Intelligent Web Usage Clustering Based Recommender System

Shafiq Alam*
Department of Computer Science
University of Auckland
Auckland, New Zealand
sala038@aucklanduni.ac.nz

ABSTRACT

Our work focuses on tackling the problem of efficiency and accuracy of web usage clustering for recommender systems. Accurate analysis and preprocessing of web usage data and efficient web usage clustering are the key factors that influence the development of clustering based implicit recommender system. We propose an analysis and preprocessing model to tackle the poor quality of web usage data. To address the problem of efficient web usage clustering, we propose a Particle Swarm Optimization (PSO) based clustering approach. Having shown our PSO based clustering performs well; we extend it for mining the usage behavior of web users. We select Java API (Application Programming Interface) documentation usage data as a case study for our recommender system.

Categories and Subject Descriptors

H.3 [**Information Storage and Retrieval**]: ClusteringInformation Filtering; H.2 [**Database Applications**]: Data mining

General Terms

Algorithms

1. INTRODUCTION

Knowledge Discovery in Databases or data mining (KDD) extracts novel, verifiable, comprehendible, and potentially useful information from huge amounts of data [3] [4]. Data mining techniques look for informative patterns such as clusters of relevant data, classification and association rules, sequential patterns and prediction models from different types of data such as textual data, audio-visual data, and microarray data. One of the areas where the use of data mining technique has seen a marked increase is the mining of web data, which is growing by leaps and bounds in three different directions i.e. usage, structure and contents. The activities of web users around web resources, the contents of the web resources and structural information of the web resources are generating petabytes of data on an hourly basis. Web mining, a sub domain of data mining, tackles the information extraction problem of the World Wide Web (WWW).

More than 2 billion internet users [1] are generating massive amounts of data. Understanding this enormous amount of data is important for finding the behavior of web-users and predicting future moves of the web user. Web-users generally follow some particular sequence while moving from one page to another or from one semantic topic to another topic. The discovery of such invisible patterns is the ultimate goal of WUM.

One of the most rapidly growing applications of patterns generated by the web usage mining process is recommender systems; tools that assist users find a particular resource based on some prior knowledge about the usage of that resource, the behavior of the user, or the behavior and usage of other similar users and resources. Recommender systems aim to generate the set of most suitable choices among these resources [9]. There are two types of recommender system, explicit recommender system and implicit recommender systems. In explicit recommender systems web users actively and willingly contribute to build the knowledge base by providing favorites lists, ratings, and responding to different surveys. In implicit systems, data is gathered implicitly from the usage logs and cookies of the users [8]. Implicit systems are complex to build as compared to explicit recommender systems because of the low quality of data, and the huge amounts of data. In order to develop an implicit recommender system, the knowledge base needs to be clean, relevant and possess large amount of data.

Data clustering being one of the most suitable data mining techniques for recommender systems, reduces the target search space for recommendations. On the one hand, the development of implicit recommender system where the users do not specifically add their interests to the knowledge base, needs huge amount of data about users' interests [7]. While on the other hand, the amount of such data poses a serious challenge in near real time pattern extraction. Ordinary data analysis tools are not efficient to analyze and extract patterns from such huge data. One solution is to enhance the efficiency of mining algorithms and generate accurate and efficient recommendations by involving optimization in

*Shafiq Alam

RecSys'11, October 23–27, 2011, Chicago, Illinois, USA.

[1]Internet usage statistics, the internet big picture http://www.internetworldstats.com/stats.htm

the web usage mining process. The involvement of intelligent optimization techniques can enhance the complex, real time, and costly web mining process. Swarm intelligence (SI) is one such intelligent optimization technique. It exploits the social and cognitive learning properties of vertebrates and insects, and models their behavior through multi agent systems in the form of software components communicating with each other in a decentralized environment. We use swarm intelligence based optimization to improve the web usage mining process for implicit recommender systems. Our project has three different phases, web usage data gathering and preprocessing, extraction of patterns and detection of abnormal observation from the data and building implicit recommender system based on the extracted patterns.

2. DATA PREPROCESSING

Web usage data is very raw and contains a huge amount of irrelevant data, which not only makes the pattern discovery process inefficient, but also hampers the techniques from finding useful patterns in the data. The analysis of web usage data and the subsequent preprocessing of the data are important tasks that need to be carried out before the actual pattern discovery starts. We propose a step by step analysis and preprocessing method, and collected Java API documentation usage data from the Department of Computer Science's web usage logs. Table 1 shows a snap shot of the data which is collected from the usage log of Java API documentation after performing some initial preprocessing and cleaning of the log file. The logs have sufficient user requests which could give us a realistic sequence of individual sequences as well as API usage sequence. The logs contain requests from 2006 to 2010, each containing raw data including Java API usage requests. Table 2 shows some of the initial statistics about the data. We will pass this data through a sophisticated preprocessing stage to extract usage sessions as we mentioned in [2] [1].

Table 1: Java API navigation requests

Java API Requests
/java/java1.5/api/java/nio/package-frame.html
/java/java1.5/api/java/awt/font/package-frame.html
/java/java1.5/api/java/rmi/server/package-frame.html
/java/java1.5/api/javax/crypto/package-frame.html
/java/java1.5/api/index.html?java/util/PriorityQueue.html
/java/java1.5/api/java/awt/event/HierarchyEvent.html

Table 2: Statistics about the data

log statistics	
Start Date:	24/12/2006
End date:	31/12/2009
Total requests:	38387774
java API Request:	5206947
Images requests:	1537978
CSS requests:	44569
Distinct IP's:	74211
Distinct Pages:	60140

3. CLUSTERING WEB USAGE DATA

Data clustering is a comprehensive and important technique in data mining which aims to group data (text, multimedia, microarray etc.) on the basis of similarities and dissimilarities among the data elements [6] [3]. Due to a increasing number of web users, and the activities of these users, web clustering has been found as one of the most useful way of understanding their activities [5] [1]. Web data is different from traditional textual and numeric data and so usage clustering is different from traditional data clustering. Web usage data needs a sophisticated pre-processing stage before it can be grouped by the clustering technique. Another important thing in such data is the selection of appropriate attributes for clustering as by default web usage data do not possess all the basic attributes needed for Web usage mining. They needed to be calculated from the basic web usage data during the selection process.

4. EPSO AND HPSO BASED DATA CLUSTERING

To tackle the problems of efficiency in the process of pattern extraction, and dealing with the accuracy issue of proposed recommendations generated by WUM-based recommender systems, we proposed Particle Swarm based clustering approaches. The proposed Evolutionary Particle Swarm Optimization based clustering (EPSO-clustering) and Hierarchical Particle Swarm Optimization based clustering (HPSO-clustering) are bio-inspired clustering techniques, which uses the cooperation and communication of swarms to perform clustering in a hierarchical manner while containing the benefits of partitional clustering. The details of these approaches can be found in [1] and [2].

We used EPSO and HPSO-clustering to tackle the problem of huge amount of data, and lack of domain knowledge for the web usage data. We used these techniques in a hierarchical manner by selecting a relatively large number of particles to be spread across the web usage data. Different generations of the swarm iterate to find the cluster solutions from the usage data to be used for generating recommendations. The Pseudocode of HPSO-clustering is shown in Algorithm 1. For initial validation of our propose clustering approaches for recommender system, we chose benchmark web usage data, i.e. NASA web log file, which contains 1891715 HTTP requests to NASA Kennedy Space Centre's web server. We also collected our own Java API documentation usage data from the University of Auckland (UOA), Department of Computer Science web log (CS-WebLog). The duration of the data is from 2006 to 2010. Some preliminary clustering results on NASA web usage data are shown in Table 3

5. HPSO-CLUSTERING BASED OUTLIER DETECTION

To tackle the quality problem in web usage data, cleaning is carried out to make the data useable for pattern extraction. Consequently analysis and preprocessing of web usage repositories is very important. Without systematic analysis, selection and preprocessing, it is impossible for the data to be used for pattern extraction. Our data shows that 60% to 80% of the data has no significant contribution in the pattern extractions. Using such data in the data mining pro-

Table 3: K-means,EPSO, and HPSO-clustering

Log	K means		PSO		HPSO	
	IntraCluster Dist.	Fitness	IntraCluster Dist.	Fitness	IntraCluster Dist.	Fitness
1	81.82	245.46	81.49	244.46	76.00	**228.00**
2	35.03	105.10	34.85	104.55	27.30	**81.90**
3	27.78	55.55	25.84	51.67	24.90	**49.80**
4	59.13	118.26	59.14	118.27	50.46	**100.92**

Algorithm 1 HPSO Clustering

Input: data file
Output: clusters of relevant data
Parameters: Swarm Size S, V_{Max}, V_{Min}, ω,q_1,q_2, and number of records N
Method:

```
INITIALIZE S,V_Max, V_Min, ω,q1,q2, and N
for Each Particle X do
    INITIALIZE X_i
end for
while (STOPPING CRITERIA(false)) do
    for each iteration do
        for Each Particle X do
            ASSIGN won data vectors to Particles
            CALCULATE pBest for each particle
            if pBest(t+1) is better then pBest(t) then
                pBest(t) ← pBest(t+1)
            end if
            CALCULATE Velocity V_i(t)
            UPDATE Position X_i(t)
        end for
        CALCULATE swarm strength
        FIND the weakest Particle
        FIND the nearest strong Particle
        MERGE both Particle
        UPDATE intra cluster distance
        UPDATE X_i(t)
        DELETE weaker particle
    end for
end while
```

cess is a waste of time and resources, however the process of identifying such data and removing it from consideration or replacing it for pattern extraction requires careful examination of the data. Some of the abnormal observation can be detected by the help of outlier detection methods. Web bot's requests are one of the examples which can deviate the analysis process if not treated accordingly. We propose an outlier detection mechanism to detect such abnormal observations. The proposed technique identifies outliers by using hierarchical clustering based outlier detection mechanism. The technique uses a distance measure to identify potential outliers. We call this measure, D_t. For different clusters this measure could be different as each cluster has their own relative outliers. The outlier threshold distance for particle X_i is calculated as:

$$OD(X_i) = \frac{D_t}{k} \times \sqrt{\sum_{j=1}^{k}(Y_j - X_i)^2} \quad (1)$$

where Y_i represents data vectors and k is the number of total data associated with a particular particle X_j. The

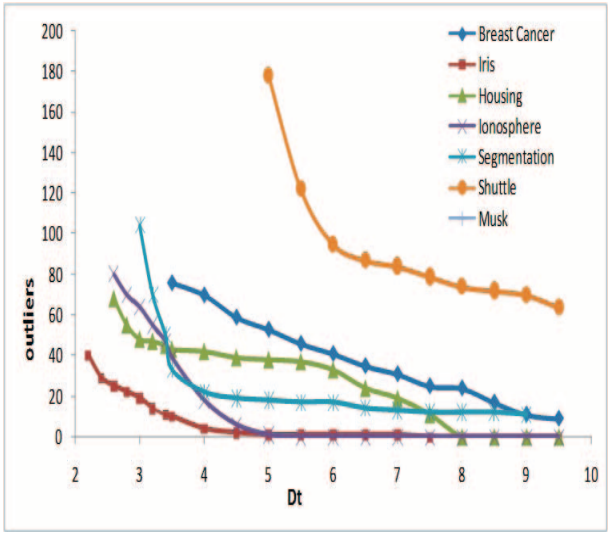

Figure 1: Outliers vs D_t

outlier distance evolves through different generations. Initial generations have a smaller outlier distance compared to the later generations where the intra cluster distance of the individual clusters increases due to merging of clusters and the population of clusters increases.

To validate the proposed approach, we tested it on some benchmark data sets from UCI machine learning repository. The findings of the experiments are shown in Figure 1 which depicts the number of detected outliers based on different values of D_t as well as the range of the optimal value. For small values of D_t, a relatively large number of outliers are detected which is evidence of the fact that the centroids are well adjusted in the center of the cluster and possess more data around them. In such cases a small change in D_t will add a larger number of data elements to the outliers list. For large values of D_t the number of outliers is small and a change in the value of D_t doesn't significantly increase the number of outliers. The optimal value of D_t for finding outliers in the given dataset was found to be between 3 and 7. The preliminary results show that the proposed approach is efficient and can be used for detecting abnormal observations in web usage data.

6. SWARM BASED RECOMMENDER SYSTEM

For the user of Java API documentation, the recommendations should be implicit, accurate, timely, and personalized, which means we need a sufficient amount of usage data and efficient algorithms to generate and rank recommendations. Once the data is preprocessed, it can be used for pattern

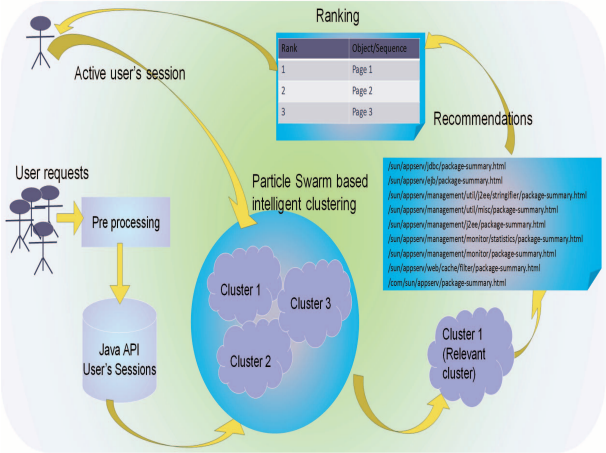

Figure 2: Clustering based recommender system

extractions for our recommender system. The clustering approach uses a distance measure which helps in clustering, it can be used to detect abnormal observations, generate distance based recommendations, and rank the proposed recommendations. Figure 2 shows the overall system which is based on the analysis presented in this paper.

A cluster based recommendation system helps reduce the problem space of huge web usage data and generate recommendations for a group of user and for a group of resources. Our proposed approach clusters Java API usage sessions on the basis of similarity amongst the usage behavior and then incorporates this clustering information with the usage patterns found in the Java API elements.

As the individual web user requests are not enough to make a knowledge base for providing quality recommendations. We consider sessions instead individual requests as an input for our recommender system. Usage profile in the form of sessions are extracted from the logs and treated as data points of a cluster. We represent each corresponding cluster as an agent or particle which searches for recommendations an active user session within its corresponding cluster and ranks them on the basis of its similarity. The nearer objects have higher rank in the recommendation. The history of the current user of the API will also be observed to add value to the clustering. This approach helps reduce the input search space for recommendation and ranking. The corresponding recommendation enables the user to be directed to their desired page.

Apart from the experimentation on benchmark clustering data and NASA web usage data, the recommender system will be evaluated on different precision and recall measurements, and empirical measures to verify the efficiency and accuracy of the recommendation system.

7. CONCLUSION AND FUTURE WORK

We analyze a subject focused web usage data from the University of Auckland (UOA), Department of Computer Science web log (CS-WebLog) containing web requests to various parts of the Java API documentation by different java API users.

To tackle the problems of efficiency in the process of pattern extraction, and dealing with the accuracy issue of proposed recommendations generated by WUM-based recommender systems, we proposed Particle Swarm based clustering approach called EPSO-clustering and HPSO-clustering approaches. Both approaches exploit the cooperation and communication of swarms to perform clustering in an hierarchical manner while containing the benefits of partitional clustering. Comparison results of both these approaches show that the proposed approaches are accurate and efficient and are capable of being used for pattern extraction for web usage mining based recommender system. We tackle the problem of detecting abnormal observations in the web logs such as requests from web crawlers as outliers by using an HPSO-clustering based outlier detection method. The technique is based on HPSO clustering which generates clusters and detects abnormal observation simultaneously.

For our future work, we will evaluate our WUM based recommender system and provide a benchmark repository of web usage data to be used for WUM and recommender systems.

8. ACKNOWLEDGMENTS

I would like to acknowledge and thank the contributions, guidance and advice of my supervisor Gillian Dobbie, co-supervisor Patricia Riddle, and colleague Yun Sing Koh for helping me conduct this research.

9. REFERENCES

[1] S. Alam, G. Dobbie, and P. Riddle. Particle swarm optimization based clustering of web usage data. In *Web Intelligence and Intelligent Agent Technology, 2008. WI-IAT '08. IEEE/WIC/ACM International Conference on*, volume 3, pages 451 –454, 2008.

[2] S. Alam, G. Dobbie, and P. Riddle. Exploiting swarm behaviour of simple agents for clustering web users session's data. In L. Cao, editor, *Data Mining and Multi-agent Integration*, pages 61–75. Springer US, 2009.

[3] U. M. Fayyad, G. Piatetsky-Shapiro, and P. Smyth. From data mining to knowledge discovery: An overview. In *Advances in Knowledge Discovery and Data Mining*, pages 1–34. 1996.

[4] W. J. Frawley, G. Piatetsky-Shapiro, and C. J. Matheus. Knowledge discovery in databases: An overview. In *Knowledge Discovery in Databases*, pages 1–30. AAAI/MIT Press, 1991.

[5] Y. Fu, K. Sandhu, and M.-Y. Shih. A generalization-based approach to clustering of web usage sessions. In *Revised Papers from the International Workshop on Web Usage Analysis and User Profiling*, WEBKDD '99, pages 21–38, London, UK, 2000. Springer-Verlag.

[6] A. K. Jain, M. N. Murty, and P. J. Flynn. Data clustering: a review. *ACM Comput. Surv.*, 31:264–323, September 1999.

[7] B. Mobasher, R. Cooley, and J. Srivastava. Automatic personalization based on web usage mining. *Commun. ACM*, 43:142–151, August 2000.

[8] A. W. Neumann. Classification and mechanism design of recommender systems. In *Recommender Systems for Information Providers*, Contributions to Management Science, pages 1–8. Physica-Verlag HD, 2009.

[9] P. Resnick and H. R. Varian. Recommender systems. *Commun. ACM*, 40:56–58, March 1997.