DOI 10.1007/s10115-004-0194-1

Knowledge and Information Systems (2005) 8: 374–384

Knowledge and Information Systems

Short Paper

Generative model-based document clustering: a comparative study

Shi Zhong[1], Joydeep Ghosh[2]

[1]Department of Computer Science and Engineering, Florida Atlantic University, Boca Raton, FL, USA
[2]Department of Electrical and Computer Engineering, University of Texas at Austin, Austin, TX, USA

Abstract. This paper presents a detailed empirical study of 12 generative approaches to text clustering, obtained by applying four types of document-to-cluster assignment strategies (hard, stochastic, soft and deterministic annealing (DA) based assignments) to each of three base models, namely mixtures of multivariate Bernoulli, multinomial, 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 coclustering 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 coclustering algorithm fares worse than vMF-based methods for a majority of the datasets.

Keywords: Comparative study; Document clustering; Model-based clustering

1. Introduction

Document clustering has become a fundamental operation in unsupervised document organisation, automatic topic extraction, and fast information retrieval or filtering. Until the mid-1990s, hierarchical agglomerative clustering using a suitable similarity measure such as cosine, Dice, or Jaccard, formed the dominant paradigm for clustering documents. 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 (multilevel) self-organising map (Kohonen et al. 2000), spher-

Received 18 November 2003
Revised 28 October 2004
Accepted 25 November 2004
Published online 10 February 2005

ical k-means (Dhillon and Modha 2001), bisecting k-means (Steinbach et al. 2000), mixture of multinomials (Vaithyanathan and Dom 2000; Meila and Heckerman 2001), multilevel graph partitioning (Karypis 2002), mixture of vMFs (Banerjee et al. 2003), information bottleneck (IB) clustering (Slonim and Tishby 2000), and coclustering 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.

Among this large diversity of approaches, probabilistic model-based clusterings are particularly attractive, as each iteration is linear in the size of the input. We recently introduced a unified framework for such approaches (Zhong and Ghosh 2003), 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 reestimation step and a data reassignment 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 (vMF), in conjunction with four types of data assignments, thus leading to a total of 12 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 12 instantiated algorithms are compared on a number of document datasets derived from the Text REtrieval Conference (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) and a bipartite spectral coclustering method (Dhillon 2001). A comparison to recent Kullback-Leibler (KL) clustering or IB clustering is not made because of the equivalence between IB text clustering and multinomial model-based clustering (Banerjee et al. 2004).

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 (2004). They both focussed 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. This paper provides a substantially expanded empirical study in terms of both model coverage and dataset variety, yielding additional insights into the problem of large-scale text clustering.

2. Data reassignment strategies and base models

In this section, we briefly review the four data (re)-assignment strategies that are at the core of four related clustering algorithms—model-based k-means (*mk-means*), EM clustering, stochastic *mk*-means and deterministic annealing, respectively. We also summarise the three base generative models used to represent single clusters. A more detailed exposition of the ideas in this section can be found in Zhong and Ghosh (2003). 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 because it may represent individual words, stemmed words, tokenised words or short phrases.

Data assignment: Let $\Lambda = \{\lambda_1, ..., \lambda_K\}$ be the set of K models, one per cluster (typically they come from the same family, e.g. multinomials, and just differ in the parameter values). The *mk-means* algorithm assigns each data object x to only one cluster—the cluster y that gives the maximum likelihood, $P(x|\lambda_y)$. The *EM clustering* algorithm assigns a fraction of x to each cluster y, with the fraction calculated as $P(y|x, \Lambda) = \frac{P(y)P(x|\lambda_y)}{\sum_{y'} P(y')P(x|\lambda_{y'})}$. 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 $P(y|x, \Lambda)$. 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 (Zhong and Ghosh 2003) extends EM clustering by parameterising the E-step with a temperature parameter T, that is, $P(y|x, \Lambda) = \frac{P(y)P(x|\lambda_y)^{\frac{1}{T}}}{\sum_{y'} P(y')P(x|\lambda_{y'})^{\frac{1}{T}}}$, and letting T gradually decrease during the clustering process. 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 (Zhong and Ghosh 2003).

Let $y(x) = \arg\max_y P(x|\lambda_y)$. For text data, the condition $P(x|\lambda_{y(x)}) \gg P(x|\lambda_y)$, $\forall y \neq y(x)$ is often encountered for the models discussed next, 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 $P(y)$'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.

Base models: The multivariate Bernoulli and multinomial models (with naïve Bayes assumption) have been commonly used when the document-term matrix has binary- and integer-valued entries, respectively (McCallum and Nigam 1998; Zhong and Ghosh 2004). So we just describe the third model, which is based on the vMF distribution. This distribution is the analogue of the Gaussian distribution for directional data (Mardia 1975), and is given by

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

where x is a normalised (unit-length in L_2 norm) document vector and the Bessel function $Z(\kappa_y)$ is a normalisation term. The parameter κ measures the directional variance (or dispersion) and, the higher its value, the more peaked is the distribution. Hard assignment on a vMF mixture with the same κ yields the spherical k-means (Dhillon and Modha 2001; Banerjee and Ghosh 2004), which has performed well for several text collections. For vMF-based algorithms, we use TF-IDF (Term Frequency-Inverse Document Frequency)-weighted document vectors that are normalised to unit length.

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.

Detailed parameter estimation formulas are omitted for these three models, but can be found in McCallum and Nigam (1998), Zhong and Ghosh (2003) and Banerjee and Ghosh (2004).

3. Experimental results

3.1. Evaluation criteria

For document clustering, external measures are commonly used because typically the benchmark documents' category labels are actually known (but 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). Based on the arguments in Strehl et al. (2000) and in Strehl and Ghosh (2002), in our experiments, we use normalised mutual information (NMI) between cluster and class random variables as the evaluation criterion. Because the three probabilistic models use slightly different representations of documents, we cannot directly compare their objective functions (data likelihoods) under different probabilistic models.

3.2. Text datasets

We used the 20-newsgroups data[1] and a number of datasets from the CLUTO toolkit[2] (Karypis 2002). These datasets provide a good representation of different characteristics: number of documents ranges from 204 to 19,949, number of words from 5,832 to 43,586, and number of classes from 3 to 20. 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. The *NG17-19* dataset is a subset of NG20, containing $\sim$ 1,000 messages from each of the three categories on different aspects of politics. These three categories are expected to be difficult to separate. All the datasets associated with the CLUTO toolkit have already been preprocessed (Zhao and Karypis 2004) and we further removed those words that appear in two or fewer documents.

3.3. Experimental setting

The four algorithms based on the Bernoulli model are *k*-Bernoullis, stochastic *k*-Bernoullis, mixture-of-Bernoullis and Bernoulli-based deterministic annealing (DA),

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

[2] 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 words, K the number of classes, $\bar{n}_c$ the average number of documents per class, and balance the size ratio of the smallest class to the largest class)

Data	Source	n_d	n_w	K	$\bar{n}_c$	Balance
NG20	20 newsgroups	19,949	43,586	20	997	0.991
NG17-19	3 overlapping groups from NG20	2,998	15,810	3	999	0.998
classic	CACM/CISI/Cranfield/Medline	7,094	41,681	4	1,774	0.323
ohscal	OHSUMED-233,445	11,162	11,465	10	1,116	0.437
k1b	WebACE	2,340	21,839	6	390	0.043
hitech	San Jose Mercury (TREC)	2,301	10,080	6	384	0.192
reviews	San Jose Mercury (TREC)	4,069	18,483	5	814	0.098
sports	San Jose Mercury (TREC)	8,580	14,870	7	1,226	0.036
la1	LA Times (TREC)	3,204	31,472	6	534	0.290
la12	LA Times (TREC)	6,279	31,472	6	1,047	0.282
la2	LA Times (TREC)	3,075	31,472	6	513	0.274
tr11	TREC	414	6,429	9	46	0.046
tr23	TREC	204	5,832	6	34	0.066
tr41	TREC	878	7,454	10	88	0.037
tr45	TREC	690	8,261	10	69	0.088

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 Sect. 2, 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, the 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 the *daberns* and *damnls* algorithms, an inverse temperature parameter $\beta = 1/T$ is used to parameterise 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 10 times, each time starting from a different random initialisation. 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, 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 algorithm. We use *vcluster* in the toolkit with the default setting, which is a bisecting graph partitioning-based algorithm. The other one is the bipartite spectral coclustering algorithm (Dhillon 2001), modified according to Ng et al. (2002)[3] so as to

[3] Use K instead of $\log K$ eigen directions and normalise each projected data vector.

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**
cocluster	.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**
cocluster	.40 ± .07	.56 ± .02	.41 ± .05	.42 ± .07	.41 ± .02

generate slightly better results than the original algorithm. We run each algorithm 10 times, each run using a different order of documents.

3.4. 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* ± *1 standard deviation*. Boldface entries highlight the best algorithms in each column. To save space, we show the NMI results for one specific K only for each dataset (results for other datasets are shown in Tables 3 and 4).

Table 5 shows 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*

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 ± .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
daberns	.04 ± .00	.09 ± .00	.08 ± .01	.02 ± .00	.07 ± .00
kmnls	.55 ± .04	.39 ± .07	.15 ± .03	.49 ± .03	.43 ± .05
skmnls	.55 ± .05	.39 ± .08	.15 ± .02	.50 ± .04	.43 ± .05
mixmnls	.56 ± .04	.39 ± .07	.15 ± .03	.50 ± .03	.43 ± .05
damnls	.61 ± .04	.61 ± .02	.31 ± .03	.61 ± .05	.56 ± .03
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
cocluster	.60 ± .01	.53 ± .03	.22 ± .01	.51 ± .02	.50 ± .03

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**

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 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 range 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 indicates significant evidence for the reverse of the hypothesis to be true. All significant *p*-values are highlighted in boldface in the table.

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 it 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. For the vMF models, however, one should note that the exact EM clustering can achieve significant improvement over hard assignment due to an annealing effect observed in 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 of 15 datasets. 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 maximising data likelihood with Bernoulli models does not align with generating well-separated clusters.

Surprisingly, the bipartite spectral coclustering 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.

Table 6 shows the running time results on *NG20*, the largest dataset used in our experiments. All the numbers are recorded on a 2.4 GHz PC running Windows 2000 with 768 MB memory and reflect only the clustering time, not including the data I/O cost. Note that CLUTO is written in C, whereas all the other algorithms are in Matlab. Clearly, algorithms using soft assignment take longer time than those using hard assignments. Overall, the *kvmfs* algorithm is the fastest one.

3.5. 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 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 and word-variance-based selection (Salton and McGill 1983).

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,

Table 6. 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	18.6 ± 1.8	22.6 ± 1.7	25.1 ± 1.7	27.0 ± 1.7
cocluster	20.9 ± 0.5	39.9 ± 1.0	62.8 ± 0.7	102.9 ± 0.8

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 lth 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. The clustering results as well as paired t-test results with feature-selected text datasets are omitted due to space limit, but can be found in Zhong and Ghosh (2004).

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

1. In terms of NMI, the multinomial model-based methods generally produce better results for feature-selected datasets as compared with the basic vMF-based approaches. For example, 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. An explanation of this finding is obtained by reconsidering the objective functions for *kmnls* and *kvmfs*. Note that the former maximises $\sum_x \sum_l x(l) \log P_{y(x)}(l)$, 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 maximises $\sum_x \sum_l \tilde{x}(l)\mu_{y(x)}(l)$, where $\tilde{x} = \frac{x}{\|x\|}$ is a normalised document vector and $\mu_{y(x)}$ is the normalised 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 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. However, note that if κ is allowed to vary among clusters, then the vMF mixture model is still (often substantially) superior (Banerjee et al. 2003). A recent evaluation using PAC-MDL bounds instead of NMI also shows this superiority (Banerjee and Langford 2004).
2. 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. For variance-selected datasets, *damnls* is significantly better than *davmfs* on 13 datasets and worse on only two.
3. After feature selection, CLUTO seems to deliver lower NMI performance whereas cocluster seems to fare better on most datasets. For example, the performance of CLUTO significantly improves on 3 but degrades on 6 out of 15 variance-selected datasets. In contrast, the cocluster generates significantly improved results on 9 but degraded results on only 2 out of 15 variance-selected datasets.

4. Concluding remarks

The comparative study of generative models for document clustering provided several insights and some surprises. We also noted that, while the model-based algorithms (except DA) have a computational advantage over graph-partitioning based approaches, they need better initialisation strategies to generate more stable clustering results. Updating multiple solutions simultaneously based on different initialisations, using online updates of the means or performing local search around the found solutions are some viable approaches for further improving performance and robustness of model-based text clustering approaches.

Acknowledgements. 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

Banerjee A, Dhillon I, Ghosh J, Merugu S (2004) An information theoretic analysis of maximum likelihood mixture estimation for exponential families. In: Proc 21st int conf machine learning, pp 57–64. Banff, Canada, July 2004

Banerjee A, Dhillon IS, Ghosh J, Sra S (2003) Generative model-based clustering of directional data. In: Proc 9th ACM SIGKDD int conf knowledge discovery and data mining, pp 19–28, Washington, DC, August 2003

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

Banerjee A, Langford J (2004) An objective evaluation criterion for clustering. In: Proc 10th ACM SIGKDD int conf knowledge discovery and data mining, pp 702–719

Dhillon IS (2001) Co-clustering documents and words using bipartite spectral graph partitioning. In: Proc 7th ACM SIGKDD int conf knowledge discovery and data mining, pp 269–274

Dhillon IS, Modha DS (2001) Concept decompositions for large sparse text data using clustering. Mach Learn 42(1):143–175

Ghosh J (2003) Scalable clustering. In: Ye N (ed) Handbook of data mining, pp 341–364. Lawrence Erlbaum Assoc

Karypis G (2002) CLUTO—A clustering toolkit. Dept of Computer Science, University of Minnesota, http://www-users.cs.umn.edu/~karypis/cluto/

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

Mardia KV (1975) Statistics of directional data. J Roy Stat Soc Ser B 37(3):349–393

McCallum A, Nigam K (1998) A comparison of event models for naive Bayes text classification. In: AAAI workshop on learning for text categorization, pp 41–48

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

Meila M, Heckerman D (2001) An experimental comparison of model-based clustering methods. Mach Learn 42:9–29

Ng AY, Jordan MI, Weiss Y (2002) On spectral clustering: analysis and an algorithm. In: Dietterich TG, Becker S, Ghahramani Z (eds) Advances in neural information processing systems 14, pp 849–856. MIT

Salton G, McGill MJ (1983) Introduction to modern information retrieval. McGraw-Hill

Slonim N, Tishby N (2000) Document clustering using word clusters via the information bottleneck method. In: Research and development in information retrieval, pp 208–215

Steinbach M, Karypis G, Kumar V (2000) A comparison of document clustering techniques. In: KDD workshop on text mining, Boston, MA

Strehl A, Ghosh J (2002) Cluster ensembles—a knowledge reuse framework for combining partitions. J Mach Learn Res 3:583–617

Strehl A, Ghosh J, Mooney RJ (2000) Impact of similarity measures on web-page clustering. In: AAAI workshop on AI for web search, pp 58–64

Vaithyanathan S, Dom B (2000) Model-based hierarchical clustering. In: Proc 16th conf uncertainty in artificial intelligence, pp 599–608

Zhao Y, Karypis G (2004) Empirical and theoretical comparisons of selected criterion functions for document clustering. Mach Learn 55(3):311–331

Zhong S, Ghosh J (2003) A unified framework for model-based clustering. J Mach Learn Res 4:1001–1037, November 2003

Zhong S, Ghosh J (2004) Generative model-based document clustering: A comparative study. Technical report, Department of Computer Science and Engineering, Florida Atlantic University, June 2004. Available at http://www.cse.fau.edu/∼zhong/papers/textc-full.pdf

Correspondence and offprint requests to: Shi Zhong, Department of Computer Science and Engineering, Florida Atlantic University, Boca Raton, FL, USA