Clustering the Users of Large Web Sites into Communities

Georgios Paliouras PALIOURG@IIT.DEMOKRITOS.GR
Institute of Informatics and Telecommunications, NCSR "Demokritos", Aghia Paraskevi, GR 15310 GREECE

Christos Papatheodorou PAPATHEODOR@LIB.DEMOKRITOS.GR
Division of Applied Technologies, NCSR "Demokritos", Aghia Paraskevi, GR 15310 GREECE

Vangelis Karkaletsis VANGELIS@IIT.DEMOKRITOS.GR
Constantine D. Spyropoulos COSTASS@IIT.DEMOKRITOS.GR
Institute of Informatics and Telecommunications, NCSR "Demokritos", Aghia Paraskevi, GR 15310 GREECE

Abstract

In this paper we analyze the performance of clustering methods on the task of constructing community models for the users of large Web sites. Community models represent patterns of usage of the Web site, which can be associated with different types of user. Knowledge of this type is clearly valuable for commercial sites, where each user is a potential customer. We argue that it is equally valuable for non-commercial sites, because it can assist greatly in the improvement of the site. We evaluate three clustering methods on usage data from a large site that covers on-line resources in Chemistry. The size of the site and its high hit rate impose a serious constraint on the scalability of the methods. We also examine two ways of encoding usage data, which give complementary information about the behavior of the users. Finally, the emphasis is on the construction of meaningful community models, by identifying the descriptive characteristics of communities, at a post-processing stage.

1. Introduction

Interest in the analysis of user behavior on the Internet has been increasing rapidly, especially since the advent of electronic commerce. New concepts, such as *electronic customer relationship management* (e-CRM), *web usage analysis* and *web mining* have appeared recently in the literature. All of these share the same goal: understanding the needs, interests and knowledge of the users of Web sites. The motivation stems from the fact that added value is not gained merely through larger quantities of data on a site, but through easier access to the required information at the right time and in the most suitable form. Seen from a different viewpoint:

"*The quantity of people visiting your site is less important than the quality of their experience*" (Schwartz, 1997)

Commerce on the Internet provides a unique opportunity for businesses to meet their customers and adapt their service to them. The same holds for sites that provide non-commercial services, the success of which depends largely on their understanding of their users' interests and needs. The use of the computer as an intermediary in the provision of services allows the collection of transaction data, with limited cost and effort. However, the transformation of these data to useful knowledge is not simple.

Machine learning techniques have been shown to address this issue well, leading to the creation of a separate field of study, i.e., that of knowledge discovery from data (KDD). Our approach is an attempt to apply the same ideas to usage data from Internet-based services. In this effort we make use of basic concepts and ideas from the area of user modeling. The motivation for borrowing these ideas is the fact that a Web site is still a computer-based system, being *used* by people on the Internet.

In particular, we focus our work on the concept of a user *community* (Orwant, 1995), which seems to apply best to a public Web site. Alternative approaches and other related work is presented in Section 2. A community corresponds to a group of users who exhibit common behavior in their interaction with the system. Our approach to the construction of community models is to use clustering methods on the usage data and then use a post-processing method to identify the distinguishing characteristics of each cluster. We give particular emphasis on the interpretation stage, because we believe that the descriptive characterization of a community in terms of a model is what the site owner needs. Section 3 of this paper explains our approach further. For the clustering process itself, we use existing techniques, which are briefly described in Section 4. An important criterion for the choice of a clustering method is its scalability to large sets of data, which is a

requirement for any method that we hope to apply to real-life Web sites. In this paper, we evaluate three clustering methods on a large site containing information for researchers in Chemistry. Section 5 describes the usage data acquired from this site and the pre-processing that we performed to them. Section 6 presents our evaluation of the three clustering methods on the data and Section 7 concludes our work, introducing some interesting issues that are still open.

2. Related Work

As mentioned above, our work focuses on the construction of models for user communities, i.e., groups of users with common behavior. One alternative concept is that of a personal user model, i.e., a model corresponding to a single user. Personal user modelling has already been studied extensively with the use of machine learning methods (e.g., Langley, 1999; Joachims et al., 1997; Pazzani & Billsus, 1997). The difficulty in adopting this approach for Web sites is the requirement for user identification at each interaction of a user with the system. Although registration procedures have been adopted by some Web sites, the need for identification is widely considered a "scarecrow" for potential users. Another alternative to communities, which we have looked at in the past (Paliouras et al. 1999) is the concept of a user *stereotype* (Rich, 1983), which corresponds to a community associated with common personal characteristics of the users, such as age, gender, etc. Although stereotypes are more informative than communities, the collection of personal information, exogenous to the system, in a way that does not violate the privacy of the users, is a thorny issue.

The broader research area in which our work contributes mostly is *Web usage mining* (Cooley et al., 1997), which applies data mining techniques to Web usage data, aiming to identify interesting usage patterns. As in most applications of data mining, different techniques extract different types of knowledge from usage data. This knowledge can be combined to provide a detailed picture of the visitors of a site. The site administrator can use this knowledge to improve the service, in terms of its presentation in the site, e.g., directing the users quickly to the most appropriate pages, or even by shifting the focal points of the service, i.e., reacting to the visitors' implicit requests. The work that has been presented so far in Web usage mining has looked mainly at the construction of association rules and sequence mining (e.g., Cooley et al., 1999; Büchner et al., 1999). The only work that we are aware of using a clustering method is that of Fu et al. (1999). We extend this work by looking at other clustering methods and two different ways to encode the data. Furthermore, we focus on the characterization of the resulting clusters, which is not addressed by Fu et al. (1999).

Another research area that is related to Web usage mining is *collaborative filtering*. At its outset, collaborative filtering was mainly used in information filtering services and aimed to relate one particular user to a group of other users with similar information needs. Research in collaborative filtering has been very active (e.g., Basu et al., 1998; Balabanovic & Shoham, 1997) and has also provided commercial products. There is substantial overlap between collaborative filtering and Web usage mining, but there is also an important difference of goals. Collaborative filtering is a user-centered approach, aiming to help the user directly, while Web usage mining aims to extract valuable knowledge for the system owner. Thus, the two approaches complement each other. Technically, most of the work in collaborative filtering uses instance-based methods, while Web usage mining and in particular our work is based on an active search for general models. A notable exception is the work of Breese et al. (1998), who used model-based methods for collaborative filtering. The primary goal of this work remained the ability to help the users directly, rather than to analyze the characteristics of the models that were generated.

3. Constructing and Evaluating Community Models

The original definition of a user community assumed the identification of individual users, i.e., a community model corresponded to an identifiable group of users. Some of our earlier work on this problem (Paliouras et al., 1998) was based on the same assumption. In that work, we used conceptual clustering algorithms, such as COBWEB (Fisher, 1987) and ITERATE (Biswas et al., 1998) to build community models by clustering personal user profiles. However, as mentioned above, the assumption of user identification is unrealistic for the majority of the existing Web sites. Despite this, community models can still be constructed on usage data, providing valuable information to the owner of a Web-based service. Implicitly community models built in this manner still correspond to groups of users with common behavior, although one may not be able to identify individual community members.

Conceptual clustering algorithms can still be used for the construction of the modified type of community that we are interested in. However, there are two problems in doing that: a theoretical and a practical one. The theoretical problem arises from the clustering process used by conceptual clustering algorithms, which constructs non-overlapping clusters. This is not desirable when clustering users into communities, since a user may belong to more than one community. The practical problem that we encountered is that the publicly-available versions of the two algorithms that we had could only handle small sets of data. This restriction was prohibitive for the work presented here. Due to these problems we decided to examine the three algorithms that are presented in Section 4, which allow overlapping clusters and can handle the large quantities of data that we have available.

The final goal of our approach is a set of models, which correspond to behavioral patterns for different types of

user. Clustering the users into communities with common behavioral characteristics is a first step towards this goal, but does not provide the desired patterns. In order to obtain these patterns we need to identify the descriptive characteristics of each cluster. The way in which we achieve this differs for different clustering methods, but the underlying idea is common.

A community model is expressed in terms of the same parameters as the underlying usage data. For instance, if the usage data simply record the pages in a site that are visited by a user within an access session, the communities will also be described in terms of the pages in the site. In a more formal language, given a set of boolean attributes A describing the instances in the data set, the model of a community C_j consists of a subset of A, A_j, which characterizes the members of the community, i.e., which are usually true for the members of the community. If A does not contain boolean attributes, characteristic attribute values (for nominal attributes) or ranges of values (for numeric attributes) will provide the community model. For the sake of simplicity we use boolean attributes here.

The selection of the descriptive attributes is done with the aid of simple metrics, which are based on the idea that an attribute is special for a community if its frequency within the community is significantly higher than its frequency in the whole data set. The natural choice of metric differs for different clustering methods. For the conceptual clustering algorithms, we have used a squared-difference measure, called *frequency increase* (Paliouras et al., 1998), motivated by the *category utility* search heuristic that these algorithms use. The choices for the clustering algorithms that are used here are explained in Section 4.

Having decided on a method to obtain the community models, we need to decide on the desired properties of these models. Our primary objective is to provide useful community models. In order for the models to be useful, they need to be relatively few in number and small in size. As a result, the *number of models* and their *average size* are two important measurable criteria for the success of a method. The exact figures for these criteria depend on the nature of the problem, e.g., the size of the attribute set.

However, a digestible set of models is not necessarily interesting. When there are only small differences between the models, accounting for variants of the same community, the segmentation of users into communities is not interesting. Thus, we are interested in community models that are as distinct from each other as possible. We measure the *distinctiveness* of a set of models M by the ratio between the number of distinct attributes that are covered and the size of the model set M. Thus, if there are J communities in M, A_j the attributes used in the j-th model and $A_{\square}$ the attributes appearing at least in one model, distinctiveness is given by the following equation:

$$Distinctiveness(M) = \frac{|A'|}{\sum_j |A_j|} \qquad (1)$$

Since we are interested in a small number of models that are distinct, the empty model set trivially satisfies our criteria. In a more realistic situation, we might have a small set of distinct models, which account for only a small part of the usage of the system. In order to avoid this problem, we introduce a further criterion that counteracts distinctiveness and size. The new criterion is the overall *coverage* of the community models, i.e., the proportion of attributes covered by the models. If $A_{\square}$ is the set of attributes appearing at least in one model, the coverage of the set of models M is:

$$Coverage\ (M) = \frac{|A'|}{|A|} \qquad (2)$$

The simultaneous optimization of distinctiveness and coverage by a set of community models indicates the presence of useful knowledge in the set. All of the four measures, introduced in this section, i.e., number of models, average model size, distinctiveness and coverage, are independent of the biases of the clustering algorithms. For this reason, they constitute objective criteria and they are used in the evaluation of the three clustering methods.

4. Introduction of the Clustering Methods

This section describes briefly three clustering methods that we use in our work. Two of them have been used widely in machine learning research. These are: Autoclass (Hanson et al., 1991) and self-organizing maps (Kohonen, 1997). Our presentation of those two methods is a short summary of the descriptions one can find in the original references. The third method is a fairly new one, which we have also modified to some extent. It is called *cluster mining* (Perkowitz & Etzioni 1998) and it will be presented in more detail than the other two.

4.1 Autoclass

Autoclass is an unsupervised classification algorithm using mixture modeling as the basic clustering method, supplemented by a Bayesian method for discovering efficiently the optimal classes in large data sets. As all unsupervised classifiers, its goal is to find the most probable class descriptions of a data set. Specifically the algorithm considers that each class C_j has its own probability distribution T_j. Its fundamental model is the finite mixture distribution which deals with two main probabilities: (i) the interclass probability of an instance X_i being a member of class C_j, $P(X_i \in C_j)$ and (ii) the class probability of observing the instance attribute values X_{ik} conditional on the assumption that X_i is a member of C_j, $P(X_{ik} | X_i \in C_j)$.

Autoclass performs two levels of search: (i) the model-level search, which determines the number of classes J and alternate class models for each T_j and (ii) the search for maximum posterior parameter values, which determines, for any fixed T_j, the set of the corresponding parameters that are maximally probable.

An important difference of Autoclass to other unsupervised classifiers is that it does not assign instances to the classes. Since it holds that no finite amount of evidence can determine an instance's class membership, it uses a weighted assignment, weighting on the probability of class membership. As explained above, this probabilistic assignment of instances to classes is very suitable to community modeling.

Autoclass has a built-in metric to assist in the descriptive characterization of the clusters, i.e., the construction of community models. This metric is called *influence* and it is defined for an attribute a_i and a class C_j as follows:

$$I(a_i \mid C_j) = \frac{P(a_i \mid C_j)}{P(a_i)} \log\left(\frac{P(a_i \mid C_j)}{P(a_i)}\right) \qquad (3)$$

This is a measure of how important an attribute is for a particular class. The community model should consist of the attributes with the highest influence values. The question that arises is how many attributes to keep per model. Clearly, negative influence values indicate that the attribute should not be used in the model. We decided to investigate this parameter further by normalizing the influence value ($I_n(a_i|C_j)=I(a_i|C_j)/\max_h I(a_h|C_j)$) and examining the effect of different threshold values.

In the described experiments we used the public-domain program Autoclass-C version 3-3-2, for Windows NT/95, downloaded from the following Web site: http://ic-www.arc.nasa.gov/ic/projects/bayes-group/autoclass.

4.2 Self-Organizing Maps

The self-organizing map (SOM) method is one of the most popular neural network approaches to unsupervised learning. We use the batch implementation of the SOM included in the Intelligent Miner™ software by IBM.

SOM performs a *k*-means type of clustering, by trying to identify prototype vectors for the *k* clusters. Prototypes act as centers of gravity for the clusters and their position in the input space is optimized iteratively. This process is common to a large family of other clustering methods. However, the power of SOM lies in its ability to search efficiently for the optimal prototypes. This is achieved by allowing neighboring clusters to affect the choice of a new prototype vector. The end-result is similar to a Voronoi tessellation of the input space, the boundaries of which approximate the theoretical Bayesian decision boundaries. In this sense, the SOM method provides a different path to the same destination as Autoclass does.

For the task of community modeling, each cluster corresponds to a community and communities that are close to each other tend to have similar models. We construct community models using the same metric as we do for Autoclass, i.e., influence. One technical difficulty with SOM is that one needs to specify a fixed number of communities. In practice, this number does not need to be accurate, but it needs to be large enough to cover the number of real communities in the data (IBM, 1998).

4.3 Cluster Mining

The cluster mining algorithm is a simple graph-based clustering method. Cluster mining discovers patterns of common behavior, by looking for all fully-connected subgraphs (cliques) of a graph that represents the users' characteristic attributes. It starts by constructing a weighted graph $G(A,E,W_V,W_E)$. The set of vertices A corresponds to the descriptive attributes used in the input data. The set of edges E corresponds to attribute co-occurrence as observed in the data. For instance, in the Web site on Chemistry that we examine, if the user visits pages concerning "Organic Chemistry" and "Polymers" an edge is added between the relevant vertices. The weights on the vertices W_A and the edges W_E are computed as the attribute frequencies and attribute co-occurrence frequencies respectively. Edge frequencies are normalized by dividing them with the maximum of the frequencies of the two vertices that they connect. The effect of normalization is to remove the bias for attributes that appear very often in all users. The resulting graph is given in Figure 1.

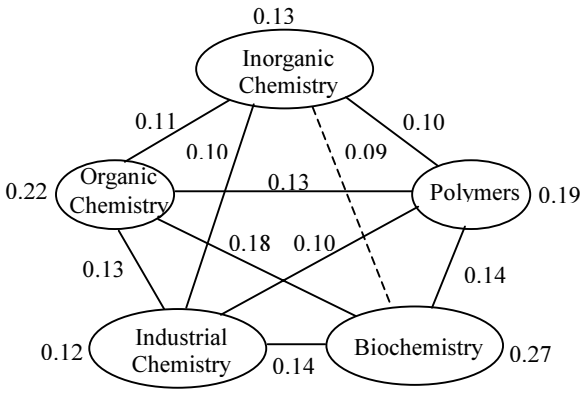

Figure 1. Normalized graph for cluster mining.

The connectivity of the graph is usually very high. For this reason we make use of a *connectivity threshold* aiming to reduce the edges of the graph. In our example in Figure 1, if the threshold equals 0.1 the edge ("Inorganic Chemistry", "Biochemistry") is dropped.

The cluster mining method was introduced in the PageGather system (Perkowitz & Etzioni 1998). The algorithm that we use differs in two ways from the original algorithm: (a) PageGather does not normalize the weights W_E and (b) it restricts its search to cliques of size k and to connected components. Despite the large theoretical complexity of the clique-finding problem, in practice the algorithm that we implemented (Bron & Kerbosch, 1973) is fast.[1] The efficiency of the algorithm allowed a full investigation of the effect of the connectivity threshold.

[1] It generates all cliques (approx. 200) of a large graph (239 vertices), with an average clique size of 100 vertices, in about 5 mins on a common SparcServer.

In contrast to the other clustering methods, such as Autoclass and SOM, the clusters generated by cluster mining group together characteristic features of the users directly. Each clique discovered by cluster mining is already a behavioral pattern. Therefore there is no need to post-process the clusters to construct descriptive models.

5. Description of the Usage Data

For this experiment, we used the access logs of the site "Information Retrieval in Chemistry" (http://macedonia.chem.demokritos.gr), which consists of a few thousand pages with a high hit rate. The log files consisted of 137,150 Web-server calls (log file entries) and covered the period between January and August 1999. Each log entry recorded access date and time, the visitor's IP address and domain name, and the target page (URL).

In order to construct a training set for the clustering algorithms, the data in the log files passed through two stages of pre-processing. First we extracted access sessions and then we translated the paths recorded in the access sessions into attribute vectors.

Access sessions were extracted from log files using the following procedure:

1. Grouping the logs by date and IP address.
2. Selecting a time-frame within which two hits from the same IP address can be considered to belong in the same access session.
3. Grouping the pages accessed by the same IP address within the selected time-frame to form a session.

In order to select the appropriate time-frame, we generated the frequency distribution of the page transitions in minutes. According to this distribution, transitions from one page to another, made with a time interval longer than one hour, had very low frequency. Thus, a sensible definition of the *access session* is a sequence of page transitions for the same IP address, where each transition is done at a time interval smaller than one hour. Based on this definition, our log files consisted of 11,893 access sessions.

Concerning the translation of access sessions to attribute vectors, we examined two alternative approaches. In the first approach each attribute in the vector represented the presence of a particular page of the Web site in the session. In the second approach, we used transitions between pages, rather than individual pages as the basic path components. In both cases the attribute vector consisted of boolean features, representing whether an attribute (a page or a transition) was present in a session or not.

There were 1,027 pages in the site that were visited at least once. Clearly the number of all possible transitions between these pages is prohibitively large. Even the number of different transitions that appear in the log files is very large. Thus, we needed a method to reduce the number of attributes in both experiments. This reduction was achieved by examining the frequency distributions of the pages and the transitions from one page to another. The two distributions were highly skewed, i.e., there was a small number of very frequent pages and transitions. Thus, we decided on a cut-off frequency of 30 for pages and 20 for transitions, which were the points where the corresponding distributions were becoming flat. Additionally we removed all transitions from a page to itself. As a result, 229 pages and 251 transitions survived this selection and were used to form attribute vectors. We also tried a method that uses Mutual Information, as a criterion for selecting attributes for unsupervised learning (Sahami, 1997). More than 90% of the attributes selected by this method, were within the high-frequency range that we selected. However, some of the attributes that were eliminated were clearly important, e.g., pages covering major research areas of chemistry, such as pharmaceutical chemistry. For this reason, we preferred the simple frequency-threshold approach.

The three clustering methods were applied to both representations of the data. In the first representation, the resulting model for a community is a group of pages, which are popular for users within the community. In the second representation, each community model is a set of page transitions. The page-based representation provides static models of user interests, similar to the user profiles used in collaborative filtering. For instance, one community of chemists may be found to be interested in organic chemistry, polymers and biochemistry. On the other hand, the transition-based representation provides navigational models, which show the paths through the site that users usually follow. For instance, one community may start from the Index page, then move to a high-level category and then navigate horizontally through the thematic categories. Both types of model are of interest in an information retrieval site, like the one we are examining. In this respect the two representations can be seen to provide complementary knowledge.

6. Evaluation Results

Each of the three clustering methods presented in Section 4 were applied to the usage data from the site "Information Retrieval in Chemistry", using both types of data representation. The results in this section examine the behavior of the three methods, in terms of the four criteria introduced in Section 3: number of communities, average size of community models, distinctiveness and coverage. Subsections 6.1 and 6.2 present the results, Subsection 6.3 explains how these results can be used to choose the desired set of community models and Subsection 6.4 illustrates the use of the models by the site administrator.

6.1 Number and Size of Community Models

Autoclass decides automatically on the number of clusters that it creates, during the model-level search. For the page-based representation, Autoclass created 13 communities, while for the transition-based representation only 4.

Both numbers are reasonably low, but one needs to combine these with the average size of community models, in order to judge whether the results are usable. The choice of influence threshold has a dramatic effect on model size. Both in the page-based and in the transition-based representations, there is a large number of attributes in each model that have very low influence values. As a result at threshold level of 0.2 there are just above 10 attributes per model, which corresponds to about 5% of the attribute set. For the transition-based representation, where the number of models is small, this sharp fall is bound to have a significant effect on coverage.

For the SOM method, as mentioned in Section 4, we had to fix the number of communities. For both types of representation we asked for 16 communities to be constructed, chosen to be slightly higher than the larger number of communities generated by Autoclass. Similar to Autoclass, the number of influential attributes in each community is very small. Setting the threshold to 0.2, the average size of the community models is less than 5.

In the cluster mining algorithm, the graph-connectivity threshold affects the number as well as the size of the community models. For small values of the threshold, the graph is highly connected and contains many cliques. It is really above the threshold value 0.2 that the number of cliques drops to manageable levels (below 40). The effect is almost identical for both types of representation. At the same time, the average size of the generated models is very small. For threshold values above 0.1 almost all cliques are pairs of pages or transitions. The result of this phenomenon is that the associations found between the attributes are really co-occurrence patterns, rather than substantial page groups or transition sequences.

6.2 Distinctiveness and Coverage of the Models

As explained in Section 3, a measurable indication that a set of community models is interesting can be obtained by optimizing the distinctiveness and the coverage of the models. The coverage measure captures also the combined effect on the number and the size of the models. For instance, low coverage is usually observed when both the number and the size of the models are small. Since, we are interested in the combined optimization of distinctiveness and coverage, we would like to present the results along those two dimensions in a combined manner. A good choice for such a presentation is the use of Receiver Operating Characteristic (ROC) curves. ROC curves are commonly used for cost-sensitive classification tasks, such as medical diagnosis, in order to present the trade-off between two types of error, e.g., over-diagnosis and under-diagnosis. The two types of error are measured by two corresponding measures, called *sensitivity* and *specificity.* A ROC curve is a plot of sensitivity against (1-specificity). Adapting this idea to our measures, we plot coverage against (1-distinctiveness). We name this type of plot a *trade-off* curve, in order to avoid confusion with ROC curves, since we are not measuring sensitivity and specificity. The results that we obtained for the two types of data representation are shown in Figures 2 and 3.

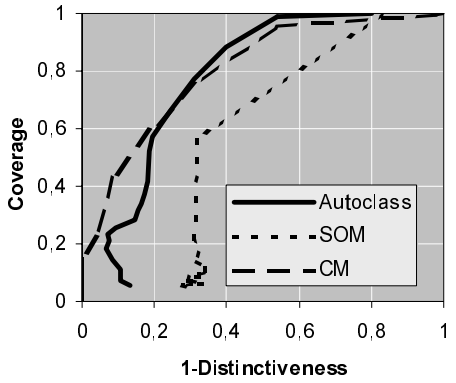

Figure 2. Trade-off curves for the page-based representation.

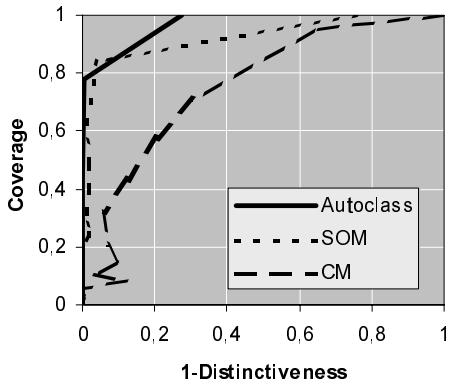

Figure 3. Trade-off curves for the transition-based representation.

Each curve in the two figures is generated by measuring coverage and distinctiveness for different values of the influence and the connectivity thresholds. In the experiments, we varied the threshold values from 0 to 1, with an interval of 0.5. A large proportion of the results lie in the low coverage – high distinction area, because the community models are usually small for threshold values above 0.2, as mentioned in Subsection 6.1. This is the reason for the "noisy" look of some of the curves at the lower left side of the graph.

Similar to the ROC curves, the optimal position is the top-left corner, where coverage and distinctiveness reach their maximum values. Thus, the surface underneath each curve is an indication of the overall performance of the method in the experiment. In this respect, Autoclass is doing well in both representations. In the page-based representation, Autoclass outperforms cluster mining only for high levels of coverage. However, due to the decrease in distinctiveness, the models become too large to be of use to the site administrator. For this reason, we need to trade off coverage for distinctiveness, moving to areas of the graph where the two methods become comparable. SOM is clearly doing worse than the other methods, using this representation. Its performance indicates that the number of clusters that we chose (16) is too large, result-

ing in low distinctiveness, even when the coverage is low. In the transition-based representation, Autoclass clearly achieves the best performance. The behavior of SOM is similar to that of Autoclass, despite the much larger number of clusters that SOM generates (16 instead of 4).

6.3 Choosing a set of Community Models

The results of the experiments presented above can help us choose a threshold value for each method and each data encoding. Since we are interested in a high level of coverage, we should reject high threshold values (above 0.2) that reduce coverage drastically. On the other hand, setting the threshold value too low is likely to increase significantly the size of the models and decrease their distinctiveness. Table 1 presents our preferred choices for the three methods and for the two types of representation. The choice of low threshold values (0.05) for Autoclass and SOM in the transition-based representation (T) is due to the high level of distinctiveness that the two methods achieve. On the other hand, the choice of a low threshold value (0.05) for SOM in the page-based representation (P) is dictated by the sharp fall in coverage for higher threshold values. The results in Table 1 are in accordance to the observations made in Subsection 6.3. Autoclass seems to have the best overall performance, while SOM follows closely when using the transition-based representation.

Table 1. Model set properties for the selected thresholds.

METHOD (REPRESENTATION)	THRESHOLD	MODELS	SIZE	COVERAGE	DISTINCTIVENESS
AUTOCLASS (P)	0.20	13	15.45	0.66	0.78
SOM (P)	0.05	16	11.75	0.56	0.68
CM (P)	0.20	76	2.18	0.59	0.81
AUTOCLASS (T)	0.05	4	49.25	0.78	1.00
SOM (T)	0.05	16	13.75	0.84	0.96
CM (T)	0.15	83	2.18	0.43	0.79

6.4 Using Community Models

Although the selection of community models goes some way towards delivering useful knowledge about the usage of the system, the presentation of the models to the site administrator in a digestible format requires further effort. It is not in the scope of this paper to address this issue fully, but we provide here some indication of the type of feedback that we have received from the site administrators, who are also Chemistry scientists. In order to introduce our results to the administrators, we have selected the "strongest" community model for each method and each type of representation. We define the strongest model as the largest model that survives for large threshold values. Table 2 presents our choices. Page-based models are presented as lists of page names, separated by semicolons. Transition-based models are presented as transition sequences connected by an arrow.

Table 2. The strongest community models, discovered by the three algorithms.

METHOD (REPRESENTATION)	COMMUNITY MODEL
AUTOCLASS (P)	Atmospheric-Chem Databases; Pharmaceutical-Chem Databases; Medicinal-Chem Databases; Chemistry Overview Databases; Hellas; Internet Overview.
SOM (P)	Computational Chemistry Journals; Chemistry Overview Databases; Thermochemistry.
CM (P)	Engineering; Environmental Sciences; Crystallography; Other Topics.
AUTOCLASS (T)	Chemistry ->related -> Internet; demok -> akno.
SOM (T)	Chemistry -> Internet -> WWW; Chemistry -> Biochemistry.
CM (T)	Internet -> Institute; Akno -> stats -> stats_all -> awards.

The administrators have identified patterns that were expected, as well as interesting "surprises". An example of the former type is the page-based model for Autoclass, which corresponds to a group of Chemistry scientists, who know how to use the site. Their specialties are quite technical, justifying familiarity with Internet-based services. On the other hand, the page-based model identified by cluster mining was a surprise that has caused thought. The explanation that was given to this pattern was that some fields, such as 'Environmental Sciences', are not covered sufficiently for the engineers in the field, causing them to navigate to more general-theme pages, such as 'Engineering' and 'Other Topics'. This issue is worth further consideration and could cause a change in the site.

7. Conclusions

Unsupervised machine learning methods seem to be a good choice for extracting useful knowledge from usage data on a Web site. We have looked at three clustering methods, two of which are popular in machine learning research and have already been used successfully in practice. We have included the clustering methods in a methodology, which consists of preprocessing the usage data in two different ways, constructing the communities using clustering and most importantly extracting community models. Using four evaluation criteria and data from a large Web site, we have examined the behavior of the three methods and have shown how the four criteria can be used to select a set of models. In the site that we looked at, Autoclass seemed to outperform the other methods, but further study with data from different sites is necessary, in order to draw a general conclusion.

In addition to further empirical evaluation, we are interested in associating this methodology to domain knowl-

edge about the structure and the content of Web sites. In particular, we are interested in providing guidelines about the use of the community models in different types of site. Additionally, we are looking into the systematization of the parts in our approach, which still require manual intervention, e.g., attribute selection and threshold selection. Finally, the important issue of presenting the models to the administrator remains open. Visualization techniques will be of use in this problem.

Acknowledgements

We would like to thank the members of the team "Information Retrieval in Chemistry" (E. Varveri, A. Varveris and P. Telonis) for providing the data and P. Tzitziras for his help with the experiments.

References

Balabanovic, M., & Shoham, Y. (1997). Content-based, collaborative recommendation. *Communications of the ACM*, *4*, 66-72.

Basu, C., Hirsh, H., & Cohen W. (1998). Recommendation as classification: Using social and content-based information in recommendation. *Proceedings of the Fifteenth National Conference on Artificial Intelligence* (pp. 714-720). Cambridge, MA: AAAI Press.

Biswas, G., Weinberg, J.B., & Fisher, D. (1998). ITERATE: A conceptual clustering algorithm for data mining. *IEEE Transactions on Systems, Man and Cybernetics*, *28*, 100-111.

Breese, J.S., Heckerman, D., & Kadie, K. (1998) Empirical analysis of predictive algorithms for collaborative filtering. *Proceedings of the Fourteenth Conference on Uncertainty in Artificial Intelligence* (pp. 43-52). San Francisco: Morgan Kaufmann Publishers.

Bron, C., & Kerbosch, J. (1973). Finding all cliques of an undirected graph. *Communications of the ACM*, *16*, 575-577.

Büchner, A.G., Baumgarten, M., Anand, S.S., Mulvenna, M.D., & Hughes, J.G. (1999). Navigation pattern discovery from Internet data. *Proceedings of the KDD-99 Workshop on Web Usage Analysis and User Profiling*.

Cooley, Mobasher, B., & Srivastava, J. (1997). Web mining: Information and pattern discovery on the World Wide Web. *Proceedings of the Ninth IEEE International Conference on Tools with Artificial Intelligence* (pp. 558-567). New York: IEEE.

Cooley, R., Tan, P-N., & Srivastava, J. (1999). WebSIFT: The Web site information filter system. *Proceedings of the KDD-99 Workshop on Web Usage Analysis and User Profiling*.

Fisher, D. (1987). Knowledge acquisition via incremental conceptual clustering, *Machine Learning*, *2*, 139-172.

Fu, Y., Sandhu, K., & Shih, M-Y. (1999). Clustering of Web users based on access patterns. *Proceedings of the KDD-99 Workshop on Web Usage Analysis and User Profiling*.

Hanson, R., Stutz, J., & Cheeseman, P. (1991). *Bayesian classification theory* (Technical Report FIA-90-12-7-01). AI Branch, NASA Ames Research Center, CA.

IBM (1998). *Using the Intelligent Miner for Data, Version 2 Release 1.* IBM Corporation.

Joachims, T., Freitag, D., & Mitchell, T. (1997). WebWatcher: A tour guide for the World Wide Web. *Proceedings of the Fifteenth International Joint Conference on Artificial Intelligence* (pp. 770-777). San Francisco: Morgan Kaufmann Publishers.

Kohonen, T. (1997). *Self-organizing maps* (second edition). Berlin: Springer Verlag.

Langley, P. (1999). User modelling in adaptive interfaces. *Proceedings of the Seventh International Conference on User Modelling* (pp. 357-370). New York: Springer Verlag.

Orwant, J. (1995). Heterogeneous learning in the Doppelgänger user modelling system. *User Modelling and User-Adapted Interaction*, *4*, 107-130.

Paliouras, G., Papatheodorou, C., Karkaletsis, V., Spyropoulos, C.D., & Malaveta, V. (1998). Learning user communities for improving the services of information providers. *Proceedings of the Second European Conference on Digital Libraries* (pp. 367-384). Berlin: Springer Verlag.

Paliouras, G., Karkaletsis, V., Papatheodorou, C., & Spyropoulos, C.D. (1999). Exploiting learning techniques for the acquisition of user stereotypes and communities. *Proceedings of the Seventh International Conference on User Modeling* (pp. 169-178). New York: Springer Verlag.

Pazzani, M., & Billsus, D. (1997). Learning and revising user profiles: The identification of interesting Web sites. *Machine Learning, 27*, 313-331.

Perkowitz, M., & Etzioni, O. (1998). Adaptive Web sites: Automatically synthesizing Web pages. *Proceedings of the Fifteenth National Conference in Artificial Intelligence* (pp. 727-732). Cambridge, MA: AAAI Press.

Rich, E. (1983). Users are individuals: Individualizing user models. *International Journal of Man-Machine Studies*, *18*, 199-214.

Sahami, M. (1998). *Using machine learning to improve information access*. Doctoral dissertation, Department of Computer Science, Stanford University, Stanford, CA.

Schwartz, E. I. (1997). *Webonomics*. New York: Broadway books.