# Web Page Categorization and Feature Selection Using Association Rule and Principal Component Clustering *

Jerome Moore, Eui-Hong (Sam) Han,
Daniel Boley, Maria Gini, Robert Gross, Kyle Hastings,
George Karypis, Vipin Kumar, and Bamshad Mobasher

Department of Computer Science and Engineering / Army HPC Research Center
University of Minnesota, Minneapolis, MN 55455

**Abstract**

Clustering techniques have been used by many intelligent software agents in order to retrieve, filter, and categorize documents available on the World Wide Web. Clustering is also useful in extracting salient features of related web documents to automatically formulate queries and search for other similar documents on the Web. Traditional clustering algorithms either use *a priori* knowledge of document structures to define a distance or similarity among these documents, or use probabilistic techniques such as Bayesian classification. Many of these traditional algorithms, however, falter when the dimensionality of the feature space becomes high relative to the size of the document space. In this paper, we introduce two new clustering algorithms that can effectively cluster documents, even in the presence of a very high dimensional feature space. These clustering techniques. which are based on generalizations of graph partitioning, do not require pre-specified ad hoc distance functions, and are capable of automatically discovering document similarities or associations. We conduct several experiments on real Web data using various feature selection heuristics, and compare our clustering schemes to standard distance-based techniques, such as *hierarchical agglomeration clustering*, and Bayesian classification methods, *AutoClass*.

*This work was supported in part by NSF ASC-9634719 & CCR-9405380, by Army Research Office contract DA/DAAH04-95-1-0538, by Army High Performance Computing Research Center cooperative agreement number DAAH04-95-2-0003/contract number DAAH04-95-C-0008, the content of which does not necessarily reflect the position or the policy of the government, and no official endorsement should be inferred. Additional support was provided by the IBM Partnership Award, and by the IBM SUR equipment grant. Access to computing facilities was provided by AHPCRC, Minnesota Supercomputer Institute.

# 1 Introduction

The World Wide Web is a vast resource of information and services that continues to grow rapidly. Powerful search engines have been developed to aid in locating unfamiliar documents by category, contents, or subject. Relying on large indexes to documents located on the web, search engines determine the URLs of those documents satisfying a user's query. Often queries return inconsistent search results, with document referrals that meet the search criteria but are of no interest to the user.

While it may not be currently feasible to extract in full the meaning of an HTML document, intelligent software agents have been developed which extract semantic features from the words or structure of an HTML document. These extracted features are then employed to classify and categorize the documents. Clustering offers the advantage that *a priori* knowledge of categories is not needed, so the categorization process is unsupervised. The results of clustering could then be used to automatically formulate queries and search for other similar documents on the Web, or to organize bookmark files, or to construct a user profile.

In this paper, we present two new clustering algorithms based on graph partitioning and compare their performance against more traditional clustering algorithms used in information retrieval. Traditional clustering algorithms either define a distance or similarity among documents, or use probabilistic techniques such as Bayesian classification. Many of these algorithms, however, break down as the size of the document space, and hence, the dimensionality of the corresponding feature space increases. High dimensionality is characteristic of the type of information retrieval applications which are used to filter and categorize hypertext documents on the World Wide Web. In contrast, our partitioning-based algorithms do not rely on pre-specified or ad hoc notions of distance, and they perform well in the presence of a high dimensional space.

In the rest of the paper we describe our new clustering algorithms, the results of a number of experiments using different methods to select a set of features from the documents, and then compare the results of the different clustering algorithms. We show that partitioning clustering methods perform better than traditional distance based clustering.

# 2 Clustering Methods

Most of the existing approaches to document clustering are based on either probabilistic methods, or distance and similarity measures (see [9]). Distance-based methods such as $k$-means analysis, hierarchical clustering [12] and nearest-neighbor clustering [15] use a selected set of words (features) appearing in different documents as the dimensions. Each such feature vector, representing a document, can be viewed as a point in this multi-dimensional space.

There are a number of problems with clustering in a multi-dimensional space using traditional distance- or probability-based methods. First, it is not trivial to define a distance measure in this space. Feature vectors must be scaled to avoid skewing the result by different document lengths or possibly by how common a word is across many documents. Techniques such as TFIDF [20] have been proposed precisely to deal with some of these problems.

Second, the number of different words in all the documents can be very large. Distance-based schemes generally require the calculation of the mean of document clusters, which are often chosen initially at random. In a high dimensional space, the cluster means will do a poor job at separating documents. We have found that *hierarchical agglomeration clustering* (HAC) [7], based on distances between sample cluster means, does a poor job on our examples. Similarly, probabilistic methods such as Bayesian classification used in AutoClass [6, 21] do not perform well when the size of the feature space is much larger than the size of the sample set or may depend on the independence of the underlying features. Web documents suffer from both high dimensionality and high correlation among the feature values. We have found AutoClass has performed poorly on our examples.

Our proposed clustering algorithms, described below, are designed to efficiently handle very high dimensional spaces, and do not depend on the use of ad hoc distance or similarity metrics.

### 2.1 Association Rule Hypergraph Partitioning (ARHP)

In [11], a new method was proposed for clustering related items in transaction-based databases, such as supermarket bar code data, using association rules and hypergraph partitioning. This method first finds set of items that occur frequently together in transactions using association rule discovery methods [1]. These frequent item sets are then used to group items into hypergraph edges, and a hypergraph partitioning algorithm [13] is used to find the item clusters. The similarity among items is captured implicitly by the frequent item sets.

In document clustering, each document corresponds to an item and each possible feature corresponds to a transaction. The association rule discovery algorithm is used to find sets of documents with many features in common (*frequent item sets*). Each frequent item sets must satisfy a user-specified minimum support criteria which specifies a threshold on the minimum number of features common among documents in the set. A hypergraph [3] $H = (V, E)$ is formed with vertices $V$

consisting of the documents and hyperedges $E$ representing the frequent item sets. Hyperedges are edges that can connect more than 2 vertices. To each hyperedge we associate a weight, which is calculated as the average confidence of all the association rules involving the related documents of the hyperedge, where each individual confidence level is the conditional probability that a feature occurs in a document or group of documents given that it occurs in the remaining documents in that hyperedge.

Next, a hypergraph partitioning algorithm is used to partition the hypergraph such that the weight of the hyperedges that are cut by the partitioning is minimized. We are essentially minimizing the relations that are violated by partitioning the documents into different clusters. Similarly, this method can be applied to word clustering. In this setting, each word would correspond to an item and each document would correspond to a transaction. This method uses the *Apriori* algorithm [1] which has been shown to be very efficient in finding frequent item sets and HMETIS [13] which can partition very large hypergraphs (of size $>$ 100K nodes) in minutes on personal computers. Furthermore, the support criteria on frequent item sets can be used to filter out documents that are less likely to be related to other documents in each cluster.

### 2.2 Principal Component Analysis (PCA) Partitioning Algorithm

In the principal component algorithm, each document is represented by a feature vector of word frequencies, scaled to unit length. TFIDF scaling could be used, but simple scaling to unit length was found to achieve better clustering results in less time, at least on the data sets we tried. The algorithm proceeds by cutting the entire space of documents with a hyperplane passing through the overall arithmetic mean of the documents, and normal to the the principal direction (direction of maximum variance) for the document set. The documents are split into two separate groups by means of the hyperplane in a manner similar to a linear discriminant function, then each group is further split in the same manner. This is repeated as many times as desired, resulting in a tree-like hierarchy. The leaves of the tree are document clusters, each with a computed mean and principal direction. We use a *scatter value*, measuring the average distance from the documents in a cluster to the mean [7], to determine the next cluster to split at each stage.

The definition of the hyperplane is based on principal component analysis, similar to the Hotelling or Karhunen-Loeve Transformation [7]. We compute the principal direction as the leading eigenvector of the sample covariance matrix. This is the most expensive part, for which we use a fast Lanczos-based singular value solver [10].

# 3 Experimental Results

## 3.1 Experimental Setup

For the experiments we present here we selected 98 web pages in four broad categories: business and finance, electronic communication and networking, labor, and manufacturing. These pages were downloaded, labeled, and archived. The labeling facilitates an entropy calculation and subsequent references to any page were directed to the archive. This ensures a stable data sample since some pages are fairly dynamic in content.

| Experiment | Selection Criteria | Datasets Size | Comments |
|---|---|---|---|
| F1 | All words | 98x5623 | We select all words. |
| F2 | Quantile filtering | 98x619 | Quantile filtering selects the most freqently occuring words until the accumulated frequencies exceed a threshold of 0.25. In addition, we include all words from the partition that contributes the word that exceeds the threshhold. |
| F3 | Top 20+ words | 98x1239 | We select the 20 most frequently occuring words and include all words from the partition that contributes the 20th word. |
| F4 | Top 5+ words plus emphasized words | 98x1432 | We select the top 5+ words as in F3. This list is then augmented with any word that was emphasized in the html document. That is, words appearing in <TITLE>, <H1>, <H2>, <H3>, <I>, <BIG>, <STRONG>, or <EMPHASIZE> tags were added to the list. |
| F5 | Frequent item sets | 98x399 | We select words from the document word lists that appear in a-priori word clusters. That is, we use an object measure to identify important groups of words. |
| F6 | All words with text frequency > 1 | 98x2641 | We prune the words selected for F1 to exclude those occuring only once. |
| F7 | Top 20+ with text frequency > 1 | 98x1004 | We prune the words selected for F3 to exclude those occuring only once. |

Table 1: Setup of experiments.

The word lists from all documents were filtered with a stop-list and "stemmed" using Porter's suffix-stripping algorithm [18] as implemented by [8]. We derived seven experiments and clustered the documents using the four algorithms described earlier. Our objective is to reduce the dimensionality of the clustering problem and retain the important features of the documents. The seven experiments are distinguished by further selection rules, as shown in Table 1.

## 3.2 Evaluation of Clustering Results

Validating clustering algorithms and comparing performance of different algorithms is complex because it is difficult to find an objective measure of quality of clusters. We decided to use entropy [19] as a measure of goodness of the clusters. When a cluster contains documents from one class only, the entropy value is 0.0 for the cluster and when a cluster contains documents from many different classes, then entropy of the cluster is higher. The total entropy is calculated as the weighted sum of entropies of the clusters. We compare the results of the various experiments by comparing their entropy across algorithms and across feature selection methods (Fig. 1). Note that the hypergraph partitioning method does not cluster all the documents, so the entropy is computed only for the documents clustered.

Our experiments suggest that clustering methods based on partitioning seem to work best for this type of information retrieval applications, because (1) they do not depend in a choice of a distance function; (2) they do not require calculation of the mean of the clusters, and so the issue of having cluster means very close in space does not apply; (3) they are not sensitive to the dimensionality of the data sets. In particular, both the hypergraph partitioning method and the principle component methods performed much better than the traditional methods regardless of the feature selection criteria used.

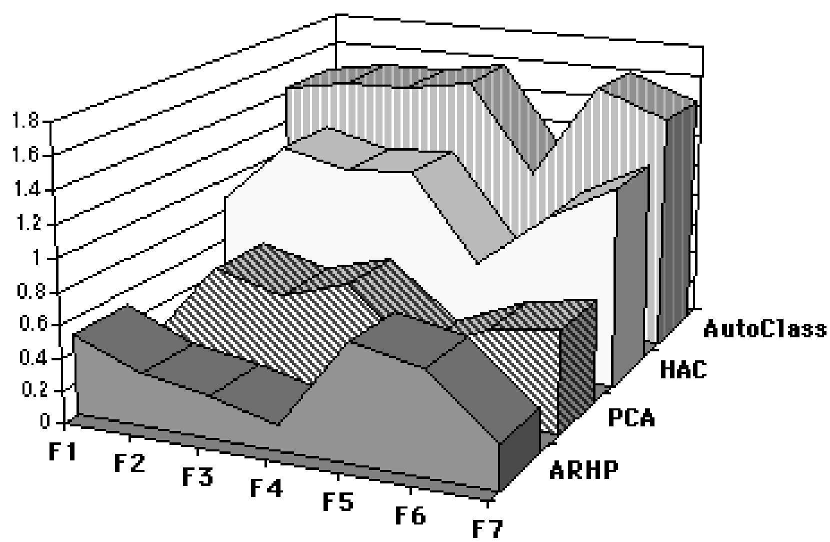


Figure 1: Entropy Comparison Across Algorithms and Feature Selection Methods.

As evident from the experiments the hypergraph partitioning method seems to benefit significantly from careful selection of a small number of representative features from each document. This is partly due to the fact that the method does not automatically take word frequencies within the text into account. The method worked best with F4 (top 5 plus HTML emphasized words) and also well with F3 (top 20 words). The worst results are with F5 (frequent item set). This makes sense, since the hypergraph partitioning includes the computation of the frequent item set as its first set. Applying this method twice may remove too much useful information. The principal component method works well with data sets that are not filtered based on text frequency. The best results are obtained with F1 (all words).

In general, AutoClass, HAC method, and Principal Component methods had similar behavior across the experiments in that they performed better when less filtering based on word frequencies was done. However, algorithms such as AutoClass and HAC become computationally prohibitive as the dimensionality is increased. It is clear from the F5 results, however, that filtering features based on frequent word sets (across documents) significantly improves the performance of traditional clustering algorithms, while significantly reducing dimensionality. This method of feature selection could therefore be very useful in a variety of information retrieval applications.

For any particular experiment, we can better judge the quality of the clustering by looking at the distribution of class labels among clusters. For example, Figure 2 shows the class distribution for the F4 experiment using the Hypergraph Partitioning algorithm. Full results (including the data sets) are available on the Web at `http://www.cs.umn.edu/~jmoore/wap2.html`.

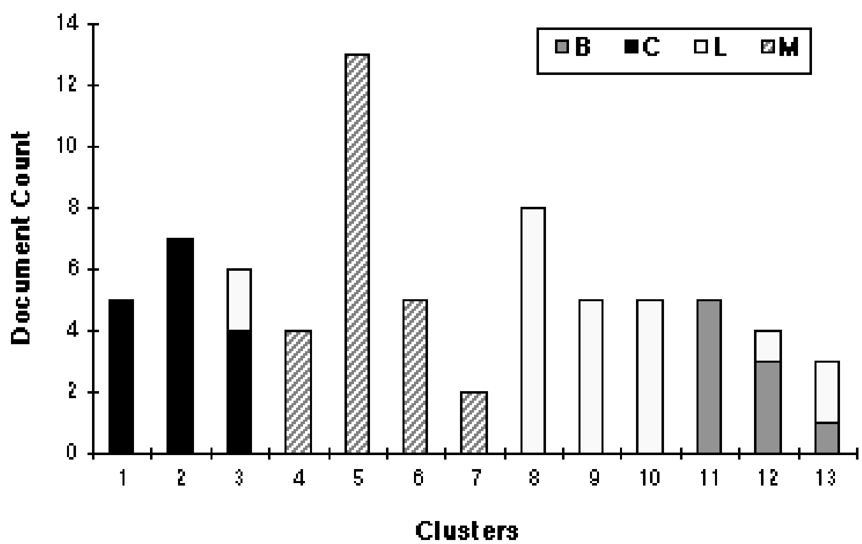


Figure 2: Distribution of class labels in F4 using ARHP Algorithm (B = Business, C = Communication, L = Labor, M = Manufacturing).

It should be noted that both PCA and ARHP were quite effective in isolating subcategories within each of the general categories. For example, in Figure 2, "labor-related" clusters 8, 9, and 10, correspond to the subcategories General Labor, Occupational Safety, and Affirmative Action, respectively.

## 4 Related Work

A number of Web agents use various information retrieval techniques [9] and characteristics of open hypertext Web documents to automatically retrieve, filter, and categorize these documents [5, 4, 16, 23, 22]. For example, HyPursuit [22] uses semantic information embedded in link structures as well as document content to create cluster hierarchies of hypertext documents and structure an information space. Pattern recognition methods and word clustering using the Hartigan's K-means partitional clustering algorithm are used in [23] to discover salient HTML document features (words) that can be used in finding similar HTML documents on the Web.

Syskill & Webert [17] represents an HTML page with a Boolean feature vector, and then uses naive Bayesian classification to find web pages that are similar, but for only a given single user profile. Also, Balabanovic [2] presents a system that uses a single well-defined profile to find similar web documents for a user. Candidate web pages are located using best-first search, comparing their word vectors against a user profile vector, and returning the highest -scoring pages. A TFIDF scheme [20] is used to calculate the word weights, normalized for document length. The system needs to keep a large dictionary and is limited to one user.

The Kohonen Self-Organizing Feature Map [14] is a neural network based scheme that projects high dimensional input data into a feature map of a smaller dimension such that the proximity relationships among input data are preserved. On data sets of very large dimensionality such as those discussed here, convergence could be slow, depending upon the initialization.

## 5 Conclusion

In this paper we have presented two new methods for clustering, namely, Association Rule Hypergraph Partitioning and Principal Component Partitioning, that are particularly suitable for the type of information retrieval applications discussed above. These methods do not depend on distance measures, and perform well in high dimensional spaces.

Our experiments suggest that both of these methods perform better than other traditional clustering algorithms regardless of the techniques used for feature selection. In particular, they both perform well, even when all of the features from each document are used in clustering. In addition, the experiments suggest that if the features selected are restricted to those present in frequent item sets, such as those derived from the Apriori Algorithm, then the traditional methods tend to perform better. It is also evident that, the hypergraph partitioning method may perform better, if the features selected include those words emphasized by document authors through the use of HTML tags.

Our future research plans include developing methods for incremental clustering or classification of documents after discovering an initial set of clusters. Furthermore, we plan to investigate the use of clustering techniques proposed here for word clustering. These word clusters can then be used to classify new documents or to search for related documents on the Web.

## References


[1] A. Agrawal, H. Mannila, R. Srikant, H. Toivonen, and A. Verkamo. Fast discovery of association rules. In U. Fayyad, G. Piatetsky-Shapiro, P. Smith, and R. Uthurusamy, editors, *Advances in Knowledge Discovery and Data Mining*, pages 307–328. AAAI/MIT Press, 1996.

[2] M. Balabanovic, Y. Shoham, and Y. Yun. An adaptive agent for automated Web browsing. *Journal of Visual Communication and Image Representation*, 6(4), 1995. http://www-diglib.stanford.edu/cgi-bin/WP/get/SIDL-WP-1995-0023.

[3] C. Berge. *Graphs and Hypergraphs.* American Elsevier, 1976.

[4] A. Z. Broder, S. C. Glassman, M. S. Manasse, and G. Zweig. Syntactic clustering of the web. In *Proc. of 6th International World Wide Web Conference*, 1997.

[5] C. Chang and C. Hsu. Customizable multi-engine search tool with clustering. In *Proc. of 6th International World Wide Web Conference*, 1997.

[6] P. Cheeseman and J. Stutz. Baysian classification (autoclass): Theory and results. In U. Fayyad, G. Piatetsky-Shapiro, P. Smith, and R. Uthurusamy, editors, *Advances in Knowledge Discovery and Data Mining*, pages 153–180. AAAI/MIT Press, 1996.

[7] R. O. Duda and P. E. Hart. *Pattern Classification and scene analysis.* Wiley, 1973.

[8] W. B. Frakes. Stemming algorithms. In W. B. Frakes and R. Baeza-Yates, editors, *Information Retrieval Data Structures and Algorithms*, pages 131–160. Prentice Hall, 1992.

[9] W. B. Frakes and R. Baeza-Yates. *Information Retrieval Data Structures and Algorithms.* Prentice Hall, Englewood Cliffs, NJ, 1992.

[10] G. Golub and C. Van Loan *Matrix Computations.* Johns Hopkins, 1996.

[11] E. Han, G. Karypis, V. Kumar, and B. Mobasher. Clustering based on association rule hypergraphs. In *Workshop on Research Issues on Data Mining and Knowledge Discovery*, pages 9–13, Tucson, Arizona, 1997.

[12] A. Jain and R. C. Dubes. *Algorithms for Clustering Data.* Prentice Hall, 1988.

[13] G. Karypis, R. Aggarwal, V. Kumar, and S. Shekhar. Multilevel hypergraph partitioning: Application in VLSI domain. In *Proceedings ACM/IEEE Design Automation Conference*, 1997.

[14] T. Kohonen. *Self-Organization and Associated Memory.* Springer-Verlag, 1988.

[15] S. Lu and K. Fu. A sentence-to-sentence clustering procedure for pattern analysis. *IEEE Transactions on Systems, Man and Cybernetics*, 8:381–389, 1978.

[16] Y. S. Maarek and I. B. Shaul. Automatically organizing bookmarks per content. In *Proc. of 5th International World Wide Web Conference*, 1996.

[17] M. Pazzani, J. Muramatsu, and D. Billsus. Syskill & Webert: Identifying interesting Web sites. In *National Conference on Artificial Intelligence*, pages 54–61, Aug. 1996. http://www.ics.uci.edu/ pazzani/RTF/AAAI.html.

[18] M. F. Porter. An algorithm for suffix stripping. *Program*, 14(3):130–137, 1980.

[19] J. R. Quinlan. *C4.5: Programs for Machine Learning.* Morgan Kaufmann, San Mateo, CA, 1993.

[20] G. Salton and M. J. McGill. *Introduction to Modern Information Retrieval.* McGraw-Hill, 1983.

[21] D. Titterington, A. Smith, and U. Makov. *Statistical Analysis of Finite Mixture Distributions.* John Wiley & Sons, 1985.

[22] R. Weiss, B. Velez, M. A. Sheldon, C. Nemprempre, P. Szilagyi, A. Duda, and D. K. Gifford. Hypursuit: A hierarchical network search engine that exploits content-link hypertext clustering. In *Seventh ACM Conference on Hypertext*, Mar. 1996. http://paris.lcs.mit.edu/ rweiss/.

[23] M. R. Wulfekuhler and W. F. Punch. Finding salient features for personal Web page categories. In *6th International World Wide Web Conference*, Apr. 1997. http://proceedings.www6conf.org/HyperNews/get/PAPER118.html.