., & Sierra, B. (May 2015). "Dynamic selection of the best base classifier in One versus One". In: *Knowledge-Based Systems* 85.

Merkle, E. C. & Steyvers, M. (Dec. 2013). "Choosing a Strictly Proper Scoring Rule". In: *Decision Analysis* 10.4, pp. 292–304.

Montañés, E., Barranquero, J., Díez, J., & Coz, J. J. del (Feb. 2013). "Enhancing directed binary trees for multi-class classification". In: *Information Sciences* 223, pp. 42–55.

Morales-Ramirez, I., Kifetew, F. M., & Perini, A. (Dec. 2019). "Speech-acts based analysis for requirements discovery from online discussions". In: *Information Systems* 86, pp. 94–112.

Morán-Fernández, L., Bolón-Canedo, V., & Alonso-Betanzos, A. (2016). "Selection of the Best Base Classifier in One-Versus-One Using Data Complexity Measures". In: *Advances in Artificial Intelligence*. Ed. by Luaces, O., Gámez, J. A., Barrenechea, E., Troncoso, A., Galar, M., Quintián, H., & Corchado, E. Vol. 9868. Cham: Springer International Publishing, pp. 110–120.

Moreira, M. & Mayoraz, E. (1998). "Improved pairwise coupling classification with correcting classifiers". In: *Machine Learning: ECML-98*. Ed. by Carbonell, J. G., Siekmann, J., Goos, G., Hartmanis, J., Leeuwen, J. van, Nédellec, C., & Rouveirol, C. Vol. 1398. Berlin, Heidelberg: Springer Berlin Heidelberg, pp. 160–171.

Murphy, A. H. & Winkler, R. L. (1977). "Reliability of Subjective Probability Forecasts of Precipitation and Temperature". In: *Applied Statistics* 26.1, pp. 41–47.

Naeini, M. P. (2016). "Obtaining Accurate Probabilities using Classifier Calibration". PhD thesis. University of Pittsburgh.

Naeini, M. P., Cooper, G. F., & Hauskrecht, M. (2015a). "Obtaining Well Calibrated Probabilities Using Bayesian Binning". In: *Proceedings of the Twenty-Ninth AAAI Conference on Artificial Intelligence*. Austin, Texas, pp. 2901–2907.

Naeini, M. P. & Cooper, G. F. (Nov. 2015). "Binary Classifier Calibration using an Ensemble of Near Isotonic Regression Models". In: *arXiv:1511.05191 [cs, stat]*. arXiv: 1511.05191.

Naeini, M. P. & Cooper, G. F. (June 2016). "Binary Classifier Calibration Using an Ensemble of Linear Trend Estimation". In: *Proceedings of the 2016 SIAM International Conference on Data Mining*. Society for Industrial and Applied Mathematics, pp. 261–269.

Naeini, M. P. & Cooper, G. F. (Jan. 2018). "Binary classifier calibration using an ensemble of piecewise linear regression models". In: *Knowledge and Information Systems* 54.1, pp. 151–170.

Naeini, M. P., Cooper, G. F., & Hauskrecht, M. (Jan. 2014). "Binary Classifier Calibration: A Bayesian Non-Parametric Approach". In: *arXiv:1401.2955 [cs, stat]*. arXiv: 1401.2955.

Naeini, M. P., Cooper, G. F., & Hauskrecht, M. (June 2015b). "Binary Classifier Calibration Using a Bayesian Non-Parametric Approach". In: *Proceedings of the 2015 SIAM International Conference on Data Mining*. Ed. by Venkatasubramanian, S. & Ye, J. Philadelphia, PA: Society for Industrial and Applied Mathematics, pp. 208–216.

Niculescu-Mizil, A. & Caruana, R. (2005a). "Obtaining Calibrated Probabilities from Boosting". In: *Proceedings of the Twenty-First Conference on Uncertainty in Artificial Intelligence*. Edinburgh, Scotland: AUAI Press, pp. 413–420.

Niculescu-Mizil, A. & Caruana, R. (2005b). "Predicting good probabilities with supervised learning". In: *Proceedings of the 22nd international conference on Machine learning - ICML '05*. Bonn, Germany: ACM Press, pp. 625–632.

Nixon, J., Dusenberry, M. W., Zhang, L., Jerfel, G., & Tran, D. (2019). "Measuring Calibration in Deep Learning". In: *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition Workshops*, pp. 38–41.

Ou, G. & Murphey, Y. L. (Jan. 2007). "Multi-class pattern classification using neural networks". In: *Pattern Recognition* 40.1, pp. 4–18.

Park, S.-H. & Fürnkranz, J. (2007). "Efficient Pairwise Classification". In: *Machine Learning: ECML 2007*. Ed. by Kok, J. N., Koronacki, J., Mantaras, R. L. d., Matwin, S., Mladenič, D., & Skowron, A. Vol. 4701. Berlin, Heidelberg: Springer Berlin Heidelberg, pp. 658–665.

Park, S.-H., Huh, S.-Y., Zhang, P., & Shi, Y. (2009). "A Study on Identifying Essential Hyperplanes for Constructing a Multiclass Classification Model". In: *2009 Fifth International Joint Conference on INC, IMS and IDC*. Seoul, South Korea: IEEE, pp. 1798–1804.

Parmigiani, G. & Inoue, L. Y. T. (2009). *Decision Theory: Principles and Approaches*. Wiley Series in Probability and Statistics. Wiley.

Parzen, E. (1962). "On Estimation of a Probability Density Function and Mode". In: *JSTOR: The Annals of Mathematical Statistics* 33.3, pp. 1065–1076.

Passerini, A., Pontil, M., & Frasconi, P. (Jan. 2004). "New Results on Error Correcting Output Codes of Kernel Machines". In: *IEEE Transactions on Neural Networks* 15.1, pp. 45–54.

Pawara, P., Okafor, E., Groefsema, M., He, S., Schomaker, L. R., & Wiering, M. A. (Dec. 2020). "One-vs-One classification for deep neural networks". In: *Pattern Recognition* 108, p. 107528.

Plataniotis, K. N. & Hatzinakos, D. (Jan. 2000). "Gaussian Mixtures and their Applications to Signal Processing". In: *Advanced Signal Processing Handbook: Theory and Implementation for Radar, Sonar, and Medical Imaging Real Time Systems*.

Platt, J. C. (1999). *Probabilistic Outputs for Support Vector Machines and Comparisons to Regularized Likelihood Methods*. Tech. rep. Redmond, WA 98052: Microsoft Research.

Platt, J. C. (2000). "Probabilities for Support Vector Machines". In: *Advances in Large Margin Classifiers*. Ed. by Smola, A. J., Bartlett, P. L., Schölkopf, B., & Schuurmans, D. Cambridge: MIT Press, pp. 61–74.

Platt, J. C., Cristianini, N., & Shawe-Taylor, J. (1999). "Large Margin DAGs for Multiclass Classification". In: *Proceedings of the 12th International Conference on Neural Information Processing Systems*. NIPS'99. Denver, CO: MIT Press, pp. 547–553.

Price, D., Knerr, S., Personnaz, L., & Dreyfus, G. (1995). "Pairwise Neural Network Classifiers with Probabilistic Outputs". In: *Advances in Neural Information Processing Systems 7*. Ed. by Tesauro, G., Touretzky, D. S., & Leen, T. K. MIT Press, pp. 1109–1116.

Pujol, O., Radeva, P., & Vitria, J. (June 2006). "Discriminant ECOC: a heuristic method for application dependent design of error correcting output codes". In: *IEEE Transactions on Pattern Analysis and Machine Intelligence* 28.6, pp. 1007–1012.

Qin, G., Huang, X., & Chen, Y. (Mar. 2017). "Nested One-to-One Symmetric Classification Method on a Fuzzy SVM for Moving Vehicles". In: *Symmetry* 9.4, p. 48.

Qin, Z. & Lu, Y. (July 2021). "Self-organizing manufacturing network: A paradigm towards smart manufacturing in mass personalization". In: *Journal of Manufacturing Systems* 60, pp. 35–47.

Quost, B., Denœux, T., & Masson, M.-H. (Apr. 2007). "Pairwise Classifier Combination Using Belief Functions". In: *Pattern Recognition Letters* 28.5, pp. 644–653.

Quost, B. & Destercke, S. (May 2018). "Classification by pairwise coupling of imprecise probabilities". In: *Pattern Recognition* 77, pp. 412–425.

Rahman, A. F. R. & Fairhurst, M. C. (1997). "A novel pair-wise recognition scheme for handwritten characters in the framework of a multi-expert configuration". In: *Image Analysis and Processing*. Ed. by Goos, G., Hartmanis, J., Leeuwen, J. van, & Del Bimbo, A. Vol. 1311. Berlin, Heidelberg: Springer Berlin Heidelberg, pp. 624–631.

Rätsch, G., Smola, A. J., & Mika, S. (Jan. 2002). "Adapting Codes and Embeddings for Polychotomies". In: *Advances in Neural Information Processing Systems*. Vol. 15, pp. 513–520.

Reid, S. R. (2010). "Model Combination in Multiclass Classification". PhD thesis. University of Colorado.

Reineking, T. (2014). "Belief Functions: Theory and Algorithms". PhD thesis. Bremen: Universität Bremen.

Ribeiro, F. C., Carvalho, R. T. S., Cortez, P. C., De Albuquerque, V. H. C., & Filho, P. P. R. (2018). "Binary Neural Networks for Classification of Voice Commands From Throat Microphone". In: *IEEE Access* 6, pp. 70130–70144.

Rifkin, R. & Klautau, A. (Dec. 2004). "In Defense of One-Vs-All Classification". In: *Journal of Machine Learning Research* 5, pp. 101–141.

Rifkin, R. M. (2002). "Everything Old Is New Again - A Fresh Look at Historical Approaches in Machine Learning.pdf". PhD thesis. Sloan School of Management Science.

Robertson, T., Wright, F. T., & Dykstra, R. L. (1988). *Order restricted statistical inference*. Wiley Series in Probability and Mathematical Statistics.

Rocha, A. & Goldenstein, S. K. (Feb. 2014). "Multiclass From Binary: Expanding One-Versus-All, One-Versus-One and ECOC-Based Approaches". In: *IEEE Transactions on Neural Networks and Learning Systems* 25.2, pp. 289–302.

Rossi, P. E. (2014). *Bayesian Non- and Semi-parametric Methods and Applications*. Princeton: Princeton University Press.

Saez, J. A., Galar, M., & Krawczyk, B. (2019). "Addressing the Overlapping Data Problem in Classification Using the One-vs-One Decomposition Strategy". In: *IEEE Access* 7, pp. 83396–83411.

Sáez, J. A., Galar, M., Luengo, J., & Herrera, F. (2012). "A First Study on Decomposition Strategies with Data with Class Noise Using Decision Trees". In: *Hybrid Artificial Intelligent Systems*. Ed. by Hutchison, D. et al. Vol. 7209. Berlin, Heidelberg: Springer Berlin Heidelberg, pp. 25–35.

Sáez, J. A., Galar, M., Luengo, J., & Herrera, F. (Jan. 2014). "Analyzing the presence of noise in multi-class problems: alleviating its influence with the One-vs-One decomposition". In: *Knowledge and Information Systems* 38.1, pp. 179–206.

Scott, D. W. & Sain, S. R. (2005). "Multidimensional Density Estimation". In: *Handbook of Statistics*. Vol. 24. Elsevier, pp. 229–261.

Seo, S., Seo, P. H., & Han, B. (Apr. 2019). "Learning for Single-Shot Confidence Calibration in Deep Neural Networks through Stochastic Inferences". In: *arXiv:1809.10877*. arXiv: 1809.10877.

Shafer, G. (1976). *A Mathematical Theory of Evidence*. Princeton and London: Princeton University Press.

Shafer, G. (Dec. 2016). "A Mathematical Theory of Evidence turns 40". In: *International Journal of Approximate Reasoning* 79, pp. 7–25.

Sheather, S. J. & Jones, M. C. (July 1991). "A Reliable Data-Based Bandwidth Selection Method for Kernel Density Estimation". In: *Journal of the Royal Statistical Society: Series B (Methodological)* 53.3, pp. 683–690.

Silverman, B. W. (1986). *Density Estimation for Statistics and Data Analysis*. London: Chapman & Hall.

Simon, R. (July 2014). "Class probability estimation for medical studies". In: *Biometrical Journal* 56.4, pp. 597–600.

Smets, P. & Kennes, R. (Apr. 1994). "The transferable belief model". In: *Artificial Intelligence* 66.2, pp. 191–234.

Sun, W. W., Cheng, G., & Liu, Y. (2018). "Stability Enhanced Large-Margin Classifier Selection". In: *Statistica Sinica*.

Szegedy, C., Vanhoucke, V., Ioffe, S., Shlens, J., & Wojna, Z. (June 2016). "Rethinking the Inception Architecture for Computer Vision". In: *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. Las Vegas, NV, USA: IEEE, pp. 2818–2826.

Tchamova, A. & Dezert, J. (Sept. 2012). "On the Behavior of Dempster's Rule of Combination and the Foundations of Dempster-Shafer Theory". In: *2012 IEEE 6th International Conference Intelligent Systems*. Sofia, Bulgaria: IEEE, pp. 108–113.

Tran, G.-L., Bonilla, E. V., Cunningham, J. P., Michiardi, P., & Filippone, M. (May 2018). "Calibrating Deep Convolutional Gaussian Processes". In: *arXiv:1805.10522 [cs, stat]*. arXiv: 1805.10522.

Tsujinishi, D., Koshiba, Y., & Abe, S. (2004). "Why pairwise is better than one-against-all or all-at-once". In: *2004 IEEE International Joint Conference on Neural Networks (IEEE Cat. No.04CH37541)*. Budapest, Hungary: IEEE, pp. 693–698.

Venables, W. N. & Ripley, B. D. (2002). *Modern Applied Statistics with S*. Springer.

Voorbraak, F. (Mar. 1991). "On the justification of Dempster's rule of combination". In: *Artificial Intelligence* 48.2, pp. 171–197.

Walt, C. M. van der & Barnard, E. (2017). "Variable Kernel Density Estimation in High-Dimensional Feature Spaces". In: *Proceedings of the Thirty-First AAAI Conference on Artificial Intelligence*. San Francisco, California, USA, pp. 2674–2680.

Wang, J., Shen, X., & Liu, Y. (Jan. 2008). "Probability estimation for large-margin classifiers". In: *Biometrika* 95.1, pp. 149–167.

Wang, P. (1994). "A Defect in Dempster-Shafer Theory". In: *Uncertainty Proceedings 1994*. Elsevier, pp. 560–566.

Wang, Y., Li, L., & Dang, C. (Aug. 2019). "Calibrating Classification Probabilities with Shape-Restricted Polynomial Regression". In: *IEEE Transactions on Pattern Analysis and Machine Intelligence* 41.8, pp. 1813–1827.

Wang, Z. & Xue, X. (Mar. 2014). "Multi-Class Support Vector Machine". In: *Support Vector Machines Applications*. Ed. by Ma, Y. & Guo, G. Springer Publishing Company, Incorporated.

Wasserman, L. (2006). *All of nonparametric statistics*. 1st ed., 3rd print. Springer texts in statistics. OCLC: 69992043. New York, NY: Springer.

Weston, J. & Watkins, C. (May 1998). *Multi-class Support Vector Machines*. Tech. rep. CSD-TR-98-04. Egham, Surrey TW20 0EX, England: Department of Computer Science, Royal Holloway University of London, p. 10.

Wilson, N. (1993). "The Assumptions Behind Dempster's Rule". In: *Uncertainty in Artificial Intelligence*. Elsevier, pp. 527–534.

Witt, C., Bux, M., Gusew, W., & Leser, U. (May 2019). "Predictive Performance Modeling for Distributed Computing using Black-Box Monitoring and Machine Learning". In: *Information Systems* 82, pp. 33–52.

Wu, D., Li, C., Chen, J., You, D., & Xia, X. (2014). "A Novel Multi-class Support Vector Machines Using Probability Voting Strategy and Its Application on Fault Diagnosis of Gearbox". In: *International Journal of Performability Engineering* 10.2, p. 14.

Wu, T.-F., Lin, C.-J., & Weng, R. C. (Jan. 2004). "Probability Estimates for Multi-class Classification by Pairwise Coupling". In: *Journal of Machine Learning Research* 5, pp. 975–1005.

Xiao, H., Xiao, Z., & Wang, Y. (June 2016). "Ensemble classification based on supervised clustering for credit scoring". In: *Applied Soft Computing* 43, pp. 73–86.

Xu, P., Su, X., Mahadevan, S., Li, C., & Deng, Y. (Oct. 2014). "A non-parametric method to determine basic probability assignment for classification problems". In: *Applied Intelligence* 41.3, pp. 681–693.

Xu, P., Davoine, F., Zha, H., & Denœux, T. (May 2016). "Evidential calibration of binary SVM classifiers". In: *International Journal of Approximate Reasoning* 72, pp. 55–70.

Xu, R., Qian, T., & Kwan, C. (2005). "An Improved Optimal Pairwise Coupling Classifier". In: *Advances in Neural Networks – ISNN 2005*. Ed. by Hutchison, D. et al. Vol. 3497. Berlin, Heidelberg: Springer Berlin Heidelberg, pp. 32–38.

Xu, T. & Wang, J. (Sept. 2012). "An efficient model-free estimation of multiclass conditional probability". In: *arXiv:1209.4951 [cs, stat]*. arXiv: 1209.4951.

Yang, F. & Barber, R. F. (2019). "Contraction and uniform convergence of isotonic regression". In: *Electronic Journal of Statistics* 13.1, pp. 646–677.

Yang, G., Destercke, S., & Masson, M.-H. (Dec. 2017). "Cautious classification with nested dichotomies and imprecise probabilities". In: *Soft Computing* 21.24, pp. 7447–7462.

Yang, X., Yu, Q., He, L., & Guo, T. (Aug. 2013). "The one-against-all partition based binary tree support vector machine algorithms for multi-class classification". In: *Neurocomputing* 113, pp. 1–7.

Zadeh, L. A. (Mar. 1979). *On the Validity of Dempster's Rule of Combination of Evidence*. Tech. rep. UCB/ERL M79/24. Berkeley, CA: EECS Department, University of California, Berkeley.

Zadeh, L. A. (1984). "Book review: A mathematical theory of evidence". In: *The AI Magazine* 5.3, pp. 81–83.

Zadeh, L. A. (1986). "A Simple View of the Dempster-Shafer Theory of Evidence and its Implication for the Rule of Combination". In: *The AI Magazine* 7.2, pp. 85–90.

Zadrozny, B. (2002). "Reducing multiclass to binary by coupling probability estimates". In: *Advances in Neural Information Processing Systems 14*. Ed. by Dietterich, T. G., Becker, S., & Ghahramani, Z. MIT Press, pp. 1041–1048.

Zadrozny, B. & Elkan, C. (2001a). "Learning and making decisions when costs and probabilities are both unknown". In: *Proceedings of the seventh ACM SIGKDD international conference on Knowledge discovery and data mining - KDD '01*. San Francisco, California: ACM Press, pp. 204–213.

Zadrozny, B. & Elkan, C. (2001b). "Obtaining calibrated probability estimates from decision trees and naive Bayesian classifiers". In: *Proceedings of the Eighteenth International Conference on Machine Learning*, p. 8.

Zadrozny, B. & Elkan, C. (2002). "Transforming Classifier Scores into Accurate Multiclass Probability Estimates". In: *Proceedings of the eighth ACM SIGKDD international conference on Knowledge discovery and data mining*. Edmonton, Alberta, Canada, p. 6.

Zambom, A. Z. & Dias, R. (Dec. 2012). "A Review of Kernel Density Estimation with Applications to Econometrics". In: *arXiv:1212.2812 [stat]*. arXiv: 1212.2812.

Zhang, C.-H. (Apr. 2002). "Risk bounds in isotonic regression". In: *The Annals of Statistics* 30.2, pp. 528–555.

Zhang, J. & Yang, Y. (2004). "Probabilistic Score Estimation with Piecewise Logistic Regression". In: *Proceedings of the 21st International Conference on Machine learning*. Banff, Alberta, Canada, p. 8.

Zhang, J., Zhao, X., & Du, L. (Dec. 2013). "Solving multi-class problems by data-driven topology-preserving output codes". In: *Neurocomputing* 121, pp. 556–568.

Zhang, Z., Krawczyk, B., Garcìa, S., Rosales-Pérez, A., & Herrera, F. (Aug. 2016). "Empowering one-vs-one decomposition with ensemble learning for multi-class imbalanced data". In: *Knowledge-Based Systems* 106, pp. 251–263.

Zhao, H., Sun, D., Zhao, M., & Cheng, S. (Apr. 2016). "A Multi-Classification Method of Improved SVM-based Information Fusion for Traffic Parameters Forecasting". In: *PROMET - Traffic&Transportation* 28.2, pp. 117–124.

Zhong, L., Li, Z., Ding, Z., Guo, C., & Song, H. (2008). "Multiple Sources Data Fusion Strategies Based on Multi-class Support Vector Machine". In: *Advances in Neural Networks - ISNN 2008*. Ed. by Sun, F., Zhang, J., Tan, Y., Cao, J., & Yu, W. Vol. 5263. Berlin, Heidelberg: Springer Berlin Heidelberg, pp. 715–722.

Zhou, J., Peng, H., & Suen, C. Y. (Jan. 2008). "Data-driven decomposition for multi-class classification". In: *Pattern Recognition* 41.1, pp. 67–76.