A Comparative Study of Generative Models for Document Clustering

Shi Zhong and Joydeep Ghosh
Department of Electrical and Computer Engineering
The University of Texas at Austin
Austin, TX 78712-1084

Abstract

Generative models based on the multivariate Bernoulli and multinomial distributions have been widely used for text classification. Recently, the spherical k-means algorithm, which has desirable properties for text clustering, has been shown to be a special case of a generative model based on a mixture of von Mises-Fisher (vMF) distributions. This paper compares these three probabilistic models for text clustering, both theoretically and empirically, using a general model-based clustering framework. For each model, we investigate three strategies for assigning documents to models: maximum likelihood (k-means) assignment, stochastic assignment, and soft assignment. Our experimental results over a large number of datasets show that, in terms of clustering quality, (a) The Bernoulli model is the worst for text clustering; (b) The vMF model produces better clustering results than both Bernoulli and multinomial models; (c) Soft assignment leads to comparable or slightly better results than hard assignment. We also use deterministic annealing (DA) to improve the vMF-based soft clustering and compare all the model-based algorithms with the state-of-the-art discriminative approach to document clustering based on graph partitioning (CLUTO) and a spectral co-clustering method. Overall, CLUTO and DA perform the best but are also the most computationally expensive; the spectral co-clustering algorithm fares worse than the vMF-based methods.

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 and Vivisimo. 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.

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 & Modha, 2001), bi-secting k-means (Steinbach et al., 2000), mixture of multinomials (Vaithyanathan & Dom, 2000; Meila & Heckerman, 2001), multi-level graph partitioning (Karypis, 2002), and co-clustering using bipartite spectral graph partitioning (Dhillon, 2001). Most clustering methods proposed for data mining (Berkhin, 2002) can be divided into two categories: *discriminative* (or similarity-based) approaches (Indyk, 1999; Scholkopf & Smola, 2001; Vapnik, 1998) and *generative* (or model-based) approaches (Blimes, 1998; Rose, 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.

In this paper we focus on model-based approaches since they provide several advantages. First, model-based partitional clustering algorithms often have a complexity of $O(n)$, where n is the number of data samples. 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., Banerjee and Ghosh (2002) and Sikkonen and Kaski (2001). 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 & Ghosh, 2002), which includes a generic treatment of model-based partitional clustering methods. Basically, a generic model-based partitional clustering algorithm centers around two steps—a model re-estimation step and a data re-assignment step. This two-step view allows one to easily combine different models with different assignment strategies. We shall exploit this property in this paper to assess several data assignment strategies from an objective function point of view.

Specifically, we shall describe and compare three probabilistic models—multivariate Bernoulli, multinomial, and von Mises-Fisher, for clustering documents, for three types of data assignments each, leading to a total of nine algorithms. All the three models directly handle high dimensional vectors without dimensionality reduction, and have been recommended for document clustering, which involves grouping of vectors that are high-dimensional, sparse with only non-negative entries, and directional (i.e. only the vectors' directions are important as they are typically normalized to unit length). In contrast, Gaussian based models such as k-means perform very poorly for such datasets (Strehl et al., 2000). All nine instantiated algorithms are compared on a number of document datasets derived from the TREC collections and internet newsgroups. 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 used deterministic annealing (DA) as a more sophisticated soft clustering approach on vMF models and compared all the model-based algorithms with the state-of-the-art graph-based approaches, the CLUTO (Karypis, 2002) algorithm and a bipartite spectral method.

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, Karypis, and Kumar (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) have 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.

The organization of this paper is as follows. Section 2 summarizes a general model-based partitional clustering framework and analyzes several data assignment strategies. Section 3 describes

```
Algorithm: mk-means
Input: Data samples $O = \{o_1, ..., o_n\}$, and model structure $\Lambda = \{\lambda_1, ..., \lambda_k\}$, where $\lambda_j$ is the set of
    parameters for model $j$ and $\Lambda$ the set of all parameters. Each model represents a cluster.
Output: Trained models $\Lambda$ and a partition of the data samples given by the cluster identity vector
    $Y = \{y_1, ..., y_n\},\ y_i \in \{1, ..., k\}$ .
Steps:
    1. Initialization: initialize the cluster identity vector $Y$;
    2a. Model re-estimation: for each cluster $j$, let $O_j = \{o_i | y_i = j\}$, the parameters of each model
        $\lambda_j$ are re-estimated as $\lambda_j = \arg\max_{\lambda} \sum_{o \in O_j} \log P(o|\lambda)$ ;
    2b. Sample re-assignment: for each data sample $i$, set $y_i = \arg\max_j \log P(o_i|\lambda_j)$ ;
    3. Stop if $Y$ does not change, otherwise go back to Step 2a.
```

Figure 1: Model-based k-means algorithm.

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

The model-based k-means (*mk-means*) algorithm (Fig. 1) is a generalized version of the standard k-means. It assumes that there are k parameterized model, one for each cluster. The set of parameters of the i-th model is denoted by λ_i, and typically all the models are from the same family, e.g., the family of exponential distributions. Basically, the algorithm iterates between a model re-estimation step 2a and a sample re-assignment step 2b. In Fig. 1, the maximum likelihood (ML) assignment is used for the latter step. Alternatively, one can employ a soft assignment, as in EM clustering, where a sample o gets fractionally assigned to all k clusters according to the posterior probability $P(j|o, \Lambda)$, and each model is trained using the posterior probability weighted samples. An information-theoretic analysis of these two assignment strategies was given by Kearns, Mansour, and Ng (1997), and an empirical study was made by Meila and Heckerman (2001) for multinomial models. A variant of the ML assignment strategy is stochastic assignment, where a sample o is stochastically assigned to exactly one of the k clusters following the posterior probability $P(j|o, \Lambda)$. We call the clustering algorithm with this assignment strategy *stochastic mk-means*.

In soft clustering, an additional set of parameters, the mixture weights that specify the prior probabilities of component models, are introduced. One can generalize these weight parameters to construct a general objective function (to be maximized) for model-based partitional clustering (Zhong & Ghosh, 2003):

$$\log P(O|\Lambda) = \sum_{i=1}^{n} \log \left(\sum_{j=1}^{k} \alpha_{ij} P(o_i|\lambda_j) \right), \tag{1}$$

where $\Lambda = \{\lambda_j, \alpha_{ij}\}_{i=1,...,n,j=1,...,k}$ is the set of all model parameters to be estimated, and α_{ij}'s are the model mixture weights that are subject to the constraints $\sum_j \alpha_{ij} = 1,\ \forall i$. Applying EM algorithm to maximize (1), one can derive the general re-estimation formula for Λ as follows:

$$\lambda_j^{(new)} = \arg\max_{\lambda} \sum_i P(j|o_i, \Lambda) \log P(o_i|\lambda) \ , \tag{2}$$

$$\alpha_{ij}^{(new)} = P(j|o_i, \Lambda) \ , \tag{3}$$

where $P(j|o_i, \Lambda)$ is the posterior probability of cluster j given sample o_i and parameters Λ,

$$P(j|o_i, \Lambda) = \frac{\alpha_{ij} P(o_i|\lambda_j)}{\sum_{j'} \alpha_{ij'} P(o_i|\lambda_{j'})} \ . \tag{4}$$

Equation (2) is the model re-estimation step and the posterior probabilities $P(j|o_i, \Lambda)$'s are the sample re-assignment weights.

Several popular algorithms differ in how the α_{ij}'s in (3) or $P(j|o_i, \Lambda)$'s in (4) are set. Setting $\alpha_{ij} = I(y_i = j)$[1], where the cluster identity $y_i = \arg\max_{j'} \log P(o_i|\lambda_{j'})$, leads to the mk-means algorithm. Constraining α_{ij}'s to be independent of individual data samples, i.e., $\alpha_{ij} = \alpha_j, \forall i$, results in the EM clustering algorithm. In this case, the re-estimation of α's (3) needs to be modified to $\alpha_j^{(new)} = \frac{1}{n} \sum_i P(j|o_i, \Lambda)$. For stochastic mk-means, $\alpha_{ij} = I(\xi_i = j)$, where ξ_i is a discrete random variable that takes value j with the posterior probability (4). It is worth emphasizing that different assignment strategies do not change the generic model re-estimation formula in (2) and we can plug any probabilistic model into the formulation and get a series of model-based clustering algorithms with different assignment methods.

The computational complexity for all the algorithms described above is linear in the number of data samples n, provided that we use a constant (maximum) number of iterations and a model training algorithm that has linear complexity. Specifically, the complexity is $O(kn)$ for mk-means and stochastic mk-means and $O(k^2 n)$ for soft EM clustering.

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"[2] counts in the document. The dimensionality equals the number of words in the vocabulary used. Next we briefly introduce the three generative models studied in our experiments.

3.1 Multivariate Bernoulli model

In the multivariate Bernoulli model (McCallum & Nigam, 1998), a document is represented as a binary vector over the space of words. The l-th dimension of the vector representing document d_i is denoted by b_{il}, 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. With naïve Bayes assumption, the probability of a document d_i in cluster j is

$$P(d_i|\lambda_j) = \prod_l P_j(w_l)^{b_{il}} (1 - P_j(w_l))^{1-b_{il}} \ , \tag{5}$$

[1] $I(.)$ is an indicator function that takes value 1 when the predicate argument is true and 0 otherwise.

[2] Used in a broad sense since it may represent individual words, stemmed words, tokenized words, or short phrases.

where $\lambda_j = \{P_j(w_l)\}$, $P_j(w_l)$ is the probability of word w_l being present in cluster j, and $(1-P_j(w_l))$ the probability of word w_l not being present in cluster j. To avoid zero probabilities when estimating $P_j(w_l)$, one can employ a Laplacian prior and derive the solution as (McCallum & Nigam, 1998)

$$P_j(w_l) = \frac{1 + \sum_i P(j|d_i, \Lambda) b_{il}}{2 + \sum_i P(j|d_i, \Lambda)} \ , \tag{6}$$

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

3.2 Multinomial model

Based on the naïve Bayes assumption, a multinomial model for cluster j represents a document d_i by a multinomial distribution of the words in the document

$$P(d_i|\lambda_j) = \prod_l P_j(w_l)^{n_{il}} \ , \tag{7}$$

where n_{il} is the number of word w_l's occurrences in document d_i. Note the $P_j(w_l)$'s here represent the word distribution in cluster j and are subject to $\sum_l P_j(w_l) = 1$. They are different from the $P_j(w_l)'s$ in (5) and can be estimated by counting the number of documents in each cluster and the number of times w_l occurs in all documents in the cluster j (Nigam, 2001). With Laplacian smoothing, the parameter estimation of multinomial models amounts to

$$P_j(w_l) = \frac{1 + \sum_i P(j|d_i, \Lambda) n_{il}}{\sum_l (1 + \sum_i P(j|d_i, \Lambda) n_{il})} = \frac{1 + \sum_i P(j|d_i, \Lambda) n_{il}}{|V| + \sum_l \sum_i P(j|d_i, \Lambda) n_{il}} \ , \tag{8}$$

where $|V|$ is the size of the word vocabulary, i.e., the dimensionality of document vectors.

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). It has recently been shown that the spherical k-means algorithm that uses the cosine similarity metric (to measure the closeness of a data point to its cluster's centroid) can be derived from a generative model based on the vMF distribution under certain restrictive conditions (Banerjee & Ghosh, 2002; Banerjee et al., 2003). The vMF distribution for cluster j can be written as

$$P(d_i|\lambda_j) = \frac{1}{Z(\kappa_j)} \exp\left(\kappa_j \frac{d_i^T \mu_j}{\|\mu_j\|}\right) \ , \tag{9}$$

where d_i is a normalized (unit-length in L_2 norm) document vector and the Bessel function $Z(\kappa_j)$ is a normalization term. The κ measures the directional variance (or dispersion) and the higher it is, the more peaked the distribution is. For the vMF-based k-means algorithm, we assume κ is the same for all clusters, i.e., $\kappa_j = \kappa, \forall j$. This results in the spherical k-means (Dhillon & Modha, 2001; Dhillon et al., 2001). The model estimation in this case simply amounts to $\mu_j = \frac{1}{n_j} \sum_{i:y_i=j} d_i$, where n_j is the number of documents in cluster j. 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. Even using an approximation for estimating κ's, however, it is computationally much more expensive than the vMF-based k-means algorithm. 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 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 $I(X;Y)$ between a *r.v.* X, governing the cluster labels, and a *r.v.* Y, governing the class labels, is a superior measure than purity or entropy (Strehl & Ghosh, 2002). Moreover, by normalizing this measure to lie in the range [0,1], it becomes quite impartial to k. There are several choices for normalization based on the entropies $H(X)$ and $H(Y)$. We shall follow the definition of normalized mutual information (NMI) using geometrical mean, $NMI = \frac{I(X;Y)}{\sqrt{H(X) \cdot H(Y)}}$, as given in (Strehl & Ghosh, 2002), In practice, we use a sample estimate

$$NMI = \frac{\sum_{h,l} n_{h,l} \log\left(\frac{n \cdot n_{h,l}}{n_h n_l}\right)}{\sqrt{\left(\sum_h n_h \log \frac{n_h}{n}\right)\left(\sum_l n_l \log \frac{n_l}{n}\right)}} \;, \tag{10}$$

where n_h is the number of data samples in class h, n_l the number of samples in cluster l and $n_{h,l}$ the number of samples 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 & Ghosh, 2002). In our experiments, we use NMI as the evaluation criterion.

4.2 Text datasets

We used the 20-newsgroups data[3] and a number of datasets from the CLUTO toolkit[4] (Karypis, 2002). These datasets provide a good representation of different characteristics: number of documents ranges from 204 to 19949, number of terms 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.

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

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

Table 1: Summary of text datasets (for each dataset, n_d is the total number of documents, n_w the total number of terms, 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

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 $\sim$ 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 & 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[5]. 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 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.

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

4.3 Experimental setting

We compare nine model-based clustering algorithms, each resulting from the combination of one of the three models with one of the three assignment strategies. For example, the three algorithms based on the Bernoulli model are k-Bernoullis, stochastic k-Bernoullis, and mixture-of-Bernoullis, abbreviated as *kberns*, *skberns*, and *mixberns*, respectively. Similarly, the abbreviated names are *kmns*, *skmns*, and *mixmns* for multinomial-based algorithms, and are *kvmfs*, *skvmfs*, and *softvmfs* 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. Realizing that this is purely ad-hoc, we also implemented the standard deterministic annealing for the *softvmfs*: (a) a constant κ is used for all clusters at each iteration; (b) the algorithm runs until convergence for each κ; (c) κ follows 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 use log(IDF)-weighted and normalized document vectors.

For all the model-based algorithms (except for *davmfs*), we use a maximum number of iterations of 20 (to make a fair comparison). 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.

Two state-of-the-art graph-based clustering algorithms are also included in our experiments. The first one is CLUTO (Karypis, 2002), a clustering toolkit based on the Metis graph partitioning algorithms (Karypis & Kumar, 1998). It is worth mentioning that CLUTO is positioned for clustering and drops the strong balance constraints in the original Metis partitioning. We use *vcluster* in the toolkit with the default setting. The other one is a modification of the bipartite spectral co-clustering algorithm (Dhillon, 2001). The modification is according to (Ng et al., 2002)[6] and generates slightly better results than the original bipartite clustering algorithm. Both graph partitioning algorithms uses fast heuristics and thus is dependent on the order of nodes from the input graph. We run each algorithm ten times, each run using a different order of documents.

4.4 Results

Table 2 shows the NMI results on the *NG20* and *NG17-19* datasets and Table 3 the NMI results on the classic and ohscal datasets, across different number of clusters for each dataset. All numbers in the table are shown in the format *average* ± *standard deviation*. To save space, we show the NMI results on all other datasets for one specific k only (Table 4 and Table 5).

Of the three models, the vMF model appears to be the best and the multivariate Bernoulli model the worst. The Bernoulli-based algorithms significantly underperform the other methods for all the datasets except for *ohscal*. This indicates that noting only whether a word occurs or not 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*, *tr11*, *tr23*, *tr41*, and *tr45*, etc. The deterministic annealing algorithm improves the performance of *softvmfs*, sometimes significantly, as shown in Table 2 & 5.

The three different data assignment strategies produce very comparable clustering results across all datasets. The soft assignment is only slightly better than the other two (except for the *classic* dataset where the soft assignment with the multinomial model is clearly the best). For the vMF

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

Table 2: *NMI* Results on *NG20* and *NG17-19* datasets

	NG20				NG17-19			
k	10	**20**	30	40	**3**	5	7	9
kberns	.18 ± .03	.20 ± .04	.18 ± .03	.18 ± .02	.03 ± .01	.09 ± .05	.08 ± .03	.09 ± .05
skberns	.19 ± .04	.21 ± .03	.19 ± .02	.20 ± .03	.03 ± .01	.08 ± .05	.09 ± .04	.09 ± .05
mixberns	.18 ± .05	.19 ± .03	.17 ± .02	.18 ± .03	.03 ± .01	.08 ± .04	.08 ± .04	.08 ± .05
kmns	.50 ± .02	.53 ± .03	.53 ± .02	.54 ± .02	.23 ± .08	.26 ± .05	.23 ± .04	.23 ± .04
skmns	.51 ± .02	.53 ± .03	.54 ± .02	.55 ± .02	.22 ± .08	.26 ± .06	.24 ± .05	.23 ± .04
mixmns	.52 ± .02	.54 ± .03	.54 ± .02	.56 ± .02	.23 ± .08	.27 ± .05	.25 ± .04	.25 ± .04
kvmfs	.53 ± .02	.55 ± .02	.52 ± .01	.50 ± .01	.37 ± .10	.37 ± .02	.33 ± .03	.32 ± .03
skvmfs	.54 ± .01	.56 ± .01	.48 ± .16	.52 ± .01	.37 ± .08	.37 ± .05	.38 ± .03	.35 ± .03
softvmfs	.55 ± .02	.57 ± .02	.56 ± .01	.55 ± .01	.39 ± .10	.40 ± .04	.39 ± .04	.37 ± .02
davmfs	**.57 ± .03**	**.59 ± .02**	.57 ± .01	.56 ± .01	**.46 ± .01**	**.40 ± .02**	.41 ± .03	.39 ± .02
CLUTO	.55 ± .02	.58 ± .01	**.58 ± .01**	**.57 ± .01**	**.46 ± .01**	**.40 ± .01**	**.45 ± .01**	**.43 ± .01**
co-cluster	.36 ± .01	.46 ± .01	.50 ± .01	.51 ± .01	.02 ± .01	.16 ± .07	.36 ± .03	.37 ± .01

Table 3: *NMI* Results on *classic* and *ohscal* datasets

	classic				ohscal			
k	**4**	6	8	10	5	**10**	15	20
kberns	.23 ± .10	.25 ± .11	.25 ± .08	.26 ± .07	.37 ± .02	.37 ± .02	.38 ± .02	.38 ± .03
skberns	.23 ± .11	.22 ± .13	.21 ± .10	.27 ± .16	.38 ± .01	.38 ± .02	.39 ± .02	.39 ± .03
mixberns	.20 ± .15	.18 ± .15	.18 ± .12	.18 ± .17	.38 ± .01	.37 ± .02	.38 ± .02	.38 ± .03
kmns	.56 ± .06	.58 ± .03	.58 ± .03	.58 ± .02	.37 ± .01	.37 ± .02	.37 ± .01	.36 ± .01
skmns	.57 ± .06	.58 ± .03	.58 ± .02	.56 ± .02	.37 ± .01	.37 ± .02	.37 ± .02	.37 ± .01
mixmns	**.66 ± .04**	**.65 ± .04**	**.64 ± .03**	**.65 ± .02**	.37 ± .01	.37 ± .02	.38 ± .02	.37 ± .01
kvmfs	.54 ± .03	.60 ± .04	.57 ± .05	.56 ± .04	.40 ± .03	.43 ± .03	.41 ± .01	.39 ± .01
skvmfs	.54 ± .02	.63 ± .04	.61 ± .03	.57 ± .03	.39 ± .02	.44 ± .02	.41 ± .01	.38 ± .01
softvmfs	.55 ± .03	.61 ± .06	.60 ± .03	.58 ± .02	.40 ± .02	.44 ± .02	.41 ± .01	.41 ± .01
davmfs	.51 ± .01	.62 ± .01	.60 ± .01	.59 ± .01	.41 ± .01	**.47 ± .02**	**.45 ± .01**	.42 ± .01
CLUTO	.54 ± .02	.64 ± .01	.60 ± .01	.58 ± .01	**.44 ± .01**	.44 ± .02	.44 ± .01	**.43 ± .01**
co-cluster	.01 ± .01	.43 ± .02	.43 ± .02	.59 ± .03	.39 ± .01	.39 ± .01	.36 ± .01	.38 ± .01

models, however, the exact EM clustering (Banerjee et al., 2003) can achieve significant improvement over hard assignment.

Surprisingly, the bipartite spectral co-clustering algorithm mostly underperforms the vMF-based methods and sometimes gives very poor results (with NMI values close to 0). The other graph-based algorithm, CLUTO, performs much better and is overall the best among all the algorithms we have compared. This is not surprising since it is built on a very sophisticated multi-level graph partitioning engine (Karypis & Kumar, 1998). The disadvantage of this approach, and similarity-based algorithms in general, lies in its $O(n^2)$ computational complexity.

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. It also means that if we can develop a good initialization strategy to make the results of model-based clustering lean towards the upper end of the performance range, we shall see results comparable to the CLUTO results. 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 6 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 memory large enough to hold an individual dataset, and reflect only the clustering time, not including the data I/O cost.

Table 4: *NMI* Results on *hitech*, *reviews*, *sports*, *la1*, *la12*, and *la2* datasets

	hitech	reviews	sports	la1	la12	la2
k	6	5	7	6	6	6
kberns	.11 ± .05	.30 ± .05	.39 ± .06	.04 ± .04	.06 ± .06	.17 ± .03
skberns	.11 ± .03	.30 ± .04	.37 ± .05	.06 ± .05	.07 ± .06	.19 ± .03
mixberns	.11 ± .04	.29 ± .05	.37 ± .05	.05 ± .05	.06 ± .05	.20 ± .04
kmns	.23 ± .03	.55 ± .08	.59 ± .06	.39 ± .05	.42 ± .04	.47 ± .04
skmns	.23 ± .04	.55 ± .08	.58 ± .06	.41 ± .05	.43 ± .04	.47 ± .05
mixmns	.23 ± .03	**.56 ± .08**	.59 ± .06	.41 ± .05	.43 ± .05	.48 ± .04
kvmfs	.28 ± .02	.53 ± .06	.57 ± .08	.49 ± .05	.50 ± .03	.54 ± .04
skvmfs	.29 ± .02	.53 ± .07	.61 ± .04	.51 ± .04	.51 ± .04	.52 ± .03
softvmfs	.29 ± .01	**.56 ± .06**	.60 ± .05	.52 ± .04	.53 ± .05	.49 ± .04
davmfs	.30 ± .01	**.56 ± .09**	.62 ± .05	.53 ± .03	.52 ± .02	.52 ± .04
CLUTO	**.33 ± .01**	.52 ± .01	**.67 ± .01**	**.58 ± .02**	**.56 ± .01**	**.56 ± .01**
co-cluster	.22 ± .03	.40 ± .07	.56 ± .02	.41 ± .05	.42 ± .07	.41 ± .02

Table 5: *NMI* Results on *k1b*, *tr11*, *tr23*, *tr41*, and *tr45* datasets

	k1b	tr11	tr23	tr41	tr45
k	6	9	6	10	10
kberns	.32 ± .25	.07 ± .02	.11 ± .01	.27 ± .05	.13 ± .06
skberns	.36 ± .24	.08 ± .02	.11 ± .01	.27 ± .06	.13 ± .05
mixberns	.31 ± .24	.07 ± .02	.11 ± .01	.27 ± .04	.13 ± .06
kmns	.55 ± .04	.39 ± .07	.15 ± .03	.49 ± .03	.43 ± .05
skmns	.55 ± .05	.39 ± .08	.15 ± .02	.50 ± .04	.43 ± .05
mixmns	.56 ± .04	.39 ± .07	.15 ± .03	.50 ± .03	.43 ± .05
kvmfs	.60 ± .03	.52 ± .03	.33 ± .05	.59 ± .03	.65 ± .03
skvmfs	.60 ± .02	.57 ± .04	.34 ± .05	.62 ± .03	.65 ± .05
softvmfs	.60 ± .04	.60 ± .05	.36 ± .04	.62 ± .05	.66 ± .03
davmfs	**.67 ± .04**	.66 ± .04	.41 ± .03	**.69 ± .02**	**.68 ± .05**
CLUTO	.62 ± .03	**.68 ± .02**	**.43 ± .02**	.67 ± .01	.62 ± .01
co-cluster	.60 ± .01	.53 ± .03	.22 ± .01	.51 ± .02	.50 ± .03

Clearly, algorithms using soft assignment take longer time than those using hard assignments, suggesting that we should choose the hard versions in practice when the soft version does not buy much performance (this seems to be the case for Bernoulli and multinomial models according to the NMI results presented above). 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 the first nine model-based algorithms, if re-written in C, will be considerably faster than CLUTO.

5 Concluding remarks

At present, we are unaware of any comprehensive comparative study of generative models for document clustering, or a comparison of such models with discriminative ones. A central goal of this work is to fill this void. We have presented a general framework for model-based partitional clustering that is then used to describe and compare three probabilistic models—multivariate Bernoulli, multinomial, and von Mises-Fisher—for clustering documents. Empirical results across a large number of high dimensional text datasets highlighted the following trends: (a) the Bernoulli model is not appropriate for text clustering; (b) the von Mises-Fisher model often outperforms the multi-

Table 6: Running time Results on *NG20* dataset (in seconds)

	NG20				
k	10	20	30	40	50
kberns	26.8 ± 10.6	43.0 ± 19.0	81.6 ± 37.6	125.4 ± 43.6	132.0 ± 54.6
skberns	30.2 ± 9.8	65.9 ± 22.1	92.3 ± 35.2	144.7 ± 51.8	153.5 ± 60.7
mixberns	28.5 ± 11.4	77.8 ± 25.4	102.0 ± 38.9	164.9 ± 38.9	165.7 ± 70.3
kmns	17.5 ± 2.9	36.7 ± 4.9	54.8 ± 7.0	78.5 ± 8.4	97.1 ± 11.2
skmns	19.7 ± 3.0	39.1 ± 5.6	68.4 ± 7.0	94.9 ± 9.9	106.7 ± 10.0
mixmns	23.8 ± 3.6	47.7 ± 6.8	74.2 ± 10.0	99.5 ± 12.7	125.6 ± 16.0
kvmfs	11.4 ± 1.3	17.5 ± 0.3	21.7 ± 0.1	25.5 ± 0.1	29.1 ± 0.1
skvmfs	16.1 ± 0.1	24.4 ± 0.2	29.0 ± 9.2	39.1 ± 0.1	46.2 ± 0.1
softvmfs	34.5 ± 2.2	76.8 ± 1.8	121.7 ± 0.1	178.8 ± 0.2	225.5 ± 0.5
davmfs	288.4 ± 10.0	671.4 ± 21.4	1050.7 ± 26.2	1584.0 ± 39.7	1973.8 ± 56.5
CLUTO[a]	18.6 ± 1.8	22.6 ± 1.7	25.1 ± 1.7	27.0 ± 1.7	28.9 ± 1.7
co-cluster	20.9 ± 0.5	39.9 ± 1.0	62.8 ± 0.7	102.9 ± 0.8	148.5 ± 4.0

nomial model for clustering documents; (c) the algorithms using soft assignment run slower and only slightly improve the clustering performance for the Bernoulli and multinomial models.

Note that the *softvmfs* used in this paper is not a full-fledged EM algorithm. Concurrent work at UT 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 (Banerjee et al., 2003). Preliminary investigation indicates that the superior results are due to an annealing effect produced by using small initial κ's which are then automatically determined/annealed by the EM procedure. This is analogous to using very large initial variances for a mixture-of-Gaussians model. Our results on using deterministic annealing for soft vMF-based clustering also show that significant improvements can be obtained. We are currently studying this issue further to determine how annealing can be used to augment the other model-based soft clustering methods.

All the model-based algorithms (without DA) have a computational advantage over graph-partitioning based approaches but need better initialization strategies to generate clustering results that are comparable to CLUTO. 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. (2002) 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 & Ghosh, 2002), so online extensions of the other model-based approaches need to be investigated.

There are several observations that are currently not fully explained and need more examination. The mixture-of-multinomials, though not so competitive in general, perform the best for the *classic* dataset. Is this because there are a small number of well separated clusters? Secondly, why does the spectral co-clustering perform well on the *k1b* dataset and *classic* dataset (when k=10), but very poorly on other datasets? Can other spectral clustering methods, e.g., Kannan et al. (2000) and Ng et al. (2002), fare better? This matter also deserves further investigation.

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

Acknowledgments

We thank Inderjit Dhillon, Yuqiang Guan, and Arindam Banerjee for helpful discussions on the results, and thank anonymous reviewers for helpful comments that have improved the technical quality of this paper. This work was partly supported by an IBM Faculty Partnership Award from IBM/Tivoli and IBM ACAS, and research grant #8032 from Intel.

References

Banerjee, A., Dhillon, I. S., Ghosh, J., & Sra, S. (2003). *Clustering on hyperspheres using Expectation Maximization* (Technical Report TR-03-07). Department of Computer Sciences, University of Texas.

Banerjee, A., & Ghosh, J. (2002). Frequency sensitive competitive learning for clustering on high-dimensional hyperspheres. *Proc. IEEE Int. Joint Conf. Neural Networks* (pp. 1590–1595).

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

Blimes, J. A. (1998). *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.

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

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

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

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

Dhillon, I. S., Fan, J., & Guan, Y. (2001). Efficient clustering of very large document collections. In R. L. Grossman, C. Kamath, P. Kegelmeyer, V. Kumar and R. R. Namburu (Eds.), *Data mining for scientific and engineering applications*, 357–381. Kluwer Academic publishers.

Dhillon, I. S., Guan, Y., & Kogan, J. (2002). Iterative clustering of high dimensional text data augmented by local search. *Proc. IEEE Int. Conf. Data Mining.*

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

Ghosh, J. (2003). Scalable clustering methods for data mining. In N. Ye (Ed.), *Handbook of data mining.* Lawrence Erlbaum.

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

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

Indyk, P. (1999). A sublinear-time approximation scheme for clustering in metric spaces. *Proc. 40th Symposium on Foundations of Computer Science* (pp. 154–159).

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

Karypis, G. (2002). *CLUTO - a clustering toolkit.* Dept. of Computer Science, University of Minnesota.

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

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

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

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

McCallum, A., & Nigam, K. (1998). A comparison of event models for naive Bayes text classification. *AAAI Workshop on Learning for Text Categorization* (pp. 41–48).

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

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

Ng, A. Y., Jordan, M. I., & Weiss, Y. (2002). On spectral clustering: analysis and an algorithm. *Advances in Neural Information Processing Systems* (pp. 849–856). MIT Press.

Nigam, K. (2001). *Using unlabeled data to improve text classification.* Doctoral dissertation, School of Computer Science, Carnegie Mellon University.

Rasmussen, E. (1992). Clustering algorithms. In W. Frakes and R. Baeza-Yates (Eds.), *Information retrieval: Data structures and algorithms*, 419–442. Prentice Hall.

Rose, K. (1998). Deterministic annealing for clustering, compression, classification, regression, and related optimization problems. *Proceedings of IEEE*, *86*, 2210–2239.

Scholkopf, B., & Smola, A. (2001). *Learning with kernels.* MIT Press.

Sinkkonen, J., & Kaski, S. (2001). Clustering based on conditional distributions in an auxiliary space. *Neural Computation*, *14*, 217–239.

Steinbach, M., Karypis, G., & Kumar, V. (2000). A comparison of document clustering techniques. *KDD Workshop on Text Mining.*

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

Strehl, A., Ghosh, J., & Mooney, R. J. (2000). Impact of similarity measures on web-page clustering. *AAAI Workshop on AI for Web Search* (pp. 58–64).

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

Vaithyanathan, S., & Dom, B. (2000). Model-based hierarchical clustering. *Proc. 16th Conf. Uncertainty in Artificial Intelligence* (pp. 599–608).

Vapnik, V. (1998). *Statistical learning theory.* New York: John Wiley.

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

Zhong, S., & Ghosh, J. (2002). *A unified framework for model-based clustering and its applications to clustering time sequences* (Technical Report). Department of ECE, University of Texas at Austin.

Zhong, S., & Ghosh, J. (2003). Scalable, balanced model-based clustering. *The 3rd SIAM Int. Conf. Data Mining.* To appear.