Clustering Documents with an Exponential-Family Approximation of the Dirichlet Compound Multinomial Distribution

Charles Elkan ELKAN@CS.UCSD.EDU
Computer Science and Artificial Intelligence Laboratory, Massachusetts Institute of Technology, Cambridge MA 02139
Department of Computer Science and Engineering, University of California, San Diego, La Jolla, CA 92093-0404

Abstract

The Dirichlet compound multinomial (DCM) distribution, also called the multivariate Polya distribution, is a model for text documents that takes into account burstiness: the fact that if a word occurs once in a document, it is likely to occur repeatedly. We derive a new family of distributions that are approximations to DCM distributions and constitute an exponential family, unlike DCM distributions. We use these so-called EDCM distributions to obtain insights into the properties of DCM distributions, and then derive an algorithm for EDCM maximum-likelihood training that is many times faster than the corresponding method for DCM distributions. Next, we investigate expectation-maximization with EDCM components and deterministic annealing as a new clustering algorithm for documents. Experiments show that the new algorithm is competitive with the best methods in the literature, and superior from the point of view of finding models with low perplexity.

1. Introduction

In a text document, if a word occurs once, it is likely that the same word will occur again. This phenomenon is distinct from the obvious phenomenon that different words are more common for different topics. For example, consider a collection of documents that are all about the car industry. Naturally, words like "automotive" are more common than words like "aerospace," but suppose that the words "Toyota" and "Nissan" are equally common overall for this topic. Nevertheless, if "Toyota" appears once, a second appearance of "Toyota" is much more likely than a first appearance of "Nissan."

The phenomenon just explained is called burstiness. The multinomial distribution is used very widely to model text documents, but it does not account for burstiness. As an alternative model for documents, a recent paper proposed the so-called Dirichlet compound multinomial distribution (DCM) (Madsen et al., 2005). The present paper investigates the DCM further. We have three main contributions. First, we provide additional insight into why the DCM is appropriate for modeling documents. Second, we derive a new distribution that is a close approximation to the DCM and, unlike the DCM, is a member of the exponential family of distributions. Third, we use expectation-maximization (EM) with the new distribution to obtain a new clustering algorithm for documents. Experiments show that the new algorithm is competitive with the best methods in the literature, and superior from the point of view of finding low-perplexity models.

Previous work has argued the case for the DCM persuasively, and has shown that classifiers using Bayes' rule and the DCM are competitive with the best-known classification methods on standard document collections (Madsen et al., 2005). Mixtures of DCM components have been proposed independently for language modeling (Yamamoto et al., 2003). However, the DCM approach is not without problems. First, although Dirichlet distributions constitute an exponential family, DCM distributions do not. Exponential families have many desirable properties (Banerjee et al., 2005b, Section 4) which DCM distributions fail to share. Second, the expression for a DCM distribution lacks intuitiveness, so understanding its behavior qualitatively is difficult. Third, DCM parameters cannot be estimated quickly; gradient descent in high dimensions is necessary (Minka, 2003). Fast training is important not only for modeling large document collections, but also for using DCM distributions in more complex models including the mixture models discussed in this paper and hierarchical models such as LDA (Blei et al., 2003).

This paper presents a new family of distributions that we call EDCM distributions. EDCM distributions approximate DCM distributions, while overcoming each of the three dis-

Appearing in *Proceedings of the 23^{rd} International Conference on Machine Learning*, Pittsburgh, PA, 2006. Copyright 2006 by the author(s)/owner(s).

advantages of DCM distributions just mentioned. Although the focus here is on modeling text documents, the DCM is applicable in many other domains also where burstiness, sometimes called contagion, is important. We expect that the results of this paper will be useful in these other domains also, for example (Kvam & Day, 2001).

This paper is organized as follows. First, in Section 2 we revisit the DCM distribution and explain two different generative models that both lead to it. Next, Section 3 derives the EDCM distribution and discusses insights obtainable from it. Section 4 applies expectation-maximization with the EDCM to obtain a new clustering algorithm for text documents. Section 5 describes the design of experiments to evaluate the performance of the new algorithm, while Section 6 presents the results of these experiments. Finally, Section 7 concludes the paper.

2. New perspectives on the DCM

Given a document x, let x_w be the number of appearances of word w, where w ranges from 1 to the vocabulary size W. The DCM distribution, also called the multivariate Polya distribution, is

$$p(x) = \frac{n!}{\prod_{w=1}^{W} x_w!} \frac{\Gamma(s)}{\Gamma(s+n)} \prod_{w=1}^{W} \frac{\Gamma(x_w+\alpha_w)}{\Gamma(\alpha_w)} \qquad (1)$$

where the length of the document is $n = \sum_{w=1}^{W} x_w$ and $s = \sum_{w=1}^{W} \alpha_w$ is the sum of the parameters.

Like a multinomial distribution, a DCM is a distribution over all possible count vectors that sum to a fixed value n. When a DCM or a multinomial is used to model a collection of documents of different lengths, formally there is a different distribution for each different length, with all distributions sharing the same parameter values. Also, with a DCM or with a multinomial, $p(x)$ for a document x is really the probability of the equivalence class of all documents that have the same word counts, that is all documents that have the same bag-of-words representation. The cardinality of this equivalence class is given by the multinomial coefficient $n!/\prod_{w=1}^{W} x_w!$.

The DCM arises naturally from at least two different perspectives, both of which are generative. The first perspective, which is the one presented by (Madsen et al., 2005), is that a document is generated by drawing a multinomial from a Dirichlet distribution, then drawing words from that multinomial. Equation (1) is obtained by integrating over all possible multinomials: $p(x) = \int_\theta p(x|\theta)p(\theta|\alpha)$ where $p(\theta|\alpha)$ is a Dirichlet distribution and $p(x|\theta)$ is a multinomial distribution. The intuition behind this perspective is that the Dirichlet represents a general topic, while each multinomial is a document-specific subtopic that makes certain words especially likely for this particular document. For example, some articles about the car industry may be generated from a multinomial that gives high probability to the word “Toyota,” while others are generated from a multinomial that emphasizes the word “Nissan.” The Dirichlet that represents the entire car industry topic gives high probability to both these multinomials.

The second perspective is that a document is generated following a so-called urn scheme. Consider an urn filled with colored balls, with one color for each word in the vocabulary. The simplest scheme is to draw balls with replacement, and to count for each color how many times a ball of that color is drawn. This scheme yields a multinomial distribution, where the parameters of the multinomial are the fractions of ball colors. Note that although the drawing process is sequential, only the total number of balls drawn of each color is recorded.

An alternative urn scheme was first studied by Polya and Eggenberger (Johnson et al., 1997, Chapter 40). In this scheme, each time a ball is drawn it is replaced *and* one additional ball of the same color is placed in the urn. Following this scheme, words that have already been drawn are more likely to be drawn again. One can make the urn process more bursty or less bursty by decreasing or increasing the number of balls in the urn initially, without changing the proportions of ball colors. For example, consider an urn with equal numbers of balls of two colors. If the urn contains k balls of each color, the chance that the second ball drawn has the same color as the first is $(k+1)/(2k+1)$. The smaller k is, the more the urn scheme will be bursty.

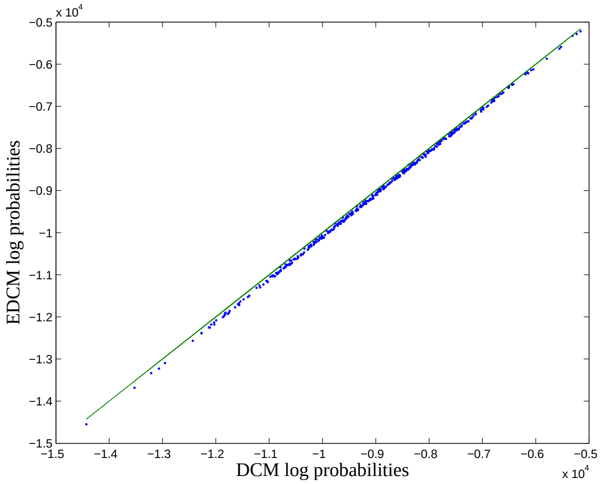
Sampling following the Polya-Eggenberger urn scheme results in count vectors that follow a DCM distribution. Each DCM parameter α_w is the number of balls of color w in the urn initially. (Each α_w may be less than one and need not be an integer.) The sum $s = \sum_{w=1}^{W} \alpha_w$ measures the overall burstiness of the distribution. Increasing s decreases burstiness and vice versa. As $s \to \infty$ the DCM tends towards a multinomial.[1]

3. Approximating the DCM

In this section we derive a family of distributions that is an exponential family and is also an approximation to the DCM family. We call the members of the new family EDCM distributions. We investigate their properties, and we show how maximum likelihood parameter values can be computed efficiently for them.

[1] It is not obvious that the two generative processes described above give rise to the same distribution. Indeed, this fact is not mentioned by the standard monograph (Johnson et al., 1997), where the first perspective is described on pages 80–83 and the second perspective is described separately on pages 200–211. However, rearrangements show that Equations 35.152 on page 80 and 40.12 on page 202 are equivalent.

Figure 1. DCM probabilities versus EDCM probabilities.



Given any natural collection of documents, most words do not appear in most documents, that is most counts are zero. For computational efficiency, it should be possible to evaluate $p(x)$ as a function of non-zero x_w values only. This is the case for a DCM distribution:

$$p(x) = \frac{n!}{\prod_{w:x_w \geq 1} x_w!} \frac{\Gamma(s)}{\Gamma(s+n)} \prod_{w:x_w \geq 1} \frac{\Gamma(x_w + \alpha_w)}{\Gamma(\alpha_w)} \tag{2}$$

since $x_w! = 1$ and $\Gamma(x_w + \alpha_w)/\Gamma(\alpha_w) = 1$ if $x_w = 0$.

Empirically, given a DCM fitted by maximum likelihood to a set of documents, $\alpha_w \ll 1$ for almost all words w. For example, for a DCM trained on the NIPS document collection described below (Section 5), the average α_w is 0.0636. Of the 6,871 parameters, 84% are less than 0.1 and only 25 are above 1.0. For small α, a useful fact is that

$$\lim_{\alpha \to 0} \frac{\Gamma(x+\alpha)}{\Gamma(\alpha)} - \Gamma(x)\alpha = 0.$$

for $x \geq 1$. Replacing $\Gamma(x_w + \alpha_w)/\Gamma(\alpha_w)$ by $\Gamma(x_w)\alpha_w$ in Equation (2) and using the fact that if z is an integer then $\Gamma(z) = (z-1)!$, we obtain the EDCM distribution:

$$q(x) = \frac{n!}{\prod_{w:x_w \geq 1} x_w} \frac{\Gamma(s)}{\Gamma(s+n)} \prod_{w:x_w \geq 1} \beta_w. \tag{3}$$

For clarity, we denote the EDCM parameters β_w.

Using $\Gamma(x)\alpha$ instead of $\Gamma(x+\alpha)/\Gamma(\alpha)$ is a highly accurate approximation for small α and small integers x. In the NIPS collection, 92.7% of all x_w values are $x_w = 0$, 4.2% are $x_w = 1$, for which the approximation is exact, and only 3.1% are $x_w \geq 2$. In practice the probabilities given by Equation (3) are very close to those given by Equation (1). Figure 1 shows the probability assigned to each document in the NIPS collection by the maximum-likelihood EDCM distribution compared to the probability assigned by the maximum-likelihood DCM distribution. The average difference is less than 1%.

Since $q(x)$ is an approximation, the normalization constant $Z(\beta) = n!\Gamma(s)/\Gamma(s+n)$ is not exact. In principle $q(x)$ could be summed over all values of x to get the exact normalization constant. We believe $q(x)$ is always a good approximation for real text data because text shows consistent burstiness behavior, and zero or small counts for most words. The results in Table 1 below indicate that the approximation has no statistically significant impact on performance clustering documents.

Since the EDCM expression is relatively simple, we can gain insights from it, and since the EDCM approximates the DCM, insights for one distribution carry over to the other. The EDCM can also be written

$$q(x) = n! \frac{\Gamma(s)}{\Gamma(s+n)} \prod_{w:x_w \geq 1} \frac{\beta_w}{x_w}. \tag{4}$$

This form makes clear that for fixed s and n, the probability of a document is proportional to $\prod_{w:x_w \geq 1} \beta_w / x_w$. This means that the first appearance of a word w reduces the probability of a document by β_w, a word-specific factor that is almost always much less than 1.0, while the mth appearance of any word reduces the probability by $(m-1)/m$, which tends to 1 as m increases. This behavior reveals how the EDCM, and hence the DCM, allow multiple appearances of the same word to have high probability. In contrast, with a multinomial each appearance of a word reduces the probability by the same factor.

Another point of view on Equation (4) is that it distinguishes between word types and word tokens. The β_w factors are word-type parameters while the $1/x_w$ factors are word-token parameters. It has been argued recently that modeling both word type frequencies and word token frequencies is useful for capturing the statistical properties of natural language (Goldwater et al., 2005). Unlike DCM and EDCM models, multinomial models ignore the type-token distinction, because given a document collection, the parameters of the maximum-likelihood multinomial are the same regardless of where the boundaries between documents are in the collection. On the other hand, maximum-likelihood DCM and EDCM models are sensitive to which words appear in which documents, i.e. to document boundaries and not just to total word counts.

The members of an exponential family of distributions have the form $f(x)g(\beta)\exp[t(x) \cdot h(\beta)]$ where $t(x)$ is a vector of sufficient statistics and $\theta = h(\beta)$ is the vector of so-called “natural” parameters; for details see (Banerjee et al., 2005b, Section 4.1). We can write $q(x)$ in this form as

$$\left(\prod_{w:x_w \geq 1} x_w^{-1}\right) n! \frac{\Gamma(s)}{\Gamma(s+n)} \exp[\sum_{w=1}^{W} I(x_w \geq 1) \log \beta_w].$$

The sufficient statistics for a document x are $\langle t_1(x), \cdots, t_W(x)\rangle$ where $t_w(x) = I(x_w \geq 1)$. We can obtain maximum likelihood parameter estimates for the EDCM by taking the derivative of the log-likelihood function. From (4) the log-likelihood of one document is

$$\log n! + \log \Gamma(s) - \log \Gamma(s+n) + \sum_{w:x_w \geq 1} \log \beta_w - \log x_w.$$

Given a collection D of documents, where document number d has length n_d and word counts x_{dw}, the partial derivative of the log-likelihood of the collection is

$$\frac{\partial l(D)}{\partial \beta_w} = |D|\Psi(s) - \sum_d \Psi(s+n_d) + \sum_d I(x_{dw} \geq 1)\frac{1}{\beta_w}$$

where $\Psi(z)$ is the digamma function, which is the derivative of $\log \Gamma(z)$. Setting this expression to zero and solving for β_w gives

$$\beta_w = \frac{\sum_d I(x_{dw} \geq 1)}{\sum_d \Psi(s+n_d) - |D|\Psi(s)}. \tag{5}$$

We can compute $s = \sum_w \beta_w$ by summing each side of Equation (5) over all words, giving

$$s = \frac{\sum_w \sum_d I(x_{dw} \geq 1)}{\sum_d \Psi(s+n_d) - |D|\Psi(s)} \tag{6}$$

where the numerator is the number of times a word appears at least once in a document. This equation involves only a single unknown, s, so it can be solved numerically efficiently by Newton's method. Once s is known, each individual β_w can be computed directly using Equation (5).[2]

4. Mixtures of EDCMs

For learning a mixture model, expectation-maximization (EM) can be summarized as follows (Banerjee et al., 2005b). In the E step, one computes weights using Bayes' rule, i.e. with the same equation regardless of what the distribution is. In the M step, one uses a maximum-likelihood estimator for a weighted log-likelihood function. The equations that implement this process are as follows. For the E step, the probability m_{id} that document d is generated by mixture component i is

$$m_{id} = p(i|x_d) = \frac{\mu_i p(x_d|\theta_i)}{\sum_j \mu_j p(x_d|\theta_j)}$$

where μ_i is the prior probability of component i of the mixture, and θ_i is the parameter vector of component i. For the M step, μ_i and θ_i are re-estimated as follows with the weights m_{id} held fixed:

$$\mu_i = \frac{M}{D} \quad \text{and} \quad \theta_i = \arg\max_\theta \sum_d m_{id} \log p(x_d|\theta)$$

where $M = \sum_d m_{id}$. For the EDCM, we can obtain the weighted maximum-likelihood parameter vector easily:

$$\frac{\partial}{\partial \beta_w} \sum_d m_{id} \log p(x_d|\theta) =$$

$$M\Psi(s) - \sum_d m_{id}\Psi(s+n_d) + \sum_d m_{id} I(x_{dw} \geq 1)\frac{1}{\beta_w}.$$

As before, first we solve an equation in one unknown for s,

$$s = \frac{\sum_w \sum_d m_{id} I(x_{dw} \geq 1)}{\sum_d m_{id}\Psi(s+n_d) - M\Psi(s)},$$

and then we compute each β_w for component i as

$$\beta_w = \frac{\sum_d m_{id} I(x_{dw} \geq 1)}{\sum_d m_{id}\Psi(s+n_d) - M\Psi(s)}.$$

The equations above are the foundation of clustering using the EDCM, but as always with EM, algorithmic details are important. First, as is usual with likelihood computations, all probabilities are represented as logarithms to avoid underflow. In addition, to avoid overflow without losing precision, weights m_{id} are computed as

$$m_{id} = \frac{\exp(\log \mu_i + \log p(x_d|\theta_i) - c)}{\sum_j \exp(\log \mu_j + \log p(x_d|\theta_j) - c)} \tag{7}$$

where $c = \max_j\{\log \mu_j + \log p(x_d|\theta_j)\} - 100$.

Second, a deterministic annealing procedure allows EM to find better local optima of the likelihood function (Ueda & Nakano, 1998). This procedure has three phases. Each phase runs EM until convergence with $\log p(x_d|\theta_i)$ replaced by $(1/T) \log p(x_d|\theta_i)$ in Equation (7), where T is a temperature parameter. The final parameter values in each phase are used as initial values in the next phase. The three phases use $T = 25$, $T = 5$, and finally $T = 1$. We find that slower annealing schedules provide no significant additional benefit.

General justifications for the annealing procedure are given by (Ueda & Nakano, 1998). We can add two additional points of view here. The first point of view is that dividing $\log p(x_d|\theta_i)$ by T is a heuristic way of calibrating the membership probabilities m_{id}, i.e. of making them reflect genuine membership uncertainty more realistically. Although the EDCM has the virtue of not assuming that different appearances of the same word are independent, it still does

[2] Equations (5) and (6) are similar to (118) and (108) in (Minka, 2003).

assume that different words provide independent evidence for the membership of a document in a class. This assumption is not completely true, so the EDCM (also the DCM, and the multinomial) tends to give probabilities $p(x_d|\theta_i)$ that are excessively confident, i.e. too close to zero, and hence weights m_{id} that are too close to zero or one. Dividing $\log p(x_d|\theta_i)$ by T where $T > 1$ makes all weights further away from zero and one, i.e. more realistic.

The second point of view is that making weights m_{id} be further away from zero and one slows down convergence of the EM algorithm, which allows it to explore a larger region of the parameter space, instead of fixating quickly on a local optimum close to whatever the initial parameter values are. Slower convergence has been argued to be an important factor in good performance of a soft clustering algorithm (Banerjee et al., 2005a, Section 7).

Given that the annealing procedure broadens exploration of the parameter space, we use a simple initialization method that is designed to maximize the uniformity of the weights computed in the E step of the first iteration of EM. One EDCM is fitted to the entire document collection, and then the parameters of each component are set to be a different random small perturbation of this EDCM. The initial mixing proportions are uniform, $\mu_i = 1/k$, where k is the number of clusters to be discovered.

5. Experimental design

The goal of the experiments described here is to investigate whether clustering with mixtures of EDCM components can summarize well the diversity in heterogeneous collections of documents. The experimental design choices include: (1) which algorithms to compare against? (2) which document collections to use? (3) which performance metrics to use? (4) exactly which experiments to run?

For (1) we select as a baseline method EM for a mixture of multinomials. We also select an especially elegant and interesting algorithm that has been reported to perform well recently, namely the soft-movMF method of (Banerjee et al., 2005a, Section 5). This method is EM using components that are von Mises-Fisher (vMF) distributions. Details of our implementation of it are explained below. The experiments of (Banerjee et al., 2005a) are careful and thorough, using twelve different document collections. On all these collections, the soft-movMF method performs better than the three methods it is compared against. Banerjee *et al.* write

> It has already been established that k-means using Euclidean distance performs much worse than spherical k-means for text data (Strehl et al., 2000), so we do not consider it here. Generative model based algorithms that use mixtures of Bernoulli or multinomial distributions, which have been shown to perform well for text data sets, have also not been included in the experiments. This exclusion is done as a recent empirical study over 15 text data sets showed that simple versions of vMF mixture models (with κ constant for all clusters) outperform the multinomial model except for only one data set (Classic3), and the Bernoulli model was inferior for all data sets (Zhong & Ghosh, 2003).

For this paper, we *do* include EM for mixtures of multinomials. Importantly, our implementation of multinomial EM uses deterministic annealing as described in the previous section, so it performs much better than previously reported.

Implementing the soft-movMF algorithm is straightforward following (Banerjee et al., 2005a), except that the von Mises-Fisher distribution requires computing modified Bessel functions $I_\nu(\cdot)$ of the first kind of high order ν. Widely available implementations of $I_\nu(z)$ yield underflow or overflow for large ν and z. To overcome this problem, we use an approximation of $\log I_\nu(z)$ from (Abramowitz & Stegun, 1974, Equation 9.7.7):

$$\log I_\nu(z) \approx -\log\sqrt{2\pi\nu} + \nu\eta - 0.25\log\alpha$$

with $\alpha = 1+(z/\nu)^2$ and $\eta = \sqrt{\alpha}+\log(z/\nu)-\log(1+\sqrt{\alpha})$. This approximation is highly accurate and gives essentially the same clustering results as using the exact value of $I_\nu(z)$ whenever the latter is computable. Preliminary experiments suggest that the soft-movMF algorithm does not benefit from annealing, so we run it without annealing.

For (2), we show detailed results for two especially interesting document collections. The first is the Classic400 collection from (Banerjee et al., 2005a). This collection is chosen because it reveals the biggest performance differences between methods among all twelve collections used by (Banerjee et al., 2005a), and soft-movMF performs particularly well on it. The second collection contains the OCRed text of all papers published in the 2002 and 2003 NIPS proceedings (Globerson et al., 2004). This collection has relatively long documents and subtle differences between topics. We use only papers from 2002 and 2003 since the organization of papers into research areas was different in earlier and later years. Papers that seem to be incompletely captured, i.e. that are less than 700 words long, are eliminated. The properties of these two document collections are summarized in the first columns of Table 1 below. Both collections have no feature selection and no stemming, but have had stop words removed.

We also report summarized results for fifteen collections used in previous work (Zhong & Ghosh, 2005). These collections seem to contain spelling mistakes and other nonwords. To reduce the impact of these, for each collection we remove all terms that appear in just one document, or in more than half of all documents. Since stop words have already been removed in these collections, we are likely not removing any additional genuine words.

Question (3) asks which performance metrics to measure. A good clustering algorithm is one that identifies groups that seem meaningful to people. In experiments, a proxy for this subjective criterion is whether the algorithm finds groups that have been recognized previously as meaningful by humans. Therefore, one performance metric quantifies how much the clustering found by an algorithm agrees with a prespecified clustering. Of the many such metrics, we choose to use mutual information (MI) as used by (Banerjee et al., 2005a). Let m_i be the number of documents said to be in cluster i, let n_j be the number of documents with prespecified label j, and let c_{ij} be the number of documents in cluster i with label j. With D total documents, define $p_i = m_i/D$, $q_j = n_j/D$, and $r_ij = c_{ij}/D$. The MI metric is then

$$\sum_i \sum_j r_{ij} \log \frac{r_{ij}}{p_i q_j}.$$

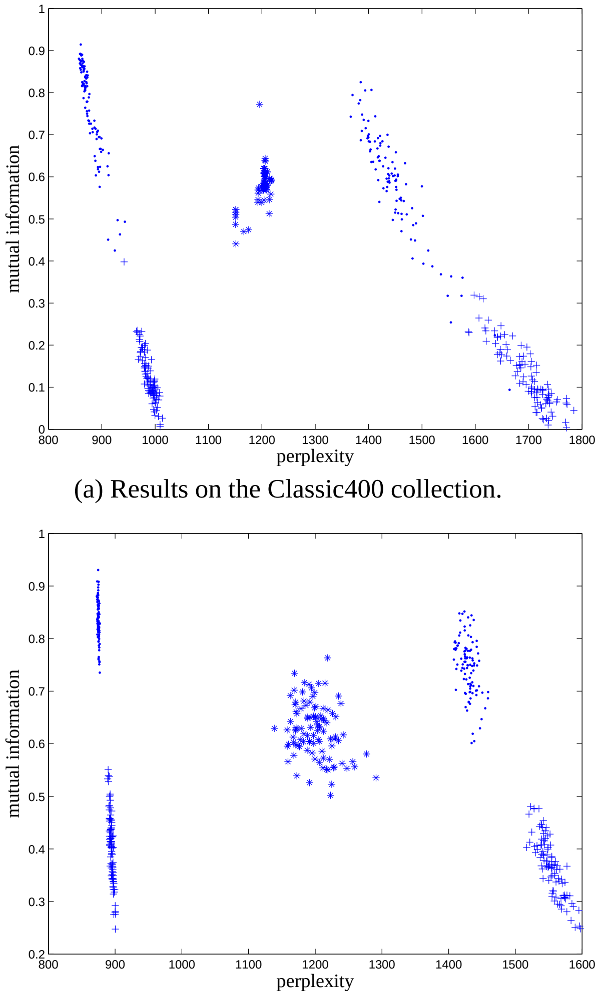
We also want to know how well a method models the content of a document collection. To answer this question, we measure perplexity. Intuitively, perplexity is the average uncertainty the model assigns to each word in a document collection. Perplexity is not well-defined for all models. First, it requires a discrete probability mass function, not a continuous probability density function. For this reason we cannot talk about the perplexity of a von Mises-Fisher model. Second, perplexity is a per-word metric. Distributions like the multinomial and the EDCM assign a probability mass to an equivalence class of documents. To get a meaningful perplexity number, the mass assigned to a single document must be defined. The most straightforward approach is to divide the probability mass equally between all documents in the equivalence class. Specifically, for a multinomial distribution θ we define $p(x|\theta) = \prod_w \theta_w^{x_w}$ and for an EDCM with parameters β we define

$$p(x|\beta) = \left(\frac{n!}{\prod_{w=1}^{W} x_w!} \right)^{-1} n! \frac{\Gamma(s)}{\Gamma(s+n)} \prod_{w:x_w \geq 1} \frac{\beta_w}{x_w}.$$

The perplexity of a model for a document collection is then $\exp\left(\sum_d \log p(x_d) / \sum_d n_d\right)$. If the cardinality of the vocabulary is W, a uniform multinomial model has perplexity W. A model that successfully assigns higher probability to documents in the collection than to others will have perplexity less than W. Since a mixture model with more components has more free parameters, it will have lower (or equal) perplexity than one with fewer components.

As is customary in research on clustering, the metrics used in this paper are all evaluated on the experimental data directly. In other words, no separate test set of documents is used. If one did use separate training and test sets, it would be necessary to smooth or regularize the distributions found by maximum likelihood.

Figure 2. Results of five clustering algorithms.



(a) Results on the Classic400 collection.

(b) Results on the NIPS collection.

Finally, question (4) above concerns exactly which experiments to perform. We keep this design as simple as possible. For the Classic400 and NIPS collections, we run each algorithm 100 times with different random initializations. For the other fifteen collections, we run ten times with different random initializations. We set the number k of clusters to be found to be the same as the number of prespecified classes. We report the average MI with standard errors, and also the average perplexity. As a rule of thumb, differences between methods are significant at around the 5% level if their mean ± standard error ranges do not overlap.

Table 1. Clustering results on two document collections (D: number of documents, $\bar{n}_d$: average document length, W: vocabulary size, k: number of classes, p: perplexity ± standard error, MI: mutual information ± standard error, time measured in seconds).

					DCM			EDCM		
name	D	$\bar{n}_d$	W	k	p	MI	time	p	MI	time
Classic	400	78.8	6205	3	853.96 ± 1.91	0.77197 ± 0.01063	49.7	877.64 ± 1.87	0.77203 ± 0.01136	2.35
NIPS	391	1332.4	6871	9	804.19 ± 0.24	0.84364 ± 0.00659	751.3	875.17 ± 0.15	0.83705 ± 0.00737	6.61

6. Experimental results

An important preliminary question is whether the EDCM approximation leads to as good clustering results as the DCM distribution. Table 1 indicates that the answer to this question is yes, since the differences in mutual information values achieved are not statistically significant. The perplexity values shown for the two methods are different because these values depend on the normalization constant for each distribution. This constant is not known exactly for the EDCM, and computed EDCM perplexity values are overestimates. The times shown in Table 1 are for one clustering run to convergence, for comparably optimized Matlab code. DCM-based clustering is 21 and 114 times slower.

Figure 2 shows the outcome of running five different clustering algorithms on the Classic400 and NIPS document collections. For each algorithm 100 points are plotted in each panel. The dots in the top left cloud are measurements for EM with the EDCM distribution, with annealing. The stars in the middle cloud correspond to EM with the von Mises-Fisher distribution, while the dots in the top right cloud correspond to EM with the multinomial distribution, again with annealing. The bottom left cloud of plus signs corresponds to EM without annealing with the EDCM; the bottom right cloud is for EM without annealing with the multinomial.

Several conclusions are clear from Figure 2. First, the best perplexity and the best mutual information are achieved by EDCM-based clustering with annealing. For all EDCM and multinomial methods, there is a strong correlation between good perplexity and good mutual information, within the confines of each method. This means that for these methods optimizing perplexity, which can always be measured, is a good way to maximize mutual information, which cannot be measured in real applications.

As mentioned above, perplexity is not defined for continuous models such as mixtures of vMF distributions. Therefore, the horizontal axis for vMF results in each panel of Figure 2 shows scaled negative log-likelihood, not perplexity. In panel (b) the correlation between better log-likelihood and better mutual information is weak, while in panel (a) the direction of correlation is reversed. These correlations suggest that when clustering with a mixture of vMF distributions, optimizing log likelihood is unfortunately not a good way to optimize mutual information.

Table 2. Alternative clusterings of the Classic400 collection.

Medline	91	1	8
CISI	1		99
Cranfield		194	6

(a) EDCM clustering

Medline	99		1
CISI	100		
Cranfield	25	134	41

(b) vMF clustering

For the Classic400 collection, the average mutual information achieved by vMF clustering is 0.582 ± 0.004 (mean plus/minus standard error), almost exactly the same as reported previously for this algorithm on this collection (Banerjee et al., 2005a). The average mutual information achieved by EDCM clustering with annealing is 0.772 ± 0.011, and by multinomial clustering with annealing 0.588 ± 0.013, but only 0.127 ± 0.007 without annealing. All differences are statistically significant, except between the vMF method and the multinomial with annealing method. One may ask whether these performance differences matter in practice. Table 2 shows the confusion matrix of the clustering that has best log likelihood among all 100 plotted in panel (a) of Figure 2. It is clear that the EDCM method succeeds in separating the three true groups, while the vMF method does not.

Table 3 shows the average perplexity and average mutual information achieved with ten runs of each algorithm on each of the fifteen document collections used in (Zhong & Ghosh, 2005). For comparability, mutual information is normalized as explained in (Zhong & Ghosh, 2005). Standard errors are not given for perplexity averages to improve readability; they are typically around ± 10 or less. EDCM-based clustering yields statistically significantly superior perplexity for every document collection. Results as measured by mutual information are not clearcut: each algorithm is the best on some collections. For both multinomial and EDCM-based clustering, for all collections except two, better perplexity is correlated with better mutual information. For vMF-based clustering, better perplexity is unfortunately correlated with worse mutual information just as often as with better mutual information.

Table 3. Perplexity (p), normalized mutual information (NMI ± standard error), and correlation (r) results on fifteen document collections.

					multinomial			EDCM			vMF	
name	D	$\bar{n}_d$	W	k	p	NMI	r	p	NMI	r	NMI	r
NG20	19949	120.0	43585	20	2944	0.590 ± 0.004	0.73	1953	0.546 ± 0.005	0.75	0.249 ± 0.006	0.07
NG17-19	2998	164.9	15808	3	3857	0.375 ± 0.028	0.69	2594	0.150 ± 0.036	0.47	0.149 ± 0.001	0.94
classic	7089	42.9	12009	4	1370	0.699 ± 0.020	0.36	912	0.729 ± 0.028	0.82	0.315 ± 0.006	0.99
ohscal	11162	104.2	11465	10	1235	0.384 ± 0.010	0.93	558	0.387 ± 0.004	0.24	0.261 ± 0.008	0.31
hitech	2301	228.3	10080	6	2086	0.282 ± 0.004	-0.07	1328	0.235 ± 0.005	-0.15	0.283 ± 0.005	-0.80
reviews	4069	281.2	18482	5	2866	0.593 ± 0.028	0.67	2062	0.509 ± 0.007	0.09	0.508 ± 0.016	-0.16
sports	8580	200.8	14866	7	1632	0.587 ± 0.013	0.48	1211	0.561 ± 0.009	0.86	0.448 ± 0.014	-0.98
la1	3204	248.3	17265	6	2714	0.498 ± 0.015	0.91	1579	0.417 ± 0.011	0.56	0.417 ± 0.014	0.60
la12	6279	246.3	11231	6	2882	0.525 ± 0.014	0.69	1624	0.445 ± 0.006	0.44	0.385 ± 0.016	-0.27
la2	3075	244.2	15203	6	2543	0.487 ± 0.014	0.89	1488	0.407 ± 0.011	0.77	0.390 ± 0.014	-0.18
k1b	2340	194.3	13856	6	2129	0.621 ± 0.014	0.64	1421	0.609 ± 0.013	0.27	0.585 ± 0.014	-0.53
tr11	414	1025.4	6412	9	1088	0.444 ± 0.025	0.55	964	0.382 ± 0.012	0.61	0.591 ± 0.011	0.76
tr23	204	2416.2	5814	6	1068	0.142 ± 0.015	-0.35	927	0.189 ± 0.016	0.86	0.258 ± 0.020	0.42
tr41	878	404.9	7445	10	1337	0.624 ± 0.013	0.72	1043	0.520 ± 0.010	-0.04	0.512 ± 0.015	-0.06
tr45	690	937.0	8249	10	1194	0.499 ± 0.022	0.22	1020	0.492 ± 0.015	0.17	0.535 ± 0.013	0.00

7. Discussion

The experimental results above show that mixtures of EDCM distributions always achieve much lower perplexity than mixtures of multinomials, when modeling document collections. The reason for this success is that EDCM models account correctly for the fact that if a word appears once in a document, it is likely to appear again, even if the first appearance is unlikely. In contrast, multinomial models tend to be excessively "surprised" by later occurrences.

We expect that the perplexity improvements achieved by mixtures of EDCMs will carry through to more complex models such as LDA and correlated topic models (Blei et al., 2003; Blei & Lafferty, 2005), when multinomials are replaced by EDCMs in these. Any document model that is built on top of multinomials either will suffer from improper modeling of burstiness, or will have to devote additional parameters to capturing burstiness, when this phenomenon could be captured by basing the complex model on DCM or EDCM models instead of on multinomials.

Acknowledgments: With thanks to Rasmus Madsen and David Kauchak for early work, to Fair Isaac, Inc. for funding, and to Tom Minka and others for useful comments.

References

Abramowitz, M., & Stegun, I. A. (1974). *Handbook of mathematical functions*. Dover Publications.

Banerjee, A., Dhillon, I. S., Ghosh, J., & Sra, S. (2005a). Clustering on the unit hypersphere using von Mises-Fisher distributions. *Journal of Machine Learning Research*, *6*, 1345–1382.

Banerjee, A., Merugu, S., Dhillon, I. S., & Ghosh, J. (2005b). Clustering with Bregman divergences. *Journal of Machine Learning Research*, *6*, 1705–1749.

Blei, D. M., & Lafferty, J. D. (2005). Correlated topic models. *Advances in Neural Information Processing Systems*.

Blei, D. M., Ng, A. Y., & Jordan, M. I. (2003). Latent Dirichlet allocation. *Journal of Machine Learning Research*, *3*, 993–1022.

Globerson, A., Chechik, G., Pereira, F., & Tishby, N. (2004). Euclidean embedding of co-occurrence data. *Advances in Neural Information Processing Systems* (pp. 497–504).

Goldwater, S., Griffiths, T. L., & Johnson, M. (2005). Interpolating between types and tokens by estimating power-law generators. *Advances in Neural Information Processing Systems*.

Johnson, N. L., Kotz, S., & Balakrishnan, N. (1997). *Discrete multivariate distributions*. John Wiley & Sons, Inc.

Kvam, P., & Day, D. (2001). The multivariate Polya distribution in combat modeling. *Naval Research Logistics*, *48*, 1–17.

Madsen, R., Kauchak, D., & Elkan, C. (2005). Modeling word burstiness using the Dirichlet distribution. *Proceedings of the Twenty-Second International Conference on Machine Learning* (pp. 545–552).

Minka, T. P. (2003). Estimating a Dirichlet distribution. Unpublished paper available at `http://research.microsoft.com/∼minka`.

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

Ueda, N., & Nakano, R. (1998). Deterministic annealing EM algorithm. *Neural Networks*, *11*, 271–282.

Yamamoto, M., Sadamitsu, K., & Mishina, T. (2003). Context modeling using Dirichlet mixtures and its applications to language models. *Information Processing Society of Japan-SIGSLP*, *104*, 29–34. In Japanese, available at `http://www.mibel.cs.tsukuba.ac.jp/∼sadamitsu/archive/ipsj2003/ipsj2003.pdf`.

Zhong, S., & Ghosh, J. (2003). A comparative study of generative models for document clustering. *Workshop on Clustering High Dimensional Data, Third SIAM Conference on Data Mining*.

Zhong, S., & Ghosh, J. (2005). Generative model-based document clustering: A comparative study. *Knowledge and Information Systems*, *8*, 374–384.