International Journal of Tomography & Statistics,
Winter 2008, Vol. 8, No. W08; Int. J. Tomogr. Stat.
ISSN 0972-9976 (Print); ISSN 0973-7294 (Online)

Model Based Document Classification and Clustering

A. Murua[1], **W. Stuetzle**[2], **J. Tantrum**[3], **and S. Sieberts**[2]

[1]Département de mathématiques et de statistique
Université de Montréal
CP 6128, succ. Centre-ville
Montréal, Québec H3C 3J7, Canada
murua@dms.umontreal.ca

[2]Department of Statistics
University of Washington, Seattle, USA

[3]Microsoft adCenter Labs, Redmond, USA

ABSTRACT

In this paper we develop a complete methodology for document classification and clustering. We start by investigating how the choice of document features influences the performance of a document classifier and then use our findings to develop a clustering method suitable for document collections. From our study of the effect of frequency transformation, term weighting and dimensionality reduction through principal components analysis on the performance of a simple K-nearest-neighbors classifier, we conclude that: (a) applying a logarithm or square-root transformation to the term frequencies reduces error rates; (b) term weighting of the transformed frequencies does not appear to help much; and (c) increasing the feature space dimension beyond 50 does not improve performance. We use these findings in the construction of a Gaussian Mixture Document Clustering (GMDC) algorithm. This algorithm models the data as a sample from a Gaussian mixture. The model is used to build clusters based on the likelihood of the data, and to classify documents according to Bayes rule. One main advantage of our approach is the ability to automatically select the number of clusters present in the document collection. Our experiments with the Topic Detection and Tracking Corpus demonstrates the ability of GMDC to choose a sensible number of clusters and to generate meaningful partitions of the data.

Keywords: clustering, classification, text mining, dimensionality reduction, Gaussian mixture.

2000 Mathematics Subject Classification: 62H30, 62P30, 68T50.

*This work has been supported by NSA grant 62-1942, NSF grant DMS-9803226 and NSERC grant 327689-06

1 Introduction

With the dramatic growth of digital document collections (news, journal articles, e-mail, manuals, etc.) comes an increasing need for tools facilitating various kinds of access such as query-based retrieval, browsing, or generating synopses.

The primary focus of this paper is topic detection, i.e. assigning the documents in a collection C to "topics". For example, the documents might be coming from a newswire agency, and topics might be "Carter in Bosnia" (referring to the negotiations to cease fi re in Bosnia in December of 1994), "DNA evidence in the OJ Simpson trial" (referring to the DNA blood analysis in the OJ Simpson murder trial), "Kobe earthquake" (referring to the severe earthquacke that took place in Kobe, Japan, on January 17, 1995), etc. There are two versions of the topic detection problem: we can require that each document be assigned to exactly one topic, in which case we are constructing a partition of the collection; or we can allow each document to be assigned to one or more topics. In this paper we focus on the case when the topics form a partition of the collection.

We also briefly touch on "topic tracking". In its simplest form the goal of topic tracking is to assign a new document to one of the topics detected in C, or to decide that it is about a new topic not represented in C.

Document detection and tracking is different from document retrieval. The goal of document retrieval is to fi nd the documents in a collection that best match some query. The query might be considered a very short document consisting of a few keywords, and the goal then is to fi nd the documents in the collection that are most similar to the query document.

In Statistics terminology, topic detection is a clustering problem: we want to partition C into groups such that documents in each group are similar to each other, and dissimilar from documents in other groups.

In its simplest form, topic tracking is a classifi cation problem. We have a collection C of documents, each labeled with a topic, and we want to assign a label to a new document. The unusual aspect of the problem is that our answer could be "none", in which case the document is taken to represent a new topic.

Clustering and classifi cation methods play a central role in the reduction of both the number of operations needed for document classifi cation, and the retrieval time. Also, they can be designed to make accurate decisions on whether or not a document represents a new topic.

In order to apply clustering and classifi cation methods, we fi rst map documents to vectors in some p-dimensional space. This is not strictly speaking necessary. Most clustering methods and some classifi cation methods (for example K-nearest-neighbor classifi cation) only require similarities or dissimilarities between documents. However, the distinction is not as important as it might seem at fi rst glance. Given a representation of documents as p-dimensional points we can always defi ne dissimilarity as interpoint distance. Given a dissimilarity matrix, on the other hand, we can use multi-dimensional scaling (Kruskal, 1964a; Kruskal, 1964b; Kruskal

and Wish, 1978; Shepard, 1980) to find points in p-dimensional Euclidean space such that the interpoint distances approximately or exactly match the dissimilarities.

The influence of the mapping from documents to feature vectors (or, equivalently, the influence of the similarity measure) on the performance of document retrieval methods has been extensively studied, and has been found to be considerable (Zobel and Moffat, 1998; Dumais, 1991; Berry, Dumais and O'Brien, 1995). In contrast, the influence on clustering and classification methods appears to have attracted little attention (see however (Schütze and Silverstein, 1997)).

It is clear that in order to successfully address the topic tracking and detection problem, one need not only design a good clustering method for documents, but also discover a map from documents to feature vectors that eases the task for the clustering method. Thus, in this paper we develop a complete methodology for document clustering. We start by investigating how the choice of document features influences the performance of a document classifier (Section 2); and then use our findings to develop a clustering method suitable for document collections (Section 3).

Our study of the effect of document feature selection, e.g. word frequency transformation and weighting, and document dimensionality reduction (Subsection 2.1), is based on the K-nearest-neighbor classifier. We choose to work with this classifier because of its simplicity and lack of assumptions on the distributional properties of the documents. We expect that a choice of features resulting in good performance of the K-nearest-neighbor classifier is also a good choice for clustering. Although this admittedly requires a leap of faith, our experiments on document clustering in Subsection 3.2 seem to confirm this belief. Our findings suggest that applying a square-root or logarithm transformation to word frequencies results in substantial gains for classification. The combination of any of these transformations with the so-called *inverse document frequency* or *entropy* weighting schemes also improve classification performance. Similar results have been found in the information retrieval literature (Zobel and Moffat, 1998; Dumais, 1991; Berry et al., 1995) but with somewhat less general experiment designs that ours (Section 2).

Our document clustering method borrows ideas from the model based clustering literature (Banfield and Raftery, 1993; Celeux and Govaert, 1995). It explicitly models the data as a sample from a Gaussian mixture. Each of the components in the mixture distribution is assumed to be a multivariate Gaussian distribution with uncorrelated components. This assumption fits the data well and greatly simplify the computations involved, including the estimation of the parameters. These are efficiently estimated through the Expectation-Maximization (EM) algorithm (Dempster, Laird and Rubin, 1977). We propose several ways of initializing the EM algorithm; these include efficient and accurate variations of the K-means algorithm (Ward, 1963), as well as the more popular agglomerative hierarchical clustering techniques.

The model is used to build clusters based on the likelihood of the data, and to classify documents according to Bayes rule (Section 3). We call this approach to document clustering *Gaussian Mixture Document Clustering* (GMDC). One main advantage of our approach is the

ability to automatically estimate the number of clusters (topics) present in the document collection via Bayes factors (Raftery, 1995).

Our experiments in Subsection 3.2, with the TDT Corpus (Allan, Carbonell, Doddington, Yamron and Yang, 1998), are extremely encouraging, demonstrating the ability of GMDC to choose a reasonable number of clusters as well as to generate meaningful partitions of the data.

Our ideas have been successfully applied to large collections of documents and in general to large data sets, through a simple procedure that combines "fractionation" (Cutting, Karger, Pedersen and Tukey, 1992) with Gaussian mixture document clustering. The study of this extension has been published elsewhere (Tantrum, Murua and Stuetzle, 2004; Tantrum, Murua and Stuetzle, 2002). The present work focuses on the foundations of our methodology. Similar ideas to deal with large datasets, but not necessarily documents, have been reported in the literature recently. But unlike our comprenhensive treatment of the problem for document collections, they focus on scalability of the original EM algorithm used to fi t Gaussian mixtures (Jin, Wong and Leung, 2005), or on feature extraction procedures for general data sets (Hsieh, Wang and Hsu, 2006).

This paper is organized as follows. Section 2 describes our study of the influence of document dimensionality reduction, term weighting and transformation (feature selection) on the performance of the K-nearest-neighbors document classifi er. Within this section, we summarize a number of term frequency transformation and term weighting schemes used in the conversion of documents into high dimensional vectors. In Subsection 2.2 we focus on dimensionality reduction methods like latent semantic indexing and principal component analysis, and discuss connections between them. In Subsection 2.3 we describe our experiment on the effect of feature selection on the performance of the K-nearest-neighbors classifi er, present the results and our conclusions concerning feature selection. In Section 3, we describe our model based approach to document classifi cation and clustering. Our experiments regarding GMDC are presented in Subsection 3.2. Finally, Section 4 contains an overall summary of our fi ndings.

2 The effect of feature selection on document classification

In this section, we report the results of an experiment investigating how the choice of document features influences the performance of a document classifi er. We chose the K-nearest-neighbor classifi er due to its simplicity and lack of assumptions on the distributional properties of the documents. As our test data we use a subset of 1131 documents from the TDT corpus (Allan et al., 1998) that have been manually partitioned into 25 topics. Our fi ndings in this section are used later in Section 3 to select feature vectors that ease the classifi cation and clustering of documents. We start with a brief overview of widely used pre-processing techniques on documents.

2.1 Converting documents into vectors

The documents in a collection are first decomposed into word or sub-word units usually referred to as *terms,*. The terms are arbitrarily assigned sequence numbers between 1 and the number of terms p. Each document in the collection is then represented by a p-dimensional vector of term frequencies, and the collection of n documents is represented by a $n \times p$ term-frequency matrix $F = \{f_{ij}\}$. In applications such as topic detection and tracking, (on-line) customer support, product catalog navigation, and web-surfing, the vocabulary size typically is on the order of tens of thousands of terms, giving rise to extremely sparse term frequency matrices.

The raw term-frequency matrix F is then subjected to various transformations. The first step often is to replace the term frequencies by their square-root or logarithm. This reduces the influence of high counts, which is motivated by the belief that the difference between a term occuring 10 times versus 11 times is not as significant as the difference between a term occuring once versus not occuring at all. A more extreme step in the same direction is to convert F to a binary matrix indicating whether a term does or does not occur in a document.

A document is usually characterized by a few key terms; these terms indicate what topic the document is covering, and do not necessarily appear more than once within the document. Hence total term frequency is not necessarily indicative of a term's information content; for example a rare term (e.g. "OJ", from"OJ Simpson") immediately reveals what topic the document is about. To account for this disparity between terms, several global weighting schemes have been proposed. They are global in the sense that the weights reflect the distribution of terms over the entire document collection. Some proposed choices for the weight assigned to the j-th term are:

Identity: $w_j = 1$

Normal: $w_j = 1/\sqrt{\sum_i f_{ij}^2}$

Global frequency Inverse document frequency (Gfldf):
$w_j = \sum_i f_{ij} / \sum_i I(f_{ij} > 0)$, where I denotes the indicator function.

Inverse document frequency: $w_j = \log(n / \sum_i I(f_{ij} > 0))$

Entropy: $w_j = 1 + \sum_i p_{ij} \log p_{ij} / \log n$, with $p_{ij} = f_{ij} / \sum_i f_{ij}$.

The normal weighting scheme normalizes the term counts over the document collection. Hence a term which occurs infrequently will make the same contribution to the distance between documents as a very common term. The global frequency inverse document frequency (Gfldf) weighting scheme weights each term by the average frequency of the term in documents containing the term. Among two terms with equal total frequency $\sum_i f_{ij}$, Gfldf favors the one that occurs in a smaller number of documents. The inverse document frequency (Idf) weighting scheme gives lower weights to terms occurring in a large number of documents. The entropy weighting scheme is based on information-theoretic ideas. Basically, the entropy of a frequency distribution is maximized if all the frequencies are the same. This case is thought of as least informative: if a given term is equally likely to be present in all documents, then this term is not

telling anything about a particular document. A frequency distribution that is concentrated at a single document is at the other extreme (entropy $= 0$): the term completely distinguishes a particular document from the others.

Previous studies by Dumais (Dumais, 1991) suggest that entropy weighting outperforms other weighting schemes in the context of information retrieval; however her study focuses only on untransformed frequencies and log-transformed entropy weighted frequencies. In our experiment we evaluate all 15 combinations of the three term frequency transformations (i.e. untransformed, square root and logarithm) with the five weighting schemes listed above. We first transform the term frequencies, multiply the transformed frequencies by their global weights, and then normalize each document vector to have Euclidean norm equal to one. The last step eliminates the influence of document length on distance. More precisely, let $(f_{i1}, f_{i2}, \ldots, f_{ip})$ be the term frequency vector for the i-th document in the collection. The transformed and weighted term frequencies are given by

$$x_{ij} = \frac{w_j \times g(f_{ij})}{\sqrt{\sum_k (w_k \times g(f_{ik}))^2}},$$

where w_j is the global weight associated with the j-th term, and $g(\cdot)$ is the term transformation (square-root, log, identity). In the following, X denotes the transformed and weighted term frequency matrix.

2.2 Dimensionality reduction

The number p of terms occuring in a document collection can easily be in the thousands and may be larger than the number n of documents. Representing each document by a p-dimensional vector of (tranformed and weighted) term frequencies has at least two disadvantages. First, it is costly. Storing a document vector requires space proportional to the number of terms occuring in the document. Finding the distance between two document vectors requires work that is proportional to the number of terms occuring in the two documents. This assumes that sparse matrix techniques are used. Representing documents by vectors of dimensionality lower than the average number of terms in a document results in savings of space as well as time. Principal component analysis is a standard statistical tool for mapping a collection of high-dimensional vectors into some lower dimensional space while (hopefully) preserving the essential structure.

Second, representing documents by high dimensional term frequency vectors might even be detrimental to performance. This was first noted in the context of document retrieval and led to the discovery of latent semantic indexing. Latent semantic indexing was conceived with the goal of obtaining a measure of similarity between documents that is more invariant to "semantic content" than lexical matching (Berry et al., 1995). Lexical matching between words has been observed to be quite ineffective in information retrieval, since only documents having at least one word in common with the query are retrieved. In fact, lexical matching yields low *recall* (many relevant documents are missed) and low *precision*(many unrelated documents are retrieved). According to (Berry et al., 1995), the philosophy behind LSI is that "there is

an underlying or latent structure in the pattern of word usage that is partially obscured by the variability of word choice." Latent semantic indexing is designed to uncover this structure.

There is a close connection between principal component analysis and latent semantic indexing, which we will now discuss.

Principal component analysis Given document vectors $\mathbf{x}_1, \ldots, \mathbf{x}_n \in R^p$ and a target dimensionality q, principal component analysis finds the q-dimensional affine subspace S of $\mathbb{R}$ that is closest to the document vectors, i.e. that minimizes $\sum \mathbf{d}^2(\mathbf{x}_i, S)$, where $\mathbf{d}(\cdot, \cdot)$ denotes euclidean distance. It is a standard result ((Mardia, Kent and Bibby, 1979), Chapter 8) that S passes through the mean of the document vectors and is spanned by the q eigenvectors of the term covariance matrix Σ with the largest eigenvalues. The term covariance matrix is defined as

$$\begin{aligned} \Sigma &= 1/n\, \tilde{X}^t \tilde{X}, \qquad \text{where} \\ \tilde{X} &= \tilde{X} = (I - 1/n\, \mathbf{1}\mathbf{1}^t)\, X \end{aligned}$$

is obtained from X by mean centering the columns. Here $\mathbf{1} = (1, \ldots, 1)$. Let

$$\Sigma = A\, \Lambda\, A^t$$

be the eigen-decomposition of Σ. The columns of A are the normalized eigenvectors of Σ, and $\Lambda = \mathbf{diag}(\lambda_1, \ldots, \lambda_p)$ is the diagonal matrix of eigenvalues, in decreasing order. The projection $\mathbf{y}$ of a document vector $\mathbf{x}$ on the space spanned by the first q eigenvectors of Σ is given by $\mathbf{y} = A_q^t \mathbf{x}$, where A_q denotes the $p \times q$ matrix consisting of the q leading columns of A.

Dimensionality reduction by principal component analysis has another interesting property: it preserves distances between documents to the largest extent possible ((Mardia et al., 1979), Chapter 14.4). For a set of feature vectors $\mathbf{z}_1, \ldots, \mathbf{z}_n \in R^q$, define

$$E(\mathbf{z}_1, \ldots, \mathbf{z}_n) = \sum_{ij} (\mathbf{d}^2(\mathbf{x}_i, \mathbf{x}_j) - \mathbf{d}^2(\mathbf{z}_i, \mathbf{z}_j))^2$$

The figure of merit $E(\mathbf{z}_1, \ldots, \mathbf{z}_n)$ measures how well the interpoint distances of the q-dimensional feature vectors $\mathbf{z}_1, \ldots, \mathbf{z}_n$ match those of the p-dimensional document vectors $\mathbf{x}_1, \ldots, \mathbf{x}_n$. It is optimized by choosing $\mathbf{z}_i = A_q^t \mathbf{x}_i$, i.e. by projecting the document vectors on the space spanned by the q largest principal components. As the "structure" of the document cloud is captured by its interpoint distance matrix it is justified to say that, in a sense, dimensionality reduction by principal component analysis preserves structure to the largest extent possible.

There is an alternative algorithm for principal component analysis that brings out the similarity to latent semantic indexing: Find the singular value decomposition $\tilde{X} = \tilde{U}\, \tilde{\Phi}\, \tilde{V}^t$ of $\tilde{X}$. Here $\tilde{U}$ is $n \times n$ orthogonal, $\tilde{V}$ is $p \times p$ orthogonal, and $\tilde{\Phi}$ is the diagonal matrix of singular values, in decreasing order. We have

$$\tilde{X}^t \tilde{X} = \tilde{V} \tilde{\Phi}^2 \tilde{V}^t$$

which shows that $\tilde{V} = A$ and $\tilde{\Phi}^2 = \Lambda$ (up to sign changes).

Latent semantic indexing Latent semantic indexing is a dimensionality reduction tool very similar to principal component analysis. In contrast to principal component analysis, it finds the singular value decomposition $X = U\Phi V^t$ of X — the columns of X are not mean-centered. Dimensionality reduction is achieved as in principal component analysis: $\mathbf{y}_i = V_q^t \mathbf{x}_i$, where V_q is the matrix consisting of the leading q columns of V. *Latent semantic indexing reduces dimensionality by projecting the document vectors onto the closest q-dimensional linear subspace, whereas principal component analysis projects them onto the closest affine subspace.*

Computational considerations At first glance it might seem that principal component analysis requires the eigen-decomposition of the $p \times p$ matrix $\tilde{X}^T \tilde{X}$. Often this is infeasible because the number of terms in the document collection is too large. If the number n of documents is small compared to p it is more efficient to instead compute the eigen-decomposition of the $n \times n$ matrix $\tilde{X}\tilde{X}^t$. In fact, it is straightforward to verify that if UDU^t is the eigen-decomposition of $\tilde{X}\tilde{X}^t$, then the columns of $\tilde{X}^t U$ are the eigenvectors of $\tilde{X}^t\tilde{X}$ associated with the largest n eigenvalues (since the non-zero eigenvalues of $\tilde{X}^t\tilde{X}$ coincide with those of $\tilde{X}\tilde{X}^t$). When both n and p are very large, we propose to estimate $\tilde{X}\tilde{X}^t$ from a large but manageable sample of documents.

Compared to principal component analysis, latent semantic indexing has the advantage that it uses the singular value decomposition of X, which is sparse, while $\tilde{X}$ is not. There are efficient algorithms for computing the singular value decomposition of a sparse matrix and the projections of the document vectors (Berry, Drmac and Jessup, 1999; Berry et al., 1995).

2.3 Experiments on document dimensionality reduction

The goal of this experiment was to assess the influence of three factors — frequency transformation, term weighting, and dimensionality q of the feature space — on the ability to predict the topic of a document from its feature vector.

We tried all 15 frequency transformation / term weighting combinations described in Section 2.1. We used principal component analysis for dimensionality reduction, with a range of dimensions between 5 and 500.

The Data. The data used in the experiment were the 1131 labeled documents in the TDT corpus. The corpus consists of 15,863 news stories (documents) taken from Reuters and CNN between July 1, 1994, and June 30, 1995 (Allan et al., 1998). The TDT project investigators classified 1131 of these documents into 25 topics (e.g. Carter in Bosnia, Comet into Jupiter, DNA in the OJ trial, Kobe Japan quake, etc.). The number of documents on a given topic ranges from 2 to 273, and most topics appear in between 10 and 60 documents. We carried out a visual exploration of the data using the data exploration tool XGobi (Swayne, Cook and Buja, 1998). Some relevant findings concerning the shape of the clusters comprising the labeled TDT data are pointed out in Subsection 3.2.

The desing of the experiment. We used a K-nearest-neighbor classifi er for document classifi cation. To classify a document with feature vector **x**, a K-nearest-neighbor classifi er fi nds its k nearest neighbors among the training observations and takes majority vote. The number k of neighbors is a parameter of the procedure. We used cross-validation to estimate the optimal k from the training sample. Here is a detailed description of the experiment:

- Randomly partition the $n = 1131$ labeled documents into fi ve groups $\mathcal{D}_1, \ldots, \mathcal{D}_5$ of roughly equal size.
- Choose a combination of the experimental factors (frequency transformation, term weighting, and dimensionality).
- For $i = 1, \ldots, 5$, use group $\mathcal{D}_i$ as the test set and the union $\mathcal{D}_{-i}$ of the remaining groups as the training set. Estimate the optimal k from $\mathcal{D}_{-i}$ by cross-validation. Classify the documents in $\mathcal{D}_i$ using the optimal k. Let E_i denote the number of errors. Measure the merit of the current combination of experimental factors by the error rate $E = \sum_{i=1}^{5} E_i/n$.

Figures 1 and 2, and Table 1 show the results of the experiment. In both fi gures the error rate E is plotted on the vertical axis, and the dimension of the feature space is plotted on the horizontal axis. Each fi gure contains 15 curves, one for each of the 15 frequency transformation/term weighting combinations. The grey band is a ± 2 standard error band for log-ldf.

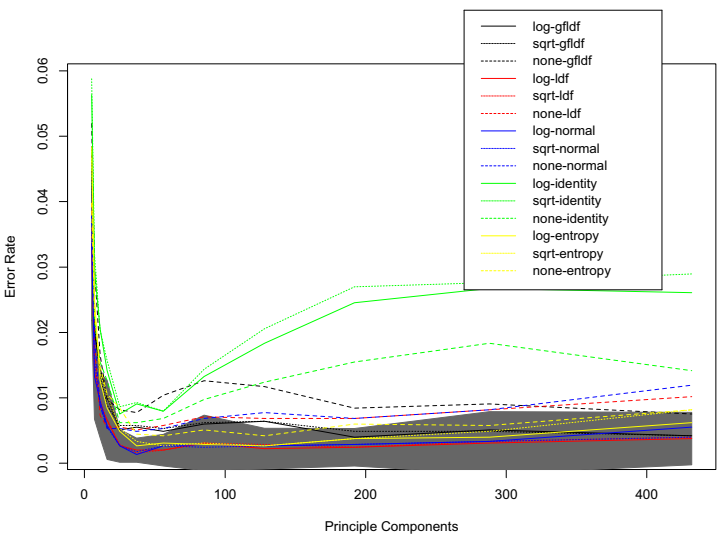

Figure 1: Performance of the 15 transformation/weighting scheme combinations as a function of the number of components utilized for data reduction.

We conclude that:

- Applying a square-root or log transformation to the term frequencies results in substantial

transformation weight combination	Number of Principal Components					
	5	7	11	16	25	37
log-gfdf	47.5 (6.86)	19.9 (4.49)	14.6 (4.71)	10.2 (4.01)	5.1 (2.99)	5.5 (2.34)
sqrt-gfdf	48.4 (6.87)	21.2 (4.52)	13.3 (2.34)	8.8 (3.66)	5.7 (2.40)	5.7 (2.13)
none-gfdf	51.9 (5.12)	29.8 (7.70)	16.6 (6.44)	9.7 (2.86)	8.2 (2.42)	7.7 (2.59)
log-idf	33.4 (5.76)	14.1 (3.70)	8.8 (2.47)	6.6 (3.03)	2.7 (1.26)	2.0 (0.92)
sqrt-idf	36.2 (7.26)	15.9 (2.54)	9.7 (2.13)	6.4 (2.75)	2.9 (0.99)	1.5 (0.99)
none-idf	39.8 (10.60)	18.1 (4.03)	7.3 (4.03)	5.5 (2.82)	5.5 (0.78)	5.1 (2.15)
log-entropy	33.6 (7.14)	13.3 (3.22)	9.5 (2.01)	5.7 (1.64)	2.7 (0.61)	1.3 (1.21)
sqrt-entropy	37.6 (9.21)	15.7 (3.61)	8.8 (3.03)	5.5 (1.35)	2.7 (0.61)	1.8 (0.61)
none-entropy	37.8 (11.23)	19.2 (4.85)	8.8 (4.88)	5.3 (2.64)	5.3 (1.98)	4.9 (2.01)
log-normal	56.4 (3.41)	29.6 (4.09)	20.6 (3.18)	14.1 (4.01)	7.5 (2.26)	9.1 (2.13)
sqrt-normal	58.8 (6.22)	32.0 (4.42)	20.3 (3.09)	15.7 (3.16)	8.6 (2.75)	9.3 (1.26)
none-normal	47.7 (6.96)	21.9 (4.97)	15.5 (2.47)	12.6 (1.48)	6.2 (1.68)	6.2 (2.29)
log-identity	48.4 (4.97)	25.2 (3.70)	12.6 (3.88)	8.4 (1.26)	4.9 (2.29)	2.7 (0.99)
sqrt-identity	46.6 (4.09)	24.3 (3.98)	13.7 (2.99)	10.4 (2.29)	5.3 (1.98)	3.3 (0.78)
none-identity	39.8 (5.79)	23.0 (6.32)	14.8 (5.16)	10.6 (2.99)	8.4 (1.48)	4.4 (2.21)
	56	85	128	192	288	432
log-gfdf	4.9 (2.29)	6.0 (1.68)	6.4 (2.75)	4.0 (1.68)	5.1 (2.88)	4.2 (1.64)
sqrt-gfdf	5.3 (2.40)	6.2 (2.15)	6.4 (3.07)	4.9 (1.85)	4.9 (3.09)	4.2 (3.07)
none-gfdf	10.4 (1.85)	12.6 (4.18)	11.7 (3.80)	8.4 (3.18)	9.1 (3.70)	7.5 (5.03)
log-idf	2.0 (1.21)	3.1 (2.13)	2.2 (1.56)	2.4 (1.44)	3.1 (2.40)	3.8 (2.01)
sqrt-idf	2.0 (0.49)	3.1 (2.75)	2.9 (2.01)	2.4 (1.44)	3.3 (1.56)	4.0 (1.85)
none-idf	5.7 (1.21)	7.1 (2.54)	6.9 (1.82)	6.9 (2.13)	8.2 (3.09)	10.2 (2.64)
log-entropy	2.7 (0.99)	2.9 (1.68)	2.7 (1.85)	2.9 (2.01)	3.3 (2.21)	5.5 (1.10)
sqrt-entropy	2.7 (0.99)	2.4 (1.82)	2.7 (2.29)	2.9 (1.26)	3.3 (2.21)	4.0 (2.42)
none-entropy	5.3 (2.13)	6.9 (2.64)	7.7 (4.13)	6.9 (1.98)	8.2 (3.80)	11.9 (3.07)
log-normal	8.0 (2.75)	13.3 (3.91)	18.3 (7.23)	24.5 (4.52)	26.7 (2.75)	26.1 (2.42)
sqrt-normal	8.0 (1.82)	14.4 (5.36)	20.6 (6.84)	27.0 (4.60)	27.6 (6.58)	29.0 (2.13)
none-normal	6.9 (2.40)	9.7 (2.40)	12.4 (3.78)	15.5 (2.47)	18.3 (4.53)	14.1 (3.53)
log-identity	3.1 (2.40)	2.9 (1.26)	2.7 (1.48)	3.8 (1.48)	4.0 (2.01)	6.2 (2.54)
sqrt-identity	2.7 (1.68)	3.1 (1.21)	2.7 (1.48)	3.8 (1.68)	4.9 (2.77)	8.2 (4.32)
none-identity	4.2 (2.13)	5.1 (1.85)	4.2 (2.13)	6.0 (2.29)	5.7 (2.86)	8.2 (3.55)

Table 1: Classifi cation error rates ($\times 1000$) as a function of both the frequency transformation/term weighting scheme and the number of principal components (dimensionality of the feature space). The fi gures in parenthesis correspond to the associated standard deviations of the classifi cation error rates; they were estimated from the fi ve error rate estimates arising in the cross-validated K-nearest-neighbors procedure. See text for further details.

performance improvement over using untransformed frequencies. The two transformations give virtually identical results in all cases.

- When combined with square-root or log transformation of term frequencies, entropy and Idf term weighting do slightly better than identity weighting. Gfldf and normal weighting do worse than identity weighting.
- Except for the case of normal weighting, going beyond a feature space dimension of 50 does not improve the error rate. Normal weighting seems to be the worst performer; the number of errors reaches a minimum roughly for dimensionality 30 and then increases with dimensionality.
- Misclassifi cation error rates for the best performing combinations of frequency transfor-

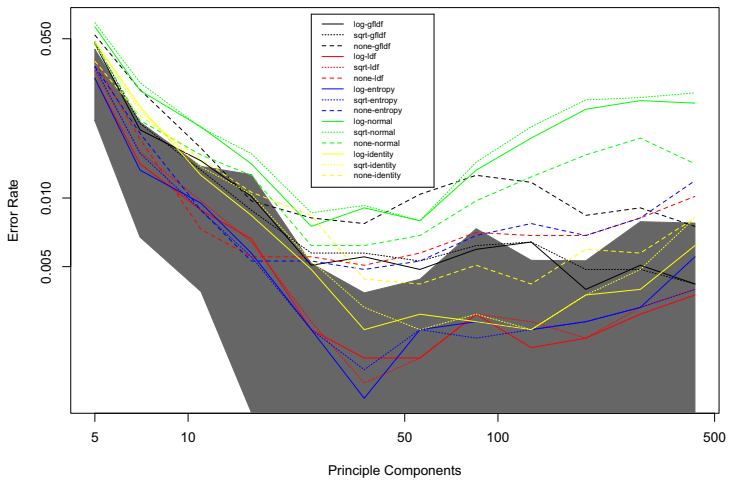

Figure 2: Performance of the 15 transformation / weighting scheme combinations as a function of the number of components utilized for data reduction (log-log scale).

mation / term weighting / feature space dimension are less than 1%; most of the errors correspond to misclassifi cation of documents representing very rare topics, e.g. Cuban riot in Panama (two documents), Karrigan/Harding (two documents), and Pentium chip flaw (four documents).

3 Gaussian Mixture Document Clustering

In this section we use model based clustering ideas (Banfi eld and Raftery, 1993; Celeux and Govaert, 1995) to cluster documents. The documents are assumed to be mapped to feature vectors of reduced dimension as explained in the previous section. This mapping involves term weighting, term transformation, and data reduction through principal components analysis.

Our clustering procedure explicitly models the data as being drawn from a Gaussian mixture. This mixture distribution is used to construct clusters based on the likelihood of the data and to classify documents according to Bayes rule. We call this approach to document clustering *Gaussian Mixture Document Clustering* (GMDC). One main advantage of our approach is the ability to automatically select the number of clusters present in the document collection via Bayes factors (Raftery, 1995).

The Gaussian mixture model.

The underlying assumption of GMDC is that the document feature vectors $x_1, x_2, \ldots, x_n$ are sampled from a Gaussian mixture density of the form

$$f(x) = \sum_{g=1}^{G} p_g \mathcal{N}_q(x; \mu_g, \Sigma_g), \tag{3.1}$$

where G is the number of components in the mixture, q is the dimension of the reduced document feature vector, $\{p_g\}_{g=1}^{G}$ are the mixture proportions, and $\mathcal{N}_q(x; \mu_g, \Sigma_g)$ denotes the q-variate normal density with mean μ_g, and variance-covariance matrix Σ_g, evaluated at x, $g = 1, \ldots, G$.

With high-dimensional data, as in our case with documents, the density given by (3.1) has many parameters. For example, The variance-covariance matrix of a 50-dimensional feature vector requires the estimation of 1275 covariances. Hence model complexity reduction is needed. Based on our visual exploration of the document feature vectors (as in Section 2) in multiple dimensions using the XGobi statistical visualization package (Swayne et al., 1998), it seems adequate (and reasonable) to model each component in the mixture density (3.1) as a multivariate Gaussian distribution with a diagonal variance-covariance matrix, i.e. with $\Sigma_g = \textbf{diag}\,(\sigma_{1g}^2, \ldots, \sigma_{qg}^2)$. Furthermore, the variance-covariance matrices cannot be reduced from diagonal matrices to multiples of the identity matrix (i.e. spherical Gaussian densities). This is also corroborated by the classifi cation results below (see experiments in Subsection 3.2). We note that when the variance-covariance matrices Σ_g are restricted to be diagonal matrices the number of parameters associated with a Gaussian mixture (3.1) is $r = G(1 + 2q)$, which increases only linearly with the dimension of the document feature vector, giving rise to very parsimonious models.

Selecting the Number of Clusters

One main advantage of GMDC over nonparametric clustering methods, is its ability to explicitly compute the likelihood of the model, and hence compare different models (i.e. different clusterings of the data) through Bayes factors (Raftery, 1995). Let $\mathcal{D}$ denote the data, and $\mathcal{M}_1, \mathcal{M}_2$ be two different mixtures models, e.g. models with different covariance structure, or different number of components. The Bayes factor for model $\mathcal{M}_2$ against model $\mathcal{M}_1$ is the ratio

$$P(\mathcal{D}|\mathcal{M}_2)/P(\mathcal{D}|\mathcal{M}_1);$$

it corresponds to the posterior odds for $\mathcal{M}_2$ against $\mathcal{M}_1$, assuming that *a priori* any of the two models is equally likely.

Since in document clustering we are interested in estimating the number of groups that give rise to the data, Bayes factors will be utilized to suggest the appropriate number of components in the mixture model. Our models are Gaussian mixture densities with similar covariance structure but with different number of components.

In (Raftery, 1995) Raftery shows that Bayes factors can be reasonably approximated by the Bayes information criterion, BIC, when the prior on the parameters is a multivariate normal distribution with mean equal to the MLE of the parameters, and variance-covariance matrix equal to the inverse of the Fisher information matrix given by the model. In this case choosing the most likely model as hinted by the Bayes factors, is equivalent to choosing the mixture model maximizing the BIC:

$$\text{BIC } = 2 \text{ log-likelihood } - r \log(n),$$

where $n = |\mathcal{D}|$ is the data size, and r, the number of parameters in the model, as defined above. Thus BIC will prefer those models that balance the likelihood of the model and the number of parameters in the model.

Assigning documents to clusters

Once a model is selected, each mixture component of the Gaussian mixture density gives rise to a cluster. The cluster structure or partition of the data is formed by assigning a cluster to each document. The usual cluster assignment corresponds to the Bayes classification rule: let d be a document and x be its associated feature vector (note: in what follows, we will refer to d and x as documents); then Bayes rule *assigns document d to cluster g, if*

$$g = \arg \max_{g'=1,\ldots,G} p_{g'} \mathcal{N}_q(x; \mu_{g'}, \Sigma_{g'}).$$

We will refer to the clustering structure arising from the mixture density model as *apparent partition.* The clusters in the apparent partition will be referred simply as *clusters.* In contrast, the true clusters generating the data will be referred to as *topics.*

Goodness-of-fit: Empirical measures

In order to objectively evaluate the performance of a clustering method we need labeled data. We can then compare the apparent partition generated by the clustering method with the true partition defined by the labels (topics). We will now describe two measures of similarity between partitions, the Fowlkes-Mallows-Wallace index and the F1 index.

The Fowlkes-Mallows-Wallace Index. The Fowlkes-Mallows-Wallace (Hubert and Arabie, 1985; Wallace, 1983; Fowlkes and Mallows, 1983) index is a measure of similarity between two partitions of the same data. Let $\{A_1, A_2, \ldots, A_g\}$ be the apparent partition (generated by the clustering method), and let $\{T_1, T_2, \ldots, T_J\}$ be the true partition (defined by topic labels). Each cell in Table 2 contains the number n_{ig} of documents in topic T_i assigned to cluster A_g.

The Fowlkes-Mallows-Wallace index associated to this table is

$$\sum_{i,g} \binom{n_{ig}}{2} \Big/ \sqrt{\sum_i \binom{n_{i\cdot}}{2} \sum_g \binom{n_{\cdot g}}{2}}.$$

This index is the geometric mean of the following two indexes:

$$\sum_{i,g}\binom{n_{ig}}{2}\Big/\sum_{i}\binom{n_{i\cdot}}{2}, \text{ and } \qquad \sum_{i,g}\binom{n_{ig}}{2}\Big/\sum_{g}\binom{n_{\cdot g}}{2}.$$

The first index can be interpreted as the probability that a randomly chosen pair of documents is assigned to the same cluster given that the pair addresses the same topic. Similarly, the second index can be interpreted as the probability that a randomly chosen pair of documents addresses the same topic given that the pair was assigned to the same cluster. Notice that apparent partitions with large Fowlkes-Mallows-Wallace index are the desirable ones.

The F1 measure: combining recall and precision. When considering retrieval of documents, a more natural measure of goodness-of-fit of a model can be created by combining recall and precision. Given a query document d, *recall* is the proportion of documents relevant to d retrieved from the collection, and *precision* is the proportion of documents relevant to d among those retrieved. A retrieval model, and in general, a clustering model, will be preferred if it generates good recall and precision. More specifically, associated to each document d there are a recall $r(d)$ and a precision $p(d)$. These are defined as follows. Suppose that document d in topic T_i is assigned to cluster A_g (see Table 2). When the query is d, all documents is cluster A_g are retrieved. So,

$$r(d) = n_{ig}/n_{i\cdot} \text{ and } \quad p(d) = n_{ig}/n_{\cdot g}$$

A reasonable measure of goodness-of-fit of a model is some sort of average that takes into account both the recall and precision of all documents in a given collection. Several ways of combining recall and precision in a single measure are possible. The most popular one in document retrieval applications is the so-called F_1 index (Allan et al., 1998; Van Rijsbergen, 1979), which is given by

$$F_1(d) = 2\frac{p(d)r(d)}{p(d)+r(d)} = \{\frac{1}{2}(\frac{1}{r(d)}+\frac{1}{p(d)})\}^{-1}$$

giving rise to the F_1 average

$$F_1 = \sum_{d\in \text{ data collection}} F_1(d)\times\frac{1}{n} = 2\sum_{i,g}\frac{n_{ig}^2}{n_{i\cdot}+n_{\cdot g}}\frac{1}{n}.$$

	Apparent Partition				
Topics	$\mathbf{A_1}$	$\mathbf{A_2}$	...	$\mathbf{A_G}$	**Total**
$\mathbf{T_1}$	n_{11}	n_{12}	$\cdots$	n_{1G}	$n_{1\cdot}$
$\mathbf{T_2}$	n_{21}	n_{22}	$\cdots$	n_{2G}	$n_{2\cdot}$
$\mathbf{\ldots}$	$\cdots$	$\cdots$	$\cdots$	$\cdots$	$\cdots$
$\mathbf{T_J}$	n_{J1}	n_{J2}	$\cdots$	n_{JG}	$n_{J\cdot}$
Total	$n_{\cdot 1}$	$n_{\cdot 2}$	$\cdots$	$n_{\cdot G}$	n

Table 2: Comparison between the apparent partition (columns) and the topics forming the true partition (rows). Each cell count n_{ig} corresponds to the number of common elements in cluster A_g and topic T_i.

In our experiments (Subsection 3.2), the Fowlkes-Mallows-Wallace and F_1 indexes give very similar results.

3.1 Estimating the mixture model

Clustering documents through GMDC requires estimating the parameters of the Gaussian mixture model (3.1). This estimation is usually done via the Expectation-Maximization (EM) algorithm (Dempster et al., 1977). EM gives explicit iterative updating formulas for the parameters associated to Gaussian mixtures for most constrained structures of variance-covariance matrices. We now develop these updating formulas for our particular case of interest, namely, when the variance-covariance matrices in the mixture are constrained to be diagonal matrices (Celeux and Govaert, 1995).

EM updating formulas. Let θ denote the parameters of the model, and $\mathcal{D} = \{x_i\}_{i=1}^n$ be the (observed) data (i.e. document feature vectors). Defi ne the variables

$$z_{ig} = \begin{cases} 1 & \text{if } g \text{ is the cluster containing data item } x_i, \\ 0 & \text{otherwise,} \end{cases}$$

$i = 1, \ldots, N$, $g = 1, \ldots, G$. Then the complete log-likelihood of the model given $\{(x_i, z_i)\}_1^n$ is

$$l(\theta|(x_1, z_1), \ldots, (x_n, z_n)) = \sum_{i=1}^{n} \sum_{g=1}^{G} z_{ig} \log(p_g \mathcal{N}_q(x_i; \mu_g, \Sigma_g)). \tag{3.2}$$

Now using the diagonal variance-covariance matrix constraint, one can write $\Sigma_g = \lambda_g D_g$, where D_g is a unitary diagonal matrix, i.e. the determinant of D_g is 1, and $\lambda_g = |\Sigma_g|^{1/q}$ (here and throughout this paper, $|A|$ stands for the determinant of the squared matrix A).

Let $n_g = \sum_{i=1}^n z_{ig}$, $g = 1, \ldots, G$. A straightforward computation shows that the EM updating formulas for the mixture proportions and means are

$$\hat{p}_g = \frac{n_g}{n}, \qquad \hat{\mu}_g = \frac{1}{n_g} \sum_{i=1}^{n} z_{ig} x_i.$$

Substituting these into (3.2), one sees that maximizing the log-likelihood of the model is equivalent to minimizing

$$\begin{aligned} \ell(\theta|(x_1, z_1), \ldots, (x_n, z_n)) &= \sum_{g=1}^{G} q n_g \log \lambda_g + \sum_{g=1}^{G} \sum_{i=1}^{n} \frac{1}{\lambda_g} z_{ig} (x_i - \hat{\mu}_g)^t D_g^{-1} (x_i - \hat{\mu}_g) \\ &= \sum_{g=1}^{G} \{q\, n_g \log \lambda_g + \frac{1}{\lambda_g} \textsf{trace}\,(W_g D_g^{-1})\}, \end{aligned} \tag{3.3}$$

where $W_g = \sum_{i=1}^n z_{ig} (x_i - \hat{\mu}_g)(x_i - \hat{\mu}_g)^t$, $g = 1, \ldots, G$. Another straightforward computation shows that the minimization of (3.3) leads to the updating estimates

$$\lambda_g = \frac{1}{n_g} |\textbf{diag}\,(W_g)|^{1/q}, \qquad D_g = \frac{\textbf{diag}\,(W_g)}{|\textbf{diag}\,(W_g)|^{1/q}}.$$

Initialization Step. The EM algorithm is a nonlinear optimization method and its results depend heavily on the initial guesses for the parameters. Several ways of initializing the parameters have been suggested in the literature. We have tried the following two methods:

(a) **Initializing via K-means a large mixture model.** In this case we form a model with a large number of clusters through slight modifi cations of the K-means algorithm (Ward, 1963), e.g. by considering Mahalanobis distances associated to diagonal variance-covariance matrices instead of the usual Euclidean distance; once the EM estimates for this large mixture are obtained, the number of clusters is iteratively reduced by one by merging two components from the mixture at a time, according to some criterion of optimality. For example, one could look for those two components whose merging least decreases the likelihood, or for the two closest components in the sense of the arc-cosine distance, as is done in the functional merging algorithm (Coates and Fitzgerald, 1999) (see below). Once the number of components has been reduced, new estimates are obtained through a new run of the EM algorithm. This process is iterated until only two components remain in the mixture

(b) **Initialization via an agglomerative hierarchical clustering method.** In this case the data is clustered by iteratively merging groups of data points according to some measure of similarity between the groups. This gives rise to a tree structure (similar to the one obtained through single linkage analysis) that describes the hierarchy of the clusters arising from coarse to fi ne partitions of the data. In agglomerative hierarchical clustering (Fraley, 1998; Posse, 1999) the tree is constructed in a bottom-up fashion, i.e. from the leaves up to root node; the leaves correspond to single data points (every single data point is seen as a singleton cluster, i.e. a cluster by itself), and the root to a cluster containing all the data.

As in the case (a) above, the criterion for merging two clusters is based on a measure of similarity between clusters; we have employed both likelihood based and functional merging methods for the merging of clusters. The results obtained with the TDT collection indicate that likelihood based merging performs better than functional merging for this task.

Next we briefly describe the likelihood based and the functional merging algorithms used in our experiments.

Likelihood based algorithm for merging. In GMDC the relevant part of the likelihood of a given cluster structure is given by equation (3.3). Hence, the merging of two clusters, say clusters g_1 and g_2, decreases the likelihood by the quantity

$$\begin{aligned}\Delta(g_1, g_2) &= q\,(n_{g_1} + n_{g_2}) \log \lambda_{\text{new}} + \frac{1}{\lambda_{\text{new}}} \text{trace}\,(W_{\text{new}} D_{\text{new}}^{-1}) \\ &\quad - \left(q\, n_{g_1} \log \lambda_{g_1} + q\, n_{g_2} \log \lambda_{g_2} + \frac{1}{\lambda_{g_1}} \text{trace}\,(W_{g_1} D_{g_1}^{-1}) + \frac{1}{\lambda_{g_2}} \text{trace}\,(W_{g_2} D_{g_2}^{-1}) \right) \\ &= q(n_{g_1} + n_{g_2}) \log \lambda_{\text{new}} - q n_{g_1} \log \lambda_{g_1} - q n_{g_2} \log \lambda_{g_2} \quad (3.4)\end{aligned}$$

where the updating formulas for the new cluster formed by the merging of clusters g_1 and g_2 are

$$\hat{\mu}_{\text{new}} = \frac{n_{g_1}}{n_{g_1}+n_{g_2}}\hat{\mu}_{g_1} + \frac{n_{g_2}}{n_{g_1}+n_{g_2}}\hat{\mu}_{g_2},$$
$$W_{\text{new}} = W_{g_1} + W_{g_2} + n_{g_1}(\hat{\mu}_{\text{new}} - \hat{\mu}_{g_1})(\hat{\mu}_{\text{new}} - \hat{\mu}_{g_1})^t + n_{g_2}(\hat{\mu}_{\text{new}} - \hat{\mu}_{g_2})(\hat{\mu}_{\text{new}} - \hat{\mu}_{g_2})^t,$$
$$\lambda_{\text{new}} = \frac{1}{n_{g_1}+n_{g_2}}|\mathbf{diag}\,(W_{\text{new}})|^{1/q}.$$

We note that (3.4) corresponds to the log-likelihood ratio between the hypotheses H_0 : clusters g_1 and g_2 are in the same cluster, and H_a : clusters g_1 and g_2 are distinct clusters. At each iteration of the agglomerative hierarchical clustering procedure the merging of the two current clusters that least decreases the likelihood (i.e. that minimizes $\Delta(g_1, g_2)$ over all possible pairs (g_1, g_2) is performed; the algorithm stops when there are only two clusters left in the model.

Functional merging algorithm. In functional merging (Coates and Fitzgerald, 1999), each current cluster is represented by its corresponding Gaussian density in the current mixture model. The procedure is based on the cosine between two Gaussian densities $f_{g_1}(x) = \mathcal{N}_q(x; \mu_{g_1}, \lambda_{g_1} D_{g_1})$, $f_{g_2}(x) = \mathcal{N}_q(x; \mu_{g_2}, \lambda_{g_2} D_{g_2})$, which corresponds to the cosine between f_{g_1} and f_{g_2} as vectors in L^2, i.e.

$$\cos(f_{g_1}, f_{g_2}) = \frac{\int f_{g_1} f_{g_2}}{\sqrt{\int f_{g_1}^2}\sqrt{\int f_{g_1}^2}}.$$

In our particular case, this quantity can be easily shown to be equal to

$$\frac{\lambda_{g_1}^{q/4}\lambda_{g_2}^{q/4}}{|\frac{1}{2}(\lambda_{g_1} D_{g_1} + \lambda_{g_2} D_{g_2})|^{1/2}} \exp\{-\frac{1}{2}(\mu_{g_1}^t \lambda_{g_1}^{-1} D_{g_1}^{-1} \mu_{g_1} + \mu_{g_2}^t \lambda_{g_2}^{-1} D_{g_2}^{-1} \mu_{g_2} - \mu_{\text{merge}}^t \lambda_{\text{merge}}^{-1} D_{\text{merge}}^{-1} \mu_{\text{merge}})\} \quad (3.5)$$

where

$$\lambda_{\text{merge}} = \lambda_{g_1}\lambda_{g_2}/|\lambda_{g_1} D_{g_1} + \lambda_{g_2} D_{g_2}|^{1/q}$$
$$D_{\text{merge}} = \{|\lambda_{g_1} D_{g_1} + \lambda_{g_2} D_{g_2}|^{1/q}(\lambda_{g_2} D_{g_1}^{-1} + \lambda_{g_1} D_{g_2}^{-1})\}^{-1}$$
$$\mu_{\text{merge}} = \lambda_{\text{merge}} D_{\text{merge}}(\lambda_{g_1}^{-1} D_{g_1}^{-1}\mu_{g_1} + \lambda_{g_2}^{-1} D_{g_2}^{-1}\mu_{g_2})$$

The two clusters with the largest cosine value between their corresponding densities are merged. In our experiments (see Subsection 3.2) functional merging performs rather poorly in comparison with the likelihood based hierarchical clustering.

3.2 Experiments with Gaussian mixture document clustering

We applied our GMDC techniques to the clustering of the 1131 labeled news event documents in the TDT corpus. The labels were manually assigned by people who read the news stories, giving rise to 25 topics in the data.

We followed the procedure in Section 2 for dimensionality reduction, term weighting, and term frequency transformation of these data. According to our conclusions in Subsection 2.3 (see Figure 2), we chose to work with 50 principal components (i.e. $q = 50$), inverse document frequency weighting, and logarithm transformation of the resulting reduced and weighted "terms".

GMDC performance

Figure 3 shows the BIC scores for seven "flavors" of model based document clustering. The suffi x **EM** refers to the EM algorithm being applied after the initialization step in order to refi ne the mixtures; its absence implies that this step was not performed.

- **KM-fm-EM:** refers to GMDC with initialization through K-means followed by agglomerative hierarchical clustering based on functional merging.
- **KM-Lr-EM:** refers to GMDC with initialization through K-means followed by agglomerative hierarchical clustering based on the likelihood ratio criterion.
- **He-fm, He-fm-EM:** refers to GMDC where the initialization step consists on assigning each data item to a distinct singleton cluster; these were then iteratively merged by applying an agglomerative hierarchical clustering method based on functional merging. The Gaussian components in the mixtures were assumed to be ellipsoidal.
- **He-Lr, He-Lr-EM:** refers to GMDC where the initialization step consists on assigning each data item to a distinct singleton cluster; these were then iteratively merged by applying an agglomerative hierarchical clustering method based on the likelihood ratio criterion. The Gaussian components in the mixtures were assumed to be ellipsoidal.
- **Hs-Lr:** refers to GMDC where the initialization step consists on assigning each data item to a distinct singleton cluster; these were then iteratively merged by applying an agglomerative hierarchical clustering method based on the likelihood ratio criterion. The Gaussian components in the mixtures were assumed to be spherical.

In terms of computational cost, K-means initialization methods are cheaper than the usual agglomerative hierarchical initialization methods, i.e. bottom-up algorithms that start with each data item as a singleton cluster (from now on we will refer to these initialization methods as fully bottom-up hierarchical methods). In fact, if n is the data size, and k is the initial number of clusters, then K-means initialization based methods are of the order $O(nk)$, while fully bottom-up hierarchical methods are of order $O(n^2)$. However, fully bottom-up hierarchical methods produce consistently higher BIC scores, as well as consistently higher Fowlkes-Mallows-Wallace and F_1 indexes (see Figures 3–4).

One can see from Figure 3 that the best method is He-Lr-EM, closely followed by He-Lr. The two variations of the K-means initialization method perform similarly, and comparable to the best two performers when the number of clusters is close to the number of topics (i.e. 25). The Gaussian mixtures with spherical covariance structure closely follows the KM-fm-EM and KM-Lr-EM curves; however, in contrast to the latter two curves, it steadily increases with the

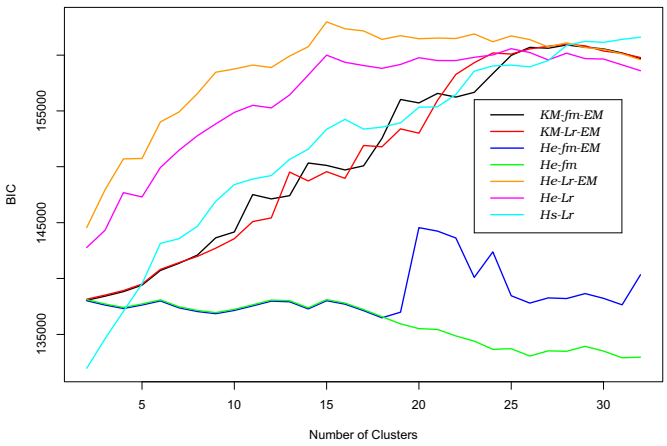

Figure 3: Values of BIC for seven different “flavors” of GMDC.

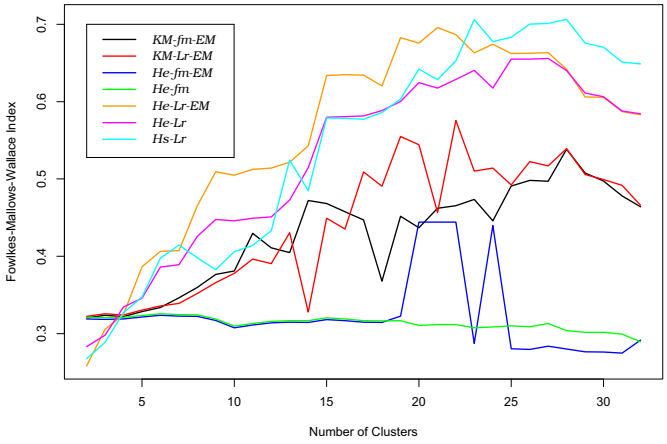

Figure 4: Fowlkes-Mallows-Wallace index associated to the seven “flavors” of GMDC as a function of the number of clusters in the models.

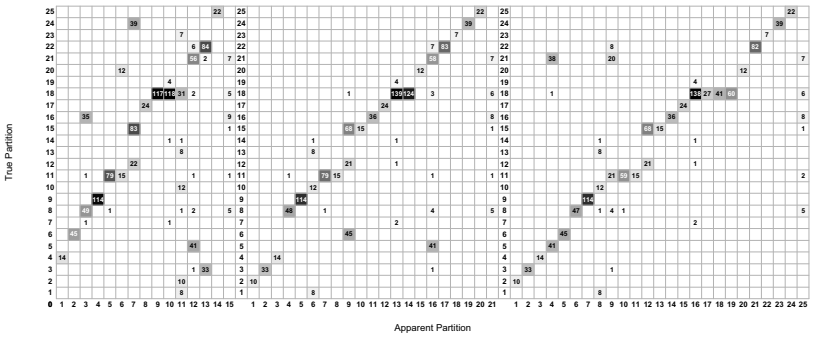

Figure 5: Comparison between three apparent partitions (columns) and the true partition (rows). The apparent partitions correspond to the three highest values of BIC for He-Lr-EM; these partitions contain 15, 21, and 25 clusters, respectively. The cells are color-coded according their absolute counts. Darker cells correspond to larger counts.

number of clusters. This phenomenon can be explained as follows: assume that the true nature of the covariance structure of the clusters is ellipsoidal; then mixture models with spherical covariance structure will tend to divide each ellipsoidal cluster in smaller more spherical-shaped clusters, thus producing a large number of sub-clusters within clusters. The worst performers are the two fully bottom-up hierarchical methods based on functional merging. It appears that functional merging is very sensitive to the initialization step; in contrast, judging from our experiments, the performance of likelihood based methods is more "stable", in the sense that the initialization step is not too critical.

In order to illustrate the clustering structure "discovered" by the best method, He-Lr-EM, Figures 5, 6, and 7, show the association between the *true partition* (rows) and the *apparent partitions* (columns), for the models with the largest BIC scores, i.e. models with 15, 21, and 25 clusters. In these fi gures, each cell number n_{ig} corresponds to the number of common documents in topic i and cluster g. The cells have been colored according to (a) their absolute counts in Figure 5, (b) their relative counts in the apparent partition (i.e. in each column) in Figure 6, and (c) their relative counts in the true partition (i.e. in each row) in Figure 7 (darker cells correspond to larger cells). Figures 6 and 7 give us an idea of the precision and recall, respectively, associated with the apparent partitions.

These fi gures show the evolution of the clusters as their number is increased. The fi rst partition is very coarse, containing only 15 clusters (recall that there are 25 topics). Several of the topics are grouped together in this partition. The second and third partitions contain 21 and 25 cluster, respectively. In these latter partitions, some topics are split into separate clusters. The third partition adds only a little to the second one, at the cost of splitting up some of the topics. This shows the classic trade-off between precision and recall in clustering: the *precision* is increased as the number of clusters is increased, but the *recall* is reduced. The Fowlkes-Mallows-Wallace index suggests that the optimal choice is the more parsimonious second

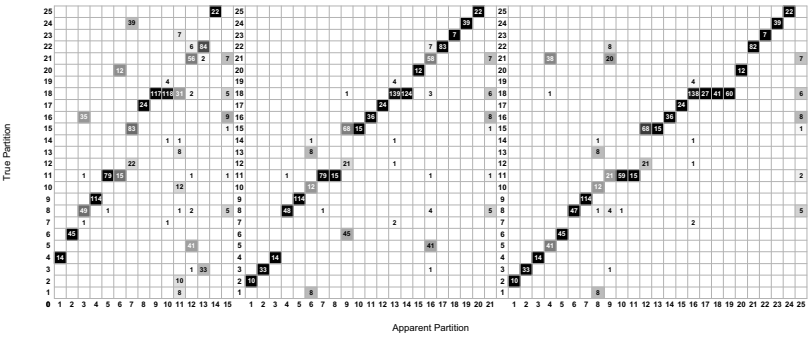

Figure 6: Precision associated with the three apparent partitions with the highest values of BIC for He-Lr-EM; these partitions contain 15, 21, and 25 clusters, respectively. The cells are color-coded according to their relative counts on the apparent partitions (columns). Darker cells correspond to larger counts.

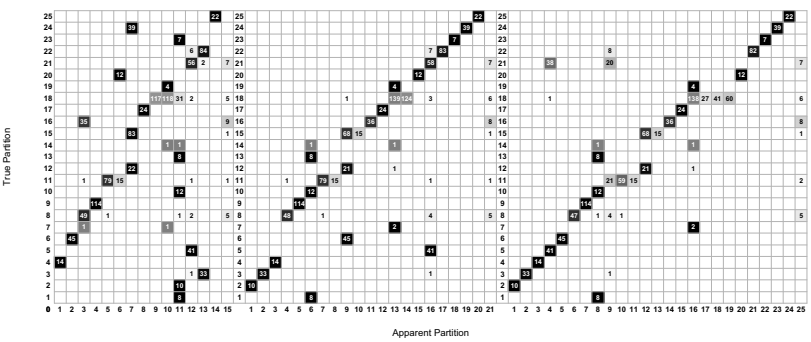

Figure 7: Recall associated with the three apparent partitions with the highest values of BIC for He-Lr-EM; these partitions contain 15, 21, and 25 clusters, respectively. The cells are color-coded according their relative counts on the true partition (rows). Darker cells correspond to larger counts.

partition (containing 21 clusters).

In general we observe that the apparent partitions found by GMDC agree well with the topics and that clusters may show patterns in the data that are not necessarily evident without reading all of the documents. Indeed a closer look at the cases were differences between the clusters and the topics are observed reveals that the documents from different topics that are merged into single clusters present large similarities in their wording. For example, cluster 9 in the second partition is representative of natural disasters: it contains almost all of topics 6 (*comet crashes into Jupiter*) and 12 (*Humble, Texas flooding*), as well as part of topic 15 (*Kobe, Japan earthquake*).

4 Summary

We present a complete methodology for document clustering and classifi cation. This is based on mapping documents to feature vectors in an Euclidean space followed by Gaussian mixture modeling of the distribution associated with the feature vectors.

Our study shows that (1) good classifi cation and clustering performance is achieved by using feature vectors derived through principal component analysis of log or square-root-transformed term frequencies; (2) increasing the feature space dimension beyond 50 principal components does not improve performance; (3) a model based on Gaussian mixture densities with diagonal covariance structure is suffi cient for clustering the TDT labeled corpus, and (4) the BIC criterion suggests a reasonable number of clusters.

Crucial to the estimation of the parameters associated to GMDC is the choice of the starting values of the EM algorithm. Our experiments show that hierarchical clustering initialization gives the best results. However, when the number of components is close to the number of topics, K-means initialization becomes a competitive alternative. The main advantage of K-means is its speed, which could be made linear in the number of documents if a binary split strategy is used on each iteration of the K-means algorithm (Rabiner and Juang, 1993, pp. 126–127).

Also essential in fi nding the initial clustering structure through an agglomerative hierarchical procedure is the cluster merging criterion. Our experiments clearly show the superiority of the likelihood ratio criterion over the functional merging criterion.

We have extended our methodology to large collections of documents such as the complete TDT corpus (Tantrum et al., 2004; Tantrum et al., 2002) via a divide-and-conquer strategy based on fractionation (Cutting et al., 1992). An alternative strategy based on a scalable EM algorithm is suggested by Jin, Wong and Leung in (Jin et al., 2005).

References

Allan, J., Carbonell, J., Doddington, G., Yamron, J. and Yang, Y. 1998. Topic detection and tracking pilot study fi nal report.

Banfi eld, J. D. and Raftery, A. 1993. Model-based Gaussian and non-Gaussian clustering, *Biometrics* **49**: 803–821.

Berry, M., Drmac, Z. and Jessup, E. 1999. Matrices, vector spaces, and information retrieval, *SIAM Review* **41**(2): 335–362.

Berry, M., Dumais, S. and O'Brien, G. 1995. Using linear algebra for intelligent information retrieval, *SIAM Review* **37**(4): 573–595.

Celeux, G. and Govaert, G. 1995. Gaussian parsimonious clustering models, *Pattern Recognition* **28**: 781–793.

Coates, M. J. and Fitzgerald, W. J. 1999. Regionally optimised time-frequency distributions using fi nite mixture models, *Signal Processing* **77**: 247–260.

Cutting, D. R., Karger, D. R., Pedersen, J. O. and Tukey, J. W. 1992. Scatter/gather: A cluster-based approach to browsing large document collections, *15th Ann. International SIGIR 92/Denmark-6/92*, p. 318329.

Dempster, A. P., Laird, N. M. and Rubin, D. B. 1977. Maximum likelihood from incomplete data via the EM algorithm, *J. R. Stat. Soc.* **76**: 341–353.

Dumais, S. 1991. Improving the retrieval of information from external sources, *Behavior Research Methods, Instruments & Computers* **23**(2): 229–236.

Fowlkes, E. B. and Mallows, C. L. 1983. A method for comparing two hierarchical clusterings, *J. American Statistical Association* **78**: 553–569.

Fraley, C. 1998. Algorithms for model-based Gaussian hierarchical clustering, *SIAM J. Sci. Comput.* **20**: 270–281.

Hsieh, P.-F., Wang, D.-S. and Hsu, C.-W. 2006. A linear feature extraction for multiclass classifi caion problems based on class mean and covariance discriminant information, *IEEE Trans.Pattern Analysis and Machine Intelligence* **28**: 223–235.

Hubert, L. and Arabie, P. 1985. Comparing partitions, *J. Classification* **2**: 193–218.

Jin, H., Wong, M.-L. and Leung, K.-S. 2005. Scalable model-based clustering for large datasets based on data summarization, *IEEE Trans.Pattern Analysis and Machine Intelligence* **27**: 1710–1719.

Kruskal, J. B. 1964a. Multidimensional scaling by optimizing goodness of fi t to a nonmetric hypothesis, *Psychometrika* **29**(1): 1–27.

Kruskal, J. B. 1964b. Non-metric multidimensional scaling: A numerical method, *Psychometrika* **29**(1): 115–129.

Kruskal, J. B. and Wish, M. 1978. *Multidimensional Scaling. Sage University Paper Series on Quantitative Applications in the Social Sciences*, 07-011, Sage Publications, Beverly Hills and London.

Mardia, K., Kent, J. and Bibby, J. 1979. *Multivariate Analysis*, Academic Press.

Posse, C. 1999. Hierarchical model-based clustering for large datasets, *Technical Report 363*, Department of Statistics, University of Washington, Seattle, Washington.

Rabiner, L. and Juang, B. 1993. *Fundamentals of Speech Recognition*, PTR Prentice Hall, Inc.

Raftery, A. 1995. Bayesian model selection in social research (with discussion), *Sociological Methodology* pp. 111–196.

Schütze, H. and Silverstein, C. 1997. Projections for effi cient document clustering, *SIGIR Forum* pp. 74–81.

Shepard, R. N. 1980. Multidimensional scaling, tree-fi tting, and clustering, *Science* **210**(4468): 390–398.

Swayne, F., Cook, D. and Buja, A. 1998. XGobi: Interactive dynamic data visualization in the X window system, *J. Computational and Graphical Statistics* **7**: 113–130.

Tantrum, J., Murua, A. and Stuetzle, W. 2002. Hierarchical model-based clustering of large datasets through fractionation and refrectionaton, *KDD Conference 2002, Edmonton, Canada*.

Tantrum, J., Murua, A. and Stuetzle, W. 2004. Hierarchical model-based clustering of large datasets through fractionation and refrectionaton, *Information Systems* **29**: 315–326.

Van Rijsbergen, C. J. 1979. *Information Retrieval*, Butterworths, London.

Wallace, D. L. 1983. Comment, *J. American Statistical Association* **78**: 569–579.

Ward, J. H. 1963. Hierarchical groupings to optimize an objective function, *J. American Statistical Association* **58**: 234–244.

Zobel, J. and Moffat, A. 1998. Exploring the similarity space, *SIGIR Forum* **32**(1): 18–34.