Generative Model-based Document Clustering: A Comparative Study

Shi Zhong
Department of Computer Science and Engineering
Florida Atlantic University
777 Glades Road, Boca Raton, FL 33431

Joydeep Ghosh
Department of Electrical and Computer Engineering
The University of Texas at Austin
1 University Station, Austin, TX 78712

Abstract

This paper presents a detailed empirical study of twelve generative approaches to text clustering obtained by applying four types of document-to-model assignment strategies (hard, stochastic, soft and deterministic annealing (DA) based assignments) to each of three base models, namely mixtures of multivariate Bernoulli, multinomials, and von Mises-Fisher (vMF) distributions. A large variety of text collections, both with and without feature selection, are used for the study, which yields several insights, including (a) showing situations wherein the vMF centric approaches, which are based on directional statistics, fare better than multinomial model-based methods, and (b) quantifying the trade-off between increased performance of the soft and DA assignments and their increased computational demands. We also compare all the model-based algorithms with two state-of-the-art discriminative approaches to document clustering based respectively on graph partitioning (CLUTO) and a spectral co-clustering method. Overall, DA and CLUTO perform the best but are also the most computationally expensive. The vMF models provide good performance at low cost while the spectral co-clustering algorithm fares worse than vMF-based methods for a majority of the datasets.

Keywords: Document Clustering, Model-based Clustering, Comparative Study

1 Introduction

Document clustering has become an increasingly important technique for unsupervised document organization, automatic topic extraction, and fast information retrieval or filtering. For example, a web search engine often returns thousands of pages in response to a broad query, making it difficult for users to browse or to identify relevant information. Clustering methods can be used to automatically group the retrieved documents into a list of meaningful categories, as is achieved by search engines such as Northern Light (http://www.northernlight.com) and Vivisimo (http://www.vivisimo.com), or an automated news aggregator/organizer such as Google News (http://news.google.com). Similarly, a large database of documents can be pre-clustered to facilitate query processing by searching only the cluster that is closest to the query.

If the popular vector space representation is used, a text document, after suitable pre-processing, gets mapped to a high dimensional vector with one dimension per "term" (Salton and McGill, 1983).

Such vectors tend to be very sparse, and they have only non-negative entries. Moreover, it has been widely observed that the vectors have directional properties, i.e., the length of the vector is much less important than their direction. This has led to the widespread practice of normalizing the vectors to unit length before further analysis, as well as to the use of the cosine between two vectors as a popular measure of similarity between them.

Till the mid-nineties, hierarchical agglomerative clustering using a suitable similarity measure such as cosine, Dice or Jaccard, formed the dominant paradigm for clustering documents (Rasmussen, 1992; Cutting et al., 1992). The increasing interest in processing larger collections of documents has led to a new emphasis on designing more efficient and effective techniques, leading to an explosion of diverse approaches to the document clustering problem, including the (multi-level) self-organizing map (Kohonen et al., 2000), mixture of Gaussians (Tantrum et al., 2002), spherical k-means (Dhillon and Modha, 2001), bi-secting k-means (Steinbach et al., 2000), mixture of multinomials (Vaithyanathan and Dom, 2000; Meila and Heckerman, 2001), divisive information-theoretic KL clustering (Dhillon and Guan, 2003), multi-level graph partitioning (Karypis, 2002), mixture of vMFs (Banerjee et al., 2003), information bottleneck (IB) clustering (Slonim and Tishby, 2000) and co-clustering using bipartite spectral graph partitioning (Dhillon, 2001). This richness of approaches prompts a need for detailed comparative studies to establish the relative strengths or weaknesses of these methods.

Most clustering methods proposed for data mining (Berkhin, 2002; Ghosh, 2003) can be divided into two categories: *discriminative* (or similarity-based) approaches (Indyk, 1999; Vapnik, 1998) and *generative* (or model-based) approaches (Blimes, 1998; Cadez et al., 2000). In similarity-based approaches, one optimizes an objective function involving the pairwise document similarities, aiming to maximize the average similarities within clusters and minimize the average similarities between clusters. Model-based approaches, on the other hand, attempt to learn generative models from the documents, with each model representing one particular document group. The empirical study in this paper focuses on model-based approaches since they provide several advantages. First, model-based partitional clustering algorithms have a complexity of $O(KNM)$, where K is the number of clusters, N the number of data objects, and M the number of iterations. In similarity-based approaches, just calculating the pairwise similarities requires $O(N^2)$ time. Second, each cluster is described by a representative model, which provides a richer interpretation of the cluster. Third, online algorithms can be easily constructed for model-based clustering using competitive learning techniques, e.g., see Banerjee and Ghosh (2004). Online algorithms are useful for clustering a stream of documents such as news feeds, as well as for incremental learning situations.

We recently introduced a unified framework for probabilistic model-based clustering (Zhong and Ghosh, 2003b), which allows one to understand and compare a vast range of model-based partitional clustering methods using a common viewpoint that centers around two steps—a model re-estimation step and a data re-assignment step. This two-step view enables one to easily combine different models with different assignment strategies. We now apply this unified framework to design a set of comparative experiments, involving three probabilistic Models suitable for clustering documents: multivariate Bernoulli, multinomial, and von Mises-Fisher, in conjunction with four types of data assignments, thus leading to a total of twelve algorithms. Note that all the three models directly handle high dimensional vectors without dimensionality reduction, and have been recommended for dealing with the peculiar characteristics of document clustering. In contrast, Gaussian-based algorithms such as k-means perform very poorly for such datasets (Strehl et al., 2000). All twelve instantiated algorithms are compared on a number of document datasets derived from the TREC collections and internet newsgroups, both with and without feature selection. Our goal is to empirically investigate the suitability of each model for document clustering and identify which model works better in what situations. We also compare all the model-based algorithms

with two state-of-the-art graph-based approaches, the *vcluster* algorithm in the CLUTO toolkit (Karypis, 2002) algorithm and a bipartite spectral co-clustering method (Dhillon, 2001). The comparison to recent KL clustering or IB clustering is not needed, given the equivalence between Information Bottleneck text clustering and multinomial model-based clustering demonstrated in Section 3.

McCallum and Nigam (1998) performed a comparative study of Bernoulli and multinomial models for text classification but not for clustering. Comparisons of different document clustering methods have been done by Steinbach et al. (2000), and by Zhao and Karypis (2001). They both focused on comparing partitional with hierarchical approaches either for one model, or for similarity-based clustering algorithms (in the CLUTO toolkit). Meila and Heckerman (2001) compared hard vs. soft assignment strategies for text clustering using multinomial models. To the best of our knowledge, however, a comprehensive comparison of different probabilistic models for clustering documents has not been done before except in our previous work (Zhong and Ghosh, 2003a), which is now substantially expanded in this paper.

Section 2 reviews the four data assignment strategies and Section 3 describes the three probabilistic models for clustering text documents. Section 4 compares the clustering performance of different models and data assignment strategies on a number of text datasets. Finally, section 5 concludes this paper.

2 Model-based partitional clustering

In this section, we briefly review the four data assignment strategies that are at the core of four related clustering algorithms—model-based k-means (mk-means), "EM clustering",[1] stochastic mk-means, and deterministic annealing, respectively. A more detailed exposition of the ideas in this section can be found in Zhong and Ghosh (2003b), where virtually all existing model based clustering approaches, both partitional and hierarchical, are captured within a unified framework.

Model-based k-means

The model-based k-means (*mk-means*) algorithm is a generalization of the standard k-means algorithm, with the cluster centroid vectors being replaced by probabilistic models. Let $X = \{x_1, ..., x_N\}$ be the set of data objects and $\Lambda = \{\lambda_1, ..., \lambda_K\}$ the set of cluster models. The mk-means algorithm locally maximizes the log-likelihood objective function

$$\log P(X|\Lambda) = \sum_{x \in X} \log p(x|\lambda_{y(x)}) \ , \tag{1}$$

where $y(x) = \arg\max_y \log p(x|\lambda_y)$ is the cluster identity of object x.

When equi-variance spherical Gaussian models are used in a vector space, mk-means reduces to the standard k-means algorithm (MacQueen, 1967). As another example, the spherical k-means algorithm developed specifically for text (Dhillon and Modha, 2001; Banerjee and Ghosh, 2002) uses the von Mises-Fisher distribution as its underlying probabilistic model.

Clustering via Mixture Modeling

The generic EM clustering algorithm (Banfield and Raftery, 1993; Cadez et al., 2000) is a generalization of the mixture-of-Gaussians clustering (Blimes, 1998) that uses a mixture of probabilistic

[1] This term signifies a specific application of the more general EM algorithm (Dempster et al., 1977), where one treats the cluster identities of data objects as the hidden indicator variables and then tries to maximize the objective function in Equation 2 using the EM algorithm.

models for which a maximum likelihood estimation is possible (e.g., probabilistic models in the exponential family), to model the data. Given a set of K probabilistic models Λ, EM is applied to find a local maximum of the data log-likelihood

$$\log p(X|\Lambda) = \sum_x \log \left(\sum_{y=1}^{K} \alpha_y p(x|\lambda_y) \right) , \tag{2}$$

where the parameters α's are cluster priors. The algorithm amounts to iterating between the following E-step and M-step until convergence:
E-step:

$$P(y|x,\Lambda) = \frac{\alpha_y p(x|\lambda_y)}{\sum_{y'} \alpha_{y'} p(x|\lambda_{y'})} ; \tag{3}$$

M-step:

$$\lambda_y^{(new)} = \arg\max_{\lambda} \sum_x P(y|x,\Lambda) \log p(x|\lambda), \tag{4}$$

$$\alpha_y^{(new)} = \frac{1}{N} \sum_x P(y|x,\Lambda) . \tag{5}$$

A partition of the data objects is actually a byproduct of the maximum likelihood estimation process.

Stochastic model-based k-means

The *stochastic mk-means* is a stochastic variant of the mk-means. It *stochastically* assigns each data object entirely to one cluster (and not fractionally, as in soft clustering), with the probability of object x going to cluster y set to be the posterior probability $P(y|x,\Lambda)$. Kearns et al. (1997) described this algorithm as *posterior assignment*. The stochastic mk-means can be viewed as a sampled version of EM clustering, where one uses a sampled E-step based on the posterior probability.

Model-based deterministic annealing

Model-based deterministic annealing (Zhong and Ghosh, 2003b) extends EM clustering by parameterizing the E-step in (3) with a temperature parameter T, which gradually decreases during the clustering process. Let Y be the set of cluster indices, and the joint probability between X and Y be $P(x,y)$. Model-based DA clustering aims to maximize the expected log-likelihood with entropy constraints

$$\begin{aligned} L &= E_{P(x,y)}[\log p(x|\lambda_y)] + T \cdot H(Y|X) - T \cdot H(Y) \\ &= \sum_x P(x) \sum_y P(y|x) \log p(x|\lambda_y) - T \cdot I(X;Y) . \end{aligned} \tag{6}$$

For each T, the E-step can be shown to become

$$P(y|x,\Lambda) = \frac{\alpha_y p(x|\lambda_y)^{\frac{1}{T}}}{\sum_{y'} \alpha_{y'} p(x|\lambda_{y'})^{\frac{1}{T}}} . \tag{7}$$

The M-step is the same as (4) in the EM clustering algorithm.

Discussion

Model-based k-means and EM clustering can be viewed as two special stages of a model-based deterministic annealing process, with $T = 0$ and $T = 1$, respectively, and they optimize two different objective functions.

In practice, we often have the condition $P(x|\lambda_{y(x)}) \gg P(x|\lambda_y), \forall y \neq y(x)$ (this is often true for the models discussed in the next section), which means that $P(y|x, \Lambda)$ will be dominated by the likelihood values and be very close to 1 for $y = y(x)$, and 0 otherwise, independent of most choices of T's and α's. This suggests that the difference between hard and soft versions is small, i.e. their clustering results will be fairly similar. This is also confirmed by the experimental results presented in this paper.

The complexities of the above model-based clustering algorithms are linear in K, number of clusters, N, number of data objects, and M, number of iterations. In our experiments, we typically used $M_{max} = 20$, which is large enough for most of our experimental runs to converge.

3 Probabilistic models for text documents

The traditional vector space representation is used for text documents, i.e., each document is represented as a high dimensional vector of "word" counts in the document. The "word" here is used in a broad sense since it may represent individual words, stemmed words, tokenized words, or short phrases. The dimensionality of document vectors equals the vocabulary size. Depending on whether the vectors are binarized or not, the popular generative models for such a representation are multivariate Bernoulli and multinomial mixtures. Recently, a third model, inspired by the directional nature of text data, was proposed that uses a mixture of vMF distributions. Thus these three models, which are briefly discussed below, are the focus of our study.

3.1 Multivariate Bernoulli model

In a multivariate Bernoulli model (McCallum and Nigam, 1998), a document is represented as a binary vector over the space of words. The l-th dimension of a document vector x is denoted by $x(l)$, and is either 1 or 0, indicating whether word w_l occurs or not in the document. Thus the number of occurrences is not considered, i.e., the word frequency information is lost.

With naïve Bayes assumption, the probability of a document x in cluster y is

$$P(x|\lambda_y) = \prod_l P_y(w_l)^{x(l)}(1 - P_y(w_l))^{1-x(l)} \ , \tag{8}$$

where $\lambda_y = \{P_y(w_l)\}$, $P_y(w_l)$ is the probability of word w_l being present in cluster y, and $(1-P_y(w_l))$ the probability of word w_l not being present in cluster y. To avoid zero probabilities when estimating $P_y(w_l)$, one can employ a Laplacian prior (i.e., $P(\lambda_y) = C \cdot P_y(w_l)(1-P_y(w_l))$, C is a normalization constant) and derives the solution as (McCallum and Nigam, 1998)

$$P_y(w_l) = \frac{1 + \sum_x P(y|x, \Lambda)x(l)}{2 + \sum_x P(y|x, \Lambda)} \ , \tag{9}$$

where $P(y|x, \Lambda)$ is the posterior probability of cluster y.

3.2 Multinomial model

Standard description of multinomial models is available in many statistics or probability books (e.g., Stark and Woods, 1994); here we briefly discuss it in the context of clustering text documents.

Based on the naïve Bayes assumption, a multinomial model for cluster y represents a document x by a multinomial distribution of the words in the vocabulary

$$P(x|\lambda_y) = \prod_l P_y(l)^{x(l)} \ ,$$

where $x(l)$ is the l-th dimension of document vector x, indicating the number of occurrences of the l-th word in document x. To accommodate documents of different lengths, we use a normalized (log-)likelihood measure

$$\log \tilde{P}(x|\lambda_y) = \frac{1}{|x|} \log P(x|\lambda_y) \ , \tag{10}$$

where $|x| = \sum_l x(l)$ is the length of document x. The $P_y(l)$'s are the multinomial model parameters and represent the word distribution in cluster y. They are subject to the constraint $\sum_l P_y(l) = 1$ and can be estimated by counting the number of documents in each cluster and the number of word occurrences in all documents in the cluster y (Nigam, 2001). With Laplacian smoothing, i.e., with model prior $P(\lambda_y) = C \cdot \prod_l P_y(l)$, the parameter estimation of multinomial models amounts to

$$P_y(l) = \frac{1 + \sum_x P(y|x,\Lambda)x(l)}{\sum_i (1 + \sum_x P(y|x,\Lambda)x(i))} = \frac{1 + \sum_x P(y|x,\Lambda)x(l)}{|V| + \sum_i \sum_x P(y|x,\Lambda)x(i)} \ , \tag{11}$$

where $|V|$ is the size of the word vocabulary, i.e., the dimensionality of document vectors. The posterior $P(y|x,\Lambda)$ can be estimated from (7).

Connection to KL Clustering

A connection between multinomial model-based clustering and the divisive Kullback-Leibler clustering (Dhillon et al., 2002b; Dhillon and Guan, 2003) is worth mentioning here. It is briefly mentioned in Dhillon and Guan (2003) but they did not explicitly reveal the equivalence between divisive KL clustering and multinomial model-based k-means. Let $P_x(l) = \frac{x(l)}{|x|}$ and $y(x)$ be the cluster identity of document x. The objective function (to be minimized) for divisive KL clustering is the sum of KL divergence between a document (represented by word distribution P_x) and its cluster distribution $P_{y(x)}$

$$\begin{aligned} \sum_x D_{KL}(P_x|P_{y(x)}) &= \sum_x \sum_l P_x(l) \log \frac{P_x(l)}{P_{y(x)}(l)} \\ &= -\sum_x \left(H(P_x) + \sum_l \frac{x(l)}{|x|} \log P_{y(x)}(l) \right) \\ &= -\sum_x \left(H(P_x) + \frac{1}{|x|} \log P(x|\lambda_{y(x)}) \right) . \end{aligned} \tag{12}$$

Since $\sum_x H(P_x) = -\sum_{x,l} P_x(l) \log P_x(l)$ is a constant w.r.t. λ and y, minimizing the above objective is equivalent to maximizing the objective for multinomial model-based k-means

$$\frac{1}{N} \sum_x \frac{1}{|x|} \log P(x|\lambda_{y(x)}) = \frac{1}{N} \sum_x \log \tilde{P}(x|\lambda_{y(x)}) \ . \tag{13}$$

This also indicates that multinomial model-based DA clustering algorithms described below can be viewed as a deterministic annealing extension of soft divisive KL clustering.

Multinomial Model-based DA Clustering and the Information Bottleneck Method

Substituting the generic M-step (4) in the model-based DA clustering with (11) gives a multinomial model-based DA clustering algorithm, abbreviated as *damnl*. The normalized log-likelihood measure (10) is used since it accommodates different document lengths and leads to a stable annealing process in our experiments.

Based on (6) and the above analysis on relationship between multinomial model-based clustering and KL clustering, it is easy to see that the objective function of *damnl* can be written as

$$L = -\sum_{x,y} P(x,y) D_{KL}(P_x|P_y) - T \cdot I(X;Y) + \sum_x H(P_x) \ , \tag{14}$$

where the last term is a constant. With this representation, one can show that, when applied to clustering, the Information Bottleneck method is just a special case of model-based DA clustering with the underlying probabilistic models being multinomial models. This has also been mentioned by Slonim and Weiss (2003) when they explored the relationship between maximum likelihood formulation and information bottleneck. A more formal treatment, which shows both IB and *damnl* as special cases of an even broader framework, and precisely states the assumptions behind the IB technique, can be found in Banerjee et al. (2004).

The IB method aims to minimize the objective function

$$\begin{aligned} F &= I(X;Y) - \beta I(Z;Y) \\ &= I(X;Y) + \beta(I(Z;X) - I(Z;Y)) - \beta I(Z;X) \\ &= I(X;Y) + \beta E_{p(x,y)}[D_{KL}(p(z|x)|p(z|y))] - \beta I(Z;X) \end{aligned} \tag{15}$$

and represents the tradeoff between minimizing the mutual information between data X and compressed clusters Y and preserving the mutual information between Y and a third variable Z. Both X and Z are fixed data but Y represents the cluster structure that one tries to find out. The last term in (15) can be treated as a constant w.r.t. to the Y and thus to the clustering algorithm. One can easily see that minimizing (15) is equivalent to maximizing (14), with β being the inverse of temperature T and Z being a random variable representing the word dimension.

3.3 von Mises-Fisher model

The von Mises-Fisher distribution is the analogue of the Gaussian distribution for directional data in the sense that it is the unique distribution of L_2-normalized data that maximizes the entropy given the first and second moments of the distribution (Mardia, 1975). The vMF distribution for cluster y can be written as

$$P(x|\lambda_y) = \frac{1}{Z(\kappa_y)} \exp\left(\kappa_y \frac{x^T \mu_y}{\|\mu_y\|}\right) \ , \tag{16}$$

where x is a normalized (unit-length in L_2 norm) document vector and the Bessel function $Z(\kappa_y)$ is a normalization term. The parameter κ measures the directional variance (or dispersion) and the higher its value, the more peaked the distribution is. For the vMF-based k-means algorithm, we assume κ is the same for all clusters, i.e., $\kappa_y = \kappa, \forall y$. This results in the spherical k-means (Dhillon and Modha, 2001; Dhillon et al., 2001), which uses cosine similarity to measure the closeness of a data point to its cluster's centroid and has shown good results for text clustering. The model estimation in this case simply amounts to $\mu_y = \frac{1}{N_y} \sum_{x \in C_y} x$, where N_y is the number of documents in cluster C_y. The estimation for κ in the mixture-of-vMFs clustering algorithm, however, is rather difficult due to the Bessel function involved.

In Banerjee et al. (2003), the EM based maximum likelihood solution has been derived, including updates for κ. While it provides markedly better results than those obtained with a fixed κ, it is computationally much more expensive even if an approximation for estimating κ's is used. In this paper, for convenience, we use a simpler soft assignment scheme that is similar to deterministic annealing. We use a κ that is constant across all models at each iteration, start with a low value of κ, and gradually increase the κ (i.e. make the distributions more peaked) in unison with each iteration. Note that κ has the effect of an "inverse temperature" parameter.

4 Experimental results

4.1 Evaluation criteria

Objective clustering evaluation criteria can be based on internal measures or external measures. An internal measure is often the same as the objective function that a clustering algorithm explicitly optimizes, as is the sum-squared error criteria used for the standard k-means. For document clustering, external measures are more commonly used, since typically the benchmark documents' category labels are actually known (but of course not used in the clustering process). Examples of external measures include the confusion matrix, classification accuracy, F1 measure, average purity, average entropy, and mutual information (Ghosh, 2003).

In the simplest scenario where the number of clusters equals the number of categories and their one-to-one correspondence can be established, any of these external measures can be fruitfully applied. However, when the number of clusters differs from the number of original classes, the confusion matrix is hard to read and the accuracy difficult or impossible to calculate. It has been argued that the mutual information $I(Y;\hat{Y})$ between a *r.v.* Y, governing the cluster labels, and a *r.v.* $\hat{Y}$, governing the class labels, is a superior measure than purity or entropy (Strehl and Ghosh, 2002; Dom, 2001). Moreover, by normalizing this measure to lie in the range [0,1], it becomes relatively impartial to K. There are several choices for normalization based on the entropies $H(Y)$ and $H(\hat{Y})$. We shall follow the definition of normalized mutual information (NMI) using geometrical mean, $NMI = \frac{I(Y;\hat{Y})}{\sqrt{H(Y)\cdot H(\hat{Y})}}$, as given in (Strehl and Ghosh, 2002). In practice, we use a sample estimate

$$NMI = \frac{\sum_{h,l} n_{h,l} \log\left(\frac{n \cdot n_{h,l}}{n_h n_l}\right)}{\sqrt{\left(\sum_h n_h \log\frac{n_h}{n}\right)\left(\sum_l n_l \log\frac{n_l}{n}\right)}}\ , \tag{17}$$

where n_h is the number of documents in class h, n_l the number of documents in cluster l and $n_{h,l}$ the number of documents in class h as well as in cluster l. The NMI value is 1 when clustering results perfectly match the external category labels and close to 0 for a random partitioning. This is a better measure than purity or entropy which are both biased towards high K solutions (Strehl et al., 2000; Strehl and Ghosh, 2002).

In our experiments, we use NMI as the evaluation criterion. Since the three probabilistic models use slightly different representations of documents, we cannot directly compare their objective functions (data likelihoods) under different probabilistic models.

Table 1: Summary of text datasets (for each dataset, n_d is the total number of documents, n_w the total number of words, K the number of classes, and $\bar{n}_c$ the average number of documents per class)

Data	Source	n_d	n_w	K	$\bar{n}_c$	Balance
NG20	20 Newsgroups	19949	43586	20	997	0.991
NG17-19	3 overlapping subgroups from NG20	2998	15810	3	999	0.998
classic	CACM/CISI/CRANFIELD/MEDLINE	7094	41681	4	1774	0.323
ohscal	OHSUMED-233445	11162	11465	10	1116	0.437
k1b	WebACE	2340	21839	6	390	0.043
hitech	San Jose Mercury (TREC)	2301	10080	6	384	0.192
reviews	San Jose Mercury (TREC)	4069	18483	5	814	0.098
sports	San Jose Mercury (TREC)	8580	14870	7	1226	0.036
la1	LA Times (TREC)	3204	31472	6	534	0.290
la12	LA Times (TREC)	6279	31472	6	1047	0.282
la2	LA Times (TREC)	3075	31472	6	513	0.274
tr11	TREC	414	6429	9	46	0.046
tr23	TREC	204	5832	6	34	0.066
tr41	TREC	878	7454	10	88	0.037
tr45	TREC	690	8261	10	69	0.088

4.2 Text datasets

We used the 20-newsgroups data[2] and a number of datasets from the CLUTO toolkit[3] (Karypis, 2002). These datasets provide a good representation of different characteristics: number of documents ranges from 204 to 19949, number of words from 5832 to 43586, number of classes from 3 to 20, and balance from 0.036 to 0.998. The balance of a dataset is defined as the ratio of the number of documents in the smallest class to the number of documents in the largest class. So a value close to 1(0) indicates a very (un)balanced dataset. A summary of all the datasets used in this paper is shown in Table 1.

The *NG20* dataset is a collection of 20,000 messages, collected from 20 different usenet newsgroups, 1,000 messages from each. We preprocessed the raw dataset using the Bow toolkit (McCallum, 1996), including chopping off headers and removing stop words as well as words that occur in less than three documents. In the resulting dataset, each document is represented by a 43,586-dimensional sparse vector and there are a total of 19,949 documents (after empty documents being removed). The *NG17-19* dataset is a subset of NG20, containing ~ 1000 messages from each of the three categories on different aspects of politics. These three categories are expected to be difficult to separate. After the same preprocessing step, the resulting dataset consists of 2,998 documents in a 15,810 dimensional vector space.

All the datasets associated with the CLUTO toolkit have already been preprocessed (Zhao and Karypis, 2001) and we further removed those words that appear in two or fewer documents. The *classic* dataset was obtained by combining the CACM, CISI, CRANFIELD, and MEDLINE abstracts that were used in the past to evaluate various information retrieval systems[4]. The *ohscal* dataset was from the OHSUMED collection (Hersh et al., 1994). It contains 11,162 documents from the following ten categories: antibodies, carcinoma, DNA, in-vitro, molecular sequence data, pregnancy, prognosis, receptors, risk factors, and tomography. The *k1b* dataset is from the WebACE project (Han et al., 1998). Each document corresponds to a web page listed in the subject

[2] http://kdd.ics.uci.edu/databases/20newsgroups/20newsgroups.html .

[3] http://www.cs.umn.edu/∼karypis/CLUTO/files/datasets.tar.gz .

[4] Available from ftp://ftp.cs.cornell.edu/pub/smart.

hierarchy of Yahoo! (http://www.yahoo.com). The other datasets are from TREC collections (http://trec.nist.gov). In particular, the *hitech*, *reviews*, and *sports* were derived from the San Jose Mercury newspaper articles. The *hitech* dataset contains documents about computers, electronics, health, medical, research, and technology; the *reviews* dataset contains documents about food, movies, music, radio, and restaurants; the *sports* dataset contains articles about baseball, basketball, bicycling, boxing, football, golfing, and hockey. The *la1*, *la12*, and *la2* datasets were obtained from articles of the Los Angeles Times in the following six categories: entertainment, financial, foreign, metro, national, and sports. Datasets *tr11*, *tr23*, *tr41*, and *tr45* are derived from TREC-5, TREC-6, and TREC-7 collections.

4.3 Experimental setting

The four algorithms based on the Bernoulli model are k-Bernoullis, stochastic k-Bernoullis, mixture-of-Bernoullis, and Bernoulli-based DA, abbreviated as *kberns*, *skberns*, *mixberns*, and *daberns* respectively. Similarly, the abbreviated names are *kmnls*, *skmnls*, *mixmnls*, and *damnls* for multinomial-based algorithms, and are *kvmfs*, *skvmfs*, *softvmfs*, and *davmfs* for vMF-based algorithms. We use *softvmfs* instead of *mixvmfs* for the soft vMF-based algorithm for the following reason. As mentioned in Section 3, the estimation of parameter κ in a vMF model is difficult but is needed for the mixture-of-vMFs algorithm. As a simple heuristic, we use $\kappa^{(m)} = 20m$, where m is the iteration number. So κ is a constant for all clusters at each iteration, and gradually increasing over iterations.

For the *davmfs* algorithm, the temperature parameter T can be assimilated into κ, which has an interpretation of inverse temperature. We set κ to follow an exponential schedule $\kappa^{(m+1)} = 1.1\kappa^{(m)}$, starting from 1 and up to 500. We call this algorithm *davmfs*. For vMF-based algorithms, we also use log(IDF)-weighted and normalized document vectors.

For the *daberns* and *damnls* algorithms, an inverse temperature parameter $\beta = 1/T$ is used to parameterize the E-step in the *mixberns* and *mixmnls* algorithms. The annealing schedule for *daberns* is set to $\beta(m+1) = 1.2\beta(m)$, and β increases from 0.002 up to 1; for *damnls* it is set to $\beta(m+1) = 1.3\beta(m)$, and β grows from 0.5 up to 200.

For all the model-based algorithms (except for the DA algorithms), we use a maximum number of iterations of 20 (to make a fair comparison). Our results show that most runs converge within 20 iterations if a relative convergence criterion of 0.001 is used. Each experiment is run ten times, each time starting from a different random initialization. The averages and standard deviations of the NMI and running time results are reported.

After surveying a range of spectral or graph-based partitioning techniques, (Meila and Shi, 2001a,b; Kannan et al., 2000), we picked two state-of-the-art graph-based clustering algorithms as leading representatives of this class of similarity-based approaches. in our experiments. The first one is CLUTO (Karypis, 2002), a clustering toolkit based on the Metis graph partitioning algorithms (Karypis and Kumar, 1998). We use *vcluster* in the toolkit with the default setting, which is a bisecting graph partitioning-based algorithm. The other one is a modification of the bipartite spectral co-clustering algorithm (Dhillon, 2001). The modification is according to Ng et al. (2002)[5] and generates slightly better results than the original bipartite clustering algorithm. The *vcluster* algorithm is greedy and thus dependent on the order of nodes from the input graph. The spectral co-clustering algorithm uses the standard k-means algorithm in its last step, which introduces randomness into the co-clustering process. We run each algorithm ten times, each run using a different order of documents.

[5]Use K instead of $\log K$ eigen-directions and normalize each projected data vector.

4.4 Clustering results without feature selection

Table 2 shows the NMI results on the *NG20*, *NG17-19*, *classic*, *ohscal*, and *hitech* datasets. All numbers in the table are shown in the format *average* $\pm$ 1 *standard deviation*. Boldface entries highlight the best algorithms in each column. To save space, we show the NMI results on for one specific K only for each dataset (results for other datasets are shown in Table 3 and Table 4).

Table 2: NMI Results on *NG20*, *NG17-19*, *classic*, *ohscal*, and *hitech* datasets

	NG20	NG17-19	classic	ohscal	hitech
K	20	3	4	10	6
kberns	.20 ± .04	.03 ± .01	.23 ± .10	.37 ± .02	.11 ± .05
skberns	.21 ± .03	.03 ± .01	.23 ± .11	.38 ± .02	.11 ± .03
mixberns	.19 ± .03	.03 ± .01	.20 ± .15	.37 ± .02	.11 ± .04
daberns	.03 ± .00	.03 ± .01	.05 ± .08	.00 ± .00	.01 ± .00
kmnls	.53 ± .03	.23 ± .08	.56 ± .06	.37 ± .02	.23 ± .03
skmnls	.53 ± .03	.22 ± .08	.57 ± .06	.37 ± .02	.23 ± .04
mixmnls	.54 ± .03	.23 ± .08	.66 ± .04	.37 ± .02	.23 ± .03
damnls	.57 ± .02	.36 ± .12	**.71 ± .06**	.39 ± .02	.27 ± .01
kvmfs	.55 ± .02	.37 ± .10	.54 ± .03	.43 ± .03	.28 ± .02
skvmfs	.56 ± .01	.37 ± .08	.54 ± .02	.44 ± .02	.29 ± .02
softvmfs	.57 ± .02	.39 ± .10	.55 ± .03	.44 ± .02	.29 ± .01
davmfs	**.59 ± .02**	**.46 ± .01**	.51 ± .01	**.47 ± .02**	.30 ± .01
CLUTO	.58 ± .01	**.46 ± .01**	.54 ± .02	.44 ± .02	**.33 ± .01**
co-cluster	.46 ± .01	.02 ± .01	.01 ± .01	.39 ± .01	.22 ± .03

Table 3: NMI Results on *reviews*, *sports*, *la1*, *la12*, and *la2* datasets

	reviews	sports	la1	la12	la2
K	5	7	6	6	6
kberns	.30 ± .05	.39 ± .06	.04 ± .04	.06 ± .06	.17 ± .03
skberns	.30 ± .04	.37 ± .05	.06 ± .05	.07 ± .06	.19 ± .03
mixberns	.29 ± .05	.37 ± .05	.05 ± .05	.06 ± .05	.20 ± .04
daberns	.04 ± .01	.02 ± .00	.01 ± .00	.01 ± .00	.01 ± .00
kmnls	.55 ± .08	.59 ± .06	.39 ± .05	.42 ± .04	.47 ± .04
skmnls	.55 ± .08	.58 ± .06	.41 ± .05	.43 ± .04	.47 ± .05
mixmnls	**.56 ± .08**	.59 ± .06	.41 ± .05	.43 ± .05	.48 ± .04
damnls	.51 ± .06	.57 ± .04	.49 ± .02	.54 ± .03	.45 ± .03
kvmfs	.53 ± .06	.57 ± .08	.49 ± .05	.50 ± .03	.54 ± .04
skvmfs	.53 ± .07	.61 ± .04	.51 ± .04	.51 ± .04	.52 ± .03
softvmfs	**.56 ± .06**	.60 ± .05	.52 ± .04	.53 ± .05	.49 ± .04
davmfs	**.56 ± .09**	.62 ± .05	.53 ± .03	.52 ± .02	.52 ± .04
CLUTO	.52 ± .01	**.67 ± .01**	**.58 ± .02**	**.56 ± .01**	**.56 ± .01**
co-cluster	.40 ± .07	.56 ± .02	.41 ± .05	.42 ± .07	.41 ± .02

Table 5 show the results for a series of paired t-tests. In particular, we test the following seven hypotheses: *bb>wb* – the best of *kberns, skberns*, and *mixberns* is better than the worst of them (in terms of NMI performance); *bm>wm* – the best of *kmnls, skmnls*, and *mixmnls* is better than the worst of them; *bv>wv* – the best of *kvmfs, skvmfs*, and *mixvmfs* is better than the worst of them; *dam>bm* – *damnls* is better than the best of *kmnls, skmnls*, and *mixmnls*; *dav>bv* – *davmfs* is better

Table 4: NMI Results on *k1b*, *tr11*, *tr23*, *tr41*, and *tr45* datasets

	k1b	tr11	tr23	tr41	tr45
K	6	9	6	10	10
kberns	$.32 \pm .25$	$.07 \pm .02$	$.11 \pm .01$	$.27 \pm .05$	$.13 \pm .06$
skberns	$.36 \pm .24$	$.08 \pm .02$	$.11 \pm .01$	$.27 \pm .06$	$.13 \pm .05$
mixberns	$.31 \pm .24$	$.07 \pm .02$	$.11 \pm .01$	$.27 \pm .04$	$.13 \pm .06$
daberns	$.04 \pm .00$	$.09 \pm .00$	$.08 \pm .01$	$.02 \pm .00$	$.07 \pm .00$
kmnls	$.55 \pm .04$	$.39 \pm .07$	$.15 \pm .03$	$.49 \pm .03$	$.43 \pm .05$
skmnls	$.55 \pm .05$	$.39 \pm .08$	$.15 \pm .02$	$.50 \pm .04$	$.43 \pm .05$
mixmnls	$.56 \pm .04$	$.39 \pm .07$	$.15 \pm .03$	$.50 \pm .03$	$.43 \pm .05$
damnls	$.61 \pm .04$	$.61 \pm .02$	$.31 \pm .03$	$.61 \pm .05$	$.56 \pm .03$
kvmfs	$.60 \pm .03$	$.52 \pm .03$	$.33 \pm .05$	$.59 \pm .03$	$.65 \pm .03$
skvmfs	$.60 \pm .02$	$.57 \pm .04$	$.34 \pm .05$	$.62 \pm .03$	$.65 \pm .05$
softvmfs	$.60 \pm .04$	$.60 \pm .05$	$.36 \pm .04$	$.62 \pm .05$	$.66 \pm .03$
davmfs	$\mathbf{.67 \pm .04}$	$.66 \pm .04$	$.41 \pm .03$	$\mathbf{.69 \pm .02}$	$\mathbf{.68 \pm .05}$
CLUTO	$.62 \pm .03$	$\mathbf{.68 \pm .02}$	$\mathbf{.43 \pm .02}$	$.67 \pm .01$	$.62 \pm .01$
co-cluster	$.60 \pm .01$	$.53 \pm .03$	$.22 \pm .01$	$.51 \pm .02$	$.50 \pm .03$

than the best of *kvmfs, skvmfs*, and *mixvmfs*; *dav>dam* – *davmfs* is better than *damnls*; *dav>cluto* – *davmfs* is better than CLUTO. The p-values shown in the table ranges from 0 to 1. A value of 0.05 or lower indicates significant evidence for the hypothesis to be true, while a value of 0.95 or higher indicate significant evidence for the reverse of the hypothesis to be true. All significant p-values are highlighted in boldface in the table.

Table 5: Summary of paired t-test results.

	Hypothesis tested						
Dataset	bb>wb	bm>wm	bv>wv	dam>bm	dav>bv	dav>dam	dav>cluto
NG20	0.229	0.076	**0.021**	**0.013**	**0.007**	**0.006**	**0.019**
NG17-19	0.277	0.453	0.364	**0.005**	**0.017**	**0.012**	0.54
classic	0.277	**<0.001**	0.147	**0.027**	**0.999**	**>0.999**	**>0.999**
ohscal	0.324	0.223	0.246	**0.04**	**<0.001**	**<0.001**	**<0.001**
hitech	0.228	0.421	0.255	**0.001**	0.089	**<0.001**	**>0.999**
reviews	0.337	0.449	0.128	0.907	0.493	0.135	0.124
sports	0.188	0.395	0.132	0.784	0.243	**0.011**	**0.995**
la1	0.253	0.178	**0.033**	**0.001**	0.267	**<0.001**	**0.999**
la12	0.098	0.28	**0.005**	**<0.001**	0.72	0.911	**0.999**
la2	0.289	0.259	**0.043**	0.133	0.764	**<0.001**	**0.998**
k1b	0.336	0.278	0.436	**0.007**	**<0.001**	**0.003**	**0.001**
tr11	0.225	0.49	**<0.001**	**<0.001**	**0.002**	**<0.001**	0.915
tr23	0.439	0.44	0.084	**<0.001**	**0.002**	**<0.001**	**0.963**
tr41	0.454	0.328	0.075	**<0.001**	**<0.001**	**<0.001**	**0.023**
tr45	0.403	0.417	0.163	**<0.001**	0.203	**<0.001**	**<0.001**

Of the three types of models, vMF leads to the best performance and multivariate Bernoulli the worst. The Bernoulli-based algorithms significantly underperform the other methods for all the datasets except for *ohscal*. This indicates that noting only whether or not a word occurs in a document, but not the number of occurrences, is a limited representation. The vMF-based algorithms perform better than the multinomial-based ones, especially for most of the smaller datasets, i.e., *NG17-19*, *k1b*, *hitech*, *tr11*, *tr23*, *tr41*, and *tr45*. The paired t-tests show that *davmfs*

significantly outperforms *damnls* on 12 out of 15 datasets while significantly underperforms on only one dataset (*classic*).

The three different data assignment strategies, k-means, EM, and stochastic k-means, produce very comparable clustering results across all datasets. The soft EM assignment is only slightly better than the other two. The *t*-test results also show that, for most datasets, there is no significant difference in NMI performance between soft and hard assignment strategies. Specifically, for none of the 15 datasets, the test *bb*>*wb* is significant; and for one and five out of the 15 datasets, the tests *bm*>*wm* and *bv*>*wv* are significant, respectively. For the vMF models, however, one should note that the exact EM clustering can achieve significant improvement over hard assignment (Banerjee et al., 2003).

For multinomial and vMF models, the deterministic annealing algorithm improves the performance of corresponding soft clustering algorithms, sometimes significantly. For example, the *t*-test results show that: *damnls* significantly outperforms the best of *kmnls, skmnls, and mixmnls* on 12 out of 15 datasets; *davmfs* does so on 7 out 15 datasets. Table 6 shows the performance gains of *damnls* over *mixmnls* and *davmfs* over *softvmfs*, on a sorted list of the datasets according to data sizes. A trend seen is that the DA clustering algorithms gain more on medium to small ($n_d \leq 3,000$) datasets.

The deterministic annealing algorithm seems to degrade the performance of *mixberns*, however, as shown by the NMI results. By further looking into the log-likelihood objective values and actual resulting clusters, we observed that deterministic annealing improves the objective value but puts most documents in one cluster, indicating that maximizing data likelihood with Bernoulli models does not align with generating well-separated clusters.

Table 6: Summary of results. (For each dataset, n_d is the total number of documents, $Gain_{mnls} = \frac{NMI_{damnls} - NMI_{mixmnls}}{NMI_{mixmnls}}$ the performance improvement of *damnls* over *mixmnls*, and $Gain_{vmfs} = \frac{NMI_{davmfs} - NMI_{softvmfs}}{NMI_{softvmfs}}$ the performance improvement of *davmfs* over *softvmfs*.)

Data	n_d	Best three algorithms	$Gain_{mnls}$	$Gain_{vmfs}$
NG20	19949	davmfs, CLUTO, damnls	5.6%	3.5%
ohscal	11162	davmfs, CLUTO, softvmfs	5.4%	6.8%
sports	8580	CLUTO, davmfs, softvmfs	-3.4%	3.3%
classic	7094	damnls, mixmnls, skmnls	7.8%	-7.3%
la12	6279	CLUTO, damnls, softvmfs	25.6%	1.2%
reviews	4069	davmfs, softvmfs, mixmnls	-8.9%	0%
la1	3204	CLUTO, davmfs, softvmfs	19.5%	1.9%
la2	3075	CLUTO, kvmfs, skvmfs	-6.3%	6.1%
NG17-19	2998	davmfs, CLUTO, softvmfs	56.5%	17.9%
k1b	2340	davmfs, CLUTO, damnls	8.9%	11.7%
hitech	2301	CLUTO, davmfs, softvmfs	60.9%	3.3%
tr41	878	davmfs, CLUTO, skvmfs	22.0%	11.3%
tr45	690	davmfs, softvmfs, kvmfs	30.2%	3.0%
tr11	414	CLUTO, davmfs, damnls	56.4%	10.0%
tr23	204	CLUTO, davmfs, softvmfs	106.7%	13.9%

Surprisingly, the bipartite spectral co-clustering algorithm mostly underperforms the vMF-based methods and sometimes gives very poor results (with most documents grouped into one cluster and NMI values close to 0). The other graph-based algorithm, CLUTO (actually the *vcluster* algorithm with default setting), performs much better and is overall one of the best among all the algorithms

we have compared. The t-test results show that CLUTO significantly outperforms *davmfs* on 7 out of the 15 datasets, but also significantly underperforms on five of them.

Note that the standard deviations of the model-based clustering results are much larger than that of the CLUTO results, indicating that the initialization effect of model-based methods is larger. Deterministic annealing improves the local solutions but still sees moderate variation over 10 runs. How to substantially improve the initialization or robustness of model-based clustering remains a challenging problem.

Table 7 shows the running time results on *NG20*, the largest dataset used in our experiments. All the numbers are recorded on a 2.4GHz PC running Windows 2000 with 768MB memory, and reflect only the clustering time, not including the data I/O cost. Clearly, algorithms using soft assignment take longer time than those using hard assignments. Overall, the kvmfs algorithm is the fastest one. Since that the CLUTO software package is written in C but all the other algorithms are in Matlab, we expect that most of the model-based algorithms, if re-written in C, will be faster than CLUTO.

Table 7: Running time Results on *NG20* dataset (in seconds)

	NG20			
K	10	20	30	40
kberns	26.8 ± 10.6	43.0 ± 19.0	81.6 ± 37.6	125.4 ± 43.6
skberns	30.2 ± 9.8	65.9 ± 22.1	92.3 ± 35.2	144.7 ± 51.8
mixberns	28.5 ± 11.4	77.8 ± 25.4	102.0 ± 38.9	164.9 ± 38.9
daberns	125.0 ± 0.1	234.6 ± 3.7	352.2 ± 4.6	491.1 ± 5.1
kmnls	17.5 ± 2.9	36.7 ± 4.9	54.8 ± 7.0	78.5 ± 8.4
skmnls	19.7 ± 3.0	39.1 ± 5.6	68.4 ± 7.0	94.9 ± 9.9
mixmnls	23.8 ± 3.6	47.7 ± 6.8	74.2 ± 10.0	99.5 ± 12.7
damnls	78.6 ± 4.3	172.1 ± 7.4	252.5 ± 8.3	362.5 ± 17.9
kvmfs	11.4 ± 1.3	17.5 ± 0.3	21.7 ± 0.1	25.5 ± 0.1
skvmfs	16.1 ± 0.1	24.4 ± 0.2	29.0 ± 9.2	39.1 ± 0.1
softvmfs	34.5 ± 2.2	76.8 ± 1.8	121.7 ± 0.1	178.8 ± 0.2
davmfs	288.4 ± 10.0	671.4 ± 21.4	1050.7 ± 26.2	1584.0 ± 39.7
CLUTO[a]	18.6 ± 1.8	22.6 ± 1.7	25.1 ± 1.7	27.0 ± 1.7
co-cluster	20.9 ± 0.5	39.9 ± 1.0	62.8 ± 0.7	102.9 ± 0.8

[a]CLUTO is written in C whereas all the other algorithms are in Matlab.

4.5 Clustering results with feature selection

In text information retrieval applications, feature selection techniques are often used to select a subset of words, to achieve more compact representation of text documents and reduced computational complexity for manipulating text data. Feature selection has been researched extensively for classification problems (Guyon and Elisseeff, 2003) where each feature dimension can be evaluated based on its ability to differentiate different target labels. In contrast, for clustering problems, there are relatively small number of feature selection techniques. The traditional principal component analysis has difficulty on document clustering due to very high dimensionality. Some recent proposals are quite complicated (Dash et al., 2002; Law et al., 2003).

Feature selection is not the focus of this paper; rather, we intend to see how dimensionality reduction for text documents will affect the model-based clustering results. Therefore, we employ two simple feature selection methods—word frequency-based selection (Dhillon, 2001) and word variance-based selection (Salton and McGill, 1983).

Table 8: Summary of text datasets after feature selection. For each dataset, n_d is the total number of documents, n_w the total number of words.

Feature Selection	frequency-based		variance-based	
Data	n_d	n_w	n_d	n_w
NG20	19949	11941	19949	19949
NG17-19	2998	15791	2998	2998
classic	7085	3244	7086	7094
ohscal	11162	5126	11162	11162
k1b	2340	10337	2340	2340
hitech	2301	9978	2301	2301
reviews	4069	13840	4069	4069
sports	8580	8588	8580	8580
la1	3204	10930	3204	3204
la12	6279	11153	6279	6279
la2	3075	10546	3075	3075
tr11	414	6030	414	414
tr23	204	5277	204	204
tr41	878	7244	878	878
tr45	690	7882	690	690

For frequency-based selection, we simply keep only the words that occur in more than 0.1% and less than 15% of all documents. For the second selection method, we sort all the words based on their variances and keep only the N words with the highest variances. That is, we reduce the number of dimensions to be the same as the number of documents. The variance of the l-th word is defined as

$$\sigma_l^2 = \frac{1}{N}\sum_x x^2(l) - (\frac{1}{N}\sum_x x(l))^2 \ ,$$

where $x(l)$ is the number of occurrences of word w_l in document x. Table 8 shows the dimensionality of each dataset and the number of documents (with empty ones removed) after feature selection step. The clustering results as well as paired t-test results with feature-selected text datasets are presented in Table 9–17 in the Appendix. The hypothesis *fs>nfs* tests whether feature selection improves the clustering results.

The main notable changes in clustering results on feature-selected datasets are:

1. The Bernoulli model-based algorithms—*kberns*, *skberns*, and *mixberns*—perform exceptionally well on feature-selected *classic* datasets, which are relatively simple to cluster. By further examining the *classic* dataset, we see that, of the four classes (MEDLINE, CRANFIELD, CACM, and CISI), CACM class contains documents that are of much shorter length (with an average length of 4.7 vs. 60–80 for the other three classes), and that overlap with the CISI class is in terms of content. Without the CACM class, the rest three classes can be easily identified by many clustering algorithms (Dhillon, 2001; Dhillon et al., 2002a). With CACM, the main difficulty is to separate CACM from CISI. So one reason for the exceptionally good performance of *kberns, skberns*, and *mixberns* on *classic* could be that, after feature selection, i.e., with over 30,000 "irrelevant" features removed, the Bernoulli models can separate CACM from CISI by effectively capturing the length difference. In general, however, the Bernoulli model-based methods are still much inferior to multinomial and vMF model-based algorithms.

2. Just by looking at the NMI numbers, one can see that multinomial model-based methods generally produce better results for feature-selected datasets. For example, as shown in

Table 13 and 18, the average NMI values of *damnls* algorithm significantly improves on 12 out of the 15 frequency-selected datasets and on 14 out of the 15 variance-selected datasets. On the other hand, vMF model-based methods generate lower NMI values for most of the feature-selected datasets. These surprises encourage us to go back and examine the properties of different models to find an explanation. Reconsidering the objective functions for *kmnls* and *kvmfs*, we note that the former maximizes

$$\sum_x \sum_l x(l) \log P_{y(x)}(l) \ ,$$

where $y(x)$ is the cluster index for document x, l is the word index, and $P_{y(x)}$ is the word distribution for cluster $y(x)$. The latter minimizes

$$\sum_x \sum_l \tilde{x}(l) \mu_{y(x)}(l) \ ,$$

where $\tilde{x} = \frac{x}{\|x\|}$ is a normalized document vector and $\mu_{y(x)}$ is the normalized mean of cluster $y(x)$. All quantities are of course empirically estimated based on the training data. Note that *kmnls* involves a $\log(\cdot)$ function, which magnifies the magnitude of $P_{y(x)}(l)$ when the probabilities are small. That is, when the dimensionality of document vectors is high, the discrete word distribution will be diluted, most $P_{y(x)}(l)$'s will be small, and $\log P_{y(x)}(l)$'s will be large negative numbers that may dominate the objective function. If this is the case, the cluster assignment of x based on $\sum_l x(l) \log P_y(l)$ will not be accurate. But if dimensionality decreases (e.g., after feature selection), the discriminative power of x will likely increase in the objective function (relative to $\log P_{y(x)}$), thus improve the partitioning of documents into clusters. Though feature selection may remove words that contain useful discriminating information, our results (especially the *damnls* results in Table 13 and 18) suggest that the benefits from dimensionality reduction outweigh the possible information loss from reduced features for the multinomial model. On the other hand, there is no corresponding benefit for the vMF model and thus feature selection starts hurting the clustering performance earlier on.

3. The relative performance between *damnls* and *davmfs* changes to the opposite—the former is now significantly better than the latter on many datasets. For frequency-selected datasets, *damnls* significantly outperforms *davmfs* on six datasets and underperforms on only three datasets (see Table 12). For variance-selected datasets, *damnls* is significantly better than *davmfs* on 13 datasets and worse on only two (see Table 17).

4. After feature selection, CLUTO seems to deliver lower NMI performance whereas co-cluster seems to fare better on most datasets. For example, as shown in Table 18, the performance of CLUTO significantly improves on three but degrades on six out of 15 variance-selected datasets. In contrast, the co-cluster generates significantly better results on nine and worse on only two out of 15 variance-selected datasets. Overall, *damnls*, *davmfs*, and CLUTO are still the three best algorithms for the feature-selected datasets, and no one is dominating the other two.

5 Concluding remarks

The comparative study of generative models for document clustering provided several insights and some surprises. First, though both EM-based and stochastic assignments seem more sophisticated than hard assignment, in practice they provide little performance improvement over the

corresponding simple and fast "winner-take-all" method. However, deterministic annealing does often significantly improve the performance of both multinomial and vMF model-based clustering algorithms, and are worthwhile if the added computational demands can be tolerated. Second, the Bernoulli model is clearly not appropriate for text clustering unless the clusters are very well separated. More of a surprise, however, was the relatively poor performance of a theoretically well-motivated spectral clustering approach.

By incorporating directional constraints, the von Mises-Fisher model typically provides better results than the popular multinomial model when the number of training samples is small compared to the input dimensionality. By carefully selecting only a subset of the words, this advantage can be neutralized however, and the feature reduction studies show the multinomial performing quite well. Note however that the *softvmfs* used in this paper is not a full-fledged EM algorithm. Concurrent work at UT-Austin on an EM algorithm that allows different dispersion (κ) values for different clusters and lets EM re-estimate these values after each iteration, indicates that substantial gains can be achieved with this added capability (Banerjee et al., 2003). It has been observed that, if small initial κ's are used, the EM procedure gradually increases the κ values, and the final values obtained are typically very high. Since different clusters can have different levels of dispersion during the iterative process the effect is similar to localized deterministic annealing.

All the model-based algorithms (except DA) have a computational advantage over graph-partitioning based approaches but need better initialization strategies to generate more stable clustering results. Meila and Heckerman (2001) compared several initialization techniques and found none to be clearly better, so the quest for more effective techniques continues. Bradley and Fayyad (1998) employed sampling and meta-clustering (clustering of multiple solutions on sampled datasets) to refine initial cluster centroids. This technique deserves more investigation in the future. A second direction on improving the local solution of model-based algorithms is to tweak the clustering process. For example, local search has been employed by Dhillon et al. (2002a) to improve the performance of the spherical k-means algorithm. Also online updates have been reported to work better than batch updates for both spherical k-means (Dhillon et al., 2001) and soft vMF-based clustering (Banerjee and Ghosh, 2002), so online extensions of the other model-based approaches need to be investigated.

Acknowledgments

We thank Inderjit Dhillon, Yuqiang Guan, and Arindam Banerjee for helpful discussions on the results. This research was supported in part by an IBM Faculty Partnership Award from IBM/Tivoli and IBM ACAS, and by NSF grant IIS-0307792.

References

A. Banerjee, I. Dhillon, J. Ghosh, and S. Merugu. An information theoretic analysis of maximum likelihood mixture estimation for exponential families. In *Proc. 21st Int. Conf. Machine Learning*, Banff, Canada, July 2004. To appear.

A. Banerjee, I. S. Dhillon, J. Ghosh, and S. Sra. Generative model-based clustering of directional data. In *Proc. 9th ACM SIGKDD Int. Conf. Knowledge Discovery and Data Mining*, pages 19–28, Washington D.C., August 2003.

A. Banerjee and J. Ghosh. Frequency sensitive competitive learning for clustering on high-dimensional hyperspheres. In *Proc. IEEE Int. Joint Conf. Neural Networks*, pages 1590–1595, May 2002.

A. Banerjee and J. Ghosh. Frequency sensitive competitive learning for balanced clustering on high-dimensional hyperspheres. *IEEE Trans. Neural Networks*, 15(3):702–719, May 2004.

J. D. Banfield and A. E. Raftery. Model-based Gaussian and non-Gaussian clustering. *Biometrics*, 49(3): 803–821, September 1993.

P. Berkhin. Survey of clustering data mining techniques. Unpublished manuscript, available from Accrue.com, 2002.

J. A. Blimes. A gentle tutorial of the EM algorithm and its application to parameter estimation for Gaussian mixture and hidden Markov models. Technical report, University of California at Berkeley, April 1998.

P. S. Bradley and U. M. Fayyad. Refining initial points for k-means clustering. In *Proc. 15th Int. Conf. Machine Learning*, pages 91–99, 1998.

I. V. Cadez, S. Gaffney, and P. Smyth. A general probabilistic framework for clustering individuals and objects. In *Proc. 6th ACM SIGKDD Int. Conf. Knowledge Discovery and Data Mining*, pages 140–149, 2000.

D. Cutting, D. Karger, J. Pedersen, and J. W. Tukey. Scatter/Gather: A cluster-based approach to browsing large document collections. In *Proc. ACM SIGIR*, pages 318–329, 1992.

M. Dash, K. Choi, P. Scheuermann, and H. Liu. Featre selection for clustering — a filter solution. In *Proc. IEEE Int. Conf. Data Mining*, pages 115–122, Maebashi City, Japan, 2002.

A. P. Dempster, N. M. Laird, and D. B. Rubin. Maximum-likelihood from incomplete data via the EM algorithm. *Journal of the Royal Statistical Society B*, 39(1):1–38, 1977.

I. Dhillon and Y. Guan. Information theoretic clustering of sparse co-occurrence data. In *Proc. IEEE Int. Conf. Data Mining*, pages 517–520, Melbourne, FL, November 2003.

I. S. Dhillon. Co-clustering documents and words using bipartite spectral graph partitioning. In *Proc. 7th ACM SIGKDD Int. Conf. Knowledge Discovery and Data Mining*, pages 269–274, 2001.

I. S. Dhillon, J. Fan, and Y. Guan. Efficient clustering of very large document collections. In R. L. Grossman, C. Kamath, P. Kegelmeyer, V. Kumar, and R. R. Namburu, editors, *Data Mining for Scientific and Engineering Applications*, pages 357–381. Kluwer Academic publishers, 2001.

I. S. Dhillon, Y. Guan, and J. Kogan. Iterative clustering of high dimensional text data augmented by local search. In *Proc. IEEE Int. Conf. Data Mining*, pages 131–138, Maebashi City, Japan, 2002a.

I. S. Dhillon, S. Mallela, and R. Kumar. Enhanced word clustering for hierarchical text classification. In *Proc. 8th ACM SIGKDD Int. Conf. Knowledge Discovery and Data Mining*, pages 446–455, July 2002b.

I. S. Dhillon and D. S. Modha. Concept decompositions for large sparse text data using clustering. *Machine Learning*, 42(1):143–175, 2001.

B. Dom. An information-theoretic external cluster-validity measure. Technical Report RJ10219, IBM, 2001.

J. Ghosh. Scalable clustering. In N. Ye, editor, *Handbook of Data Mining*, pages 341–364. Lawrence Erlbaum Assoc., 2003.

I. Guyon and A. Elisseeff. An introduction to variable and feature selection. *Journal of Machine Learning Research*, 3:1157–1182, 2003.

E. H. Han, D. Boley, M. Gini, R. Gross, K. Hastings, G. Karypis, V. Kumar, B. Mobasher, and J. Moore. WebACE: A web agent for document categorization and exploration. In *Proc. 2nd Int. Conf. Autonomous Agents*, pages 408–415, May 1998.

W. Hersh, C. Buckley, T. J. Leone, and D. Hickam. OHSUMED: An interactive retrieval evaluation and new large test collection for research. In *Proc. ACM SIGIR*, pages 192–201, 1994.

P. Indyk. A sublinear-time approximation scheme for clustering in metric spaces. In *40th Annual IEEE Symp. Foundations of Computer Science*, pages 154–159, 1999.

R. Kannan, S. Vempala, and A. Vetta. On clusterings — good, bad and spectral. In *41st Annual IEEE Symp. Foundations of Computer Science*, pages 367–377, 2000.

G. Karypis. *CLUTO - A Clustering Toolkit*. Dept. of Computer Science, University of Minnesota, May 2002. `http://www-users.cs.umn.edu/~karypis/cluto/`.

G. Karypis and V. Kumar. A fast and high quality multilevel scheme for partitioning irregular graphs. *SIAM Journal on Scientific Computing*, 20(1):359–392, 1998.

M. Kearns, Y. Mansour, and A. Y. Ng. An information-theoretic analysis of hard and soft assignment methods for clustering. In *Proc. 13th Conf. Uncertainty in Artificial Intelligence*, pages 282–293, 1997.

T. Kohonen, S. Kaski, K. Lagus, J. Salojrvi, J. Honkela, V. Paatero, and A. Saarela. Self organization of a massive document collection. *IEEE Trans. Neural Networks*, 11(3):574–585, 2000.

M. H. Law, A. K. Jain, and M. A. T. Figueiredo. Feature selection in mixture-based clustering. In S. T. S. Becker and K. Obermayer, editors, *Advances in Neural Information Processing Systems 15*, pages 625–632, Cambridge, MA, 2003. MIT Press.

J. MacQueen. Some methods for classification and analysis of multivariate observations. In *Proc. 5th Berkeley Symp. Math. Statistics and Probability*, pages 281–297, 1967.

K. V. Mardia. Statistics of directional data. *J. Royal Statistical Society. Series B (Methodological)*, 37(3): 349–393, 1975.

A. McCallum and K. Nigam. A comparison of event models for naive Bayes text classification. In *AAAI Workshop on Learning for Text Categorization*, pages 41–48, 1998.

A. K. McCallum. Bow: A toolkit for statistical language modeling, text retrieval, classification and clustering. Available at `http://www.cs.cmu.edu/~mccallum/bow`, 1996.

M. Meila and D. Heckerman. An experimental comparison of model-based clustering methods. *Machine Learning*, 42:9–29, 2001.

M. Meila and J. Shi. Learning segmentation by random walks. In T. K. Leen, T. G. Dietterich, and V. Tresp, editors, *Advances in Neural Information Processing Systems 13*, pages 873–879. MIT Press, 2001a.

M. Meila and J. Shi. A random walks view of spectral segmentation. In *AI and STATISTICS (AISTATS) 2001*, 2001b.

A. Y. Ng, M. I. Jordan, and Y. Weiss. On spectral clustering: analysis and an algorithm. In T. G. Dietterich, S. Becker, and Z. Ghahramani, editors, *Advances in Neural Information Processing Systems 14*, pages 849–856. MIT Press, 2002.

K. Nigam. *Using Unlabeled Data to Improve Text Classification*. PhD thesis, School of Computer Science, Carnegie Mellon University, May 2001.

E. Rasmussen. Clustering algorithms. In W. Frakes and R. Baeza-Yates, editors, *Information Retrieval: Data Structures and Algorithms*, pages 419–442. Prentice Hall, 1992.

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

N. Slonim and N. Tishby. Document clustering using word clusters via the information bottleneck method. In *Research and Development in Information Retrieval*, pages 208–215, 2000.

N. Slonim and Y. Weiss. Maximum likelihood and the information bottleneck. In S. T. S. Becker and K. Obermayer, editors, *Advances in Neural Information Processing Systems 15*, pages 335–342, Cambridge, MA, 2003. MIT Press.

H. Stark and J. W. Woods. *Probability, Random Processes, and Estimation Theory for Engineers*. Prentice Hall, Englewood Cliffs, New Jersey, 1994.

M. Steinbach, G. Karypis, and V. Kumar. A comparison of document clustering techniques. In *KDD Workshop on Text Mining*, Boston, MA, August 2000.

A. Strehl and J. Ghosh. Cluster ensembles — a knowledge reuse framework for combining partitions. *Journal of Machine Learning Research*, 3:583–617, 2002.

A. Strehl, J. Ghosh, and R. J. Mooney. Impact of similarity measures on web-page clustering. In *AAAI Workshop on AI for Web Search*, pages 58–64, July 2000.

J. Tantrum, A. Murua, and W. Stuetzle. Hierarchical model-based clustering of large datasets through fractionation and refractionation. In *Proc. 8th ACM SIGKDD Int. Conf. Knowledge Discovery and Data Mining*, pages 239–246, 2002.

S. Vaithyanathan and B. Dom. Model-based hierarchical clustering. In *Proc. 16th Conf. Uncertainty in Artificial Intelligence*, pages 599–608, July 2000.

V. Vapnik. *Statistical Learning Theory*. John Wiley, New York, 1998.

Y. Zhao and G. Karypis. Criterion functions for document clustering: experiments and analysis. Technical Report #01-40, Department of Computer Science, University of Minnesota, November 2001.

S. Zhong and J. Ghosh. A comparative study of generative models for document clustering. In *SIAM Int. Conf. Data Mining Workshop on Clustering High Dimensional Data and Its Applications*, San Francisco, CA, May 2003a.

S. Zhong and J. Ghosh. A unified framework for model-based clustering. *Journal of Machine Learning Research*, 4:1001–1037, November 2003b.

Appendix

Table 9: NMI Results on *NG20*, *NG17-19*, *classic*, *ohscal*, and *hitech* datasets with frequency-selected words

	NG20	NG17-19	classic	ohscal	hitech
K	20	3	4	10	6
kberns	.38 ± .02	.03 ± .01	**.92 ± .00**	.39 ± .02	.11 ± .02
skberns	.39 ± .02	.03 ± .01	**.92 ± .00**	.39 ± .02	.11 ± .02
mixberns	.38 ± .02	.03 ± .02	**.92 ± .00**	.38 ± .03	.11 ± .02
daberns	.04 ± .00	.03 ± .00	.13 ± .11	.00 ± .00	.01 ± .00
kmnls	.55 ± .02	.31 ± .09	.56 ± .05	.38 ± .02	.18 ± .02
skmnls	.56 ± .01	.31 ± .10	.57 ± .07	.38 ± .02	.19 ± .02
mixmnls	.56 ± .01	.32 ± .08	.63 ± .06	.38 ± .02	.18 ± .03
damnls	**.62 ± .01**	.46 ± .05	.66 ± .03	.43 ± .01	.23 ± .02
kvmfs	.54 ± .02	.36 ± .11	.57 ± .04	.45 ± .02	.22 ± .03
skvmfs	.46 ± .05	.37 ± .10	.55 ± .03	.43 ± .03	.22 ± .02
softvmfs	.48 ± .01	.45 ± .04	.54 ± .04	.39 ± .02	**.26 ± .02**
davmfs	.58 ± .01	**.47 ± .01**	.49 ± .00	**.47 ± .01**	**.26 ± .02**
CLUTO	.56 ± .00	.46 ± .01	.53 ± .00	**.47 ± .00**	.24 ± .01
co-cluster	.58 ± .01	.13 ± .02	.47 ± .02	.33 ± .00	.23 ± .01

Table 10: NMI Results on *hitech*, *reviews*, *sports*, *la1*, *la12*, and *la2* datasets with frequency-selected words

	reviews	sports	la1	la12	la2
K	5	7	6	6	6
kberns	.33 ± .04	.37 ± .05	.17 ± .03	.23 ± .01	.20 ± .03
skberns	.34 ± .05	.38 ± .05	.18 ± .02	.23 ± .02	.21 ± .03
mixberns	.35 ± .04	.37 ± .05	.18 ± .03	.23 ± .01	.20 ± .03
daberns	.05 ± .01	.03 ± .00	.01 ± .00	.01 ± .00	.01 ± .00
kmnls	.49 ± .10	.58 ± .06	.43 ± .02	.48 ± .04	.41 ± .04
skmnls	**.50 ± .10**	.59 ± .05	.42 ± .04	.47 ± .04	.43 ± .04
mixmnls	.49 ± .10	.58 ± .06	.43 ± .03	.48 ± .04	.42 ± .03
damnls	**.50 ± .01**	.59 ± .04	**.53 ± .02**	**.58 ± .02**	**.54 ± .04**
kvmfs	.46 ± .09	.57 ± .06	.48 ± .05	.49 ± .04	.45 ± .03
skvmfs	.45 ± .08	.57 ± .04	.49 ± .04	.49 ± .06	.50 ± .04
softvmfs	.43 ± .03	.59 ± .04	.47 ± .04	.47 ± .03	.48 ± .03
davmfs	.45 ± .06	.59 ± .03	.52 ± .03	.53 ± .02	**.54 ± .02**
CLUTO	**.50 ± .00**	**.61 ± .00**	**.53 ± .02**	.53 ± .01	.53 ± .01
co-cluster	.34 ± .01	.53 ± .01	.36 ± .01	.45 ± .01	.43 ± .01

Table 11: NMI Results on *k1b*, *tr11*, *tr23*, *tr41*, and *tr45* datasets with frequency-selected words

	k1b	tr11	tr23	tr41	tr45
K	6	9	6	10	10
kberns	$.42 \pm .10$	$.07 \pm .01$	$.11 \pm .01$	$.13 \pm .04$	$.08 \pm .02$
skberns	$.47 \pm .02$	$.07 \pm .01$	$.11 \pm .01$	$.13 \pm .05$	$.09 \pm .02$
mixberns	$.42 \pm .09$	$.07 \pm .01$	$.10 \pm .01$	$.13 \pm .04$	$.08 \pm .02$
daberns	$.05 \pm .00$	$.08 \pm .01$	$.10 \pm .01$	$.03 \pm .00$	$.07 \pm .01$
kmnls	$.50 \pm .03$	$.23 \pm .04$	$.16 \pm .03$	$.44 \pm .06$	$.41 \pm .04$
skmnls	$.50 \pm .04$	$.23 \pm .04$	$.16 \pm .03$	$.44 \pm .07$	$.42 \pm .03$
mixmnls	$.51 \pm .04$	$.24 \pm .04$	$.16 \pm .04$	$.45 \pm .06$	$.42 \pm .04$
damnls	$.64 \pm .03$	$\mathbf{.53 \pm .05}$	$.32 \pm .04$	$\mathbf{.69 \pm .03}$	$.67 \pm .04$
kvmfs	$.56 \pm .04$	$.27 \pm .04$	$.18 \pm .02$	$.54 \pm .04$	$.55 \pm .07$
skvmfs	$.59 \pm .03$	$.39 \pm .04$	$.22 \pm .03$	$.59 \pm .03$	$.62 \pm .05$
softvmfs	$.59 \pm .03$	$.51 \pm .02$	$.37 \pm .03$	$.61 \pm .02$	$\mathbf{.71 \pm .05}$
davmfs	$.65 \pm .04$	$.48 \pm .02$	$.34 \pm .03$	$.65 \pm .02$	$.70 \pm .02$
CLUTO	$\mathbf{.67 \pm .03}$	$.51 \pm .02$	$\mathbf{.39 \pm .02}$	$.66 \pm .02$	$.70 \pm .01$
co-cluster	$.64 \pm .01$	$.43 \pm .01$	$.24 \pm .01$	$.58 \pm .02$	$.56 \pm .03$

Table 12: Summary of t-test results (word features selected by word frequencies).

	Hypothesis tested						
Dataset	bb>wb	bm>wm	bv>wv	dam>bm	dav>bv	dav>dam	dam>cluto
NG20	0.1655	0.2237	**<0.001**	**<0.001**	**<0.001**	**>0.999**	**<0.001**
NG17-19	0.46	0.338	0.009	**<0.001**	0.083	0.336	0.44
classic	**0.007**	**0.009**	0.089	0.062	**>0.999**	**>0.999**	**<0.001**
ohscal	0.216	0.413	**<0.001**	**<0.001**	**0.011**	**>0.999**	**>0.999**
hitech	0.359	0.391	**<0.001**	**<0.001**	0.648	**<0.001**	0.914
reviews	0.263	0.382	0.103	0.499	0.662	0.078	0.502
sports	0.339	0.292	0.132	0.654	0.496	0.427	0.911
la1	0.206	0.298	0.155	**<0.001**	**0.049**	0.884	0.507
la12	0.339	0.369	**0.043**	**<0.001**	**0.017**	**>0.999**	**<0.001**
la2	0.228	0.207	**0.004**	**<0.001**	**0.002**	0.576	0.24
k1b	0.074	0.174	0.054	**<0.001**	**<0.001**	0.138	**0.997**
tr11	0.397	0.181	**<0.001**	**<0.001**	**0.983**	**0.998**	0.108
tr23	0.283	0.439	**<0.001**	**<0.001**	**0.976**	0.076	**0.999**
tr41	0.485	0.323	**<0.001**	**<0.001**	**0.002**	**0.999**	**0.018**
tr45	0.3	0.225	**<0.001**	**<0.001**	0.553	**0.013**	**0.99**

Table 13: Paired t-test results on feature selection (word features selected by word frequencies).

	Hypothesis tested: fs > nfs					
Dataset	kmnls	damnls	kvmfs	davmfs	CLUTO	co-cluster
NG20	**0.006**	**<0.001**	0.789	0.57	**>0.999**	**<0.001**
NG17-19	**0.029**	**<0.001**	0.568	**<0.001**	0.295	**<0.001**
classic	0.486	**<0.001**	**0.046**	**>0.999**	0.9139	**<0.001**
ohscal	**0.044**	**<0.001**	0.065	**0.006**	**<0.001**	**>0.999**
hitech	**>0.999**	0.456	**>0.999**	**0.998**	**>0.999**	0.38
reviews	0.932	0.803	**0.977**	**>0.999**	**>0.999**	**>0.999**
sports	0.585	0.183	0.613	0.548	**>0.999**	**0.985**
la1	**0.031**	**<0.001**	0.666	0.118	**>0.999**	**0.967**
la12	0.452	**<0.001**	**0.991**	0.382	**>0.999**	**>0.999**
la2	0.584	**<0.001**	**0.998**	**<0.001**	**>0.999**	**0.024**
k1b	**0.997**	**<0.001**	**0.975**	**<0.001**	**<0.001**	**0.043**
tr11	**>0.999**	**<0.001**	**>0.999**	**0.995**	**>0.999**	**>0.999**
tr23	0.498	**<0.001**	**>0.999**	0.756	**0.997**	**0.006**
tr41	**0.99**	**<0.001**	**0.998**	**<0.001**	0.817	**<0.001**
tr45	0.785	**<0.001**	**>0.999**	**<0.001**	**<0.001**	**<0.001**

Table 14: NMI Results on *NG20*, *NG17-19*, *classic*, *ohscal*, and *hitech* datasets with variance-selected words

	NG20	NG17-19	classic	ohscal	hitech
K	20	3	4	10	6
kberns	$.34 \pm .02$	$.05 \pm .04$	$\mathbf{.90 \pm .00}$	$.37 \pm .01$	$.20 \pm .02$
skberns	$.34 \pm .02$	$.05 \pm .05$	$.89 \pm .03$	$.38 \pm .02$	$.20 \pm .03$
mixberns	$.33 \pm .03$	$.04 \pm .04$	$\mathbf{.90 \pm .00}$	$.38 \pm .01$	$.20 \pm .03$
daberns	$.02 \pm .00$	$.03 \pm .00$	$.01 \pm .01$	$.00 \pm .00$	$.17 \pm .01$
kmnls	$.55 \pm .02$	$.34 \pm .10$	$.57 \pm .04$	$.37 \pm .02$	$.27 \pm .02$
skmnls	$.55 \pm .02$	$.34 \pm .10$	$.58 \pm .03$	$.37 \pm .02$	$.27 \pm .02$
mixmnls	$.56 \pm .02$	$.34 \pm .11$	$.61 \pm .04$	$.37 \pm .02$	$.27 \pm .02$
damnls	$\mathbf{.62 \pm .01}$	$\mathbf{.49 \pm .04}$	$.67 \pm .04$	$.42 \pm .02$	$.28 \pm .02$
kvmfs	$.40 \pm .02$	$.32 \pm .10$	$.55 \pm .02$	$.44 \pm .02$	$.29 \pm .01$
skvmfs	$.39 \pm .03$	$.36 \pm .05$	$.52 \pm .03$	$.43 \pm .04$	$.29 \pm .02$
softvmfs	$.42 \pm .01$	$.42 \pm .06$	$.53 \pm .03$	$.42 \pm .02$	$.31 \pm .01$
davmfs	$.43 \pm .02$	$.42 \pm .03$	$.49 \pm .00$	$\mathbf{.47 \pm .02}$	$.31 \pm .02$
CLUTO	$.57 \pm .00$	$.44 \pm .00$	$.54 \pm .01$	$.44 \pm .00$	$\mathbf{.34 \pm .01}$
co-cluster	$.51 \pm .00$	$.38 \pm .00$	$.17 \pm .00$	$.35 \pm .01$	$.27 \pm .01$

Table 15: NMI Results on *reviews*, *sports*, *la1*, *la12*, and *la2* datasets with variance-selected words

	reviews	sports	la1	la12	la2
K	5	7	6	6	6
kberns	.39 ± .05	.35 ± .04	.24 ± .04	.24 ± .02	.24 ± .01
skberns	.39 ± .05	.36 ± .07	.23 ± .02	.23 ± .02	.25 ± .01
mixberns	.39 ± .05	.35 ± .06	.23 ± .02	.24 ± .02	.24 ± .01
daberns	.35 ± .04	.03 ± .00	.19 ± .02	.07 ± .07	.22 ± .02
kmnls	.53 ± .05	.54 ± .06	.45 ± .07	.50 ± .05	.43 ± .04
skmnls	.56 ± .06	.54 ± .06	.46 ± .06	.50 ± .04	.45 ± .03
mixmnls	.53 ± .05	.55 ± .06	.46 ± .06	.50 ± .04	.44 ± .03
damnls	**.60 ± .05**	.57 ± .04	.55 ± .03	**.59 ± .02**	.53 ± .03
kvmfs	.51 ± .07	.61 ± .04	.48 ± .06	.53 ± .03	.51 ± .03
skvmfs	.54 ± .07	.63 ± .03	.52 ± .05	.50 ± .05	.52 ± .04
softvmfs	.55 ± .07	.60 ± .05	.50 ± .02	.48 ± .04	.50 ± .03
davmfs	**.60 ± .07**	.61 ± .06	.52 ± .02	.55 ± .02	.53 ± .03
CLUTO	.51 ± .00	**.67 ± .01**	**.59 ± .00**	.55 ± .01	**.55 ± .01**
co-cluster	.40 ± .02	.57 ± .01	.50 ± .01	.50 ± .01	.50 ± .01

Table 16: NMI Results on *k1b*, *tr11*, *tr23*, *tr41*, and *tr45* datasets with variance-selected words

	k1b	tr11	tr23	tr41	tr45
K	6	9	6	10	10
kberns	.54 ± .03	.34 ± .04	.18 ± .02	.43 ± .05	.41 ± .03
skberns	.56 ± .03	.36 ± .03	.17 ± .02	.42 ± .04	.42 ± .04
mixberns	.56 ± .03	.36 ± .03	.18 ± .02	.43 ± .05	.43 ± .04
daberns	.55 ± .03	.27 ± .05	.16 ± .01	.35 ± .01	.33 ± .07
kmnls	.57 ± .06	.53 ± .04	.22 ± .06	.58 ± .06	.50 ± .05
skmnls	.57 ± .06	.53 ± .03	.22 ± .05	.59 ± .05	.50 ± .05
mixmnls	.57 ± .05	.54 ± .04	.22 ± .05	.59 ± .05	.50 ± .05
damnls	.64 ± .03	.62 ± .03	.29 ± .02	.68 ± .02	**.71 ± .02**
kvmfs	.58 ± .04	.58 ± .02	.31 ± .04	.63 ± .02	.66 ± .06
skvmfs	.61 ± .03	.61 ± .03	.31 ± .02	.65 ± .02	.68 ± .06
softvmfs	.61 ± .02	**.63 ± .03**	**.32 ± .03**	.66 ± .04	.66 ± .03
davmfs	.63 ± .03	**.63 ± .03**	**.32 ± .03**	**.69 ± .02**	.65 ± .04
CLUTO	.60 ± .01	.59 ± .00	.28 ± .00	.64 ± .00	.67 ± .01
co-cluster	**.69 ± .03**	.53 ± .02	.17 ± .03	.58 ± .02	.49 ± .01

Table 17: Summary of t-test results (word features selected by word variances).

	Hypothesis tested						
Dataset	bb>wb	bm>wm	bv>wv	dam>bm	dav>bv	dav>dam	dam>cluto
NG20	0.133	0.115	**<0.001**	**<0.001**	**<0.001**	**0.999**	**<0.001**
NG17-19	0.364	0.446	**<0.001**	**<0.001**	0.612	**>0.999**	**<0.001**
classic	0.289	**0.048**	**0.002**	**0.003**	**>0.999**	**>0.999**	**<0.001**
ohscal	0.094	0.254	**0.015**	**<0.001**	0.385	**>0.999**	**0.996**
hitech	0.301	0.336	**0.005**	0.098	0.509	**0.005**	**>0.999**
reviews	0.4	0.141	**0.007**	0.057	**0.014**	**0.026**	**<0.001**
sports	0.396	0.444	**<0.001**	0.155	0.352	**0.997**	**>0.999**
la1	0.215	0.371	0.309	**<0.001**	**0.983**	**>0.999**	**0.999**
la12	0.265	0.415	0.003	**<0.001**	**0.996**	**>0.999**	**<0.001**
la2	0.227	0.208	0.11	**<0.001**	0.252	**>0.999**	**0.987**
k1b	0.086	0.45	0.081	**0.003**	**0.02**	**>0.999**	**0.002**
tr11	0.131	0.449	0.059	**<0.001**	0.065	**0.981**	**0.015**
tr23	0.313	0.398	0.45	**<0.001**	0.583	**>0.999**	0.083
tr41	0.377	0.314	**<0.001**	**<0.001**	**0.996**	**>0.999**	**<0.001**
tr45	0.127	0.414	0.249	**<0.001**	0.516	**>0.999**	**<0.001**

Table 18: Paired *t*-test results on feature selection (word features selected by word variances).

	Hypothesis tested: fs > nfs					
Dataset	kmnls	damnls	kvmfs	davmfs	CLUTO	co-cluster
NG20	**0.029**	**<0.001**	0.494	**0.003**	**>0.999**	**<0.001**
NG17-19	**0.01**	**<0.001**	**0.981**	**0.009**	**>0.999**	**<0.001**
classic	0.283	**<0.001**	0.133	**>0.999**	0.616	**<0.001**
ohscal	0.415	**<0.001**	**>0.999**	**>0.999**	0.33	**>0.999**
hitech	**0.007**	**<0.001**	0.109	**0.002**	**0.002**	**<0.001**
reviews	0.754	**0.003**	0.228	**<0.001**	**0.963**	0.424
sports	0.948	0.456	**0.998**	**>0.999**	**0.016**	0.108
la1	**0.022**	**<0.001**	**>0.999**	**>0.999**	0.063	**<0.001**
la12	0.092	**<0.001**	**>0.999**	**>0.999**	0.841	**<0.001**
la2	0.164	**<0.001**	**>0.999**	**>0.999**	0.926	**<0.001**
k1b	0.137	**<0.001**	**0.996**	0.069	0.933	**<0.001**
tr11	**<0.001**	**<0.001**	**<0.001**	**<0.001**	**>0.999**	0.523
tr23	**0.002**	**0.005**	**>0.999**	**>0.999**	**>0.999**	**>0.999**
tr41	**<0.001**	**<0.001**	**0.999**	**0.995**	**>0.999**	**<0.001**
tr45	**0.001**	**<0.001**	**>0.999**	**>0.999**	**<0.001**	0.687