Model-Based Document Clustering with a Collapsed Gibbs Sampler

Daniel D. Walker and Eric K. Ringger
Natural Language Processing Lab
Computer Science Department
Brigham Young University
Provo, Utah, USA
{danl4,ringger}@cs.byu.edu

ABSTRACT
Model-based algorithms are emerging as a preferred method for document clustering. As computing resources improve, methods such as Gibbs sampling have become more common for parameter estimation in these models. Gibbs sampling is well understood for many applications, but has not been extensively studied for use in document clustering. We explore the convergence rate, the possibility of label switching, and chain summarization methodologies for document clustering on a particular model, namely a mixture of multinomials model, and show that fairly simple methods can be employed, while still producing clusterings of superior quality compared to those produced with the EM algorithm.

Categories and Subject Descriptors
I.5.3 [Pattern Recognition]: Clustering—Algorithms; G.3 [Probability and Statistics]: Probabilistic algorithms

General Terms
Algorithms, Experimentation, Performance

Keywords
document clustering, Gibbs sampling, MCMC, EM, collapsed samplers, practical guidelines

1. INTRODUCTION
Document clustering is an unsupervised learning task that partitions a set of documents \( D = d_1, ..., d_N \) to produce disjoint subsets called clusters. Members of the same cluster should be similar, while members of different clusters are dissimilar. Similarity and dissimilarity are subjective, and might refer to similarities based on topic, style, authorship, sentiment, or any number of criteria as dictated by the requirements of the specific clustering problem.

Practical applications of document clustering include the discovery of genres in large heterogeneous corpora, the automatic organization of document collections, novelty detection, exploratory data analysis, organizing search results, and text mining. Many of the same techniques that are used to cluster other types of data can be used for document clustering with varying degrees of success. The document feature space is high-dimensional and sparse, distinguishing it from other data clustering problems and requiring techniques suited to these properties to achieve good results. Research interest is transitioning from vector-space algorithms to statistical model-based algorithms for clustering [1, 10, 19, 24, 25]. As this transition occurs, many questions involving the methodology of model-based clustering need to be considered and answered.

Although there are models that are more elaborate, and potentially more robust than the one presented here [19], past research has focused mostly on developing new models and has not empirically explored the implications of the choices made during the implementation of a parameter estimation algorithm. This paper sets out to empirically investigate many of these questions and hopefully to serve as a set of guidelines for those who would apply MCMC (Markov Chain Monte Carlo) techniques to model-based document clustering, allowing them to make informed decisions as they implement their own algorithms. Specifically, we explore the use of a collapsed Gibbs sampler on a mixture of multinomials document model.

We will show the empirical effects on cluster quality of using various sample summarization methods and will demonstrate that within-chain label switching (non-identifiability) does not appear to be an issue when using the collapsed sampler with our model. We also provide evidence that the sampler converges quickly, within a relatively small number of samples. Finally, we show that the collapsed sampler clustering algorithm presented here produces better clusters than an EM clustering algorithm on the same model, according to five cluster quality metrics.

2. RELATED WORK
Many approaches have been proposed for clustering in general and document clustering in particular. Some of these techniques treat each datum as a vector in \( n \)-dimensional space. These techniques use measures such as Euclidean distance and cosine similarity to group the data together using agglomerative clustering, \( k \)-means clustering, and other similar methods.

Another set of techniques uses generative statistical models, such as Bayesian networks. These models usually include a latent variable denoting cluster assignment. One such model is the mixture of multinomials model [14], which we employ in this paper. In this model, the values of the individual features of each document are assumed to be conditionally independent and exchangeable, given the label of the document. This is called the “bag of words” assumption. The model is explained in Section 3.
Many methods can be used to estimate parameters in order to conduct inference and produce clusterings in model-based schemes. One such method is to begin with a random initialization of the parameters and then refine them using the EM algorithm. This approach has been shown superior to some vector-space solutions, such as agglomerative clustering [14]. However, as a hill-climbing algorithm EM is subject to becoming trapped by local maxima in the likelihood surface [3]. An alternative proposed in the literature is to use a Markov Chain Monte Carlo method, such as Gibbs sampling, to sample from the posterior distribution of the model parameters, given the data to be clustered [19, 24]. This approach is more correct from a Bayesian perspective, and has been shown to perform better than maximum likelihood on some problems such as the estimation of parameters for probabilistic grammars [8]. Like EM, Gibbs is naturally attracted to higher points on the likelihood surface, which are more likely to be sampled from. Unlike EM, an MCMC sampler is capable of leaving local maxima to explore less likely configurations. In this way, it is able to transition from local maxima to other areas of high likelihood. Thus, regardless of initialization, a Gibbs sampler may eventually come close to a true global maxima of the distribution in question, while an unfortunate initialization of EM can preclude it from ever reaching that mode, regardless of the number of iterations of the algorithm that are run.

Initialization of EM can preclude it from ever reaching that mode, regardless of the number of iterations of the algorithm that are run. Although MCMC-based document clustering techniques have been compared with EM-based techniques in the past by Banerjee and Basu [1], their work did not hold the model constant.

Despite the apparent advantages of MCMC methods, it is not clear how one should proceed when applying these methods specifically to document clustering, where high dimensionality, sparseness, and large numbers of data points are the rule.

3. MODEL

![Figure 1: Graphical representation of the generative document model used in this paper for clustering. Square nodes are constants, shaded nodes are observed values.](image)

The model used here is a mixture of multinomials, with Dirichlet priors. The parameters of the model are distributed as follows:

\[
\begin{align*}
\theta_j & \sim \text{Dirichlet}(\beta) \\
\lambda & \sim \text{Dirichlet}(\alpha) \\
z_{d} & \sim \text{Categorical}(\lambda) \\
w_{d|z_{d}} & \sim \text{Multinomial}(\theta_{j=z_{d}})
\end{align*}
\]

This model is illustrated in Figure 1. Here, \(w\) is a matrix representing the words in all \(M\) documents in the data. Document \(d\) from cluster \(j\) consists of a vector of \(N\) words \(w_{d|j}\), distributed according to a multinomial distribution parameterized by the vector \(\theta_{j}\). Assuming that \(z_{d} = j, \theta_{j,v}\) is the probability that word \(i\) of document \(d\) is the word \(v\) in the vocabulary \(V\):

\[
p(w_{d,i} = v) = \theta_j, v
\]

Likewise, each \(z_{d}\) is distributed according to a categorical distribution with parameter vector \(\lambda\). This means that the prior distribution on the value of \(z_{d}\) is given by:

\[
p(z_{d} = j) = \lambda_j
\]

Since both \(\lambda\) and \(\theta\) are latent variables in the model, we must select prior distributions for each. We choose the uniform Dirichlet distribution. This is also known as an uninformative Dirichlet prior and is achieved by setting the Dirichlet parameter vector (in our case \(\alpha\) and \(\beta\)) to all ones. The uninformative distribution was chosen because we assume that \textit{a priori} nothing is known about the word distributions for each class, nor about the marginal distribution of the document labels. As is common, we chose the Dirichlet because it is the conjugate prior for the Dirichlet.

Finally, it should be noted that the model includes an unexpressed parameter \(K\), the number of mixture components. Other work has explored model selection [14] and automatic estimation of this parameter using a Dirichlet process prior [15]. In this work we do not focus on this aspect of the problem. We instead choose a suitable constant value of \(K\) for each data set used for experimentation.

4. COLLAPSED SAMPLER

A collapsed sampler is one which eschews sampling all of the variables from the joint distribution. Instead, sampling is conducted over a simplified distribution from which all but the specific variables of interest have been marginalized out. Specifically, the variables which will not be sampled are marginalized from the complete conditionals of the variables that will be sampled. A complete conditional is the distribution of a single variable, conditioned on all of the other variables in the model.

In the case of clustering, the only specifically relevant variables are the hidden document cluster labels, \(z\). There may be some reason to sample the values for the \(\theta\) matrix as well, but this is not strictly part of the clustering task.

In addition to the storage benefit of not sampling from uninteresting variables, collapsed samplers have also been shown to converge relatively quickly because they contain fewer dependencies between sampled parameters and treat the marginalized parameters exactly [4, 9, 22].

Collapsed samplers cannot be used in all cases but have the limiting requirement that the variables to be marginalized out must be marginalizable in closed form. Fortunately, conjugacy between the Dirichlet and Multinomial distributions makes this possible for the model shown in Figure 1.

After marginalizing out \(\theta\) and \(\lambda\), the complete conditional over the label \(z_{d}\) for document \(d\) is:

\[
p(z_{d} = j|w, z_{-d}) = \frac{p(z, w)}{p(z_{-d}, w)}
\]

where \(z_{-d} = [z_1, \ldots, z_{d-1}, z_{d+1}, \ldots, z_M]\). We also have omitted \(\alpha\) and \(\beta\) from the conditions in each term for the sake of brevity. The right-hand-side of Equation (1) can also be expanded by explicitly showing that \(z_{d} = j\) giving:

\[
p(z_{d} = j|w, z_{-d}) = \frac{p(z_{d} = j, z_{-d}, w)}{p(z_{-d}, w)}
\]
In order to continue the derivation, it is necessary to have the marginalized joint distribution \( p(z, w) \). Because of the conjugacy between the Dirichlet distribution and the multinomial distribution, this consists of a product of Euler beta functions \( B(X) \):

\[
p(z_d = j | w, z_{-d}) \propto \frac{B(\alpha_s) \prod_{k=1}^{K} B(\beta_{ks})}{B(\alpha_{-ds}) \prod_{k=1}^{K} B(\beta_{-dks})}
\]

where we assume, for the purpose of computing counts, that \( z_d = j \) and that,

\[
\alpha_s = [\alpha_1 + n_1, ..., \alpha_K + n_K],
\]

\[
\beta_{ks} = [\beta_j + n_{jk}, ..., \beta_W + n_{Wk}],
\]

\[
\alpha_{-ds} = [\alpha_1 + n_1, ..., \alpha_j + n_j - 1, ..., \alpha_K + n_K],
\]

and \( \beta_{-dks} = \beta_{ks} \) for all other \( k \). In addition, \( n_j \) is the number of times that the label \( j \) has been applied to any document, \( n_{vj} \) is the number of times that word \( v \) occurs in a document labeled \( j \), and \( d_v \) is the number of times that word \( v \) occurs in document \( d \).

The derivation that produces this result is lengthy, and can be found in the Appendix. It follows a similar derivation for a related model, as presented in [19]. After simplifying we arrive at the following expression for the complete conditional for the label of document \( d \):

\[
p(z_d = j | w, z_{-d}) \propto (\alpha_j + n_j - 1) \frac{B(\beta_{ds})}{B(\beta_{-djs})}
\]

Using Equation 4, one may sample from \( p(z_d = j | w, z_{-d}) \), giving the labels of every other document. This complete conditional can now be used as part of a Gibbs sampling algorithm. The sampling algorithm used to conduct the experiments presented in this paper is shown in Algorithm 1.

### Algorithm 1:Collapsed Gibbs sampling algorithm.

In this section, we will discuss three issues as they relate to using \( Z \) to choose a clustering for \( D \).

1. How long does it take to converge?
2. How many samples should be taken?
3. How should the collected samples be summarized?

#### 5.1 Experiments

In this section we show empirically the effects that various algorithmic choices have on cluster quality in the text document domain. We also compare the quality of clusters produced with Gibbs sampling to that of clusters produced with EM. The experiments described here were conducted on three data sets labeled by topic. The first data set is the 20 News groups set which has become a standard in classification and clustering research [12].

The second data set is a portion of the Enron e-mail corpus which has been annotated and made available through LDC [2]. This set consists of about 5000 e-mail messages from the Enron corpus, which have each been identified as being related to one of thirty-two possible topics.

The third data set is a corpus of web pages annotated based on user tags found on the popular collaborative bookmarking website del.icio.us [23]. Del.icio.us allows users to tag each site they bookmark with a set of arbitrary tags. For this data set, 50 topics were selected, and each was converted to a del.icio.us tag intuitively derived from the topic label. For example, one of the topics is “Natural language processing”; documents are assigned to this topic if they have been given the three tags “natural”, “language” and “processing”. To be assigned to the topic “Apple”, a page merely has to have been tagged with the single tag “apple”.

We designed a feature selection process to facilitate topical clustering. Before clustering, each document undergoes feature extraction and vectorization. A list of stop-words were first removed from each document. Next, an unsupervised feature selection process was conducted in which the TF-IDF value for each word in each document was first calculated, then the 10 words in each document with the highest TF-IDF value were added to the feature set. Finally, each document was converted to a feature vector where each element in the vector is the frequency with which a given feature occurs in that document.

To evaluate cluster quality, five metrics were chosen from the literature. These metrics are all external metrics, meaning that they require a reference, or gold-standard, partitioning of the data. These metrics are the F-Measure [20], the V-Measure [18], and the \( Q_2 \) metric [6]. These metrics will be used below in any experiments where the quality of various partitionings of the same data set are compared.

#### 5.2 Convergence

MCMC sampling techniques are guaranteed to converge in the limit to the target distribution, given some reasonable assumptions. However, because consecutive draws in the chain can be highly correlated, the samples from the beginning of the chain can be highly influenced by the random initialization state. To correct this, MCMC algorithms often include a parameter called “burn” or “burn-in”, which specifies the number of initial samples that should be discarded from the beginning of the chain to reduce the influence of random initialization on the samples used for parameter estimation and inference.

Some authorities recommend general rule-of-thumb guidelines as to how many samples should be burned. Gelman [7] recom-
mends burning up to 50% of the collected samples, which is excessive when collecting samples is expensive, and when steady-state is achieved rapidly. Principled diagnostics have been proposed to help choose good values for burn \cite{16}. These diagnostics are mostly suited for the case where the variables being sampled are continuous, and they do not handle categorical variables well.

Because the variables being sampled in our algorithm are all categorical, we chose to avoid these formal diagnostics, and instead choose burn using likelihood time-series plots. Although there is no direct proof that this should be the case, we have found that the point at which this plot appears to approach an asymptote corresponds with the convergence of the chain to steady-state (c.f. \cite{9}). Several examples of these plots are shown in Figure 2. After examining several such plots for each data set, it was decided that the burn should be 1000 samples for the Enron data, 100 samples for the 20 Newsgroups data, and 50 samples for the del.icio.us data.

5.3 Identifiability

The model proposed in Section 3 does not contain any a priori knowledge of the clusters which are to be discovered through parameter estimation and inference. The labels applied by the clustering algorithm do not have any particular meaning before the clusters form, and are therefore completely interchangeable. For example, assume a data set with two distinct clusters, one consisting of documents about space exploration, and another consisting of documents about dogs. The model does not distinguish the case where all of the space exploration documents are labeled $1$, all of the dog related documents are labeled $2$, or vice versa. Solutions to the non-identifiability problem have been proposed. These often take the form of a re-labeling scheme \cite{17, 21}, and often feel slightly ad hoc.

We wanted to determine the extent to which label switching occurs using the collapsed sampler with our model. If within-chain label switching occurs often, then any summarization technique that attempts to use more than one sample will require steps for label switching diagnosis and compensation to be effective. To test for label switching, we ran a chain on the Enron data set, collecting a total of 6000 samples. The first sample after burn was stored as a reference iteration and we computed estimates for $p(w|z)$ given that sample. We used add-one smoothing to calculate this estimate:

$$p_0(w = v|z = j) = \frac{n_{w,j} + 1}{|V| + \sum_{w \in V} n_{w,j}}$$

where $n_{w,j}$ is the count of the number of times that word $v$ occurs in a document where the label of that document in the current sample is $j$.

After that, for the $i^{th}$ sample after burn a new distribution $p_i$ over words is estimated in the same way. Next, we compute the Kullback-Leibler divergence between $p_0(w|h)$ and $p_i(w|j)$ for all possible combinations of $h$ and $j$. These values were then scaled to the range $[0, 1]$ by dividing by the maximum and plotted in a quilt plot such that the square at square $(h, j)$ is the normalized value of the K-L divergence between $p_0(w|z = h)$ and $p_i(w|z = j)$.

This plot shows the similarity between clusters produced by different samples of the same MCMC chain. For example, the plot of $p_0$ against itself will obviously always have zeros (black squares) along the diagonal, and various other random values in the off-diagonal cells. If label switching occurs in a chain, however, we would expect that for larger values of $i$, the plot of $p_i$ against $p_0$ will begin to have higher values (lighter squares) along the diagonal. If labels $h$ and $j$ were to swap meaning, then we would expect $(j, j)$ and $(h, h)$ to have high divergence values, while $(j, h)$ and $(h, j)$ will both have divergence values close to zero.

The plots of $p_0$ against $p_0$ and $p_0$ against $p_{5000}$ for this experiment are shown in Figure 3. It can be seen that, even after 5000 samples, the diagonal elements are still close to zero. In fact, the differences between the two plots are minor, and noticeable under close scrutiny. This indicates that little, if any, label switching is occurring in this chain. This result is not unique to the Enron data set, nor to this particular run of the algorithm, though we omit presenting others because of space constraints. This result is likely the result of the high dimensionality of the data, together with the relatively high correlation between variables.
5.4 Summarizing Samples

We summarized sample chains produced by the collapsed sampler in three ways. The first method is called the marginal posterior method, because it considers the posterior distribution over labels for each document independent of all other documents. To summarize with the marginal posterior method the label selected for document \(d\), \(\hat{z}_d\), is chosen to be the label which was most frequently assigned to that document during sampling:

\[
\hat{z}_d = \arg\max_j \sum_i \delta(\hat{z}_{d,i}, j)
\]

where \(\delta()\) is the Kronecker delta function, which returns one if its two arguments are the same and zero otherwise.

The second method is the MAP (maximum a posteriori) sample method. This method takes the sample (a column vector \(Z_{i,:}\) from \(Z\)) with the highest posterior joint value as the selected partitioning of the data set. Since the priors specified in the model are uniform, this is the same as choosing the sample which maximizes the joint probability of the labeling and the data:

\[
\hat{z} = \arg\max_{Z_{i,:}} p(Z_{i,:}, w)
\]

The final method is the random method. In the random method, the sample at an arbitrary point in the chain is chosen to be the selected partitioning of the data set.

In order to determine which of these methods produces the best clusterings, 100 chains were run for 80 hours each on each of the three data sets on Dell Poweredge 1955s with two Dual-core 2.6GHz Intel Xeon processors and 8 GB of RAM. Because of the imposed time limit, different numbers of samples were collected for each run, though chains for the same data set tended to have roughly the same number of samples. For each data set, we rounded the number of samples found in the shortest chain down to the nearest 10. Consequently, the Social Bookmarking data chains had 200 samples each, the 20 Newsgroups chains had 750 samples each, and the Enron chains had 7300 samples each. The three summarization methods were then used to cluster the data given these 300 MCMC chains. The number \(K\) of clusters in each experiment was chosen to be the same as the number of natural classes for each data set: 50 for the Social Bookmarking data set, 20 for the Newsgroups data set, and 32 for the Enron data set. The metrics for the 100 chains, across each data set and summarization method pair were averaged. The results are shown in Tables 1-3 and indicate that there is little difference in the performance of the summarization methods.

5.5 Comparison to EM Clustering

To compare the clusters produced by the collapsed Gibbs sampler to those produced by EM, an EM algorithm on the same model was used to cluster each of the three data sets 100 times from random starting points. The average values of the metrics computed on the partitionings produced by EM are compared with the averaged
metrics for the best summarizations produced by the Gibbs sampler in Tables 4-6. It can be seen that the metrics consistently indicate that the Gibbs sampling algorithm produces better clusterings than EM.

Table 3: Results comparing the performance of the 3 summarization techniques on 100 chains of 7300 samples each, produced using the Enron data set.

<table>
<thead>
<tr>
<th>Metric</th>
<th>Marginal</th>
<th>MAP</th>
<th>Rand</th>
</tr>
</thead>
<tbody>
<tr>
<td>F-Measure</td>
<td>0.35023</td>
<td>0.35100</td>
<td>0.34989</td>
</tr>
<tr>
<td>VI</td>
<td>3.54033</td>
<td>3.49994</td>
<td>3.50624</td>
</tr>
<tr>
<td>ARI</td>
<td>0.13409</td>
<td>0.13564</td>
<td>0.13461</td>
</tr>
<tr>
<td>V-Measure</td>
<td>0.32366</td>
<td>0.32348</td>
<td>0.32322</td>
</tr>
<tr>
<td>Q2</td>
<td>0.71328</td>
<td>0.71315</td>
<td>0.71309</td>
</tr>
</tbody>
</table>

Table 4: Results comparing the performance of the best configuration of the Gibbs sampling clustering algorithm to the performance of an EM clustering algorithm on the del.icio.us data.

<table>
<thead>
<tr>
<th>Metric</th>
<th>Gibbs</th>
<th>EM</th>
</tr>
</thead>
<tbody>
<tr>
<td>F-Measure</td>
<td>0.38591</td>
<td>0.30536</td>
</tr>
<tr>
<td>VI</td>
<td>3.74822</td>
<td>4.08036</td>
</tr>
<tr>
<td>ARI</td>
<td>0.22803</td>
<td>0.16483</td>
</tr>
<tr>
<td>V-Measure</td>
<td>0.47867</td>
<td>0.41225</td>
</tr>
<tr>
<td>Q2</td>
<td>0.72949</td>
<td>0.69472</td>
</tr>
</tbody>
</table>

Table 5: Results comparing the Gibbs clustering algorithm to the EM clustering algorithm on the 20 Newsgroups data.

<table>
<thead>
<tr>
<th>Metric</th>
<th>Gibbs</th>
<th>EM</th>
</tr>
</thead>
<tbody>
<tr>
<td>F-Measure</td>
<td>0.42174</td>
<td>0.33012</td>
</tr>
<tr>
<td>VI</td>
<td>2.14728</td>
<td>2.60149</td>
</tr>
<tr>
<td>ARI</td>
<td>0.27730</td>
<td>0.20405</td>
</tr>
<tr>
<td>V-Measure</td>
<td>0.56037</td>
<td>0.47627</td>
</tr>
<tr>
<td>Q2</td>
<td>0.73058</td>
<td>0.68064</td>
</tr>
</tbody>
</table>

Table 6: Results comparing the Gibbs sampling clustering algorithm to the EM clustering algorithm on the Enron data.

<table>
<thead>
<tr>
<th>Metric</th>
<th>Gibbs</th>
<th>EM</th>
</tr>
</thead>
<tbody>
<tr>
<td>F-Measure</td>
<td>0.35100</td>
<td>0.30351</td>
</tr>
<tr>
<td>VI</td>
<td>3.49994</td>
<td>3.86434</td>
</tr>
<tr>
<td>ARI</td>
<td>0.13564</td>
<td>0.09739</td>
</tr>
<tr>
<td>V-Measure</td>
<td>0.32366</td>
<td>0.27483</td>
</tr>
<tr>
<td>Q2</td>
<td>0.71328</td>
<td>0.68704</td>
</tr>
</tbody>
</table>

Table 7: Example contingency table showing the relationship of natural classes to clusters produced by the collapsed Gibbs sampler, using the 100th sample as the summary for the 20 Newsgroups data.

<table>
<thead>
<tr>
<th>Source</th>
<th>1</th>
<th>2</th>
<th>3</th>
<th>4</th>
<th>5</th>
<th>6</th>
<th>7</th>
</tr>
</thead>
<tbody>
<tr>
<td>comp.graphics</td>
<td>0</td>
<td>1</td>
<td>2</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>comp.os.ms-windows.misc</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>2</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>comp.sys.ibm.zSeries</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>2</td>
<td>0</td>
</tr>
<tr>
<td>comp.sys.mac.hardware</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>2</td>
<td>0</td>
</tr>
<tr>
<td>misc.literature</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>2</td>
<td>0</td>
</tr>
<tr>
<td>misc.forsale</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>misc.inet</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>rec.autos</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>rec.sport.baseball</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>rec.sport.hockey</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>sci.crypt</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>sci.electronics</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>sci.med</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>sci.space</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>soc.religion.recreation</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>talk.politics.guns</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>talk.politics.military</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>talk.politics.iran</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>talk.religion.misc</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
</tbody>
</table>

5.7 Qualitative Confirmation

The external metrics used here attempt to measure the similarity between partitionings of the data. While they do tend to be useful for comparing clustering algorithm variations on the same data set, there is no way to know whether a clustering is actually good based solely on these numbers. The fact that the Gibbs sampler described here produces clusterings with better external metrics than the EM algorithm does not necessarily mean that these clusterings make sense. It could be the case that the EM algorithm is producing very poor clusterings, and the Gibbs algorithm is producing only slightly less poor clusterings.

In order to verify that the clusterings produced are sensible, we examined contingency tables for several runs of the Gibbs sampling algorithm on the various data sets. For the most part, the algorithm appears to make sensible partitionings of the data, though they are
not necessarily the same as those present in the reference partition.

A randomly selected contingency table is shown in Table 7. The clustering contains many empty or small clusters, such as clusters 1, and 5. We have culled the majority of these very small clusters from the table. The bulk of the data have been placed into seven clusters: 2, 3, 4, 8, 9, 12, and 16. These clusters, for the most part, make a great deal of sense. For example, two of the clusters are fairly pure; cluster 2 is mostly about cryptography, and cluster 12 is mostly about medical issues. The other clusters consist of intuitive groupings of the remaining classes. Cluster 3 has to do with automotive and electronic hardware. Cluster 4 has to do with politics and sports. Cluster 8 deals mostly with computer-related topics. Cluster 9 is about space exploration and the middle-east and Cluster 16 is about religion.

Another interesting way to view the contingency table is to examine how the classes end up distributed across the clusters. For example, the misc.forsale and sci.electronics documents are almost evenly split between the automotive and computer clusters. This makes sense because computer equipment and automobiles are the most likely products to be sold on newsgroups, and these are the types of forums where discussions of electronics would be most prevalent. Also interesting is the way that the talk.politics.mideast class is divided into a politics cluster, a religious cluster and a cluster about space and the middle east.

In general these trends indicate that the partitions are of good quality, perhaps even better than the metric scores would suggest. Although the cluster labels do not match the gold-standard perfectly, the clusters appear coherent and mostly correspond to a valid way of organizing the documents, one that a human annotator might settle on, if the newsgroup labels were withheld.

6. CONCLUSION

The experiments show a great deal of promise for using MCMC methods over EM for clustering documents with a mixture of multinomials model.

The absence of label switching shows that summarization methods that use more than one sample in their summary (like the marginal posterior method presented here) can be quite effective in this domain. This simplifies clustering algorithms that use MCMC on a mixture of multinomials for clustering. However, it does indicate that the sampler is not sampling from the entire distribution, as that would necessarily involve a certain amount of label switching. This is most likely because of the large number of variables present in the model, together with a relatively high amount of correlation between those variables. This failure to explore other modes would not be acceptable for applications for which true estimates of the posterior distributions are needed. In the case of document clustering, however, the true objective is to maximize the quality of the partitions produced by the algorithm, not to maximize the accuracy of the posterior distribution estimates.

It appears that the marginal posterior summarization method was generally superior to the two single-sample summarization techniques, except in the case of the Enron data set where the results were mixed, as can be seen in Table 3. It is possible that this is due to the smaller number of documents in the Enron data. The resulting longer sample chains might have allowed the sampler to find MAP samples that were closer to the true MAP values. Although not optimal, much simpler summarization techniques yield partitionings of comparable quality.

In summary, we have found the following guidelines that practitioners may consult as they create their own sampler-based clustering algorithms:

1. The point at which likelihood plots approach an asymptote corresponds with the convergence of the chain to steady-state.
2. Little or no label-switching occurs with our model and data. The methodology from Section 5.3 can be used to verify the presence or absence of label switching.
3. Once samples have been collected, there is little difference in the performance of reasonable summarization methods.
4. The Gibbs sampling algorithm presented here consistently produces better clusterings than EM on the same model.
5. Longer chains, together with either the marginal posterior, or MAP summarization methods produce the best results. However, much simpler strategies yield competitive results.

7. FUTURE WORK

We were able to compare MCMC and EM in their more basic forms, but it would be informative to evaluate how other variants of these algorithms that are used in practice (e.g. variational EM, and EM combined with local search techniques [5]) perform in comparison.

In addition, while it is likely that the results presented here generalize to a broad class of situations where similar models are used on similarly sized data sets, it is important to evaluate how well they extend to more complex models, such as LDA.

Scalability is also an issue that needs to be addressed, as some real-world data sets can be orders of magnitude larger than those used for experimentation here. It is unknown whether the results shown here scale to such large data sets.

Finally, an approach that combines EM and MCMC might perform better than either technique in isolation. One approach would be to first collect a limited number of samples using an MCMC sampler, and then to use a summary of these samples as a starting point for EM. This approach would leverage the exploratory power of a sampling-based approach to find a more promising starting configuration to begin maximizing with EM. That is, instead of choosing a random hill to climb, MCMC helps find a hill that’s likely to be one of the higher hills to climb.

8. ACKNOWLEDGMENTS

We would like to thank Dr. Shane Reese and Robbie Haertel for their feedback and help related to the techniques and mathematical derivations presented in this paper.

9. REFERENCES


APPENDIX

A. DERIVATION OF COMPLETE CONDITIONAL

The derivation uses the following variables:

<table>
<thead>
<tr>
<th>Symbol</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>$M$</td>
<td>Number of documents in data set</td>
</tr>
<tr>
<td>$N_d$</td>
<td>Length of document $d$</td>
</tr>
<tr>
<td>$\theta_d$</td>
<td>The $z^\text{th}$ column of $\theta$</td>
</tr>
<tr>
<td>$w_{d:n}$</td>
<td>The $n^\text{th}$ word of document $d$</td>
</tr>
<tr>
<td>$V$</td>
<td>the vocabulary</td>
</tr>
<tr>
<td>$K$</td>
<td>The number of clusters/mixture components</td>
</tr>
<tr>
<td>$n_k$</td>
<td>$\sum_d \delta(z_d,k)$, the number of documents labeled $k$.</td>
</tr>
<tr>
<td>$n_{kv}$</td>
<td>$\sum_d \sum_n \delta(z_d,k) \cdot \delta(w_{d:n},v)$, the number of times $v$ occurs in documents with the label $k$.</td>
</tr>
</tbody>
</table>

We begin with the full joint distribution over the data, latent variables, and model parameters, given our fixed priors.

$$p(\lambda, z, w, \theta | \alpha, \beta) = p(\lambda | \alpha) \prod_{k=1}^{K} p(\theta_k | \beta) \prod_{d=1}^{M} \left( p(z_d | \lambda) \prod_{n=1}^{N_d} p(w_{d:n} | z_d, \theta z_d) \right)$$  \hspace{1cm} (A-1)

Next, integrate out $\lambda$ and $\theta$.

$$p(z, w | \alpha, \beta) = \int \ldots \int p(\lambda | \alpha) \prod_{k=1}^{K} p(\theta_k | \beta) \left( \prod_{d=1}^{M} p(z_d | \lambda) \right) \prod_{n=1}^{N_d} p(w_{d:n} | z_d, \theta z_d) \, d\lambda, d\theta_1, \ldots, d\theta_K$$ \hspace{1cm} (A-2)

Expand the nested product

$$= \int \ldots \int p(\lambda | \alpha) \prod_{k=1}^{K} p(\theta_k | \beta) \left( \prod_{d=1}^{M} p(z_d | \lambda) \right) \prod_{n=1}^{N_d} p(w_{d:n} | z_d, \theta z_d) \, d\lambda \prod_{1}^{K} \prod_{d=1}^{M} \prod_{n=1}^{N_d} p(w_{d:n} | z_d, \theta z_d) \, d\theta_1, \ldots, d\theta_K$$ \hspace{1cm} (A-3)

and then group like terms

$$= \int p(\lambda | \alpha) \prod_{d=1}^{M} p(z_d | \lambda) \, d\lambda \int \ldots \int \prod_{k=1}^{K} \prod_{d=1}^{M} \prod_{n=1}^{N_d} p(w_{d:n} | z_d, \theta z_d) \, d\theta_1, \ldots, d\theta_K$$ \hspace{1cm} (A-4)

Expand the multinomial and Dirichlet distributions

$$= \int B^{-1}(\alpha) \prod_{j=1}^{V} \lambda_j^{\alpha_j-1} \prod_{d=1}^{M} \lambda_{z_d} \, d\lambda \int \ldots \int \prod_{k=1}^{K} \left( B^{-1}(\beta) \prod_{v=1}^{V} \theta_{kv}^{\beta_{kv}-1} \right) \prod_{d=1}^{M} \prod_{n=1}^{N_d} \theta_{w_{d:n}} \, d\theta_1, \ldots, d\theta_K$$ \hspace{1cm} (A-5)

Now, change the product indices from products over documents and word sequences, to products over cluster labels and token counts

$$= \int B^{-1}(\alpha) \prod_{j=1}^{V} \lambda_j^{\alpha_j-1} \prod_{k=1}^{K} \lambda_{k}^{n_k} \, d\lambda \int \ldots \int \prod_{k=1}^{K} \left( B^{-1}(\beta) \prod_{v=1}^{V} \theta_{kv}^{\beta_{kv}-1} \right) \prod_{k=1}^{K} \prod_{v=1}^{V} \theta_{kv}^{n_{kv}} \, d\theta_1, \ldots, d\theta_K$$ \hspace{1cm} (A-6)

this is possible because $\prod_{d=1}^{M} \lambda_{z_d} = \prod_{k=1}^{K} \lambda_{k}$ and $\prod_{n=1}^{N_d} \theta_{w_{d:n}} = \prod_{v=1}^{V} \theta_{kv}^{n_{kv}}$, because they are products over the same quantities, in different orders.

Next, simplify by combining products, adding exponents and pulling constant multipliers outside of integrals.

$$= B^{-1}(\alpha) \prod_{j=1}^{K} \lambda_j^{\alpha_j+n_k-1} \, d\lambda \cdot B^{-K}(\beta) \int \ldots \int \prod_{k=1}^{K} \prod_{v=1}^{V} \theta_{kv}^{\beta_{kv}+n_{kv}-1} \, d\theta_1, \ldots, d\theta_K$$ \hspace{1cm} (A-7)

At this point we have integrals over terms that are in the form of the kernel of the Dirichlet distribution, we can therefore complete the integrals, leaving

$$p(z, w | \alpha, \beta) = B^{-1}(\alpha) B(\alpha_*) \cdot B^{-K}(\beta) \prod_{k=1}^{K} B(\beta_{k*})$$ \hspace{1cm} (A-8)

Where $\alpha_*$ and $\beta_{k*}$ are defined in Section 4. Equation 3 is directly derived from Equation A-8.