Finding Usage Patterns from Generalized Weblog Data

Tahira Hasan

A Thesis

in

The Department

of

Computer Science and Software Engineering

Presented in Partial Fulfillment of the Requirements

for the Degree of Master of Computer Science at

Concordia University

Montréal, Québec, Canada

April 2009

Your file Votre référence
ISBN: 978-0-494-63172-0
Our file Notre référence
ISBN: 978-0-494-63172-0

ABSTRACT

Finding Usage Patterns from Generalized Weblog Data

Tahira Hasan

Buried in the enormous, heterogeneous and distributed information, contained in the web server access logs, is knowledge with great potential value. As websites continue to grow in number and complexity, web usage mining systems face two significant challenges - *scalability* and *accuracy*. This thesis develops a web data generalization technique and incorporates it into the web usage mining framework in an attempt to exploit this information-rich source of data for effective and efficient pattern discovery. Given a concept hierarchy on the web pages, generalization replaces actual page-clicks with their general concepts. Existing methods do this by taking a level-based cut through the concept hierarchy. This adversely affects the quality of mined patterns since, depending on the depth of the chosen level, either significant pages of user interests get coalesced, or many insignificant concepts are retained. We present a usage driven concept ascension algorithm, which only preserves significant items, possibly at different levels in the hierarchy. Concept usage is estimated using a small stratified sample of the large weblog data. A usage threshold is then used to define the nodes to be pruned in the hierarchy for generalization. Our experiments on large real weblog data demonstrate improved performance in terms of quality and computation time of the pattern discovery process. Our algorithm yields an effective and scalable tool for web usage mining.

To My Mother

your dreams are my aspirations

ACKNOWLEDGEMENTS

It's a special feeling when a long journey, that you are a part of, comes to an end. I take immense pleasure in acknowledging the dedication, prayers and support of all those people who have helped me travel through the two years of my Masters program and made writing this thesis possible.

First of all, my deepest gratitude goes to my supervisors Dr. Nematollaah Shiri and Dr. Sudhir Mudur. I am grateful for their commitment to my thesis and for the amount of time they have invested in it. Thanks to Dr. Shiri for introducing me to the most challenging intellectual experience that I have embarked on and for providing me with many helpful suggestions and important advice during the course of this work. Thanks to Dr. Mudur for the constructive comments and the constant encouragements during times of difficulties. Your ability to select and to approach compelling research problems set an example and our discussions frequently led to key insights. It has been a distinct privilege for me to work with both of you.

I am indebted to the Department of Computer Science at Concordia University for providing state-of-the-art computing resources and a supportive environment making the time working on my research a pleasurable experience. A special thanks to Bhushan Suryavanshi for his contributions in the development of the RFSC algorithm used in this research.

I must acknowledge the invaluable contributions of all my undergraduate teachers at International Islamic University, Islamabad, with whom I began to learn about the theories of data analysis and the art of computer programming. It was their belief

and unfaltering encouragements that set me on the right track. Thanks to all my friends for bearing with my little availability during this time. Thankyou Auj, for all those interesting talks which helped me keep life in context; this thesis hasn't always been your greatest friend, but I really appreciate your support when I needed it the most.

I wish to thank my brother and his family for their patience, love and encouragement that kept me going during hard times of this study. Thanks Bhai, for being that one person around who thinks that research is fun and trying with full enthusiasm to make me believe it. I must admit, that you did not manage to convince me but you definitely killed my fear of diving into this world of endless readings and writings. Bhabi, thanks for listening to my complaints and frustrations, for keeping me motivated and of course for stuffing me with delicious food. Thankyou, Minahil for being my best friend and confidant, Sireen, for making me smile even when I was irritated by my work and Saad for being my little bundle of joy during the most testing phase of my research.

Finally, I wish to say a special thankyou to the two most special people in my life. It would take another thesis to express my deep love for my parents. I am thankful for their love, their support, and their confidence throughout the past twenty-four years of my life. Abba, thanks for inspiring me to set high goals in life and for guiding me to achieve them. Mama, it has been your sheer conviction and unfathomable faith that drives me to work hard and attain my goals with dignity. I owe all my accomplishments to both of you.

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF TABLES

LIST OF ABBREVIATIONS

AOI	Attribute Oriented Induction
LCS	Longest Common Subsequence
OLAP	Online Analytical Processing
RFSC	Relational Fuzzy Subtractive Clustering
URL	Uniform Resource Locator

Chapter 1

Introduction

"The sheer volume of information

dissolves the information"

Günter Wilhelm Grass

The World Wide Web, known as the *web*, has made a phenomenal contribution in disseminating information across the globe. Over the years, the number of information seekers and providers has grown exponentially, creating a highly competitive environment. This has made it indispensable for the providers to recognize the information needs of their users in order to attract new visitors and retain the existing ones.

With more than 180 million websites and 1.5 billion web users [52, 36], the resulting web browsing activities generate huge amount of data, recorded in web server access log files. Since a weblog keeps track of the users' browsing behavior

down to individual mouse clicks, manual inspection of this immense amount of data becomes virtually impossible. Therefore, the use of data mining techniques (known as *web usage mining*) for the modeling and analysis of such data, was the natural alternative step and is now the focus of an increasing number of researchers.

Web Usage Mining [12, 21] is formally defined as the application of data mining techniques to discover and analyze interesting patterns in the browsing behavior of web users. Over the years, it has emerged as an essential tool for providing more personalized, user friendly and improved Web services. This has resulted in the development of many successful applications such as web personalization [18], site improvements [44], marketing decision support [6], web caching and prefetching [49].

1.1 Motivation

The ultimate goal of web usage mining is to capture, model and analyze the behavioral patterns (such as association rules or classification rules) and profiles of the users' interactions with a website, that are of potential interest and also reflective of the users' preferences. However, in some of the earlier proposals for web usage mining (*e.g.* [14]) the quality of the extracted patterns was poor due to the shallowness of the data available to these systems. Relying on the web usage data alone can be ineffective due to two reasons. Firstly, in cases where little or no usage information is available (such as in the case of newly added pages), the system fails to draw reasonable conclusions about such items. Moreover, as a result of insufficient usage data two pages that

are never used together will not be identified as similar even if they are semantically (content-wise or structurally) related to each other. Secondly, the use of dynamic URLs hampers the ability of these systems to interpret or reason about the discovered usage models, since the resulting profiles only contain usage patterns at the URL level. For example [53], consider the following cryptic association rule, in the context of an online bookstore: "If `http://www.the shop.com/show.html?item=123`, then `http://www.the shop.com/show.html?item=456`, *support* = 0.5, *confidence* = 0.4". This does not give any useful interpretation of the user intentions. On the other hand, actionable patterns like "*Users who bought* 'Hamlet' *also tended to buy* 'How to stop worrying and start living'," could give significant insights into the underlying reasons for particular user behaviors.

A common approach to compensate for these limitations of the usage data is to integrate content characteristics of pages into the web usage mining process [17]. Generally, in this approach, URLs are either mapped to keywords extracted from the content on the Web site or classified into various content categories obtained from externally available concept hierarchies. The integration of domain knowledge can capture patterns of user interests at a deeper semantic level, exploit the underlying dependencies among the users' navigational behaviors and in general improve the quality of the web usage mining.

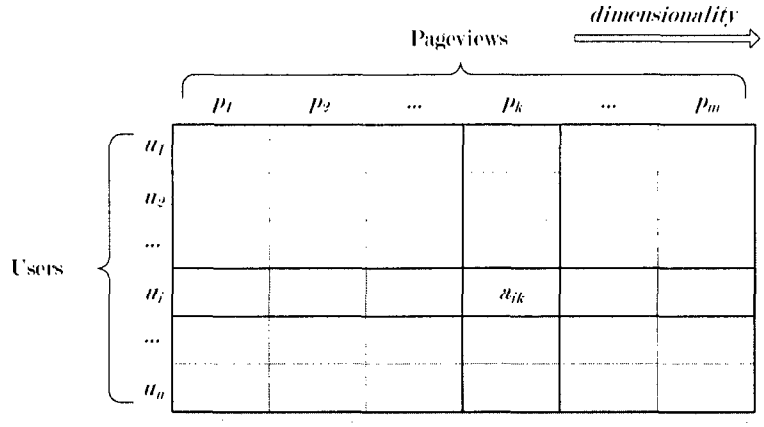


Figure 1.1: Skeletal Structure of User-Pageview Matrix

1.1.1 Challenges in Web Usage Mining

It is clear from the published literature that semantically enriched web usage mining is successful in overcoming the shortcomings of the traditional systems, but there are still some challenges that need to be addressed.

The primary choice of data representation for web usage mining procedures has been the *vector space model* [62, 50, 47], where each visit of a user to a website is encoded as a vector whose dimensions correspond to the total number of distinct URLs in a website. Figure 1.1 shows a skeletal structure of the user-pageview matrix composed of all such vectors, capturing the past transactions. Although this simple representation is claimed to facilitate various data mining algorithms (*e.g.*, clustering and association rule mining), its usability is limited by the following two inherent problems of web data which are generally aggravated by its semantic enrichment.

1. **High dimensionality**: Web usage data is characterized by high dimension

(c.f., Fig. 1.1) as it is composed of quite heterogeneous and granular features such as URLs. The semantic enrichment of these URLs further amplifies the dimensionality of the underlying data. As a result, the mining process requires an excessive amount of storage space, is often very time-consuming, scales poorly in practice, and eventually encounters the inherent *curse of dimensionality* [4]. The curse of dimensionality is a term commonly used in data mining to express the fact that the complexity of many existing data mining algorithms is exponential in the number of dimensions. Therefore, with increasing dimensionality, these algorithms soon become computationally intractable and effectively impractical in many real applications.

2. **Sparsity**: For high-dimensional representations, the data distribution is usually sparse which means that the data points are located in different dimensional subspaces. In other words, as the number of both the web pages and users continues to grow, the likelihood that different users access common pages decreases. Therefore, eventually we have insufficient usage data as many web pages are not accessed by most users. This corresponds to many empty cells in the user-pageview matrix shown in Fig. 1.1. Friedman [23] discusses that as data sparsity increases, data points tend to become equidistant from one another. This phenomenon may render many data mining tasks (e.g., clustering) ineffective because the model becomes vulnerable to the presence of noise and fails to correctly capture the underlying structure of data.

This shows that along with introducing greater flexibility to the process of web usage mining the integration of domain knowledge also makes the process susceptible to two significant challenges, i.e., *quality* and *scalability*. However, it is possible to take advantage of this additional semantic information and use it to project the high-dimensional data to an aggregated lower-dimensional representation. Commonly known as *data generalization*, it is an effective data reduction technique which subsequently minimizes the effects of sparseness as well. Generalization is a data compression technique that helps remove excessive and non-relevant details and derives from a source dataset, a target dataset at a reduced scale such that the structural characteristics of the source data are maintained for a given application. Most known generalization techniques utilize a concept hierarchy to replace detailed concepts in the given dataset by their general concepts in an attempt to decrease the data dimensionality and increase the data density. Therefore, restricting mining tasks to concepts at a higher level of abstraction (in a concept hierarchy) is believed to help in improving not only the scalability of the mining process but also the quality of the patterns discovered. This improvement in quality is expected not only due to the possibility of identifying certain patterns which otherwise might be missed due to the insufficient usage, but also due to pruning away many patterns which were redundant and uninteresting.

In the context of web usage mining, the presence of millions of web pages that provide diverse information, impairs the ability of the mining algorithm to capture overlapping user interests which is essential for reflecting the trends of accessing the

website. For example, it is likely that many users are interested in "programming languages", but at a finer granularity they will either access pages on "Java", "C++", or "Ruby." This diversity of choice at the page-click level increases the sparsity of the web data making it difficult for the mining algorithm to identify interesting and meaningful usage patterns. The problem occurs when the discovered patterns do not include important items which may have occured less frequently in the usage data. It often happens that references to pages containing specific concepts occur far less frequently than pages containing general content [40]. For effective knowledge discovery, however, it is also important to capture those patterns that contain these less frequently used items. By aggregating the usage data and focusing on higher level concepts of such items, we can identify patterns of potential interest that were hidden in the originally sparse dataset. For example, Mobashar [16] suggests that although a movie site may not provide enough support for an association rule such as: "*If* Spiderman, Xmen *then* Xmen2", mining at a higher level may have enough usage support to capture a rule like: "*If* Sci-Fiction and Action, Xmen *then* Xmen2."

A variety of generalization based web usage mining techniques have been proposed in the literature (c.f. Section 2.4). As expected, they significantly decrease the dimensionality of the web data, thus reducing the computation time and space requirements. However, these improvements in scalability seem to come at the cost of the quality of patterns discovered. We noted that existing techniques have one characteristic in common, i.e., every concept is generalized to the same hierarchy level irrespective of its significance. At a macro level, this *significance* is defined as the

degree of interest of the web users in a concept. Since, concept hierarchies on web data usually have a high branching factor, desirable rates of data compression are only achieved by generalizing to higher levels in the hierarchy. As a result, some significant concepts get generalized in the process leading to a potential loss of patterns that existed in the original data, or extraction of false/misleading patterns. For example, suppose in a university site we find an association rule: "*If* GradStudent *then* GradStudentAssociation", suggesting that graduate students are interested in information about their association. However, if we mine at a higher level of abstraction where, {{GradStudent, UnderGradStudent} $\subset$ Student} we get a generalized rule: "*If* Student *then* GradStudentAssociation" which is misleading, since this pattern is mainly a characteristic of GradStudent. For this reason, it is desirable to develop a generalization technique that can retain the significant concepts while pruning away the unnecessary details so that we can improve both the scalability and accuracy of web usage mining.

1.1.2 Objectives and Proposed Methodology

Our work is based on the requirement to devise a flexible generalization technique, suitable for web data, that is able to achieve desirable data compressions without loosing the essence of the users' navigational behavior on the website. Previous studies [24, 51, 3, 57, 66] about generalization based web usage mining have observed or implied that the expected quality improvements in the patterns cannot be achieved at the desired rates of data compression. Our criticism of these techniques is that taking

a level-based cut through the given concept hierarchy results in reduced quality since either significant concepts get coalesced (*overgeneralization*) and/or insignificant ones are retained (*undergeneralization*).

In a more general sense, the existing techniques of web data generalization, to which we refer to as *level-driven generalization techniques*, suffer from a problem which has been studied in literature in the context of content-based personalization techniques [46]. These techniques were not effective because they relied on content similarity alone for making recommendations and missed more significant semantic relationships among objects such as *usage*. Similarly, merging similar items in a concept hierarchy as one pseudo-concept without taking into account the much relevant usage information makes the level-based generalization techniques less useful in the web usage mining domain where the goal is to capture and model user *behavioral* patterns.

In this thesis, we propose an extension to the existing techniques for generalization of web data by controlling the generalization of a concept based on its significance rather than its level in the hierarchy. For this, we estimate the significance of concepts in an intuitive way from a stratified sample of the large usage dataset. While this yields only an approximation of user interests (*i.e.*, usage), it is fast, as we only need to use a small sample, from the large dataset. Further, our experiments on a large weblog data have shown that an approximate usage estimate is sufficient to yield significant improvements in scalability and quality. A usage threshold is defined to control the

concept hierarchy pruning (interchangeably called *compression*) process , which allows to flexibly manipulate the trade-off between data compression and the quality of the mined patterns. Web data is generalized by merging the insignificant concepts into their higher level concepts along with an aggregation of significance at the generalized nodes. Numerous experiments and analysis of the mined profiles obtained from the generalized web data confirm that our technique can extract interesting and meaningful patterns from large and sparse weblog datasets, more efficiently.

1.2 Contributions

The thesis primarily focuses on the development of a web usage data generalization module, and evaluates its impact by incorporating it as an additional step in the conventional web usage mining framework. The aim of this generalization is to considerably reduce the large quantity of the web usage data available and, at the same time, to improve the quality of the subsequent pattern discovery. In summary, we are interested in developing a web usage mining process that serves towards fulfilling the much desired goals of a higher quality pattern discovery and scalability. The contributions of this research are as follows:

1. Development of an efficient method to estimate significance of concepts in a hierarchy based on a sample of the large usage data contained in the web access logs.

2. Introducing a parameter, called *usage threshold*, to control generalization of each

concept in the hierarchy. When the usage of a concept is less than the threshold, it is generalized to the next level in the hierarchy.

3. Development of a usage driven compression algorithm, which preserves the items of significance, possibly at different levels in a concept hierarchy.

4. A comprehensive experimental study comparing our usage-driven generalization technique with existing level-driven techniques.

To the best of our knowledge, we are the first to propose a usage-driven generalization of concept hierarchies underlying web usage data to facilitate better mining of user navigation patterns. We adapt and use a fuzzy clustering algorithm, which implements and tests the proposed idea.

1.3 Thesis Organization

The rest of the thesis is organized as follows. In Chapter 2, we provide background and review related work. In Chapter 3, we give a detailed description of our web data generalization technique. Chapter 4 presents experiments and results. Finally, chapter 5 provides concluding remarks and outlines some directions for future research.

Chapter 2

Background and Related Work

In this chapter, we describe the web usage mining process along with a review of the relevant knowledge discovery techniques. It also describes the notion of data generalization and explains the attribute oriented induction based approach to generalizing large datasets. The intent is to introduce the basic concepts and some notations that are going to be used in the rest of the thesis. A review of the related work is given in the end.

2.1 Web Usage Mining

Web Usage Mining is defined as the process of automatically discovering meaningful usage patterns from the web server access logs, using data mining techniques [12]. The discovered patterns are usually represented as a collection of pages, objects, or resources that are frequently accessed by groups of users with common needs and

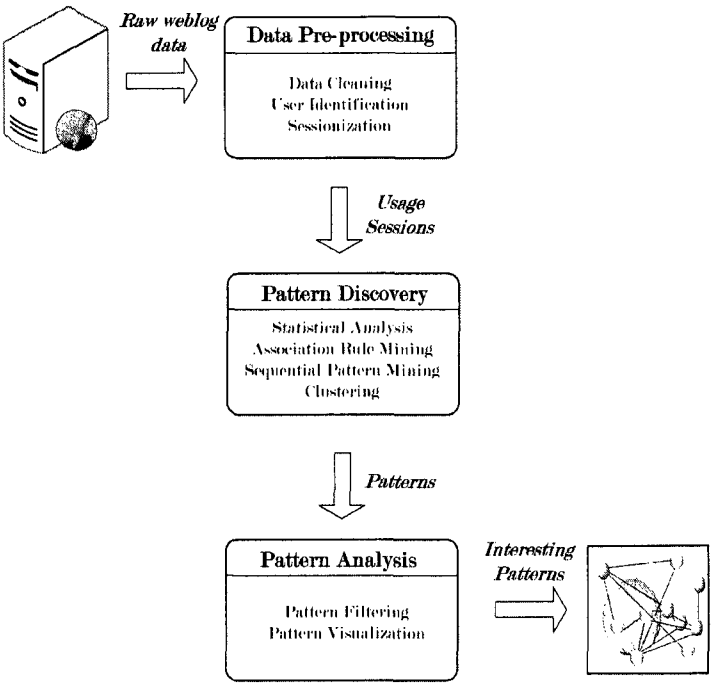


Figure 2.1: The Classical Web Usage Mining Process

interests. In essence, web usage mining allows the web-based organizations to gather interesting information about the users navigational behavior which can be later used to perform activities such as personalizing the web content, enhancing the system performance, understanding the nature of web traffic, determining effective marketing strategies, identifying potential customers for E-commerce related applications, developing adaptive websites, etc.

Similar to the standard data mining process, the overall web usage mining process is usually divided into three inter-dependent stages: data pre-processing, pattern discovery, and pattern analysis, as shown in Fig. 2.1. Next, we discuss the relevant concepts and techniques used in each stage:

2.1.1 Pre-Processing

Web server logs are the primary source of data for capturing the navigational behavior of web users. Moreover, data can also be collected at the client-level, proxy-level or obtained from an organization's database (i.e. business/operational data). Due to caching and network transmission times the information in web server logs may not be entirely reliable. In such situations, the use of navigation information available in the proxy servers and/or at the client side can contribute towards better quality and "completeness" of the usage information. The use of remote hosts (JavaScripts or java applets) or modified browsers at the client side, provided the client's corporate, can capture detailed information about user behaviors at the actual source. Similarly, the web proxy server, which acts as an intermediate level of caching between the client browsers and web servers, may serve as a good data source for characterizing the browsing behavior of a group of anonymous users sharing a common proxy server. For more details on the sources and types of data for web usage mining, see [12].

Web servers register a *log* entry for every click made by a user which keeps track of, among other things, the IP address from which the request is originated, the URL requested, and a timestamp. The following is a fragment of a typical log file from a web server, which is based on the Common Logfile Format [41]:

```
24.203.44.21 - - [31/Dec/2004:12:14:32 -0500]
''GET /help/homepage.html HTTP/1.1" 200 5459
```

Table 2.1 describes the different components of this standard format. It should be noted that the fields *logname* and *username* are usually not recorded.

Component	Description
remotehost	The remote hostname or IP address of the client
logname	A character string which identifies the identity of the user from a particular TCP connection
username	The name with which the user authenticates himself
date	Date and time when the request was made
request	The resource requested by the client specifying the method, file name, and the protocol of the request
status	The HTTP status code returned to the client
bytes	The size of the returned file

Table 2.1: The Common Logfile Format

Depending on the goals of modeling and analysis, the raw weblog data has to be cleaned, condensed, and transformed in order to retrieve and analyze significant and useful information. The process is briefly explained below. For details interested readers are referred to [15, 47].

Data cleaning results in the removal of useless log entries including references to multimedia data and scripts, and/or requests performed by robots and web spiders. Moreover, irrelevant fields like *status* and *bytes* may be removed as well. The next step is to group together the browsing activities belonging to the same user. *User Identification* is necessary since a user may visit a site more than once. For websites where a user authentication mechanism is not available, a simple heuristic is to identify each IP address as a unique user, although it is known that an IP address can be used by several users. Depending on the information available, other user identification heuristics can be used as suggested in [37]. Once the users are identified, the user activity record must be divided into sessions, which are segments of user activities

performed during individual visits of the user to the site from the moment of entering the site to time of leaving it. This is referred to as the *sessionization* process which attempts to reconstruct the real sequence of actions performed by a user during one visit to the site with the help of different heuristics. Generally, a timeout is used as the default method of breaking a user's activity record into consecutive sessions. Various heuristics based on parameters derived from time or the linkage structure of the website are discussed in [5] and evaluated for effectiveness in [59].

Based on the requirements of the mining application, the sessions may be viewed as either a *set* or a *sequence* of pages accessed by users. Many applications, such as market basket analysis and study of usage profiles, consider a session as a set of pages accessed by users irrespective of the order in which they were accessed. However, for applications where establishment of sequential or frequently browsed patterns is needed, each session is modeled and viewed as a sequence of web pages accessed. In this work, we consider sessions as sets not sequences.

At the end of the data pre-processing step, we obtain a set of N sessions, S = $\{s_1, s_2, s_3 \ldots s_N\}$. We use P = $\{p_1, p_2, p_3 \ldots p_M\}$, to denote the set of all pages, each of which is a URL (i.e., *Uniform Resource Locator*). Conceptually, a session s_i includes a subset of all pages requested by the same user in a "single" visit, i.e.,

$$s_i = (s_{i_{ID}}, \{\langle p_{k_{ID}}, w^i_{p_k}\rangle\}) \tag{2.1}$$

where $w^i_{p_k}$ is the weight associated with each page p_k in session s_i, representing its significance. Usually, this weight is either binary (i.e. representing the existence or

non-existence of a page in the session) or is a function of the actual time that the user spent on the page.

The pre-processing tasks ultimately result in a collection of user sessions, each corresponding to a delimited set of pages accessed by a user. However, as discussed in Chapter 1 recent research efforts have focused on integrating a variety of other data sources, such as web content, web structure and other semantic domain knowledge from site ontologies, with the pre-processed usage data at this stage to ensure effective pattern discovery.

2.1.2 Pattern Discovery and Analysis

In general, methods and algorithms in different fields such as statistics, data mining, machine learning and pattern recognition can be adopted and used to discover meaningful patterns of user navigational behavior from pre-processed usage data. The choice of the technique, however, depends on the goals and the desired outcomes. Major techniques, in the context of web usage mining domain, are discussed below [62]:

1. *Statistical Analysis*: Gains knowledge about visitor behavior by applying standard statistical techniques such as mean, median, frequency, etc. on various data items available in the log files such as requested resources, time duration, and domain.

2. *Association Rule Mining*: Finds groups of items or pages that are commonly

accessed together usually, but not exclusively, using a modification of the Apriori Algorithm [1].

3. *Sequential Patterns*: Allows the discovery of patterns of co-occurrence (of web pages), by incorporating the notion of time.

4. *Clustering*: Groups together pages or users that have similar characteristics based on the general idea of a distance function which computes the similarity between groups. This yields a usage model.

5. *Classification*: Assigns the users to predefined classes based on their characteristics. Classification requires extraction of features that have high discriminative ability as referred to the given classes or categories.

In this thesis we have focused on clustering techniques discussed in Section 2.2 in detail.

Once patterns are discovered, the final phase of the web usage mining process involves converting the discovered rules, patterns and statistics into *knowledge* (or insight) about the website being analyzed. The exact analysis methodology depends on the application for which the web mining is carried out. In most cases visualization techniques, such as graphs, are used to communicate the knowledge in a more convenient format to the human analysts and users.

2.2 Web Data Clustering

Web data clustering is the process of organizing web data into groups whose members are similar in some way. It not only allows to facilitate the accessibility of web information but is also used to improve the content delivery on the web.

Clustering of usage records (i.e., sessions), usually termed as *usage-based clustering*, is one of the most commonly used analysis tasks in web usage mining. It mainly consists of three steps:

1. *Representing sessions as vectors*: Since, in our context of usage modeling, the ordering of URLs accessed is not relevant in clustering, we can represent each session s_i as a binary vector over the M-dimensional space i.e., the number of the available pages: $s_i = \langle w^i_{p_1}, w^i_{p_2}, w^i_{p_3} \dots w^i_{p_M} \rangle$, where $w^i_{p_k}$ (see Formula 2.1) is 1 if p_k appears in the session s_i; otherwise $w^i_{p_k} = 0$.

2. *Computing similarities*: Measuring similarities among objects is a primary task in many data mining algorithms. In the context of web data clustering this involves computing strength of the relationship between the attributes of two usage sessions using measures such as cosine similarity, Euclidean distance, Pearson correlation, etc.

   Cosine similarity is a well-known and commonly used method to measure the similarity between two sessions[50]. It is computed by normalizing the dot product of the two corresponding session vectors s_i and s_j, with respect to their vector

norms. Formally,

$$cosinesim_{ij} = \frac{\sum_{k=1}^{M} s_{ik} s_{jk}}{\sqrt{\sum_{k=1}^{M} s_{ik}^2 \sum_{k=1}^{M} s_{jk}^2}} \tag{2.2}$$

3. *Clustering*: The final step is to employ a clustering technique, which is basically an unsupervised learning algorithm for discovering different usage profiles, each representing the interests or behavior of a significant *'interest group'* of users. There are a number of well-known clustering algorithms [69]; e.g., leader, k-means, hierarchical, fuzzy clustering, etc. The wide variety of clustering approaches proposed in the literature aim at solving problems in different application domains but their common objective is to determine the classes/clusters to which each session will be assigned.

   Given a dataset $S = \{s_1, s_2, s_3, \ldots, s_N\}$ consisting of N sessions, clustering attempts to partition S into C groups $\{c_1, c_2, c_3 \ldots, c_C\}$, such that the following conditions hold [69]:

   (a) $c_m \neq \emptyset, 1 \leq m \leq C$

   (b) $\bigcup_{m=1}^{C} c_m = S$

   The clustering results can be represented by a $(C \times N)$ partition matrix (also referred to as a *membership matrix*) $U = [u_{mi}]$, $m = 1, \ldots, C$ and $i = 1, \ldots, N$, where u_{mi} is the membership of the user session s_i in cluster m. For a *hard* (or crisp) clustering algorithm, where each session is assigned to a single cluster, $u_{mi} \in \{0, 1\}$. However, in a *fuzzy* clustering every session has a variable degree of membership to each of the output clusters, i.e., $u_{mi} \in [0, 1]$. Moreover, a

fuzzy clustering can be converted to a crisp clustering by assigning each session to the cluster with the largest membership grade.

The ultimate goal in clustering sessions is to provide the ability to analyze each segment for deriving business intelligence, or to use them for tasks such as personalization or recommendation. This requires the derivation of good quality and actionable usage profiles from the patterns obtained in the clustering process. A usage profile, which is a weighted collection of pages accessed in a cluster, helps in capturing an aggregate view of the navigational behavior of users with the same interest group.

More formally, given a cluster c_m, a usage profile is a set of URL-weight pairs, $UP_{c_m} = \{\langle p_k, up_{km} \rangle\}$, where up_{km} is the ratio of the sum of the weights of the k^{th} page across the sessions to the total number of sessions, in cluster m [48]:

$$up_{km} = \frac{\sum_{s_i \in c_m} w^i_{p_k}}{|c_m|} \tag{2.3}$$

where i is a session in cluster m, and $|c_m|$ is the total number of sessions in this cluster. More specifically, up_{km} is the support of URL k in cluster m. If the page weights in the original sessions are binary, then up_{km} is the percentage of sessions cluster m in which the page k occurs.

Further analysis and filtering of this aggregate representation of user behavior may be required to guarantee the reliability of the extracted knowledge. Experts in the relevant fields can then use this information for predictive modeling and in applications such as recommender systems.

In our work, we have used the Relational Subtractive Fuzzy Clustering algorithm

[65] which is discussed below in detail.

2.2.1 Relational Fuzzy Subtractive Clustering

The Relational Fuzzy Subtractive Clustering (RFSC) algorithm, proposed by Suryavanshi *et al.* [65], is a relational data clustering technique for partitioning web access logs into soft classes capturing the fuzziness inherent in the web log data.

Relational data describes objects by specifying pairwise dissimilarities (or similarities) between them [33]. Being a relational clustering algorithm the input to RFSC is a collection of sessions represented by a matrix R, where R_{ij} is the dissimilarity between the i^{th} and j^{th} sessions. It also holds that $0 \leq R_{ij} \leq 1$, $R_{ij} = R_{ji}$, and $R_{ii} = 0$.

RFSC is based on the subtractive clustering algorithm, proposed by Chiu [10], which is an efficient algorithm for estimating the number and locations of the clusters. Extending the idea in [10], RFSC starts by computing the potential for each session based on its dissimilarity to all other sessions in the dataset. A session with many similar sessions will have a high potential value. The potential of a session s_i is calculated using the formula:

$$P_i = \sum_{j=1}^{N} e^{-\alpha R_{ij}^2} \tag{2.4}$$

where R_{ij} is the dissimilarity between sessions s_i and s_j, N is the total number of sessions to be clustered, and $\alpha = 4/\gamma^2$. Here, γ is essentially the neighborhood-dissimilarity value calculated from R. The dissimilarity(γ_i) of the i^{th} session is defined

as the median of dissimilarity values of session i to all other sessions. The median of all γ_i's, then forms the dissimilarity for the entire dataset, denoted by γ, which is a value in the range [0,1].

Once the potential of every session is computed, the session with the highest potential P_1^* is selected as the first cluster center. This is followed by a series of subtractive steps which start by reducing the potential of each session proportional to the degree of similarity with the cluster center found in the preceding step. The sessions which are highly similar to the previous cluster center will have a higher degree of reduction in their potentials, and thus are unlikely to be selected as the next cluster center. Each subtractive step identifies the next candidate cluster center, i.e. the session with the highest potential P_t after subtraction. This process of acquiring new cluster centers and revising potentials repeats until $P_t < \underline{\in} P_1^*$, where $\underline{\in}$, called *reject ratio*, is a threshold below which a candidate cluster center s_t is definitely rejected. The condition for termination/acceptence is relaxed by introducing an additional threshold $\overline{\in}$, called the *accept ratio*. If $P_t > \overline{\in} P_1^*$, then s_t is definitely selected as the next cluster center, and this is followed by the next subtractive step. However, if the potential falls in the gray region, i.e., $\underline{\in} P_1^* < P_t < \overline{\in} P_1^*$, a check is performed to see if acceptance of the cluster center provides a good trade-off between having a sufficient potential and being sufficiently far from the existing cluster centers.

When the subtractive step terminates, RFSC yields a clustering of the form $\{s_{c_1}, s_{c_2}, \cdots, s_{c_C}\}$, where s_{c_m} is a user session serving as the m^{th} cluster center. The membership u_{mj} of the j^{th} session in the dataset [1 ... N] with respect to the m^{th}

cluster center is calculated using the formula:

$$u_{mj} = e^{-\alpha R^2_{cmj}} \tag{2.5}$$

where α is defined by Eq. 2.4 and R_{c_mj} is the dissimilarity between sessions s_{c_m} and s_j. Sessions that are close to each other have high memberships in the same clusters.

RFSC uses fuzzy techniques for web data clustering which is preferable since usage, categories and associations in web mining do not have crisp boundaries. A web user can belong to multiple interest groups with some degree of membership to each group. For instance, he/she might access the same page for different purposes in different sessions or can also have conflicting sub-goals in the same session. This inherent ambiguity and fuzziness of the underlying data can be better represented using overlapping (or soft) clusters, i.e., founded on fuzzy sets [71] to reflect *degrees* of user interests in the different classes.

2.2.2 Cluster Evaluation

After a set of clusters is found, we need to assess the goodness of the clusters. Clustering evaluation is a very difficult problem. Clustering is an unsupervised process since there are no predefined classes and no examples that would indicate grouping properties in the dataset, therefore it is difficult to find an appropriate metric for measuring if the derived cluster configuration is acceptable or not. We discuss the two commonly used evaluation methods below:

1. **User Inspection**: Human inspection on the clustering output may be the most

intuitive clustering validation method as it compares the clustering result with the user's intention in a natural way. It refers to the subjective process of manual inspection of the resulting clusters by experts on the data. This is obviously a labor intensive and time consuming task. However, in most applications, some level of manual inspection is necessary because no other existing evaluation methods are able to guarantee the quality of the final clusters.

2. **Quantitative Measures**: In these methods, internal information in the clusters is used to evaluate the clustering results. Two measurement criteria have been used for evaluating and selecting an optimal clustering scheme [39]:

   - **Compactness** (intra-cluster cohesion): This measures how near the data points in a cluster are to to the cluster centroid. A common measure of compactness is the variance.
   - **Separation** (inter-cluster isolation): This measures how far apart are different clusters from one another. There are three common approaches measuring the distance between two different clusters: distance between the closest member of the clusters, distance between the most distant members and distance between the centers of the clusters. Any distance functions can be used for the purpose.

Using these quantitative measures, the objective of clustering is defined as follows: to minimize the distances among the data points in individual clusters and to maximize the distances between the clusters. We should note, however,

that good values for these measurements do not always mean good clusters. In most applications, expert judgements are still the key.

2.3 Data Generalization

Data generalization is a process of grouping data by transforming concrete item sets into more abstract (high-level) conceptual representations based on similarity [26]. It is a form of *descriptive data mining*, which aims at identifying general patterns or properties that lie within the massive set of task-relevant data. The volume of data and human user's inability to comprehend the large sets of data have driven the development of such data analysis techniques that provide concise and summarized information at multiple levels of abstraction.

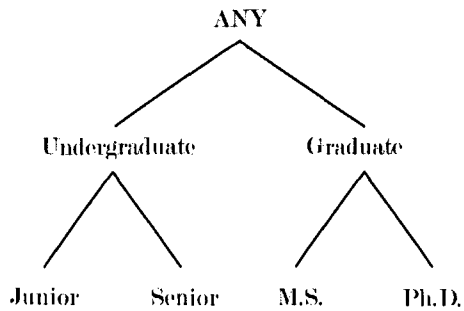
Data generalization can be implemented using data cube (OLAP-based) approach [26] and the attribute-oriented induction (AOI) approach [7, 27, 28]. The data cube approach is essentially based on materialized views (called "data cube") of the data, which are typically pre-computed and stored in a data warehouse. Generalization is performed using a roll-up operation which reduces the number of dimensions in the data cube and captures an aggregate view of the underlying data. However, data cube computations are challenging as they are resource intensive, requiring ample computational time and storage space. Furthermore, it is a user-controlled process which confines its applications to simpler data analysis problems, in general. On the

other hand, the AOI approach which was introduced by Cai *et al.* [7] a few years before the data cube approach is relatively a more automated process that provides the flexibility to determine the degree of generalization in order to produce an interesting and feasible summarization of the data. We next discuss the AOI approach which forms the basis of our generalization technique.

2.3.1 Attribute Oriented Induction

AOI is an iterative process which generalizes the values of tuples in a relation to their corresponding higher level concepts, and subsequently merges the tuples having the same generalized value into a generalized tuple [26]. This generalized tuple typically reflects some common characteristics of the original tuples from which it is generalized.

In essence, AOI is a data summarization technique that integrates the generalization rule known as *concept ascension* which is defined as climbing up a concept hierarchy. In the AOI algorithm [7], each attribute to be generalized is associated with a concept hierarchy [29] that is provided by knowledge engineers, domain experts or users prior to the process. A concept hierarchy is represented as a tree (i.e., a multi level taxonomy) in which concepts are ordered in a general-to-specific ordering. Fig. 2.2(a) shows a concept hierarchy for a categorical attribute-student *status*. The root node corresponds to the most general domain value, whereas the leaf nodes correspond to the most specific values in the database. The hierarchical character of induction permits gradual, similarity-based, aggregation of attribute values stored in the original tuples. The choice of concepts has a fundamental influence on the



(a) A categorical concept hierarchy for the attribute *status*

name	*major*	*status*
Bach	Math	Junior
Fraser	Physics	M.S.
Gupta	Math	Ph.D.
Jackson	Computing	Senior
Monk	Computing	Ph.D.

major	*status*	*count*
Science	Undergraduate	2
Science	Graduate	3

(b) An instance of relation *student* and it's generalization

Figure 2.2: An Example Concept Hierarchy and Attribute Oriented Induction

retrieved results. For details on how to choose concept hierarchies, interested readers are referred to [29].

AOI can be effectively utilized as an initial step in the data mining process followed by further knowledge extraction from the generalized data. This may lead to the discovery of patterns that provide a general description of the original data. These general patterns are not only easier to understand but most often are stronger, (i.e., have higher *support*) as well as interesting/meaningful. Besides this, it is an effective data reduction technique since it compresses the raw data without totally omitting even rare attribute values from the data. Although each higher level in the hierarchy has a smaller number of descriptors, their broad character captures the general meaning of domain values from the lower abstraction levels. Furthermore, to preserve the original dependencies among the data, each generalized tuple is associated with an added attribute, called *count*, which keeps track of the number of objects

represented by the abstract object. This count is accumulated when merging identical objects in generalization. Other popular aggregate functions (e.g., *sum* or *average*) can also be associated with the generalized tuples given there are numeric attributes in the working relation.

The generalization process is composed of a sequence of (iterative) replacements of attribute values with their general descriptors present at the next abstraction level in the concept hierarchy, followed by an accumulation of the merged tuples to keep track of the original values which are now characterized by a particular abstract concept. The concept ascension continues until a reasonable level in the concept hierarchy is attained. Two parameters, the attribute generalization threshold and the generalized relation threshold, can be used to define a *desired level* for generalization. The former threshold regulates the maximum number of distinct values that are allowed in an attribute after generalization, and the second threshold regulates the maximum number of tuples. If, at the end of an iteration, either of the thresholds is not satisfied, the process terminates. An alternative termination condition is to define a level up to which objects are merged based on their degree of similarity to each other. Angryk *et al.* [2] refer to this level as an α-cut level, where α is the similarity between the values of an attribute. This can be used when there is a similarity relation defined on the attribute which needs to be generalized. The control of how high an attribute should be generalized, irrespective of the approach, is typically quite subjective. If the attribute is generalized "too high", it may lead to *overgeneralization*, and the resulting patters may be erroneous (not reflecting reality). On the other hand, if the

attribute is generalized "too low", then *undergeneralization* may result, and the patterns obtained may not be informative either. Therefore, the termination condition has to be applied with caution in order to attain a desirable balance in AOI.

Let us consider an example student relation [27] and the conceptual hierarchy shown in Fig. 2.2(a). Fig. 2.2(b) shows the AOI based generalization of student relation using this concept hierarchy and an attribute generalization threshold value of 3. The value of count for each tuple depicts the number of tuples in the original relation generalized after applying AOI.

2.4 Related Work

Web usage mining has been a topic of extensive research in the recent years. A number of data mining techniques are proposed in order to efficiently model user behavior [43, 42, 70, 68, 38, 65]. However, as discussed in Chapter 1, the success of such techniques in truly discovering the usage model cannot be demonstrated, due to insufficient usage information at the page-click level.

In the following sections, we review the related work in the domain of semantically enriched web usage mining focusing on the content characterization using concept hierarchies. We also provide a quick overview of the generalization techniques used for data analysis. Finally, we discuss the conventional techniques of generalizing web usage data in detail.

2.4.1 Concept Hierarchies and Web Usage Mining

A number of recent studies have shown the usefulness of exploiting semantics for mining. The mining process is enhanced by mapping the navigational data to either content features [22] or to concepts within an ontology [53]. Eirinaki *et al.* [19] suggest to characterize web content using a concept hierarchy (*i.e.* a taxonomy), in order to bring uniformity in the semantic enrichment of web data we need

Concept hierarchies have been considered extensively in the contexts of data mining, data warehousing, and other areas and their incorporation has improved research results and produced useful systems. Harinarayan *et al.* [32] improved the performance of OLAP operations using concept hierarchies to express dependencies among views. Focusing on text databases, Chakrabarti *et al.* [9] demonstrated that taxonomies provide a means for designing enhanced searching, browsing and filtering systems. Moreover, concept hierarchies have been used by McCallum *et al.* [45] to improve the accuracy and scalability of classification algorithms, and by Han *et al.* [30] for the discovery of interesting and strong association rules in large databases. Concept hierarchies have also been effectively applied in other areas including information retrieval to allow web users to formulate more expressive queries [55], and in language processing for effective document indexing [25].

A number of recent studies have discussed the usefulness of exploiting concept hierarchies for mining meaningful and interesting usage patterns. Oberle *et al.* [53] have used this as a vehicle to incorporate semantics into the mining process by mapping

URLs to concept labels reflecting application events. This enhances both visibility of the mined patterns and their usefulness to the users. Eirinaki *et al.* [19] extended this framework by using the mined profiles for web personalization and noted that the patterns found produced a broader yet semantically focused set of recommendations.

Nasraoui *et al.* [51] intuitively identifies two kinds of concept hierarchies that can be used in a web usage mining framework. An *implicit* concept hierarchy is available in the form of the website structure (i.e., *a page hierarchy*) and can be exploited for computing the similarity between two web pages [50]. The explicit concept hierarchies, such as product categories, are either hand-crafted by domain experts, or automatically created through a variety of machine learning techniques, such as agglomerative clustering [64] or association rule mining on the feature space to identify composite features [11].

In our work we exploit the implicit concept hierarchy derived from the website's directory structure. Different studies have shown that there is a lot of effort, manual or automated, involved in constructing the explicit concept hierarchies [35, 56]. Therefore, if the web pages are represented by meaningful URL prefixes and the website design reflects the underlying semantics, a page hierarchy is simple to use and is a preferable choice. Moreover, it is generally readily available for many websites.

2.4.2 Applications of Attribute Oriented Induction

The incorporation of concept hierarchies into AOI [7] has by far been the most significant formal use of this background knowledge in data mining which has lead AOI

to become one of the most popular class description methods.

Many studies in the domain of data mining have demonstrated that abstracting raw data to a higher conceptual level, and discovering and expressing knowledge at higher abstraction levels have superior advantages over data mining at a primitive level. Han *et al.* [31] have resolved the semantic heterogeneity problem, in building of cooperative information systems, by mapping low level heterogeneous data to high level homogenous data. Srikant *et al.* [60] have performed association rule mining across different levels of item taxonomies. They show that the generalized association rules are valuable as they include many interesting rules that had poor support at lower levels. On the other hand, many redundant rules are pruned away. Later, this work was extended for generalized sequential pattern mining in [61]. AOI has also been effectively applied in various other fields of data analysis, including geographical information systems for discovering associations rules between geographic data and non-geographic data [20], and multimedia data mining for analyzing patterns from multimedia data [72].

All these works are good examples for arguing in favor of the application of generalization techniques in the domain of web mining, where diversity of usage due to insufficient information is a challenging problem. The next section looks at existing techniques of generalization based web usage mining.

2.4.3 Level-Driven Generalization Techniques

With the emerging techniques related to semantically enriched web usage mining and the explosive growth of the web, usage data has become an ideal target for effective data reduction techniques. In the recent years, some researchers have been studying effectiveness of data generalization techniques for faster mining and reducing the sparsity of data. An overview of the work in this context is as follows.

Fu *et al.* [24] proposed the idea of generalizing web session data and integrated this with a clustering algorithm to find and analyze web access patterns. For this, they categorize the web pages in a page hierarchy created automatically via URL tokenization. This categorized representation is used to generalize the session data by replacing actual pages accessed with general URLs appearing as higher level concepts in the hierarchy. For example, a page like `/programs/ugrad/cs/` is replaced by `/programs/ugrad/` or `/programs/` depending on a pre-determined generalization level. This generalization level, which is critical to both the efficiency and effectiveness of the approach, is a user-specified parameter. The authors do not specify any automated method for estimating this parameter, but only suggested to try several levels, which is not a feasible option, especially if dealing with huge data sizes. The generalized web sessions are later clustered using a hierarchical clustering algorithm. The approach is tested for scalability and it is shown that the generalization of sessions greatly reduces the dimensionality but there is no quantitative evaluation of the validity and/or quality of the clustering results.

Simpler versions of the above-mentioned technique are applied in [57, 3]. They have only used pages at the level right below the root as categories to the remaining URLs. Rossi *et al.* [57] propose a method to cluster these categories of pages based on similarity of their usage. They provide a visual analysis of the clustering result and find some significant high level patterns. On the other hand, Banerjee *et al.* [3] use these categories to transform the raw Longest Common Subsequence (LCS) paths of user navigation into concept-based LCS path, which are later clustered based on the navigation time information using a graph-partitioning algorithm called Metis. They report some anecdotal evidence of effectiveness but suggest that generalization to a lower level will further enhance the quality of user clusters, provided additional information complexity can be managed. Both of these papers have not considered generalization with different compression factors (at lower generalization levels), and thus fail to provide an evaluation of the impact of generalizing to other levels of the hierarchy.

Nasraoui *et al.* [51] study the impact of data compression on quality of knowledge discovery by incorporating simple cues from the page hierarchy into the mining process. They parameterize the generalization process by defining a similarity threshold. Pages are recursively merged into their ancestor URLs as long as their content similarity with the general page is above the defined threshold. The metric used for similarity calculation is a measure of the amount of overlap in the paths of two URLs. Therefore, for a particular threshold value the roll-up is consistent for all the navigation trails, and hence this yields a level based cut of the hierarchy. For example, if

the depth (*i.e.* number of levels) of the given page hierarchy is 4 and the similarity threshold suggests to merge URLs that are 50% similar, compression is performed by merging all URLs that share a common prefix path into general URLs at the second level of the hierarchy. They have formally evaluated the impact of varying the threshold on the quality of the discovered patterns. Precision, coverage and F1 measures used, confirm the negative impact of higher data compression on the quality of the mined profiles.

There are also reports of generalization-based web usage mining using explicit concept hierarchies (c.f. Section 2.4.1. Tanasa *et al.* [66] proposed the idea of 'semantic generalization' where they map pages to manually identified semantic topics. The mapping is done only for the first and second level of the page hierarchy due to limited resources. For example, all pages under `/authors/index.html` would be classified as '*peoples*' pages. The page accesses in each user session are then described using these page categorizations. Generalization is incorporated as an advanced step in the data preprocessing phase. This study primarily focuses on the data preparation tasks and does not discuss the impact of generalization on the pattern discovery. Similarly, Dai *et al.* [16] discuss the usefulness of integrating domain knowledge in the pattern discovery phase and suggest to focus on higher level concepts in the concept hierarchy to improve the scalability issues that can be endemic at this stage. They do not indulge into the details of the impact of such aggregations on the quality of the mined patterns. We have observed that while using the explicit concept hierarchies, the mining algorithms have to perform relatively more complex similarity

computations. However, the idea of concept ascension is similar in all kinds of concept hierarchies. Therefore, our proposed technique is applicable to explicit concept hierarchies, given there are efficient algorithms to perform the underlying computational tasks. However, the explicit concept hierarchies are outside the scope of this thesis and in the remaining part of the thesis we will only focus on the page hierarchies. Readers interested in details of mining with explicit hierarchies are referred to [16].

2.4.4 Discussion and Perspective

In each of the techniques of session generalization discussed above, a generalization level is either explicitly specified by the user or implied by a threshold. It is also noted that typically the generalization levels are kept high, closer to the root of the hierarchy, in order to achieve desirable data compression. Since the number of concepts increase exponentially in the hierarchy as the level increases, a lower level of generalization may neither give desirable data compression nor would it significantly reduce the data sparsity. Nasraoui *et al.* [51] have suggested that similarity thresholds for generalization be chosen such that the process achieves a URL compression of 90%. For hierarchies which are deep and have a high branching factor, such a compression ratio is generally achieved only at the top most levels. However, the higher the level, the greater the risk of truncating some significant portions of the hierarchy. As already mentioned, this overgeneralization may have a negative impact on the quality of the profiles discovered for loosing patterns to which the truncated concepts might have belonged, and also generating some false patterns due to the misrepresentation of

information. Therefore, instead of rolling-up a fixed number of hierarchy levels, the need is to prune away the infrequently accessed areas of the page hierarchy. In the next chapter we introduce our technique of usage driven generalization and show its effectiveness to address the issues.

Chapter 3

Usage Driven Generalization of Web Sessions

The goal of generalization based web usage mining is to prune away the unnecessary items from the high dimensional data space, which introduces sparsity in the underlying data. This in turn effectively impairs the ability of the mining algorithms to capture meaningful patterns. For example, Strehl [63] suggests that when the data becomes extremely sparse, data points located in different dimensions can be considered as all equally distanced, making the similarity measure, which is essential for a clustering technique, meaningless.

The usage driven approach to generalizing web data is motivated by the inverse power law properties found in the user navigation behavior [34]. Basically, a page hierarchy is a collection of trails that users follow to satisfy their goals. Levene *et al.* [40] demonstrate that the probability of accessing a URL decreases as we descend in a

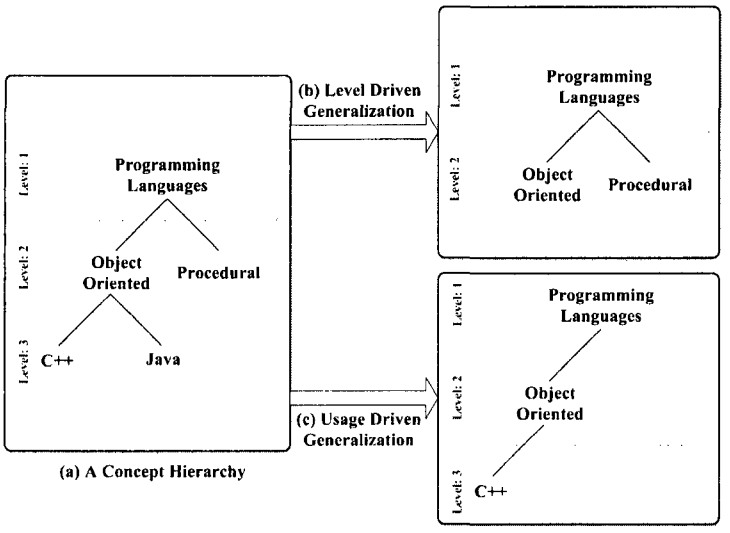


Figure 3.1: Approaches to Generalizing a Concept Hierarchy

trail. Our belief is that merging such infrequently accessed URLs into a general URL appearing at a higher level in the hierarchy and using aggregate access frequencies of such merged groups would reduce both the dimensionality and sparseness of the data. This should not only make the knowledge extraction process efficient but also improve the quality of the patterns discovered. However, as discussed in Section 2.4.3, existing techniques of web data generalization are unable to achieve the expected quality improvement at desired rates of data compression. This is primarily due to generalizing all the trails to the same higher level of the page hierarchy, irrespective of the availability of usage information for the truncated URLs. The following example illustrates this point.

Example: Fig. 3.1(a) illustrates a concept hierarchy. Assuming each node in this hierarchy is a web page and most visitors followed the path `Programming Languages` to `Object Oriented` to `C++`. As shown in Fig. 3.1(b) the existing *level*

driven web data generalization techniques, in this situation, will generalize a dense item like 'C++.' In most cases, this either causes loss of patterns that existed in the original data, or extracts false or misleading patterns. Another related problem is the inability to alleviate the sparsity of some trails even after generalization, e.g., an insignificant item 'Procedural' is retained. As mentioned earlier, mining algorithms extract uninteresting and redundant patterns if the underlying data is sparse. On the contrary, with usage-driven generalization, we can identify frequently accessed portions of the navigation trails and have the generalization process preserve the corresponding URLs for subsequent participation in the pattern discovery process. As shown in Fig. 3.1(c), each trail is trimmed from lower levels to a higher level where the aggregate access frequency of the merged URLs qualify the remaining trail for retention in the compressed hierarchy. Therefore, not all trails are generalized to the same level in the concept hierarchy.

As the first attempt to utilize the rich information contained in the weblog data for web data generalization, our research paves the way for further improving the accuracy and scalability of existing web usage mining systems. This chapter presents the revised web usage mining framework incorporating data generalization as an additional module. Following that, we discuss the proposed data generalization methodology in detail.

3.1 Generalization of Web Usage Data

As discussed in Section 2.1, the classical web usage mining process consists of three stages [62]: *data preprocessing*, *pattern discovery*, and *pattern analysis*. Traditionally, web data generalization has been performed implicitly as part of the data preprocessing [24]. Later, Tanasa *et al.* [66] considered this as an advanced step in the data preprocessing phase. In this thesis, we propose our usage driven generalization as an intermediary step between data pre-processing and pattern discovery, called *data generalization*. It is separated from data pre-processing because usage information, which drives the generalization of URLs, is extracted from the pre-processed user sessions using data mining techniques. Fig. 3.2 illustrates the extended four step web usage mining process. Usage driven data generalization comprises of three steps: *usage estimation*, *URL compression*, and *session transformation*. The idea of compressing the page hierarchy and subsequently transforming the session data is identical to all web data generalization techniques. However, in our usage driven generalization, instead of horizontally cutting the page hierarchy at a particular level, we propose to prune away the insignificant portions explicitly based on the estimated usage information.

At the end of the data pre-processing stage, we have the following set of information:

1. ***User sessions***, which consists of the page requests originating from the same IP address within a pre-defined time-out (c.f. Section 2.1.1).

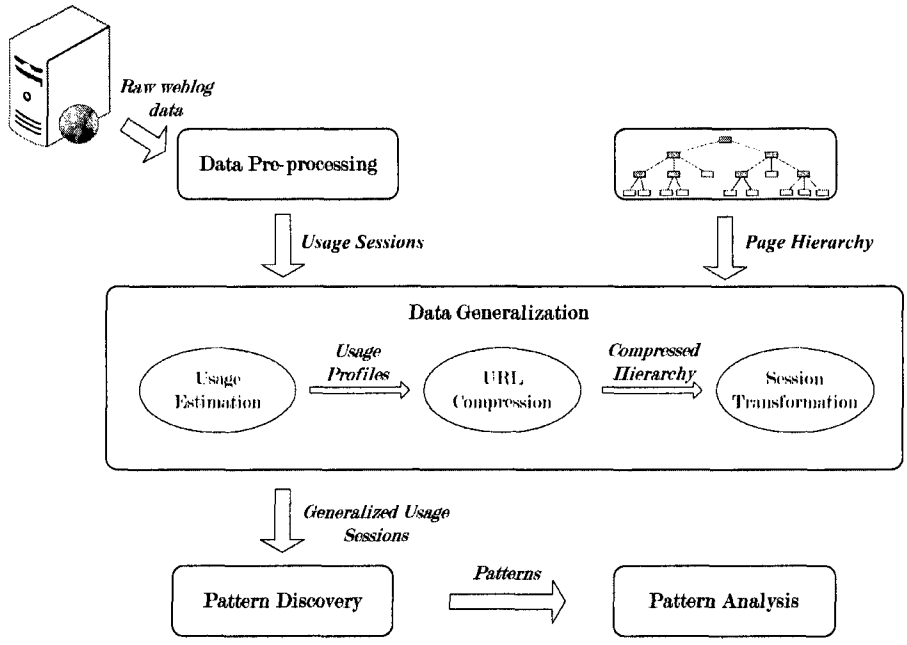


Figure 3.2: The Extended Web Usage Mining Framework Driven by Usage-Based Generalization of Web Sessions

2. The ***page hierarchy***, which is a reflection of the overall content of the website at different levels of abstraction. Fig. 3.3 provides a snapshot of the page hierarchy consisting of some pages present in the web server of a computer science and software engineering department at a university. Each node, represented by a URL, reflects a concept. The hierarchy is arranged in a *general-specific* ordering with the root, which is the home page of the website, serving as the most general concept. These nodes participate in an *ancestor-descendant* relationship. The preceding URLs in the path from the root to URL j, are called the ancestors of j. Similarly, the succeeding URLs which can be reached from a URL k, are called the descendants of k. If semantics form the basis of the page organization, than each node abstracts the content of its descendent nodes. Under this assumption,

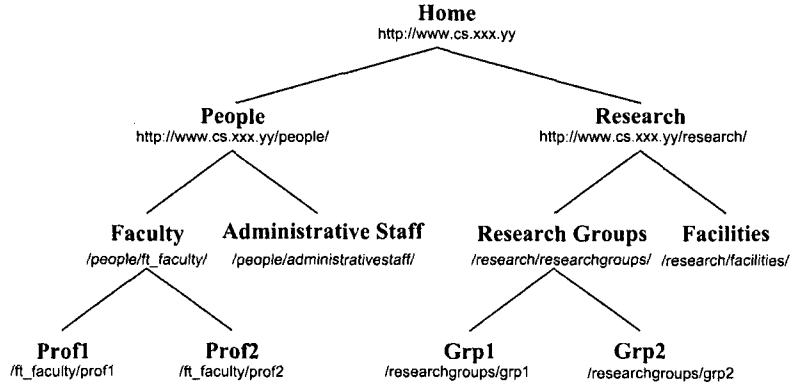


Figure 3.3: An Example Page Hierarchy

the general intent of users can be identified by looking at the ancestor URLs of the actual user clicks in the session data. Fu *et al.* [24] provide a simple method of constructing a page hierarchy using the URLs of the pages.

This information forms the input to the 'data generalization' stage (c.f. Fig. 3.2). The following sections provide a detailed description of our usage-driven generalization methodology, and explain the relevant concepts and techniques applied in its various stages.

3.2 Proposed Methodology

Web logs are the most commonly used implicit source of user behavioral data. Over the years, they have been used to derive sets of user-centric data models (i.e., user profiles) that represent the activities and interests of all the users of a website. In this work, we exploit information contained in the weblog data to derive a generalization scheme which reduces the dimensionality and sparsity of the data available to the

web usage mining systems. We show how this resolves the drawbacks (c.f. Section 2.4.4) of existing web data generalization techniques. The following sections provide a detailed description of the usage-driven web data generalization methodology.

3.2.1 Usage Estimation

In this section, we propose an intuitive method of efficiently estimating the usage of URLs (i.e., nodes in the page hierarchies). We use this information to guide the process of compressing the concept hierarchy. Since the weblog data is very large, usage computation with the entire weblog would take too much computation time and space. Instead, we extract a stratified random sample of the user sessions, which is the main source of usage information. The reason behind this is twofold. Firstly, using a sample makes sense since we are primarily interested in identifying pages that occur *frequently* in the sessions. A sample can help capture this information quickly and precisely as it is very likely that a popular URL in the original dataset, will also be present in a representative subset of the data. Secondly, the use of stratification is motivated due to the seasonal properties inherent in the weblogs. Generally web usage data is collected over an extended period of time t, whereas the popularity of pages is subject to change over t. A small non-stratified random sample is likely to be less representative of usage than a stratified sample of *time ordered* sessions.

The process of usage estimation begins with ordering the session data according to relative time of page access. This ordered dataset is then divided into time frames and sessions are selected randomly from each of these frames. This allows us to

capture the desired information even with relatively small sample sizes.

Once a sample has been selected, any appropriate data mining technique can be applied to capture the URL usage. However, we estimate this usage by analyzing the Usage Profiles obtained from clustering of the sample data. Clustering is preferred over other techniques since in addition to the added advantage of removing noisy or irrelevant data, it helps identify popular URLs among user groups based on their interests. It is such items of user interest that need to be preserved in the compressed hierarchy. It is important to note that this initial clustering is performed with the purpose of estimating the usage of URLs and using this information to improve the actual knowledge discovery.

The clustering of the sample dataset results in a set of URL-weight pairs called usage profiles (c.f. Section 2.2). As a result, every URL k in the page hierarchy is associated with a list of C_d components,

$$UP_k = (up_{k1}, \cdots, up_{kC_d}) \tag{3.1}$$

where C_d is the number of clusters obtained from the sample clustering and up_{km} is the significance of URL k in cluster m (c.f. Eq. 2.3).

The Usage Threshold

A *usage threshold*, denoted as μ, regulates the minimum usage value which will qualify a URL along with its ancestral path to be retained in the compressed hierarchy. In other words, it will control the number of levels that can be pruned in each navigation trail of the hierarchy.

The following parameters need to be considered while selecting a usage threshold:

1. The average size of clusters (denoted by $|c^d|_{avg}$) of the sample dataset, defined as $\frac{d}{C_d}$, where d is the size of the sample dataset and C_d is the number of clusters.

2. The sparsity level of the sample dataset, defined as $1 - \frac{|s^d|_{avg}}{N}$, where $|s^d|_{avg}$ is the average session length, i.e., the average number of nonzero items in the sample session vectors and N is the total number of URLs.

Using these parameters, μ can be expressed as $\frac{w_{min}}{|c^d|_{avg}}$, where $w_{min} \in (0, |c^d|_{avg}]$ is the minimum weight that a URL k should have in a cluster of uniform size, to be preserved in the compressed hierarchy. When sessions are binary vectors, this is defined as the minimum number of sessions in a uniform cluster. The choice of w_{min} depends on the sparsity of the sample dataset. As the sparsity increases, the weights of URLs decrease in the clusters, i.e., fewer sessions access a URL (for binary weights). Consequently, w_{min} should approach 0. This implies that even if a URL is accessed by a small percentage of sessions, its presence in a user interest group makes it significant enough to be retained. In the next chapter, we study the impact of different threshold values on the quality of the patterns discovered.

3.2.2 URL Compression

The usage driven URL compression is an iterative process. Given a usage threshold $\mu \in (0, 1]$, and the list of usage profiles, UP_k for each URL k, compression begins at

the leaf nodes of the hierarchy. A URL k is merged into its parent k_a if there is no cluster m such that $up_{km} \geq \mu$. The usage profile list of the parent URL k_a is updated to capture this increase in the significance of node k_a (c.f. Eq. 2.3).

$$up_{k_a m} = up_{k_a m} + \sum_{k \in desc(k_a)} up_{km} \tag{3.2}$$

where $up_{k_a m}$ is the aggregate usage profile of a general URL k_a for cluster m and $desc(k_a)$ returns the descendants of URL k_a which satisfy the generalization condition.

The compression continues upwards for each navigation trail in the hierarchy. A trail discontinues to participate in the compression if any of the following conditions holds:

1. A URL's usage estimate is above the usage threshold. This qualifies both the URL and its ancestral path to be retained in the compressed hierarchy.

2. Compression for a trail reaches the second level of the hierarchy. It is not recommended to continue compression to the root as this will loose the semantic boundaries of the web data, altogether.

A compressed list of URL pairs of the form $\{\langle k, k_a \rangle\}$, is constructed and maintained during the process, where k_a is an ancestor of URL k. This is used to guide the transformation of session data where the actual page-clicks are replaced by their general concepts.

Our URL compression algorithm is described as follows, in which + denotes the merge operation, χ is the set of nodes in the page hierarchy at certain level (initially

set to include the leaves), and parent(k_a) returns the immediate parent node of a URL k_a. This set is regenerated at each level as we roll-up the hierarchy.

Algorithm: URL Compression
Input: A page hierarchy, C usage profiles from sample clustering, and the usage threshold μ
Output: γ: A list of compressed URL pairs

D = the maximum depth of the page hierarchy
χ = a set of "leaf" nodes at level D

do while $D > 2 \wedge \chi \neq \phi$
  for each URL k in χ
    if $up_{kl} < \mu$, $\forall l = 1..C$
      Update γ by replacing k with $parent(k)$
      Add the pair $(k, parent(k))$ to γ
      Update $up_{parent(k)l} =$
        $up_{parent(k)l} + up_{kl}$, $\forall l = 1..C$
  decrement D by 1 and regenerate χ

To illustrate the steps of this algorithm, consider the page hierarchy in Fig. 3.3. A level driven generalization of this hierarchy, under the settings of level = 2 and 3, generates compressed hierarchies shown in Fig. 3.4(a) and 3.4(b). (We assume that the root is at level 1 and the levels increase as we move from top to bottom). Suppose Table 3.1 captures the usage profile obtained from the clustering of a sample session data for the given page hierarchy. Assuming a usage threshold $\mu = 0.3$, the leaf nodes {`Administrative Staff`, `Facilities`, `Prof2`, `Grp1`, `Grp2`} satisfy the compression condition and thus are merged into their corresponding parent URLs. However, {`Prof1`} has usage above the threshold in Cluster 1. This qualifies the path `Home` to `People` to `Faculty` to `Prof1` to be frequently accessed and

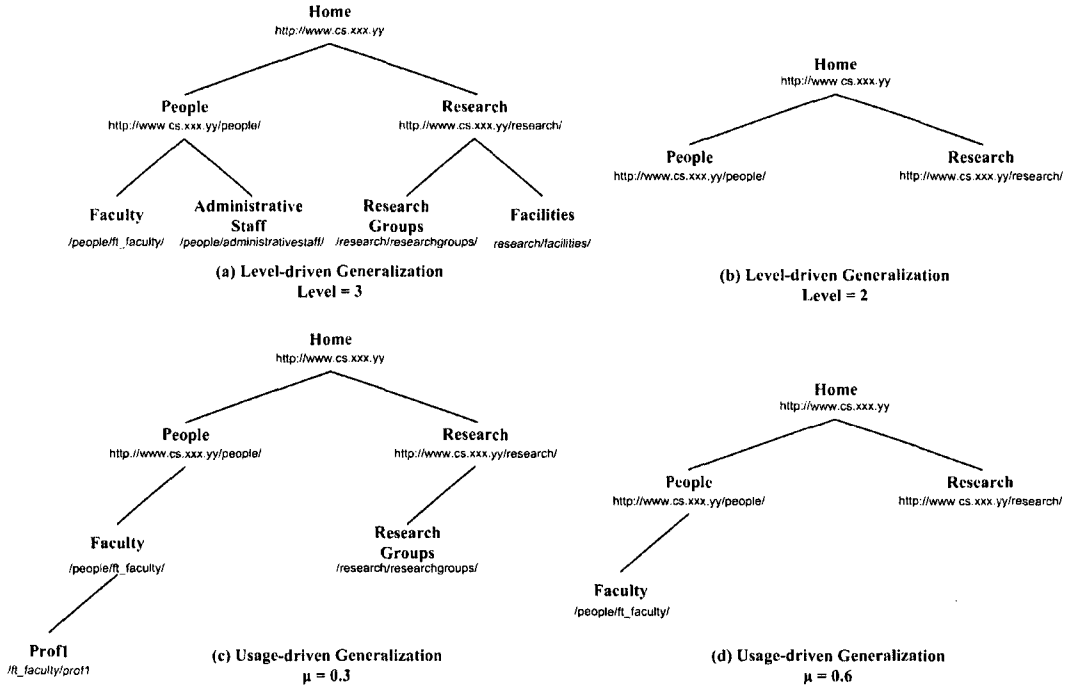


Figure 3.4: Compressed Hierarchies using Generalization Techniques

hence retained in the compressed hierarchy shown in Fig. 3.4(c). Similarly, considering the aggregate usage for node {`Research Groups`} in Cluster 1, we have that $up_{researchgroup,c_1} = 0.4$ which is more than μ, and hence this node is retained in the compressed hierarchy along with its path. The final list of compressed URL pairs we get includes the following pairs {`(Prof2, Faculty)`,`(Administrative Staff, People)`, `(Grp1, Research Groups)`, `(Grp2, Research Groups)`, `(Facilities, Research)`}. A higher usage threshold value would result in higher data compression, as shown in Fig. 3.4(d), where $\mu = 0.6$.

		Usage Profiles (UPs)	
ID	Page Category	Cluster 1	Cluster 2
1	Home	1.0	0.8
2	People	0.3	0.6
3	Research	0.4	0.2
4	Faculty	0.2	0.05
5	Admin. Staff	0	0.2
6	Research Groups	0.2	0
7	Facilities	0	0
8	Prof1	0.4	0.1
9	Prof2	0	0
10	Grp1	0.1	0
11	Grp2	0.1	0.05

Table 3.1: Sample Usage Profiles Data

3.2.3 Session Transformation

Once the list of compressed URL pairs is obtained, it can be used to generate the generalized sessions. The original session is structured as $(s_i, \langle k \rangle)$, where s_i and k are unique identifiers for sessions and URLs, respectively. This is a compact structure which is generally used to implement sessions that associate binary weights with pages.

The session transformation is a two step process.

- In the first step, each URL k in a session s_i is replaced with its ancestor URL k_a if a pair (k, k_a) is found in the list of compressed URL pairs. This results in duplicate entries in the same session.

- In the second step, the session obtained from the first step is restructured as $(s_i, \langle (k_a, count_{s_i k_a}) \rangle)$, where k is a distinct URL in session s_i and $count_{s_i k_a}$ is

the number of times k_a appears in s_i. For example, a session of pages in Fig. 3.3, (People, Faculty, Prof2, Research, Grp2), is transformed into ((People, 1) (Faculty, 2) (Research, 1) (Research Groups, 1)) based on the usage driven URL compression, shown in Fig. 3.4(c).

This has the desired effect of reducing the sparsity of the session data.

In the original session data, the session s_i is represented as a binary vector of size N (the total number of URLs in the page hierarchy), where the k^{th} entry is 1 if the user accessed the k^{th} URL during this session, and 0 otherwise. As discussed in Section 2.2, a common session similarity measure used to calculate the similarity between two such sessions is the Cosine similarity measure (c.f. Eq. 2.2).

However, the cosine measure is not effective for measuring the similarity between two generalized sessions since it tends to misrepresent information. For example, suppose that we have two generalized sessions characterized by the following values for count : $s_1 = [2, 4], s_2 = [4, 8]$. This reflects that user 2 was more interested in the two generalized concepts as compared to user 1. However, it is easy to verify that the two users are maximally similar according to the cosine angle, i.e. $cosinesim_{s_1,s_2} = 1$.

This limitation of the cosine measure is because it does not take into account the magnitude of user interest (*i.e.*, count) associated with each URL. To incorporate this additional (semantic) information in measuring the session similarity, we define below a similarity measure based on the one proposed in [58], which utilizes fuzzy sets. This measure is commonly used to calculate the similarity between sessions which are represented as vectors of the normalized time spent on each page instead of binary

values. Castellano [8] demonstrate the effectiveness of implementing this measure. The similarity of two sessions i and j is defined as:

$$fuzzysim_{ij} = \frac{\sum_{k=1}^{M} min\{\omega_{ik}, \omega_{jk}\}}{\sum_{k=1}^{M} max\{\omega_{ik}, \omega_{jk}\}} \tag{3.3}$$

where ω_{ik} is the degree of interest for URL k in session s_i and M is the total number of available URLs.

This degree of interest is a value in the range $[0, 1]$, and can be defined as a function of *count*, where 0 indicates count=0, and 1 indicates the page is surely preferred by the user. Ideally, ω_{ik} can be modeled using an exponential function, since the degree of user interest is believed to grow rapidly with each increment in *count* and level off to 1 after some number of clicks, making it a highly interesting page for that user.

$$\omega_{ik} = 1 - e^{-count_{ik}} \tag{3.4}$$

where $count_{ik}$ is the number of times the URL k or any of its descendants were clicked in session s_i.

This similarity measure (c.f. Eq. 3.3) is reflexive (i.e. $fuzzysim_{ii} = 1$) and symmetric (i.e. $fuzzysim_{ij} = fuzzysim_{ji}$). Defined as the ratio of the cardinality of the intersection of fuzzy sets to the cardinality of union of fuzzy sets [58], this measure gives a much more acceptable evaluation of the similarity between two sessions. Reconsidering the example, we can calculate the similarity between s_1 and s_2 as follows:

$$S_{fuzz,s_1,s_2} = \frac{0.6 + 0.8}{0.8 + 0.9} = 0.82 \tag{3.5}$$

This stage results in a set of generalized sessions and a matrix of pairwise similarity values computed among all. This information forms the input to the next stage of the web usage mining process described in the next chapter.

Chapter 4

Clustering Generalized Sessions and Evaluation

In this chapter we address the problem of discovering useful patterns efficiently from a large set of web access log data. The aim is to capture not only the patterns comprised of items that are frequently accessed in the website (that are usually apparent and thus uninteresting), but also patterns identifying trends of using the infrequently accessed but significant items of the website.

4.1 Pattern Discovery

Clustering [70] has been recognized as a principal technique for mining web usage data. In our work, we cluster the generalized sessions for evaluation of the proposed technique. While we could use any clustering algorithm, we used Relational Fuzzy

Subtractive Clustering (RFSC) algorithm (c.f. Section 2.2.1) for its scalability to very large datasets. Note that generalization is recommended in general when the underlying dataset is large, as is in our case. Using RFSC, we can study the impact of generalization on large dimensional data. Besides, RFSC does not require user specified input parameters and is less sensitive to noise. In what follows we briefly describe RFSC clustering technique which serves as basis for testing our proposed generalization technique.

Similarity Measures: As discussed in Section 2.2.1, RFSC operates on a relational matrix R for N sessions, where R_{ij} is the dissimilarity between sessions i and j. This dissimilarity is defined as $D_{ij} = 1 - S_{ij}$, where S_{ij} is the similarity between sessions s_i and s_j and is calculated using two measures [50]. The first measure is the fuzzy similarity measure proposed in Eq. 3.3:

$$S_{1,ij} = fuzzysim_{ij} \tag{4.1}$$

The second measure incorporates the syntactic URL similarity, defined as:

$$S_{2,ij} = \frac{\sum_{k=1}^{M} \sum_{l=1}^{M} s_{ik} s_{jl} S_u(k,l)}{\sum_{k=1}^{M} s_{ik} \sum_{l=1}^{M} s_{jl}} \tag{4.2}$$

where

$$S_{u(k,l)} = min\left(1, \frac{|p_k \cap p_l|}{max(1,\ max(|p_k|, |p_l|) - 1)}\right) \tag{4.3}$$

in which p_k denotes the path traverses from the root to the node which corresponds to the k^{th} URL. The length of path p_k is denoted as $|p_k|$.

The final similarity is defined by Equations 4.1 and 4.2:

$$S_{ij} = max(S_{1,ij}, S_{2,ij}) \tag{4.4}$$

After construction of relation R, RFSC is applied to group together similar generalized sessions. A significant advantage of using the implicit concept hierarchy (page hierarchy) for generalization is that it does not require any modification to the underlying clustering algorithm. A session is still represented as a vector over pageviews and the implicit semantics are effectively incorporated and isolated within the similarity measure used to compare sessions.

The Membership Matrix: Once the cluster centers are identified over the generalized session data, a membership matrix $C \times N$ is constructed for the entire weblog data i.e. for the original pre-processed, non-generalized set of sessions. Here, C is the number of clusters and N is the total number of sessions. An entry u_{mi} (c.f. Eq. 2.5) in the membership matrix indicates the membership value of session s_i in cluster m. The reason for incorporating the original sessions into the discovered patterns at this stage of the web usage mining process is to avoid loosing significant information in the applications of the mined knowledge. Note that some URLs may not have played any part during pattern discovery because they had negligible usage in the stratified sample. However, it is important that this does not make them insignificant to the model. They are a part of the underlying dataset and thus need to participate in subsequent applications of the derived model.

Post-processing Tasks: Finally, a defuzzification step is performed which assigns each session to a cluster to which its degree of membership is the highest. Every session which has a low membership to all the clusters is considered as noise and hence ignored. The resulting crisp clusters are used to generate the usage profiles for interpretation purposes.

4.2 Experiments and Results

We have studied performance of the proposed usage driven generalization using access log data collected by the server in our department at Concordia collected(*i.e.* from December 31, 2004 to March 05, 2005. Standard data pre-processing tasks (c.f. Section 2.1.1) were performed on the available log data. The maximum elapsed time between two consecutive accesses in the same session was set to 45 minutes. The root node "/" was filtered out from all sessions as it was accessed by more than 80% of the sessions. We also removed short sessions of length 1 or 2, since they have insignificant contribution towards users' access patterns. At the end of the preprocessing stage, the data was segmented in 64530 user sessions with 12685 distinct URLs. It is important to note that this is a much larger dataset compared to those used by other generalization-based clustering techniques [24, 51]. The average length of sessions was 6.997 pages and the sparsity level (c.f. 3.2.1) was 0.999329. The page hierarchy has 9 levels. Clustering results obtained by applying RFSC on the generalized sessions was used in our evaluation. We have used the discovered profiles to provide a quantitative

as well as qualitative comparisons between our usage-driven and level-driven generalization techniques. All experiments were performed on a typical desktop computer with a dual core 3GHz Pentium IV processor with 2GB of RAM running Windows XP. We used C++ in our implementations.

4.2.1 Experimental Setup

Our experiments started with the selection of stratified samples from the 64530 user sessions, to estimate the usage of URLs. For comparison purposes, we used three sampling fractions f=0.03, 0.07, 0.15, which generated samples of sizes 2500, 5000, and 10000 sessions respectively, where f is the ratio of the size of the sample to the total number of usage sessions. Each sample was clustered using the RFSC algorithm. The resulting usage profiles were used to guide the URL compression algorithm. For enabling a study of the effect of a varying usage threshold, experiments were carried out over a range of the usage threshold μ from 0 to 1.0. Fig. 4.1 depicts the URL compression ratios obtained at some threshold values for each stratified sample. The URL compression ratio r is defined as [51]:

$$r = \frac{M - M'}{M} \tag{4.5}$$

where M is the total number of actual URLs and M' is the total number of retained URLs.

There are two important points to note here. Firstly, r increases as μ approaches 1. It is not totally unexpected that this increased data compression might result in the

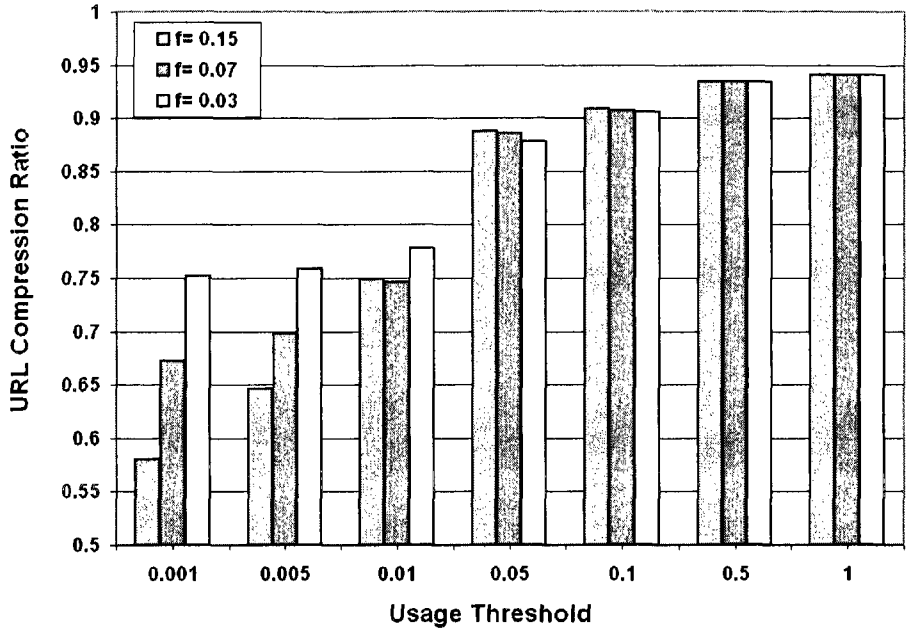


Figure 4.1: Effect of varying the Usage Threshold on the Compression Ratio for Different Values of the Sampling Fraction

loss of some significant URLs and thus adversely effect the quality of the patterns. Secondly, r also increases when f approaches 0. For example, a sample of 10000 sessions approximately retains 40% of the URLs at $\mu = 0.001$, compared to a smaller sample of size 2500 which retains only 30% of the original URLs for the same threshold (Fig. 4.1). We also noted that the URLs participating in a sample approximately form a subset of the URLs participating in a larger sample. This property holds due to the use of stratified samples. Randomly chosen sample sets of the same sizes have been tested and were found to give inconsistent results, again not surprising given the time dependent variation in preferences of users. As a consequence of this property, at higher values of μ all three samples generate similar compression hierarchies, both in terms of size and content, as shown in Fig. 4.1.

4.2.2 Comparison on Cluster Validity

In this section, we compare the quality of the clusters generated from applying RFSC on a collection of generalized sessions. We carried out a number of clustering experiments to evaluate the effectiveness of our usage driven generalization as compared to the level driven generalization technique. Firstly, the non-generalized original sessions were clustered. The results were used as reference in performing the intended comparison. Next, for the usage driven generalization, sessions were generalized using three sample sizes ($f = 0.15, 0.07, 0.03$) and varying μ = 0.001, 0.005, 0.01, 0.05, 0.1, 0.5, 1.0 for each sample. Finally, for the level driven generalization, sessions were generalized using the generalization levels 2, 3, 4. Since, at level 4 r was less than 0.5, the remaining levels were not considered. Further lowering of compression ratios will defeat one of the main purposes of generalization. In carrying out these experiments, the only varying factor was the input of generalized sessions, produced by the corresponding technique being tested. The remaining steps were kept identical.

The goal of clustering is to group similar sessions together such that intra-cluster distance is maximized and inter-cluster distance is minimized (c.f. Section 2.2.2. This property can be quantified using a compactness to separation ratio used as the clustering validity index [54]. In this thesis, we have used the validity index proposed in [67].

The Validity Index

Given a clustering of C clusters over a dataset of N sessions, we assume that cluster c_m is non-singleton, where $1 \leq m \leq C$.The Xie Index SC is defined in [67] as follows:

$$SC = SP \times CP \tag{4.6}$$

The optimal clustering solution is obtained where SC is maximized. The compactness CP for clustering with textitC clusters, is defined as:

$$CP = C \Big/ \sum_{m=1}^{C} \left(\frac{\sum_{s_j \in c_m, s_j \neq s_{c_m}} u_{mj}^2 R_{c_m j}^2}{\sum_{s_j \in c_m, s_j \neq s_{c_m}} u_{mj}^2} \right) \tag{4.7}$$

where u_{mj} is the membership of session s_j in cluster m, and $R_{c_m j}$ is the dissimilarity between sessions c_m and j. Similarly, separation SP is defined as:

$$SP = \left(\frac{\sum_{m=1}^{C} min_{1 \leq p \leq C, m \neq p} \{R_{c_m c_p}\}}{C} \right)^2 \tag{4.8}$$

It is important to note that the session dissimilarity R_{ij} is calculated using similarity measures proposed in Equations 2.2 and 4.2. As mentioned above, the use of original sessions and their corresponding similarity measures during pattern analysis will enable us to conduct a more focused, useful and comparable evaluation of the clustering results.

Evaluation

Fig. 4.2 compares the quality obtained from clustering of the sessions generalized through usage and level driven generalization techniques. For each technique, the

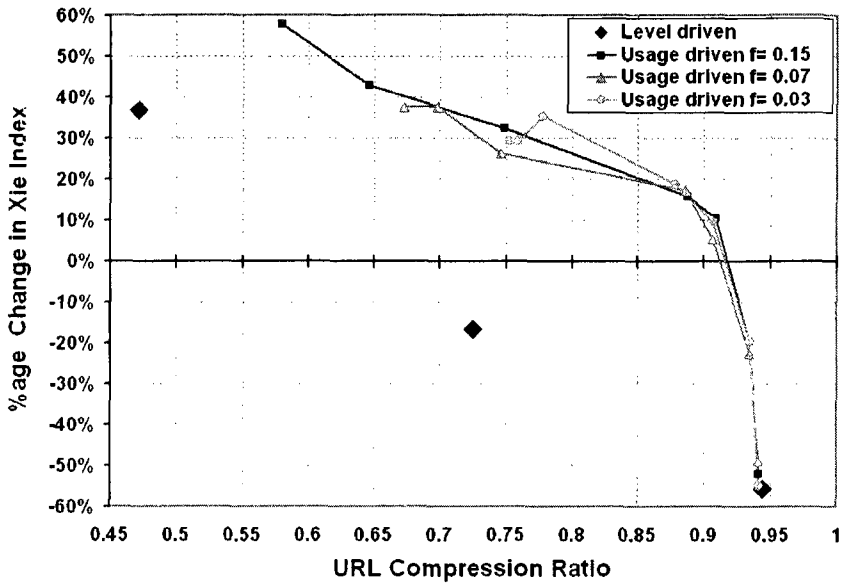


Figure 4.2: Comparison of Cluster Validity

figure shows the amount of change (in percentage) obtained in the Xie Index (relative to the non-generalized session clustering) at different compression ratios.

It should be noted that for the level driven generalization the attainable compression ratios are equal to the number of levels in the hierarchy. Since in these experiments we use three levels 2, 3, 4, it generates three corresponding compression ratios, $r = 0.94$, 0.72, and 0.47, which are shown as single points in Fig. 4.2. On the other hand, usage driven generalization is more flexible allowing to choose from a range of compression ratios. For a given sample, the highest compression ratio is achieved at $\mu = 1$ and declines as μ approaches 0. For example, when $f = 0.15$, the range of the compression ratio is $[0.58, 0.94]$, as shown in Fig. 4.1. This range shrinks for smaller sample sizes, due to the reduction in the number of URLs participating in the corresponding sample.

The results of our experiments indicate that usage driven generalization outperforms the level driven generalization in terms of cluster validity, at all compression ratios. Moreover, usage driven generalization indicates slight improvement in the quality, relative to the non-generalized session clustering, at very high rates of data compression, i.e., $r > 0.9$. On the other hand, it is only at the 4th level ($r = 0.47$) that the level driven generalization indicates improved quality compared to the clustering of non-generalized sessions.

Comparing the three usage driven generalization scenarios, Fig. 4.2 shows that the quality of the clusters is similar for higher compression ratios. This makes sense, since the high support URLs are approximately identical for all three samples. As μ approaches 0, the compression ratios decline and a larger number of URLs are retained, as shown in Fig. 4.1. If these URLs are actually popular in the original dataset, retaining them will improve the quality of the mined profiles. This is reflected by the general increase in the cluster validity at lower compression ratios. However, it is likely for some of these URLs to belong to the infrequently accessed portions of the navigation trails. Retaining such URLs will introduce sparsity in the data and could negatively impact the quality of the resulting clusters. This can be seen at $f = 0.03$ as the quality declines when $r < 0.78$.

The usage threshold μ is a critical parameter, but easy to understand and control by any user, as its impact on compression and quality is consistent. Since the dataset being used for evaluation is highly sparse, we obtain the best quality of clustering as μ approaches 0 (c.f. Section 3.2.1). We also believe that this assumption will

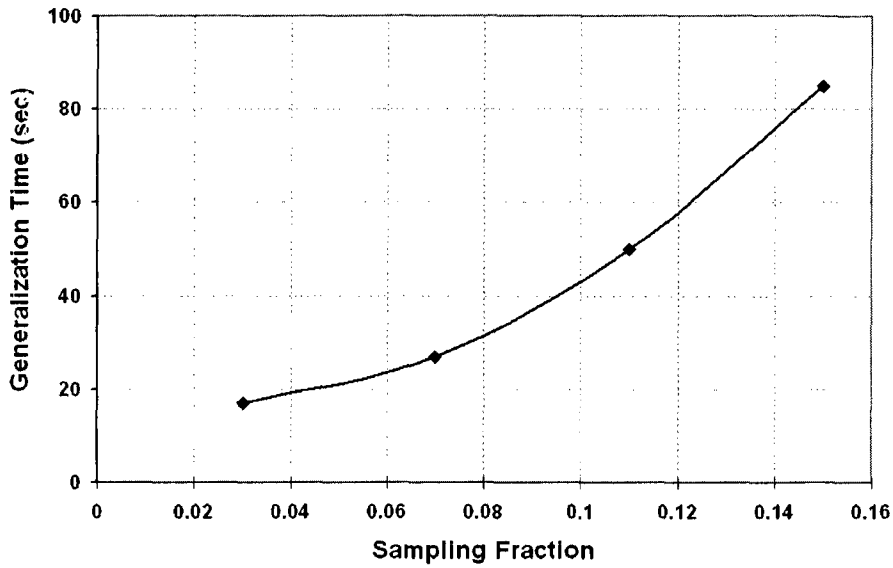


Figure 4.3: The Total Time taken in Web Data Generalization for Different Sample Sizes

hold for other datasets which include a very large number of items and sparse usage. However we also maintain that the choice of the threshold can also be driven by the available computational resources for the web usage mining process by making a feasible selection from the range of compression ratios available. It should be noted that lower values of μ require relatively higher computational resources, memory and time.

4.2.3 Execution Time and Scalability

The time spent on the generalization process should not overshadow the time savings it yields. The generalization time is defined as the total time taken during usage estimation, URL compression and session transformation. Among these, usage estimation is found to be the most time consuming step. Therefore, the processing time

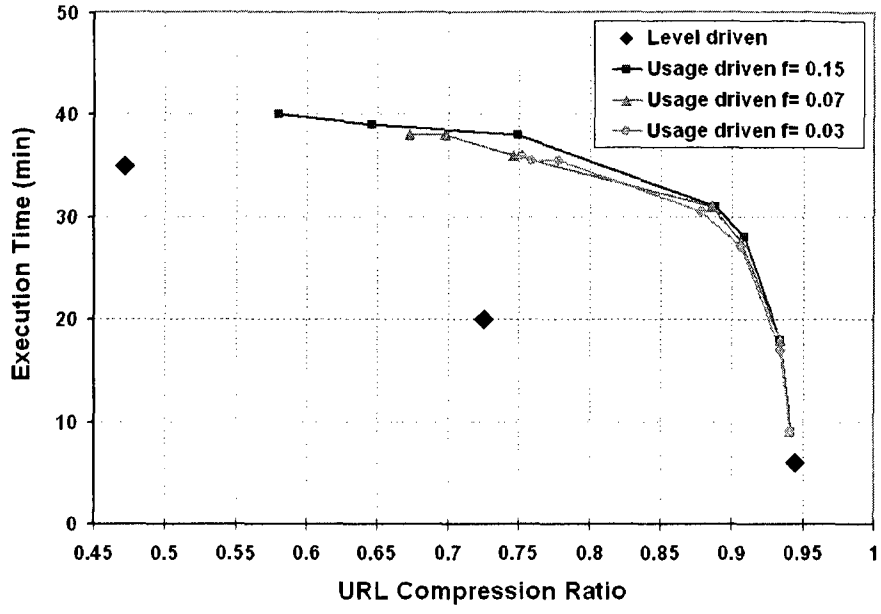


Figure 4.4: Comparison of Execution Time in Minutes

depends mainly on the size of the session samples derived for usage estimation, as shown in Fig 4.3. However, it will not take much time if we use an efficient algorithm for clustering sample data, since the samples are quite small. The figure shows that for the largest sample size of 10000 sessions, it took about 80 seconds to generalize the web data. This is a small fraction of the time required for clustering non-generalized sessions, which took 45 minutes to complete. On the other hand, by varying r, usage driven generalization gains a 20% to 80% reduction in clustering time compared to the clustering of non-generalized sessions. This suggests that the proposed generalization is an effective data compression technique in web usage mining.

One of the underlying objectives of web data generalization is to reduce the time taken for modeling and pattern discovery. Fig. 4.4 compares this time in minutes, using generalized sessions either derived from usage ($f = 0.15, 0.07, 0.03$) or

from level (level = $2, 3, 4$). As can be seen, at similar compression levels, level driven generalization takes slightly less time compared to usage driven generalization. This is because level driven generalization merges many frequently accessed URLs, which reduces the average length of the generalized sessions, resulting in faster session similarity calculation. This relatively reduces the overall execution time for the level driven generalization technique. It is important to note that this saving in processing time comes at the cost of reduced quality of the mined profiles, as shown in Fig. 4.2.

4.2.4 Profile Analysis

The Xie Index, discussed in Section 4.2.2, gives an idea of the cluster validity. However, it would be helpful to better understand the impact of usage driven generalization in a subjective manner, by actually examining the profiles it produced. We did this exercise manually by analyzing the clusters generated. Our experiments confirm the quality improvements achieved due to generalization driven by usage information. For this, we compared the profiles obtained from clustering of sessions generalized through the usage driven and level driven generalization techniques. For the level driven technique, sessions were generalized to the third level of the hierarchy, corresponding to a compression ratio of 73%. For comparison purposes, a similar compression ratio was obtained from the usage driven generalization by adjusting the usage threshold appropriately. Let UP_{usage} and UP_{level} be the profiles obtained after applying the usage driven and the level driven generalizations on the session data, respectively. Also let $UP_{original}$ denote the profiles generated from the clustering of non-generalized session

UP_{level}	UP_{usage}
Profile: up_{l1}	**Profile:** up_{u1}
	/programs/ugrad/courses.html 0.3646
	/programs/ugrad/cs/cs.html 0.1895
	/prospective_students.html 0.1656
/programs/ugrad/courses.html 0.3079	/current_students.shtml 0.1454
/programs/ugrad/cs/cs.html 0.1787	/programs/ugrad/cs/curriculum.html 0.1341
/prospective_students.html 0.1549	/programs/ugrad/soen/soen.html 0.1160
/programs/grad/courses.html 0.1507	**Profile:** up_{u2}
/programs/ugrad/cs/curriculum.html 0.1283	/programs/grad/courses.html 0.3438
/programs/ugrad/soen/soen.html 0.1130	/programs/grad/masters/master.html 0.2106
	/current_students.shtml 0.1524
	/programs/grad/diploma/courses.html 0.1041

Table 4.1: Comparison of Discovered Usage Profiles

data.

RFSC algorithm finds clusters in descending order of their potentials in the dataset. Therefore, each cluster has its potential relative to the first cluster. This feature of RFSC is helpful in identifying significance of some differences among the profiles compared. We summarize our observations from this analysis.

The Impact of Overgeneralization: Table 4.1 depicts an example of overgeneralization by the level driven generalization technique. The usage profile shows the URLs whose popularity is greater than 0.1. In UP_{usage}, Profile up_{u1} and Profile up_{u2} capture the user's interests in the undergraduate and graduate courses offered by the department, respectively. Although, they reflect semantically dissimilar concepts,

they are merged into Profile up_{l1} of UP_{level}. This is a good example of the curse of overgeneralization. At the third level of the hierarchy we have the general URLs /programs/ugrad/ and /programs/grad/. The frequently accessed subsuming portions of the navigation trail are lost in the process of generalization. The high structural similarity of the general URLs overshadows the dissimilarity of their usage and thus the clustering algorithm merges them into one profile. As a result, not only an important profile is lost but the generated profile also misrepresents the usage behavior. This explains the negative impact of the quality of any subsequent applications of the model.

Similarly, we identified another instance where a profile related to a professor, corresponding to a high potential cluster in UP_{usage} and $UP_{original}$ is lost in UP_{level}. A detailed analysis of the compression indicates that usage driven generalization retained some URLs in the lower portions of the navigation trails associated with that professor. These URLs were obviously merged into their ancestors at the third level of the hierarchy by the level driven generalization. We believe that these URLs were significant for identification of this profile.

The Impact of Sparsity: It is found that the number of repeating profiles in UP_{level} was much higher than in UP_{usage}. Data sparsity is identified as a reason for this repetition. The clustering algorithm in general cannot identify items to be similar if data is sparse. It is likely that generalizing to level 3 does not alleviate the sparsity of some items in the web data. On the other hand, usage driven generalization

correctly identifies the sparse, less used portions of the hierarchy and generalizes the corresponding URLs. This allows to extract general patterns which are more meaningful.

Apart from the aforementioned aspects, UP_{usage} ignores the three lowest potential profiles in $UP_{original}$, which are identified as noise after careful inspection. On the other hand, UP_{level} ignores only one of these profiles and retains the other two. As a result, clusters in UP_{usage} are better separated.

Chapter 5

Conclusions and Future Work

"*What does ALL my data say?*" This is the main question that a web usage mining system tries to answer. However, as the complexity of the web applications and user's interaction in these applications increases, we need to either exploit new algorithms or optimize an established approach in order to find a scalable and accurate answer. In recent years, semantic enrichment of weblogs has been perhaps the most promising development in the area of web usage mining. As affirmed in Cooley [13], "not only is the web usage mining process enhanced by content and structure, it cannot be completed without it." However, for complex websites which have numerous users, the integration of semantics further aggravates the dimensionality and sparseness of the feature space. This not only impairs the performance of many data mining algorithms, but also the comprehensibility of the mined patterns by a human analyst. Restricting mining to concepts at a chosen level of abstraction, using concept hierarchies, is perhaps the most common, yet implicit, approach to dimensionality reduction. Although

dealing with granularity is, in fact, one of the techniques exploited by many recent semantic web usage mining studies; in Section 2.4, we argued that this alone is not sufficient to prevent the *curse of dimensionality* while retaining enough detail.

The idea of applying generalization techniques to the web data in order to achieve data compression and sparsity alleviation was first introduced in [24]. Since then, a number of techniques have been proposed for incorporating generalization with semantic web usage mining; while much progress has been made, improved quality is still desired for extracted patterns. This is because, in an attempt to reduce the data sparsity, existing techniques tend to overgeneralize the usage information.

In this thesis, we proposed a web data generalization methodology which incorporates the information-rich usage data to guide the process of concept hierarchy ascension. For this, we utilize the page hierarchy to replace the actual clicks with high level URLs only when they are infrequently accessed. In other words, it exploits the usage information to guide the generalization of the sparse portions of the web data. This addresses the problem of overgeneralization in existing level based techniques. A usage threshold is introduced to guide this generalization process. We also developed a new session similarity metric, based on fuzzy sets, which better captures the similarity of the generalized sessions. We show that usage driven generalization can extract significantly better quality web usage patterns at high compression ratios compared to existing level driven generalization techniques. It is important to note here that the choice of the usage threshold has a significant role in maximizing these underlying benefits of usage driven generalization. If the threshold is set too high,

overgeneralization may occur and the validity of the patterns may be in question. On the other hand, if the threshold is set too low, the generalization would not effectively alleviate the sparsity of the dataset. However, we have derived parameters in this study which will help an experienced data analyst to select a suitable threshold value, with an underlying assumption that the clustering used to derive the usage values is approximately uniform.

We established the effectiveness of the proposed technique through numerous experiments and analysis of the results, both qualitatively and quantitatively, for alleviating data sparsity and significant improvement in quality of mined patterns. Our results indicate that usage driven generalization of web sessions is an effective technique and when coupled with an efficient pattern discovery algorithm, it can serve to fulfil the much desired goals of higher quality pattern discovery and scalability.

Some of the potential extensions and improvement strategies for the work done in this thesis are outlined below:

- The usage-driven generalization of explicit concept hierarchies would also be a significant contribution. With the advent of dynamic URLs and multi-framed sites, semantic enrichments using explicit concept hierarchies is becoming indefensible. It would be highly useful to measure the impact of our technique on the scalability and accuracy of such systems.

- We have mainly focused on clustering as the primary data mining technique for the discovery of usage profiles. However, a potential future direction would

be to evaluate the quality improvements in a variety of other techniques, e.g., association rule mining and/or sequential pattern mining.

- It would be interesting to compliment the cluster evaluation methods used in this study with statistical significance testing techniques to assess the strength of pattern changes comparing the non-generalized clustering results to those obtained from our proposed technique. In particular, we would like to verify whether an observed change is statistically significant or not.

- In this work we used a large weblog dataset for testing purposes. It would be interesting to apply the proposed technique on datasets from other websites and domains (such as e-commerce) to study the effectiveness of the proposed approach. Although we firmly believe that the usage driven generalization will significantly improve the pattern discovery for a website in any domain, given it has a very large number of items and the usage data tends to be sparse.

Bibliography

[1] Rakesh Agrawal and Ramakrishnan Srikant. Fast algorithms for mining association rules in large databases. In *Proceedings of Twentieth International Conference on Very Large Data Bases*, pages 487–499, September 1994.

[2] Rafal Angryk, Roy Ladner, and Frederick Petry. Generalization data mining of fuzzy object-oriented databases. In *Advances in Fuzzy Object-oriented Databases: Modeling and Applications*, pages 85–112, November 2004.

[3] Arindam Banerjee and Joydeep Ghosh. Concept-based clustering of clickstream data. In *Proceedings of the Third International Conference on Information Technology*, pages 145–150, December 2000.

[4] Richard Bellman. *Adaptive Control Processes: A Guided Tour.* Princeton University Press, Princeton, NJ, USA, 1961.

[5] Bettina Berendt, Bamshad Mobasher, Miki Nakagawa, and Myra Spiliopoulou. The impact of site structure and user environment on session reconstruction

in web usage analysis. In *WEBKDD-Mining Web Data for Discovering Usage Patterns and Profiles*, pages 159–179, July 2002.

[6] Alex G. Buchner and Maurice D. Mulvenna. Discovering internet marketing intelligence through online analytical web usage mining. *ACM SIGMOD Record*, 27(4):54–61, December 1998.

[7] Yandong Cai, Nick Cercone, and Jiawei Han. Attribute-oriented induction in relational databases. In *Knowledge Discovery in Databases*, pages 213–228. AAAI/MIT Press, December 1991.

[8] Giovanna Castellano, A. Maria Fanelli, Corrado Mencar, and M. Alessandra Torsello. Similarity-based fuzzy clustering for user profiling. In *Proceedings of the International Conferences on Web Intelligence and Intelligent Agent Technology - Workshops*, pages 75–78, November 2007.

[9] Soumen Chakrabarti, Byron Dom, Rakesh Agrawal, and Prabhakar Raghavan. Using taxonomy, discriminants, and signatures for navigating in text databases. In *Proceedings of the Twenty-Third International Conference on Very Large Data Bases*, pages 446–455, August 1997.

[10] Stephen L. Chiu. Fuzzy model identification based on cluster estimation. *Journal of Intelligent and Fuzzy Systems*, 2(3):267–278, December 1994.

[11] Patrick Clerkin, Pádraig Cunningham, and Conor Hayes. Ontology discovery for the semantic web using hierarchical clustering. In *Semantic Web Mining Workshop at ECML/PKDD*, pages 27–38, September 2001.

[12] Robert Cooley. *Web usage mining: discovery and application of interesting patterns from web data.* PhD thesis, University of Minnesota, 2000. Major Adviser-Jaideep Srivastava.

[13] Robert Cooley. The use of web structure and content to identify subjectively interesting web usage patterns. *ACM Transacions on Internet Technology*, 3(2):93–116, May 2003.

[14] Robert Cooley, Bamshad Mobasher, and Jaideep Srivastava. Web mining: information and pattern discovery on the world wide web. In *Proceedings of the Ninth International Conference on Tools with Artificial Intelligence*, pages 558–567, November 1997.

[15] Robert Cooley, Bamshad Mobasher, and Jaideep Srivastava. Data preparation for mining world wide web browsing patterns. *Knowledge and Information Systems*, 1(1):5–32, February 1999.

[16] Honghua Dai and Bamshad Mobashar. Integrating semantic knowledge with web usage mining for personalization. In *Web Mining: Applications and Techniques*, pages 276–306, August 2004.

[17] Honghua Dai and Bamshad Mobasher. Using ontologies to discover domain-level web usage profiles. *Proceedings of the Second Semantic Web Mining Workshop at ECML/PKDD*, August 2002.

[18] Magdalini Eirinaki and Michalis Vazirgiannis. Web mining for web personalization. *ACM Transactions on Internet Technology*, 3(1):1–27, February 2003.

[19] Magdalini Eirinaki, Michalis Vazirgiannis, and Iraklis Varlamis. Sewep: using site semantics and a taxonomy to enhance the web personalization process. In *Proceedings of the Ninth ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, pages 99–108, August 2003.

[20] Martin Ester, Hans-Peter Kriegel, and Jörg Sander. Spatial data mining: A database approach. In *Proceedings of the Fifth International Symposium on Advances in Spatial Databases*, pages 47–66, July 1997.

[21] Federico Michele Facca and Pier Luca Lanzi. Recent developments in web usage mining research. In *Data Warehousing and Knowledge Discovery, Fifth International Conference, DaWaK*, pages 140–150, September 2003.

[22] Sergio Flesca, Sergio Greco, Andrea Tagarelli, and Ester Zumpano. Mining user preferences, page content and usage to personalize website navigation. *World Wide Web*, 8(3):317–345, September 2005.

[23] Jerome H. Friedman. An overview of predictive learning and function approximation. In *From Statistics to Neural Networks: Theory and Pattern Recognition Applications*, pages 1–61, September 1994.

[24] Yongjian Fu, Kanwalpreet Sandhu, and Ming Shih. A generalization-based approach to clustering of web usage sessions. In *Revised Papers from the International WEBKDD Workshop on Web Usage Analysis and User Profiling*, pages 21–38, August 1999.

[25] Alexander F. Gelbukh, Grigori Sidorov, and Adolfo Guzmán-Arenas. Document indexing with a concept hierarchy. In *Proceedings of the First International Workshop on New Developments in Digital Libraries*, pages 47–54, July 2001.

[26] Jiawei Han. *Data mining: concepts and techniques.* Morgan Kaufmann Publishers Inc., San Francisco, CA, USA, 2005.

[27] Jiawei Han, Yandong Cai, and Nick Cercone. Knowledge discovery in databases: An attribute-oriented approach. In *Proceedings of the Eighteenth International Conference on Very Large Data Bases*, pages 547–559, August 1992.

[28] Jiawei Han, Yandong Cai, and Nick Cercone. Data-driven discovery of quantitative rules in relational databases. *IEEE Transactions on Knowledge and Data Engineering*, 5(1):29–40, February 1993.

[29] Jiawei Han and Yongjian Fu. Dynamic generation and refinement of concept hierarchies for knowledge discovery in databases. In *Knowledge Discovery in Databases Workshop*, pages 157–168, July 1994.

[30] Jiawei Han and Yongjian Fu. Discovery of multiple-level association rules from large databases. In *Proceedings of the Twenty-First International Conference on Very Large Data Bases*, pages 420–431, September 1995.

[31] Jiawei Han, Raymond T. Ng, Yongjian Fu, and Son K. Dao. Dealing with semantic heterogeneity by generalization-based data mining techniques. *Papazoglou and Schlageter (Eds): Cooperative Information Systems*, pages 207–231, November 1997.

[32] Venky Harinarayan, Anand Rajaraman, and Jeffrey D. Ullman. Implementing data cubes efficiently. *ACM SIGMOD Record*, 25(2):205–216, June 1996.

[33] Richard J. Hathaway, James C. Bezdek, and John W. Davenport. On relational data versions of c-means algorithms. *Pattern Recognition Letters*, 17(6):607–612, May 1996.

[34] Bernardo A. Huberman, Peter L.T. Pirolli, James E. Pitkow, and Rajan. M. Lukose. Strong regularities in world wide web surfing. In *Science*, pages 95–97, September 1998.

[35] Suk hyung Hwang, Hong-Gee Kim, Myeng-Ki Kim, Sung-Hee Choi, and Hae Sool Yang. A data-driven approach to constructing an ontological concept hierarchy

based on the formal concept analysis. In *Proceedings of the International Conference of Computational Science and Its Applications*, pages 937–946, May 2006.

[36] InternetWorldStats.com. *Internet Usage Statistics.* `http://www.internetworldstats.com/stats.htm`, January 2009.

[37] Renta Ivncsy and Sndor Juhsz. Analysis of web user identification methods. *International Journal of Computer Science*, 2(3):212–219, August 2007.

[38] Anupam Joshi and Raghu Krishnapuram. On mining web access logs. In *ACM SIGMOD Workshop on Research Issues in Data Mining and Knowledge Discovery*, pages 63–69, May 2000.

[39] Csaba Legány, Sándor Juhász, and Attila Babos. Cluster validity measurement techniques. In *Proceedings of the Fifth WSEAS International Conference on Artificial Intelligence, Knowledge Engineering and Data Bases*, pages 388–393, February 2006.

[40] Mark Levene, José Borges, and George Loizou. Zipf's law for web surfers. *Knowledge and Information Systems*, 3(1):120–129, February 2001.

[41] Ari Luotonen, Henrik Frystyk Nielsen, and Tim Berners-Lee. *W3C User's Guide.* Available at: `http://www.w3.org/Daemon/User`, May 1996.

[42] Alice Marascu and Florent Masseglia. Mining sequential patterns from data streams: a centroid approach. *Journal of Intelligent Information Systems*, 27(3):291–307, November 2006.

[43] Florent Masseglia, Pascal Poncelet, and Rosine Cicchetti. An efficient algorithm for web usage mining. *Networking and Information Systems Journal*, 2(5-6):571–603, October 1999.

[44] Florent Masseglia, Pascal Poncelet, and Maguelonne Teisseire. Using data mining techniques on web access logs to dynamically improve hypertext structure. *ACM SIGWEB Newsletter*, 8(3):13–19, October 1999.

[45] Andrew McCallum, Ronald Rosenfeld, Tom M. Mitchell, and Andrew Y. Ng. Improving text classification by shrinkage in a hierarchy of classes. In *Proceedings of the Fifteenth International Conference on Machine Learning*, pages 359–367, July 1998.

[46] Robin Van Meteren and Maarten Van Someren. Using content-based filtering for recommendation. In *ECML/MLNET Workshop on Machine Learning and the New Information Age*, pages 47–56, May 2000.

[47] Bamshad Mobashar. Web usage mining. In *Web Data Mining: Exploring Hyperlinks, Contents and Usage Data*, pages 450–483, December 2006.

[48] Bamshad Mobasher, Honghua Dai, Tao Luo, Yuqing Sun, and Jiang Zhu. Integrating web usage and content mining for more effective personalization. In *Proceedings of the First International Conference on Electronic Commerce and Web Technologies*, pages 165–176, September 2000.

[49] Alexandros Nanopoulos, Dimitrios Katsaros, and Yannis Manolopoulos. Exploiting web log mining for web cache enhancement. In *Revised Papers from the Third International Workshop on Mining Web Log Data Across All Customers Touch Points*, pages 68–87, August 2002.

[50] Olfa Nasraoui, Hichen Frigui, Raghu Krishnapuram, and Anupan Joshi. Extracting web user profiles using relational competitive fuzzy clustering. *International Journal on Artificial Intelligence Tools*, 9(4):509–526, December 2000.

[51] Olfa Nasraoui and Esin Saka. Web usage mining in noisy and ambiguous environments: exploring the role of concept hierarchies, compression, and robust user profiles. In *From Web to Social Web, Workshop on Web Mining, WebMine*, pages 82–101, Septemeber 2006.

[52] Netcraft.com. *Web Server Survey.* `http://news.netcraft.com/archives/2009/01/16/january_2009_web_server_survey.html`, January 2009.

[53] Daniel Oberle, Bettina Berendt, Andreas Hotho, and Jorge Gonzalez. Conceptual user tracking. In *Proceedings of the First International Atlantic Web Intelligence Conference*, pages 155–164, May 2003.

[54] Nikhil R. Pal and James C. Bezdek. On cluster validity for the fuzzy c-means model. *IEEE Transactions on Fuzzy Systems*, 3(3):370–379, August 1995.

[55] Scott Parent, Bamshad Mobasher, and Steve Lytinen. An adaptive agent for web exploration based on concept hierarchies. In *Proceedings of the Ninth International Conference on Human Computer Interaction*, August 2001.

[56] Carsten Pohle and Myra Spiliopoulou. Building and exploiting ad hoc concept hierarchies for web log analysis. In *Proceedings of the Fourth International Conference on Data Warehousing and Knowledge Discovery*, pages 83–93, September 2002.

[57] Fabrice Rossi, Aïcha El Golli, and Yves Lechevallier. Usage guided clustering of web pages with the median self organizing map. In *Proceedings of the Thirteenth European Symposium on Artificial Neural Networks*, pages 351–356, April 2005.

[58] Simone Santini and Ramesh Jain. Similarity measures. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 21(9):871–883, September 1999.

[59] Myra Spiliopoulou, Bamshad Mobasher, Bettina Berendt, and Miki Nakagawa. A framework for the evaluation of session reconstruction heuristics in web-usage analysis. *INFORMS Journal on Computing*, 15(2):171–190, April 2003.

[60] Ramakrishnan Srikant and Rakesh Agrawal. Mining generalized association rules. In *Proceedings of Twenty-First International Conference on Very Large Data Bases*, pages 407–419, September 1995.

[61] Ramakrishnan Srikant and Rakesh Agrawal. Mining sequential patterns: generalizations and performance improvements. In *Proceedings of the Fifth International Conference on Extending Database Technology*, pages 3–17, March 1996.

[62] Jaideep Srivastava, Robert Cooley, Mukund Deshpande, and Pang Tan. Web usage mining: Discovery and applications of usage patterns from web data. *SIGKDD Explorations*, 1(2):12–23, January 2000.

[63] Alexander Strehl. *Relationship-based clustering and cluster ensembles for high-dimensional data mining.* PhD thesis, 2002. Supervisor-Joydeep Ghosh.

[64] Gerd Stumme, Rafik Taouil, Yves Bastide, Nicolas Pasquier, and Lotfi Lakhal. Fast computation of concept lattices using data mining techniques. In *Proceedings of the Seventh International Workshop on Knowledge Representation meets Databases*, pages 129–139, August 2000.

[65] Bhushan Shankar Suryavanshi, Nematollaah Shiri, and Sudhir P. Mudur. An efficient technique for mining usage profiles using relational fuzzy subtractive clustering. In *Proceedings of the International Workshop on Challenges in Web Information Retrieval and Integration*, pages 23–29, April 2005.

[66] Doru Tanasa and Brigitte Trousse. Advanced data preprocessing for intersites web usage mining. *IEEE Intelligent Systems*, 19(2):59–65, March 2004.

[67] Ying Xie, Vijay V. Raghavan, and Xiaoquan Zhao. 3m algorithm: finding an optimal fuzzy cluster scheme for proximity data. In *Proceedings of the IEEE International Conference on Fuzzy Systems*, pages 627–632, May 2002.

[68] Yunjuan Xie and Vir V. Phoha. Web user clustering from access log using belief function. In *Proceedings of the First International Conference on Knowledge Capture*, pages 202–208, October 2001.

[69] Rui Xu and Donald Wunsch II. Survey of clustering algorithms. *IEEE Transactions on Neural Networks*, 16(3):645–678, May 2005.

[70] Tak Woon Yan, Matthew Jacobsen, Hector Garcia-Molina, and Umeshwar Dayal. From user access patterns to dynamic hypertext linking. *Computer Networks and ISDN Systems*, 28(7-11):1007–1014, May 1996.

[71] Lotfi A. Zadeh. Fuzzy sets. *Information and Control*, 8(3):338–353, June 1965.

[72] Osmar R. Zaïane, Jiawei Han, Ze-Nian Li, Sonny Han Seng Chee, and Jenny Chiang. Multimediaminer: A system prototype for multimedia data mining. In *Proceedings ACM SIGMOD International Conference on Management of Data*, pages 581–583, June 1998.