**Hasso–Plattner–Institut für Softwaresystemtechnik
an der Universität Potsdam**

X-tracking the Usage Interest on Web Sites

Dissertation

zur Erlangung des akademischen Grades
"Doctor rerum naturalium"
(Dr. rer. nat.)
am Fachgebiet Internet Technologien und Systeme

eingereicht an der
Mathematisch–Naturwissenschaftlichen Fakultaet
der Universitaet Potsdam

von
Long Wang

Potsdam, December 8, 2009

Contents

Acknowledgments

First and foremost, I greatly appreciate my supervisor, Prof. Dr. Christoph Meinel, not only for his help to get the financial support with which I could go on with my PhD study in Hasso Plattner Institute and in Germany, but also for his encouragement, guidance and discussions throughout my studies from which I learnt the spirits of research and the skills on how to work in a team. I give my great respect on his rich vitality on the work, and specially on his contributions on the cooperations between Germany and China in a long term.

I thank my parents, Xiuzhi Zhang and Baonuan Wang, for their support and understanding for my not visiting them in the last several years; for their encouragement and warm letters when I am depressed in the study. I thank a lot my colleagues and my Chinese colleagures in our research group, for sharing their success, happiness and sadness during working, for our cooperations in a team. I thank as well other good friends in daily life, for giving me the warm home feeling in Germany, for helping me understand, learn and integrate with this culture.

Abstract

The exponential expanding of the numbers of web sites and Internet users makes WWW the most important global information resource. From information publishing and electronic commerce to entertainment and social networking, the Web allows an inexpensive and efficient access to the services provided by individuals and institutions. The basic units for distributing these services are the web sites scattered throughout the world. However, the extreme fragility of web services and content, the high competence between similar services supplied by different sites, and the wide geographic distributions of the web users drive the urgent requirement from the web managers to track and understand the usage interest of their web customers. This thesis, ”**X-tracking the Usage Interest on Web Sites**”, aims to fulfill this requirement. ”X” stands two meanings: one is that the usage interest differs from various web sites, and the other is that usage interest is depicted from multi aspects: internal and external, structural and conceptual, objective and subjective. ”Tracking” shows that our concentration is on locating and measuring the differences and changes among usage patterns.

This thesis presents the methodologies on discovering usage interest on three kinds of web sites: the public information portal site, e-learning site that provides kinds of streaming lectures and social site that supplies the public discussions on IT issues. On different sites, we concentrate on different issues related with mining usage interest.

The educational information portal sites were the first implementation scenarios on discovering usage patterns and optimizing the organization of web services. In such cases, the usage patterns are modeled as frequent page sets, navigation paths, navigation structures or graphs. However, a necessary requirement is to rebuild the individual behaviors from usage history. We give a systematic study on how to rebuild individual behaviors. Besides, this thesis shows a new strategy on building content clusters based on pair browsing retrieved from usage logs. The difference between such clusters and the original web structure displays the distance between the destinations from usage side and the expectations from design side. Moreover, we study the problem on tracking the changes of usage patterns in their life cycles. The changes are described from internal side integrating conceptual and structure features, and from external side for the physical features; and described from local side measuring the difference between two time spans, and global side showing the change tendency along the life cycle. A platform, Web-Cares, is developed to discover the usage interest, to measure the difference between usage interest and site expectation and to track the changes of usage patterns.

E-learning site provides the teaching materials such as slides, recorded lecture videos and ex-

ercise sheets. We focus on discovering the learning interest on streaming lectures, such as real medias, mp4 and flash clips. Compared to the information portal site, the usage on streaming lectures encapsulates the variables such as viewing time and actions during learning processes. The learning interest is discovered in the form of answering 6 questions, which covers finding the relations between pieces of lectures and the preference among different forms of lectures. We prefer on detecting the changes of learning interest on the same course from different semesters. The differences on the content and structure between two courses leverage the changes on the learning interest. We give an algorithm on measuring the difference on learning interest integrated with similarity comparison between courses. A search engine, TASK-Moniminer, is created to help the teacher query the learning interest on their streaming lectures on tele-TASK site.

Social site acts as an online community attracting web users to discuss the common topics and share their interesting information. Compared to the public information portal site and e-learning web site, the rich interactions among users and web content bring the wider range of content quality, on the other hand, provide more possibilities to express and model usage interest. We propose a framework on finding and recommending high reputation articles in a social site. We observed that the reputation is classified into global and local categories; the quality of the articles having high reputation is related with the content features. Based on these observations, our framework is implemented firstly by finding the articles having global or local reputation, and secondly clustering articles based on their content relations, and then the articles are selected and recommended from each cluster based on their reputation ranks.

Long Wang
December 8, 2009

Chapter 1

Introduction

I never waste memory on things that can easily be stored and retrieved from elsewhere.

Albert Einstein (1879 - 1955)

Microsoft has made an investigation in 04.2007 [1]: over 200 students in Germany from different facilities, were asked their remarks on the usage of their universities' web sites. 90% percent students said that the web sites from their universities are very or partly useful, but only 7% students gave the positive feedback or the high remarks on the web sites from their universities. Figure 1.1 shows the results on this investigation: 82.6% students wanted online-Test and online exercise, but only 6.1% said they had such services; 47.4% students needed a virtual communication with other students, however, only 5.7% said they had such facility. The value of this investigation is not only the revealed result that the big gap between the students' expectations and the supplied services of the web sites, but the huge hardness of finding proper ways to collect and understand the online usage requirements from the students.

This direct investigation made by Microsoft reveals much valuable usage interest on the university web sites, however, for other educational sites, government or public portal sites, how can they retrieval the usage feedback on their services? Although the importance of knowing the usage interest on web sites is already well recognized, for most web sites, especially the public information portals, on which it is not known who are the right visitors, the direct investigation on the usage feedback can not be economically implemented. In this case, we can only seek other ways to discover the usage interest on the web sites.

Generally, there are **direct** and **indirect** ways to investigate the usage interest or patterns on a web site. Since the direct and indirect investigation have different strengths, they can be best considered as complements rather than replacement to each other (Yi 1989).

The **direct** way is the traditional approach, and might include:

1. observation of user interaction on a web site in a usability laboratory or in a field setting, using video and audio taping of free-form and/or predetermined tasks, recording navigation patterns and user comments for observers; and
2. using online or offline questionnaires such as the one by Microsoft mentioned above, or telephone interviews.

The primary advantage of the direct way is directness: the purpose is clear, the responses are straightforward, and the corresponding rules between consumer satisfaction and measure are unequivocal. While the disadvantages are:

[1] http://www.microsoft.com/germany/presseservice/detail.mspx?id=531887

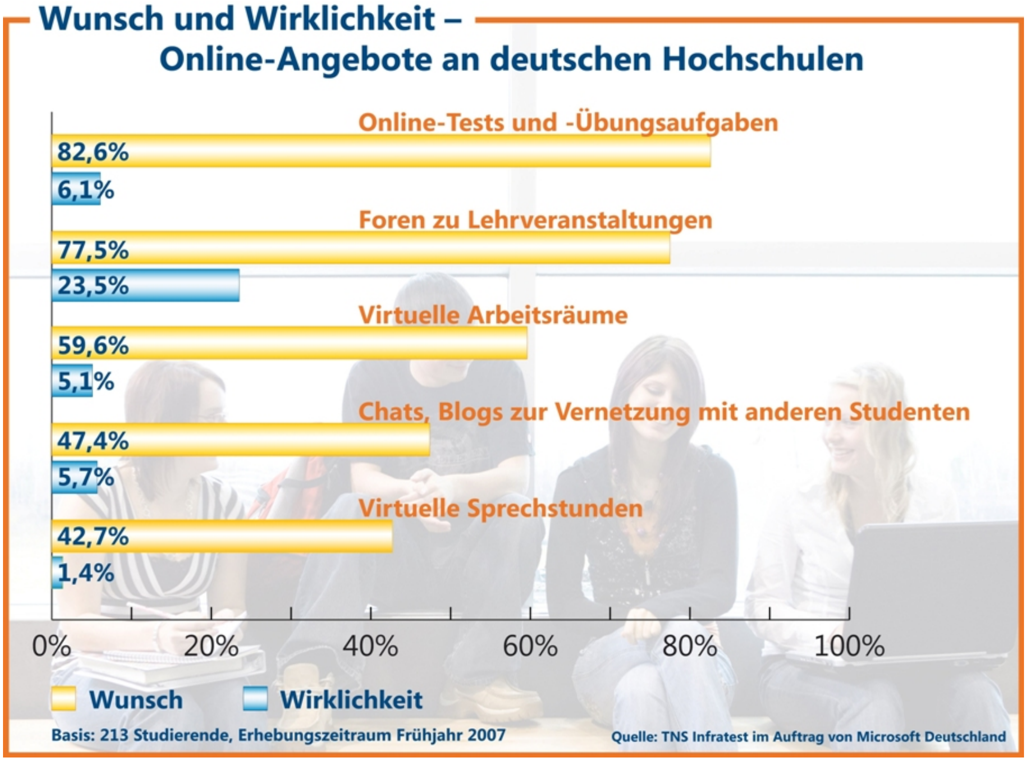


Figure 1.1: *Investigation by Microsoft on German Universities' Sites*

1. incompleteness: compared with the population of accessing the web site, the number of attendants in the direct investigation is small, which could overlook the great variance among the visitors;

2. possible fraud: an attendant in the direct investigation could distort, hide and forge his/her real behavior and preference, this is even much possible in the investigation on private information like incoming and family status;

3. low unexpectedness: the design of questionnaires and interviews is subjective, and the unexpected information or knowledge unknown before could not be found by the direct investigation; and

4. high expense: the expense of the direct investigation is a big financial and time burden for the web site.

The **indirect** ways are the methods of analyzing the visitors' interest from their usage history on a web site. The usage history records the interactions between users and the site and is usually stored as usage logs. Based on the location, the usage logs are classified into server-side and client-side logs. The usage logs are the history on how and what did the visitors interact with the web site, which remedy the disadvantages ”incompleteness” and ”possible fraud” of the direct investigation. Moreover, from the usage logs, it gives the possibility to discover the detailed, hidden and unexpected knowledge about usage interest, which can not be fulfilled by the direct investigation.

The big requirement on discovering the usage interest from usage logs, drives the birth of Web Usage Mining. Web usage mining is the automatic discovery of patterns in clickstreams and associated data collected or generated as a result of user interactions with one or more Web sites (Mobasher 2006). Web usage mining is the application of data mining in the web usage area. Data mining or KDD (Knowledge Discovery in Database) process is *"the nontrivial extraction of implicit, previously unknown, and potentially useful information from data"* (Piatetsky-Shapiro and Frawley 1991). The information discovered by data mining is represented as patterns, models or relations.

1.1 Related Works

A large number of work related to data mining has been conducted by various researchers. Much work in this area may be divided into two major categories: supervised learning and unsupervised learning. Supervised learning aims at discovering relationships between attributes and a response variable. In other words, it can be used to classify and predict unknown outputs corresponding to the known categories. On the contrary, unsupervised learning aims at creating groups that share common characteristics even though the collected data do not have pre-classified categories. Another branch of unsupervised learning is associate rule mining, which includes the frequent structures mining like sequences, trees and graphs. This section lists the related systems and methodologies on web usage mining.

WebTrends and Weblizer are two simple and commercial web analysis tools providing graphical reports on how frequently web sites are accessed. However, they focus on volume-level analysis (ex: "how many hits did this WWW page get?") rather than on the analysis of individual user-level data (ex: "how did one user navigate the site?"). In addition, most of these tools do not support the filtering mechanism in order to clean and select the target data for analysis.

Perkowitz investigated the problem of index page synthesis, which is the automatic creation of the pages that facilitate a visitor's navigation of a web site (Perkowitz and Etzioni 2000). By analyzing the web logs, their cluster mining algorithm finds collections of pages that tend to co-occur in visits and puts them under one topic. They then generate the index page consisting of links to the pages pertaining to a particular topic.

Nakayama and his colleagues tried to discover the gap between the web site's designer's expectations and visitors' behaviors (Nakayama et al. 2000). Their approach uses the inter-page conceptual relevance to estimate the former, and the inter-page access co-occurrence to estimate the latter. They focus on the web site design improvement by using multiple regression to predict hyperlink traversal frequency from page layout features.

An algorithm was given on automatically finding pages in a web site whose location is different from which the visitors expect to find (Srikant and Yang 2001). The key assumption is that the visitors will backtrack if they do not find the information where they expect it: the point from which they backtrack is the expected location for the page.

"Web utilization miner" was proposed to find interesting navigation patterns (Berendt 2005). The interestingness criteria for navigation patterns are dynamically specified by the human expert using a mining language which supports the specification of statistical, structural and textual criteria.

As will be explained later in this thesis, the usage interest has a huge scope, which means that the usage interest varies on different services, has different forms, differs on individuals and is described by multiple aspects. The difficulties and challenges web usage mining faces are:

1. diverse web content and services: web x.0 covers various types of content from text, picture, voice and video, which supply different services such as information portals, e-learning, online-shopping, or online social communities. Users behave different ways in searching and browsing the different formats content and services;

2. weak relations between user and site: visitors could access the web site at any time from any place through any path. And during their visiting, they even do not have any clear idea about what they want from the web. On the other hand, it is not easy for the site to discriminate different users. WWW brings great freedom and convenience for users and sites, and great varieties among them as well. So the relation between supply and demand becomes weak and vague; and

3. complicated usage behaviors: hyperlink and back tracking are the two important characteristics in web navigation, which make users' activities more complicated. For the same visited content, different users can access them with different patterns. On the other hand, the users' behaviors are recorded as visiting sequences in web logs, which can not exactly and directly reflect the users' real behaviors and web site structures.

1.2 Contributions

In this thesis, we investigate the methodologies on discovering the usage interest on three different kinds of web sites. The sites are introduced in the followings:

- public information portal site: a portal site is used by a university, government or company to present the publics information about their organizations, services or products. Such web site supplies very few interactions between users and web site. We take HPI site (www.hpi.uni-potsdam.de) and ECCC site (eccc.hpi-web.de) as the examples of tracking usage interest. HPI site is an educational portal site for Hasso Plattner Institute, and ECCC site is an electronic journal on ideas, techniques, and research in computational complexity;

- e-learning site: it supplies the teaching materials in forms of slides, videos to the students via a web site. The content is always organized based on the structure of the courses or the events. The learning interest from online students is shown from their staying time and actions during the learning processes. We implemented the learning interest mining method on tele-TASK site (www.tele-task.de), which provides the lecture videos in different formats and is proved as an efficient e-learning platform; and

- online social site: web 2.0 witnesses plenty of social sites and online communities, in which the interactions between users and content and among groups of users bring the new features and frameworks to discover the usage interest. We discuss this problem based on IT-Gipfelblog site (it-gipfelblog.hpi-web.de), which is a webblog in German discussing the topics on information and communication technologies.

Three projects have been implemented to discover the usage interest on different kinds of web sites: Web-Cares, TASK−Moniminer and Re−Blog. The contributions of this thesis are listed:

1. to solve the problem of recovering individual navigation behaviors in browsing an information portal site, which is the necessary premise for the posterior usage pattern mining;
2. to give a general and unified method on tracking the changes of web navigation patterns, which not only locates the nearest version of a pattern in posterior time spans, but measures its internal and external variances from the structural and semantic side;
3. to model the learning interest on browsing kinds of streaming lectures and to discover the learning interest by answering several questions;
4. to measure the changes of learning interest on the same courses from different semesters, which integrates the variance of usage interest and the difference of the courses in different years; and
5. to present a framework on evaluating and recommending high reputation articles in a social site, which is based on the proof that the reputation the articles received is classified into local and global categories. The proposed framework considers the balance between the concept and usage feedback, between the interest from major users and minor users.

1.3 Thesis Structure

This thesis illustrates the topic on ″X-tracking the Usage Interest on Web Sites″ using three parts:

1. Part I is ″Discovering the Changes of Usage Interest on a Portal Site″, which answers the question of discovering usage interest on a public information portal site, such as HPI site (www.hpi.uni-potsdam.de) and ECCC site (eccc.hpi-web.de);
2. Part II is ″Mining the Learning Interest in a Web-Streaming E-learning Site″, which discusses mining the learning interest on the e-learning site such as tele-TASK; and
3. Part III is ″Recommending High Reputation Articles in a Social Site″, which provides a framework on evaluating and recommending high reputation articles in a social site by using the example ″IT-Gipfelblog″.

Every part starts with a chapter introducing the project on implementing the mining methodology on one kind of web site, and proceeds by the chapters explaining the modeling and algorithms on measuring usage interest and mining usage patterns, and ends with the chapter discussing the experiments and the findings of our projects.

Part I

Discovering the Changes of Usage Interest on a Portal Site

Chapter 2

Web-Cares: A Platform to Track the Web Usage Interest

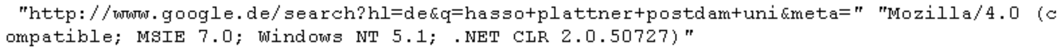
Web-Cares is a GUI-based platform aiming to mine usage interest on a portal site, to track the changes of usage interest over a period of time, and to discover the gap between web structures and usage patterns. Such information will help web managers to know the interest of their web visitors and to further optimize their web structures and services.

Chapter Organization In Section 2.1 we give the data resource that Web-Cares processes. We describe the outputs and the work flow of Web-Cares in Section 2.2. We present the summary of this chapter in Section 2.3.

2.1 Input of Web-Cares

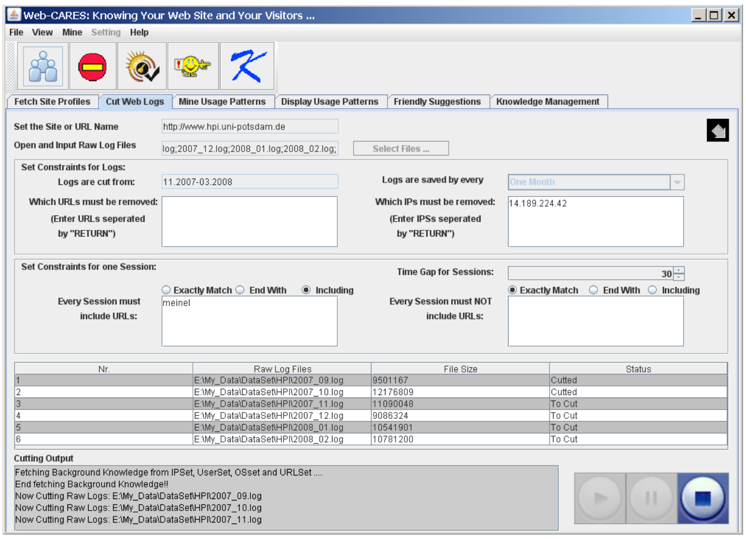
The main source data web-cares takes are the usage log files on the server side recording the information requests from its web visitors. We concentrate on the log files confirming to the W3C Common Log Format extended by the Agent Log and the Referee Log. Figure 2.1 gives an example of a client HTTP request recorded in server logs, it tells that when, who and how was the request on ″willkommen.html″ asked: a user found this page from Google search engine by using ″hasso plattner potsdam uni″ keywords, and his client browser is Mozilla and 200 is the return code showing that this request was successfully answered. Every request is written as a unique line in a chronological order in the server log files.

In our system, we treat ″willkommen.html″ in Figure 2.1 a **pageview**. A **pageview** is an aggregate representation of a collection of Web objects contributing to the display on a user's browser resulting from a single user action (such as a click-through). Thus, each pageview can be viewed as a collection of Web objects or resources representing a specific ″user event,″ e.g., reading an article, viewing a product page, or adding a product to the shopping cart. In the server logs, after this logline as in Figure 2.1, there are other requests on the images and formats for displaying the page ″willkommen.html″ on the client side, but we don't treat the requested images and format files as pageviews. For simplicity, **pageview** is replaced by ″**page**″ in this thesis. A logline on a page request is an **access**, which describes who, when, what and how was a page requested. We call a number of **accesses** asked by the same user within a time period (e.g. 30 minutes) a **session**.



```
77.224.131.65 - - [31/Dec/2008:23:55:07 +0100] "GET /willkommen.html HTTP/1.1" 200 3876
 "http://www.google.de/search?hl=de&q=hasso+plattner+postdam+uni&meta=" "Mozilla/4.0 (c
ompatible; MSIE 7.0; Windows NT 5.1; .NET CLR 2.0.50727)"
```

Figure 2.1: *An example: a client HTTP request recorded in logs*



Set the Site or URL Name: http://www.hpi.uni-potsdam.de

Open and Input Raw Log Files: log;2007_12.log;2008_01.log;2008_02.log;

Set Constraints for Logs:

Logs are cut from: 11.2007-03.2008

Logs are saved by every: One Month

Which URLs must be removed: (Enter URLs seperated by "RETURN")

Which IPs must be removed: (Enter IPSs seperated by "RETURN") 14.189.224.42

Set Constraints for one Session:

Time Gap for Sessions: 30

Every Session must include URLs: ○ Exactly Match ○ End With ● Including
meinel

Every Session must NOT include URLs: ● Exactly Match ○ End With ○ Including

Nr.	Raw Log Files	File Size	Status
1	E:\My_Data\DataSet\HPI\2007_09.log	9501167	Cutted
2	E:\My_Data\DataSet\HPI\2007_10.log	12176809	Cutted
3	E:\My_Data\DataSet\HPI\2007_11.log	11090048	To Cut
4	E:\My_Data\DataSet\HPI\2007_12.log	9086324	To Cut
5	E:\My_Data\DataSet\HPI\2008_01.log	10541901	To Cut
6	E:\My_Data\DataSet\HPI\2008_02.log	10781200	To Cut

Cutting Output

```
Fetching Background Knowledge from IPSet, UserSet, OSset and URLSet ....
End fetching Background Knowledge!!
Now Cutting Raw Logs: E:\My_Data\DataSet\HPI\2007_09.log
Now Cutting Raw Logs: E:\My_Data\DataSet\HPI\2007_10.log
Now Cutting Raw Logs: E:\My_Data\DataSet\HPI\2007_11.log
```

Figure 2.2: *Data preparation: log filtering and session reconstruction*

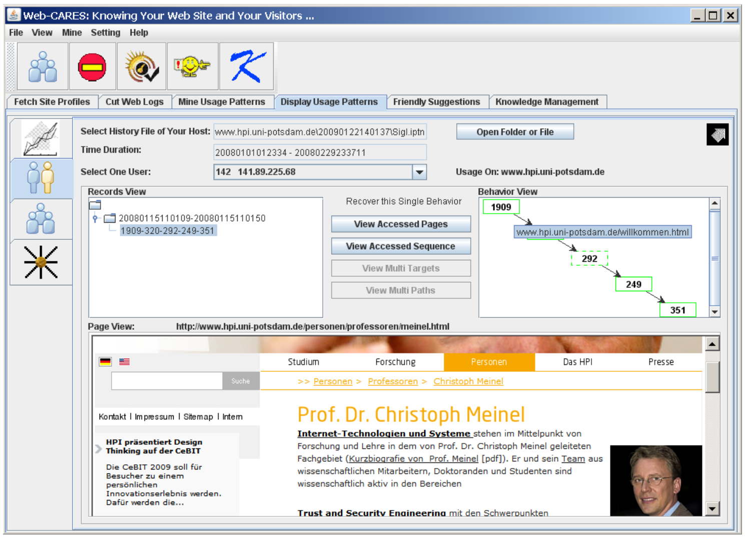
2.2 Work Flow of Web-Cares

Generally, data mining includes four steps: data preparation, pattern mining, and pattern analysis and pattern application. As an implementation of data mining, Web-Cares obeys this process.

Step 1: Data Preparation To discover the usage interest, Web-Cares firstly cleans the server logs and reconstructs the sessions of human visitors. The target sessions are selected by setting the constraints of IP or URLs and other conditions. This step generates the target session set for usage mining, in which a session can be simply described as a sequence of web pages visited by a user. Figure 2.2 shows the interface of data preparation in this step. Figure 2.3 helps us to query the usage history of a single user.

Step 2: Usage Pattern Mining Usage behavior can be described in different usage models, depending on the aspects of interactions between the visitors and web pages. For example, if the frequently co-accessed pages are to be discovered, association rules need to be used; when the frequently click sequences are asked, sequence patterns satisfy this requirement; if our target is to find the pages leading to different navigation paths, frequent tree structures should be modeled; if we want to extract the user clusters having similar navigation interest, clustering is the choice. Web-Cares fulfills these possibilities, as shown in Figure 2.4.

Step 3: Pattern Investigation The mined patterns are greatly condensed and restructured compared to the original log files. However, mining algorithms only give the possibility to find these condensed and structured information, interesting and useful patterns need to be further interpreted and filtered. To judge if a pattern is useful, a friendly query and intuitive display are needed. Web-Cares tries to bridge the gap between the patterns and human understanding. Figure 2.5 displays the usage patterns in a graphic way.



Select History File of Your Host: www.hpi.uni-potsdam.de\20090122140137\Sigl.iptn

Time Duration: 20080101012334 - 20080229233711

Select One User: 142 141.89.225.68

Usage On: www.hpi.uni-potsdam.de

Records View

20080115110109-20080115110150

1909-320-292-249-351

Behavior View

www.hpi.uni-potsdam.de/willkommen.html

Page View: http://www.hpi.uni-potsdam.de/personen/professoren/meinel.html

>> Personen > Professoren > Christoph Meinel

Prof. Dr. Christoph Meinel

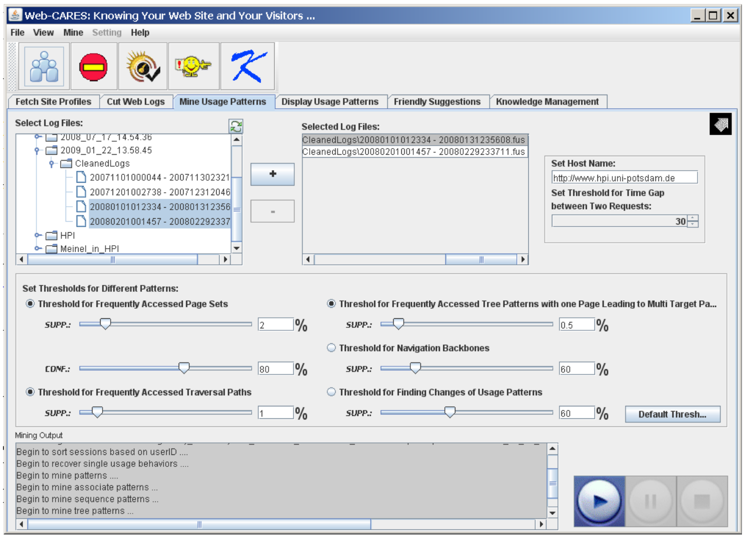
Internet-Technologien und Systeme stehen im Mittelpunkt von Forschung und Lehre in dem von Prof. Dr. Christoph Meinel geleiteten Fachgebiet (Kurzbiografie von Prof. Meinel [pdf]). Er und sein Team aus wissenschaftlichen Mitarbeitern, Doktoranden und Studenten sind wissenschaftlich aktiv in den Bereichen

Trust and Security Engineering mit den Schwerpunkten

HPI präsentiert Design Thinking auf der CeBIT

Die CeBIT 2009 soll für Besucher zu einem persönlichen Innovationserlebnis werden. Dafür werden die...

Figure 2.3: *Individual usage activity*



Select Log Files:

```
2008_07_17_14.54.36
2009_01_22_13.58.45
  CleanedLogs
    20071101000044 - 200711302321
    20071201002738 - 200712312046
    20080101012334 - 200801312356
    20080201001457 - 200802292337
HPI
Meinel_in_HPI
```

Selected Log Files:

```
CleanedLogs\20080101012334 - 20080131235608.fus
CleanedLogs\20080201001457 - 20080229233711.fus
```

Set Host Name: http://www.hpi.uni-potsdam.de

Set Threshold for Time Gap between Two Requests: 30

Set Thresholds for Different Patterns:

● Threshold for Frequently Accessed Page Sets

SUPP.: 2 %

CONF.: 80 %

● Threshold for Frequently Accessed Traversal Paths

SUPP.: 1 %

● Threshol for Frequently Accessed Tree Patterns with one Page Leading to Multi Target Pa...

SUPP.: 0.5 %

○ Threshold for Navigation Backbones

SUPP.: 60 %

○ Threshold for Finding Changes of Usage Patterns

SUPP.: 60 %

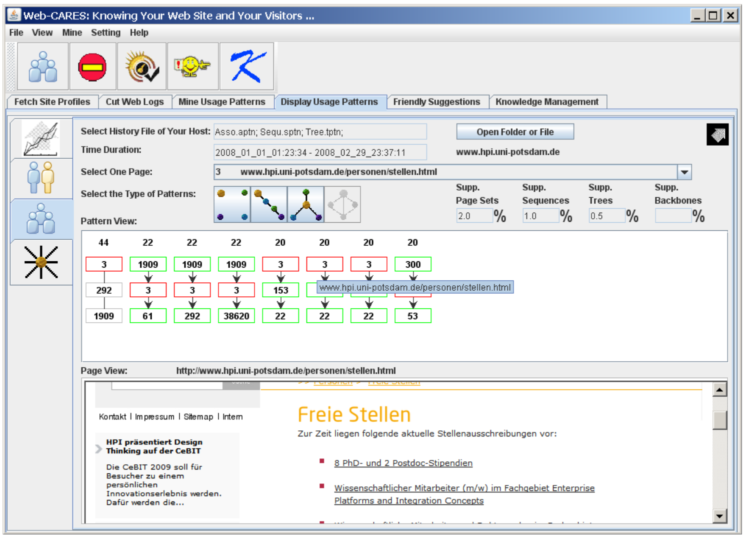
Mining Output

```
Begin to sort sessions based on userID ....
Begin to recover single usage behaviors ....
Begin to mine patterns ....
Begin to mine associate patterns ...
Begin to mine sequence patterns ...
Begin to mine tree patterns ...
```

Figure 2.4: *Usage pattern mining*

A web manager may need to be aware of the gap between his web site structure and the commonly used navigation structures, Web-Cares presents this functionality. To display this gap, Web-Cares fetches the web site structure as shown in Figure 2.6. Figure 2.7 gives a view on investigating the difference between site structures and navigation patterns: selecting a URL, Web-Cares displays the pages linked to and from this URL, as well the pages usually visited by users before and after this URL.

To understand the concepts of mined patterns, Web-Cares gives the interface for URL annotation. Though the keywords used in computing conceptual difference are extracted automatically from web pages by our parser, Web-Cares still gives the manual annotation interface. This is



Select History File of Your Host: Asso.aptn; Sequ.sptn; Tree.tptn;

Time Duration: 2008_01_01_01:23:34 - 2008_02_29_23:37:11

www.hpi.uni-potsdam.de

Select One Page: 3 www.hpi.uni-potsdam.de/personen/stellen.html

Select the Type of Patterns:

Supp. Page Sets 2.0 %

Supp. Sequences 1.0 %

Supp. Trees 0.5 %

Supp. Backbones %

Pattern View:

Page View: http://www.hpi.uni-potsdam.de/personen/stellen.html

Kontakt | Impressum | Sitemap | Intern

HPI präsentiert Design Thinking auf der CeBIT

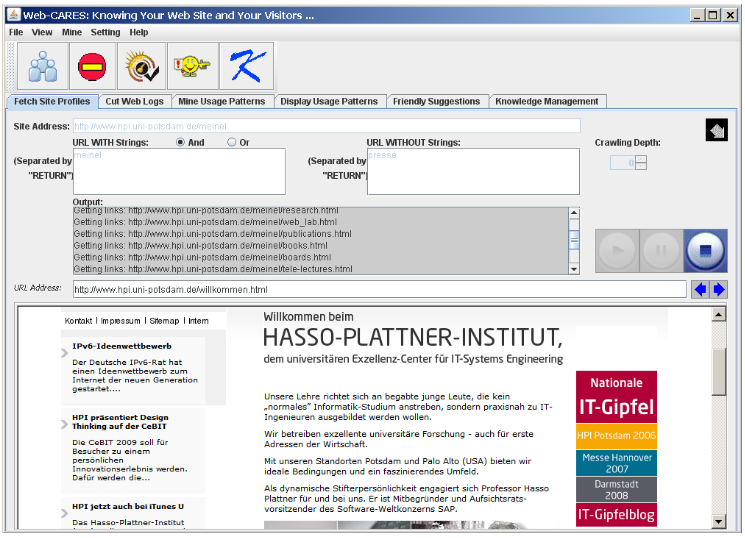
Die CeBIT 2009 soll für Besucher zu einem persönlichen Innovationserlebnis werden. Dafür werden die...

Freie Stellen

Zur Zeit liegen folgende aktuelle Stellenausschreibungen vor:

- 8 PhD- und 2 Postdoc-Stipendien
- Wissenschaftlicher Mitarbeiter (m/w) im Fachgebiet Enterprise Platforms and Integration Concepts

Figure 2.5: *Frequent usage patterns*



Site Address: http://www.hpi.uni-potsdam.de/meinel

URL WITH Strings: ● And ○ Or

(Separated by "RETURN") meinel

URL WITHOUT Strings:

(Separated by "RETURN") presse

Crawling Depth: 0

Output:

```
Getting links: http://www.hpi.uni-potsdam.de/meinel/research.html
Getting links: http://www.hpi.uni-potsdam.de/meinel/web_lab.html
Getting links: http://www.hpi.uni-potsdam.de/meinel/publications.html
Getting links: http://www.hpi.uni-potsdam.de/meinel/books.html
Getting links: http://www.hpi.uni-potsdam.de/meinel/boards.html
Getting links: http://www.hpi.uni-potsdam.de/meinel/tele-lectures.html
```

URL Address: http://www.hpi.uni-potsdam.de/willkommen.html

Kontakt | Impressum | Sitemap | Intern

IPv6-Ideenwettbewerb

Der Deutsche IPv6-Rat hat einen Ideenwettbewerb zum Internet der neuen Generation gestartet....

HPI präsentiert Design Thinking auf der CeBIT

Die CeBIT 2009 soll für Besucher zu einem persönlichen Innovationserlebnis werden. Dafür werden die...

HPI jetzt auch bei iTunes U

Das Hasso-Plattner-Institut

Willkommen beim

HASSO-PLATTNER-INSTITUT,

dem universitären Exzellenz-Center für IT-Systems Engineering

Unsere Lehre richtet sich an begabte junge Leute, die kein „normales" Informatik-Studium anstreben, sondern praxisnah zu IT-Ingenieuren ausgebildet werden wollen.

Wir betreiben exzellente universitäre Forschung - auch für erste Adressen der Wirtschaft.

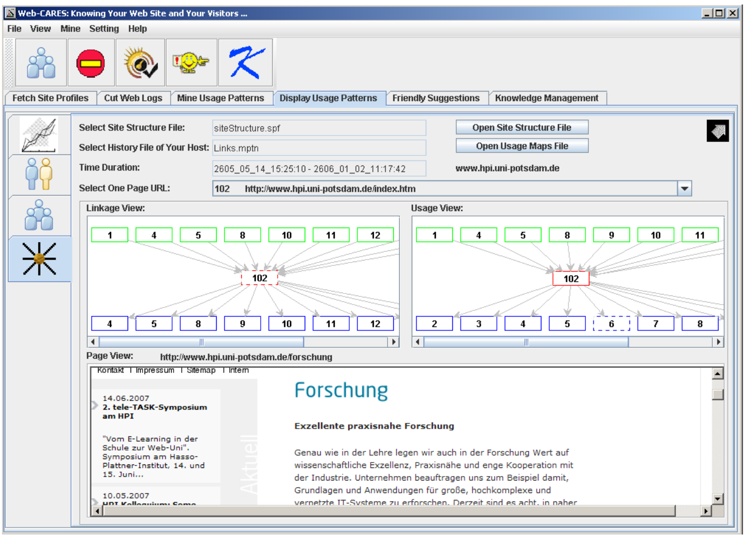
Mit unseren Standorten Potsdam und Palo Alto (USA) bieten wir ideale Bedingungen und ein faszinierendes Umfeld.

Als dynamische Stifterpersönlichkeit engagiert sich Professor Hasso Plattner für und bei uns. Er ist Mitbegründer und Aufsichtsrats-vorsitzender des Software-Weltkonzerns SAP.

Figure 2.6: *Structure crawling*

especially necessary to investigate the semantic dependency among web pages, as displayed in Figure 2.8.

Step 4: Pattern Application After the useful usage patterns have been understood and accepted, the next step is pattern application. Usually, there is little distance before making the strategy on implementing the acceptable patterns and suggestions, because it would cause the deletion, modification and optimization on the current organization and service. And this will cost some extra expense and may have some risks. It is an acceptable way to reach the compromised solutions by the intern negotiation within a team.



Select Site Structure File: siteStructure.spf

Select History File of Your Host: Links.mptn

Time Duration: 2605_05_14_15:25:10 - 2606_01_02_11:17:42

www.hpi.uni-potsdam.de

Select One Page URL: 102 http://www.hpi.uni-potsdam.de/index.htm

Page View: http://www.hpi.uni-potsdam.de/forschung

Kontakt | Impressum | Sitemap | Intern

14.06.2007
2. tele-TASK-Symposium am HPI

"Vom E-Learning in der Schule zur Web-Uni". Symposium am Hasso-Plattner-Institut, 14. und 15. Juni...

10.05.2007

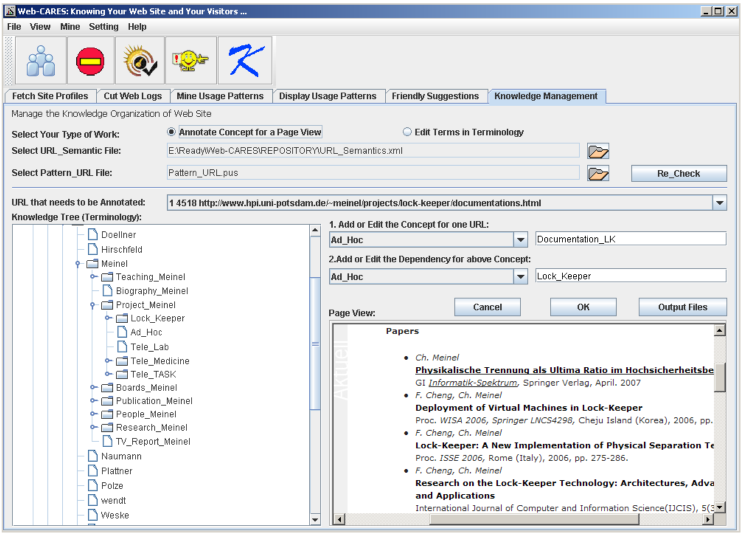
Aktuell

Forschung

Exzellente praxisnahe Forschung

Genau wie in der Lehre legen wir auch in der Forschung Wert auf wissenschaftliche Exzellenz, Praxisnähe und enge Kooperation mit der Industrie. Unternehmen beauftragen uns zum Beispiel damit, Grundlagen und Anwendungen für große, hochkomplexe und vernetzte IT-Systeme zu erforschen. Derzeit sind es acht, in naher

Figure 2.7: *Difference between usage behavior and site structure*



Manage the Knowledge Organization of Web Site

Select Your Type of Work: ● Annotate Concept for a Page View ○ Edit Terms in Terminology

Select URL_Semantic File: E:\Ready\Web-CARES\REPOSITORY\URL_Semantics.xml

Select Pattern_URL File: Pattern_URL.pus

URL that needs to be Annotated: 1 4518 http://www.hpi.uni-potsdam.de/~meinel/projects/lock-keeper/documentations.html

Knowledge Tree (Terminology):

```
Doellner
Hirschfeld
Meinel
  Teaching_Meinel
  Biography_Meinel
  Project_Meinel
    Lock_Keeper
    Ad_Hoc
    Tele_Lab
    Tele_Medicine
    Tele_TASK
  Boards_Meinel
  Publication_Meinel
  People_Meinel
  Research_Meinel
  TV_Report_Meinel
Naumann
Plattner
Polze
wendt
Weske
```

1. Add or Edit the Concept for one URL: Ad_Hoc Documentation_LK

2.Add or Edit the Dependency for above Concept: Ad_Hoc Lock_Keeper

Page View:

Papers

- *Ch. Meinel*
  Physikalische Trennung als Ultima Ratio im Hochsicherheitsbe
  GI *Informatik-Spektrum*, Springer Verlag, April. 2007
- *F. Cheng, Ch. Meinel*
  Deployment of Virtual Machines in Lock-Keeper
  Proc. *WISA 2006, Springer LNCS4298,* Cheju Island (Korea), 2006, pp.
- *F. Cheng, Ch. Meinel*
  Lock-Keeper: A New Implementation of Physical Separation Te
  Proc. *ISSE 2006,* Rome (Italy), 2006, pp. 275-286.
- *F. Cheng, Ch. Meinel*
  Research on the Lock-Keeper Technology: Architectures, Adva and Applications
  International Journal of Computer and Information Science(IJCIS), 5(3

Figure 2.8: *Annotating semantics for URLs*

2.3 Summary for This Chapter

This chapter gives a brief view on Web-Cares. Web-Cares is designed for those sites that are unauthenticated for the human visitors, for example, for the educational and public information sites, in which the usage data are stored as server logs. The algorithms and models used in Web-Cares can be easily adapted in other scenarios like online shops and communities, by adapting the relations between visitors and pages to the customers and goods or topics. Manipulating Web-Cares needs some auxiliary work such as setting proper thresholds and selecting right keywords for URL annotation. Finding interesting patterns and information is an iterative process affected by subjective personal understanding, though the main target of data mining is to automatically and directly discover the hidden information from the large data. In the following three chapters, this thesis will discuss in details the data preparation, models and algorithms used in Web-Cares.

Chapter 3

Data Preparation in Web Usage Mining

The log data on the web server side are the most common resources for web usage mining, as they are easy to collect. They contain the useful data from which a well-designed data mining system can discover beneficial information, unfortunately, raw sever side data contain much noise and are usually incomplete. In most cases, a log entry is automatically added each time when a resource request reaches the web server. Though this may reflect the actual use of the resource on a site, it does not record the real behaviors like frequent backtracking or frequent reloading of the same resource when the resource is cached by the browser or a proxy; moreover, the log sequence can not be directly mapped to the site structure. These drawbacks of logs are primarily attributed to the specifications of the HTTP protocol, which uses a connection for every file requested from the web server.

The target of data preparation is to rebuild access sessions for human visitors from server logs. A user session, according to the definition in W3C, is a delimited set of user clicks across one or more Web servers. Tanasa stated that two thirds of data mining analysts consider that data cleaning and preparation consume more than 60 percent of total analysis time (Tanasa and Trousse 2004).

Chapter Organization We briefly describe the tasks of collecting server logs in Section 3.1. Robots removal and sessions reconstruction are discussed in Section 3.2 and Section 3.3 separately. We focus on rebuilding individual accessing behaviors in Section 3.4. After these, to help understand the final mined usage patterns and interest, URL annotation is presented in Section 3.5. Finally we give the summary of this chapter in Section 3.6.

3.1 Collecting Server Logs

A web site may be allocated physically in multi servers, which means that the requests within the same user session would be fulfilled by the distributed servers. In this case, the log files from different servers have to be mixed and realigned in a chronological order.

In most cases, the web pages are requested and delivered to web users via HTTP protocols and these requests are recorded on the server side as HTTP server logs. However, the HTTP server logs include all the HTTP requests, which do NOT have to be the requests on web pages.

Moreover, the URL in an HTTP request may be an invalid or an outdated URL for a web page. So uniforming URLs is another task in collecting server logs. It is inevitable that some different URLs have exactly the same web content, and the reasons are possibly:

- URL redirection: a URL ending with "/" is automatically redirected to a default index page;
- URL updating: server logs always span over a period, during which the same web page could have different URL versions; and

- multi copies for the same web content.

In this step, the URLs sharing the same web content are replaced by one unique valid URL.

3.2 Robots Removal

The most important step of log data cleaning is the removal of robot accesses from the log data. We use the term ”robot” to refer to any programmable software agent that does not access a site interactively. These requests can mislead the analyst, because these sequences do not reflect the way human visitors navigate the site. To exclude these accesses, we employed several heuristic methods based on the indicators of non-human behaviors. These indicators are:

- a time interval between two requests is too short to apprehend the content of a page;
- the referee URLs that a series of requests from one host are empty; and
- the ”client agent” name is recognized as a robot in the robot database.

Though these techniques can be used to remove the noise made by robots, recognizing robots and their activities is time consuming and controversial. In the implementation of data preparation, robots removal is executed before log realignment, and this will reduce greatly the time complexity on data cleaning. In our investigation, we found that conservatively 80% visits recorded on the server side are NOT performed by human visitors, and the percentage could be even higher on public portal sites.

3.3 Sessions Reconstruction

Analyzing the sites that are free to unauthenticated access, we cannot rely on cookies or other similar measures to identify unique visitors. In this case, IP address, agent and version of OS and browsers are used to identify users. A session can be interpreted as a visit performed by a user from the time she enters the web site till she leaves. There are two fundamentally different methods of reconstructing sessions: by duration and by structure. However, the second is less appropriate because it is based on an assumption that a user has the semantical target during his session. The first method of time-based limitation of a session’s duration is the only feasible way.

Two time criteria are offered for this reconstruction: time spent on visiting a page $Timeout_l$, and time on the whole session $Timeout_g$. The first is more appropriate for the applications where visitors may spend a long time on the content of a page, such as online video watching or banking; while the second is intended for the web sites where browsing through pages or data items is a usual activity, for instance, 30 minutes.

3.4 Rebuilding Individual Accessing Behaviors

A session stores the URLs a user requested in a chronological order, but hides the actions like circular moves, query refinements. Hence, it is the necessary step to rebuild diverse individual accessing behaviors from usage sessions before pattern mining. The diversity of individual behaviors and the reconstruction technologies have been recognized by (Herder and Juvina 2004) and (Graff 2005) as well. In our research, depending on the target usage patterns, the individual user behaviors through the web site can be discovered into five different categories: granular accessing behavior, linear sequential behavior, tree structure behavior, acyclic routing behavior and cyclic routing behavior. In this section, we give the different algorithms for rebuilding these behaviors. This work is refereed in (Wang and Meinel 2004) and (Wang et al. 2005). The experimental studies will be discussed in Chapter 5.

3.4.1 Problem statements

To model the individual accessing behaviors, we firstly give the basic definitions of the terms that will be used in this section:

- W - the set of URLs of a web site;
- U - the set of visitors;
- T - a time span, for instance, from 01.01.2008 to 31.12.2008;
- L - the usage logs of U on W during T;
- S - the set of sessions reconstructed from L;
- l - one logline, or request in L;
- s - one session from S.

We use $l.visitor$ to denote the visitor of l, and $l.time$ to name the request time of l, and $l.url$ to represent the requested URL in l. It is obvious that for each l, $l.visitor \in U$, $l.time \in T$, and $l.url \in W$.

1. Definition (Session). *A session s is a set:*

$$s = \{l_1, \ldots, l_m\} : l \leq i < j \leq m, l_i.visitor = l_j.visitor, l_i.time < l_j.time.$$

And also s should satisfy the following conditions:

$$\forall i (1 \leq i < m) : l_{i+1}.time - l_i.time \leq Timeout_l, l_m.time - l_1.time \leq Timeout_g.$$

$Timeout_l$ and $Timeout_g$ are the two time criteria to identify sessions discussed in section 3.3. As the same definition of l, we also use $s.visitor$ to name the visitor of s, and $s.length$ is the number of URLs in s.

The session set S is a mapping from web logs L, for each $l \in L$, l belongs to exactly one session, and this ensures that S partitions L in an order-preserving way. We go on giving the necessary definitions to describe the individual accessing behaviors.

2. Definition (Repeated URLs in a Session). *Set of repeated URLs in a session s is defined:*

$$RepeatedURLs(s) = \{url_1, \ldots, url_n\},$$
$$\forall k (1 \leq k \leq n), \exists i, j (1 \leq i <> j \leq s.length) : l_i.url = l_j.url = url_k.$$

3. Definition (Unique Accessed URLs in a Session). *Set of unique accessed URLs in a session s is defined:*

$$UniqueURLs(s) = \{url_1, \ldots, url_n\},$$
$$\forall i, j(1 \le i <> j \le n), \exists i', j'(1 \le i' <> j' \le s.length) :$$
$$l_{i'}.url = url_i, l_{j'}.url = url_j, url_i <> url_j.$$

4. DEFINITION (AN ACCESS SEQUENCE IN A SESSION). *An access sequence in a session s is:*

$$Sequence(s) = \{url_1, url_2, \ldots, url_n\},$$
$$\forall i, j(1 \le i < j \le n), \exists i', j'(1 \le i' < j' \le s.length) :$$
$$l_{i'}.url = url_i, l_{j'}.url = url_j andl_{i'}.time < l_{j'}.time.$$

5. DEFINITION (AN ACCESS PATH IN A SESSION). *An access path in a session s is:*

$$Path(s) = \{url_1, url_2, \ldots, url_n\},$$
$$\exists i'(1 \le i' < s.length), \forall i, j(1 \le i <> j \le n) : l_{i+i'}.url = url_i, l_{j+i'}.url = url_j, url_i <> url_j.$$

The differences between an accessed sequence and a path are:

1. the URLs in a sequence are accessed in the same time order as in the original session, while in a path, all the URLs must be **continuously** accessed one by one as the order in the session;
2. a sequence could have repeated URLs, while a path has no repeated URLs.

An access path can be seen as a special sequence. Path is defined to find the URLs at which a user turned back to the previously accessed ones. This helps to investigate the deepest click streams over visitors, and is called as well Maximal Forward Reference (Chen et al. 1998).

It is well acknowledged that a web site is a complex graph, web pages are the vertices and the hyper links among them are edges. A visit of a user can be seen as a continuously or a broken roaming on this graph, and the history he requested web pages is stored in a session in a time sequence. The researchers determined that the users in their study interacted in a small area of a web site and frequently backtracked (Burton and Walther 2001). Tauscher and Greenberg reported on the patterns of user revisitation to web pages: user revisit web pages at a rate of 58% (Tauscher et al. 1997).

Thus, a sequential session could hide much information about the web graph, such as tree structure which characterized by the pages trigging different pages accessed afterwards. Corresponding the definition of a path, a tree structure hidden in a session must have a URL diverting different access paths. We call such tree structure relationship "divert paths tracking". Moreover, in the routing on a graph, it is possible that there are multi paths between two pages, which looks like a rhombus structure. We call this structural navigation "parallel path tracking".

6. DEFINITION (A DIVERT PATHS TRACKING IN A SESSION). *A divert path in a session s is:*

$$DivertPath(s) = \{Path(s)_1, \ldots, Path(s)_k\},$$
$$\forall i, j(1 \le i <> j \le k) : url_1^i = url_1^j, url_1^i \in Path(s)_i \textit{ and } url_1^j \in Path(s)_j.$$

7. DEFINITION (A PARALLEL PATHS TRACKING IN A SESSION). *A parallel paths tracking in a session s is:*

$$ParallelPath(s) = \{Path(s)_1, \ldots, Path(s)_k\},$$
$$\forall i, j(1 \le i <> j \le k) : url_1^i = url_1^j, url_{|P_i|}^i = url_{|P_j|}^j.$$

$|P_i|$ and $|P_j|$ are the lengthes of $Path(s)_i$ and $Path(s)_j$.

The definitions above reveal the diversities of individual activities endowed with backtracking, circular moves, query refinement, bookmarks and so on. It is possible for an individual accessing session to be interpreted with one of these definitions or the combination of several definitions, which is far more than object sets and sequences patterns as discussed in (Cooley et al. 1999)(Pei et al. 2000). A plain term "Action" is used to uniform these 6 different basic functions, because each of them characterizes the unique activity performed by a visitor on some objects in a session.

We now give the definition of individual access behavior: *An individual accessing behavior is the combination of several actions performed by a visitor during his session with the web server*(Wang et al. 2005).

3.4.2 Algorithms on rebuilding individual access behaviors

An individual access behavior is the combination of actions extracted from a session and displays not only the accessed web pages and part of site structure, but also the concept hierarchies and the routing activities on these pages.

Individual access behaviors can be discovered by several techniques. The choice of proper discovering methods depends on what kind of access patterns will be mined. From simple to complex, we show here some strategies of behaviors discovery. For example, if the sets of pages usually co-accessed are asked, unique accessed URLs need to be filtered firstly from every session; if the frequently clicked pages leading to different paths are the mining targets, divert paths tracking should be built for all sessions. We illustrate this problem by using a session reconstructed from server logs. Every URL is titled with its ID and this session is simplified as the following:

$$s = \{0, 292, 300, 304, 350, 326, 512, 510, 513, 512, 515, 513, 292, 319, 350, 517, 286\}$$

0 and 286 were accessed separately as entrance and leaving pages, and the repeated URLs are:

$$RepeatedURLs(s) = \{292, 350, 513, 512\}.$$

Any piece of the session without repeated pages can form a path, for example:

$$Path(s)_1 = \{300, 304, 350, 326, 512\}, \text{and } Path(s)_2 = \{512, 515, 513, 292\}.$$

3.4.2.1 Rebuilding simple behaviors

This strategy overlooks all the repeated pages in a session. The behavior of this visitor can be simply discovered into the largest set of accessed URLs, and the longest access sequence. These two kinds of behaviors are the extensions of the definitions of "unique accessed URLs" and "access sequence" in section 3.4.1; and the former is defined as $UniqueURLs_L(s)$ and the latter is $Sequence_L(s)$. To get the largest set of accessed URLs, the repeated URLs have to be removed; the longest access sequence equals to the session itself. In some work from other researchers (Agrawal and Srikant 1995), the repeated URLs are not allowed in forming an accessing sequence. With or without repeated URLs in a sequence depends on the concrete applications and personal understanding. For the above session, we remove the 10th, 12th, 13th, and 15th pages:

$$UniqueURLs_L(s) = \{0, 286, 292, 300, 304, 319, 326, 350, 510, 512, 513, 515, 517\}.$$
$$Sequence_L(s) = \{0, 292, 300, 304, 350, 326, 512, 510, 513, 512, 515, 513, 292, 319, 350, 517, 286\}.$$

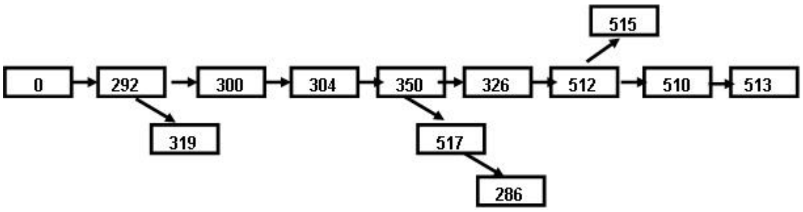


Figure 3.1: *Tree Structure behavior*

We can see that any sub set of $UniqueURLs_L(s)$ is one of the sets of accessed URLs by this visitor. Any subsequence of $Sequence_L(s)$ is one of the accessed sequences in this session. Motivated by other data mining applications in (Agrawal and Imielinski 1993)(Agrawal and Srikant 1995)(Pei et al. 2000), given a large group of accessed URLs and sequences, the set of most popularly accessed pages and the most popularly accessed page sequences can be mined.

3.4.2.2 Rebuilding tree structure behaviors

The tree structure behavior is characterized by divert paths in a session defined in section 3.4.1. From this definition, some paths in a session can form a diverged path because they share the same start accessed URL. Though all the accessed URLs are ordered by timestamp in a sequence, we can find those repeated URLs that lead to different target objects. Tree structure behavior not only displays the visiting patterns, but also reveals some conceptual hierarchy on site semantics.

During forming the tree structure t from a session s, a pointer pr is used to point to the last read URL in t. Every URL is read in the same order as in s and is inserted as the child node of pr if it firstly happens in t; but if the same URL already exists in t, we do nothing but letting pr pointing to this existing URL in t.

The tree structure behavior for the above session can be discovered with our strategy as the Figure 3.1. Based on this algorithm, there is some property in a discovered tree structure behavior:

Property 1: Given a discovered tree structure behavior, the nodes that lead to diverged paths are the repeated objects in this session.

The diverged paths in this session are:

$$DivertPath_1(s) = \{< 292 - 300 - 304 - 350 >, < 292 - 319 >\},$$
$$DivertPath_2(s) = \{< 350 - 326 - 512 >, < 350 - 517 - 286 >\},$$
$$DivertPath_3(s) = \{< 512 - 510 - 513 >, < 512 - 515 >\}.$$

The recovered navigation tree structures form the base to mine frequent tree structure access patterns (Zaki 2002) and frequent maximum forward references (Chen et al. 1998), which will be discussed in Chapter 4.

3.4.2.3 Rebuilding acyclic routing behaviors

"Acyclic routing behavior" means that in a session, there exist at least two different pages between which there are at least two different access paths. This kind of behavior is characterized by the parallel paths in a session. It shows that a visitor can access the same target object from the same start object but via different paths. This kind behavior happens frequently in the web

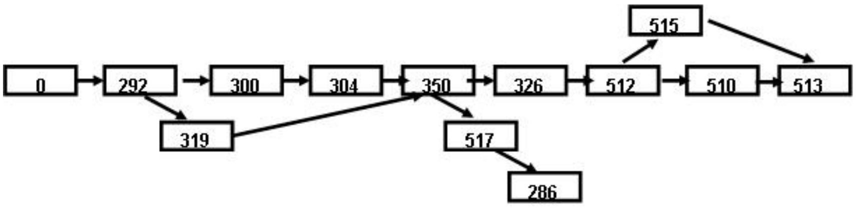


Figure 3.2: : *Acyclic routing behavior*

sites supplying rich interactions between users and web content such as online shopping, query refinement and formula submitting. With acyclic routing behaviors, we can further query the shortest path and most popular path between two pages.

The final discovered behavior is like a lattice structure defined as l, and pr is used to point to the last read URL in l. The URLs are read in the same sequence as in s, and for every URL, we check if the same URL exists in l. If this URL firstly happens in l, we insert this as a new child node of pr, and let pr point to this new node. If this URL already exists in l, there are four possible relations between this URL and the last read URL:

- This URL is the same as pr:
  1. Do nothing.
- This URL can be backward tracked from pr:
  1. Set pr point to this URL.
- This URL can be forward tracked from pr:
  1. Build a new directed edge from pr to this URL, if there is not directed edge from pr to this URL;
  2. Set pr point to this URL.
- This URL can not be tracked from pr:
  1. Build a new directed edge from pr to this URL,
  2. Set pr point to this URL.

Figure 3.2 shows the recovered acyclic routing behavior from the above session. It is clear that if an acyclic routing behavior can be discovered from a session, the session must have the following property:

Property 2: An acyclic routing behavior can be recovered from a session s iff there exist url_i, url_m, url_j, url_v, url_k and url_w($1 \leq i < m < j < v < w \leq s.length$) in s, and url_i==url_v; url_j==url_w; $url_m <> url_k$.

The parallel paths in this session are:

$$ParallelPath_1(s) = \{< 292 - 300 - 304 - 350 >, < 292 - 319 - 350 >\},$$
$$ParallelPath_2(s) = \{< 512 - 515 - 513 >, < 512 - 510 - 513 >\}.$$

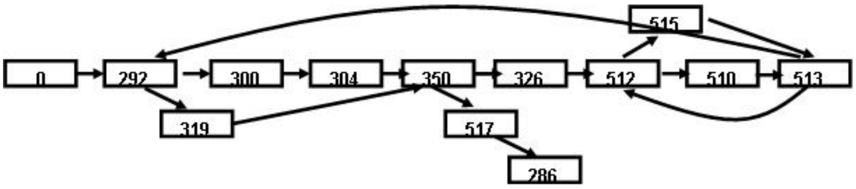


Figure 3.3: *Cyclic routing behavior*

3.4.2.4 Rebuilding cyclic routing behaviors

If there are revisited objects in a session, directed links will be built from every revisited object to one of its source object. From the semantic level, we think these two objects can be mutually heuristically evoked. Such individual behavior can be discovered as cyclic routing behavior and is characterized by the circle path hidden in the session.

The strategy for discovering cyclic routing behavior is similar to discovering acyclic routing. The following Figure 3.3 gives the cyclic routing behavior discovered from the same session.

The circle paths in this session are:

$$CirclePath_1(s) = 292 - 300 - 304 - 350 - 326 - 512 - 510 - 513 - 292,$$
$$CirclePath_2(s) = 512 - 510 - 513 - 512.$$

Cyclic routing behavior describes individual activities more colorful than simple behavior, but increases the complexity on judging the existence of such behavior. On the other hand, from simple behavior to cyclic routing behavior, the more complicated a usage model is defined, the lower possible it is to find enough support over a user group. Chapter 4 will discuss mining different usage patterns from the rebuilt individual access behaviors.

3.5 URL Annotation

Another data preparation work is URL annotation. Every URL is annotated with the most significant contexts extracted from its corresponding web document. These annotations are necessary for understanding the semantic information one usage pattern has in the next mining steps. Extracting representative topics from the documents has been explored in language processing and information retrieval: a document is represented as a term vector, where each term (usually word) is a basic concept, and each element of the vector corresponds to a term weight reflecting the importance of the term. There are three kinds of methods to index terms for documents: full-text indexing, keywords indexing and human indexing. Tf*Idf characterized by computing term frequency is the big algorithm family to weight terms in documents. However, in web hyperspace, the information contained in the anchor texts, meta-tags, header or title, capitalization, query logs (tau Yih et al. 2006) and in-linked or out-linked web pages (Sugiyama et al. 2003) effects greatly the weights of the terms of web pages.

The feature selection differs due to the characteristics of the web documents from the target web site, and in some cases, it is necessary to have a domain-specific taxonomy dedicated to a web site influencing the keywords extraction. The weight for one term o in a web document d is computed by:

$$w(o) = tf(o) \cdot idf(o) \cdot b(o)$$

where

1. $tf(o)$ is the term frequency of o in d,
2. $idf(o)$ is the inverse document frequency for o in d, and computed by $idf(o) = log(N/df(o))$ in which N is the total number of documents and $df(o)$ is the frequency of documents in which w appears.
3. $b(o)$ encapsulates the boosts of the features that strengthen the importance of o to d, and can be simply computed as: $b(o) = \sum(\beta_i \cdot o.has(f_i))$, in which

   $$o.has(f_i) = \begin{cases} 1, & \text{o has the feature } f_i \\ 0, & \text{o does not has the feature } f_i \end{cases}$$

   β_i is the boost for the feature f_i and $\sum \beta_i = 1$.

Based on the weight computation, a web document d is represented as a vector of K terms: $d =< w(o_1), ..., w(o_K) >$. However, the dimension of the vectors has to be reduced due to the possible large lengths of web documents and the redundancy of extracted terms. The selection on "K" dimension is equal to that on the semantically discriminative terms of web documents, and on the other hand, it is the compromise between the representability of the terms and the computations on the semantic distances among web documents. Because the creation of web documents is subjectively free work though a well designed web site has the clear intention on it services, we pursue the restricted site semantics having their relatively clearly boundary to compute the semantic similarities among web documents. The restricted site semantics are represented by a relatively stable set of terms.

3.6 Summary for This Chapter

In this chapter, we gave the necessary work in data preparation, but in reality, the data preparation is not restricted to those tasks listed above. Rebuilding individual accessing behaviors is one light spot in this thesis. The data preparation is the most time consuming work in the whole data mining process, which highly depends on the manual work in most cases, on the contrary, pattern mining process consumes the fewest time. Data preparation is highly related with the quality of the final discovered patterns. The whole data mining process is a cycle process, which means that the quality of the mined patterns decides if it needs re-preparing data or not.

Internet is an open environment, in which most web sites supply the unauthorized access to the Internet users. This helps to attract as many as possible visitors for the web sites, however, it produces large usage information having low quality. The data preparation becomes much harder and more important in web usage mining than those in other applications. Data preparation highly depends on the concrete applications, and this thesis will also give the related data preparation in tele-teaching and social communities in part II and part III.

Chapter 4

Modeling and Discovering Usage Patterns

In Chapter 3, we have discussed the data preparation in usage mining, especially the necessary work on rebuilding individual access behaviors from server logs. Various usage patterns are to be mined from the rebuilt individual behaviors. A pattern is *"an expression in some language describing a subset of the data or a model applicable to that subset"*(Fayyad et al. 1996). But pattern is not equal to knowledge, only if a pattern is interesting (according to a user-imposed interest measure) and certain enough (again according to the users criteria), though a pattern is expressed in a high level language. In the area of web usage mining, the navigation patterns, which are extracted in a nontrivial way (as defined in Chapter 1) and can be further interpreted and accepted by the people, are the discovered usage interest. So in the high level, the goal of data mining is to discover the patterns that can be interpreted into knowledge.

Chapter Organization This chapter begins with the theoretical principles drawn from four scenarios in Section 4.1. And then the methods on mining different patterns are discussed in Section 4.2. We propose a content clustering based on pair browsing in Section 4.3. Section 4.4 presents our work on discovering the changes of usage patterns. The necessary interpretation and evaluation of usage patterns are discussed in Section 4.5. Finally, Section 4.6 gives the summary of this chapter.

4.1 Starting from Four Scenarios

Scenario 1 *A web master wants to know the pages that are frequently co-accessed but not hyper linked with each other.*

Most web browsers supply the "back" button for the users to access their previously visited pages stored in local cache during one session, however, such usage information maybe not recorded in server logs until a new page is requested. Thence, this newly requested page is not directly linked to the last page before "back" button was clicked, but two requests are successively neighbored in server logs. To find the frequently co-accessed but not linked pages, co-accessed relationship among pages is modeled for navigation behaviors.

Scenario 2 *A professor wants to know how did the students apply bachelor studies from online.*

Let's assume that filling and submitting online application are within the same page A and A is hyper linked by several pages. A student could reach A via following different pages: directly from the homepage of department; from the project introduction page related with bachelor position; from the page about student life; or from the external search engine. So one solution for this question is using navigation sequence to model this usage behavior, in which all the pages along the navigation to A form a sequence in a time order.

Scenario 3 *A manager wants to grasp the mostly used cliques of the web graph.*

Web graph is formed with the pages as nodes, and links among pages as edges. The linkage information among pages could be the hyperlinks in the web graph or the consecutive usage clicks in a session. Filtering out the effect of backward references which are mainly made for the ease of web navigation, the session having repeated accessed pages could be transformed to a tree structure based on the techniques discussed in Chapter 3, and this tree structure is an efficient way to describe the clique that one user navigated on web graph.

Scenario 4 *A teacher wants to get the variance of the students' learning interest on the same lectures in different years.*

In first three scenarios, co-accessed, time sequence among pages and tree structures are used to model the web usage navigations. In Scenario 2, ″time sequence″ functions as a ″pattern template″ for pattern discovery, and the mined concrete traversal sequences ending with A are ″patterns″. However, in this scenario, the ″changes″ are measured on the multi aspects of a pattern. For example, if the learning interest on the lecture is depicted by the number of attendants who viewed the lecture, the time that the attendants spent on the lecture and the actions such as pause, stop and replaying, we can measure the changes computing the variance of learning interest from three aspects.

From the above four scenarios, we can see that the usage patterns differ in the form and template, and the selecting appropriate pattern template depends on the applications. Pattern template, or pattern type(Rizzi et al. 2003), represents the form of patterns, giving a formal description of their structures and relationships within the source data.

Generally, discovering usage patterns (in mining step) obeys the following steps:

```
Step 1:
   filtering the sessions which satisfy the defined
   constraints, for example, in Scenario 2, the sessions
   without page A are removed because they are not related
   with bachelor application;
Step 2:
   reorganizing the relationships among pages based on
   "pattern template" (co-accessed page set, sequential model
   or tree structure) for every session;
Step 3:
   designing the strategies to scan the rebuilt individual
   behaviors and computing the popular traversal patterns with
   the comparable parameters defined by the pattern template,
   for example, the frequency of one pattern in the session set;
   and
Step 4:
   interpreting and selecting useful patterns and displaying
   them in a user-understandable way.
```

Within one visit of a user, there may exist more than one different access patterns. Even for the same access pattern, there may be more than one explanation. Access patterns are the reflection of the site content and structures and must be interpreted by them. For the same page set but with

User ID	Session
100	abdac
200	eaebcac
300	babfaec
400	afbacfc

Table 4.1: *A database of sessions*

different user behaviors, we can get different access patterns. This thesis focus on the structural usage patterns, which are described by the dependency relations among web pages during users' visits. Due to the complicated structure of web site and the varieties in user behaviors, it is the challenging work to numerate all the kinds of access patterns. The simplest is to ignore all link information in the sessions, and to mine only the frequent sets of pages co-accessed by users. If considering the entire forward accesses of a user, frequently access subtrees or subgraphs are to be mined. In this thesis, the structural usage patterns can be grouped into frequent page sets, sequential access patterns, tree structures and more complex graphic patterns.

4.2 Mining Usage Patterns on a Web Site

Data preparation process generates a set of individual behaviors transformed from the session set based on the pattern template, which covers the first two steps shown in section 4.1. Then types of patterns $\{X_1, ..., X_m\}$ are mined from the set of individual behaviors $\{b_1, ..., b_k\}$:

$$\{b_1, ..., b_k\} \xrightarrow{f(\Delta)} \{X_1, ..., X_m\}.$$

$f(\Delta)$ represents the strategies on mining set of patterns, and a pattern X is defined as:

$$X = \{i, e, p, t\},$$

where i is a composition of data items (pages) integrating the hidden semantics with structures; e is the external description of i such as form size, depth, width, in- or out-degree of node; p is the set of values of parameters convincing the precision and assurance like support, confidence or interestingness; t is the valid time span denoting the temporal and space source dataset for X. e and p are decided by i, while the semantics of items are decided by the composition of the pattern which includes two parts: the items (pages) and the dependency among them. The semantics can be extracted by URL annotation after or before pattern mining in a separate process, which was discussed in Section 3.5.

In this section, we will give the used algorithms on mining frequent page sets, sequential patterns and tree structures, which fulfill the tasks in the first three scenarios separately. These three algorithms are the classic algorithms, we adopted them by modifying the conditions for stoping the iteration of scanning during mining process. For the convenience of understanding the algorithms, we use a synthesized example.

Example 1 Let $\{a, b, c, d, e, f\}$ be a set of five pages, and a set of 4 sessions is shown in Table 4.1.

User ID	Session	Accessed Pageset	Subpageset with frequent pages
100	abdac	a,b,c,d	a,b,c
200	eaebcac	a,b,c,e	a,b,c
300	babfaec	a,b,c,e,f	a,b,c
400	afbacfc	a,b,c,f	a,b,c

Table 4.2: *Accessed pagesets transformed from sessions*

4.2.1 Mining frequently co-accessed pages

Based on the definition of ”pattern”, a page set is assigned as:

$$X_{pageset} = \{\{p_1, ..., p_n\}, n, sup, time\},$$

where p_i is a unique page in the pageset, n is the size of this pageset, sup is the support of the pageset measuring its frequency in the dataset and $time$ is the valid period.

Two types of classical algorithms are widely used to mining frequent pagesets: apriori algrithm (Agrawal and Imielinski 1993) and FP-tree-based algorithm (Han et al. 2000). The difference between the both is with or without candidate generation during mining process: the former generates n-page length **pattern candidates** from combining $(n-1)$-page length **frequent patterns** and then scans the dataset to select the n-size length patterns having sup larger than a threshold, this process iterates until there is no more n-page patterns discovered; the latter adopts an extended aggregated tree structure to store the sup information for frequent patterns by once scanning and discovers n-page length pattern by forming this aggregated tree structure iteratively. This novel tree structure holds the condensed information: any path (a pattern candidate) from root to a leaf keeps the co-occurrence relations among pages in sessions, and the sup of any path is defined by the sup of the leaf. This tree structure successfully avoids scanning dataset during forming every candidate, which is the mostly time and storage costly step in apriori algorithm, especially when finding long patterns. However, for any of both algorithm, memory is the crisis with the increasing of session size and lowing of sup threshold.

Both algorithms discover frequent pagesets from 1-page length to n-page length, and there would be cover-set relations among these patterns. In our application, we try to mine the Maximum Frequent Patterns (Bayardo and Roberto 1998): the longest frequent pagesets (a frequent pattern X is a maximal frequent if there is no frequent pattern X' such that X' is a sub pattern of X). As will be explained in next section, the common step between apriori and FP-tree is the first scanning dataset to find frequent 1-page patterns and refine every sessions. In **Example 1**, $\{a, b, c\}$ is the 100%-pageset because it happens in all the sessions, while $\{f, c\}$ is the 50%-pageset because it gets supports from user 300 and 400. If we set 75% the sup threshold, which means a pageset is taken as a frequent pattern only if it is performed in at least 3 sessions, the first scanning filters out d, e and f for every session which is shown in Table 4.2.

To find the co-accessed but no-linked pages asked in **Scenario 1**, the discovered frequent pagesets by above techniques need to be refined based on the hyperlink infomation among pages in the target site. The web structure can be easily retrieved by crawlers, as discussed in Section 2.2. The post processing for the mined patterns will be discussed in later sections.

User ID	Session	Sequence without repeated pages	Subsequence with frequent pages
100	abdac	$a \rightarrow b \rightarrow d \rightarrow c$	$a \rightarrow b \rightarrow c$
200	eaebcac	$e \rightarrow a \rightarrow b \rightarrow c$	$a \rightarrow b \rightarrow c$
300	babfaec	$b \rightarrow a \rightarrow f \rightarrow e \rightarrow c$	$b \rightarrow a \rightarrow c$
400	afbacfc	$a \rightarrow f \rightarrow b \rightarrow c$	$a \rightarrow b \rightarrow c$

Table 4.3: *Accessed sequences transformed without repeated pages*

User ID	Session	Sequence with repeated pages	Subsequence with frequent pages
100	abdac	$a \rightarrow b \rightarrow d \rightarrow a \rightarrow c$	$a \rightarrow b \rightarrow a \rightarrow c$
200	eaebcac	$e \rightarrow a \rightarrow e \rightarrow b \rightarrow c \rightarrow a \rightarrow c$	$a \rightarrow b \rightarrow c \rightarrow a \rightarrow c$
300	babfaec	$b \rightarrow a \rightarrow b \rightarrow f \rightarrow a \rightarrow e \rightarrow c$	$b \rightarrow a \rightarrow b \rightarrow a \rightarrow c$
400	afbacfc	$a \rightarrow f \rightarrow b \rightarrow a \rightarrow c \rightarrow f \rightarrow c$	$a \rightarrow b \rightarrow a \rightarrow c \rightarrow c$

Table 4.4: *Accessed sequences with repeated pages*

4.2.2 Mining frequently accessed page sequences

A frequent page sequence depicts the time dependency among pages in a session and is an updated version of a frequent pageset: a time sequence is definitely a co-accessed relation, while the reverse is not. Similar to the algorithms on mining frequent pagesets, aprioiri (Agrawal and Srikant 1995) and FP-tree based (Pei et al. 2000) algorithms are used in mining frequent sequential patterns. Apriori algorithm promotes a generate-and-test method: first generating a set of candidate patterns and then testing whether each candidate may have sufficient support in the database (i.e., passing the minimum support threshold test). Reducing the size of pattern candidates at each iteration is the most costly work during mining. FP-tree algorithm, or WAP-tree (Pei et al. 2000), scans the access sequences twice and builds a tree structure in which access sequences with same prefix will share the same upper part of the path from the root. The height of this tree structure is one plus the maximum length of the frequent sequence, and the width is bounded the number of access sequences. Moreover, apriori removes the repeated happenings of the pages in every session, this is due to its item-growing nature by which n-length sequence candidates are formed by adding one different frequent item over $(n-1)$-length frequent sequences; while FP-tree keeps the original access sequences in every session and allows the repeated pages in the final frequent sequences.

Let's consider **Example 1**, 75% is set as the sup threshold to find the frequent access sequences. First scanning gets $\{a\}, \{b\}, \{c\}$ three frequent 1-page sequences and removes pages d, e and f as shown in Table 4.3. Apriori and FP-tree algorithm transform every session into a personal sequence composing by a, b or c, but Apriori only keeps the first happening of a repeated page in very session, for example, the second a in user 100 is removed.

The mining process of Apriori forms a lattice structure shown in Figure 4.1. 2-page sequences are generated by adding on page over 1-page sequences: $\{a \rightarrow b\}$ combines $\{a\}$ and $\{b\}$; while candidate $\{b \rightarrow a\}$ is removed due to its few 25% sup as the only support from user 300; the only 3-page sequence candidate is $\{a \rightarrow b \rightarrow c\}$ because all its 2-page subsequences are frequent. $\{a \rightarrow b \rightarrow c\}$ is the only maximum frequent sequence having 75% sup from user 100, 200 and 400.

The tree structures generated by FP-tree algorithm during discovering process are shown in

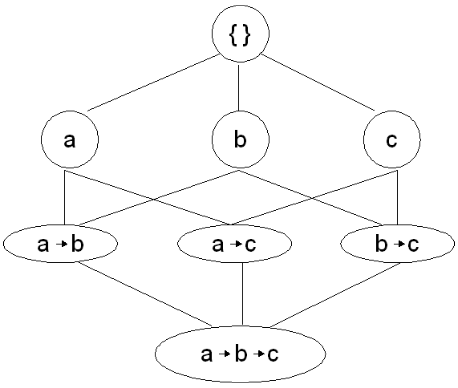


Figure 4.1: *Discovering process of Apriori*

Figure 4.2. FP-tree mining keeps the repeated pages within every session during the mining process as shown in Table 4.4. Firstly, it inserts the sequence $abac$ into the initial tree with only one empty root node. It builds a new node $a : 1$ (1 is the happening in the position) as the child of the root, and then creates the directed path following $a : 1$ "$(b : 1) \rightarrow (a : 1) \rightarrow (c : 1)$". Secondly, it inserts the second sequence $abcac$ at the root. Since the root has already a child named a, a's happening is increased by 1, $a : 2$ now. Similarly, $b : 1$ is increased to $b : 2$. The next, c, does not match the existing node a, and a new child of b is built $c : 1$. The left sequences are inserted iteratively in the same way. After reading all the four sessions, the original FP-tree is built, as the left one shown in Figure 4.2.

So now let's start finding the frequent sequences ending with c: the conditional sequences of c are listed from Figure 4.2 are:

$$(aba : 2), (ab : 1), (abca : 1), (ab : -1), (baba : 1), (abac : 1), (aba : -1)$$

The fourth sequence ab's happening is -1 because it is already included in the third sequence $abca$, and both are contributed from the same sessions. A conditional sequence must have 3 sup to be qualified as frequent. Hence, the conditional frequent sequences ending c are $(a : 4)$ and $(b : 4)$, c is removed due to only 2 sup. A conditional FP-tree structure, $a, b|c$ is created as the middle one in Figure 4.2. Recursively, the conditional sequences ending ac are built: $(ab : 3)$, $(bab : 1)$ and $(b : -1)$. When the conditional FP-tree $a, b|ac$ is created, as the right one in Figure 4.2, it has only one branch with $(a : 4)$ and $(b : 3)$. So one maximum frequent sequence is discovered: $abac : 3$.

The choice between with and without repeated pages in a sequence depends on the application: in online shopping web site, repeated operations in shopping process are necessary; while the repeated happenings of pages in a session should be removed if web master wants to investigate the most effective sequence reaching one page, such as finding how did the students apply the bachelor study in **Scenario 2**.

Another family in sequential pattern is Maximal Forward Reference (Chen et al. 1998). However, such sequential patterns are mined based on rebuilding individual navigation tree structure for every session, we classify this work in tree structure mining.

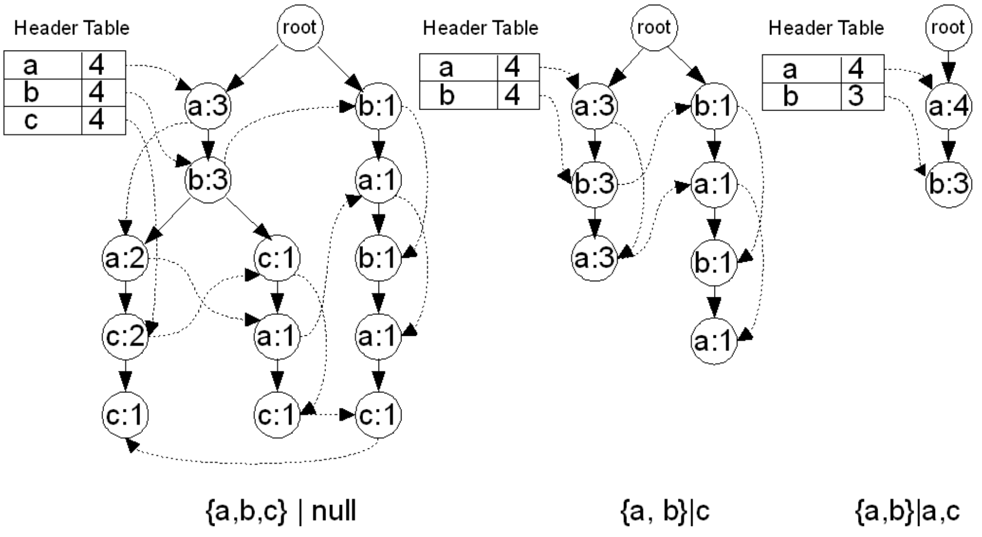


Figure 4.2: *Discovering process of FP-tree*

4.2.3 Mining frequently tree structures

Scenario 3 (seen in Section 4.1) gives the requirement to mine the frequent usage tree structures on web graph. As discussed in Section 3.4.2.2, every session, if possible, must be transformed into a tree structure by removing the backward references before discovering tree structures. Such rebuilt tree is a rooted, ordered and labeled tree: the first accessed page is the root node, the children of each node are siblings but ordered from 1th to kth child and each node has a unique label.

For efficient subtree counting and manipulation, a string representation is introduced by (Zaki 2002). A tree is denoted τ, and initialized $\tau = \emptyset$. Depth-first pre-order tracking starts at the root, adding the current node's label x to τ. Whenever we backtrack from a child to its parent we add a unique symbol -1 to the string. This string encoding is simpler to manipulate rather than adjacency list, matrix or trees for pattern counting. The recovered tree structures in **Example 1** are further transformed into string representations given in Table 4.5.

After transforming into a string encoding, each node (page) in a tree has a well-defined number, i, according to its position in a depth-first (or pre-order) traversal of the tree. Considering user 100 in Example 1, a is the 0th node and c is the 3th node. The **Scope** of a node n_l is given as $[l, r]$, where l is the position of this node and r is the position of its right-most leaf node in the subtree rooted n_l. Still considering user 100, the scope of a is $[0, 3]$ since its right-most node is c numbered 3; and the scope of c is $[3, 3]$ while its right-most node is itself. To discriminate the scopes for the same node in different trees, we add the tree label for every scope: the scope of a in user 100 is $100, [0, 3]$ and $200, [1, 1]$ in user 200. The scopes for the nodes in different trees for Example 1 is shown in Table 4.6.

The scope idea plays important in enumerating tree frequency, because the sibling, ancestor or descendant relations between nodes are judged in **constant time** by comparing their scopes: let $s_x = [l_x, u_x]$ and $s_y = [l_y, u_y]$ are the scopes for s_x and s_y in the same tree, s_x and s_y can only be siblings if $u_x < l_y$, while s_x is the ancestor of s_y if and only if $l_x \leq l_y$ and $u_x \geq u_y$.

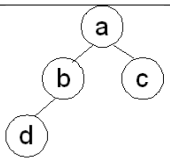
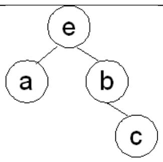
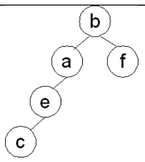
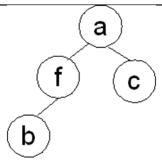
User ID	Session	Individual Tree Structure behavior	Transformed into string encoding
100	abdac		$a, b, d, -1, -1, c, -1$
200	eaebcac		$e, a, -1, b, c, -1, -1$
300	babfaec		$b, a, e, c, -1, -1, -1f, -1$
400	afbacfc		$a, f, b, -1, -1, c, -1$

Table 4.5: *Accessed tree structures*

User ID	a	b	c	d	e	f
100	100,[0,3]	100,[1,2]	100,[3,3]	100,[2,2]	X	X
200	200,[1,1]	200,[2,3]	200,[3,3]	X	200,[0,3]	X
300	300,[1,3]	300,[0,4]	300,[3,3]	X	300,[2,3]	X
400	400,[0,3]	400,[2,2]	400,[3,3]	X	X	400,[2,2]

Table 4.6: *Scope-Lists*

The detailed mining subtree strategy was discussed in (Zaki 2002). In **Example 1**, if 50% sup threshold is set, two frequent sub trees are mined finally: $a, b, -1, c$ and $e, c, -1$.

String encoding and scope are the two key ideas in mining frequent subtrees. Such transformation from a complex structure into a labeled sequence plays as well an important role in mining subgraph structures. Some efforts have been paid on mining more complex frequent usage graphs (Berendt 2005). In traditional frequent itemset discovery (pageset or sequences), the links between items are modeled in linear relationship, for example, the items can be sorted in a lexicographic order in discovering itemsets and in a chronological order in mining sequences. Clearly, this is not applicable to trees or graphs. To get total order of graphs, canonical labeling is used. A canonical label is a unique code of a given graph (Fortin 1996)(Shenoy et al. 2000), which should be always the same no matter how graphs are represented, as long as those graphs have the same topological structure and the same labels of edges and vertices. Intuitively, canonical labeling transforms a graph structure into a linear labeled sequence. Another challenge for discovering frequent graphs is expensively isomorphism testing, we don't discuss this topic in this thesis.

Short Discussion on Pattern Mining We refereed the classical algorithms on discovering frequent pagesets, sequences and trees, these techniques compress greatly the original dataset and depict the frequent usage behaviors in structural and comparable ways. More constraints are set on forming a frequent patterns, fewer patterns are discovered. Given the same session set with the same sup threshold, the number of mined frequent pagesets is larger than that of frequent sequences, and the frequent trees are fewest reversely. Considering Example 1, with 50% sup, the mined sequence $abac$ is naturally a tree structure while the mined tree structure $e, c, -1$ is a sequence as well; if the link relations between nodes are removed, every frequent sequence and tree structure is a frequent itemset. As will be discussed in Chapter 5, the mined frequent tree structures are the events of relatively low possibility and the depth of them are in average between 3 and 4. This shows that, the increasing diversity of usage patterns with the increasing of the constraints to form a pattern template, on the other side, strengthens the hardness on finding the usage interest from patterns.

During the mining process, it is noted that the frequent patterns are very sensitive, which means that small changes on the compositions of sessions could generate great different frequent patterns. Considering user 100 in Example 1, if original session $abdac$ is changed to $abdec$, $\{a, b, e, c\}$ will be the maximum frequent pageset with 75% sup and aec will be as well a maximum frequent sequence with 75% sup. This sensitivity could be even worse with the increasing of the complexity of a pattern template. Moreover, the tiny adjustment in selecting sup threshold could as well affect the composition of the frequent patterns, and this problem will be discussed in the experiment Chapter 5. The reason for this fragility is the cutoff between frequent and infrequent items, which means that one item could contribute to the final patterns only if it is frequent! Another characteristic of frequent pattern mining is the overlap on the compositions among patterns, and this is decided by the property that each subpattern of a frequent pattern is frequent as well. Considering the frequent trees in Example 1, $a, b, -1, c$ and a, e, the overlap between both is a. The overlaps between patterns bring the chaos in deciding the domain(pattern) that one page belongs to. On the other hand, the overlap helps to find the expected position of one page, which will be explained as a case study in Chapter 5.

So besides item-counting mining algorithms, there are other models used to investigate web usage patterns, such as semantic usage patterns(Berendt 2005), usage profiles clustering (Heer and Chi 2002), navigation model based on Markov Chain (Sarukkai 2000) or Bayesian Networks. These techniques produce groups of usage profiles or models and play well in the scenarios like locating potential users and advertisements. However, the web master sometimes concentrates much on the general navigation graph. In this navigation graph, each node corresponds to a page and the weighted linkage between two nodes is computed based on number of hpyerlinks or navigation referees. In the next section, we focus on building navigation graph based on page pairs from usage logs.

4.3 Page Clustering based on Pair Browsing

The usage traversals from users produce a graph structure, which reflects or distorts the original graph generated by web pages and hyperlinks among them. The motivation of this section is to investigate the difference between two graphs: one is web graph created by the site designer, and the other is the navigation graph generated by visitors.

Page pair, the binary relationship between pages, acts as the edge to form a graph. Each edge is assigned a weight measuring the closeness of two pages, hence the web graph can be divided into clusters in which the inter-cluster edges are weighted higher than than intra-cluster edges. The methods to model the binary relationship between two pages includes two categories: one is on site design, and the other is on usage feedback. Based on the former category, the binary relationship is modeled as the ″similarity″ or ″authority distribution″ (Kleinberg et al. 1999) described on the conceptual characteristics or ″linkage″ (Page et al. 1999) defined from web expert's or designer's side. The page clusters built on such methods display the initial organization of web content. On the latter category, the binary relationship is modeled from usage behaviors, which depict the seasonal and positional relations between two pages. In such cases, co-accessed (Nakayama et al. 2000) and accessed-after (Perkowitz and Etzioni 2000) relations are used to calculate the binary relations in web usage, which is the central part in personalized recommendations and improving content organization.

However, we model the binary relations by computing the **distance of their accessed positions** in usage sessions. We use this model based on the following assumption: the lately accessed content depicts the intentions and behaviors of one visitor more exactly than the early accessed content, and the lately accessed content is greatly targeted by its last previously accessed content. We use ″*heuristic importance*″ to depict the importance of one page to attract visitors to access another page. In a page cluster built on such page pairs, a web page is closely linked by those pages that have ″higher″ heuristic importance to it.

4.3.1 Heuristic importance within a page pair

A page pair is named as $Pair(p_h, p_t)$ and the heuristic importance is noted as $Hr(p_h, p_t)$, where $h < t$. We use $|s|$ to name the length of a session s, and there are at most $|s|$ different pages accessed in s, and $Pos(p, s)$ is used to denote the position of the page p in a s: $1 \leq Pos(p, s) \leq |s|$.

8. DEFINITION (POSITION RELATION). *Within a session* s, *the position relation from a page* p_h *to another page* p_t *is named as:*

$$M_s(p_h, p_t) = \frac{\sqrt{Pos(p_h,s)Pos(p_t,s)}}{|Pos(p_t,s)-Pos(p_h,s)|+1}.$$

9. DEFINITION (HEURISTIC IMPORTANCE WITHIN A PAGE PAIR). *Over the session set* S, *the heuristic importance from* p_h *to* p_t *in* $Pair(p_h, p_t)$ *is defined as:*

$$Hr(p_h, p_t) = \frac{\sum_{i=1}^{n} M_{s_i}(p_h,p_t)}{n},$$

where n *is the number of sessions in the session set* S.

In a session s, for every page pair, its position relation is $\frac{1}{\sqrt{|s|}} \leq M_s < \frac{|s|}{2}$, which shows that the later a page pair is accessed in a session, the higher position relation the page pair has; and the heuristic importance within a page pair measures the mutual relation between two pages over the whole session set. If there are multi happenings of the same page pair within a session, we use the highest position relation for this page pair. Extremely, when there is only one session in the session set, the heuristic importance for any page pair is equal to its position relation in a session.

We also use other methods to model page pairs (binary relation) from usage view, which will be used for comparing and evaluation.

Method 1 (SUP) In this model, a page pair is symbolized as two adjacently accessed pages in a session, and the support of these two adjacently accessed pages over the whole sessions is used to measure the binary relation. This support is computed as the number of happenings of this page pair in the session set divided by the size of the session set. This measurement is used as well in computing the happenings of 2-item sub-sequences in session set, which has been discussed in section 4.2.2.

Method 2 (IS) A page pair is symbolized as two adjacently accessed pages in sessions. (Tan and Kumar 2000) used

$$Br(p_h, p_t) = \frac{Pr(p_h p_t)}{\sqrt{Pr(p_h)Pr(p_t)}}$$

to compute the binary relation between page p_h and page p_t. In this formula, $Pr(p_h)$ is the possibility of the happening of p_h in session set S, which is the same to the sup of p_h over S. So does $Pr(p_t)$. $Pr(p_h, p_t)$ is defined as the support of 2-item sub sequence including adjacent pages $p_h p_t$ over S, which is as the same as that shown in Method 1.

Method 3 (CS) The binary relation is characterized by the conditional possibility:

$$Br(p_h, p_t) = Pr(p_h|p_t).$$

This measurement is also named as *confidence* in data mining and n-Markov chain and is used in personalized recommendation and adaptive web sites (Sarukkai 2000).

Surely, there are other methods to model the mutual information of two objects from usage view, as will be shown in Figure 4.4 in Section 4.5, but we only use the above three methods which are suitable and widely used in web usage mining.

4.3.2 Clustering method

The mutual relation computed based on heuristic importance is asymmetric, because p_t is accessed after p_h in a session, though the values are the same between $Hr(p_h, p_t)$ and $Hr(p_t, p_h)$. The clustering method that finds the related page communities from page pairs is introduced in this section. A general clustering process is given in follows:

```
1. recovering sessions from web usage logs;
2. scanning the recovered sessions
   and building page pairs by computing heuristic importance;
3. creating the directed graph based on
   the heuristic importance matrix for page pairs; and
4. finding the clusters in the directed graph.
```

The clustering process looks very simple, but the most expense consuming is to cluster in step 4. Generally, clustering (partitioning) over a directed or undirected graph is formulated as an optimization problem, which is NP-complete. A general point of view on clustering is to create clusters that balancing the criteria between inter-cluster (i.e. between clusters) and intra-cluster (i.e. within clusters). This is also a strong criterion for a good clustering, which guarantees strong connectedness within the clusters. However, the clustering method based on heuristic importance is different from and could not directly use the $K-means$ clustering or hierarchical agglomerative clustering, because the binary relations in the former are asymmetric while those in the latter are symmetric.

The heuristic importance matrix H is a typical asymmetric $n \times n$ matrix (n pages) with real, non-negative elements, and H forms a labeled, weighted and directed graph G. $Hr(p_i, p_j)$ represents the heuristic importance from page p_i to page p_j with $i, j \in \{1, 2, ..., n\}$. Besides, based on the support number for a page p_i over the session set, we denote by $D_i > 0$ the normalized support value functioning as the stationary probability for p_i and by D the diagonal matrix with D_i, $i \in \{1, 2, ..., n\}$ on the diagonal. One clustering family over affinity matrix is based on graph cut. The clusters found in this directed graph will be seen as a partition over the page set P, and we use $C = \{C_1, ..., C_M\}$ to note a clustering of P over H.

Two clusters C_K and $C_{K'}$ $(1 \leq K \neq K' \leq M)$ of P, such that $C_K \cap C_{K'} = \emptyset$ and $C_K \cup C_{K'} = P$ (in our case $C_K \cup C_{K'} \subset P$), define a **cut** in graph G. A real function $cut(C_K, C_{K'})$ based on heuristic importance between the pages from two clusters C_K and $C_{K'}$ represents the value of a cut:

$$cut(C_K, C_{K'}) = \textstyle\sum_{p_i \in C_K, p_j \in C_{K'}} Hr(p_i, p_j).$$

Various cut criteria, such as minimum cut (Gomory and Hu 1961), normalized cut (Shi and Malik 1997) and weighted cut (Meila and Pentney 2007), can be used for measuring the goodness of a clustering. To balancing of the cluster size and the closeness within a cluster, we use **normalized cut**:

$$Ncut(C_K, C_{K'}) = \tfrac{cut(C_K, C_{K'})}{D_K} + \tfrac{cut(C_K, C_{K'})}{D_{K'}},$$

with $D_K = \sum_{p_i \in C_K} D_i$.

Given a predefined cluster number M, the best clustering is reached by minimizing the normalized cut criterion:

$$Ncut_M = Ncut(C_1, P - C_1) + Ncut(C_2, P - C_2) + ... + Ncut(C_M, P - C_M).$$

Before clustering, we have to obtain a symmetric matrix H^* from the original asymmetric matrix H by applying kinds transformations, such as $H^* = H + H^T$. In order to reduce noise, we apply one threshold to remove the pages receiving lower support numbers. Further to reducing the clustering cost, we add the following constraints: any directed path created by pairs can be seen as the dispersion of heuristic importance along this chain, so it is reasonable to remove the loop paths and keep the heuristic importance reducing during clustering. Another convincible observation is that with the increasing of sessions, the dimension of page pairs is much controllable than that of a session set.

4.3.3 Site modeling

As written at the beginning of section 4.3, our goal is to build content clusters on page pairs from usage view, and to discover the difference between visitors' expectations and designer's intentions. The difference between two sides for the same page helps greatly to improve the content organization. This requires modeling page relations within a web site, including or excluding certain parts of a web site.

Web has been modeled by many ways, most of which are based on the graph theory. PageRank (Page et al. 1999) and HIT (Kleinberg et al. 1999) are the two famous methods. Besides the graph model, role-based model was used in (Srivastava and Cooley 2000), in which each page can be classified into navigation page or content page. In personalized recommendation system, the page relations are modeled by an n-Markov in which one page accessed by a visitor is decided by the n previous accessed pages(Sarukkai 2000)(Huang et al. 2004).

Our task here is to reveal the designer's intentions on designing a web site from the structure side. We used PageRank to model the site, which is easily to handle by the crawler introduced in Section 2.2. And the work on URL annotation discussed in Section 3.5 helps to understand the web graph from conceptual side.

Short Discussion on Page Clustering The work discussed in this section is refereed in (Wang and Meinel 2006), however, we further made improvements such as on clustering algorithm and site modeling. Page clustering based on pair browsing describes in the high level the usage interest on the web site, which is reflected by the binary page relation during browsing. Compared with the frequent usage patterns, the differences of clustering based on pair browsing are:

1. the former is sup centered, while the latter is $relation$ centered, which means that the pages in a frequent pattern are loosely connected, while the pages in a clusters are closely connected; and

2. a page could exist in multi frequent usage patterns, but belongs to at most one cluster.

Because of the close relations among the pages within a cluster, it is the direct requirement to investigate the difference between visitors' expectations and designer's intentions. Visualization on usage pattern is another topic in data mining (Liu et al. 2002)(Chen et al. 2004)(Youssefi et al. 2004). In this thesis, we use a simple but intuitive page centered way as shown in Figure 2.7 in Chapter 2, "page view-based", to show the difference: the binary relations from usage view and designer's view are displayed when a page is selected. The clustering results will be discussed in Chapter 5 together with the other usage patterns.

From the frequently co-accessed page sets, page sequences, and tree structure patterns to page clustering based on pair browsing, we have discussed the methods on discovering different kinds of usage pattern types. These patterns are tracked on a dedicated time span, which did not consider the time factor during mining process. However, the usage interest varies with the time changing due to the changes of Internet users and the Internet content in different periods. In next Section, we will give the methodology on detecting the changes of web navigation patterns.

4.4 X-tracking the Changes of Usage Patterns

To mine the same patterns, different algorithms output the same results and differ on the time and space consuming during mining. Though the novel strategies for economic mining are always necessary, we concentrate more on the post application after pattern mining. We think the interpretations and post applications of mined patterns are the bottle neck for the further acceptance of patterns. In this section, we discuss discovering the changes of usage patterns from different time spans. This work is refereed by (Wang and Meinel 2009).

We propose an X-tracking method to detect the changes of web navigation patterns. This method describes the changes on two layers: microscopic and macroscopic layers. The former concerns the differences on the composition of content and structure between single patterns,

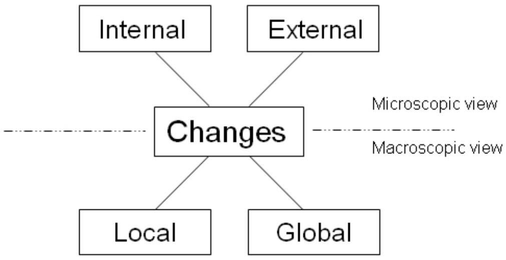


Figure 4.3: *X-tracking Changes*

while the latter is dedicated on the changes in the underlying populations of navigation patterns along the time line. On microscopic layer, we use *"internal"* change to depict the difference synthesizing the content and structure, and *"external"* change to consider the varying of its form. The changes on macroscopic layer have two views: *"local"* change depicts the change of the population coverage of a single pattern between two time spans, while *"global"* change models the life cycle of a pattern against all time spans. Intuitively, the relations among these four kinds of changes are represented by an X-shape structure in Figure 4.3.

Firstly, we define the basic terms used in the following sub sections:

- t_i: the i^{th} time span along the entire time line T, and $T = \{t_0, ..., t_n\}$;
- U_i: the usage data collected at t_i;
- ζ_i: the set of navigation patterns discovered at t_i based on a defined pattern template (or pattern type);
- X: one navigation pattern, and X_i is one pattern in ζ_i mined at t_i.

As defined in section 4.1: a pattern is a quadruple $X = \{i, e, p, t\}$, where i is a composition of data items integrating hidden semantics with structures; e is the physical description of i such as form size, depth, width, in- or out-degree of node; p is the values of parameters convincing the precision and assurance like support, confidence or interestingness; t is the valid time span denoting the temporal and space source dataset for X.

In our method, the changes are described from two layers: microcosmic and macroscopic layers. On the microcosmic layer, two kinds of changes are defined:

- ***Internal Change***: refers to the internal difference synthesizing the content and structure between two patterns.
- ***External Change***: considers the varying of the external form of a single pattern.

On the macroscopic layer, we depict the changes of underlying populations from two aspects:

- ***Local Change***: concerns the changing of popularity of a single pattern between two consecutive time spans.
- ***Global Change***: models the change of popularity over the whole time line.

4.4.1 Measuring internal change

Internal change shows the structural and conceptual difference between two patterns from different time spans. We use an edit distance approach integrating the structures with the conceptual relatedness of elements to compute the internal difference. *Edit distance* is widely used in computing structural similarity ranging from string-to-string difference (Wagner and Fischer 1974) to tree structure comparison (Zhao et al. 2004). Originally, *edit distance* between two structures is computed based on the smallest sum of cost of the basic edit operations that change one structure to the other. The basic edit operations are often defined as **insertion**, **deletion** and **updating**.

Navigation patterns have different structural formats depending on the type of models ranging from frequent page sets (associate rules), sequential navigation paths, tree structures to other directed graph based structures. Computing *edit distance* for these structures are bit different:

- *frequent page sets*: edit distance is computed by comparing the appearance or disappearance of web pages from one pattern to the other;
- *sequential navigation paths*: such case is similar to string-to-String correction problem referenced in (Wagner and Fischer 1974); and
- *tree and directed graph structures*: tree or directed graph structure firstly has to be transformed into a unique sequence representation by introducing position labels based on depth or width searching as discussed in Section 4.2.3; and then computing edit distance between two unique sequences is same to that between two navigation paths. Transforming tree and directed graph can be refereed in (Zaki 2002) and (Fortin 1996) separately.

In order to amend the precision and the accuracy of pattern changes, we integrate the semantic distance in the structure-based similarity algorithm. Semantic distance that one node changes to the other, or the distance between two different versions of the same labeled node, is used in our algorithm as the cost for the basic changing operation. As explained in section 3.5, each web page d is represented by a term vector composed by the extracted keywords with their *Tf-Idf* values plus additional annotation values like appearances in page header or title, anchor texts and query logs. Computing similarity recursively has also been explored in the specific context of database schema-matching (Melnik et al. 2002).

10. DEFINITION (SEMANTIC DISTANCE). *The semantic distance between two web documents d_1 and d_2 from two patterns in different spans is computed as:*

$$\mathrm{sDist}(d_1, d_2) = 1 - \mathrm{sSim}(d_1, d_2),$$

where $\mathrm{sSim}(d_1, d_2)$ *is the semantic similarity between d_1 and d_2.*

We use the cosine to compute the semantic similarity between two vectors of d_1 and d_2:

$$\mathit{sSim}(d_1, d_2) = \frac{\sum_{i=1}^{K} w_{d_1,i} \cdot w_{d_2,i}}{\sqrt{\sum_{i=1}^{K} w_{d_1,i}^2} \cdot \sqrt{\sum_{i=1}^{K} w_{d_2,i}^2}},$$

where $w_{d_1,i}$ and $w_{d_2,i}$ are the weights of ith word in web pages d_1 and d_2 respectively.

11. DEFINITION (INTERNAL DISTANCE). *Let X_1 and X_2 be two navigation patterns, the internal distance $iDist(X_1, X_2)$ between X_1 and X_2 is computed as the edit distance integrated with the semantic distances between the pages from two patterns.*

In computing internal distance, we use the following rules:

- the cost for one **updating** page p_1 with p_2 is the semantic distance from p_1 to p_2;
- the costs for **deletion** and **insertion** are 1 because the semantic distance between a web page and an empty is 1; and
- we overlook the effect of domain-specific taxonomy on computing importance similarity between two keywords, however, this taxonomy can be defined with the help of domain-experts (Eirinaki et al. 2003).

12. DEFINITION (INTERNAL DIFFERENTIAL). *The internal differential between two patterns X_1 and X_2 is computed as:*

$$Diff_I(X_1, X_2) = \frac{iDis(X_1,X_2)}{max(iDis(\emptyset,X_1),iDis(\emptyset,X_2))},$$

where $iDis(\emptyset, X_i)$ $(i \in \{1,2\})$ is the internal distance of building the entire X_i from empty based on the basic edit operations.

Here $Diff_I(X_1, X_2)$ is the percentage of nodes that have changed from X_1 to X_2 against the max number of nodes from X_1 and X_2. Internal differential quantifies the gap between two patterns in their structure and semantic composition, and its value is between 0 and 1.

13. DEFINITION (SIMILAR PATTERN). *Given X_i and X_j, we call X_j a "Similar Pattern" for X_i in ζ_j, if $Diff_I(X_i, X_j) < \theta_s$, where θ_s is a defined threshold for the internal differential.*

14. DEFINITION (MOST SIMILAR PATTERN). *Given X_i and X_j, we call X_j "Most Similar Pattern" for X_i in ζ_j, if X_j is one similar pattern for X_i, and $\neg\exists X_j^{'} \in \zeta_j$ that $Diff_I(X_i, X_j^{'}) < Diff_I(X_i, X_j)$, and we use $M_{X_i \dashv \zeta_j}$ to represent the most similar pattern for X_i in ζ_j.*

From this definition, X_i could have more than one most similar patterns in ζ_j. Based on the definition of most similar pattern, we give the definitions of "**internal unchanged**" and "**emerged**" pattern.

15. DEFINITION (INTERNAL UNCHANGED PATTERN). *Given X_i and its most similar pattern $M_{X_i \dashv \zeta_j}$, we call X_i "internal unchanged" pattern iff $Diff_I(X_i, M_{X_i \dashv \zeta_j}) < \theta_u$, where θ_u is the unchanged threshold. $M_{X_i \dashv \zeta_j}$ is called j^{th} "version" of X_i, and X_i is called the i^{th} "version" of $M_{X_i \dashv \zeta_j}$.*

16. DEFINITION (EMERGED PATTERN). *X_i is called "emerged pattern" if no version of X_i is found in the pattern sets from ζ_0 to ζ_{i-1}.*

Mining emerging patterns was discussed in (Dong and Li 1999)(Zhao and Bhowmick 2004), in which a pattern is considered as "emerging" if its support is over a threshold, while not considered as its conceptual and structure. From the definitions above, it is drawn that an "internal unchanged" pattern is discovered by being compared with the patterns in its posterior time spans, while an "emerged" pattern is concluded based on the patterns in its prior time spans. Given two pattern sets ζ_i and ζ_j $(i < j)$, detecting internal unchanged patterns in ζ_i is based on locating the most similar patterns in ζ_j for the patterns of ζ_i. All the detected internal unchanged patterns from ζ_i form a subset of ζ_i, and we use $\zeta_i \triangleright \zeta_j$ to denote this subset and $M_{\zeta_i \triangleright \zeta_j}$ to represent the subset of ζ_j which is composed by their corresponding most similar patterns in ζ_j.

$\zeta_i \triangleright \zeta_j$ and $M_{\zeta_i \triangleright \zeta_j}$ give the internally stable elements from ζ_i to ζ_j. On the other hand, $\zeta_i - \zeta_i \triangleright \zeta_j$ and $\zeta_j - M_{\zeta_i \triangleright \zeta_j}$ are the internal differential between ζ_i and ζ_j. Based on our definitions, no element in $\zeta_i - \zeta_i \triangleright \zeta_j$ can find its most similar pattern in $\zeta_j - M_{\zeta_i \triangleright \zeta_j}$, it is composed by all the "**emerged**" patterns in ζ_j. Further, we call the patterns in $\zeta_i - \zeta_i \triangleright \zeta_j$ "**perished patterns**" from ζ_i to ζ_j.

For internally unchanged patterns, their stable internality could cover the prominent changes in their external features like form size and the variations of their hidden popularity along the time line. This pushes us to investigate the changes in other aspects. For perished and emerged patterns, however, the changes on external and popularity features will not be discussed in the following sections because they have no internal related patterns to compare.

4.4.2 Measuring external change

External change is the variation of the physical features. Here we concern the change of pattern form size in different time spans. The form size is the basic symbol showing the information quantity of one kind of patterns.

Form size of pattern X, named as $|X|$, is the happening number of pages in X, including requests that involved revisits.

17. DEFINITION (EXTERNAL DIFFERENTIAL). *The external differential on form size between two patterns X_1 and X_2 is computed by:*

$$Diff_E(X_1, X_2) = \frac{|X_1| - |X_2|}{max(|X_1|, |X_2|)}.$$

The external change for pattern X_i at time span t_j is measured as the differential on form size between X_i and its most similar pattern at t_j, which is $Diff_E(X_i, M_{X_i \dashv \zeta_j})$. There are two possible variations of external feature of X_i in ζ_j:

- the pattern expands: if $Diff_E(X_i, M_{X_i \dashv \zeta_j}) > 0$.
- the pattern shrinks: if $Diff_E(X_i, M_{X_i \dashv \zeta_j}) < 0$.

4.4.3 Measuring local popularity change

In these two sub sections, we discuss the changes of local and global popularity for a pattern. The local concerns the popularity difference between two time spans, while the global is for that over the whole history.

The popularity support Sup hidden behind X_i has two forms: *support number* and *support ratio*. The former is the absolute number of transactions (or sessions) that compromise X_i in U_i, while the latter is *support number* against the size of U_i.

18. DEFINITION (LOCAL POPULARITY CHANGE). *The local popularity change for X_i from t_i to t_j is computed by:*

$$Diff_L(X_i, M_{X_i \dashv \zeta_j}) = \frac{Sup(M_{X_i \dashv \zeta_j}) - Sup(X_i)}{Sup(X_i)}.$$

A larger positive value of the local popularity change implies a more significant increasing number of visitors that acted as that pattern in his/her navigation behavior. Given a pre-defined positive threshold θ_l for local popularity change, a pattern suffers local popularity change in two directions:

- the pattern floats: $Diff_L(X_i, M_{X_i \dashv \zeta_j}) > \theta_l$.
- the pattern sinks: $Diff_L(X_i, M_{X_i \dashv \zeta_j}) < -\theta_l$.

Based on the local popularity change, we model the *life cycle* of a pattern to gain the insights on the evolution of the population against the entire time spans, which reflects the *global* change of its popularity.

4.4.4 Measuring global popularity change

We give the following definitions based on the local popularity change.

19. DEFINITION (DEGREE OF FLOATS). *Let $< \zeta_1, \zeta_2, ..., \zeta_n >$ be the sets of navigation patterns mined at n time spans. Suppose a pattern X_i and the time span t_i when X_i firstly emerged, the degree of floats is defined as:*

$$DoF(X_i, \theta_l) = \frac{\sum_{j=i+1}^{n} d_j}{n-1},$$
$$where\ d_j = \begin{cases} 1, if Diff_L(X_i, M_{X_i \dashv \zeta_j}) > \theta_l \\ 0, if Diff_L(X_i, M_{X_i \dashv \zeta_j}) \leq \theta_l \end{cases}$$

20. DEFINITION (DEGREE OF SINKS). *Let $< \zeta_1, \zeta_2, ..., \zeta_n >$ be the sets of navigation patterns mined at n time spans. Suppose a pattern X_i and the time span t_i when X_i firstly emerged, the degree of sinks is defined as:*

$$DoS(X_i, \theta_l) = \frac{\sum_{j=i+1}^{n} d_j}{n-1},$$
$$where\ d_j = \begin{cases} 1, if Diff_L(X_i, M_{X_i \dashv \zeta_j}) < -\theta_l \\ 0, if Diff_L(X_i, M_{X_i \dashv \zeta_j}) \geq -\theta_l \end{cases}$$

Degrees of floats and sinks for a pattern reflect the changes in two directions over the history.

21. DEFINITION (GLOBAL POPULARITY CHANGE). *The global popularity change for a pattern is measured by the pair of its DoF and DoS under threshold θ_l:*

$$Diff_G(X_i, \theta_l) = (DoF(X_i, \theta_l), DoS(X_i, \theta_l)).$$

4.4.5 Algorithms for X-tracking the changes

From the previously discussions, we can see that changes depicted in an x-view are discovered based on the locating most similar pattern for a pattern in a different time span. Suppose that X_i firstly emerged at t_i, the pseudo-code algorithm for X-tracking the changes of X_i in the rest of time spans is shown in Algorithm 1.

Before tracking the changes of a pattern, the time span in which it was firstly emerged has to be detected. This process is related with locating the most similar pattern and can be constructed as a corresponding set-covering problem. Generally, the set-covering problem is an NP problem if without any a priori background information. However, the changes of a pattern are tracked against the whole history, and a pattern needs to compute its internal differentials with all the patterns before finding its most similar pattern from another time span. Scanning the patterns in a posteriori time span for a priori pattern distinguishes the newly emerged patterns in a posteriori time span. On the other hand, the final formation of the subset E_i of newly emerged patterns at t_i ($E_i \subseteq \zeta_i$) is purified and decided by all the sub sets of newly emerged patterns before t_i:

Algorithm 1 *Tracking the Changes for a newly Emerged Pattern*

```
Given $X_i$ firstly emerged at $t_i$ and n pattern sets $< \zeta_1, \zeta_2, ..., \zeta_n >$,
$DoF(X_i, \theta_l) = DoS(X_i, \theta_l) = 0$
for $j = i + 1$ to $n$ do
  locate Most Similar Pattern $M_{X_i \dashv \zeta_j}$ for $X_i$ in $\zeta_j$
  if $M_{X_i \dashv \zeta_j} \neq$ NULL then
    compute $Diff_I(X_i, M_{X_i \dashv \zeta_j})$
    compute $Diff_E(X_i, M_{X_i \dashv \zeta_j})$
    compute $Diff_L(X_i, M_{X_i \dashv \zeta_j})$
    if $Diff_L(X_i, M_{X_i \dashv \zeta_j}) > \theta_l$ then
      $DoF(X_i, \theta_l) = DoF(X_i, \theta_l) + 1/n$
    else if $Diff_L(X_i, M_{X_i \dashv \zeta_j}) < -\theta_l$ then
      $DoS(X_i, \theta_l) = DoS(X_i, \theta_l) + 1/n$
    end if
  end if
end for
compute $Diff_G(X_i, \theta_l)$
```

$$E_i = \begin{cases} \zeta_1, & i = 1 \\ F(E_1, ..., E_{i-1}), & i > 1 \end{cases}$$

The accumulation of such scanning reduces the expense on tracking the changes for the patterns firstly emerged in a posteriori time span. The following algorithm gives the formation of the subset of newly emerged patterns E_i at ith time span.

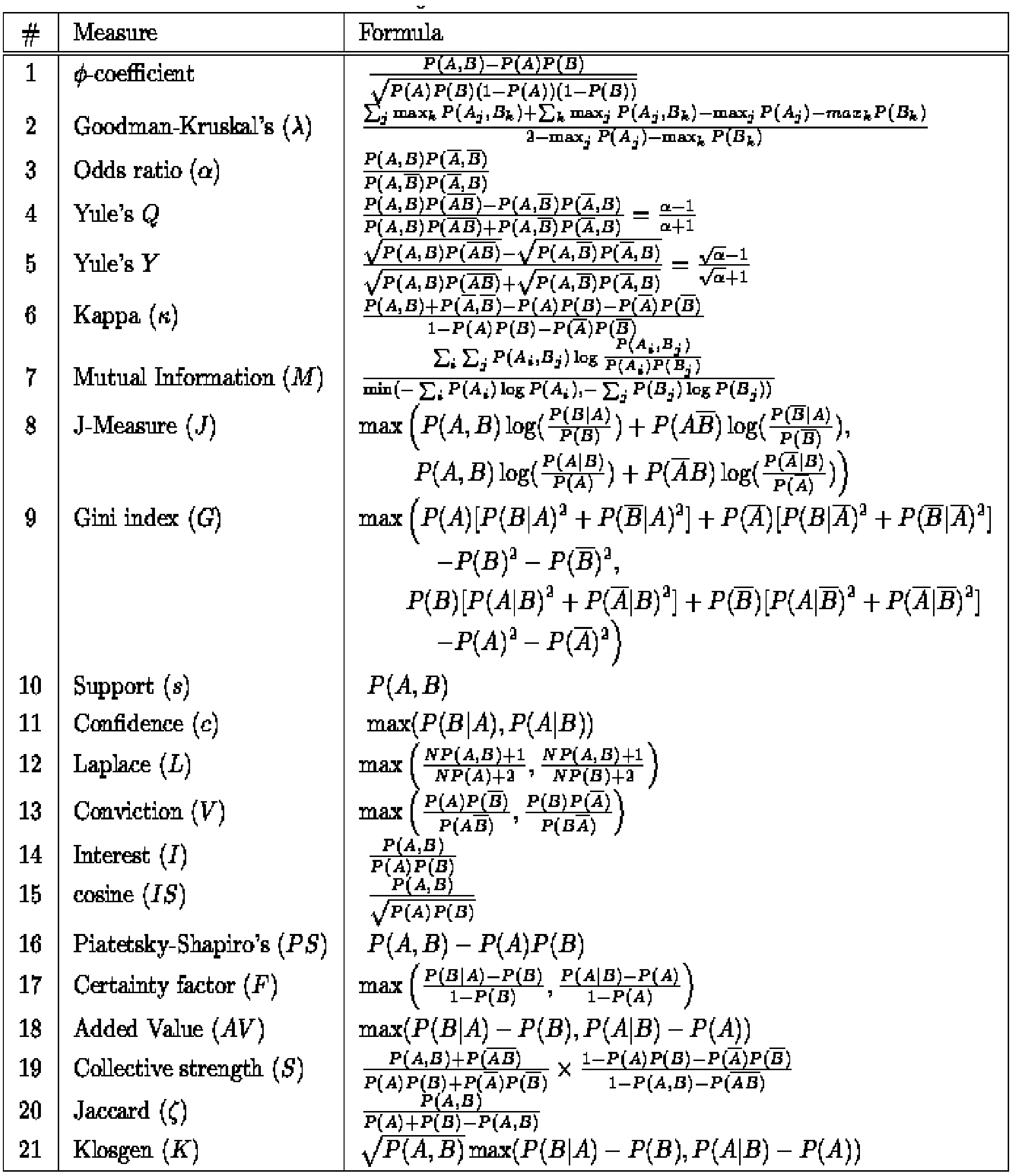
Algorithm 2 *Formation of newly Emerged Patterns at i_{th} Time Span*

```
Given $n$ pattern sets $< \zeta_1, \zeta_2, ..., \zeta_n >$
Initialize $E_1 = \zeta_1$, $E_2 = \zeta_2$, ..., $E_n = \zeta_n$
for $i = 2$ to $n$ do
  for $j = 1$ to $i - 1$ do
    $E_i$ = $E_i - M_{E_j \triangleright E_i}$
  end for
end for
```

In Algorithm 2, $M_{E_j \triangleright E_i}$ is the subset of E_i, in which each element is the most similar pattern for one pattern in E_j $(j < i)$. Based on the definition on locating most similar pattern, it can not be guaranteed that every element in E_j could find its most similar pattern in E_i. The experiment on the algorithms will be discussed in Chapter 5.

4.5 Pattern Interpretation and Evaluation

The pattern mining is the process of extraction, compression and re-organization over the large quantity source data. This process inevitably generates the new formation and the information unknown before, which requires the further steps on comparing, filtering and interpreting. To assess the difference and validity of usage patterns, evaluation standards from three different categories are used: technique standard, expert standard and task oriented standard.



#	Measure	Formula
1	ϕ-coefficient	$\frac{P(A,B)-P(A)P(B)}{\sqrt{P(A)P(B)(1-P(A))(1-P(B))}}$
2	Goodman-Kruskal's (λ)	$\frac{\sum_j \max_k P(A_j,B_k)+\sum_k \max_j P(A_j,B_k)-\max_j P(A_j)-max_k P(B_k)}{2-\max_j P(A_j)-\max_k P(B_k)}$
3	Odds ratio (α)	$\frac{P(A,B)P(\overline{A},\overline{B})}{P(A,\overline{B})P(\overline{A},B)}$
4	Yule's Q	$\frac{P(A,B)P(\overline{AB})-P(A,\overline{B})P(\overline{A},B)}{P(A,B)P(\overline{AB})+P(A,\overline{B})P(\overline{A},B)} = \frac{\alpha-1}{\alpha+1}$
5	Yule's Y	$\frac{\sqrt{P(A,B)P(\overline{AB})}-\sqrt{P(A,\overline{B})P(\overline{A},B)}}{\sqrt{P(A,B)P(\overline{AB})}+\sqrt{P(A,\overline{B})P(\overline{A},B)}} = \frac{\sqrt{\alpha}-1}{\sqrt{\alpha}+1}$
6	Kappa (κ)	$\frac{P(A,B)+P(\overline{A},\overline{B})-P(A)P(B)-P(\overline{A})P(\overline{B})}{1-P(A)P(B)-P(\overline{A})P(\overline{B})}$
7	Mutual Information (M)	$\frac{\sum_i \sum_j P(A_i,B_j)\log\frac{P(A_i,B_j)}{P(A_i)P(B_j)}}{\min(-\sum_i P(A_i)\log P(A_i),-\sum_j P(B_j)\log P(B_j))}$
8	J-Measure (J)	$\max\Big(P(A,B)\log(\frac{P(B\|A)}{P(B)})+P(A\overline{B})\log(\frac{P(\overline{B}\|A)}{P(\overline{B})}),$ $P(A,B)\log(\frac{P(A\|B)}{P(A)})+P(\overline{A}B)\log(\frac{P(\overline{A}\|B)}{P(\overline{A})})\Big)$
9	Gini index (G)	$\max\Big(P(A)[P(B\|A)^2+P(\overline{B}\|A)^2]+P(\overline{A})[P(B\|\overline{A})^2+P(\overline{B}\|\overline{A})^2]$ $-P(B)^2-P(\overline{B})^2,$ $P(B)[P(A\|B)^2+P(\overline{A}\|B)^2]+P(\overline{B})[P(A\|\overline{B})^2+P(\overline{A}\|\overline{B})^2]$ $-P(A)^2-P(\overline{A})^2\Big)$
10	Support (s)	$P(A,B)$
11	Confidence (c)	$\max(P(B\|A),P(A\|B))$
12	Laplace (L)	$\max\left(\frac{NP(A,B)+1}{NP(A)+2},\frac{NP(A,B)+1}{NP(B)+2}\right)$
13	Conviction (V)	$\max\left(\frac{P(A)P(\overline{B})}{P(A\overline{B})},\frac{P(B)P(\overline{A})}{P(B\overline{A})}\right)$
14	Interest (I)	$\frac{P(A,B)}{P(A)P(B)}$
15	cosine (IS)	$\frac{P(A,B)}{\sqrt{P(A)P(B)}}$
16	Piatetsky-Shapiro's (PS)	$P(A,B)-P(A)P(B)$
17	Certainty factor (F)	$\max\left(\frac{P(B\|A)-P(B)}{1-P(B)},\frac{P(A\|B)-P(A)}{1-P(A)}\right)$
18	Added Value (AV)	$\max(P(B\|A)-P(B),P(A\|B)-P(A))$
19	Collective strength (S)	$\frac{P(A,B)+P(\overline{AB})}{P(A)P(B)+P(\overline{A})P(\overline{B})}\times\frac{1-P(A)P(B)-P(\overline{A})P(\overline{B})}{1-P(A,B)-P(\overline{AB})}$
20	Jaccard (ζ)	$\frac{P(A,B)}{P(A)+P(B)-P(A,B)}$
21	Klosgen (K)	$\sqrt{P(A,B)}\max(P(B\|A)-P(B),P(A\|B)-P(A))$

Figure 4.4: *Objective Interestingness Measures for Patterns*

4.5.1 Objective: Technique standard

The objective measure includes ranking patterns based on statistics computed from data, such as the 21 measures (Geng and Hamilton 2006) shown in Figure 4.4.

From the above measurements, the effect of different models is closely related with the choosing proper thresholds. The best measure for the success of web structure improving is the direct positive feedback from the visitors, though this task is uncontrollable in reality.

4.5.2 Subjective: Expert standard

Although there are technique standards to evaluate the mining results, judging if a pattern is interesting or useful is a relatively subjective task, different persons could have different tastes and interpretations on the same pattern.

One kind of subjective evaluation is ranking patterns according to the user′s interpretation: a pattern is subjectively interesting if it contradicts the expectation of a user; and a pattern is subjectively interesting if it is actionable. (Siberschatz and Tuzhilin 1996).

In (Tan and Kumar 2000), *Gold standard* is named as the expert criterion in general evaluation method that is used to find ″ideal solution″ to a problem. Such evaluation is usually determined manually by one or more experts, and sometimes it is inevitable of the happening of different understandings and evaluations from different experts over the same patterns because of their subjective backgrounds. The method to reduce the subjective bias from experts is trying to get as more suitable experts as possible.

4.5.3 Task oriented standard

In web usage mining, the ideal evaluation for a content improving schema is the direct feedback from client sides. But in web applications, especially for large quantity of unprofitable web sites, such direct feedback is uncontrollable. We are pushed to raise three measurements to evaluate usage models:

```
 1. If similar patterns happen in different models,
   then these patterns are useful.
 2. If similar patterns happen in different periods of time,
   then these patterns are valuable.
 3. If a model reflects the changes of the content reorganization,
   then this model is reasonable.
```

The first and second measurements illustrate the universality and continuity of a right pattern, and third depicts the robustness of a model adaptive to the changing of the evolving situations.

4.6 Summary for This Chapter

This chapter started with four scenarios on discovering usage interest in different applications, which bring the requirement on defining and extracting different usage pattern types from usage data. We have discussed mining frequently co-accessed page sets, page sequences and tree structure patterns. We then gave our page clustering on pair browsing, which is used to detect the gap between web designer′s expectation and users′ destination. X-tracking the changes of usage patterns was further discussed, which aims to disclose and measure the variance of the usage patterns from different time spans. The contributions of our work in this chapter are from two aspects:

1. page clustering based on pair browsing;

2. tracking the changes of web navigation patterns.

We implemented the methods of mining different pattern types on two educational portal sites, and the experiments will be discussed in the next Chapter 5: experiment results discussion on Web-Cares.

Chapter 5

Results Discussion on Web-Cares

The usage data for our experiment were taken from two web sites: www.hpi.uni-potsdam.de (HPI) and eccc.hpi-web.de (ECCC). HPI site is an educational portal site, which covers the information on the study, teaching, research and management in Hasso Plattner Institute. ECCC (Electronic Colloquium on Computational Complexity) site is an electronic journal on ideas, techniques, and research in computational complexity. The quantity of information of HPI site is much larger than that of ECCC site, and as well the range of the content diversity of HPI site is wider than that of ECCC site. So the usage interest could be different on the form and the content between both sites, which will be discussed in this chapter.

Chapter Organization Section 5.1 gives the general usage statistics on both sites. We discuss the rebuilt individual behaviors in Section 5.2, further the discovered usage patterns in Section 5.3 and cases studies are shown in this section as well. Section 5.4 concentrates on the changes of usage patterns discovered from different time spans. Finally the summary on this chapter is given in Section 5.5.

5.1 General Usage Statistics

The general usage statistics, such as "how many hits did one page get in the last month" and "the geographic distribution of IP addresses of visitors", can be easily resolved by the existing commercial or open source web analysis tools. So we do not present such usage statistics in the section, however, we concentrate on finding the roles that web pages played in web usage behaviors.

The raw usage data taken from ECCC site and HPI site spanned 14 months from 2007.01 to 2008.02. Before discovering usage interest, the raw usage data have to be cleaned, integrated and reconstructed. Such data preparation work has been discussed in Chapter 3, which highly depends on manually work and consumes over 60% costs in the whole data mining process. After URL uniforming, robot removal and session construction, two sets of usage data are built. Over 80% usage data have been removed from the raw usage data due to the robots' access, and the noisy HTTP requests such as outdated and unrelated URLs. We use **Data Set 1 (DS1)** to name the usage data from HPI and **Data Set 2 (DS2)** to call the usage data from ECCC.

For investigating the entrance and exit pages, we made a further data cleaning on **DS1** and **DS2**: the sessions having only accessed one page are removed, because the entrance page and the exit page are the same for these sessions. These sessions covered 69.7% of all the sessions on ECCC site (63878/91595) and 68.6% (113549/165535) on HPI site. The following Table 5.1 gives the general information on the two usage data sets.

Table 5.1: *General information on DS1 and DS2*

	Requests	Sessions	Unique URLs
HPI Site	331526	51986	5039
ECCC Site	87617	27717	1050

Different pages could play different roles in the usage behaviors: some pages are always visited as entrance pages for sessions, some pages are much easier to be exit pages, and other pages are as middle or navigation pages. The distribution by different roles can give us more about web structure and usage information. For an entrance or exit page, its hit frequency against its rank obeys Zipf's law. The following Figure 5.1 and Figure 5.2, give the statistics on the entrance and exit pages on HPI and ECCC sites. By analyzing the accessing information of entrance pages and exit pages, HPI site and ECCC site have the similar distribution: there are more exit pages than entrance pages on both sites. This shows that the exit pages may play more importance on discriminating visitors from different interest. This partly proves our assumption raised in Section 4.3 for page clustering based on pair browsing: **the lately accessed content depicts the intentions and behaviors of one visitor more exactly than the early accessed content, and the lately accessed content is greatly targeted by its last previously accessed content**.

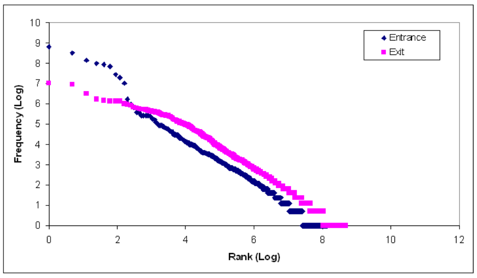


Figure 5.1: *Entrance Page vs. Exit Page on HPI*

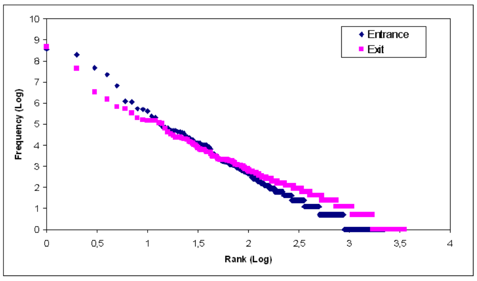


Figure 5.2: *Entrance Page vs. Exit Page on ECCC*

5.2 Rebuilding Individual Behaviors

In Chapter 3, we have especially discussed rebuilding individual behaviors: the diverse behaviors like tree structures, acyclic or cyclic navigations are reconstructed from the usage logs recorded sequentially based on time stamps on server side. The complexity of individual behaviors is due to the pressing back button on browser or back tracking along the hyperlinks on web pages, which generate the multi requests on the same page within a session. We showed that for every session a granular behavior and a linear sequence behavior can be discovered, but to rebuild a tree structure behavior there must be repeated requests in a session, and for a semi-lattice structure behavior, two or more different requests must be repeated. We compute the number of the sessions with 1 or 2 repeated pages, and also the number of the sessions that can discover tree and semi-lattice behaviors.

Simply, we use **r-page** to call the page that is **repeatedly** requested within one session: 1 r-page means one unique repeated page in a session, which produced at least 2 requests in this session;

Table 5.2: *Information on repeated pages within a session*

	DS1 (HPI)	DS2 (ECCC)
Unique r-pages in data set	1928 (38.7%)	535 (50%)
Sessions having 1 r-page	13079 (25%)	3324 (12%)
Sessions having 2 r-pages	6041 (10.4%)	1135 (4.1%)
Sessions having n r-pages ($n > 2$)	2577 (5%)	268 (1%)

and 2 r-page means that two unique pages have been repeatedly requested in a session which generated at least 4 requests in this session.

The distribution of repeated pages is closely related with the web structure. It is clear that a structure with many pages has more repeated pages than the ones with small number of pages. It can be seen from Table 5.2 that ECCC site has a small number of repeated pages than HPI site. From the same Table, we can see that a page from a site having fewer pages has a more possibility to be repeatedly requested in a session than a page from a site having more pages. The reason is that a visitor could have more choices in a site with more pages, which could reduce the happenings of repeated visits. However, on the other hand, a visitor on a small site is less possible to request the previously accessed page in a session than the one on a large site. The ratios of sessions having r-pages on HPI site is higher than those from ECCC site. This could be the several reasons:

1. the main content on ECCC is about the publications and reports related on computational complexity, however, HPI site composes kinds of information covering study, teaching, research, and other common resources; or
2. the site structure of ECCC is simple, while the pages on HPI site are hyper linked in a complex way, such as navigation bar on the top and left side.

The difference of the r-pages on both sites is also proven by the length of sessions: the average of the length of sessions on ECCC site is 3.16 ($87617/27717$), while that on HPI site is 6.37 ($331526/51986$). A longer session is more possible to have repeat requests that a shorter one.

5.3 Discovering Usage Patterns

In this section, we discuss the experiment results on mining the popular and characteristic web usage patterns from the rebuilt individual behaviors. The corresponding mining algorithms have been explained in Chapter 4.

5.3.1 Frequent navigation patterns

We discuss the results on mining frequently co-accessed pages, frequently accessed page sequences and frequently accessed tree structures in this sub section. The data from **DS1** and **DS2** for discovering navigation patterns have been made the following completion and deletion:

1. **completion**: if the referred page for the first request (seen the log format in Section 2.1) in a session came from another web site, then this referred page is inserted into the session as the first request; and

2. **deletion**: if a session has accessed only one page, then the session is removed from the data set.

The motivation for this **completion** comes from the reality that a great part of visitors accessed the web pages via Internet search engines by inputting some terms and as well via the hyperlinks refereed by other web sites. The reason for removing the sessions having only one page is that from these sessions we can only get the usage information like ”how many hits did one page receive?”. The **deletion** is executed after the **completion**. After the completion and deletion, we got 88597 sessions from **DS1** and 50840 from **DS2**. The frequent navigation patterns are mined from these two modified data sets.

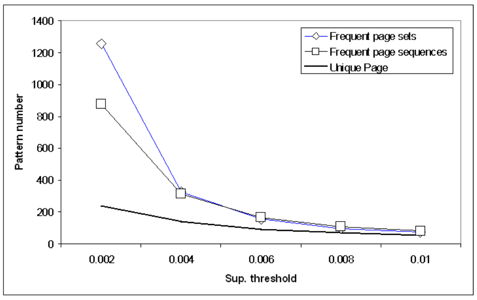


Figure 5.3: *Co-accessed pages vs. page sequences on HPI (raw)*

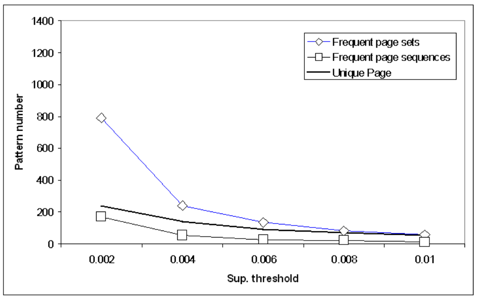


Figure 5.4: *Co-accessed pages vs. page sequences on HPI (maximal)*

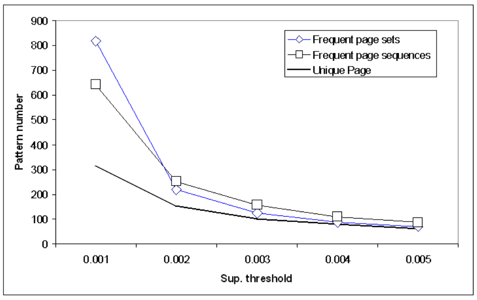


Figure 5.5: *Co-accessed pages vs. page sequences on ECCC (raw)*

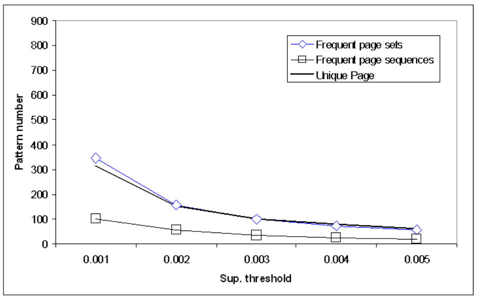


Figure 5.6: *Co-accessed pages vs. page sequences on ECCC (maximal)*

Frequently co-accessed pages and page sequences Figure 5.3 and Figure 5.4 show the number of mined frequently co-accessed page sets and page sequences on HPI site under different support thresholds. We can see that the number of mined patterns is decreasing with the increasing of support threshold. The number of the mined frequently co-accessed pages is larger than that of the mined frequently accessed page sequences under the same thresholds, and this gap is decreasing with the increasing of threshold. This is because the page sequence is a strengthened version of co-accessed page set, which means that all the pages in a page sequence were co-accessed by sessions, however, the pages in a co-accessed page set could have multi page sequences navigated by the sessions. On the statistics, this variance reduces the possibility for a page sequence to be supported by enough sessions.

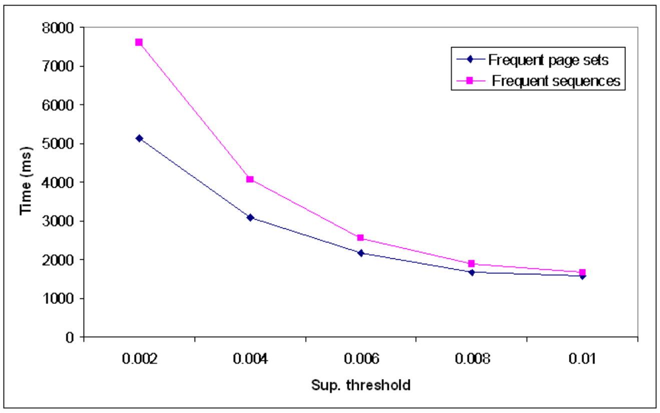


Figure 5.7: *Performance on mining patterns*

The **difference** between Figure 5.3 and Figure 5.4 is that the latter is the number of maximal patterns filtered from the former, and this ensures that there is no child-parent relations among the patterns in Figure 5.4. Compared with frequently co-accessed page sets, the number of maximal frequently accessed page sequences is more highly compressed. The reason for the number of maximal frequently co-accessed page sets larger than that of the maximal accessed page sequences is that a page sequence has at least two pages, while a co-accessed page set can have only one page. Another observation is the number of patterns and the number of unique pages: with the increasing of threshold, the gap between both numbers is decreasing, this is because the more pages, the more combinations of the pages. However, the number of maximal page sequences is lower than that of the unique pages under the same support threshold. This tells that though some pages were visited together, but usually in different sequences. The similar observations are found as well on ECCC site, which were shown in Figure 5.5 and Figure 5.6.

By investigating the composition of these frequent patterns on HPI site, we found that a big part of them started from some search engines like Google and MSN, and then followed the important pages and link structures. The important pages are the pages linked in the navigation bar or the framework of a web page, and the pages on the information about study application, courses' introduction, homepage of a research group, free position and the projects about computer graphics, computer human interaction and security.

The time performance of mining frequently co-accessed page sets and page sequences on HPI site is shown in Figure 5.7. On the same session set under the same support threshold, the time consumed on page sequences is larger than on page sets, this is due to the reordering of the pages within every session based on the bibliography before mining frequent page sets, which reduces the consuming on generating larger candidate patterns from smaller frequent page sets. However, the order among the pages within every session has to be kept during mining frequent sequences, which costs not only extra storing space in memory for the same page in different positions within the sessions, but the time on enumerating the candidate patterns.

Frequently accessed tree structures A feature of an accessed tree structure, which is different from a co-accessed page set and a page sequence, is the existence of at least one page having been followed by different pages. This is shown in the session set that enough sessions have repeatedly requested one page. However, the process of mining tree structures generates the frequently accessed page sequences as well, and such page sequences have to be filtered after the mining process. The differences between the original tree structures and the refined ones on HPI site and ECCC site are shown in Figure 5.8 and Figure 5.9.

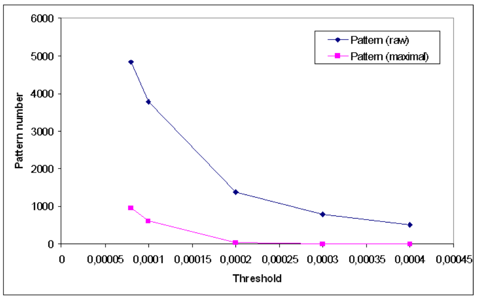


Figure 5.8: *Tree structures on HPI*

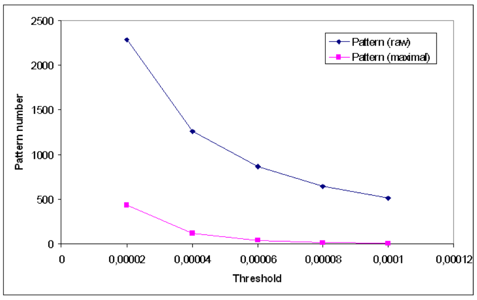


Figure 5.9: *Tree structures on ECCC*

Comparing the performance on mining tree structures on HPI site and ECCC site, which are shown in Figure 5.10 and Figure 5.11, the time consuming is not only related with the threshold, but depends on the complexity of the sessions. The lower threshold, which generates more patterns, the more time is costed on pattern mining. The complexity of a session can be simply described by the number of the pages it has.

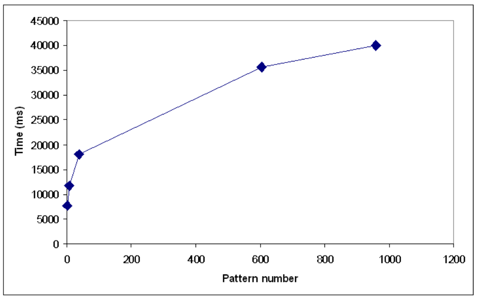


Figure 5.10: *Performance on mining trees (HPI)*

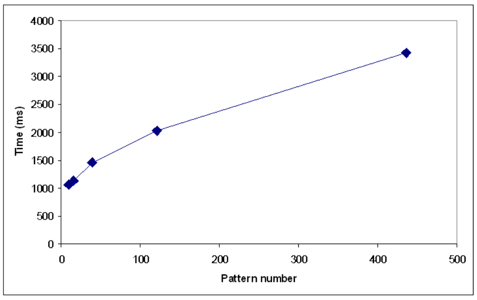


Figure 5.11: *Performance on mining trees (ECCC)*

Short Discussion on Frequent Patterns The choice of the proper threshold for pattern mining is a cyclic process, and the value of threshold is selected from large to small. In our experiments, the thresholds for mining frequent page sets and page sequences on HPI site are chosen between 0.002 and 0.01, and on ECCC site between 0.005 and 0.01. The thresholds for mining frequent tree structures are even smaller: for ECCC site between 0.00002 and 0.0001; while for HPI site between 0.00008 and 0.0004. The low thresholds tell the great variance among the individual navigation behaviors, and with the increasing of the number of sessions, discovering the similarities and the common interest over the whole visitors becomes a small probability event. On the other hand, this reveals that **the usage behaviors and interest are changing in different time spans**, which was discussed in Section 4.4 and the experiment will be shown in Section 5.4.

The above discussion gives the general and descriptive explanations on the mined patterns.

Compared with the huge original usage data on the web site, these navigation patterns are greatly compressed and structured, and they have the expressive formats which richly describe the mostly accessed pages and their relations in some time periods. The mined navigation patterns help the web masters from two levels: on the macro level to grasp and understand the usage preference, and on the micro level to adjust some concrete web pages or services. However, to find more valuable information, especially on the instruction and suggestion, we need to further interpret and evaluate the mined patterns.

5.3.2 Content clusters based on pair browsing

As discussed in Section 4.3, the content clusters built based on pair browsing are the agglomerative transformations of the binary page relations retrieved from the usage data, which in the high level depict the usage interest on the web site. During clustering, a page belongs to at most one cluster or is filtered out due to its lower support over the visitors and rare pair browsing relations with other pages. As mentioned in the clustering algorithm, the formation of page clusters is decided by the threshold for support number and the cluster number K. Because the optimal clustering is NP complete problem, we set the number of iterations I during finding K clusters, and get the clustering result at the minimized normalized cut during I iterations.

However, we concentrate more on the difference between the two graphs retrieved on pair browsing and site modeling. Given a page, showing its directed weighted linkage with other pages in both graphs is our target. We use an intuitive graphic way to show this difference as displayed in Section 2.2.

5.3.3 Case studies on frequent patterns

Here we give three case studies on the frequent patterns. They reveal and illustrate the usefulness and significance of web usage mining from different aspects.

Case 1: Optimizing Web Service Two operations, adding and removing hyperlinks, are used to optimize web services. But in reality, adding hyperlinks is much acceptable than removing hyperlinks. The reason is that judging the ″unfriendly″ and ″improperly″ links among pages is much controversial than finding and judging the frequent and potential useful accessed relations.

In the old version of our group web pages, the ″team page″ was linked by the anchor texts in the research page and the project page. However, it was not listed in the top navigation bar, which means that the visitors could not reach the ″team page″ directly from the home page. By discovering the popular web navigation paths in usage logs, we found there are two prominent paths: one ended at ″team page″ starting from the home page via the research page, while the other ended at the same ″team page″ via the project page, which are shown by the solid lines in Figure 5.12. So we prose to add a link for ″team page″ in the top navigation bar on the homepage, which is shown as the dashed line in Figure 5.12.

Case 2: Potential Commercial Model Figure 5.13 shows how the visitors found and fulfilled their bachelor applications on HPI site. The page on bachelor application from HPI site was accessed by groups of visitors via different paths ordered by support numbers. The top three paths are: the page on bachelor study; the page on study in HPI which links to the pages on

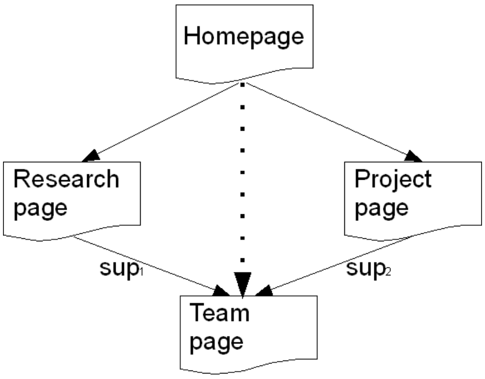


Figure 5.12: *Two Frequently Accessed Sequences on how was "team page" accessed*

bachelor, master and doctor studies; and the page on teaching information. The first path tells that some visitors accessed the bachelor study page by using Google, and after visited other pages, they reached the bachelor application page. Before reaching the bachelor application page, there could be some visitors accessed the teaching page after the bachelor page, however, the number of such visitors is less than our defined support threshold.

The different ways on fulfilling the bachelor application present the different effectiveness of different information medium. Considering that the number of applications on bachelor study is a key indicator for the success of an institute, from the patterns in Figure 5.13, we extract a model on evaluating the success of an e-Commerce, which is shown in Figure 5.14.

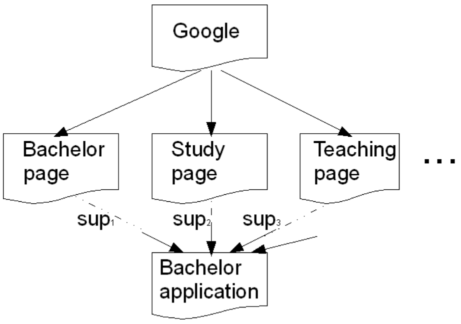


Figure 5.13: *Three Navigation Sequences on how was "bachelor application" reached*

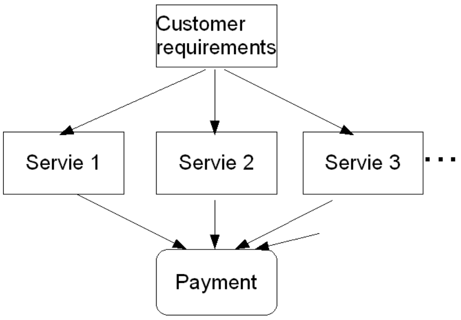


Figure 5.14: *Abstracted Model on evaluating an e-Commerce*

The navigation patterns in ***Case 1*** and ***Case 2*** are all frequently accessed page sequences, which describe the page orders on how did group of visitors access the pages. ***Case 1*** answers the question in Scenario 1 and ***Case 2*** for that in Scenario 2 raised in Chapter 4. Different from ***Case 1*** where an extra link is proposed to be added on the navigation bar, we do not suggest to add any links among the pages in ***Case 2***. This is based on the different assumptions and understandings: in ***Case 1***, we think the "team page" has the same importance and plays the same role as research page and project page in a research group; however, in ***Case 2***, it is treated that the "bachelor application" should only be executed by the visitors after they view and are familiar with the related information.

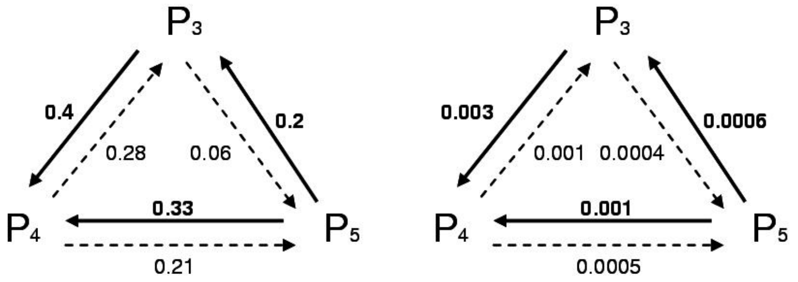


Figure 5.15: *Page Clusters base on CS and DS in March Logs*

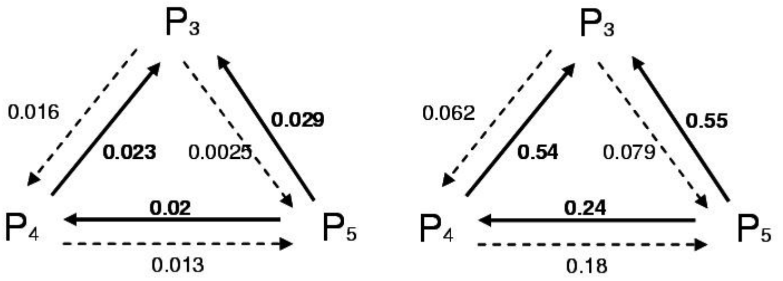


Figure 5.16: *Page Clusters base on CS and DS in April Logs*

Case 3: Content Cluster based on Pair Relations

P_3=*/lehre/vorlesungen.html*, P_4=*/lehre/bachlor.html*, and P_5=*/lehre/master.html*.

These three pages have the same semantic importance, because they are linked from the same source page "*/lehre.html*". This means that these three pages have no bias on the web designer's side. But based on page pairs modeled from usage data, these three pages have some clear bias on heuristic importance, which show the clear difference in usage view. The following two Figure 5.15 and Figure 5.16 display the binary relations based on conditional possibility (CS) an heuristic importance (DS) on March logs and April logs.

In the above two figures, we discriminate the different directions of heuristic importance within a page pair by using different lines: the bold line means a higher heuristic importance and the dashed line means a lower heuristic importance within the same page pair. From the four page clusters in these two figures, we find P_5 has a higher heuristic importance to P_3 and P_4 than those from P_3 and P_4 to P_5, which happened in two different period of logs based on two different models. Based on task-oriented evaluating measurements listed in Section 4.5, we can naturally conclude that $P_3 \leftarrow P_5 \rightarrow P_4$ is a very useful page cluster, which helps for improving content organization. This partly answers the question in Scenario 3 introduced in Chapter 4.

5.4 X-tracking the Changes of Usage Patterns

In the above sections, the usage patterns are mined on the whole data set, which overlooks the possible changes of the usage data collected from different periods. In this section, we try to gain the insights on the changes of how visitors browsed their information.

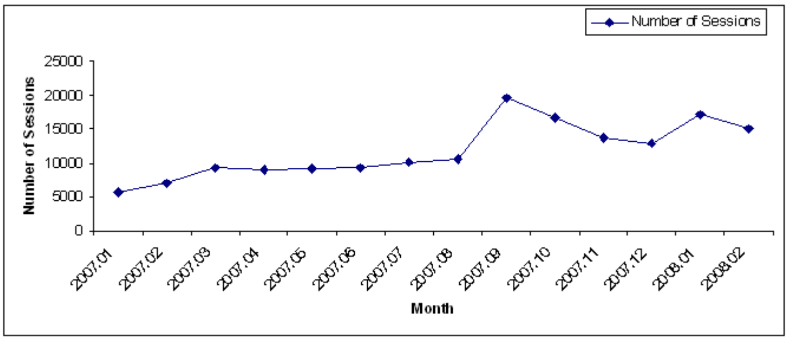


Figure 5.17: *Distribution of the Size of Session Sets*

On **Data Set 1** (DS1), the valid usage sessions are monthly split into 14 sub sets, and for each sub set, we mined three kinds of navigation patterns: frequent page sets, frequent navigation sequences and frequent tree structures. Maximum frequent patterns are further mined (a frequent pattern X is a maximal frequent if there is no frequent pattern X' such that X is a sub pattern of X'). The distribution of the sizes of session sets corresponding time spans is showed in Figure 5.17.

On each sub session set, we mine three kinds of web navigation patterns under different thresholds of support value. Figure 5.18 shows the number of patterns in different time spans. The set of frequent accessed sequences, in 2007.05, compared with other time spans, has the biggest number of sequences under threshold $\theta = 0.01$, but has a relative small number under $\theta = 0.02$. On the other hand, we can discover the same results in (a) of Figure 5.18. The numbers of navigation patterns are decided by the thresholds and the types of patterns, however, the variations or differences of these numbers are manipulated by the compositions of individual behaviors and such differences are multi-facets. From Figure 5.18, we can discover as well that 2007.09 time span has the biggest number of sessions, but a smaller number of navigation patterns under the same threshold compared with other time spans. The variations of the numbers of patterns could not show the difference of the compositions of patterns.

5.4.1 Site semantics extraction

Meanwhile, we keep the copies of web site for each time span, which are used for extracting semantics of web pages. It is time-consuming work to track and save every change and update of a web site, and some changes are the uselessness noise. For a static web site, in which the information and structure are relatively stable, it is easy to keep the "useful" changes in its content or structure. However, for a dynamic web site, such as e-learning web site and social site providing instant web services, in which the content of a web page with a stable URL is updated frequently and unexpectedly, it becomes incorrect to track these changes based on fetching URLs. In such cases, a mapping or a convertor is required to immigrate from page-unit model to item-unit (such as news, goods, e-lectures and etc.) model, we will investigated this topic in Part II and Part III of this thesis. The web site in our experiments is a static education web portal, its content and structure are relatively stable, and the changes of updates and edits happened in a small scale and were tracked by periodically crawling.

Our intention is to extract the most important terms indicating the target web documents, while not to supply convenience for search engines. As discussed in Section 3.5, the automatically indexed terms need to be condensed and purified due to the existence of synonyms, common terms for n-grams and mistypings. One hardness is the multi-linguistic environment, we kept the German words for the pages written in German while removed the "meaningful" German words for the pages in English. Another hardness is removing the negative effects of the terms extracted from the common areas in web pages like guiding, contact and advertising information. Moreover, due to the relatively free creations of web pages in which a term having a high weight would be redundant for a page, the dimension of extracted terms for every pages based on these steps has to be further reduced. The selection of representative and discriminative terms is a kind of trivial and recursively manual work because the "right" terms are task depended and domain biased. However, two basic principles are obeyed for terms selection: one is that a selected term should have a relatively high weight, which means that the common terms happening in most documents should be removed; the other is that one selected term should have the highest occurrences web documents compared with its synonyms, and its occurrences should encapsulate those of its synonyms.

We selected 2851 web pages from our target site, and *Lucene* generated 43126 searchable terms, from which 20654 terms are removed as n-grams, 2640 numeric terms are out filtered. After removing 6392 non-significant words including pronouns, articles, adjectives, adverbs and prepositions, 13440 meaningful term candidates are extracted and the average number of extracted terms for each page is 120. Table 5.3 shows the extracted terms for web pages about organization and activities of a research group.

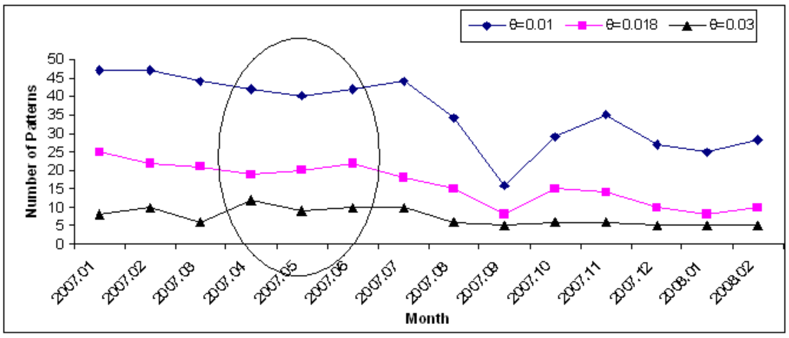
5.4.2 Frequent patterns

For the convenience to explore the changes hidden in groups of patterns from different time spans, we use t_i $(0 \leq i \leq 13)$ to represent 14 time spans from 2007.01 to 2008.02, $t_{i \to i+1}$ $(0 \leq i \leq 13)$to name the *jump* from t_i to t_{i+1}, and $t_{i \to ..j}$ for the series of jumps for from t_i to t_j $(0 \leq i < j \leq 13)$.

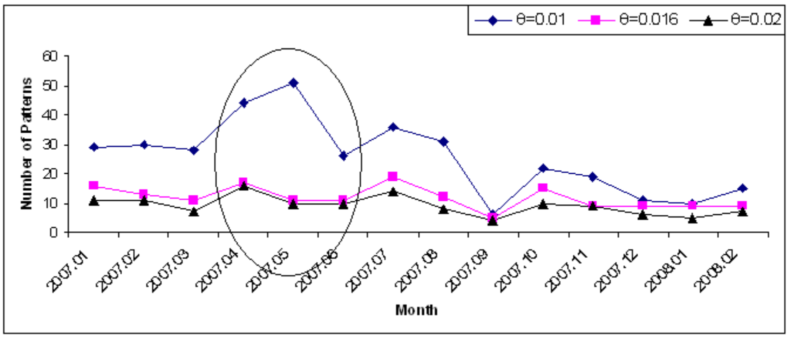
As explained in Section 4.4, after detecting internal differential, the set of navigation patterns from one time span is divided into three sub sets: *emerged*, *unchanged* and *partly changed* patterns. For an emerged pattern, its changes are computed by comparing with its unchanged patterns in posterior time spans. In this detection process, two thresholds, δ_s and δ_u, are needed: δ_s is for discriminating "*similar*" or "*emerged*" patterns, and δ_u is for selecting "*unchanged*" out of "*similar*" patterns. In our experiment, we set $\delta_u = 0.3$ and $\delta_s = 0.5$, and the distribution of emerged patterns at different time spans is given in Figure 5.19. In this figure, the gap between the "num_raw" and "num_unchanged" presents the number of emerged patterns at different time spans. The high percentage of emerged patterns compared with unchanged patterns reminds the big internal change of usage patterns, which reflects the semantical drifting of usage interest. Shown from Figure 5.19, all patterns at $t0$ are newly emerged, and $t8$ has a highest percentage of emerged patterns. Given a threshold on this percentage, pattern sets with the remarkable internal change could be selected, and the most unimportant patterns are removed. This is one significance of detecting internal changes among navigation patterns. However, only concerning the occurrence of newly emerged patterns could not reveal their evolution in the posterior time spans and the relationships among the unchanged patterns in different time spans. We track the life cycle of one emerged pattern by computing its external change, local and global popularity change with its unchanged patterns in posterior time spans.

Table 5.3: *Extracted Site Semantics for Web Pages*

URL	Top-K words with weights
/meinel/biography.html	university:28.77 complexity:21.91 computer:19.77 professor:19.50 director:18.78 science:16.10 c4:14.72 research:13.62 trier:12.99 board:12.52 ...
/meinel/boards.html	board:31.30 symposia:29.44 trier:25.98 director:25.04 confidant:22.08 symposium:17.52 series:14.73 chairmen:14.72 lecturerer:14.72 gesellschaft:14.62 university:14.38 ...
/meinel/teaching.html	ss:36.17 internet:29.11 task:17.63 ws:16.63 www:15.45 tele:15.38 beijing:14.48 grundlagen:11.79 bachelor:10.39 tu:9.043 master:6.591 lecture:6.437 ...
/meinel/research.html	internet:27.03 tele:24.16 complexity:21.91 task:21.15 computational:18.87 lab:18.80 research:17.51 work:16.83 bridge:16.70 eccc:15.66 design:15.38 exploration:14.72 ...
/meinel/research/soasecurity.html	soa:58.35 menzel:20.63 ivonne:17.24 service:15.67 michael:15.40 trust:14.48 thomas:14.15 alnemr:12.88 wolter:12.52 rehab:12.21 cross:10.97 ...
/meinel/research/web_university/tele-task_podcasts.html	task:52.89 tele:32.95 podcast:23.89 picture:22.99 video:19.65 desktop:17.92 clips:14.72 ipod:14.72 trailer:14.72 mpeg4:13.91 presenter:13.91 toc:13.91 ...
/meinel/tele-lectures.html	internet:166.3 vorlesung:88.55 www:86.54 grundlagen:82.56 view:70.23 krypto:67.17 attacks:57.04 course:51.48 task:49.36 besprochen:48.08 behandelt:47.95 summer:42.22 ...



(a) Frequent Page Sets



(b) Frequent Sequences

Figure 5.18: *Distribution of the number of different kinds Patterns*

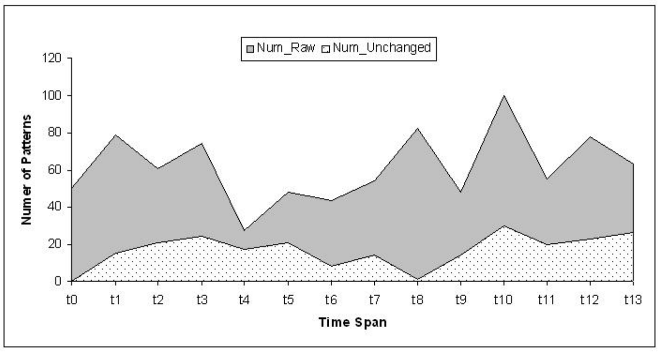
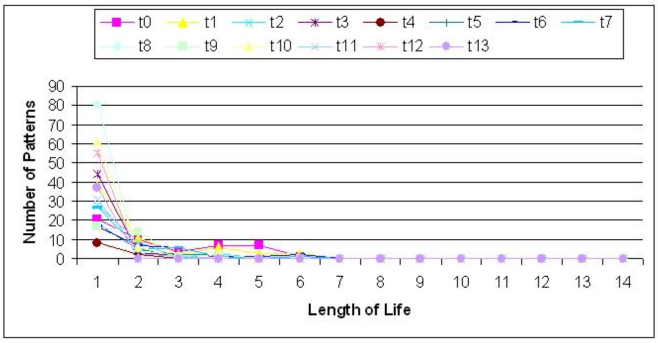


Figure 5.19: *EP v.s. UP under* $\Theta = 0.01$, $\delta_u = 0.3$ *and* $\delta_s = 0.5$

The life time of a pattern X_i is the number of time spans, in which X_i has its versions. Shown in Figure 5.20, a pattern which emerged in a priori time span has a higher possibility to have a long life time than that emerged in a posterior time span, this is usually related to the periodic updates of the web site. However, a pattern with only 1 life length indicates a temporal variation of usage interest and such variation could be unexpectedly interesting, noise from robots or due to the special events. Though $t8$ has the highest percentage of newly emerged patterns due to the



Frequent Navigation Paths

Figure 5.20: *Life length for emerged patterns at different time spans*

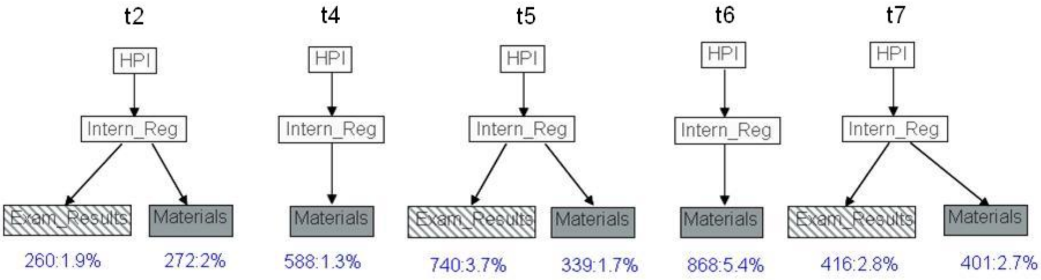


Figure 5.21: *An example for X-tracking*

big updates on the site, these patterns did not survive after $t8$. In practice, the manager usually has interest on the changing history of a runtime pattern, especially the changes on its external form and support value suffered in its life cycle. This can be easily drawn by querying the DB table about the relations between every emerged pattern and its versions with their external and local popularity changes.

5.4.3 Case studies on change detections

The example shown in Figure 5.21 gives the evolution of one emerged pattern on how did students accessed exam results and related materials. We can observe that the students accessed the content in a periodic regular way. In $t2$ time span, the students usually visited the page on the examination result, and in $t4$, this usage interest did not happen (because it did not get enough support); however, again, the usage interest on examination result appeared in $t5$ span and disappeared in $t6$ span. This periodical usage interest highly depends on the schedule of course and examination in HPI. The interesting thing is that **this periodical usage interest would not be discovered if we implemented the mining process on the whole time line, while not separately on different time spans**.

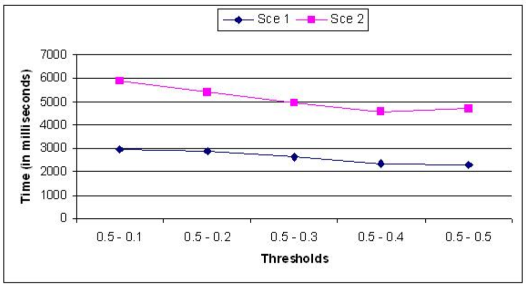


Figure 5.22: *Time results on x-tracking in two scenarios*

5.4.4 Time analysis

Here we only discuss the performance of x-tracking algorithm. The sizes of patterns and pattern sets have great impact on the effectiveness and efficiency; and the internal differential computation between two pages relies on the dimension of the extracted terms. The biggest part of time consuming is semantic extraction and data cleaning before mining and x-tracking, and depends highly on the manually purifying terms and data. Two thresholds θ_s and θ_u decide the number of emerged patterns for every time span and further affect the times on tracking the versions of internal unchanged patterns.

We used two pattern sets to investigate the time consuming on change detections: in pattern set 1, there are altogether 960 patterns in 14 time spans, and the average size of pattern is 5.8; while in pattern set 2, there are 585 patterns, and an average size for a pattern is 4.4. Figure 5.22 collects the time consuming on x-tracking changes among patterns in these two pattern sets. From this figure, we see that x-tracking algorithm is effective in a reasonable application, though the time consuming is theoretical $O(n^2)$, increasing with the increasing of pattern sets and their lengths.

5.5 Summary for This Chapter

In this chapter, we have discussed the experiment on discovering various usage patterns on a portal site, on which the content and the structure of web pages are relatively stable. We started from rebuilding the individual accessing behaviors from two session sets: one is from HPI site and the other is from ECCC site. Based on the rebuilt individual behaviors, we explained from macro and micro views the frequently co-accessed page sets, and page sequences and tree structures, and we used some case studies to show the values of usage patterns. Specially, we concentrated on the experiment discussion on the page clustering based on pair browsing and tracking the changes of navigation patterns.

From the experiment, we conclude that the usage interest and navigation patterns on a web site are modeled and mined in multi ways, and the usage patterns can not be restricted in a general and universal model. **The significance of usage mining is to depict and discover the usage patterns in a more compressed, structured, direct and understandable way from the huge, unstructured, indirect and un-interpretable usage data**. However, the judge and selection of the interesting usage patterns are highly subjective, though there exist kinds of technical measurements based on statistical possibilities.

In the first part of this thesis, we emphasized the usage mining on a relatively stable web site.

In this case, the relation between page content and page URI is unchanged within a longer time span. However, in an e-learning site like www.tele-task.de, on which the web pages are frequently updated with the newly recorded lectures, we have to shift to relations between the visitors and the lectures while not the web pages. Further, the time spent on a lecture is confidently measured and will play its importance in measuring and discovering the learning interest in an e-learning web site. This will be discussed in the Part II: mining the learning interest in a web-streaming e-learning site.

Part II

Mining the Learning Interest in a Web-Streaming E-learning Site

Chapter 6

TASK-Moniminer: An Engine to Query the Learning Interest on tele-TASK

Task-Moniminer (**Moni**tor and **Miner**) concerns on discovering the students' learning interest from their learning history recorded as usage log data in a web-streaming e-learning environment, which is the further and auxiliary project for tele-TASK. It supplies a search engine to query the usage information on e-lectures in an intuitive graphic way.

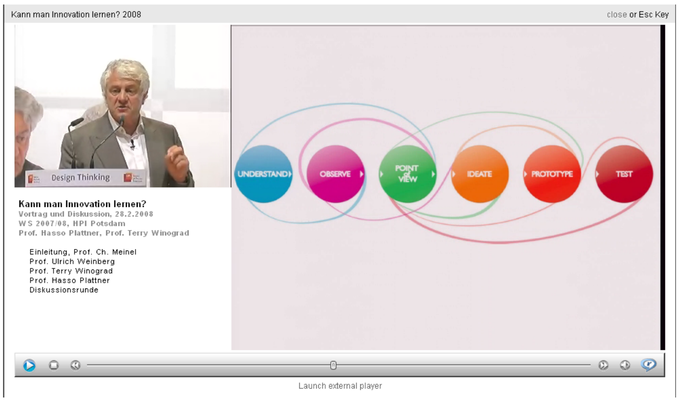
Chapter Organization We shortly introduce tele-TASK system in Section 6.1, especially focus on the attributes of e-lectures. We then give the implementation of TASK-Moniminer in section 6.2.

6.1 tele-TASK: A Web-Based e-Learning Environment

Tele-TASK (**T**eaching **A**nywhere **S**olution **K**its) (Schillings and Meinel 2002) supplies a portable and powerful solution for distance education. From 2001 till 02.2009, tele-TASK has recorded over 2180 different lectures and altogether more than 3000 hours length recordings, and it has as well served in symposiums, conferences and other public events. All the lectures, multimedia recordings and other related materials are presented on web site: www.tele-task.de, which serves as a web-based distance learning platform. In this thesis, we refer tele-TASK as its web site system, while not the lecture recording system. Students and interested surfers can freely follow the live broadcasting of the ongoing conferences or the lectures in web-streaming formats by using web browser. All the web-streaming lectures and recordings are encoded in Real streaming format, and every lecture is embedded in a web page for online learners to browse.

The layout of one lecture page is divided into two parts: the left part and the right part. The left part is the outline of one whole course, which includes all the relevant lectures, and the text of each lecture name is linked to its streaming files. In most cases, one course includes several units (or chapters), in which there are several different lectures. The right part embeds the frame of the Real formatted streaming lecture. The frame of one streaming lecture is characterized by three fields: the top left field displays the "talking head" of the teacher synchronized with audio signal; the bottom left field writes the table of content (TOC) of the lecture, and each text line in TOC links directly to the right position in the video that discusses the related knowledge, which helps students to find their interesting knowledge easily and directly; the big right field shows the presentation slides/desktop or writing pad of the teacher. The Figure 6.1 shows the snapshot of one lecture in Real format.

Each multimedia lecture has definitive attributes after recording: **Name**, **Live Stream URL**, **Lecture Stream URL**, **Duration**, **Recorded Time**, **Table of Content**, **Course Name** and other attributes such as **lecturer** and **logo**.

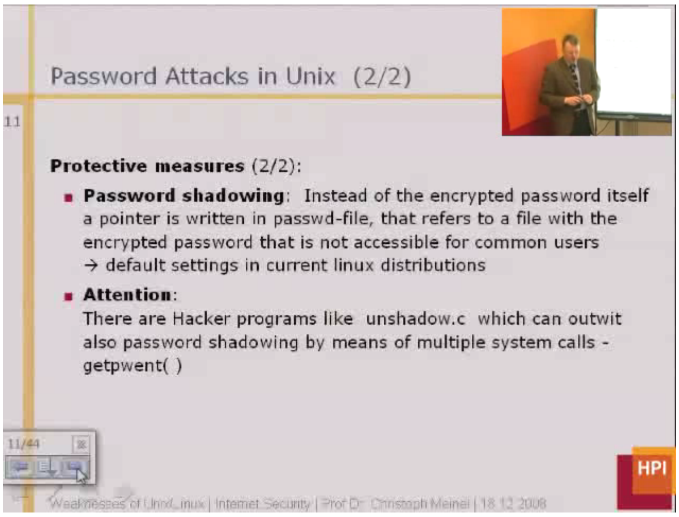

Kann man Innovation lernen? 2008

close or Esc Key

Kann man Innovation lernen?
Vortrag und Diskussion, 28.2.2008
WS 2007/08, HPI Potsdam
Prof. Hasso Plattner, Prof. Terry Winograd

Einleitung, Prof. Ch. Meinel
Prof. Ulrich Weinberg
Prof. Terry Winograd
Prof. Hasso Plattner
Diskussionsrunde

Figure 6.1: *One Lecture View on tele-TASK*


Password Attacks in Unix (2/2)

Protective measures (2/2):

- **Password shadowing**: Instead of the encrypted password itself a pointer is written in passwd-file, that refers to a file with the encrypted password that is not accessible for common users
  → default settings in current linux distributions
- **Attention**:
  There are Hacker programs like unshadow.c which can outwit also password shadowing by means of multiple system calls - getpwent( )

11
11/44
HPI
Weaknesses of Unix/Linux | Internet Security | Prof Dr. Christoph Meinel | 18.12.2008

Figure 6.2: *Lecture in mp4 or flash format*

With the wide acceptance of the web sites on i-Pods and social medias, tele-TASK also supplies the e-lectures in mp4, flash formats and IPTV. The layout of mp4 and flash formatted video is different from that of Real formatted video, which removes the TOC part and sets the video in a corner over the desktop. However, the layout and the position among the desktop and the video can be personalized during the re-encoding after the recording. Depending on the topics listed in TOC, a recorded lecture in Real format is cut and re-edited into several clips in mp4 and flash formats. A 90 minutes lecture could be divided into 9 clips, each of which is 10 minutes long on average and dedicates on illustrating one topic. A snapshot of a mp4 clip is shown in Figure 6.2.

The lectures are published on the web site in an anti-chronological order, which means the latest recorded lecture is added at the beginning of the homepage. The main content components of tele-task web site are courses. Every course is presented in a web page starting with a short description and listing all the lectures it has. One lecture listed on a course page is linked to a lecture page, which shows its TOC and icons to play the whole Real formatted lecture, or mp4 and flash clips. A user can browse one lecture by entering the homepage, or clicking the course page, or using the internal and external search engines. This brief introduction of the organization of tele-task site is necessary due to the requirement on investigating the learning interest in video-based e-learning systems.

Though the increasing of the access number on our web site and more and more recording requests convince us that tele-TASK helps to partly satisfy the great requirements in distance education, we do not know if the web lectures are well used by the students, and if there are some preference on different lectures. This is the motivation for this work: we are trying to mine students' learning interest from their browsing behaviors, which could help us to know the learners and their learning interest, and to optimize the organization of the web site and the lectures.

6.2 Implementation of TASK-Monimier

Different from HPI site in which the content and the URLs of the web pages are relatively stable, the content of the web pages on tele-TASK site is always changing because of the frequently inserting of the newly recorded lectures and videos. This shows that method on detecting learning interest on tele-TASK is different from that on HPI site.

TASK-Moniminer enlarges the territory of routine search engines: it not only outputs the e-lectures related with the query terms, but shows their usage information of visitors. TASK-Moniminer supplies two possibilities to query the usage information on e-lectures as shown in Figure 6.3: one is by selecting one course listed in the categories from the drop-drown menu; and the other is by inputting keywords. The former gives the usage of e-lectures included in the selected course, for example, usage on the course of ″Internet Weaknesses and Targets″ in summer semester 2008, or on the recorded presentations in ″HPI Colloquium″. The latter allows querying the usage on the e-lectures related by the input keywords, for instance, keywords ″windows security″ answer the related lectures in courses ″Internet Security″ and ″Operating Systems″, and the recordings on ″lock-keeper″ as well.

Figure 6.4 displays the results on querying ″TCP IP″. It gives all the videos about ″TCP/IP″ in Real, mp4 and flash format. The videos in Real format are transmitted through ″RTSP″ protocol by which clicks and staying time information can be estimated. The usage on a Real formatted lecture includes the following aspects:

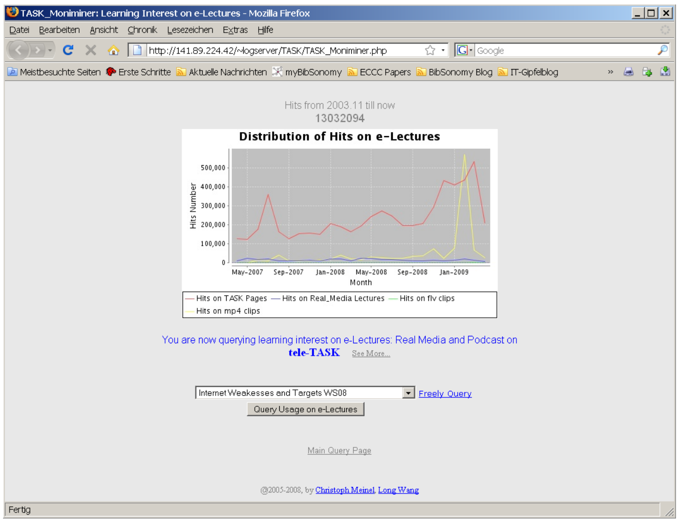
1. visitor IP distribution: the percentage of IP from the networks of HPI staff, students and others, we can see that most of the users viewed the lectures from outer HPI;

2. average hits per access: number of clicks such as pause, sliding and stopping on one lecture;

3. average time per access: time length on viewing a lecture, for instance, the visitors from outer HPI network spent 50 minutes on ″windows operation system″ of winter semester 2007; and

4. weighted usage: usage computed based on number of visitors, hits and time, for example, the weighted usage for the lecture ″Internet - First Introduction″ is 0.242 from Figure 6.4.

The weighted usage of a lecture is computed based on the linear accumulation of number of visitors, hits and time. The parameters for these three variables are adjustable in a configure


TASK_Moniminer: Learning Interest on e-Lectures - Mozilla Firefox
Datei Bearbeiten Ansicht Chronik Lesezeichen Extras Hilfe
http://141.89.224.42/~logserver/TASK/TASK_Moniminer.php
Google
Meistbesuchte Seiten Erste Schritte Aktuelle Nachrichten myBibSonomy ECCC Papers BibSonomy Blog IT-Gipfelblog
Distribution of Hits on e-Lectures
Hits Number
500,000
400,000
300,000
200,000
100,000
0
May-2007 Sep-2007 Jan-2008 May-2008 Sep-2008 Jan-2009
Month
Hits on TASK Pages
Hits on Real_Media Lectures
Hits on flv clips
Hits on mp4 clips
Fertig

Hits from 2003.11 till now
13032094

You are now querying learning interest on e-Lectures: Real Media and Podcast on **tele-TASK** See More...

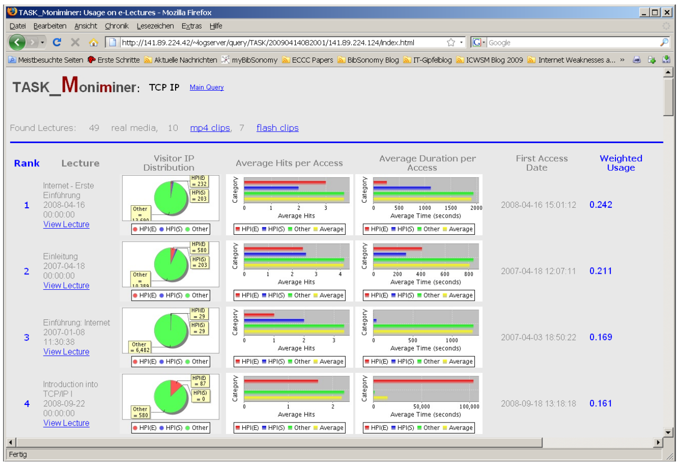
Internet Weakesses and Targets WS08 ▾ Freely Query

Query Usage on e-Lectures

Main Query Page

@2005-2008, by Christoph Meinel, Long Wang

Figure 6.3: *Query interface of TASK-Moniminer*


TASK_Moniminer: Usage on e-Lectures - Mozilla Firefox
Datei Bearbeiten Ansicht Chronik Lesezeichen Extras Hilfe
http://141.89.224.42/~logserver/query/TASK/20090414082001/141.89.224.124/index.html
Google
Meistbesuchte Seiten Erste Schritte Aktuelle Nachrichten myBibSonomy ECCC Papers BibSonomy Blog IT-Gipfelblog ICWSM Blog 2009 Internet Weaknesses a...
HPI(E) = 232
HPI(S) = 203
Other = 13,690
HPI(E) = 580
HPI(S) = 203
Other = 10,389
HPI(E) = 29
HPI(S) = 29
Other = 6,482
HPI(E) = 87
HPI(S) = 0
Other = 580
HPI(E) HPI(S) Other
Category
Average Hits
Average Time (seconds)
HPI(E) HPI(S) Other Average
0 1 2 3
0 500 1000 1500 2000
0 1 2 3 4
0 200 400 600 800
0 1 2 3
0 500 1000
0 1 2
0 50,000 100,000
Fertig

TASK_Moniminer: TCP IP Main Query

Found Lectures: 49 real media, 10 mp4 clips, 7 flash clips

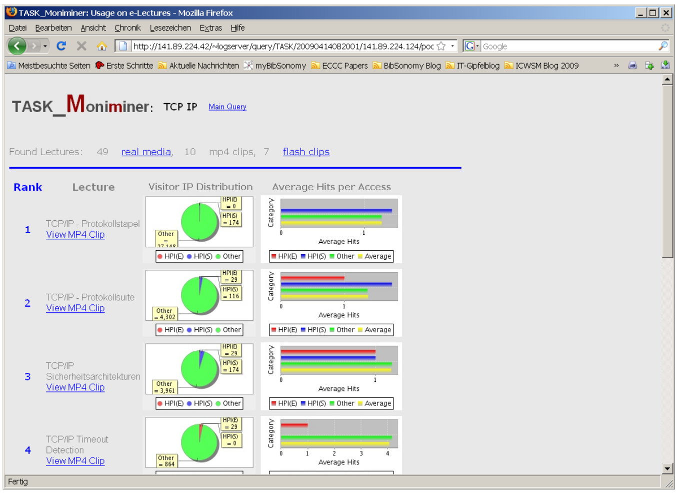
Rank	Lecture	Visitor IP Distribution	Average Hits per Access	Average Duration per Access	First Access Date	Weighted Usage
1	Internet - Erste Einführung 2008-04-16 00:00:00 View Lecture				2008-04-16 15:01:12	0.242
2	Einleitung 2007-04-18 00:00:00 View Lecture				2007-04-18 12:07:11	0.211
3	Einführung: Internet 2007-01-08 11:30:38 View Lecture				2007-04-03 18:50:22	0.169
4	Introduction into TCP/IP I 2008-09-22 00:00:00 View Lecture				2008-09-18 13:18:18	0.161

Figure 6.4: *Query results (real media) of TASK-Moniminer*

file. Besides these, first access date is showed as well for every lecture though it depends on the lecture's publish date.

The mp4 and flash clips related with the input keywords will be displayed as well, as shown in Figure 6.5. However, mp4 and flash clips are delivered by HTTP protocol by which a continuous connection could produces tens requests. And the viewing time on the clip can not be directly retrieved.

This query engine displays the usage information in a lecture-centered way which overlooks the relations between the lectures in online learning process. Moreover, an e-learning environment is different from other common web portal sites, in which the learning materials are videos in plenty lengths and big part of visitors are students who have clear browsing targets. The target of TASK-Moniminer is to investigate the learning interest in tele-teaching, which is more than



TASK_Moniminer: Usage on e-Lectures - Mozilla Firefox
Datei Bearbeiten Ansicht Chronik Lesezeichen Extras Hilfe
http://141.89.224.42/~logserver/query/TASK/20090414082001/141.89.224.124/poc
Google
Meistbesuchte Seiten Erste Schritte Aktuelle Nachrichten myBibSonomy ECCC Papers BibSonomy Blog IT-Gipfelblog ICWSM Blog 2009
Fertig

TASK_Moniminer: TCP IP Main Query

Found Lectures: 49 real media, 10 mp4 clips, 7 flash clips

Rank	Lecture	Visitor IP Distribution	Average Hits per Access
1	TCP/IP - Protokollstapel View MP4 Clip		
2	TCP/IP - Protokollsuite View MP4 Clip		
3	TCP/IP Sicherheitsarchitekturen View MP4 Clip		
4	TCP/IP Timeout Detection View MP4 Clip		

HPI(E) = 0
HPI(S) = 174
Other = 27,148
● HPI(E) ● HPI(S) ● Other
Category
0
1
Average Hits
■ HPI(E) ■ HPI(S) ■ Other ■ Average
HPI(E) = 29
HPI(S) = 116
Other = 4,302
● HPI(E) ● HPI(S) ● Other
Category
0
1
Average Hits
■ HPI(E) ■ HPI(S) ■ Other ■ Average
HPI(E) = 29
HPI(S) = 174
Other = 3,961
● HPI(E) ● HPI(S) ● Other
Category
0
1
Average Hits
■ HPI(E) ■ HPI(S) ■ Other ■ Average
HPI(E) = 29
HPI(S) = 0
Other = 864
Category
0
1
2
3
4
Average Hits

Figure 6.5: *Query results (mp4) of TASK-Moniminer*

those supplied by the query engine. And the data preparation work in e-learning environment is much trivial than processing HTTP server logs. In the next three chapters, we will discuss in details the work on data preparation, and discovering learning interest in an e-learning environment.

Chapter 7

Issues on Mining Learning Interest in the tele-Teaching Environment

Web-Streaming lectures overcome the space and time barriers between learning and teaching, but bring higher requirements on the learning feedback of students when they browse the lectures. The use of the Internet as an instructional tool in higher education is rapidly increasing. Today, there is an increase in the development of academic course web sites with huge amounts of learning materials imbedded within them.

Current distance education environments work mainly as the secondary supplement for the conventional education. In conventional education, the teachers and education supervisors can ask the students face-to-face or use anonymous questionnaires to judge if their lectures taught in the classroom are welcomed or not. In other web services, such as online shopping or e-communities, it is relatively easy to evaluate the success of their services by the changing number of online bills or the number of registered members. However, in the tele-teaching environment, there is little empirical evidence regarding the actual use of the teaching materials by the students.

In this thesis, the usage logs are used to evaluate how online content was consumed, and to identify the individual differences on the learning content that is presented in an e-learning site. The usage log files in particular are an intriguing monitoring technique for on-line courses. Through these log files, we can detect how students navigated the lectures and can receive feedback on their learning modes. In other applications, log files also permit us to dynamically record the portions of the student problem-solving process and use it as a feedback mechanism for the instructor; and can even aid us in improving our standard assignments and tests by allowing us to see how students engage these tests.

In distance education environments, especially in free and open environments only supplying web-based streaming lectures, it becomes much more difficult to evaluate the success of online lectures. Surely, the changing number of accessing is an important indicator, but it is not enough. It was said that ”using hits and page views to judge site success is like evaluating a musical performance by its volume” (Schmitt et al. 1999).

Chapter Organization This chapter presents the related work on e-learning systems in Section 7.1, and the work on the evaluation of the participation in e-learning is listed in Section 7.2. We discuss the limitations and challenges in the evaluation in Section 7.3, and give out our mining tasks at the end in Section 7.4.

7.1 Relate Works in Web Video-based e-Learning Environment

One primary task of e-Learning systems is to supply e-lectures for different online learners. The embedding of streaming videos in distance learning has received great interest (Reynolds and

Mason 2002)(Schillings and Meinel 2002)(Reisslein et al. 2005). The basic feature of a web-based e-learning environment is to supply a number of different multimedia lectures for learners in semesters, and the fresh lectures are presented at the prominent positions on the web, while the lectures for the past semesters can be accessed in the archive pages. An ideal web-based e-learning environment is that: it displays not only the multimedia lectures, but also exercises, and even serves the final exams, which were also the requirements retrieved by Microsoft's investigation shown at the beginning of this thesis. But up to our acknowledge, till now there is no such perfect e-learning environment, and the web-based distance education systems, which serve mainly on supplying web-streaming lectures, are currently the most popular, reliable and efficient solutions.

(Reynolds and Mason 2002) studied the impact of the number of windows in web-streaming distance education video: one window video showing either the instructor or presentation slides / instructor writing pad; two-window distance education video, where one window displayed the talking head of the instructor and the second window displayed the presentation slides/writing pad; three-window distance education interface, where a live chat window was added.

(Reisslein et al. 2005) stated that the web-streaming distance education offered today falls mainly into the one-way video and audio profile. The class video (along with the instructor audio) is typically recorded in a classroom studio (often filled with on-campus students) and posted on the class web site a few hours after the recording. The distance learners can then view the class video by streaming it from the class web site and interacted with the instructor asynchronously, e.g. via e-mail or web-based discussion boards.

7.2 Related Works on Evaluating Participation in e-Learning

The web-based e-learning systems facilitate teachers and learners greatly, but they can also lead to the frustration for both of them, because the visual and aural cues (eye contact, body language, facial expressions and voice tone) of online learners are missing compared to the education in face-to-face classroom. Online course evaluation is essential in order to improve the quality of teaching and learning. As discussed in Chapter 1 on the methods of investigating the usage interest, direct and indirect ways are required in evaluating participation in e-Learning. Several ways have been used to evaluate the online courses and e-learning participation, such as teacher/student reflection (questionnaires), student performance in assessments (assignments, quiz and examination) and student actions (web server log files and log databases).

It is proven that different browsing strategies are used in different types of hypertext interfaces (McAleese 1999). Therefore it is necessary to enquire whether the type of hypertext architecture employed has any effect on the browsing strategies of individuals with different cognitive styles. In e-learning environments, different mediums require different ways of evaluating student participation to ensure if the necessary knowledge or skills have been grasped during their learning.

A tool was presented aiming to track and analyze individual learner behavior during his interaction with e-learning environment (Hardy et al. 2004). And they suggested the further investigation on finding interesting patterns and navigation paths over set of single routers.

An education data mining tool was developed in (Mostow et al. 2005), which listens to the children when they read sentences and helps them learning how to read, but this application is not suitable for free and open web-based e-learning environment in high education, where the relationship between tutors and students is very loose and unstable.

A teacher's questionnaire was reported to identify the needs of teachers to know their students and to make distance learning a less detached experience (Zinn and Scheuer 2006). They showed that the current e-learning environments have to be improved to satisfy teachers needs of tracking students in distance learning contexts.

Besides, summative assessment and formative assessment have been introduced in learning performance evaluation (Torrance and Pryor 1998). The summative evaluation is generally performed after finishing an instruction unit or class, while formative assessment emphasizes the learning process.

7.3 Difficulties and Arguments on Evaluation

E-learning has been growing rapidly since the Internet and the new training technologies became widely available in the middle of the 1990s. What needs to be investigated is whether online teaching, training and learning have any tangible benefits in terms of improving student learning as measured by final grades or time consuming. However, evaluating the benefits of attending e-learning, or the participation of students is challengeable, and becomes even much controversial in unauthorized learning portal site for high education.

The reasons for the challengeable evaluation on e-learning come from two sides: **pedagogy** and **technology**. The challenges from **pedagogic** side are:

- the learning channel is not universal, which makes it harder to bounder the contributions from different channels;
- the e-learning attendants can be trainees from enterprisers, undergraduates from universities, or students from middle schools;
- the online teaching materials have different formats: PPT slides, video lectures, practical network-based lab, online exercises, or interactive discussions between teaching and learning; and
- compared with face-to-face classroom, e-learning system requires efficient technology to collect and understand the feedback from learners.

The reasons listed above decide it is not realistic to find a universal methodology to evaluate e-learning system. For instance, the evaluating method on the video-based lectures is different from that on practical lab; the strategy on judging participation of students in online discussion does not fit to assessing the language speaking in kinder garden. It is the efficient and economic way to deliver the lectures to the distributed students by using web servers to publish the recorded lectures, such as tele-TASK site. On the other hand, evaluating the e-learning benefit in this case attracts much attention and discussion. In this thesis, we concentrate on discovering the learning interest in unauthorized video-based e-learning systems, based on our tele-TASK platform.

Besides the common challenges for all e-learning systems, the challenges from **technical** side on discovering the learning interest in unauthorized video-based e-learning systems are:

- the task on discriminating the accesses of students from no-students becomes much crucial than in authorized e-learning systems;
- the data of students' actions on videos is incomplete and inaccurate since the shortage of tracking the actions on media players; and
- it is not easy to estimate the role of video-based lectures in the whole learning process.

7.4 Our Tasks

To know the learning interest in a web-based learning environment, such as tele-TASK which primarily delivers the multimedia lectures, the teachers want some methods to quantify the learning interest. For example, the questions raised by the teachers are listed in the followings:

1. *Is there any difference between viewing the live broadcasting lectures and browsing lectures after they are recorded and edited?*
2. *Is there any preference on the different lectures in a course and preference on the different pieces of one lecture?*
3. *Is there any favor among real, mp4 and flash formats?*
4. *For one lecture, is the real video viewed together with its mp4 and flash clips?*
5. *Do the students view other lectures when they access one lecture?*
6. *For the same named course supplied for different years, is there any change on the students' interest?*
7. *How often do the students browse online lectures when they do their exercise?*
8. *How different on learning interest between man and woman?*

The questions related with the learning interest could be raised more than those listed above. Different questions need different methods, direct or indirect to find the right answers. The direct ways are the methods such as questionnaires, participatory observations and interviews, while the indirect way is using the tools that can discover and evaluate student on-line activity from computer-generated log files.

However, the indirect way can not solve all the above questions, even in face-to-face classroom where direct questioning and answering are used, knowing students correctly is always the topic in pedagogy. In the next two chapters, we focus on answering the first six questions by different mining methods: general statistics, associate rules and similarity comparing. The learning interest is mined from student learning profiles, which are transformed from heterogenous usage data. Question 7 and 8 are suitable for the direct ways like questionnaires and interviews. The work discussed in Part II of this thesis can be refereed in (Wang and Meinel 2007a) and (Wang and Meinel 2007b).

Chapter 8

Modeling and Discovering Learning Interest in Different Questions

In this chapter, we discover the students learning interest from their usage data in web video-based learning environment by using multi data mining methods. The learning interest is expressed in six questions, which were asked by the teachers. We use simple statistics, associate rules mining, multi linear regression and similarity comparing to answer different questions. The usage data of online learners are heterogeneous, including HTTP server logs and REAL Helix Universal logs. And before mining learning interest, these heterogeneous usage data are transformed and uniformed into student browsing profiles.

Chapter Organization The outline of this chapter is as follows: we firstly discuss the necessary work on data preparation in Section 8.1. In Section 8.2, we present the methods on answering the six questions requested. We specially concentrate on finding the difference of learning interest on the same course serving for different years in Section 8.2.6. We give the summary of this Chapter in Section 8.3.

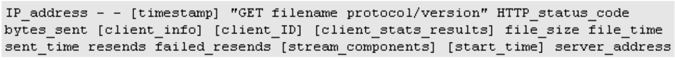
8.1 Data Preparation in e-Learning

This section explains how to filter and rebuild the browsing profiles of online students. We concentrate on this problem due to the extremely complexity and diversity of usage data in distance learning environment, which increase the difficulties to clean usage data.

8.1.1 Cleaning and integrating learning usage data

Web-based e-learning environments usually supply heterogeneous learning materials including text, audio and video, and store the usage data in different formats. In our case, tele-TASK web site records the normal surfing data on HTTP server in combined log format, and at the same time stores the usage data on streaming lectures which run on Real Helix server. Every web-streaming lecture is embedded in one page view, which means that when a student clicks one lecture link, two totally different usage log entries will be written on two different servers in different log formats.

HTTP server logs related with web usage mining have been fully discussed in Chapter 3, the useful usage information for each log entry includes: **IP address**, **request time**, **request file**, **user agent** and **referee link**. To understand the web usage patterns, it is required to know the accessing object from each log entry. In Part I of this thesis, the accessing objects are web pages, which are relatively stable on an information portal site. Dynamical web sites, which are characterized by generating the content of pages based on the input or the client configurations of the visitors, bring



```
IP_address - - [timestamp] "GET filename protocol/version" HTTP_status_code
bytes_sent [client_info] [client_ID] [client_stats_results] file_size file_time
sent_time resends failed_resends [stream_components] [start_time] server_address
```

Figure 8.1: *Helix Universal Access Log Format in Logging Style 3*

much more difficulties to retrieve the accessing objects from the requested files in server logs, and this task would be impossible if there is no extra supports, such as the site map, functionalities between the site structure and the physical file systems or the logging mechanism that records the detailed interactions between the server and the visitors.

For mining students learning patterns, the accessing objects such as lecture descriptions and content table have to be retrieved from HTTP server logs. A page view usually generates several log entries in server logs requesting different files. HTTP server logs record only the name or the path for requested files, without the content and semantic descriptions for them. Different requested paths may link to the same content, which is one of the most popular problems in WWW.

The free access on tele-TASK site brings the complexity and difficulty to recognize students' learning interest. We separate all the users into three kinds: the students, the instructors and others such as crawlers, robots and irrelevant visitors. In order to mine the students' learning interest, it is necessary to filter out the last two kinds of usage data:

- instructors and administrators have the constant IP addresses, so we can easily remove the requests sent from these IP addresses;

- recognizing the sessions made by web robots has been discussed in Chapter 3.

The lectures in real format are transmitted to the clients via RTSP protocol (Real Time Streaming Protocol), while the mp4 and flash clips are delivered via HTTP protocol. Real Helix server records RTSP requests and supplies 6 logging styles. For example, the format for logging style 3 is shown in Figure 8.1, it records the start and end time of one request, and the client actions such as stop and pause. The requests on mp4 or flash clips are stored as HTTP server logs on server side, therefore, one media request could generate tens log entries depending on the media size and client actions as well.

Another problem is that one single streaming lecture could be involved by different courses, and this happens when some courses named the same title for different semesters have some chapters sharing the same content. This is also one of the conveniences that web-based teaching brings. Such flexibilities cause the variant URLs for the single same accessing object in database. To retrieve the unique accessing objects from the log entries in tele-TASK HTTP server logs, we make use the URL generating rules written in an XML file.

8.1.2 Modeling student learning profiles

The process of browsing multimedia lectures and other relative information can be seen as the online learning session. This learning session is depicted as: **one student views one learning object, if he finds that the knowledge from the content abstract is very familiar to him, he stops viewing this object and goes on finding other learning objects or just leaves web site; if he finds the learning object is his interesting target, he goes deeply on viewing this object;**

if he finds he can not understand some pieces of his learning object in one view, he repeats viewing these pieces.

On tele-TASK site, the learning objects presented in page views are classified in two kinds: with and without embedded multimedia lectures. The former are in Real, mp4 or flash format; and the latter are pages dedicated on the outline descriptions for courses, colloquium or other topic units, and they often link to the pages with embedded multimedia lectures.

The data cleaning and integrating preprocess helps to filter and translate the heterogeneous raw usage data into a set of browsing events. Each browsing event is represented as: **session**, **student**, **learning object**, **type**, **start time**, **duration** and **operation**.

Individual learning **operation** and **duration** can not be directly measured from usage logs. We now discuss how to compute the number of operations and the duration for different kinds of learning objects. Assuming that the right usage data have been separated for one learner from the helix server logs and cut into different learning sessions for this learner (the detailed techniques have been discussed in Chapter 3).

8.1.2.1 Computing the number of operations

For a real formatted lecture, as shown in the former section, the table of content field composes several sub headlines linked to the right positions within the multimedia lecture, which facilitates online learners to directly reach the right interesting piece, and the slide bar at the bottom of media lecture helps learners select or repeat some piece of the lectures as well. When an online student clicks the hyper links in the content table, jumps over some piece or repeats some piece, stops and resumes the lecture, the server will stop the current ongoing lecture and reload the right piece in the media file if it is not cached, and the new request on a piece of lecture will be written as a new log entry in the helix server logs.

Such usage data help us to assess the operations of one online learner on multimedia lectures. The operations of this learner can be estimated from the number of log entries on the same lecture within one learning session. However, there is an exception in computing learning operations on a lecture: when there exist several records on the same live broadcasting lecture within a learning session, which means that the learner could not jump over or repeat some piece. The reason of multi recordings was mainly the network overload or just the client's clicking stop button. In this case, we induce such records within a learning session to one record and the learning operation is concluded as 1.

8.1.2.2 Computing the duration

Now we explain how to compute the **duration** of one student spent on one learning object during one learning session:

- if the learning object is a real formatted lecture, the **duration** is computed by **timestamp** subtracted by **start_time**, both are recorded in real helix server log styled 3;

- if the learning object is a mp4 or flash formatted clip, which a single file request could generate several log entries in HTTP server logs, the successive log entries requesting the same clip are compressed to 1, and the **duration** is computed as the gap between the time stamps of the first and the last log entries; and

Table 8.1: *Example of Student Browsing Profile*

Session	Student	Learning Object	Type	Start Time	Duration	Operation
...	...	...	...	...	...	...
547	736	www12	0	31/May/2006:08:48:53	00:00:20	1
548	737	www12	1	19/Jun/2006:19:27:55	00:56:33	2
548	737	TI_08	1	19/Jun/2006:20:31:55	00:21:42	6
...	...	...	...	...	...	...

- if the learning object is a normal page view like course or series description, the **duration** is defined as the **gap** between the time stamp of this learning object and that of its next object recorded in the same learning session; if this learning object is the last one in a session, the **duration** is assigned the average duration of the learning objects from the same session; if this learning object is the only one in the session, its duration is set 0.

The **gap** between the time-stamps of every two successive requests on the same lecture depends on the operations that the learner makes:

1. if the operations are clicking hyper links, jumping over and repeating, the gap will last several seconds depending on the network overloading; and
2. if the operations are stopping and resuming, the gap will be decided by the two clicks from the client learner plus the transferring delay.

Within the same learning session, one learner with more operations displays more interest than that with few operations: the former finds clear and concrete accessing object in the lecture, while the later is possibly a fresh learner on this lecture. But we can not guarantee that some operations within few learning sessions with many operations were due to the reloading of the network or the server. The following Table 8.1 shows one piece of student browsing profiles on tele-TASK.

Limitations During computing the number of operations and duration, we assume during one learning session the student kept on sitting in front of his computer and concentrated on learning, though online learners require much more maturity, more self-motivation and self-discipline than those in traditional classrooms (Zhang et al. 2004).

The **duration** of viewing one media file computed by our method, in reality, estimates the time on transmitting the pieces of media file from the server side to the client side. This time duration is affected by the local cache and the network brand width. However, if the media is live broadcasted, this duration is nearly equal to that the client viewed this media lecture; and if the client makes plenty jumps on the sliding bar of the media player, this duration is much close to the actual viewing duration.

8.2 Answering the Six Questions

Before answering the six questions showing the multi facets of online learning interest listed in Chapter 7, it is necessary to define some parameters on learning interest from a group of users on a lecture l, which are shown in the followings:

- $N_{live}(l)$: number of accessing live streaming version of l;
- $N_{real}(l)$: number of accessing post edited version of l;
- $D_{real}(l)$: average time duration of viewing l;
- $O_{real}(l)$: average number of operations of viewing l;
- $N_{flash}(l)$: number of accessing flash version of l; and
- $N_{mp4}(l)$: number of accessing mp4 version of l;

8.2.1 Is there any difference between viewing the live broadcasting lectures and browsing lectures after they are recorded and edited?

The comparison between $N_{live}(l)$ and $N_{real}(l)$ tells the preference between viewing the live broadcasting and the edited lectures. The time duration of the live broadcasting of one lecture is decided by the length of lecture's recording, and it is usually between 60 minutes and 90 minutes. It can be predicted that $N_{live}(l)$ is always less than $N_{real}(l)$, but we can use the changes of $N_{real}(l)$ based on the day, week or month to find the detailed difference between $N_{live}(l)$ and $N_{real}(l)$.

8.2.2 Is there any preference on the different lectures in a course and preference on the different pieces of one lecture?

The preference on different lectures can be computed by comparing their $N_{live}(l)$, $N_{real}(l)$, $D_{real}(l)$ and $O_{real}(l)$. One lecture with bigger $N_{live}(l)$ and $N_{real}(l)$ shows much more acceptance than that with smaller $N_{live}(l)$ and $N_{real}(l)$. Further, one with bigger $D_{real}(l)$ and $O_{real}(l)$ tells that students would like to spend more efforts on it than that with smaller $D_{real}(l)$ and $O_{real}(l)$ if there is no big difference between two lectures on $N_{live}(l)$ and $N_{real}(l)$.

$D_{real}(l)$ can be computed as follows: $D_{real}(l) = \frac{\sum duration}{N_{live}(l)+N_{real}(l)}$, where ***duration*** is the time that one online learner spent on this lecture during one learning session and can be directly fetched from student browsing profiles. $O_{real}(l)$ can be computed as: $O_{real}(l) = \frac{\sum Operatioin}{N_{live}(l)+N_{real}(l)}$.

8.2.3 Is there any favor among real, mp4 and flash formats?

A lecture is recorded into a real formatted video, cut and further re-encoded into several mp4 and flash clips based on the topics in its TOC. From two sides we investigate the favor of students among real, mp4 and flash formats: one is from the single lecture, and the other is from the whole lecture set.

On the single lecture side, $N_{real}(l)$, $N_{mp4}(l)$ and $N_{flash}(l)$ show the access numbers on different formats. However, the accesses on different mp4 or flash clips from the same lecture within one learning session should be compressed 1 due to the relation $real : (mp4 + flash)$ is $1 : n$ in our case. For a lecture, we could further find the clips that draw more attentions, which depict the attracted topics more concretely in e-learning.

On the whole lecture set, the favor on real, mp4 and flash is measured generally by the summations of $N_{real}(l)$, $N_{mp4}(l)$ and $N_{flash}(l)$ of all lectures separately.

8.2.4 For one lecture, is the real video viewed together with its mp4 and flash clips?

This question is different from that raised in Section 8.2.3. This target is to verify if a student accesses the mp4 or flash version after he views the real version, and vice versa. From the set of learning sessions, the sessions having mixture formats videos could be separated from those with homogenous formats, and further divided into two sub sets: one is composed by the sessions on single lectures, and the other includes all the sessions on multi lectures. These proportions, especially sampled from different periods, give the hint if the real video is co-viewed with mp4 or flash clips.

8.2.5 Do the students view other lectures when they access one lecture?

Answering this question can be formulated as mining the frequent lecture sub set of the lecture set. Mining such relations is a typical example of mining association rules or frequent item sets (Agrawal and Imielinski 1993)(Han et al. 2000). The implicit relations among different online lectures could help teachers to know if they need to combine some lectures or add some content from other courses.

We simplify a learning session s on some lectures as: $s = \{l_1...l_k\}$, where l_i is one lecture or lecture piece regardless its format. Transformed from the set of student browsing profiles, the set including all the learning sessions are named as P. From P, we try to mine the relations each of which is formed as $r = \{l'_1...l'_t\} : Supp_r$, where $l'_i \in L$ and $Supp_r$ is the number of sessions that viewed all the lectures in r. The methods to mine association rules or frequent item sets have been widely discussed. We used the mining method referred in Chapter 4, which integrates all the learning sessions into a highly compressed extended prefix-tree structure called frequent pattern tree stored in memory, and the complete frequent item sets can be mined from this tree structure without candidate generation.

8.2.6 For the same named course supplied for different years, is there any change on the students' interest?

We discover the changes of students' learning interest from their usage data in web video-based learning environment. Due to the effects on each other of the changes in web students and web lectures, we seek a method that integrates the changes from both sides to measure the changes of learning interest.

In Section 4.4, we have explained x-tracking the changes of web usage patterns. The changes are measured from internal and external, local and global sides. The usage patterns are in the forms of frequently co-accessed page sets, page sequences and tree structures, which describe the navigation on an information portal site whose content is mainly in textual format and relatively stable. In an e-learning site, in which the most teaching materials are videos and updated frequently, the learning interest and the changes of the learning interest have to be measured in another way. However, detecting the changes of learning interest is a special case of tracking the changes among usage patterns in Section 4.4, and the logic behind both are the same.

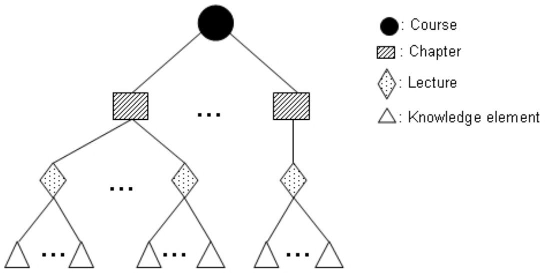
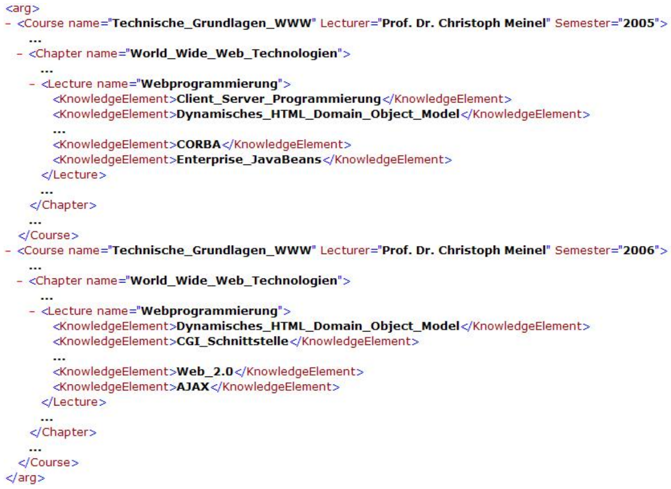


Figure 8.2: *Concept hierarchy of one couse*



```
<arg>
- <Course name="Technische_Grundlagen_WWW" Lecturer="Prof. Dr. Christoph Meinel" Semester="2005">
    ...
  - <Chapter name="World_Wide_Web_Technologien">
      ...
    - <Lecture name="Webprogrammierung">
        <KnowledgeElement>Client_Server_Programmierung</KnowledgeElement>
        <KnowledgeElement>Dynamisches_HTML_Domain_Object_Model</KnowledgeElement>
        ...
        <KnowledgeElement>CORBA</KnowledgeElement>
        <KnowledgeElement>Enterprise_JavaBeans</KnowledgeElement>
      </Lecture>
      ...
    </Chapter>
    ...
  </Course>
- <Course name="Technische_Grundlagen_WWW" Lecturer="Prof. Dr. Christoph Meinel" Semester="2006">
    ...
  - <Chapter name="World_Wide_Web_Technologien">
      ...
    - <Lecture name="Webprogrammierung">
        <KnowledgeElement>Dynamisches_HTML_Domain_Object_Model</KnowledgeElement>
        <KnowledgeElement>CGI_Schnittstelle</KnowledgeElement>
        ...
        <KnowledgeElement>Web_2.0</KnowledgeElement>
        <KnowledgeElement>AJAX</KnowledgeElement>
      </Lecture>
      ...
    </Chapter>
    ...
  </Course>
</arg>
```

Figure 8.3: *Organization of Knowledge Elements, Lectures, Chapters and Courses in XML*

8.2.6.1 Statements on types of changes

One web course has its clear conceptual intensions and extensions, and it is characterized by a set of knowledge elements. These knowledge elements will be delivered to the web students in the form of web-streaming lectures in a suitable sequence. Further, the lectures belonging to the same course are organized in several non-intersected sub sets: *units* or *chapters*. The knowledge elements, lectures, units and course are organized in a tree hierarchy structure, and we simplify this tree structure into four layers shown in Figure 8.2: *course layer, unit (chapter) layer, lecture layer* and *knowledge element layer*.

The Figure 8.3 shows a piece of XML codes that describes part of the organizations of two same titled lectures "Web Programming" in courses "Basic Technic of WWW" from different semesters.

Suppose we have a web course C and it has m different lectures: $C = \{l_1, l_2...l_m\}$. The knowledge set K is the set comprised by all the knowledge elements in C: $K = \{k_1, k_2...k_n\}$. The knowledge elements dedicated to lecture $l_i(l_i \in C)$ form a sub knowledge set of K: $l_i = \{k_{i,1}, k_{i,2}...k_{i,t}\}$, where $k_{i,j} \in K(1 \leq j \leq t)$. We use U to name the set of units (chapters) that course C includes: $U = \{u_1, u_2...u_p\}$, and $u_i(1 \leq i \leq p)$ is one chapter that includes some closely related lectures from C and formed as a sub set of C: $u_i = \{l_{i,1}, l_{i,2}...l_{i,q}\}$, where $l_{i,j} \in C(1 \leq j \leq q)$.

Similar to Section 4.4.1, we define the following *basic edit operations* for computing the changes between two courses(units, lectures):

- $Insert(x, y)$: *insert an element* x *as a leaf element of* y*;*
- $Delete(x, y)$: *delete a leaf element* x *from element* y*; and*
- $Update(x, b)$: *update an element* x *in course* C *with the new label* b *resulting that* C *is identical to course* C'*, which means that* C' *is identical to* C *except that the label of* x *is* b*.*

Based on the basic edit operations, a *structural edit script* is a sequence of basic edit operations that convert one structure to another. Here the *structure* has different levels: course level, unit level and lecture level.

22. DEFINITION (STRUCTURAL DISTANCE BETWEEN TWO LECTURES). *Let* l_1 *and* l_2 *be two lectures,* structural distance SD_{l_1,l_2} *is the number of basic edit operations in the structural edit script that can change* l_1 *to* l_2*.*

Similarly, we use SD_{C_1,C_2} for the *structural distance* between two courses, and SD_{u_1,u_2} for that between two units.

23. DEFINITION (USAGE ON ONE LECTURE). *Given a web-streaming lecture* l^f *in format* f *(real, mp4 or flash), we use* $UG_{l^f} = (N_{l^f}, D_{l^f}, O_{l^f})$ *to name the* $usage$ *from a set of web students on* l^f *during one time period.*

The explanations of the three parameters N_{l^f}, D_{l^f} and O_{l^f} are shown in Table 8.2.

.

Table 8.2: *Explanations of* N_{l^f}*,* D_{l^f} *and* O_{l^f}

Parameter	Meaning
N_{l^f}	Number of accessing l^f
D_{l^f}	Average time duration of viewing l^f
O_{l^f}	Average Number of Operations of viewing l^f

24. DEFINITION (USAGE SCORE ON ONE LECTURE). *Let* $UG_{l^f} = (N_{l^f}, D_{l^f}, O_{l^f})$ *be the usage of one lecture* l^f *during one period time, the* usage score *of* UG_{l^f} *is defined as:*

$$US_{l^f} = \alpha \times N'_{l^f} + \beta \times D'_{l^f} + \gamma \times O'_{l^f} + \delta, \tag{8.1}$$

where $\alpha + \beta + \gamma + \delta = 1$.

Before this computing, the original values of three parameters have to be normalized. Similarly, the usage and usage score (or weighted usage used by TASK-Moniminer in Chapter 6) on a course or a unit are named as UG_C and UG_u, US_C and US_u. Based on the definitions of usages on different levels, we can investigate the $changes$ of usage from one period to another period.

25. DEFINITION (CHANGES OF USAGE ON ONE LECTURE). *Let* UG_{l^f} *and* $UG'_{l'^f}$ *be two usages from two periods, changes of usage* $CH(UG_{l^f}, UG'_{l'^f})$ *are the* $increasing$ *or* $decreasing$ *of the usage parameters and usage score in one time period compared to another time period.*

8.2.6.2 Measuring changes of usages

Changes of usage $CH(UG_{l^f}, UG'_{l'^f})$ from l^f to l'^f are defined as a four-items set including the $increasing$ or $decreasing$ on N, D, O and US. Changes of *Usage score* reveal the $decreasing$ or $increasing$ of the general usage in the form of ranking, but do not reflect the *changes* on different parameters. The same changes on *usage score* may be due to different changes of four parameters. For simplicity, we use $max(|\bullet|)$ to name the $max(|CH_N|, |CH_D|, |CH_O|)$.

8.2.1. THEOREM. *Let CH_N, CH_D and CH_O be the changes of N, D and O from one student set on l'^f to another student set on l'^f, the change CH_{US} on usage score is:* $0 \leq |CH_{US}| \leq max(|\bullet|)$.

1. PROOF. For simplicity, we use N_{1^f}, D_{1^f}, O_{1^f}, US_{1^f} and $N_{1'^f}$, $D_{1'^f}$, $O_{1'^f}$, $US_{1'^f}$ to name the usages in two time periods. Based on the computations for different changes:

$$|CH_{US}| = |\frac{(\alpha \times N_{1^f} + \beta \times D_{1^f} + \gamma \times O_{1^f} + \delta) - (\alpha \times N_{1'^f} + \beta \times D_{1'^f} + \gamma \times O_{1'^f} + \delta)}{\alpha \times N_{1^f} + \beta \times D_{1^f} + \gamma \times O_{1^f} + \delta}| \tag{8.2}$$

$$= |\frac{\alpha \times CH_N \times N_{1^f} + \beta \times CH_D \times D_{1^f} + \gamma \times CH_O \times O_{1^f}}{\alpha \times N_{1^f} + \beta \times D_{1^f} + \gamma \times O_{1^f} + \delta}| \tag{8.3}$$

$$\leq \frac{max(|\bullet|) \times (\alpha \times N_{1^f} + \beta \times D_{1^f} + \gamma \times O_{1^f})}{\alpha \times N_{1^f} + \beta \times D_{1^f} + \gamma \times O_{1^f} + \delta} \tag{8.4}$$

$$\leq max(|\bullet|). \tag{8.5}$$

8.2.6.3 Similarity comparison between two learning objects

Based on the *structural distance* from l_1 to l_2, we compute the *similarity measure* between l_1 and l_2 as:

$$SM(l_1, l_2) = \frac{max(SD(\emptyset, l_1), SD(\emptyset, l_2)) - SD(l_1, l_2)}{max(SD(\emptyset, l_1), SD(\emptyset, l_2))}, \tag{8.6}$$

where $SD(\emptyset, l_i)$ $(i \in \{1, 2\})$ is the *structure distance* of building the entire l_i from an empty set based on the basic edit operations.

8.2.2. THEOREM. *Let $l_1 = \{k_{1,1}, k_{1,2}...k_{1,n}\}$ and $l_2 = \{k_{2,1}, k_{2,2}...k_{2,m}\}$ $(k_{i,j} \in K)$ be two lectures with their knowledge elements, the* similarity measure *between l_1 and l_2 is:* $0 \leq SM(l_1, l_2) \leq 1$.

2. PROOF. We use $|l_1|$ and $|l_2|$ to name the numbers of knowledge elements separately in l_1 and l_2; and $l_1 \cap l_2$ to name the intersection of l_1 and l_2, which includes all the same knowledge elements between l_1 and l_2. It is obvious that $0 \leq |l_1 \cap l_2| \leq min(|l_1|, |l_2|)$. So $max(|l_1|, |l_2|) - |l_1 \cap l_2|$ is the number of knowledge elements in l_1 that need to be updated or inserted so that l_1 can be equal to l_2, and this number is right equal to $SD(l_1, l_2)$. This means that $0 \leq SD(l_1, l_2) \leq max(|l_1|, |l_2|)$.

The algorithm to compute *structural distance* between two trees can be referenced in (Chawathe 1999). But our aim is to compute the similarity on different levels for two course structures, so we integrate these three computations in one algorithm. Another difference is that we assign identical costs to all the basic operations, due to our concentration on the *structural* difference. Though it is easily proven from the definitions of structural similarity that computing similarity between two trees in one level is symmetric, the similarity comparisons on all the levels are asymmetric. Computing the changes of one unit (lecture) is the process to find the maximal similarity of one unit (lecture) compared to all the units (lectures) of the other courses.

Algorithm 3 *Structural Similarity Algorithm*

```
Initialize the set of structural similarities $S$ as empty
Compute the sets of chapters and lectures sub tree structures $U_A$, $U_B$, $L_A$ and $L_B$ from A and B
Compute $Similarity(A, B)$ and add it in $S$
for all $u_i$ in $U_A$ do
    for all $u_j$ in $U_B$ do
        Compute $Similarity(u_i, u_j)$
    end for
    Compute $Max(Similarity(u_i, u_j))$ and add it in $S$
end for
for all $l_i$ in $L_A$ do
    for all $l_j$ in $L_B$ do
        Compute $Similarity(l_i, l_j)$
    end for
    Compute $Max(Similarity(l_i, l_j))$ and add it in $S$
end for
Output $S$
```

8.2.6.4 Measuring changes of learning interest

The *changes of usage* play importance to compute the *changes on interest*, but they are not equal to the changes of interest due to the effect of changes of learning objects. Our assumption is: **the changes of usage on the same or the similar learning objects are much useful than those on totally different learning objects**. A teachers could not make any decisions or improvements on his teaching course if he found that web students spend much more efforts on ″Food Engineering″ than on ″TCP/IP″, because there is nearly no relations and similarities between ″Food Engineering″ and ″TCP/IP″.

Given two lectures l_1 and l_2, and also the usages on them UG_{S_1,l_1} and UG_{S_2,l_2}, the *changes of learning interest* $CHI(S_1, l_1, S_2, l_2)$ from l_1 to l_2 are computed as:

$$CHI(S_1, l_1, S_2, l_2) = SM(l_1, l_2) \times CH(UG_{S_1,l_1}, UG_{S_2,l_2}). \tag{8.7}$$

We can further compute the changes of learning interest on *unit level* and *course level* similarly based on the computation on *lecture level*. The changes on *lecture* level can not reveal the changes on *unit* or *course* level, on the other hand, the stable learning interest on course level may hide the big vibrations of the learning interest on lecture level.

Given two sets of usages on the same titled courses serving in different semesters, the problem of mining the changes of learning interest is to find the learning objects from different levels (course, chapter, lecture) on which the learning interest is changed beyond the predefined thresholds. For example, we set $\tau_u = 0.05$ for US and $\tau_n = 0.08$ for N_{lf}, D_{lf} and O_{lf} to mine the lectures on which the change on US is out of [-5%,5%] and change on ***any*** of N_{lf}, D_{lf} and O_{lf} is out of [-8%,8%].

8.3 Summary for This Chapter

In this chapter, we have firstly discussed the data preparation in discovering learning interest in web video-based systems and then presented the methodologies on answering the six questions. The first four questions were solved by statistics, from which the sampling parameters display the preferences on different formats, on different topics and different learning sequences. We used association rule to discover the topics that were usually learned together, which could help optimizing the content of lectures. To discover the changes of the learning interest on one lecture/course in one year compared with the other year, we integrated the content difference with usage divergence, in which the former difference leverages the latter divergence. The methodology, integrating concept distance with usage and structural diversity, was used as well in tracking the changes of web usage patterns in Section 4.4 and will be further used in finding and recommending high reputation articles in a social site in Part III.

As the limitations discussed in previous Section 7.3, there are some defects to quantitatively evaluate any distance learning environment. In unauthorized web video-based e-learning systems, evaluating becomes more controversial due to the majority of users are not students and the lack of collecting usage actions on videos. Such problems will be further detailed in the next Chapter 9: results discussion on TASK-Moniminer.

Chapter 9

Results Discussion on TASK-Moniminer

The mining methods used in TASK-Moniminer are implemented on our web-based learning environment: tele-TASK, and the learning data includes: HTTP access logs and RTSP access logs. The usage data on web pages, mp4 and flash files is written on the HTTP server, while the requests on real formatted medias are recorded on the RTSP helix server. Moreover, the logs are distributed on different servers physically. So in data preparation, the loglines in these logs are integrated and sorted on their time stamps. The work on data preparation has been discussed in Chapter 3 and Chapter 8. In Chapter 6, TASK-Moniminer has been introduced, which serves as a search engine for teachers to query the learning interst from the online students on their courses. Such intuitive interface is developed and implemented based on the observations, and the mined results that will be discussed in this chapter. Moreover, this chapter gives other important findings.

Chapter Organization The learning interest includes the general statistics on the whole site in Section 9.1 and then we will answer the 6 questions listed in Chapter 8: the first 5 in Section 9.2 and the last in Section 9.3. The summary of this chapter is given in Section 9.4.

9.1 General Statistics

Figure 9.1 shows the distribution of monthly hits on different formate files on tele-TASK from 01.04.2007 to 31.03.2009. The hit number of web pages is nearly 4 times of those on real media, mp4 and flash files, which tells that a user usually access one real media lecture or one mp4 or flash clip after he viewed 4 web pages. Considering the back tracking in a session, this corresponds to the basic tree structure of tele-task site: home page, course page and lecture page. The summits along the curves display the intensive blowouts of visits due to some events on the site. For example:

1. the visit blows out on web pages in July 2007 is due to the massive launch of feed mechanism on tele-task, which attracted thousands of feed reader;

2. the reason for the drastic climbing of hits on web pages from November 2008 is the publish of the new version of tele-task site;

3. the sudden visits summit on MP4 clips in Feburary 2009 is because of the cooperation with iTunes, who fetched almost all the mp4 clips from tele-task.

Compared with the drastic changes of hits on web pages and mp4 clips, the visits on real media lectures are relative stable. To further discover the learning interest on e-lectures, the negative effects from the above events have to be removed because the big part of visitors under these

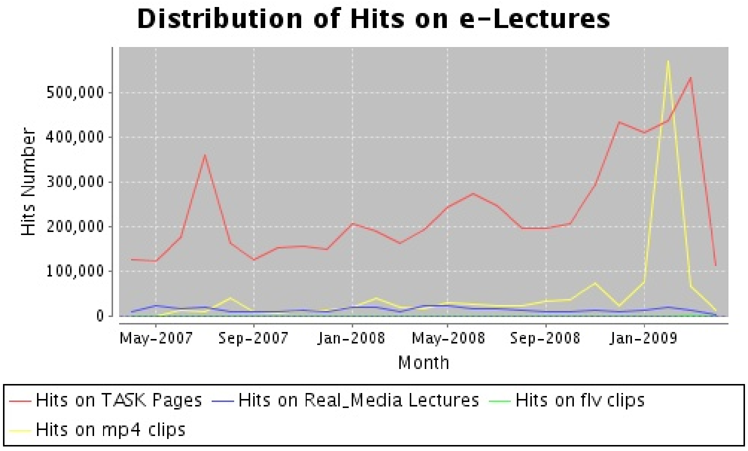


Figure 9.1: *Hits on Page, Real Media, MP4 and FLV*

situations are noises. Moreover, over 90% HTTP requests recorded on the server side are asked by robots, for instance, the frequent requests for the same resources, the extreme short intervals between two successive requests, hundreds requests in a session, or several hours a session lasted. Such massive noises affect discovering the learning interest from normal users, especially the students. The relative techniques used for removing unrelated data have been discussed in the previous Chapter 3 and Chapter 7.

TASK-Moniminer tells the geographic distribution of IP adresses of visitors of every queried lecture: HPI employee (E), HPI students (S) and Outer HPI (O). The IP distribution on employee and student are different on different lectures, but on average, 92% visitors are from out of HPI. In addition, the surveys from other researchers showed as well that more users are accessing the Web from home for educational purposes.

Data Set 3 Two kinds access logs are taken from one semester 01.04.2006∼31.07.2006 (LOG_I) and another semester 01.04.2005∼31.07.2005(LOG_{II}). Besides, we gathered the compositions of courses WWW for these two semesters. WWW from 2006 includes 26 lectures, while WWW from 2005 includes 31 lectures. Before 2007, there was no mp4 or flash clips provided on tele-TASK.

Data Set 4 We choose the usage data from the time span between 01.12.2008 and 31.01.2009, because the content updating is relatively regular and the accesses from abusive users are relatively few. We collected the usage on 2217 real media lectures, 2720 mp4 clips and 1376 flash clips, from which 13217 sessions from student were rebuilt. Intuitively, we use a bit for every lecture type to show if this kind of lecture was accessed in a session, and the usage interaction among the lecture types is represented by 3-bits. Over these sessions, we check the general usage among theses 3 kinds of e-lectures. Table 9.1 displays the distribution of sessions on different types of e-lectures. Every session accessed at least one type of e-lectures, the sessions having not accessed e-lectures are not included. The usage interaction between real media (R), mp4 clips (M) and flash clips (F) cut the set of sessions into 7 divisions. For example, $RMF(010) = 8303$ means that the number of session accessing ONLY mp4 clips is 8303; $RMF(110) = 316$ tells that 316 sessions viewed real

Table 9.1: *General interaction usage on 3 kinds of e-lectures*

R	M	F	Number of Sessions
0	0	1	187
0	1	0	8303
0	1	1	55
1	0	0	4289
1	0	1	52
1	1	0	316
1	1	1	15

medias with mp4 clips, but without flash clips; and $RMF(000) = 0$ is not included.

As discussed in the previous chapters, we investigate the learning interest on every e-lecture from hit number, viewing time and operations. The viewing time a user spent on a real media lecture is measured in seconds, and the number of operations shows his actions such as ”pause, stop, and sliding” on the lecture. The difference between **Data Set 3** and **Data Set 4** is: there were only usage on real media lectures in Data Set 3 because no mp4 or flash clips were provided on tele-task at that time; while Data Set 4 includes the usage on real lectures, mp4 and flash clips.

Recalling that each real media lecture has several mp4 and flash clips based on the content structure of the lecture, the numbers of hits and sessions on different formats are not enough to reveal the learning interest. Especially finding the difference of learning interest on different topics or the same lecture in different years is more useful for the teachers.

9.2 Results on Answering 5 Questions

In this section, we give the answers for the questions raised in Chapter 8. We use different data sets to answer different questions: Question 1 and Question 6 use Data Set 3; while Question 2 $\sim$ 5 are answered based on the Data Set 4.

Answering Q1: Is there any difference between viewing the live broadcasting lectures and browsing lectures after they are recorded and edited? To compute the average access duration of students on lectures, the various length of lectures have to to considered. The maximums, minimums and average of different usages on WWW lectures is shown in Table 9.2. Spending 25 minutes on a only 30 minutes long lecture shows more learning interest than costing 35 minutes on a 100 minutes long lecture. In our experiments, we replaced *average access duration* with *(Average Access Duration)/(Length of Lecture)*.

Table 9.2: *Maximum, Minimum and Average Usage on WWW Lectures*

	$N_{live}(l)$	$N_{real}(l)$	$D_{real}(l)$	$O_{real}(l)$
Maximum	11 (IPv4)	767 (Erst Einfuehrung)	00:30:12 (Sicherheit Internet 2)	8.11 (Sicherheit Internet 2)
Minimum	0 (TCP UDP)	34 (HTML)	00:00:19 (Erst Einfuehrung)	2.68 (Erst Einfuehrung)
Average	7	164	00:08:37	4.2

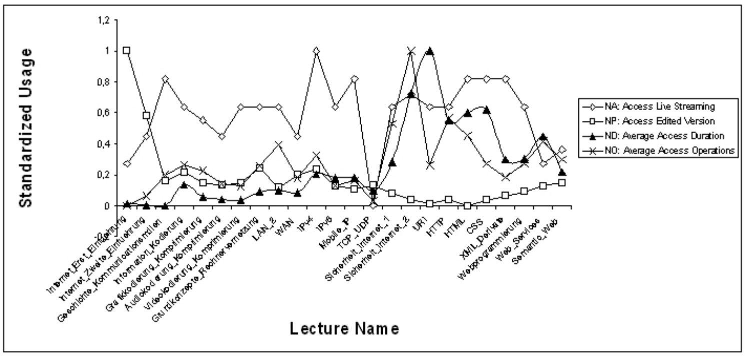


Figure 9.2: *Usage on WWW from different aspects*

The Figure 9.2 shows the statistics of usages on different lectures of WWW in 2006. We found that very few users accessed live broadcasting lectures compared with the number of viewing real lectures after being recorded and published. Though supplying more learning channels and formats is the striving target for students, motivation and self discipline in attending on-site virtual lecture or classroom is not only a technical but more a pedagogic question. We have been complained from some lecturers that fewer students are present in classroom after they introduce tele-task, but other lecturers welcome the facility that tele-task brings.

The number of accessing live broadcasting lectures is much less important than that of accessing the edited version due to the short active period of live streaming. So we use three variables to evaluate the learning interest: the number of accessing edited real lectures, time spending on one lecture and actions during viewing one lecture, which were introduced in Chapter 8.

Answering Q2: Is there any preference on the different lectures in a course and preference on the different pieces of one lecture? From our experiments, the first two lectures have the two biggest usage scores. Because these two lectures are showed at the beginning of the course web page and exist longer time than others, and the first lecture displays automatically when the course page is opened, they attract much more accesses than others.

From Data Set 3, the average time spending on a lecture from students is 487 seconds (~8 mins) while the average length of a recorded lecture is 80 minutes. This tells that students viewed only part of the one whole lecture. This gives us the idea and convience to cut every recorded lecture into mp4 or flahs clips based on the content structure of one lecture.

After supplying mp4 and flash clips on tele-task, we reinvestigate the average time spending on a real media lecture. From Data Set 4, we found 496 seconds the students spent on viewing one real media lecture, and there is almost no change compared with that from Data Set 3. Considering the stable number on viewing real media lectures from Figure 9.1, this shows that the user group on viewing real media lectures are stable, and not affected by the services on mp4 and flash clips. The latter two medias attract new users.

Answering Q3: Is there any favor among real, mp4 and flash formats? From Table 9.1, we notice the number of sessions accessing mp4 clips is nearly double that of real media lectures, and very few sessions visited flash clips. The reason for this big difference between mp4 and flash clips on attracting users is probally this: mp4 has a higher market share on Internet streaming than flash. The former allows the users to dowload on their local machines for the further achiving and viewing, while the latter is not so easy for downloading.

Answering Q4: For one lecture, is the real video viewed together with its mp4 and flash clips? The interactions among these 3 formats media display the interesting observations from Table 9.1: only 55 from 8303 (0.6%) mp4 sessions viewed flash clips, while 29.4% (55/187) flash sessions browed mp4 clips; 27.8% (52/187) flash sessions accessed real media lectures, while 1.2% (52/4289) real session clicked flash clips; 3.8% (316/8303) mp4 sessions viewed real media, and 7.3% (316/4289) real sessions accessed mp4 clips. This tells the few overlap between users on real media and mp4 clips, while big overlap between users on flash and mp4, and between flash and real as well.

Answering Q5: Do the students view other lectures when they access one lecture? On Data Set 3, we used the methods for mining frequent item sets to find if there exist some relations between the lectures during the same learning sessions. We set the threshold of the support number 1% to mine the sub sets of frequent lectures. We find that the first two lectures of WWW were always viewed together, this happened as well in other courses. We find that there is no relation among lectures belonging to different courses, and the low threshold and few mined relations suggest us that most of the online learners have clear and singular learning object during one learning session.

On the other hand, on Data Set 4, we still found that few students accessed two or more real media lectures during one session, this repeats the conclusion above that students usually did not access other real lectures after viewing one real media lecture. The situation on mp4 is bit different: the average number of mp4 clips per mp4 session is 3.5, which means that students usually clicked more than 3 mp4 clips in one learning session. Though free downloading mp4 clips could distort the actual learning process on viewing mp4 clips, the multiple downloads per visit definitely reveal the learning interest on multiple targets. Further, we used frequent item sets mining method to discover the co-accessed mp4 clips from 942 mp4 sessions which viewed more than 1 mp4 clips, and found 66.8% (630/942) sessions have the mp4 clips from the same lecture. This means that most students concentrated on the related clips during their online learing processes.

9.3 Results on Detecting Difference of Learning Interest on Similar Courses

The work on detecting the difference of learning interest on similar couses is implemented on Data Set 3.

Figure 9.3 shows the similarity comparison between the content and organizations of two same titled lectures ”WWW” from summer semesters 2005 and 2006. From this figure, it shows clearly that the similarity on the course level is not linearly decided by those on the unit or lecture levels, and also the similarity on the chapter is not linearly decided by those on the lecture level. This further proves the necessity to compute the changes of learning interest on different levels. On the course level, the similarity of course WWW between 2005 and 2006 is 0.75. It is interesting that the similarity of first chapter between two years is 0.33, which is much lower than those of

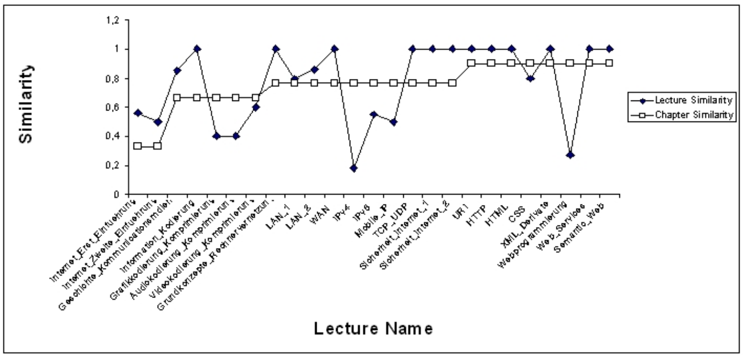


Figure 9.3: *WWW course changes on lecture and chapter level*

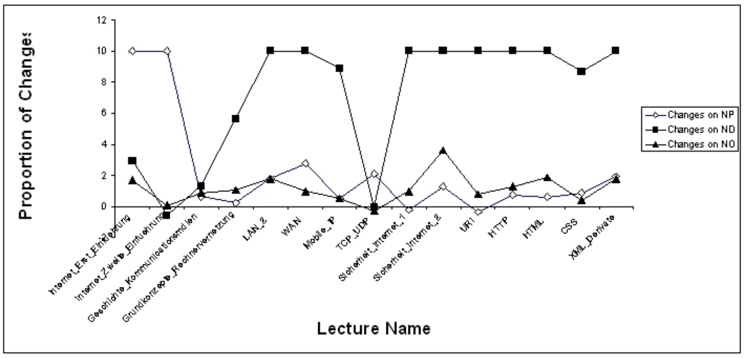


Figure 9.4: *Usage Changes on WWW from 2005 to 2006*

the two lectures in this chapter: $0.33 < min(0.5, 0.56)$. This is because there was one more lecture in the first chapter in 2006 than in 2005. The similarity of chapter 3 is 0.9, but this hides the great changes in the lecture "Web Programming", of which the similarity is only 0.27.

The changes of usage on WWW from 2005 to 2006 is shown in Figure 9.4. Due to the popular acceptance by students on e-learning and efficient arrangement of teachers on tele-teaching materials, the lectures in 2006 attract much more learning interest than those in 2005, no matter from any aspects of usage. The accessing number on the edited lectures raised explosively, and web students spent much more time than before, and their interactivities with the lectures become more active as well.

It does not show the effect of the changes on the lectures in Figure 9.4. The changes of learning interest integrating both changes on usage and lectures are shown Figure 9.5. We can see the difference between these two figures. Though explosively increasing of learning interest on most of lectures, the decreasing of that on "TCP/UDP", "URI" and "Sicherheit Internet 1" helps the teachers to think about if they know correctly the students' mastery levels.

From our investigations on web learning, we draw that the students have already been familiar to the basic knowledge such as "HTML" and "URI" before they choose this course from 2005 to 2006. We can adjust this course in the future: delete or compress the lectures on "HTTP", "HTML" and "URI", while enlarge the lectures about "Web Services", "Semantic Web" or other knowledge on the frontier of WWW. Besides of these, we also draw that the average learning duration on a

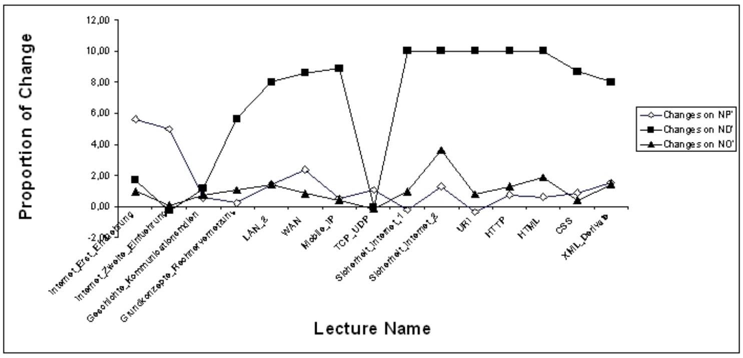


Figure 9.5: *Weighted Usage Changes on WWW from 2005 to 2006*

web lecture is nearly 8 minutes, while the average length of one web lecture is about 80 minutes. This is one of the theoritical and pratical foundations for our implementation of segmenting the lecture into small pieces of mp4 and flash clips.

9.4 Summary for This Chapter

In this chapter, we have discussed the results on discovering learning interest on streaming lectures from tele-TASK. We found that time spent on real media lecture does not change bit after introducing mp4 and flash clips: about 8 minutes per learning session. The introductions of mp4 and flash clips attract new group of users, but the former occupies most of the new users beause of the big market on internet streaming market. We especially illustrate the results on detecting the changes of learning interest on the same courses from different semesters. The detecting method combines the changes on the usage and the content.

Yet, multiple sources (log files, demographic files, academic performance files) are required to fully assess the online participation of students. Only a repository based on multiple data files from various sources can answer the broad range of questions teachers are likely to ask. "Mixing" these multiple source is a tough challeng covering the techinical and administive sides. For example, we can easily answer "how many minutes did students spend on an online lecture" through a straightforward and simple analysis of the server log file. However, the question "what other materials do the students use when viewing online lectures" can be answered only through an analysis of data from other sources such as questionnaires and direct interviews.

Part III

Recommending High Reputation Articles in a Social Site

Chapter 10

Re-Blog: A Mechanism to Recommend High Reputation Articles on IT-Gipfelblog

The type of content available on the web suffers a great overturn in the recent years. From the early 1990s onwards, a relative small amount of publishers dominated the creation of web content, while the majority of web users were only the consumers of content. From the early 2000s, user-generated content has become increasingly popular on the web: more and more users participate in content creation, rather than just consumption. Such user-generated content (or social media) is scattered in blogs and online forums, bookmarking sites, medias sharing communities, as well as social networking platforms such as Xing, LinkDB and Twitter, which emphasize on the relationships among the users of the community.

Social site's growing underscores a transformation in the web that's as fundamental as its birth: rather than simply searching for and passively information displaying, users are collaboratively creating, evaluating, and distributing the content and information. Social media sites share the four characteristics:

1. users create or contribute content in various media types;

2. users annotate content with tags;

3. users evaluate content, either actively by voting or passively by using it; and

4. users create social networks by designating other users with similar interest as contacts or friends.

However, the explosively increasing information in communities brings the wide range of the quality of user-generated content. This requires an effective mechanism to filter spam content and find high quality content, and the moderator of a community needs this mechanism as well to collect, recommend social content, and guide the public discussions.

Chapter Organization We firstly describe IT-Gipfelblog in Section 10.1 and then give the related work on content filtering in social media sites in Section 10.2. Finally we give our recommendation mechanism in Section 10.3.

10.1 IT-Gipfelblog

IT-Gipfelblog is a topic-centered social community, in which the users get connected by discussions on the heterogeneous topics. Though named as a ”blog”, it functions more as the public forum on the topics of IT technologies and policies in Germany.

10.1.1 Content of IT-Gipfelblog

IT-Gipfelblog covers the issues in IT area. The issues are classified into 4 categories and further divided into 9 sub categories.

A registered user can post his articles, comment the articles from others, mark the tags, and evaluate the articles by voting. Thus, overall, each user has a fourfold role: publisher, commenter, annotator and evaluator. Before submitting some articles, a user firstly should be a registered member getting the permission from the moderator to join the community, and getting the rights to publish and edit his own articles, to comment, vote and rate the articles from others. A user could go to his own profile space, to write an article and further upload it to the right interest group or sub column (could be some personal bias). The users may be classified into different groups having different authorities.

The central elements of IT-gipfelblog are threads. Like the definition by wiki [1], a thread is a collection of **articles**, usually displayed by default from oldest to latest, although the option for a threaded view (a tree-like view applying logical reply structure before chronological order) can be available. We use the term **article** while not the **post** as in wiki definition to avoid the confuse between post and comment used in the next chapters. A thread is usually a relatively conceptual independent set of articles, and could be closed and succeeded by another sibling thread if the article number is over a threshold.

Dedicated on IT-Gipfelblog, the articles are classified into two sets based on their dependency relations: post set and comment set. A post is usually the first article in a thread, and is the beginner advocating one discussion. A successful post can attract several comments showing their positive, neutral or negative points compared to those in the post. In a social site, sometimes the spam comments intrude the discussion, and some comments deviate the topics listed in the beginner post. One comment within one thread could draw its own comments as well. One post and its comments are often displayed in a tree structure.

Based on the content format, articles are divided into three sub sets: video related, text related and hybrid of video and text. An article having only videos expresses the author's tastes and views in the videos, and the article in pure text sometimes could be a hyperlink to other resource. The boundary between the video related and the hybrid of video and text is detected by several methods, which will be discussed in Chapter 11.

In IT-Gipfelblog, each post is marked tags by the registered users, and the tags are the condensed text representing the content of the post. Besides, it receives an average vote submitted by the users between 1 and 5 companying with the number of voters, where a higher vote shows the higher approval from the voters. A comment gets no tags, but thumbs up or down from the users.

10.1.2 Users of IT-Gipfelblog

Till the end of 2008, there are 500 registered users that have the authorities to submit posts or comments, rate posts, or give thumbs up or down to the comments. However, most of the visitors on IT-Gipfelbog are anonymous, which contribute their clicks on posts and comments. The click usage is the implicit feedback information compared to the votes for the posts, or thumbs up and down for the comments. The usage feedback is the possible metrics to evaluate the quality of articles, which could be the complementary or supplementary to the content-related evaluations. The

[1] http://en.wikipedia.org/wiki/Internet_forum

purpose of Re-blog is to investigate the relationship between content-related and usage-related metrics, to find a framework combining both sides for recommending high quality posts.

Compared with other entertainment online communities, one of the tasks of IT-Gipfelblog is to attract more young people to join the discussions on the issues about IT technology in Germany. On the other hand, the relative stable user group avoids the spam posts and abusive comments. This reduces the risk of the negative sentiment. And we could concentrate on the metrics from the technology side to evaluate the quality of articles.

10.2 Related Works

Online recommendation has attracted many discussions in personalization, internet shopping and social communities. Currently, the recommendation standards can be generally classified into four categories:

1. Content based: in communities like Youtube, the semantically related clips (in the same categories or having the same keywords or tags) are listed aside with the current clip;

2. Session based: this is much popular in online shops such as Amazon, in which a customer is supplied the items frequently co-browsed or co-bought with the item he visited;

3. Vote based: the top sold or voted items are usually presented at every page for online shops and tube communities; and

4. Relation based: in online communities like facebook and linkdb, the members having the same interest or affiliations are linked together.

However, the wide range of content quality in a social site raises the requirement of content evaluation before post recommendation. On the other hand, the rich interactions between users and content supply the possibilities to find new evaluation and recommendation methods.

10.3 Implementation of Re-blog

Here we give a mechanism on finding and recommending high reputation articles in a social site: Re-Blog. Re-Blog works in the followings:

```
Step 1: separating the high reputation articles
        into global and local reputation groups;
Step 2: clustering the content related articles in each group;
Step 3: selecting the representative articles from each cluster;
and
Step 4: the representatives are supplied
        based on the rank of their reputation.
```

We made our experiment on IT-Gipfelblog (http://it-gipfelblog.hpi-web.de), which is a weblog in German discussing the topics on information and communication technologies. Our framework is given based on the following observations during our experiments:

1. no strong correlation is observed between the numbers of hits, the number of voters, the value of votes and the number of comments an article received;
2. the number of hits and the value of votes play strongly on the reputation evaluation, while the comments have few impact on this assessment;
3. the reputation an article received can be classified into global and local categories, which means that some articles are highly accepted by a big population, while some are highly reputed in a small population;
4. the reputation evaluation from usage side can be proven by some aspects of the quality of the content an article has.

Chapter 11

Tags, Keywords, Contexts and Users in a Social Site

The online communities originate from traditional public forums where users can post their articles and feedback for the information from others, but enrich the interactions of users such as posting videos, marking tags and votes. Moreover, the online communities reflect the social networks in real societies, in which the democracy plays hidden behind the crowds. This gives the requirements that finding high quality articles in an online community should have not only the **features** representing the semantical content, but the **features** embodying their popularity and reputation from users. Generally, the features are classified into two categories: **C-Features** and **U-Features**, which mean the **c**ontent related and the **u**sage related features respectively.

The issues that we propose to solve are:

1. is there any formal or analytic relationships between these two types of features and whether these relationships could be explored for content representation? and
2. is there any unified framework to solve the problem of recommending good articles?

We concentrate on the first issue in this chapter: identifying a set of features of social medias and interactions that can be applied to the quality evaluation. And the second issue will be discussed in the next chapter. The article set, that we use to investigate the dependency among different features, are taken from IT-Gipfelblog site.

Chapter Organization Firstly we discuss the content related features in Section 11.1; and then the usage related features in Section 11.2. In both sections, we depict the features by using statistics on IT-gifelblog. After that, we give the summary in Section 11.3.

11.1 C-Features: Tags, Keywords and Contexts

In web 2.0 online communities, users are given the convenience to express their ideas by using types of content such as pictures, texts and multimedia. Content related features describe the article characteristics from intrinsic side, but depend on the content type. The content features for a picture or video post are different from those of a text post. Tagging, the meaningful feedback from the users, enriches the features to represent the content.

11.1.1 Types of content

The basic elements published by users in an online community are articles, which are classified into two sub sets based on the roles: posts and comments. The post and its comments are usually displayed in a tree view or in a reverse chronological order in a page, or several pages. The relation diagram between a post **P** and its comments **C** is illustrated in Figure 11.1.

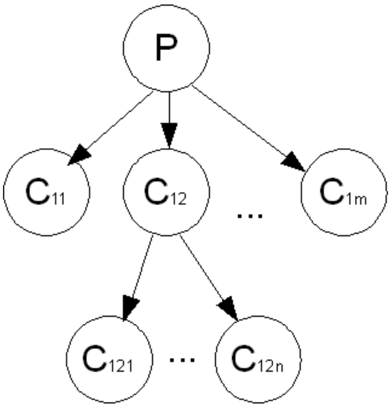


Figure 11.1: *Relation diagram between a post and its comments*

The relations among post and its comments form an ordered directed tree structure rooted by the post, in which the comments are inter nodes and leaves. A post is the beginner for a discussion, and a comment is the follower after a post or a comment. The arrow between two nodes represents the dependency between post and comment, or between two comments. This dependency has two meanings: semantic and chronological. The order among siblings represents the time order based on their publish date. If it is reasonable that the comments play the weights on evaluating the quality of one post, the comments on the lower level of the tree contribute fewer than those on the higher level. On the other hand, when the depth of a subtree rooted by one comment is deeper than that of the tree rooted by the post subtracted by this subtree, it shows that this comment starts another related topic deviated from the beginner post.

Depending on having videos or not, the articles are classified into text-related, video-related and hybrid. The hybrid article has a mixture content of text and video. Several methods are used to detect the boundary between video-related and hybrid: using the length of text in the article, using the existence of keywords, or marking manually by the editor. In IT-Gipfelblog, we use the second method to discriminate the hybrid posts from video-only posts: the word ”Themen” or ”Thema” reveal the existence of the text abstract of a post. We collect 394 posts and 371 comments from the launch date of IT-Gipfelblog till the end of march 2009. In 394 posts, there are 166 text-related posts, 145 video-only posts and 83 hybrid posts. Figure 11.2 gives the distribution of the comment number a post has.

Generally, the number of comments a post has, no matter text-related, video-related or hybrid, obeys a Zipf-like distribution; big part of posts have 0 comments and only very few posts have over 15 comments. However, different post type has clear preference on attracting comments shown in Figure 11.3: the text-related posts have a higher possibility to attract comments than the posts having videos, but there is no big taste difference between video-related and hybrid posts on attracting comments. If the existence of videos within a post suppress the users' drives on writing the comments is an interesting question. Interestingly, the similar finding has been observed by Spool et al. (Spool et al. 1998) that ”no evidence that graphics helped users to retrieve information on a web site”.

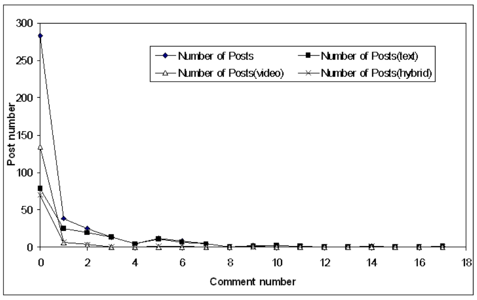


Figure 11.2: *Distribution of post vs. comment*

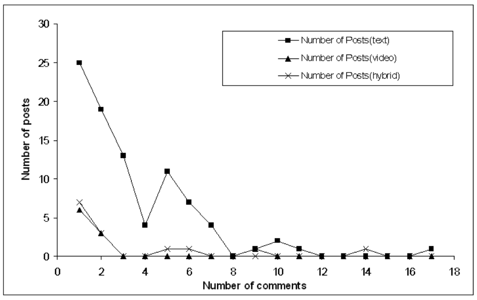


Figure 11.3: *Preference among types of posts*

11.1.2 Keywords, tags and contexts

Text corpora modeling has been already widely discussed in information retrieval, and as well in Chapter 3. The basic methodology proposed is to reducing each document in the corpus to a vector of real numbers, each of which represents the importance of a content feature to that document. In an online community, the features related to a post can be keywords, tags or topics.

Keywords Most of previous work on text modeling is keyword based. Keywords are stemmed and filtered with the weights showing their importance to a post, and the weight is usually measured by tf or $tf \times idf$ approaches. Compared with tags, the keywords related with a post could have a large scale, but each of them owns a measurable weight.

Tags Tags are marked by the readers or maintained by the editors in an online community, and a tag is usually a singular conceptual piece formed by several words for a post. Tags are especially indispensable to represent semantics for the posts with only videos. However, tags depend highly on the activities of users, though they have a higher-level abstraction on the content. Recently, discovering tag-based social interest attracted much attraction (Bateman et al. 2007)(Li et al. 2008). Xin Li (Li et al. 2008) found user-generated tags are consistent with the web content and more concise and closer to human understanding.

The dependency between posts and tags are shown in Figure 11.4. The distribution of the number of posts given n tags can be modeled by Gaussian distribution. It tells that the number of tags most posts have are in a stable scope, from 5 to 18 tags in this figure, and only small part of posts have fewer or more tags. We can see that very few tags are frequently noted in different posts, and big part of tags are marked only once. We noted that "interview" are shared in 211 posts, "IKT-Standort Deutschland" are used in 149 posts. The few overlap of tags demonstrates the big content difference among posts. This implies that it is not sufficient to filter posts by only using the content features. In our IT-Gipfelblog, only the posts accept the tags from users, but the comments do not.

Contexts Here we concentrate on how to get the topics hidden in a text post. Different from tags and keywords which are observable, the topics are latent. Latent semantic analysis (LSA) was discussed to discover the topics hidden in corpora and proven to be efficient for text modeling (Blei et al. 2003)(Hofmann 2001,). Recently LSA is used for detecting online reviews or opinions (Lu and Zhai 2008)(Titov and Mcdonald 2008). Topic feature has two advantages compared with tags

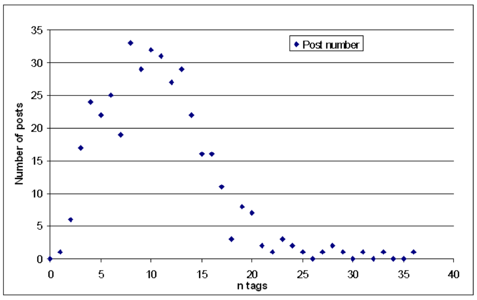


Figure 11.4: *Distribution of number of posts given n tags*

and keywords: one is that topics are closer to the semantics of text and more closer to human understanding, and the other is the scale of topics is much smaller than tags and keywords. Probabilistic latent semantic indexing(PLSI) (Hofmann 2001,) and Latent Dirichlet Allocation (LDA) (Blei et al. 2003) are the two representatives for topics discovering. In this thesis, we use LDA method. Suppose there are T topics hidden in a corpora composing by $|D|$ documents, the probability of the observed word-document pair (d, w) can be obtained by the marginalization over the laten topics:

$$P(d, w) = \sum_{i=1}^{T} \theta_d(t)\varphi_t(w),$$

where $\varphi_t(w)$ is the distribution of word w in latent topic t, while $\theta_d(t)$ is the distribution of topic t in document d. The difference between PLSI and LDA is that PLSI generates each document as a mixture of T topics (Blei et al. 2003), where the mixture coefficients are chosen individually for each document, while LDA generates a document by a word distribution $\varphi_t(w)$ from a prior Dirichlet distribtion $Dir(\beta)$ for each latent topic and by a topic distribution $\theta_d(t)$ for a document d from the symmetrical Dirichlet distribution $Dir(\alpha)$ as well.

In this thesis, we use tags, keywords and topics to represent the posts separately, and compare the difference on post clusters.

11.2 U-Features: Hits, Comments and Votes

Besides the content representativeness, the popularity is another consideration for article recommendation. In web 2.0, besides the hits (or clicks), the interactivities between the web content and the users are supplemented by the comments and the votes.

11.2.1 Hits, comments and votes

Hits The number of hits is the basic parameter to show the popularity of one article among the Internet users. Web usage mining, which aims to discover the users' interest from usage data, takes hits as a basic measurement used as support to evaluate the importance of a pattern (Agrawal and Imielinski 1993)(Agrawal and Srikant 1995)(Pei et al. 2000) (Wang and Meinel 2009). The usage behaviors are modeled as associate rules, sequence or other graphic patterns, but the parameters differentiating the usage patterns are computed based on hits.

Comments In an online community, the popularity of one article is supplemented by the number of comments and votes. We have discussed the comment in the previous section on C-Features. The users are allowed to give "thumbs up/down" on the comments following a post. The users express personal ideas on a post by using comments. And their preferences on the beginner post, good or not, are hidden in their comments.

Votes Vote is another direct feedback from users besides comment. The users express personal ideas on a post by using comments. And their preferences on the beginner post, good or not, are hidden in their comments. However, votes are the numerical judgements from the users, for instance, a registered user can give his personal evaluation on a post from 1 to 5. It is the reasonable premise that the users giving their comments or votes are included in computing the number of hits for a post, but is there any interactions among the number of hits, the goodness of comments and the value of votes?

11.2.2 Relations among hits, votes and comments

The scenario of users' reading, commenting and voting a post is similar to that of attending a research colloquium: **a section starts with a presentation attracting a group of attendants; after that, few attendants ask some questions or have some discussions; and finally some others leave their feedback in form of questionnaires. In this process, the askers and the attendants who give their feedback are the small part of the whole group. For the colloquium organizer, he would evaluate the quality of a section from the number of attendants and the satisfaction collected from the questionnaires, while the number of askers could be neglected.**

On IT-Gipfelblog, we observe that there are no clear roles between a good vote and a big number of voters or a big hits number. It is found that some posts received high hits number but fewer comments and votes, while some posts having high votes and controversial comments attracted few hits. This shows that the two groups of users who submitted their votes and comments separately for a post are only small parts of the whole users who visited this post, and these two groups are not overlapped. Another observation is that in many cases some comments after a beginner post are for some comments submitted before them, while not on the original post. So we assume that there is no dependency among the hits, comments and votes. The irrelevance among hits, votes and comments tells that a visitor concentrates more likely on the quality of the post itself than the reactions from others.

Going back to our work on evaluating the quality of posts from usage side, we select the vote of a post as one U-Feature while not the numbers of voters and comments, because they are included in and much smaller than the number of hits. Surely, the threshold for the minimum number of voters for each post is needed. Based on this observation, the post reputation (quality) from the usage side is measured by the linear function with the variables of the number of hits and vote. The rest task is to finding the suitable parameters for these two variables.

11.3 Summary for This Chapter

We have discussed the possible features for finding good posts from the content and usage aspects: C-Features and U-Features. By investigating the feature distributions and their relations, we select the proper features to evaluate the post quality. In the next chapter, we will discuss the mechanism on finding and recommending high reputation articles based on C-Features and U-Features.

Chapter 12

Finding and Recommending High Reputation Articles in a Social Site

One important difference between user-generated content in an online community and traditional content maintained by authoritative publishers such as companies and personals, is the wide range of content quality: from very high-quality items to low-quality, sometimes abusive content. This makes the task of quality evaluating in such social systems more complex than in other domains. On the other hand, the rich feedback from user side in an online community gives the potentiality to find an effective way of finding and filtering high quality content. Before selecting proper mechanism to evaluate the content quality, the content has to be represented by the quantified or description features: content related (C-Features) and usage related (U-Features).

However, what is the inter effect between C-Features and U-Features on ranking the content in online communities? One shortage of U-Features is that they can not discriminate the concept differences among content. In an online community, the balance between majority and minority is required for selecting the top K high quality content for recommendation, which means that top K content should not only cover the most important topics, but attract enough attentions from the users. So the high quality of a recommended post is shown from two sides: one is its content representing the relative topics from other posts, and the other is its acceptance from the populations. In this chapter, we try to solve the second purpose raised in the beginning of Chapter 11: is there any unified framework to solve the problem of recommending good articles?

Chapter Organization In this chapter, we firstly present the global and local reputation in a social site in Section 12.1. Then we discuss the algorithm clustering posts based on their content distance in Section 12.2. We explain how to select recommended articles from each cluster in Section 12.3. And Section 12.4 gives the summary of this Chapter.

12.1 Reputation: Global or Local?

After discussing the types of articles and the possible features used for quality evaluation, we now discuss the relations between usage related features and content related features. We especially in this section explain the global and local reputation. **Reputation is known to be a ubiquitous, spontaneous and highly efficient mechanism of social control in natural societies** (Ghose et al. 2006).

Based on the feature analysis in Chapter 11, we suppose the number of and the content of the comments one article received play few on measuring the reputation of an article in a social site. We select the vote of a post as an important feature measuring the reputation of an article, and the number of hits is another feature showing the reputation. Surely, the threshold for the minimum number of voters for each article is needed.

Table 12.1: *Classification based on hits and vote*

class	hits	vote	
c1	high	high	having global reputation
c2	high	low	need to be improved
c3	low	high	having local reputation
c4	low	low	no interest

12.1.1 Global vs. local reputation in a social site

Over the distribution between hits and votes shown in Figure 12.1, we divide the whole space into four partitions: c1, c2, c3 and c4. Different part represents an article class based on hits and votes, which is explained in Table 12.1.

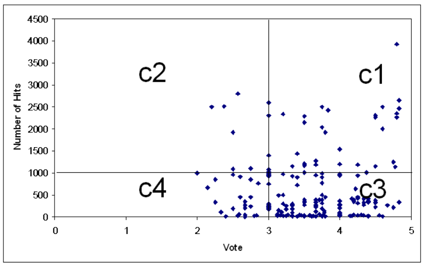


Figure 12.1: *Posts classified by hit and vote*

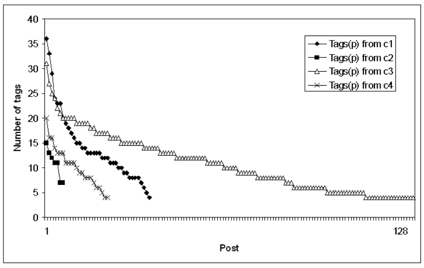


Figure 12.2: *Number of tags from 4 classes*

From this table, the posts in c1 should be recommended due to the positive reactions and those in c4 did not show their attractions to the users and should not be recommended. The post in c3 should be noticed and recommended as well, because low hits mean it has a minor group of visitors but high votes indicate it has a high reputation in this group. The post in c2 shows that it has the popular information, but needs to be improved due to the low reputation. Two methods can be used to define the boundaries between the partitions: from the expert judgement and from the statistics. The former is the empirical judgement of the experts on separating the four partitions and the latter gives the medium values of hits and votes on statistics.

We observed that most of the posts having high votes are accessed by small groups of visitors. Compared with few posts receiving high hits, each of such posts attracted only small part of visitors, but got very positive feedback from these visitors. Though it could be the abusive manipulation on the high votes by few users for some dedicated posts, the massive happenings of high votes with low hits is the reality of the discussions in online communities.

We use an example to interpret global and local reputation: assuming two posts Post 1 and Post 2 are supplied to 100 Internet users, and Post 1 was highly ranked and accessed by 95 users while Post 2 was highly evaluated and visited by 15 users, we call Post 1 has global reputation and Post 2 has local reputation. Figure 12.1 shows that most of the posts receiving high votes have local reputation, which means a minor group of users. We call this ”globally local reputation” in a social site. Global and local reputation is the majority and minority in Internet environment. However, a post having local reputation is mostly like the concept of ”majority minority”, which means the topics in this post are warmly welcomed in a small group population while not in the big rest population. The classification of posts in global or local reputation helps to filter out the uninteresting posts and to find the potential interesting posts in the high quality in a social site.

12.1.2 Does the reputation reflect the content features?

We further notice that the posts in global and local reputation reflect to some extent the semantic hierarchy among the discussed topics. For example, one post about the general policy on ”Green IT” has the global reputation, while the post on ”Green technologies in SAP” has local reputation, which is a concrete green IT policy executed by SAP. It seems that the general plain ideas are welcomed by massive public, while the concrete complex policy is interested by a small group. Moreover, the reputation from the post author affects the reputation of the article as well. In our observation, the post from ”Prof. Dr. August-Willhelm Scheer” is easily to have the global reputation than the one from an unknown interviewer, though the latter has a higher vote than the former.

The U-Features help to separate the posts having high votes from those having low votes, and further classify them into two groups: one having global reputation and the other having local reputation. However, this operation does not take account of the C-Features: if the quality of the separated posts can be proven from the aspects of content features? As discussed before, the content features depict the articles from the intrinsic facets. We now investigate the difference of the content features for these four classes.

As discussed in Section 11.1 that tags play great importance in finding social interest, we investigate the tag distribution for the posts from different classes. The distribution of the number of tags given a post from different classes is shown in Figure 12.2. Though the numbers of posts are different from 4 classes, the distribution of the number of tags per post tells the clear preference. For instance, the average number of tags per post from 24 posts in c4 having the lower hits and lower votes is 10.1; while 14.3 from 37 posts in c1 receiving higher hits and higher votes. For c3 vs. c2, the average number of tags are similar 10.8, but c3 has 133 posts much more than c2 having 7 posts. This observation ensures on some extent that evaluating the post quality from usage features can be proven from the aspects of content features. This brings us the foundation for finding and recommending top k high reputation posts in a social site.

12.2 Posts Clustering

To guarantee the concept independence among the recommended posts in a social site, we cluster the posts based on their content relevances and select the representative posts from each cluster. We tried two clustering methods to group the posts: one is the hierarchical agglomerative clustering, and the other is latent topic indexing.

12.2.1 Hierarchical agglomerative clustering

The one algorithms for document clustering is hierarchical agglomerative clustering (HAC) (Frakes and Baeza-Yates 1992). This method begins by placing each document into a distinct cluster, and pairwise similarities between every two documents are computed firstly. Then two closest clusters are merged into a new cluster. This process, computing pairwise similarities and merging the closest two clusters, is repeatedly applied. The general HAC algorithm is given in the following

Algorithm 4. For different applications, this iteration process stops when reaching one of two conditions:

1. the number of clusters reaches the predefined number;
2. all the pairwise similarities between clusters are lower than the threshold.

Algorithm 4 *The hierarchical agglomerative clustering algorithm*

```
form a list of $N$ clusters, each of which is initialized by a post $C_k \leftarrow p_k$
for $i = 1$ to $N - 1$ do
  for $j = i + 1$ to $N$ do
    compute pairwise inner cluster similarity $sim(C_i, C_j)$
  end for
end for
for $i = 1$ to $N$ do
  emerge the two closest clusters $C_{i'}$ and $C_{j'}$
  remove the similarities related with $C_{i'}$ and $C_{j'}$
  compute the pairwise similarities between this newly emerged clusters and other clusters
end for
for $i = 0$ to $N$ do
  compute log marginal likelihood under every iteration process
end for
```

Depending on how the similarity between two clusters is defined, we could obtain different clustering results. Moreover, the pairwise similarities between every two posts are the basic components to compute the similarity between two clusters. Cosine similarity,

$$sim(p_1, p_2) = \frac{\sum_{i=1}^{K} w_{p_1,i} \cdot w_{p_2,i}}{\sqrt{\sum_{i=1}^{K} w_{p_1,i}^2} \cdot \sqrt{\sum_{i=1}^{K} w_{p_2,i}^2}},$$

is used when a post is represented as a vector of tf or $tf \times idf$ showing the importance of its tags or keywords, because the post-tag or post-keyword matrix are typically sparse and cosine similarity can be fast computed.

The most common methods to compute the similarities between two clusters are single linkage, complete linkage and group average linkage. To differentiate the computing on clustering similarity, we call this similarity as "Inner Cluster Similarity". In this thesis, group average linkage is used to define the inner cluster similarity.

26. DEFINITION (INNER CLUSTER SIMILARITY). *Given two clusters C_1 and C_2 generated under the same clustering strategy, the inner cluster similarity between C_1 and C_2 is defined as:*

$$sim_I(C_1, C_2) = \frac{\sum sim(p_{1,i}, p_{2,j})}{|C_1| \times |C_2|},$$

where $p_{1,i} \in C_i$ and $p_{2,j} \in C_j$.

Besides the similarity on general distribution of the number of the clusters between tags and keywords, we investigate the difference on the composing of clusters internally. Because tags and keywords are in different scales, and setting the same inner cluster similarity threshold for both is not a prerequisite. So we compare the cluster composing under the same cluster number. We use X_T and X_K to name the cluster sets based on tags and keywords under the same cluster number constraint over the post set D, note that $|X_T| = |X_K|$ and each is one division over D. If X_T and X_K are correlated, X_T and X_K likely have large overlap. Extremely, $X_T == X_K$ when $|X_T| = |X_K| = 1$ or $|X_T| = |X_K| = |D|$. Before computing the correlation between X_T and X_K, we firstly give the definition of "Outer Cluster Similarity".

27. DEFINITION (OUTER CLUSTER SIMILARITY). *Given two clusters C_1 and C_2 generated by different strategies over the same dataset, the outer cluster similarity between C_1 and C_2 is computed by Jaccard similarity:*

$$sim_O(C_1, C_2) = \frac{|C_1 \bigcap C_2|}{|C_1 \bigcup C_2|}.$$

Based on the outer cluster similarity between two clusters, we define the clustering similarity between two cluster sets over the same post set.

28. DEFINITION (CLUSTERING SIMILARITY). *Given two cluster sets X_1 and X_2 over the same post set $D = \{p_1, ..., p_n\}$, the clustering similarity between X_1 and X_2 is defined as:*

$$sim_C(X_1, X_2) = \frac{\sum_{p_1}^{p_n} sim_O(C_{1,p_i}, C_{2,p_i})}{|D|},$$

where $C_{1,p_i} \in X_1$, $C_{2,p_i} \in X_2$, $p_i \in C_{1,p_i}$ and $p_i \in C_{2,p_i}$.

Because each post is assigned to only one cluster during any clustering strategy, this clustering similarity measures the biggest overlap between two cluster sets.

12.2.1. THEOREM. *Let X_1 and X_2 are two cluster sets over the same post set D. To measure clustering similarity between X_1 and X_2, the times of computing outer cluster similarity between X_1 and X_2 is $[\max(|X_1|, |X_2|), min(n, |X_1| * |X_2|)]$, and n is the size of the post set.*

3. PROOF. Suppose the two divisions of X_1 and X_2 on $\{p_1, ..., p_{|D|}\}$ are $\{C_{1,1}, ..., C_{1,|X_1|}\}$ and $\{C_{2,1}, ..., C_{2,|X_2|}\}$. For every $p_k \in D$, p_k has a cluster assignment in X_1 and X_2 respectively: $p_k \in C_{1,i}$ and $p_k \in C_{2,j}$. There are at most $\min(n, |X_1| * |X_2|)$ variations of $sim_o(C_{1,i}, C_{2,j})$ on D. On the other hand, every $C_{1,i}$ from X_1 is computed at least once with another cluster from X_2. And the same to every $C_{2,j}$ from X_2. So there are at least $\max |X_1|, |X_2|$ variations on computing $sim_o(C_{1,i}, C_{2,j})$.

During clustering the posts featured by tags and keywords respectively, two clusters are emerged only if their inner cluster similarity is over the predefined threshold. Under different inner cluster similarity thresholds, the log marginal likelihood (Meila and Heckerman 1998) is computed in every iteration process, which measures the distance between the cluster model and data set. The likelihood is computed as:

$$L(D|C_1, ..., C_K) = \sum_{k=1}^{K} \sum_{p_i \in C_k} logP(p_i|C_k).$$

12.2.2 Latent topics indexing for posts

Now we discuss the cluster formation based on contexts retrieved by LDA. Base on LDA, every post was taken as being generated from the dirichlet distribution of all topics, so the post-topic matrix is a matrix where each cell is valued by a real number. The same situation is for topic-keyword matrix. We follow the strategy in (Blei et al. 2003) to inference and estimate the values

Table 12.2: *Related stems in different topics*

Topic Nr.	Top K words for each cluster
1	arbeitslos steu kund softwar logisch design fehl statist
2	energi elektron gmbh verbrauch prozent intelligent entwickl energieversorg
3	moglichkeit programm ide user chanc global infrastruktur polit
4	themen spricht ikt it-gipfel it-gipfelblog bitkom prof scheer
5	it deutsch servic thema kund information frau
6	dienstleist anwend technisch internet inhalt sap technologi semant projekt
7	euro arbeit ausbild job unternehm steu sozialgesetz
8	internet it zukunft unternehm wirtschaft netz technik
9	polit heut firm spiel softwar unterstuetz selb
10	europa information bank erford eu elektron initiativ
11	entwickl internet technisch bereich wirtschaft produkt web
12	system management sap process plattn
13	deutschland deutsch polit initiativ blog diskussion bundesregier erst
14	comput nutzung schul einzeln elt digital
15	mensch gesetz polit unternehm deutsch welt international macht
16	function interview dr stell staat
17	recht bueger praxis branch manag staat
18	unternehm fachkraeft management ausbild international ikt erfolg staat
19	deutschland wirtschaft modern system dat basis staat polit
20	deutschland sap technologi wolfgang bundesregier

of these two matrixes. The number of topics is a huge reduction compared to tags and keywords, and it acts the same function as the number of clusters based on tags or keywords. For every topic, we use the top K posts based on their posterior possibilities, which forms a post cluster for this topic related with top K' words as well. In this case, one post could appear in multiple clusters, and the same to words.

The following Table 12.2 gives the top K words for each topic under $T = 20$. All the posts in IT-Gipfelblog are about the ideas on information technologies from experts and professionals, so the stemmed words are restricted in the ″expert and information″ domain. By investigating the top related stems for every topic, especially after overlooking the common stems, most of topics have their relatively clear concept boundary. For example, ″topic 1″ is about ″working position″, ″topic 2″ is about ″energy problem″ and ″topic 3″ is on ″globalization″. However, few of them have no clear topics, for example, ″topic 16″ is about ″interview″ which has no ″IT″ semantics and ″topic 9″ is about ″IT company″ but has no clear concept. It has to be admitted that the discovered topics by LDA are not so semantically intuitive as those by tags-based, and this will be further discussed in ″evaluation criteria″ section in Chapter 13. The reason for this is that the quality of selected words for each posts is highly domain biased, which is the same problem to keyword-based clustering.

In LDA, because one post could appear in multiple clusters, for the formed clusters $X = \{C_1, ..., C_T\}$ representing T discovered topics, we compute the coverage of clusters X over the whole post set D to measure its capacity.

$$coverage(X, D) = \frac{|\bigcup_1^T C_i|}{|D|}.$$

The low coverage indicates a high overlap between post clusters, which means that the current topics can not discriminate each other. Similarly, $\frac{|\bigcup_1^T c_i|}{K*T}$ is used to measure theoretically the percentage of unique posts against the happenings of all posts in clusters.

12.3 Potential Article Selection

A potential article for recommendation is defined as an article having global or local reputation. The commonness between global and local reputation is the high votes, while the difference between them is the hits number. Explained in the previous section, the content quality of post is somehow consistent with the feedback from the users, and the good feedback is classified into two groups having global and local reputation. One threshold for votes is needed to remove the posts having lower votes. To avoid the manipulation of high votes from abusive users, a minimum number of voters is set for the posts having high votes. And the threshold for hits is further used to classify the posts having higher votes into the posts having global and local reputation respectively.

After post clustering, each post has been assigned a cluster composing by other conceptually related posts. Next step is to remove the senseless clusters and select the potential clusters for recommendation. Two factors affect the refinement of clusters: one is the number of posts in a cluster, and the other is the reputation of a cluster. The reputation of a post cluster is composed by the reputation of the posts it has. The number of hits that a post cluster received is far larger than those of its comment and votes, so the reputation of a cluster can be simplified as the number of hits. The clusters having very few related posts and very little response from visitors are definitely not proper for recommendation, though in some cases the community moderator has to investigate the reason for the few usage feedback.

Suppose that a group of clusters are refined, now we focus on how to select a representative post from each cluster for recommendation. We use the simple linear integration of hits and votes each post has. The final rank of a post $R(p)$ is decided by its hit rank $R_h(p)$ and vote rank $R_v(p)$ compared with other posts in the same cluster: $R(p) = \alpha \times R_h(p) + \beta \times R_v(p)$, where $\alpha + \beta = 1$.

12.4 Summary for This Chapter

In this chapter, we have explained our strategy of finding high reputation articles in an online community: firstly clustering the conceptual related posts, and then selecting the representatives from each clusters based on their popularity ranks. The selected posts by this method guarantee the enough coverage on the discussed topics in an online community and the high quality shown by the popularity. We will further explain the experiment in the next chapter.

Chapter 13

Further Discussion on Recommendation

In this chapter, we further discuss and evaluate the experiment results on recommending the high reputation articles on IT-Gipfelblog.

Chapter Organization We firstly give the comparison among tags, keywords and contexts in Section 13.1; and based on the comparison, we discuss the clustering results on different methods in Section 13.2. Finally, the summary of chapter is given in Section 13.3.

13.1 Comparison among Tags, Keywords and Contexts

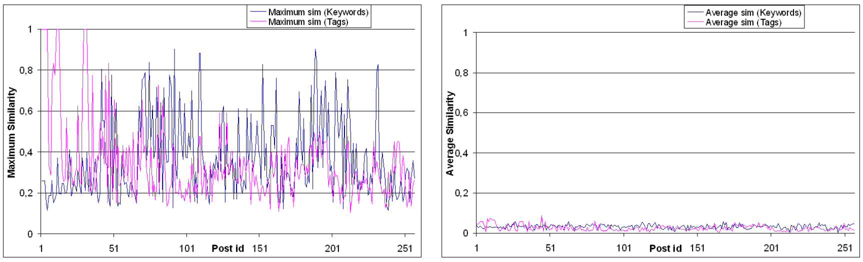
In this section, we give the comparisons between tags, keywords and contexts on post clustering. The post clustering is implemented over 264 posts covering most IT topics, and 953 tags were marked for these posts. After stemming and removing the common used stop words and those happening in less than 3 posts, we got 3180 keywords from 9447 stemmed words. The tags include the single words and pieces of combinations of words as well. Table 13.1 shows the general statistics on tags, keywords and posts.

After stemming the tags, we found the size of intersection between tags and keywords is 386, which means that 60% tags used are not included in the text of posts. Though both tags and keywords reflect the post content, they are generated from differen ways: the former are from the feedback of the users or the meta data from the moderator, while the latter come from the post content itself and are retrieved by NLP technologies. We further investigate the difference between clustering results from tags and keywords. Figure 13.1 gives the maximum and average similarity of a post compared with other posts based on tags and keywords. This figure shows that there is few dependency between tags and keywords on discriminating posts, and the big divergence between tag set and keyword set affects greatly locating the nearest partner for a post.

Figure 13.2 displays the log marginal likelihood under different inner cluster similarity thresholds based on tags and keywords. A fact should be noted that the number of posts based on tags is larger than that on keywords, because there are some posts with only videos while without texts, but they are marked with tags. From Figure 13.2, we explain the followings:

Table 13.1: *Max, Min and Avg number: tags/post, keywords/post, posts/tag and posts/keyword*

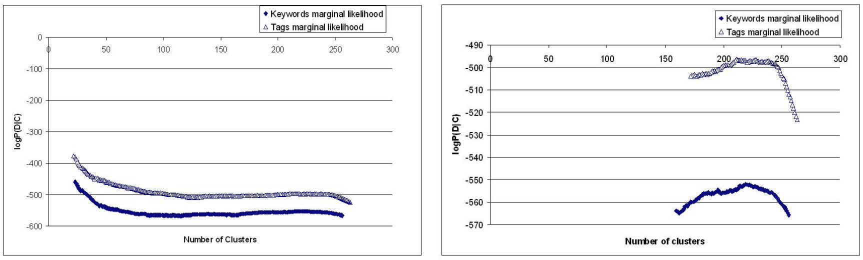
	Tags/post	Keywords/post	Posts/tag	Posts/keyword
Max.	36	314	109	117
Min.	2	1	1	3
Avg.	12	62	4	8



Maximum similarity of a post

Average similarity of a post

Figure 13.1: *Post Similarity: Tags vs. Keywords*



(a)Log marginal likelihood $\theta = 0.1$

(b)Log marginal likelihood $\theta = 0.3$

Figure 13.2: *Clusters Tags vs. Keywords*

1. the empty areas between 0 and the starting curves in both (a) and (b) mean that no new clusters are emerged after some iterations under a inner cluster similarity threshold. A lower threshold generates more clusters than a higher one;

2. with the decreasing of inner cluster similarity threshold, the maximum likelihood is caught at the stop of iterations. If similarity threshold is set 0, the maximum likelihood is got at the only one cluster including all the posts;

3. clusterings under tags and keywords generate the similar likelihoods distribution, only differ numerically, and vary at the points of iteration stop and the maximum likelihood;

4. the reason, why there is the great shifting of the number of clusters generated at the maximum likelihood from $\theta = 0.1$ to $\theta = 0.3$ (25 vs. 200 for both tags and keywords), is the compositions of posts: some of them are highly semantically related, some are in lower similarities, and some posts are totally in uninvolved topics; and

5. likelihood computed based on tags is always larger than that on keywords, this is because the dimension of tags is much smaller than that of keywords which reflects a high confidence on representing the concept of posts.

The higher inner cluster similarity threshold could generate more clusters, and lower threshold could cluster posts more compactly. However, it makes no sense to cluster all posts in one cluster by setting inner cluster similarity threshold as 0. In practice, the number of clusters formed at the maximum likelihood is even far larger than the allowed recommendation space within a web page, is 10 usually, and 20 at most. The dimensions of tags and keywords affect greatly selecting inner cluster similarity threshold.

HAC method merges two clusters with maximum similarity at each iteration, which means that the number of clusters based on tags and keywords reduce in the same pace during iteration process. Figure 13.3 shows the clustering similarity between tags and keywords at different emerging steps under inner cluster similarity threshold $\theta = 0.0$, which means that all the posts are merged into one cluster at the last iteration step. From this figure, the clustering similarity gets its minimum value when the cluster number is 17.

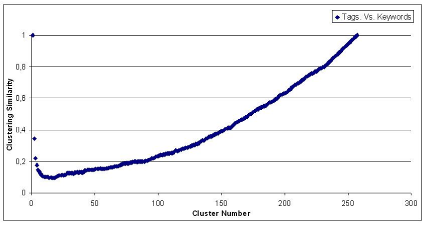


Figure 13.3: *Clustering Similarity*

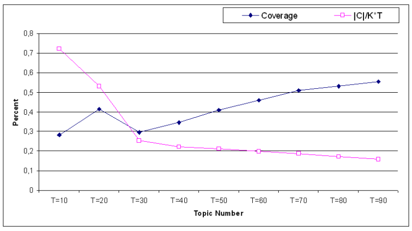


Figure 13.4: *Coverage of clusters under LDA*

Now we discuss the cluster results based on hidden topics retrieved by LDA. We tried setting different number of topics and selected the top K posts to form a post cluster for every topic. Figure 13.4 gives these two coverage measurements under different topic numbers. It is noticed that with the increasing of topic number, the coverage increases in a stable tendency (when $T > 20$, and T is the number of topics). However, a high number of topics is not practical in compacting the posts, and an idea number topics should have the ability to compact the posts effectively and have a high coverage as well. Seen from this figure, a high coverage with a high compacting ability is reached at $T = 20$, which means that partitioning the posts into 20 clusters is an ideal choice.

We observed that the discovered topics by LDA are not so semantically intuitive as those by tags-based, and this will be further discussed in next Section. The reason for this is that the quality of selected words for each posts is highly domain biased, which is the same problem to keyword-based clustering.

13.2 Criteria on Evaluating Recommendation

As discussed in Section 4.5 on pattern evaluation and interpretation, technical, expert and task oriented standards are used to judge the usage interest. To evaluate the clustering results theoretically, routine technical criteria like precision and recall are usually used. These criteria are properly necessary when clustering on huge corpus in which the independency can be regarded in the generation of documents. However, in an online community interest oriented, where the number of posts is controlled and the related topics are guided and maintained by the moderator, precision and recall criteria are not practical in the post recommendation. But it is necessary to evaluate the quality of recommended posts, especially for the discrimination between tags, keywords, contexts and users. Here we use the human reviews to approximate this goal, which is the combination of expert and task oriented standards.

In our experiment, 3 moderators were asked to do the following work:

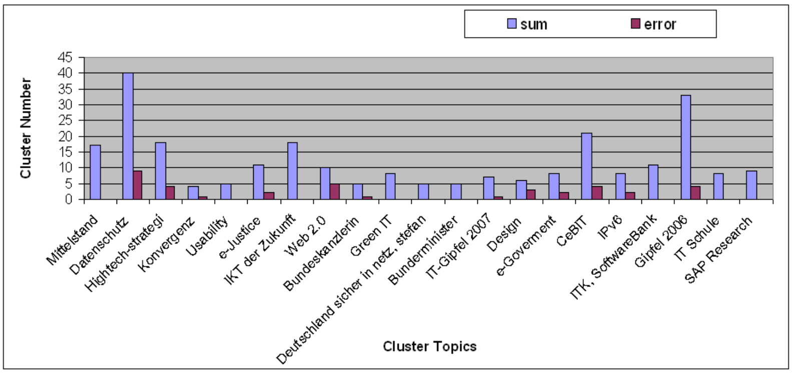


Figure 13.5: *Error in clustering based on tags*

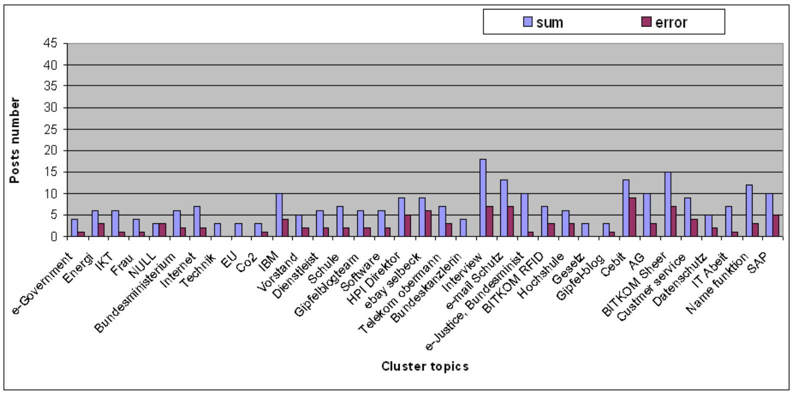


Figure 13.6: *Error in clustering based on top 10 keywords*

1. reading through the clustering results and giving the topics for every clusters; and
2. in every cluster, marking the posts that are not properly related with the cluster's topics

Based on the feedback of reviewers, we could evaluate the precision of different clustering strategies.

Figure 13.5 gives the moderators' reviews on the clusters based on tags under inner cluster similarity threshold 0.1, and 21 clusters are built. The topics every cluster is related are selected from the marked tags.

Figure 13.6 shows the reviews on the clusters based on top 10 keywords under inner cluster similarity threshold 0.1, and 36 clusters are built. The topics every cluster has are selected from the keywords. Definitely, we tried as well selecting top 20 and top 30 keywords to cluster posts. The numbers of clusters by selecting top K keywords under the same inner cluster similarity threshold are not greatly different, which is shown in Figure 13.8. When considering all the keywords stemmed from posts into clustering, $K > 61$ averagely, the number of clusters is 20 under similarity threshold 0.1. This means that the number of keywords (but must reach one value) is unimportant to represent the content of posts. However, the compositions of clusters are greatly different between top 10, 20 and 30 words. Moreover, it is hard to summarize the topics a cluster

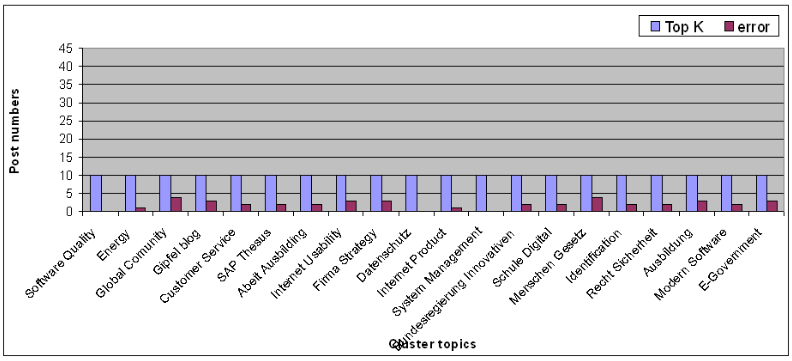


Figure 13.7: *Error in clusters based on contexts*

has when using top 20 and 30 keywords, this shows that the clustering error is very high. This is due to three reasons:

1. the first is that the risk on misclassification rises wit the increasing of K statistically;
2. the second is the quality of keywords, which depends not only on $tf \times idf$ value, but also the semantic domain; and
3. the third is the negative effect of the posts with videos, in which the text content is not enough to describe the semantics of a post.

Figure 13.7 presents the reviews on the clusters based on contexts by LDA. The number of topics is set 20, at which a relatively high capacity and compactness is reached. For every cluster, we select top 10 posts based on their posterior possibilities.

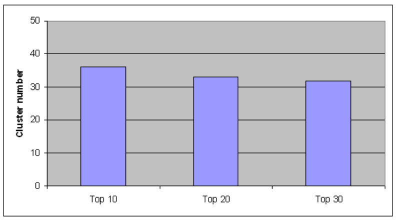


Figure 13.8: *Cluster numbers: top K words*

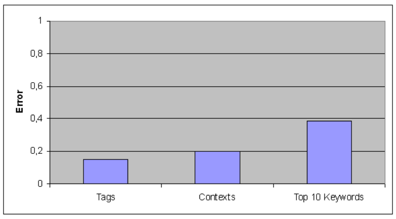


Figure 13.9: *Error: tags vs. top 10 keywords*

We compare the reviews on tags, keywords and contexts from two sides: error ratio and quality of cluster topics. The former is measured by the sum of posts divided by the number of posts clustered in error, and the comparison is given in Figure 13.9. The error on tags is much lower than those on top K keywords and contexts. The cluster topics based on tags are much human understandable than those on keywords and contexts, for example, the non semantic topics like ”women”, ”interview”, ”vorstand” and ”AG” are found from keywords.

The gap exists apparently between the ”cluster topics” marked by human reviews and those retrieved automatically by clustering or LDA, clustering based on tags supplies much intuitive and exact results than keywords and contexts based methods.

13.3 Summary for This Chapter

This chapter continued the discussion on our experiment on finding and recommending high reputation articles in a social site. The evaluation criteria for expert reviewers was explained in this chapter. The judgement from reviewers showed that HAC clustering based on tags is better than those on keywords and hidden topics. From the work discussed in Part III, we draw the following conclusions on finding and recommending high reputation articles in a social site:

1. global and local reputation is ubiquitous in an online community;

2. the reputation one article received can be somehow proven on its content features; and

3. HAC clustering based on tags is a suitable way to cluster the content related articles in a social site.

Chapter 14

Conclusions and Future Work

Usage interest analysis is a strategic important and challenging task for online business. During the data mining process, the challenging tasks are: processing huge volume data, managing high dimension relations, discovering potential patterns among entities and interpreting interesting patterns.

This thesis illustrates the methodologies and shows the evidence on finding online usage interest by analyzing the spontaneous, notably imperfect and greatly noisy usage data. We implemented the mining methodologies on different kind of web sites: on a public portal site, on a web-streaming e-learning site and on a social site. The contributions of this thesis are listed:

1. solving the problem of recovering individual navigation behaviors in browsing an information portal site, which is the necessary premise for the posterior usage pattern mining;

2. giving a general and unified method on tracking the changes of web navigation patterns, which not only locates the nearest version of a pattern in posterior time span, but measures its internal and external variances from structural and semantic sides;

3. modeling the learning interest on browsing kinds of streaming lectures and discovering the learning interest by answering six questions;

4. measuring the changes of learning interest on the same course from different semesters, which integrates the variance of usage interest with the difference of the courses in different years; and

5. presenting a framework on evaluating and recommending high reputation articles in a social site, which is based on the proof that the articles are classified into local and global categories based on their reputation. The proposed framework considers the balance between the article concept and usage feedback, between the interest from major users and minor users.

On different site, we have had different usage data, used different mining methods, implemented different platforms and discovered different formats of usage patterns and usage interest. However, the logic behind mining usage interest on different web sites is the same: **understanding the data is the key for knowledge discovery**. From the work on data investigation and discovery, we learnt that the significance of usage mining is to depict and discover the usage patterns in a more compressed, structured, direct and understandable way from the huge, unstructured, indirect and un-interpretable usage data.

This thesis concentrates on discovering the structural usage patterns from the technical side, which belongs to the indirect investigation on usage interest. However, we have to admit that there is still a long way on understanding the usage interest from their usage data, especially on letting the mined information be more readable, acceptable and applicable. Because usage interest and patterns cover a huge scope, there is no universal template and standard to model the usage interest, and treating the usefulness of the newly discovered information is a more subjective task. Moreover, for the web sites discussed in this thesis, which are open and no-profit driven, there is the lack of widely accepted standard to evaluate the exactness and success of web services and web sites. So three directions stand in front of us for the research in the future:

1. finding other methods on pattern discovery and evaluation, especially restricted on the concrete mining targets;

2. making a large scale direct investigation on the web users and comparing it with the mined results in this thesis; and

3. covering other usage data, especially the online shopping, online map search, and online gaming site, on which the users express their interest in a more direct way.

Bibliography

Agrawal, R. and Imielinski, T.: 1993, Mining association rules between sets of items in large databases, *Proc. SIGKDD*.

Agrawal, R. and Srikant, R.: 1995, Mining sequential patterns, *Proc. ICDE 1995*, pp. 3–14.

Bateman, S., Brooks, C., McCalla, G. and Brusilovsky, P.: 2007, Applying collaborative tagging to e-learning, *Proc. Tagging and Metadata for Social Information Organization Workshop, WWW07*.

Bayardo, J. and Roberto, J.: 1998, Efficiently mining long patterns from databases, *Proc. SIGMOD 1998*, pp. 85–93.

Berendt, B.: 2005, The semantics of frequent subgraphs: Mining and navigation pattern analysis, *Proc. WebKDD 2005*, pp. 21–24.

Blei, D. M., Ng, A. Y. and Jordan, M. I.: 2003, Latent dirichlet allocation, *Journal of Machine Learning Research* **3**, 993–1022.

Burton, M. C. and Walther, J. B.: 2001, A survey of web log data and their application in use-based design, *Proc. of 34th International Conference on Systems Sciences*.

Chawathe, S.: 1999, Comparing hierarchical data in external memory, *Proceedings of the Twentyfifth International Conference on Very Large Data Bases*.

Chen, J., Sun, L., Zaane, O. R. and Goebel, Y.: 2004, Visualizing and discovering web navigational patterns, *Proc. of Workshop WebDB on SIGMOD 2004*, pp. 13–18.

Chen, M. S., Member, S., Park, J. S. and Yu, P. S.: 1998, Efficient data mining for path traversal patterns, *IEEE Transactions on Knowledge and Data Engineering* **10**, 209–221.

Cooley, R., Mobasher, B. and Srivastava, J.: 1999, Data preparation for mining world wide web browsing patterns, *Knowledge and Information Systems* **1**, 5–32.

Dong, G. and Li, J.: 1999, Efficient mining of emerging patterns: Discovering trends and differences, *Proc. SIGKDD 1999*, pp. 43–52.

Eirinaki, M., Lampos, H., Vazirgiannis, M. and Varlamis, I.: 2003, Sewep: Using site semantics and a taxonomy to enhance the web personalization process, *Proc. SIGKDD 2003*, pp. 99–108.

Fayyad, U., Haussler, D. and Stolorz, P.: 1996, Mining scientific data, *Communications of the ACM* **39**(11), 51–57.

Fortin, S.: 1996, The graph isomorphism problem, *Technical report*.

Frakes, W. B. and Baeza-Yates, R. A. (eds): 1992, *Information Retrieval: Data Structures & Algorithms*, Prentice-Hall.

Geng, L. and Hamilton, H. J.: 2006, Interestingness measures for data mining: A survey, *ACM Comput. Surv.* **38**(3), 9.

Ghose, A., Ipeirotis, P. G. and Sundararajan, A.: 2006, The dimensions of reputation in electronic markets, *Social Science Research Network Working Paper Series* .

Gomory, R. E. and Hu, T. C.: 1961, Multi-terminal network flows, *Journal of the Society for Industrial and Applied Mathematics* **9**(4), 551–570.

Graff, M.: 2005, Individual differences in hypertext browsing strategies, *Journal Behaviour & Information Technology* **24**(2), 93–99.

Han, J., Pei, J. and Yin, Y.: 2000, Mining frequent patterns without candidate generation, *Proc. SIGMOD 2000*, pp. 1–12.

Hardy, J., Antonioletti, M. and Bates, S.: 2004, e-learner tracking: Tools for discovering learner behaviour, *International Conference IASTED Web-Based Education*.

Heer, J. and Chi, H.: 2002, Mining the structure of user activity using cluster stability, *Proc. of workshop on Web Analytics on SIAM 2002*, ACM Press.

Herder, E. and Juvina, I.: 2004, Discovery of individual user navigation styles, *Workshop on Individual Differences in Adative Hypermedis at AH 2004*, pp. 40–49.

Hofmann, T.: 2001,, Unsupervised learning by probabilistic latent semantic analysis, **42**(1-2), 177–196.

Huang, X., Peng, F., An, A. and Schuurmans, D.: 2004, Dynamic web log session identification with statistical language models, *J. Am. Soc. Inf. Sci. Technol.* **55**(14).

Kleinberg, J. M., Kumar, R., Raghavan, P. and Tomkins, A. S.: 1999, The web as a graph: Measurements, models and methods, *Proceedings of the International Conference on Combinatorics and Computing*, pp. 1–17.

Li, X., Guo, L. and Zhao, Y.: 2008, Tag-based social interest discovery, *Proc. WWW*.

Liu, B., Zhao, K. and Yi, L.: 2002, Visualizing web site comparisons, *Proceedings of the 11th international conference on World Wide Web*, pp. 693–703.

Lu, Y. and Zhai, C.: 2008, Opinion integration through semi-supervised topic modeling, *Proc. WWW*.

McAleese, R.: 1999, *Navigation and browsing in hypertext*, Intellect Books, Exeter, UK, UK.

Meila, M. and Heckerman, D.: 1998, An experimental comparison of several clustering and initialization methods, *Machine Learning*.

Meila, M. and Pentney, W.: 2007, Clustering by weighted cuts in directed graphs, *Seventh SIAM International Conference on Data Mining*.

Melnik, S., Garcia-molina, H. and Rahm, E.: 2002, Similarity flooding: A versatile graph matching algorithm, *Proc. ICDE 2002*, pp. 117–128.

Mobasher, B.: 2006, *Web Data Mining: Exploring Hyperlinks, Contents and Usage Data*, Springer Berlin-Heidelberg.

Mostow, J., Beck, J., Cen, H., Cuneo, A., Gouvea, E. and Heiner, C.: 2005, An educational data mining tool to browse tutor-student interactions: Time will tell!, *AAAI workshop on educational data mining*, pp. 15–22.

Nakayama, T., Kato, H. and Yamane, Y.: 2000, Discovering the gap between web site designers' expectations and users' behavior, *Comput. Netw.* **33**(1-6).

Page, L., Brin, S., Motwani, R. and Winograd, T.: 1999, The pagerank citation ranking: Bringing order to the web.

Pei, J., Han, J., Mortazavi-Asl, B. and Zhu, H.: 2000, Mining access patterns efficiently from web logs, *Proc. PAKDD 2000*, pp. 396–407.

Perkowitz, M. and Etzioni, O.: 2000, Towards adaptive web sites: Conceptual framework and case study, *Artificial Intelligence* **118**, 245–275.

Piatetsky-Shapiro, G. and Frawley, W. J. (eds): 1991, *Knowledge Discovery in Databases*, AAAI/MIT Press.

Reisslein, J., Seeling, P. and Reisslein, M.: 2005, Video in distance education: Itfs vs. webstreaming: Evaluation of student attitudes, *The Internet and Higher Education* **8**(1).

Reynolds, P. A. and Mason, R.: 2002, On-line video media for continuing professional development in dentistry, *Computer Education* **39**(1).

Rizzi, S., Bertino, E., Catania, B., Golfarelli, M. and Halkidi, M.: 2003, Towards a logical model for patterns, *Proc. ER*.

Sarukkai, R. R.: 2000, Link prediction and path analysis using markov chains, *Proc. WWW 2000*, pp. 377–386.

Schillings, V. and Meinel, C.: 2002, tele-task: teleteaching anywhere solution kit, *SIGUCCS '02: Proceedings of the 30th annual ACM SIGUCCS conference on User services*, pp. 130–133.

Schmitt, E., Manning, H., Paul, Y. and Tong, J.: 1999, Measuring web success, *Forrester Report* .

Shenoy, P., Bhalotia, G., T, J. R. H., Bawa, M., Sudarshan, S. and Shah, D.: 2000, Turbo-charging vertical mining of large databases, *Proc. SIGMOD 2000*, pp. 22–33.

Shi, J. and Malik, J.: 1997, Normalized cuts and image segmentation, *Proceedings of the 1997 Conference on Computer Vision and Pattern Recognition (CVPR '97)*.

Siberschatz, A. and Tuzhilin, A.: 1996, What makes patterns interesting in knowledge discovery systems, *IEEE Transactions on Knowledge and Data Engineering* **8**, 970–974.

Spool, J. M., Scanlon, T. and Schroeder, W.: 1998, *Web Site Usability: A Designer's Guide*, Morgan Kaufmann.

Srikant, R. and Yang, Y.: 2001, Mining web logs to improve website organization, *Proc. WWW 2001*, pp. 430–437.

Srivastava, J. and Cooley, R.: 2000, Web usage mining: Discovery and applications of usage patterns from web data, *SIGKDD Explorations* **1**, 12–23.

Sugiyama, K., Hatano, K., Yoshikawa, M. and Uemura, S.: 2003, Refinement of tf-idf schemes for web pages using their hyperlinked neighboring pages, *Proc. HYPERTEXT 2003*, pp. 198–207.

Tan, P. N. and Kumar, V.: 2000, Interestingness measures for assocation patterns: A perspective.

Tanasa, D. and Trousse, B.: 2004, Advanced data preprocessing for intersites web usage mining, *IEEE Intelligent Systems* **19**(2).

tau Yih, W., Goodman, J. and Carvalho, V. R.: 2006, Finding advertising keywords on web pages, *Proc. WWW 2006*, pp. 213–222.

Tauscher, L., Tauscher, L. and Greenberg, S.: 1997, Revisitation patterns in world wide web navigation, *Proc. of ACM CHI*, ACM Press, pp. 399–406.

Titov, I. and Mcdonald, R.: 2008, Modeling online reviews with multi-grain topic models, *Proc. WWW*.

Torrance, H. and Pryor, J.: 1998, Investigating formative assessment: Teaching, learning and assessment in the classroom, *Buckingham: Open University Press* .

Wagner, R. A. and Fischer, M. J.: 1974, The string-to-string correction problem, *J. ACM* **21**(1), 168–173.

Wang, L. and Meinel, C.: 2004, Behaviour recovery and complicated pattern definition in web usage mining, *Proc. ICWE 2004*.

Wang, L. and Meinel, C.: 2006, Building content clusters based on modelling page pairs, *Proc. APWeb 2006*.

Wang, L. and Meinel, C.: 2007a, Detecting the changes of web students' learning interest, *IEEE/WIC/ACM International Conference on Web Intelligence*, pp. 816–819.

Wang, L. and Meinel, C.: 2007b, Mining the students learning interest in browsing web-streaming lectures, *IEEE Symposium on Computational Intelligence and Data Mining*, pp. 194–201.

Wang, L. and Meinel, C.: 2009, X-tracking the changes of web navigation patterns, *Proc. PAKDD 2009*.

Wang, L., Meinel, C. and Liu, C.: 2005, Discovering characteristic individual accessing behaviors in web environment, *Proc. RSFDGrC*.

Yi, Y.: 1989, A critical review of consumer satisfaction, *Review of Marketing, Amercian Marketing Association* pp. 68–123.

Youssefi, A. H., Duke, D. J. and Zaki, M. J.: 2004, Visual web mining, *Proc. on Alternate track papers & posters on WWW 2004*, pp. 394–395.

Zaki, M.: 2002, Efficiently mining frequent trees in a forest, *Proc. SIGKDD*.

Zhang, D., Zhao, J. L., Zhou, L. and Jay F. Nunamaker, J.: 2004, Can e-learning replace classroom learning?, *Commun. ACM* **47**(5).

Zhao, Q. and Bhowmick, S. S.: 2004, Mining history of changes to web access patterns, *Proc. PKDD 2004*.

Zhao, Q., Bhowmick, S. S., Mohania, M. and Kambayashi, Y.: 2004, Discovering frequently changing structures from historical structural deltas of unordered xml, *Proc. CIKM 2004*, ACM Press, pp. 188–198.

Zinn, C. and Scheuer, O.: 2006, Getting to know your student in distance learning contexts, *European Conference on Technology Enhanced Learning*, pp. 437–451.