

# Categorization of Unlabeled Documents driven by Word Weighting

Ning Kang  
ISE Department  
George Mason University  
nkang@gmu.edu

Carlotta Domeniconi  
ISE Department  
George Mason University  
carlotta@ise.gmu.edu

Daniel Barbará  
ISE Department  
George Mason University  
dbarbara@gmu.edu

## ABSTRACT

In text mining we often have to handle large document collections. The labeling of such large corpuses of documents is too expensive and impractical. Thus, there is a need to develop (unsupervised) clustering techniques for text data, where the distributions of words can vary significantly from one category to another.

The vector space model of documents easily leads to a 30000 or more dimensions. In such high dimensionality, the effectiveness of any distance function that equally uses all input features is severely compromised. Furthermore, it is expected that different words may have different degrees of relevance for a given category of documents, and a single word may have a different importance across different categories.

In this paper we first propose a global unsupervised feature selection approach for text, based on frequent itemset mining. As a result, each document is represented as a set of words that co-occur frequently in the given corpus of documents. We then introduce a locally adaptive clustering algorithm, designed to estimate (local) word relevance and, simultaneously, to group the documents.

We present experimental results to demonstrate the feasibility of our approach. Furthermore, the analysis of the weights credited to terms provide evidence that the identified keywords can guide the process of label assignment to clusters. We take into consideration both spam email filtering and general classification datasets. Our analysis of the distribution of weights in the two cases provides insights on how the spam problem distinguishes from the general classification case.

**Keywords:** Feature selection, feature relevance and weighting, subspace clustering, document categorization, spam emails.

## 1. INTRODUCTION

The clustering problem concerns the discovery of homogeneous groups of data according to a certain similarity measure. It has been studied extensively in statistics [3], ma-

chine learning [7, 21], and database communities [23, 12, 27].

Given a set of multivariate data, (partitional) clustering finds a partition of the points into clusters such that the points within a cluster are more similar to each other than to points in different clusters. The popular  $K$ -means or  $K$ -medoids methods compute one representative point per cluster, and assign each object to the cluster with the closest representative, so that the sum of the squared differences between the objects and their representatives is minimized. Finding a set of representative vectors for clouds of multi-dimensional data is an important issue in data compression, signal coding, pattern classification, and function approximation tasks.

Clustering suffers from the curse of dimensionality problem in high dimensional spaces. In high dimensional spaces, it is highly likely that, for any given pair of points within the same cluster, there exist at least a few dimensions on which the points are far apart from each other. As a consequence, distance functions that equally use all input features may not be effective. Furthermore, several clusters may exist in different subspaces, comprised of different combinations of features. In many real world problems, in fact, some points are correlated with respect to a given set of dimensions, and others are correlated with respect to different dimensions. Each dimension could be relevant to at least one of the clusters.

In text mining we often have to handle large document collections (e.g., World Wide Web documents). The labeling of such large corpuses of documents is too expensive and impractical. Thus, there is a need to develop (unsupervised) clustering techniques for text data, where the distributions of words can vary significantly from one category to another.

The most commonly used representation for documents is the so called Vector Space Model (VSM), or Bag of Words (BOWs). Such a word level representation of documents easily leads to a 30000 or more dimensions. In this high dimensionality, the effectiveness of any distance function that equally uses all input features is severely compromised. Furthermore, one would expect that different words might have different degrees of relevance for a given category of documents, and, at the same time, a single word might have a different importance across different categories. For example, the word *result* may be more important than the word *people* to discriminate medical related documents. Similarly, the word *chip* is likely to be more important for documents on electronics than for documents related to medicine. Note that each word in a selected dictionary might be relevant

for at least one of the categories. Thus, it may not always be feasible to prune off too many dimensions without incurring a loss of crucial information. A proper feature selection procedure should operate locally in input space.

In this paper we first propose a global unsupervised feature selection approach for text, based on frequent itemset mining. As a result, each document is represented as a *bag of frequent itemsets*, that is a set of words that co-occur frequently in the given corpus of documents. This step is applied initially to documents to reduce the number of features to a feasible dimensionality for clustering and local weighting of keywords. We then introduce a locally adaptive clustering algorithm, designed to estimate (local) word relevance and, simultaneously, to group the documents. Thus, this method achieves not only a clustering of the documents, but also the identification of cluster-dependent keywords. The analysis of such keywords allows to assign labels to clusters, and therefore to use the groups as a model for prediction.

## 1.1 Our Contribution

A preliminary version of this paper appeared in [18]. Here we present an extended derivation and motivation of our algorithms, additional experiments, and a thorough analysis of the results. In particular, the contributions of this paper are as follows:

1. We introduce an unsupervised feature (word) selection approach to handle multi-class classification of documents in absence of labels. The approach is based on the mining of frequent itemsets.
2. We derive and apply a locally adaptive clustering algorithm for documents. The output of our method is twofold: it achieves not only a clustering of the documents, but also the identification of cluster-dependent keywords via a continuous term-weighting mechanism.
3. The experimental results we present demonstrate the feasibility of our approach in terms of achieved accuracy measured against the ground truth. Furthermore, the analysis of the weights credited to terms provide evidence that the identified keywords can guide the process of label assignment to clusters. Thus, the resulting groups can be used as a model for prediction.
4. We take into consideration both spam email filtering and general classification datasets. Our analysis of the distribution of weights in the two cases provides insights and further understanding on the spam email problem, and how it distinguishes from the general classification case.

## 2. RELATED WORK

In [15] the authors discuss a hierarchical document clustering approach using frequent set of words. Their objective is to construct a hierarchy of documents for browsing at increasing levels of specificity of topics. The algorithm starts constructing, for each frequent itemset (i.e., set of words) in the whole document set, an initial cluster of all the documents that contain this itemset. Then, it proceeds making the clusters disjoint. To this extent a measure of goodness of a cluster for a document is used: a cluster is “good” for a document if there are many frequent items (with respect to

the whole document set) in the document that are also frequent within the cluster. Hence, each document is removed from all the initial clusters but the one that maximizes this measure of goodness. This stage gives a disjoint set of clusters, that is used to construct a tree of groups of documents. The tree is built bottom-up by choosing for each cluster  $C_k$  at a given level the unique parent (cluster) with the largest similarity score. By merging all documents in  $C_k$  into a single conceptual document, the similarity score between  $C_k$  and its candidate parents is measured using a criterion similar to the measure of goodness of a cluster for a document.

Local dimensionality reduction approaches for the purpose of efficiently indexing high dimensional spaces have been recently discussed in the database literature [19, 6, 25]. Applying global dimensionality reduction techniques when data are not globally correlated can cause significant loss of distance information, resulting in a large number of false positives and hence a high query cost. The general approach adopted by the authors is to find local correlations in the data, and perform dimensionality reduction on the locally correlated clusters individually. For example, in [6], the authors first construct spacial clusters in the original input space using a simple technique that resembles K-means. Principal component analysis is then performed on each spacial cluster individually to obtain the principal components.

In general, the efficacy of these methods depends on how the clustering problem is addressed in the first place in the original feature space. A potential serious problem with such techniques is the lack of data to locally perform PCA on each cluster to derive the principal components. Moreover, for clustering purposes, the new dimensions may be difficult to interpret, making it hard to understand clusters in relation to the original space.

The problem of finding different clusters in different subspaces of the original input space has been addressed in [2]. The authors use a density based approach to identify clusters. The algorithm (CLIQUE) proceeds from lower to higher dimensionality subspaces and discovers dense regions in each subspace. To approximate the density of the points, the input space is partitioned into cells by dividing each dimension into the same number  $\xi$  of equal length intervals. For a given set of dimensions, the cross product of the corresponding intervals (one for each dimension in the set) is called a *unit* in the respective subspace. A unit is dense if the number of points it contains is above a given threshold  $\tau$ . Both  $\xi$  and  $\tau$  are parameters defined by the user. The algorithm finds all dense units in each  $k$ -dimensional subspace by building from the dense units of  $(k-1)$ -dimensional subspaces, and then connects them to describe the clusters as union of maximal rectangles.

While the work in [2] successfully introduces a methodology for looking at different subspaces for different clusters, it does not compute a partitioning of the data into disjoint groups. The reported dense regions largely overlap, since for a given dense region all its projections on lower dimensionality subspaces are also dense, and they all get reported. On the other hand, for many applications such as customer segmentation and trend analysis, a partition of the data is desirable since it provides a clear interpretability of the results.

Recently [24], another density-based projective clustering algorithm (DOC) has been proposed. This approach pursues an optimality criterion defined in terms of density of

each cluster in its corresponding subspace. A Monte Carlo procedure is then developed to approximate with high probability an optimal projective cluster.

The work in [11] also addresses the problem of feature selection to find clusters hidden in high dimensional data. The authors search through feature subset spaces, evaluating each subset by first clustering in the corresponding subspace, and then evaluating the resulting clusters and feature subset using the chosen feature selection criterion. The two feature selection criteria investigated are the scatter separability used in discriminant analysis [14], and a maximum likelihood criterion. A sequential forward greedy strategy [14] is employed to search through possible feature subsets. We observe that dimensionality reduction is performed globally in this case. Therefore, the technique in [11] is expected to be effective when a data set contains some relevant features and some irrelevant (noisy) ones, across all clusters.

The problem of finding different clusters in different subspaces is also addressed in [1]. The proposed algorithm (PROjected CLUstering) seeks subsets of dimensions such that the points are closely clustered in the corresponding spanned subspaces. Both the number of clusters and the average number of dimensions per cluster are user-defined parameters. PROCLUS starts with choosing a random set of medoids, and then progressively improves the quality of medoids by performing an iterative hill climbing procedure that discards the 'bad' medoids from the current set. In order to find the set of dimensions that matter the most for each cluster, the algorithm selects the dimensions along which the points have the smallest average distance from the current medoid. In contrast to the PROCLUS algorithm, our method does not require to specify the average number of dimensions to be kept per cluster. For each cluster, in fact, *all* features are taken into consideration, but properly weighted. The PROCLUS algorithm is more prone to loss of information if the number of dimensions is not properly chosen.

In [22] the authors address the problem of feature weighting in  $K$ -means clustering. Each data point is represented as a collection of vectors, with "homogeneous" features within each measurement space. The objective is to determine one (global) weight value for each feature space. The optimality criterion pursued is the minimization of the (Fisher) ratio between the average within-cluster distortion and the average between-cluster distortion. However, the proposed method does *not* learn optimal weight values from the data. Instead, different weight value combinations are ran through a K-means-like algorithm, and the combination that results in the lowest Fisher ratio is chosen. We also observe that the weights are global, in contrast to ours which are local to each cluster.

Generative approaches have also been developed for local dimensionality reduction and clustering. The approach in [16] makes use of maximum likelihood factor analysis to model local correlations between features. The resulting generative model obeys the distribution of a mixture of factor analyzers. An expectation-maximization algorithm is presented for fitting the parameters of the mixture of factor analyzers. The choice of the number of factor analyzers, and the number of factors in each analyzer (that drives the dimensionality reduction) remain an important open issue for the approach in [16].

The work in [26] extends the single PCA model to a mix-

ture of local linear sub-models to capture nonlinear structure in the data. A mixture of principal component analyzers model is derived as a solution to a maximum-likelihood problem. An EM algorithm is formulated to estimate the parameters.

While the methods in [16, 26], as well as the standard mixture of Gaussians technique, are generative and parametric, our approach can be seen as an attempt to directly estimate from the data local correlations between features. Furthermore, both mixture models in [16, 26] inherit the soft clustering component of the EM update equations. On the contrary, LAC computes a partitioning of the data into disjoint groups. As previously mentioned, for many data mining applications a partitioning of the data is desirable since it provides a clear interpretability of the results. We finally observe that, while mixture of Gaussians models, with arbitrary covariance matrices, could in principle capture local correlations along any directions, lack of data to locally estimate full covariance matrices in high dimensional spaces is a serious problem in practice.

### 3. FEATURE SELECTION BASED ON FREQUENT ITEMSETS MINING

In [4] we introduced a feature selection algorithm for text, based on frequent itemsets mining. Our method (DocMine) addresses the categorization of documents (without labels) with an unknown number of classes, with the user interested in only one of them. This is a common scenario in information retrieval, such as content-based image retrieval, web-page classification, and document retrieval.

The method presented in [4] requires multiple sets of documents to be available (e.g., collections of documents retrieved by several search engines as result of a given query), and makes the assumption that relevant documents are more frequent in the majority of the sets. By computing the itemsets (words) that are frequent in the majority of the collections, it identifies positive features. The documents that contain the identified words are labeled as positive documents.

In this work we extend our unsupervised feature selection approach to handle multi-class classification problems in absence of labels. We no longer require the existence of multiple sets of documents.

Given a document, it is possible to associate with it a *bag of words* [17, 10, 20]. Specifically, we represent a document as a binary vector  $\mathbf{d} \in \mathbb{R}^N$ , in which each entry records if a particular word stem occurs in the text. The dimensionality  $N$  of  $\mathbf{d}$  is determined by the number of different terms in the corpus of documents (size of the *dictionary*), and each entry is indexed by a specific term.

Given a sample of unlabeled documents  $\{\mathbf{d}_i\}$  of different categories, we mine them to find the frequent itemsets that satisfy a given support level. In principle, the support level is driven by the target dimensionality of the data (to make the subsequent clustering step suitable). Each resulting itemset is a set of words that *co-occur frequently* in the given corpus of documents. We consider the union of such frequent items, and represent each document as a *bag of frequent itemsets*. The actual value of the entry is the frequency of the corresponding word in the document. This provides a suitable representation since it is *compact* (the level of compactness being driven by the support), and captures keywords that

co-occur frequently within each category. We observe that additional spurious (non discriminant) features may be selected by this process (e.g., words that are frequent in documents across classes). The subsequent locally adaptive clustering algorithm is designed to estimate word relevance and, simultaneously, to group the documents. Thus, it achieves not only a clustering of the documents, but also the identification of cluster-dependent keywords. The analysis of such keywords can assist the assignment of labels to clusters, and therefore the use of groups as a model for prediction.

#### 4. LOCALLY ADAPTIVE CLUSTERING

Here we derive our locally adaptive clustering algorithm. A preliminary version of this approach appeared in [9].

We define what we call *weighted cluster*. Consider a set of points in some space of dimensionality  $n$ . A *weighted cluster*  $C$  is a subset of data points, together with a vector of weights  $\mathbf{w} = (w_1, \dots, w_n)$ , such that the points in  $C$  are closely clustered according to the  $L_2$  norm distance weighted using  $\mathbf{w}$ . The component  $w_j$  measures the degree of correlation of points in  $C$  along feature  $j$ . The problem becomes now how to estimate the weight vector  $\mathbf{w}$  for each cluster in the data set.

In this setting, the concept of *cluster* is not based only on points, but also involves a weighted distance metric, i.e., clusters are discovered in spaces transformed by  $\mathbf{w}$ . Each cluster is associated with its own  $\mathbf{w}$ , that reflects the correlation of points in the cluster itself. The effect of  $\mathbf{w}$  is to transform distances so that the associated cluster is reshaped into a dense hypersphere of points separated from other data.

In traditional clustering, the partition of a set of points is induced by a set of *representative* vectors, also called *centroids* or *centers*. The partition induced by discovering weighted clusters is formally defined as follows.

**Definition:** Given a set  $S$  of  $D$  points  $\mathbf{x}$  in the  $n$ -dimensional Euclidean space, a set of  $k$  centers  $\{\mathbf{c}_1, \dots, \mathbf{c}_k\}$ ,  $\mathbf{c}_j \in \mathbb{R}^n$ ,  $j = 1, \dots, k$ , coupled with a set of corresponding weight vectors  $\{\mathbf{w}_1, \dots, \mathbf{w}_k\}$ ,  $\mathbf{w}_j \in \mathbb{R}^n$ ,  $j = 1, \dots, k$ , partition  $S$  into  $k$  sets  $\{S_1, \dots, S_k\}$ :

$$S_j = \{\mathbf{x} | (\sum_{i=1}^n w_{ji}(x_i - c_{ji})^2)^{1/2} < (\sum_{i=1}^n w_{li}(x_i - c_{li})^2)^{1/2}, \forall l \neq j\}, \quad (1)$$

where  $w_{ji}$  and  $c_{ji}$  represent the  $i$ th components of vectors  $\mathbf{w}_j$  and  $\mathbf{c}_j$  respectively (ties are broken randomly).

The set of centers and weights is *optimal* with respect to the Euclidean norm, if they minimize the error measure:

$$E_1(C, W) = \sum_{j=1}^k \sum_{i=1}^n (w_{ji} \frac{1}{|S_j|} \sum_{\mathbf{x} \in S_j} (c_{ji} - x_i)^2) \quad (2)$$

subject to the constraints  $\sum_i w_{ji} = 1 \forall j$ .  $C$  and  $W$  are  $(n \times k)$  matrices whose column vectors are  $\mathbf{c}_j$  and  $\mathbf{w}_j$  respectively, i.e.  $C = [\mathbf{c}_1 \dots \mathbf{c}_k]$  and  $W = [\mathbf{w}_1 \dots \mathbf{w}_k]$ . For short, we set

$$X_{ji} = \frac{1}{|S_j|} \sum_{\mathbf{x} \in S_j} (c_{ji} - x_i)^2,$$

where  $|S_j|$  is the cardinality of set  $S_j$ .  $X_{ji}$  represents the average distance from the centroid  $\mathbf{c}_j$  of points in cluster  $j$  along dimension  $i$ . The solution

$$(C^*, W^*) = \operatorname{argmin}_{(C, W)} E_1(C, W)$$

will discover one dimensional clusters: it will put maximal (i.e., unit) weight on the feature with smallest dispersion ( $X_{ji}$  within each cluster  $j$ , and zero weight on all other features. Our objective, instead, is to finding weighted multi-dimensional clusters, where the unit weight gets distributed among all features according to the respective dispersion of data within each cluster. One way to achieve this goal is to add the regularization term  $\sum_{i=1}^n w_{ji} \log w_{ji}$ <sup>1</sup>, which represents the negative entropy of the weight distribution for each cluster [13]. It penalizes solutions with maximal (unit) weight on the single feature with smallest dispersion within each cluster. The resulting error function is

$$E_2(C, W) = \sum_{j=1}^k \sum_{i=1}^n (w_{ji} X_{ji} + h w_{ji} \log w_{ji}) \quad (3)$$

subject to the same constraints  $\sum_i w_{ji} = 1 \forall j$ . The coefficient  $h \geq 0$  is a parameter of the procedure; it controls the strength of the incentive for clustering on more features. Increasing (decreasing) its value will encourage clusters on more (less) features. We can solve this constrained optimization problem by introducing the Lagrange multipliers  $\lambda_j$  (one for each constraint), and minimizing the resulting (unconstrained now) error function

$$E(C, W) = \sum_{j=1}^k \sum_{i=1}^n (w_{ji} X_{ji} + h w_{ji} \log w_{ji}) + \sum_{j=1}^k \lambda_j (1 - \sum_{i=1}^n w_{ji}) \quad (4)$$

For a fixed partition  $P$  and fixed  $c_{ji}$ , we compute the optimal  $w_{ji}^*$  by setting  $\frac{\partial E}{\partial w_{ji}} = 0$  and  $\frac{\partial E}{\partial \lambda_j} = 0$ . We obtain:

$$\frac{\partial E}{\partial w_{ji}} = X_{ji} + h \log w_{ji} + h - \lambda_j = 0 \quad (5)$$

$$\frac{\partial E}{\partial \lambda_j} = 1 - \sum_{i=1}^n w_{ji} = 0 \quad (6)$$

Solving equation (5) with respect to  $w_{ji}$  we obtain  $h \log w_{ji} = -X_{ji} + \lambda_j - h$ . Thus:

$$\begin{aligned} w_{ji} &= \exp(-X_{ji}/h + (\lambda_j/h) - 1) \\ &= \exp(-X_{ji}/h) \exp((\lambda_j/h) - 1) \\ &= \frac{\exp(-X_{ji}/h)}{\exp(1 - \lambda_j/h)}. \end{aligned}$$

Substituting this expression in equation (6):

$$\begin{aligned} \frac{\partial E}{\partial \lambda_j} &= 1 - \sum_{i=1}^n \frac{\exp(-X_{ji}/h)}{\exp(1 - \lambda_j/h)} \\ &= 1 - \frac{1}{\exp(-\lambda_j/h)} \sum_{i=1}^n \exp((-X_{ji}/h) - 1) = 0. \end{aligned}$$

Solving with respect to  $\lambda_j$  we obtain

$$\lambda_j = -h \log \sum_{i=1}^n \exp((-X_{ji}/h) - 1).$$

<sup>1</sup>Different regularization terms lead to different weighting schemes.

Thus, the optimal  $w_{ji}^*$  is

$$\begin{aligned} w_{ji}^* &= \frac{\exp(-X_{ji}/h)}{\exp(1 + \log(\sum_{i=1}^n \exp((-X_{ji}/h) - 1)))} \\ &= \frac{\exp(-X_{ji}/h)}{\sum_{i=1}^n \exp(-X_{ji}/h)} \end{aligned} \quad (7)$$

For a fixed partition  $P$  and fixed  $w_{ji}$ , we compute the optimal  $c_{ji}^*$  by setting  $\frac{\partial E}{\partial c_{ji}} = 0$ . We obtain:

$$\frac{\partial E}{\partial c_{ji}} = w_{ji} \frac{1}{|S_j|} 2 \sum_{\mathbf{x} \in S_j} (c_{ji} - x_i) = \frac{2w_{ji}}{|S_j|} (|S_j|c_{ji} - \sum_{\mathbf{x} \in S_j} x_i) = 0.$$

Solving with respect to  $c_{ji}$  gives

$$c_{ji}^* = \frac{1}{|S_j|} \sum_{\mathbf{x} \in S_j} x_i. \quad (8)$$

Solution (7) puts increased weight on features along which the dispersion  $X_{ji}$  is smaller, within each cluster. The degree of this increase is controlled by the value  $h$ . Setting  $h = 0$ , places all weight on the feature  $i$  with smallest  $X_{ji}$ , whereas setting  $h = \infty$  forces all features to be given equal weight for each cluster  $j$ . By setting  $E_0(C) = \frac{1}{n} \sum_{j=1}^k \sum_{i=1}^n X_{ji}$ , we can formulate this result as follows.

**Proposition:** When  $h = 0$ , the error function  $E_2$  (3) reduces to  $E_1$  (2); when  $h = \infty$ , the error function  $E_2$  reduces to  $E_0$ .

#### 4.1 Locally Adaptive Clustering Algorithm

We need to provide a search strategy to find a partition  $P$  that identifies the solution clusters. Our approach progressively improves the quality of initial centroids and weights, by investigating the space near the centers to estimate the dimensions that matter the most. Specifically, we proceed as follows.

We start with *well-scattered* points in  $S$  as the  $k$  centroids: we choose the first centroid at random, and select the others so that they are far from one another, and from the first chosen center. We initially set all weights to  $1/n$ . Given the initial centroids  $\mathbf{c}_j$ , for  $j = 1, \dots, k$ , we compute the corresponding sets  $S_j$  as given in the definition above. We then compute the average distance  $X_{ji}$  along each dimension from the points in  $S_j$  to  $\mathbf{c}_j$ . The smaller  $X_{ji}$  is, the larger is the correlation of points along dimension  $i$ . We use the value  $X_{ji}$  in an exponential weighting scheme to credit weights to features (and to clusters), as given in equation (7). The exponential weighting is more sensitive to changes in local feature relevance [5] and gives rise to better performance improvement. Note that the technique is centroid-based because weightings depend on the centroid. The computed weights are used to update the sets  $S_j$ , and therefore the centroids' coordinates as given in equation (8). The procedure is iterated until convergence is reached. The resulting algorithm, that we call LAC (Locally Adaptive Clustering), is summarized in the following.

**Input:**  $D$  points  $\mathbf{x} \in R^n$ ,  $k$ , and  $h$ .

1. Start with  $k$  initial centroids  $\mathbf{c}_1, \mathbf{c}_2, \dots, \mathbf{c}_k$ ;
2. Set  $w_{ji} = 1/n$ , for each centroid  $\mathbf{c}_j$ ,  $j = 1, \dots, k$  and each feature  $i = 1, \dots, n$ ;
3. For each centroid  $\mathbf{c}_j$ , and for each point  $\mathbf{x}$ :

- Set  $S_j = \{\mathbf{x} | j = \arg \min_l L_w(\mathbf{c}_l, \mathbf{x})\}$ , where  $L_w(\mathbf{c}_l, \mathbf{x}) = (\sum_{i=1}^n w_{li}(c_{li} - x_i)^2)^{1/2}$ ;

#### 4. Compute new weights.

For each centroid  $\mathbf{c}_j$ , and for each feature  $i$ :

- Set  $X_{ji} = \sum_{\mathbf{x} \in S_j} (c_{ji} - x_i)^2 / |S_j|$ ;

$$\text{Set } w_{ji} = \frac{\exp(-X_{ji}/h)}{\sum_{i=1}^n \exp(-X_{ji}/h)};$$

#### 5. For each centroid $\mathbf{c}_j$ , and for each point $\mathbf{x}$ :

- Recompute  $S_j = \{\mathbf{x} | j = \arg \min_l L_w(\mathbf{c}_l, \mathbf{x})\}$ ;

#### 6. Compute new centroids.

Set  $\mathbf{c}_j = \frac{\sum_{\mathbf{x} \in S_j} \mathbf{x}}{|S_j|}$ , for each  $j = 1, \dots, k$ , where  $1_S(\cdot)$  is the indicator function of set  $S$ ;

#### 7. Iterate 3,4,5,6 until convergence.

The sequential structure of the LAC algorithm is analogous to the mathematics of the *EM* algorithm [8]. The hidden variables are the assignments of the points to the centroids. Step 3 constitutes the *E* step: it finds the values of the hidden variables  $S_j$  given the previous values of the parameters  $w_{ji}$  and  $c_{ji}$ . The following step (*M* step) consists in finding new matrices of weights and centroids that minimize the error function with respect to the current estimation of hidden variables. It can be shown that the LAC algorithm converges to a local minimum of the error function (4). The running time of one iteration is  $O(kDn)$ .

## 5. SUBSPACE CLUSTERING FOR TEXT

Our overall approach consists of the following steps:

1. Preprocess the documents by eliminating stop and rare words, and by stemming words to their root source;
2. Apply our global unsupervised feature selection approach based on frequent itemset mining. As a result, we obtain documents represented as bag of frequent itemsets;
3. Apply our locally adaptive clustering algorithm to estimate local word relevance and, simultaneously, to group documents. As a result, we obtain a clustering of the documents, and the identification of cluster-dependent keywords.

## 6. EXPERIMENTAL RESULTS

### 6.1 Datasets and Preprocessing

In our experiments we used the following datasets.

**Email-1431.** This email dataset consists of 1431 emails, falling into three categories. The categories are: conference (370), jobs (272), and spam (789). This dataset is created by Finn Aarup Nielsen. The original size of the dictionary is 38713. We run two different experiments with this dataset. In one case we consider a 2-class classification problem by merging the conference and jobs mails into one group (non-spam). In the second case we keep the three categories separate.

**Ling-Spam.** This dataset is a mixture of spam messages (453) and messages (561) sent via the linguist list, a moderated (hence, spam-free) list about the profession and science of linguistics. The original size of the dictionary is 24627.

**Table 1: Results for Email-1431 (Spam (789) - Non Spam (642))**

S	N	n	D	Ave Err	Min Err	K-means
5%	9210	791	1431	$2.0 \pm 0.3$	1.7	45.0
7%	9210	519	1431	$2.0 \pm 0.5$	1.3	45.0
10%	9210	285	1431	$2.0 \pm 0.4$	1.5	45.0

**20 Newsgroups.** This dataset is a collection of 20,000 messages collected from 20 different netnews newsgroups. One thousand messages from each of the twenty newsgroups were chosen at random and partitioned by newsgroup name. In our experiments we consider the following categories: Medical (990), Electronics (981), Autos (990), and Space (987). We consider two 2-class classification problems: (1) electronics and medical classes (the original size of the dictionary in this case is 24546); (2) autos and space classes (the original size of the dictionary in this case is 22820).

**Classic3.** This dataset is a collection of abstracts from three categories: MEDLINE (1033 abstracts from medical journals), CISI (1460 abstracts from IR papers); CRANFIELD (1399 abstracts from aerodynamics papers). (We used a preprocessed version of this dataset with  $N = 4836$ .)

The documents in each dataset were preprocessed by eliminating stop words (based on a stop words list), and stemming words to their root source. In addition, rare words that appeared in less than four documents were also removed. After the initial global feature selection step, we use as feature values for the vector space model the relative frequency of the selected words (frequent itemsets) in the corresponding document.

## 6.2 Results

Tables 1-6 report the results we obtained for the six problems under consideration. Each table includes: the support values tested ( $S$ ), the dimensionality of the data after the preprocessing step ( $N$ ), the dimensionality of the data after feature selection based on frequent itemset mining ( $n$ ), the total number of documents ( $D$ ) (as well as the number of documents per class), the average error rate computed over nine runs of LAC for  $1/h = 1, \dots, 9$  (along with the standard deviations), the minimum error rate over such nine runs, and (as baseline comparison) the error rate of K-means. Error rates are computed according to the confusion matrices based on the ground truth labels.

For increasing support values, and therefore decreasing number of selected features, we can observe an increasing trend for the minimum error rates. Increasing support values do not always result in increasing average error rates, due to the fluctuations of the error with respect to different  $h$  values. Figures 1-6 show the error rates as a function of the  $h$  parameter values given in input to the LAC algorithm (for the different support values tested). In general, lower error rates were achieved for larger  $h$  values, which favor multi-dimensional clusters. As expected, the optimal dimensionality depends on the dataset. Particularly low error rates are achieved for the three problems on spam emails, and for a wide range of dimensionalities. K-means often fails to detect any structure in the data, and provides error rates close or above 45%. It performs better on the Newsgroups and Classic3 datasets for lower dimensionalities.

## 6.3 Analysis of Weights

**Table 2: Results for Email-1431 (Spam (789) - Conference (370) - Job (272))**

S	N	n	D	Ave Err	Min Err	K-means
5%	9210	791	1431	$10.8 \pm 7.8$	4.0	44.9
7%	9210	519	1431	$15.9 \pm 6.4$	7.0	44.9
10%	9210	285	1431	$15.0 \pm 5.8$	8.8	44.9

**Table 3: Results for Ling-Spam (Spam (453) - Non Spam (561))**

S	N	n	D	Ave Err	Min Err	K-means
5%	5456	553	1014	$6.4 \pm 1.5$	4.6	44.6
6%	5456	439	1014	$6.7 \pm 1.2$	4.6	44.6
7%	5456	350	1014	$12.2 \pm 12.1$	5.2	44.6
8%	5456	287	1014	$5.5 \pm 0.9$	4.0	44.6
9%	5456	227	1014	$10.8 \pm 12.7$	5.5	44.5
10%	5456	185	1014	$6.6 \pm 0.6$	5.3	44.5

**Table 4: Results for NewsGroups (Electronic (981) - Medical (990))**

S	N	n	D	Ave Err	Min Err	K-means
1%	6217	1359	1971	$11.5 \pm 2.4$	9.5	49.6
2%	6217	583	1971	$18.1 \pm 11.8$	13.5	49.7
3%	6217	321	1971	$21.0 \pm 9.5$	16.8	49.6
4%	6217	201	1971	$21.8 \pm 0.4$	20.8	49.7
5%	6217	134	1971	$29.1 \pm 7.5$	23.3	49.6

**Table 5: Results for NewsGroups (Autos (990) - Space (987))**

S	N	n	D	Ave Err	Min Err	K-means
1%	6219	1349	1977	$30.6 \pm 11.3$	14.2	48.4
2%	6219	631	1977	$19.8 \pm 2.9$	14.6	21.2
3%	6219	366	1977	$20.6 \pm 3.2$	15.9	12.0
4%	6219	248	1977	$20.1 \pm 1.0$	18.2	32.7
5%	6219	166	1977	$23.7 \pm 2.9$	18.4	17.7

**Table 6: Results for Classic3 (Medline (1033) - Cranfield (1399) - Cisi (1460))**

S	N	n	D	Ave Err	Min Err	K-means
1%	4836	974	3892	$20.6 \pm 11.4$	4.0	23.4
2%	4836	584	3892	$11.8 \pm 7.3$	5.9	25.0
3%	4836	395	3892	$25.1 \pm 9.7$	7.6	28.6
4%	4836	277	3892	$23.7 \pm 10.4$	9.0	29.2
5%	4836	219	3892	$21.2 \pm 10.6$	10.9	29.6

The analysis of the weights credited to words provides some insights on the nature of the spam email filtering problem and the general classification case. As Figures 7-12 show, the selected keywords (and in particular those that receive largest weight values) are representative of the underlying categories, which provides evidence that our global feature selection method successfully retains discriminant words. In addition, our subspace clustering technique is capable of further sifting the most relevant ones, while discarding the additional spurious words.

Let us consider the distribution of weights obtained for the Email-1431 dataset. Figure 7 shows the weight values and corresponding keywords for the two class case (the non-spam class corresponds to both conference and jobs emails). Here we plot the top words that received highest weight for each class. We observe that words reflecting the topic of a category receive a larger weight in the *other class*. For example, the words “free”, “money”, “sales”, “marketing”, and “order” get a larger weight in the non-spam class (their weights in the spam category are very close to zero, which cause the corresponding bar not to show up in the plot). Similarly, the words “conference”, “applications”, “papers”, “science”, “committee”, “institute”, “neuroscience”, etc receive larger weights within the spam category. The weights for these words in the non-spam class are very close to zero. While surprising at first, this trend may be due to the nature of the spam and non-spam email distributions. Each of these two categories is actually a combination of subclasses. The non-spam class in this case is the union of conference and jobs emails (by construction). Likewise, the spam messages can be very different in nature, and therefore different in their word content. As a consequence, the dispersion of feature values for words reflecting the general topic of a category is larger within the same category than in the other one (e.g., the word *money* has a wider range of relative frequency values within the class spam than within the class non-spam). Since the weights computed by the LAC algorithm are inversely proportional to a measure of such spread of values (i.e.,  $X_{ji}$ ), we obtain the trend shown in Figure 7. This analysis can be interpreted as the fact that the absence of a certain term is a characteristic shared across the emails of a given category; whereas the presence of certain keywords shows a larger variability across emails of a given category.

A similar behavior is observed for the same Email-1431 dataset when the three classes are kept separate. The results for this case are shown in Figure 8 (top plot). For example, words (typical of the conference topic) such as “research”, “conference”, “applications”, “papers”, “science”, “invited”, “committee”, and “submit” receive larger weights within the spam category (and the job category as well). Similarly the words (typical of spams) “money”, “business”, “sales”, “offer”, and “credit” receive larger weights in the categories conference and job. The bottom plot of Figure 8 shows the (within-cluster) standard deviations for the features displayed in the top plot. As expected, the spread of values for “conference words” is smaller in spam messages, which results in larger weights for the same words within the spam cluster. Similarly, “spam words” such as “sales” and “offer” manifest a much larger spread of values within the spam cluster. We emphasize that, in general, the relative distribution of feature values within the various clusters has

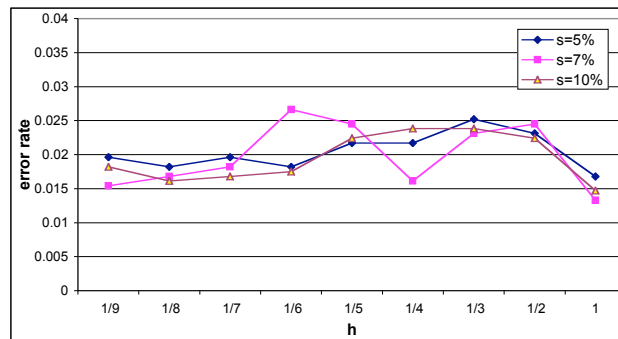


Figure 1: Email-1431 dataset (two classes): Error rate vs.  $h$  values. ( $s$ = support value.)

an inter-cluster effect, and thus the estimation of a given cluster’s spread affects the estimation of others’ as well.

Results for the Ling-Spam dataset are shown in Figure 9. In this case keywords receive largest weights within the representative class (e.g., “linguistic”, “theory”, and “abstract” for the non-spam class; “free”, “advertise”, and “win” for the spam class). Thus, this dataset shows an opposite relationship between keywords and categories with respect to the Email-1431. Nevertheless, we observe that the Ling-Spam data may not reflect the underlying category distributions of real spam and non-spam messages. This is because, by construction, the non-spam class contains only messages about the profession and science of linguistics. As such, the keywords representative of the topic “linguistics” may have a compact support (a small spread), and therefore receive larger weights as expected for the general classification case. A similar behavior is observed for the Newsgroups dataset (electronics and medical categories) (see Figure 10). The collections of terms receiving largest feature relevance weights in each cluster reflect the topic of that category. This is indeed expected in a typical categorization problem.

Results for the Newsgroups dataset (autos and space) are shown in Figure 11. For these data, the large majority of selected keywords belong to the “auto” domain (e.g., “engine”, “dealer”, “auto”, “driver”, “wheel”, etc.). As a consequence, the cluster of documents about “space” has a limited spread along the corresponding feature axes. This is the reason for having generally larger weight values credited to words within the cluster “space”.

Figure 12 shows the results for the Classic3 dataset. As for the Email-1431 dataset, words reflecting the topic of a category receive a larger weight in the other classes (e.g., the information retrieval stems “librarian”, “retriev”, “document”, “abstract” etc, receive larger weights in the crane field and medline categories than in the cisi; likewise, the stems “pressur”, “experiment” and “lead” receive larger weights in the cisi class). This result is corroborated by the plot showing the standard deviations (Figure 12, bottom plot). As for the spam email dataset, we expect the three categories (collections of medical journals, IR papers and aerodynamics papers) to contain combination of subclasses.

As an example, we also report in Figure 13 the trend of the weight values for the Email-1431 dataset. The weight values show an exponential decreasing trend. Few features account for the total (unit) weight value.

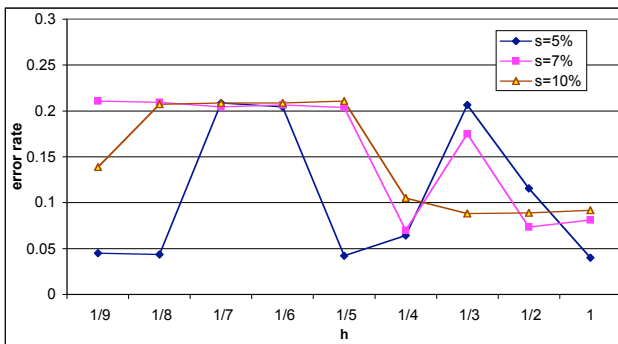


Figure 2: Email-1431 dataset (three classes): Error rate vs.  $h$  values. ( $s$ = support value.)

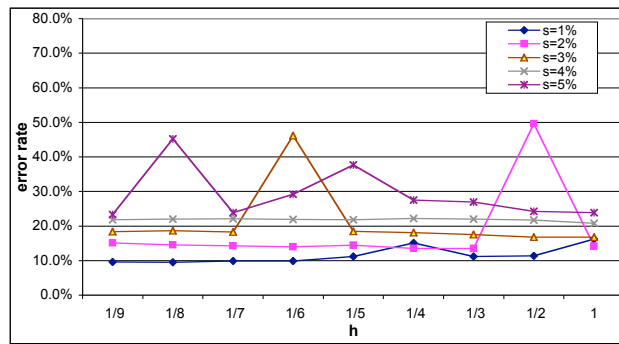


Figure 4: Newsgroups dataset (electronics-medical): Error rate vs.  $h$  values. ( $s$ = support value.)

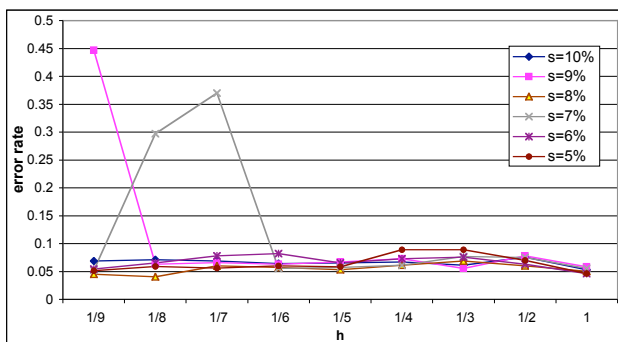


Figure 3: Ling-spam dataset: Error rate vs.  $h$  values. ( $s$ = support value.)

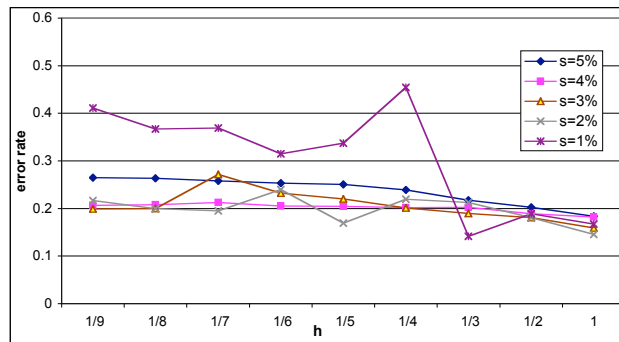


Figure 5: Newsgroups dataset (autos-space): Error rate vs.  $h$  values. ( $s$ = support value.)

## 7. CONCLUSIONS

We have introduced an unsupervised feature selection approach, based on frequent itemset mining, to handle multi-class classification of documents in absence of labels. In addition, we have derived a locally adaptive clustering algorithm that provides a clustering of the documents and the identification of cluster-dependent keywords via a continuous term-weighting mechanism. Our experimental results demonstrate the feasibility of our approach in terms of achieved accuracy measured against the ground truth. We have shown that the selected keywords are representative of the underlying categories, which provides evidence that our global feature selection method successfully retains discriminant words. Moreover, our subspace clustering technique is capable of further sifting the most relevant ones, while discarding the additional spurious words.

The analysis of weights and spread of feature values can be informative of the nature of the categorization problem. When categories are combination of subclasses, words reflecting the topic of a category receive larger weights in other classes. On the other hand, when documents of a given topic are “sufficiently focused”, the keywords representative of the topic receive larger weights. Relevant keywords, combined with the associated weight values can be used to provide short summaries for clusters and to automatically annotate documents (e.g., for indexing purposes).

In our future work we plan to consider frequent itemsets

as individual features. Thus, the LAC algorithm would estimate the relevance associated to group of keywords. In this context the corresponding weights would measure the correlation between the itemsets and the clusters, which corresponds to the definition of lift of the itemset, a widely accepted interestingness measure for association rules. We plan to conduct experiments with this representation, and compare the results with those presented in this paper.

As shown in our experiments, the clustering results achieved by the LAC algorithm depend on the value of the input parameter value  $h$ . To generate robust and stable solutions, new consensus subspace clustering methods are under investigation by the authors. The major difficulty is to find a consensus partition from the output partitions of the contributing clusterings, so that an “improved” overall clustering of the data is obtained. Since we are dealing with weighted clusters, proper consensus functions that make use of the weight vectors associated with the clusters will be investigated.

## 8. REFERENCES

- [1] C. Aggarwal, C. Procopiuc, J. L. Wolf, P. Yu, and J. S. Park. Fast algorithms for projected clustering. In *Proceedings of the ACM SIGMOD Conference on Management of Data*, 1999.
- [2] R. Agrawal, J. Gehrke, D. Gunopulos, and P. Raghavan. Automatic subspace clustering of high

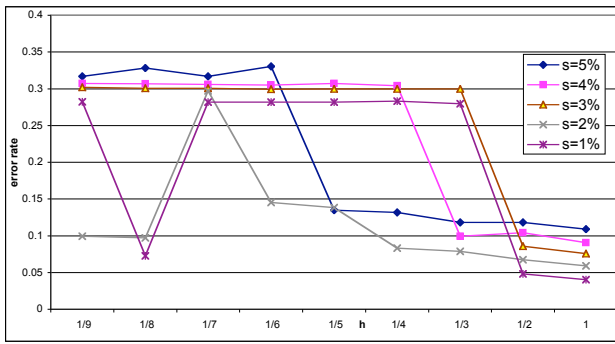


Figure 6: Classic3 dataset: Error rate vs.  $h$  values. ( $s$  = support value.)

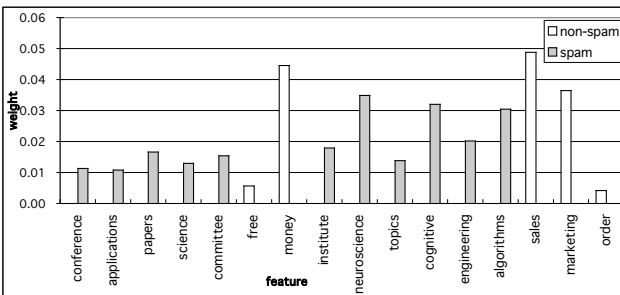


Figure 7: Email-1431 (two classes): Keywords and corresponding weight values ( $s = 10\%$ ,  $h = 1/9$ ).

dimensional data for data mining applications. In *Proceedings of the ACM SIGMOD Conference on Management of Data*, 1998.

- [3] P. Arabie and L. Hubert. *An Overview of Combinatorial Data Analysis. Clustering and Classification*. World Scientific Publications, 1996.
- [4] D. Barbará, C. Domeniconi, and N. Kang. Classifying documents without labels. In *Proceedings of the SIAM International Conference on Data Mining*, 2004.
- [5] L. Bottou and V. Vapnik. Local learning algorithms. *Neural Computation*, 4(6):888–900, 1992.
- [6] K. Chakrabarti and S. Mehrotra. Local dimensionality reduction: A new approach to indexing high dimensional spaces. In *Proceedings of the VLDB Conference*, 2000.
- [7] P. Cheeseman and J. Stutz. *Bayesian Classification (autoclass): Theory and Results. Advances in Knowledge Discovery and Data Mining*. AAAI/MIT Press, Cambridge, 1996.
- [8] A. Dempster, N. Laird, and D. B. Rubin. Maximum likelihood from incomplete data via the em algorithm. *Journal of the Royal Statistical Society*, 39(1):1–38, 1977.
- [9] C. Domeniconi, D. Papadopoulos, D. Gunopulos, and S. Ma. Subspace clustering of high dimensional data. In *Proceedings of the SIAM International Conference on Data Mining*, 2004.
- [10] S. T. Dumais, T. Letsche, M. L. Littman, and T. Landauer. Automatic cross-language retrieval using

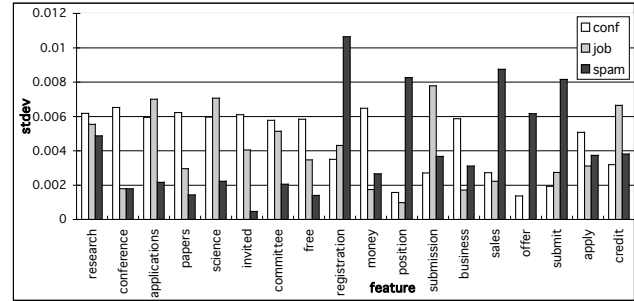
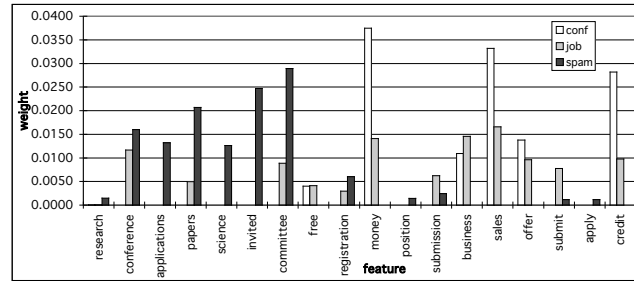


Figure 8: Email-1431 dataset (three classes): (Top) Keywords and corresponding weight values ( $s = 10\%$ ,  $h = 1/2$ ); (Bottom) Standard deviation of feature values.

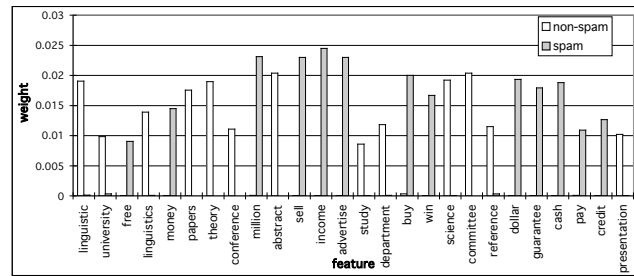


Figure 9: Ling-spam dataset: Keywords and corresponding weight values ( $s = 8\%$ ,  $h = 1/8$ ).

latent semantic indexing. In *AAAI Spring Symposium on Cross-Language Text and Speech Retrieval*, 1997.

- [11] J. Dy and C. Brodley. Feature subset selection and order identification for unsupervised learning. In *Proceedings of the International Conference on Machine Learning*, 2000.
- [12] M. Ester, H. Kriegel, and X. Xu. A database interface for clustering in large spatial databases. In *Proceedings of the International Conference on Knowledge Discovery in Databases and Data Mining*, 1995.
- [13] J. Friedman and J. Meulman. Clustering objects on subsets of attributes. In *Technical Report, Stanford University*, September 2002.
- [14] K. Fukunaga. *Introduction to Statistical Pattern Recognition*. Academic Press, 1978.
- [15] B. Fung, K. Wang, and M. Ester. Hierarchical document clustering using frequent itemsets. In *Proceedings of the SIAM International Conference on*

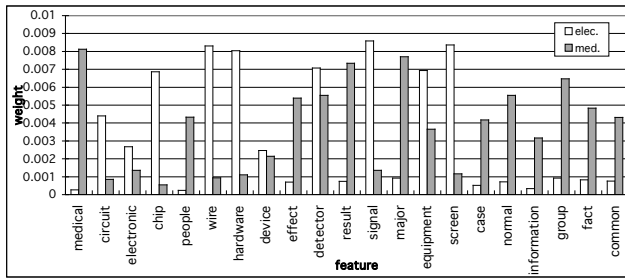


Figure 10: Newsgroups dataset (electronics-medical): Keywords and corresponding weight values ( $s=3\%$ ,  $h=1/9$ ).

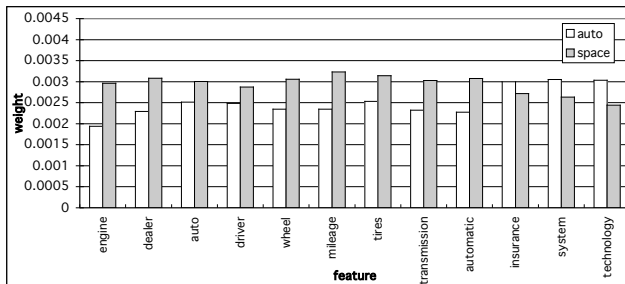


Figure 11: Newsgroups dataset (autos-space): Keywords and corresponding weight values ( $s = 3\%$ ,  $h = 1$ ).

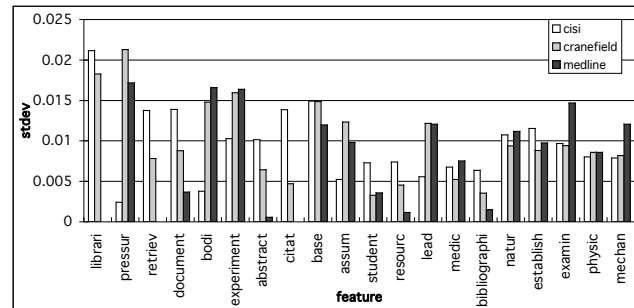
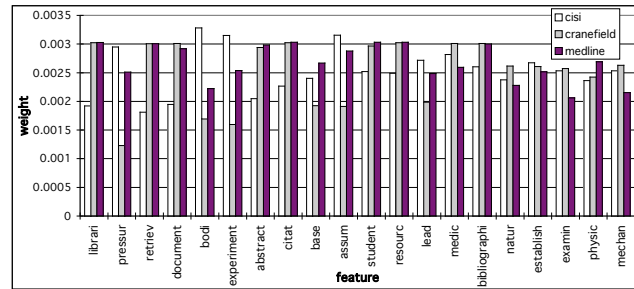


Figure 12: Classic3 dataset: (Top) Keywords and corresponding weight values ( $s = 3\%$ ,  $h = 1$ ); (Bottom) Standard deviation of feature values.

*Data Mining*, 2003.

- [16] Z. Ghahramani and G. Hinton. The em algorithm for mixtures of factor analyzers. In *Technical Report CRG-TR-96-1, Department of Computer Science, University of Toronto*, 1996.
- [17] T. Joachims. Text categorization with support vector machines. In *Proceedings of the International Conference on Machine Learning*, 1998.
- [18] N. Kang, C. Domeniconi, and D. Barabará. Categorization and Keyword Identification of Unlabeled Documents. In *Proceedings of the IEEE International Conference on Data Mining*, 2005.
- [19] E. Keogh, K. Chakrabarti, S. Mehrotra, and M. Pazzani. Locally adaptive dimensionality reduction for indexing large time series databases. In *Proceedings of the ACM SIGMOD Conference on Management of Data*, 2001.
- [20] E. Leopold and J. Kindermann. Text categorization with support vector machines. *Machine Learning*, 46:423–444, 2002.
- [21] R. Michalski and R. Stepp. *Learning from Observation: Conceptual Clustering. Machine Learning: An Artificial Intelligence Approach*. R. S. Michalski, J. G. Carbonell and T. M. Mitchell, Editors, 1996.
- [22] D. S. Modha and W. S. Spangler. Feature Weighting in k-Means Clustering. *Machine Learning*, 52(3), 2003.
- [23] R. Ng and J. Han. Efficient and effective clustering methods for spatial data mining. In *Proceedings of the VLDB Conference*, 1994.
- [24] C. Procopiuc, M. Jones, P. Agarwal, and T. Murali. A monte carlo algorithm for fast projective clustering. In *Proceedings of the ACM SIGMOD Conference on Management of Data*, 2002.
- [25] A. Thomasian, V. Castelli, and C. Li. Clustering and singular value decomposition for approximate indexing in high dimensional spaces. In *Proceedings of the CIKM Conference*, 1998.
- [26] M. E. Tipping and C. M. Bishop. Mixtures of principal component analyzers. *Neural Computation*, 1(2):443–482, 1999.
- [27] T. Zhang, R. Ramakrishnan, and M. Livny. Birch: An efficient data clustering method for very large databases. In *Proceedings of the ACM SIGMOD Conference on Management of Data*, 1996.

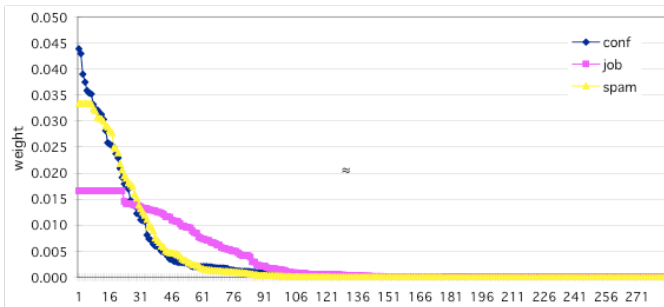
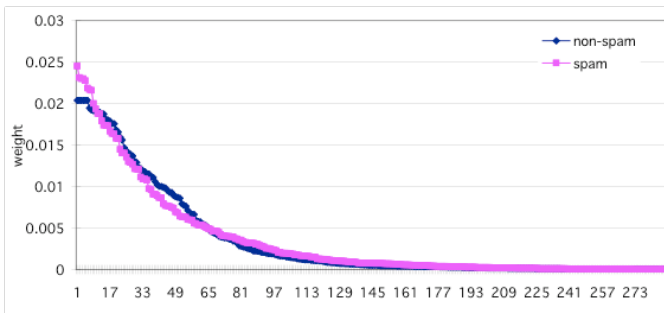


Figure 13: Email-1431 dataset: Trend of weight values (Top) Two classes; (Bottom) Three classes. (The x-axis indicates the number of features.)