2009-12-09

Bisecting Document Clustering Using Model-Based Methods

Aaron Samuel Davis
Brigham Young University - Provo

Bisecting Document Clustering Using Model-Based Methods

Aaron S. Davis

A thesis submitted to the faculty of
Brigham Young University
in partial fulfillment of the requirements for the degree of

Master of Science

Eric K. Ringger, Chair
Christophe Giraud-Carrier
Robert P. Burton

Department of Computer Science

Brigham Young University

April 2010

ABSTRACT

Bisecting Document Clustering Using Model-Based Methods

Aaron S. Davis

Department of Computer Science

Master of Science

We all have access to large collections of digital text documents, which are useful only if we can make sense of them all and distill important information from them. Good document clustering algorithms that organize such information automatically in meaningful ways can make a difference in how effective we are at using that information. In this paper we use model-based document clustering algorithms as a base for bisecting methods in order to identify increasingly cohesive clusters from larger, more diverse clusters. We specifically use the EM algorithm and Gibbs Sampling on a mixture of multinomials as the base clustering algorithms on three data sets. Additionally, we apply a refinement step, using EM, to the final output of each clustering technique. Our results show improved agreement with human annotated document classes when compared to the existing base clustering algorithms, with marked improvement in two out of three data sets.

Keywords: Document Clustering, Text Mining, Model-based

ACKNOWLEDGMENTS

I have so many people to thank for their kind contributions to this thesis. I received family support, which is the best kind, from my wife and four boys. I was directed by a great mentor, Dr. Eric Ringger, who did not give up on me and helped me create a cohesive pattern of research. Each of the members on my committee contributed to a focused and complete thesis with their feedback. There are also professors and teaching assistants who helped me to create the base I would need to do my research.

Besides all these individuals, I have had the good fortune of associating with some great classmates. I could not name them all so I will name just a few. Thank you to Nathan Davis, Daniel Walker, Matthew Smith and Brian Hawkins.

Finally, I would like to acknowledge the Marylou Fulton Supercomputing Center at Brigham Young University. This center has provided a very valuable resource to perform many thousands of long running experiments.

Contents

List of Figures

List of Tables

Chapter 1

Introduction

Given the already enormous and ever increasing number of digital documents, finding ways to provide access to so much information is a valuable pursuit. Clustering documents in large collections is one way to organize information into a more accessible form in an efficient manner. The desired output of a document clustering algorithm is a partition of documents into sets of similar documents. Such a clustering can be used in multiple applications, including the discovery of interesting patterns in document collections (Berkhin, 2006), simplifying browsing and facilitating information retrieval (Grossman, 2004). In each application, the user seeks information without expending money and time to have people read individual documents and summarize their findings, and document clustering facilitates the process.

Patterns in data help a user make sense of the data. Document clustering can be used in discovering patterns such as correlations among topics, authors, and documents (Hearst, 1999). These patterns can then be used directly or as a precursor in areas such as document classification by bootstrapping a training set which does not contain sufficient amounts of labeled data (Nigam *et al.*, 2000). Regardless of the application, the patterns discovered give structure to an otherwise unstructured set of data.

The most direct application of document clustering is browsing by humans. In this scenario, the document clustering output is consulted by a human without a specific query or pattern in mind. Typically one or more document clusters will pique their interest and result in further exploration. This application of document clustering is important because it represents a model of direct usage rather than only using document clustering as one piece in a pipeline of information discovery.

Document clustering serves two different purposes in the area of Information Retrieval. The first deals with efficiency in searching through a document set. By partitioning a document collection the size of the search space can be reduced from the entire document collection to related documents. This is especially important because of the correlation between the time to return query results and the size of the search space. The second purpose in Information Retrieval is related to the results of a query. In this case, document clustering is used as a refinement process where search results can be clustered and returned to a user. The user can then retrieve new results by selecting the cluster which is most closely related to the intent of the query (Hearst & Pedersen, 1996).

1.1 Overview

In some cases document clustering is hierarchical which leads to additional methods for browsing. Hierarchical document clustering is a very common technique for document clustering which generates high quality clusters (Zhao *et al.*, 2005). In addition to the clusters themselves, hierarchical document clustering creates a taxonomy for documents in a data set. This taxonomy can be represented as a dendrogram as seen in Figure 1.1 .

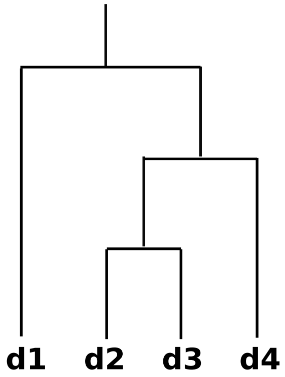


Figure 1.1: A dendrogram showing the progression from one cluster to four documents.

The two types of hierarchical clustering are agglomerative and divisive. In hierarchical agglomerative clustering, each document starts in its own cluster and clusters are combined iteratively based on similarity to each other. Divisive clustering begins with all documents combined into one cluster and iteratively breaks clusters apart. While it is natural to combine documents, hierarchical divisive clustering has been shown to produce better clusters than hierarchical agglomerative clustering (Steinbach *et al.*, 2000; Boley *et al.*, 1999). As an additional benefit, divisive clustering produces a taxonomy which increases in granularity. This matches the common browsing pattern of "drilling down" from broader concepts to more specific ones.

This work focuses on the hierarchical divisive technique where each division is fixed at two clusters. For simplicity we refer to this as bisecting throughout this work. The document clustering algorithms we propose focus on bisecting but that does not describe how to create the bisection of documents. We will refer to the clustering algorithms that bisect the document collection as base algorithms. As base algorithms we propose using a class of clustering methods commonly called model-based.

Model-based clustering works on the assumption that data is generated from a model and that the data can be used to estimate the parameters of such a model. Model-based clustering is an active area of research, producing high quality clusters (Meilă & Heckerman, 1998). The increasing appeal of statistical methods for various problems involving uncertainty is driving the improvement of model-based clustering methods and introducing alternative models which focus on other objectives. For example, topic models such as Latent Dirichlet Allocation and its variants generate word clusters representing a particular topic inside of a document (Blei *et al.*, 2003).

The effectiveness of a document clustering algorithm is a complex subject. The desired output of a document clustering algorithm is a partition of documents into sets containing similar documents. The similarity of a set of documents can be measured directly or indirectly depending on the availability of labels and the clustering objective. In order to understand if a document clustering algorithm is creating a proper partition of documents, metrics must be established. Due to the subjective nature of similarity, it is difficult to find a metric for all possible clustering objectives. Consequently, it is important to establish what metrics will be used and how the metrics are derived in order to understand how the documents of a document partition are similar.

1.2 Outline

In this work, we show that using model-based methods as a base for the hierarchical divisive clustering technique of bisecting increases the quality of document clusters over independent model-based methods. We do this by running model-based algorithms with the number of clusters set to the number of human labeled classes and running bisecting variants of these algorithms for comparison. The remainder of the thesis is organized as follows: in Chapter 2, we cover prior research related to hierarchical, bisecting and model-based document clustering. In Chapter 3 we discuss the metrics used to evaluate document clusters. In Chapter 4 we describe the process of bisecting using model-based algorithms. In Chapter 5 we show the results of our experiments. In Chapter 6 we examine the qualitative aspects of our experiments and in Chapter 7 we conclude.

Chapter 2

Related Work

Prior work in document clustering records a progression of improvement on certain metrics of interest. This progression shows that model-based text clustering is often superior to vector-based and more traditional methods (King, 1967; Sneath & Sokal, 1973; Jain & Dubes, 1988). The methods we introduce in this work are natural extensions of the work on model-based clustering.

2.1 Hierarchical Agglomerative Clustering

Hierarchical Agglomerative Clustering (HAC) is a well-known and highly-intuitive clustering method which has been in use for some time (Jain & Dubes, 1988). HAC starts with each document in its own cluster and proceeds by merging two clusters at each iteration of the algorithm. Which two clusters to merge is determined using a distance metric and a merging method (such as “single link”). The goal is to select for merging the two clusters which are most similar.

2.2 K-means

K-means can be implemented using either a vector-based algorithm or a model-based algorithm, and a variety of distance measures can be employed with the vector-based K-means algorithm. The algorithm begins with an initial set of cluster centroids. All documents are then assigned to the nearest centroid using the chosen distance measure. Next, each centroid is moved to better fit/explain the assigned data. These assignment and centroid adjustment steps are repeated in greedy fashion until the centroids no longer change, up to some tolerance. K-means has been found to produce inferior document clusters to HAC (Steinbach *et al.*, 2000; Jain & Dubes, 1988). However, the linear run-time of K-means is often an acceptable trade-off for the loss in accuracy.

2.3 Bisecting K-means

Steinbach et al. proposed a variation of K-means which scores better than HAC or K-means (Steinbach *et al.*, 2000) and that runs in linear time. This algorithm consists of an outer bisecting loop with K-means as the base clustering algorithm. The outer bisecting loop contains the criterion for selecting which cluster to split and repeats until the pre-determined number of clusters is reached. The inner bisecting loop performs the base clustering algorithm on the cluster selected in the outer loop with the desired number of sub-clusters equal to two. Inspired by this work, we explore the bisecting approach in greater detail in the next section and throughout the thesis. The quality improvement over HAC showed that K-means can be employed for higher quality clustering as well as faster asymptotic performance.

2.4 Mixture of Multinomials Model

The mixture of multinomials model is a probabilistic, generative model for clustering in which a single topic or class is assigned to each document. In addition, the model includes one variable for each document feature (i.e., word). We assume that the words are conditionally independent of each other, given the class. Documents can thus be represented as word vectors.

In this work, we follow the common assumption that words are distributed according to the multinomial event model. Particularly in the clustering setting, this model is known as a mixture of multinomials. This model has proven to be accurate for classifying documents, which suggests that it is a reasonable representation of a document for our purposes (McCallum & Nigam, 1998). By extension, if the parameters of this model could be learned in an unsupervised manner one would also expect high-quality clusters. A graphical model representing the mixture of multinomials model with its parameters made explicit is shown in Figure 2.1 .

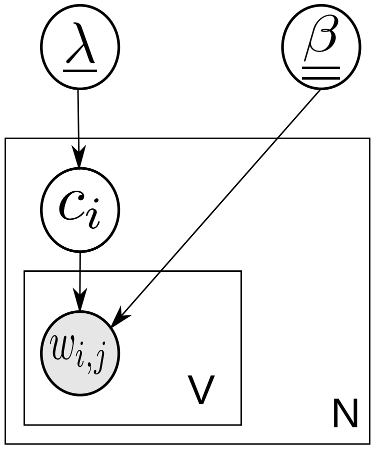


Figure 2.1: Graphical model representing the mixture of multinomials model.

In this graphical model we use plate notation to represent multiple instances of a node denoted by a variable present inside the plate. The random variables in the model are represented by c_i and $w_{i,j}$. These random variables are generated from parameterized distributions where $\underline{\lambda}$ and $\underline{\underline{\beta}}$ represent the parameters for the corresponding distributions: $c_i \sim$ Categorical($\underline{\lambda}$) and $w_{i,j} \sim$ Multinomial($\underline{\underline{\beta}}$). This mixture of multinomials model provides the basis for each of the model-based clustering algorithms explored in this work.

The mixture of multinomials is commonly used in model-based document clustering even though recent research has introduced additional models which are shown to yield superior results on several data sets (Zhong & Ghosh, 2005; Elkan, 2006). Zhong & Ghosh use the von Mises-Fisher model and Elkan uses the extended DCM (EDCM) model. These new models are foundationally similar to the mixture of multinomials but account for areas of weakness in the mixture of multinomials model with regard to document clustering. For example, Elkan explains in his work how the DCM and the EDCM model account for burstiness of words which is the increased probability of a word to be seen again once observed in a document. We assume that these models would improve the results of bisecting model-based techniques and include them in our discussion of future work.

2.5 Expectation Maximization

In the model-based document clustering of this work, we employ the mixture of multinomials model. Because the document set does not contain complete data (in fact, it is entirely unlabeled), the well-known Expectation-Maximization (EM) algorithm can be used to find maximum likelihood estimates for the parameters of the model (Dempster *et al.*, 1977; Meilă & Heckerman, 1998). After learning estimates of the model parameters, the model can be consulted for the posterior probability of any class (i.e., cluster i.d.) for a given document (i.e, set of words comprising the document). The class with the highest posterior probability is chosen as the cluster i.d. for the given document.

EM begins by initializing model parameters. Initial parameters can be either drawn randomly (from a Dirichlet distribution) or computed in a data-dependent manner. Prior work has shown that results from random initialization are roughly on par with data-driven initialization (Meilă & Heckerman, 1998). We will use random initialization as it is simpler to implement and minimizes run-time. As an iterative algorithm, EM begins with the expectation (E) step which, in practical terms, computes a posterior distribution over the class variable for each document using the given model parameters. The posteriors from the E-step can be thought of as fractional labels assigned to each document. The M-step uses the new complete data to learn a better model (i.e., re-estimate the model parameters) so as to improve the likelihood of the data. This process continues until the change in data likelihood is within a pre-specified threshold. This maximization process is a hill-climbing procedure on the (data) likelihood surface; hence, the EM algorithm is guaranteed to converge to a maximum, although the maximum may not be a global maximum (Meilă & Heckerman, 1998).

The problem of EM converging on local maxima depending on initialization suggests multiple-restarts of EM from multiple starting points. Recently deterministic annealing EM has been used successfully in document clustering (Elkan, 2006). While deterministic annealing EM does not guarantee convergence to a global maximum, it does ensure convergence to better local maxima than traditional EM. We have not included experiments using deterministic annealing in

EM but do include it in our discussion of future work. In this work we employ a brute force method of restarting EM from different randomly selected initialization points in order to approximate the more sophisticated deterministic annealing method.

2.6 Gibbs Sampling

Another common technique for estimating the parameters of our mixture of multinomials model employs sampling, namely Markov Chain Monte Carlo (MCMC) algorithms. Gibbs Sampling is one such algorithm and has proven to converge quickly (in distribution) on the mixture of multinomials model and to find high-quality document clusters (Walker & Ringger, 2008).

In each of our model-based algorithms there is a likelihood surface which represents the likelihood that the data was generated from the model. Unlike EM, Gibbs Sampling does not climb a maximum in the likelihood surface but can better explore the likelihood surface. This difference is due to the nature of sampling which does not follow an established pattern of taking a small step towards the nearest maximum. Instead, sampling tends to be drawn from the range of maximum likelihood. After a relatively short amount of time, Gibbs Sampling has been shown to settle into a good approximation of the true posterior distribution over cluster assignments and model parameters.

2.7 Bisecting Using Model-based Techinques

Any clustering algorithm which can produce two clusters from a data set can be used a a base algorithm. In a Brigham Young University class project from 2007, we showed that EM performed better than Bisecting K-Means over a range of tasks. As a result, in this work we propose using model-based methods as base algorithms to bisecting techniques. By doing so we intend to show that bisecting can be applied effectively as an enhancement to the more effective model-based algorithms.

Chapter 3

Metrics

In this section we provide information about the metrics used to assess the quality of clustering algorithms. Corresponding formulas for each of the metrics are included. Our focus is on External Metrics which are explained in this section. However, we also include the Internal Metrics which are reported in our empirical evaluation.

When evaluating clusters there are a number of metrics which represent different characteristics of a clustering. By reporting a number of different metrics in this thesis, we hope to reinforce the improvements made through bisecting techniques and clearly identify areas where additional improvement can be made.

3.1 External Metrics

External metrics are a type of metric which can be computed with the use of class labels, when available for a data set. When using class labels to calculate external metrics the documents contained in a cluster are examined to see how closely they match the documents contained in the labeled class. The five external metrics we employ in this thesis include: F-Measure, Variation of Information(VI), Adjusted Rand Index(ARI), V-Measure and Q_2.

- **F-Measure** A combined measure of precision and recall used frequently in information retrieval. F-Measure, is a good indication of the effectiveness of a clustering method to reproduce human-labeled classes (Larsen & Aone, 1999; Steinbach *et al.*, 2000). We iterate through classes (i) and clusters (j) and use the corresponding document counts (n) as follows:

$$Precision(i,j) = \frac{n_{ij}}{n_j}$$

$$Recall(i,j) = \frac{n_{ij}}{n_i}$$

$$F(i,j) = \frac{2 \times Precision(i,j) \times Recall(i,j)}{Precision(i,j) + Recall(i,j)}$$

$$\text{F-Measure} = \textstyle\sum_i \frac{n_i}{n} max\{F(i,j)\}$$

- **Variation of Information (VI)** This measure is information theory based. VI measures the distance between two proposed clusterings. The overall idea is to compare clustering variations on the same data set to determine information loss or gain (Meilă, 2002).

$$P(k) = \frac{n_k}{n}$$

$$P(k,k') = \frac{|C_k \cap C'_{k'}|}{n}$$

$$H(C) = -\sum_{k=1}^{K} P(k) log P(k)$$

$$I(C,C') = \sum_{k=1}^{K} \sum_{k'=1}^{K} P(k,k') log \frac{P(k,k')}{P(k)P(k')}$$

$$\text{VI}(C,C') = H(C) + H(C') - 2I(C,C')$$

- **Adjusted Rand Index (ARI)** This measure is the similarity of two independent partitions of the data (Yeung *et al.*, 2001). We iterate through classes (i) and clusters (j) and use the corresponding document counts (n) as follows:

$$\text{ARI} = \frac{\sum_{i,j}\binom{n_{i,j}}{2} - [\sum_i \binom{n_i}{2} \sum_j \binom{n_j}{2}]/\binom{n}{2}}{\frac{1}{2}[\sum_i \binom{n_i}{2} + \sum_j \binom{n_j}{2}] - [\sum_i \binom{n_i}{2} \sum_j \binom{n_j}{2}]/\binom{n}{2}}$$

- **V-Measure** This measure uses entropy from information theory to measure homogeneity and completeness. Homogeneity and completeness are combined as a weighted harmonic mean to produce V-Measure (Rosenberg & Hirschberg, 2007). We represent homogeneity as h and completeness as c.

$$H(C|K) = -\sum_{j=1}^{|K|}\sum_{i=1}^{|C|}\frac{n_{i,j}}{N} log \frac{n_{i,j}}{\sum_{i=1}^{|C|} n_{i,j}}$$

$$H(C) = -\sum_{i=1}^{|C|}\frac{\sum_{j=1}^{|K|} n_{i,j}}{n} log \frac{\sum_{j=1}^{|K|} n_{i,j}}{n}$$

$$H(K|C) = -\sum_{j=1}^{|K|}\sum_{i=1}^{|C|}\frac{n_{i,j}}{N} log \frac{n_{i,j}}{\sum_{j=1}^{|K|} n_{i,j}}$$

$$H(K) = -\sum_{j=1}^{|K|}\frac{\sum_{i=1}^{|C|} n_{i,j}}{n} log \frac{\sum_{i=1}^{|C|} n_{i,j}}{n}$$

$$h = \begin{cases} 1 & \text{if } H(C,K) = 0 \\ 1 - \frac{H(C|K)}{H(C)} & \text{else} \end{cases}$$

$$c = \begin{cases} 1 & \text{if } H(K,C) = 0 \\ 1 - \frac{H(K|C)}{H(K)} & \text{else} \end{cases}$$

$$\text{V-Measure} = \frac{2 \times h \times c}{h + c}$$

- Q_2 This measure uses entropy from information theory to calculate homogeneity and the model cost (Dom, 2001). We represent homogeneity as h.

$$Q_0(C,K) = H(C|K) + \frac{1}{n}\sum_{j=1}^{|K|} log \binom{h(k) + |C| - 1}{|C| - 1}$$

$$Q_2(C,K) = \frac{\frac{1}{n}\sum_{i=1}^{|C|} log \binom{h(c)+|C|-1}{|C|-1}}{Q_0(C,K)}$$

3.2 Internal Metrics

Internal metrics refer to metrics which can be computed without class labels. Internal metrics are computed solely from the data which constitute a cluster. Two types of internal metrics which we report are described as follows:

- **Total Inter-Cluster Divergence (TD)** By defining K-L Divergence between two clusters, TD can be defined as the sum (or log-sum where probabilities exist in log space) of the K-L divergence measures for all cluster combinations. We define the K-L divergence between two clusters as:

  $$logKL(c_1||c_2) = -logSum_{w_i \varepsilon V_{c_1 c_2}} (\log P(w_i|c_1) + (\log P(w_i|c_1) - \log P(w_i|c_2))$$

  Where $V_{c_1 c_2}$ is the unique word set or vocabulary of cluster 1 and cluster 2, $V_{c_1 c_2}$ is defined as $V_{c_1 c_2} = V_{c_1} \cup V_{c_2}$. To avoid issues where one cluster contains a w_i that does not exist in the other cluster, add one smoothing is implemented for the probability distributions. This gives an overall K-L divergence for the distributions that describe two clusters.

- **Total Intra-Cluster Divergence (TIAD)** Total Intra-Cluster Divergence is computed by randomly splitting a single cluster in half, then computing the divergence between the two subclusters. We repeat this randomized splitting five times and report the average upon completion. Due to the randomized nature of this metric, the outcome is not deterministic. This metric measures the degree to which a cluster is unlike itself, on average. TIAD is measured for all clusters. TIAD is the sum (or log sum where applicable) of all the self divergence values for all clusters in the final clustering.

Internal metrics provide useful information, particularly when class labels do not exist, or where class labels do not necessarily correlate with a particular clustering motivation. However, because of the non-deterministic nature of TIAD and because TD is biased towards model-based techniques which use the probability distributions of features in their calculations, we report these metrics with some tentativeness.

In addition to these internal metrics we use the **Log-Likelihood of the Data (LD)**. We use LD as a diagnostic part of the bisecting technique. After performing a number of split attempts we must decide which cluster to keep. We use the LD for the resulting clusters to determine which bisection is the best candidate. To calculate the LD we sum the log-probability of the data for each of the classes. The log-probability of the data for a given class is the sum of the log-probabilities for each of the words that are an element of that class. This is shown in the following equation.

$$
\begin{aligned}
P(D) = \sum_k P(C=k,D) &= \sum_k P(D|C=k)P(C=k) \\
&= \sum_k \prod_{w_i \in C_k} P(w_i|k)P(k) \\
log(P(D)) &= logSum_k \sum_{w_i \in C_k} log(P(w_i|k)) + log(P(k))
\end{aligned}
$$

Chapter 4

Bisecting Document Clustering

Consistent with Steinbach et al., bisecting clustering is a divisive method in which smaller clusters are formed from larger clusters. Such methods start with one cluster and repeatedly divide existing clusters into smaller clusters until a predetermined number of clusters (or some other criterion) is reached. Inside the bisecting loop is a base clustering method which produces two sub-clusters from one of the existing clusters at each iteration. In this work we employ both EM and Gibbs Sampling as base clustering methods within the bisecting loop. The following steps illustrate the process:

1. Choose a cluster to split (In this work, we select the largest cluster.).
2. Run the base clustering algorithm to find 2 sub-clusters (Bisecting step).
3. Repeat step 2 for a pre-determined number of *split attempts*, saving the 2 sub-clusters which exhibit the best score on a specified internal metric (In this work, the score is the Log-Likelihood of the Data.).
4. Repeat cluster splitting (Steps 1–3) until the number of clusters equals a pre-specified number of clusters (in this work, we set this number to the number of labeled classes in a given data set).

Internal metrics are an intrinsic part of the bisecting loop (Step 3). In order to choose an appropriate bisection, only internal metrics can be used in clustering, because they do not rely on labeled data.

Figure 4.1 is an illustration of three iterations of bisecting. In this example, the largest cluster is chosen at each iteration. The dotted line represents the cluster split chosen by the base clustering algorithm with the number of clusters equal to two.

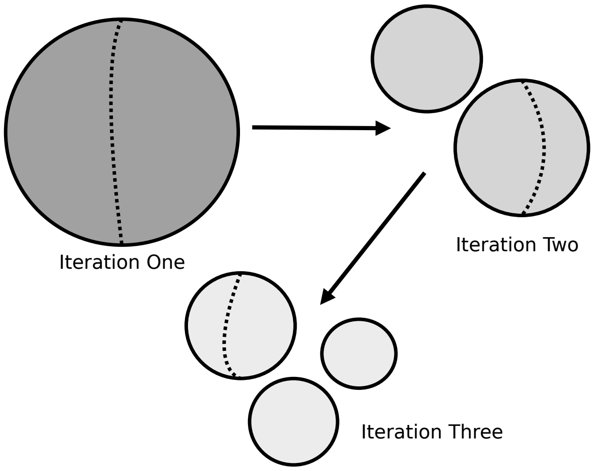


Figure 4.1: Three iterations of bisecting

4.1 Cluster Selection

Choosing the appropriate cluster for bisection is a challenging problem (Savaresi *et al.*, 2000). In our experiments we use a simple technique which has been explored previously: select the cluster containing the largest number of documents (Steinbach *et al.*, 2000; Zhao & Karypis, 2001). The largest cluster is likely to contain the greatest variety of documents. This is especially true in the early stages when the largest cluster is significantly larger than the other clusters. Future work could explore more complex techniques for this decision point.

After choosing one of the existing clusters to bisect, the base clustering algorithm is run on the data in the chosen cluster. This sub-clustering is fixed to find two clusters from the chosen cluster. The model-based clustering algorithms we have chosen, EM and Gibbs Sampling, have a degree of uncertainty to them: depending on where the model is initialized, EM may not find a global maximum. Also, like all MCMC techniques, Gibbs Sampling is guaranteed to converge to the true posterior distribution; however, there is no known way to know when a particular chain has converged to the true posterior distribution (Wasserman, 2003). We look for stability in the mixing plot of the likelihood after a pre-determined burn of 20 rounds. Burn is the sampling rounds which are discarded at the beginning of a sampling chain because the samples have not yet settled into a stable pattern.

4.2 Split Attempts

When using EM as a base clustering algorithm, the initialization directly influences which maximum will be reached. We use a random initialization of the parameters by sampling from a uniform Dirichlet distribution. Each split attempt is a run of EM to completion, beginning with a random initialization of the parameters. Increasing the number of split attempts improves the chances of reaching the global maximum but adds to the time required by the algorithm. Likewise for Gibbs sampling, each split attempt is a run of Gibbs to convergence (in distribution), beginning with a random initialization.

4.3 Refinement

Steinbach et al. further improve the results of Bisecting K-means by applying a post-clustering procedure. This clustering refinement is accomplished by using the cluster centroids from the final clustering of Bisecting K-means as initial centroids to the base K-means algorithm. Additionally, Steinbach et al. show that using base K-means to refine Bisecting K-means results in improvement on the chosen metrics. Although Steinbach et al. suggest a post-clustering refinement in their work, clustering with Bisecting K-means can also be considered an initialization method for the K-means algorithm itself.

In our experiments, we explore the degree to which bisecting model-based techniques might also show improvement through refinement. This is done with EM by taking the final clustering from Bisecting EM and using the clustering to determine initial counts in the base EM algorithm. More generally, the final clustering from any clustering algorithm could be used to initialize EM. While Gibbs sampling could also be used as a refinement algorithm, we choose to leave it for future work, focusing here on EM which is more official.

Chapter 5

Empirical Evaluation

A consistent experimentation methodology is applied to all tests in our evaluation. Each test was run on Dell Poweredge 1955s with two Quad-core 2.6GHz Intel Xeon processors and 16 GB of RAM. By using identical hardware and the same code base for the the standard and bisecting model-based techniques, we can approximate a fair comparison using run time.

We applied minimal feature selection to our experiments. We used a consistent stop word list, provided in Appendix B, for each of the three data sets and for each of the four algorithms. In addition we limited the number of features from each document by first computing a term frequency inverse document frequency score (tf-idf) for each word and then only keeping the top ten tf-idf scoring words for each document (Jones, 1972; Salton & Buckley, 1988; Manning & Schutze, 1999). To be clear, the method this work employs for computing tf-idf scores is as follows:

$$\mathrm{tf}_{\mathrm{i,j}} = \frac{t_{i,j}}{\sum_k t_{k,j}}$$

$$\mathrm{idf}_{\mathrm{i}} = \log \frac{|D|}{1 + |\{d : t_i \in d\}|}$$

$$(\text{tf-idf})_{\mathrm{i,j}} = \mathrm{tf}_{\mathrm{i,j}} \times \mathrm{idf}_{\mathrm{i}}$$

Where $t_{i,j}$ is the number of occurrences of term i in document j, $|D|$ is the total number of documents in the data set and $1 + |\{d : t_i \in d\}|$ is the number of documents where the term $t_{i,j} \neq 0$. We add one to $|\{d : t_i \in d\}|$ to ensure that we do not have a denominator of zero for terms that are not in the corpus. This is often called “add one smoothing.”

In order to visualize the results of our experiments we have included several graphs. We have chosen to focus on one external metric for our graphs and include additional external and internal metrics in our tables. By doing this we hope to clearly explain the outcome of our empirical evaluation without inundating the reader with data. Because all external metrics use class labels as the basis for their calculation, they generally track one another quite closely as we show in our tables. As F-Measure is a commonly used external metric, we will present data in our graphs using F-Measure.

Tables reporting the results for EM, Bisecting EM and Bisecting EM with refinement show the mean of five runs of Bisecting EM and the equivalent number of runs of EM as reported for each data set. In order to balance the number of random restarts with Bisecting EM and EM, run time was measured and equal run time was given to each algorithm. In addition to the mean, we report the standard deviation for each metric. This provides a clear idea of what to expect when running EM, Bisecting EM and Bisecting EM with refinement given the non-deterministic nature of EM based on initialization. We conclude with maximum and minimum values in a comprehensive graph of data sets and algorithms using the widely used F-Measure metric.

5.1 Data Collections

All experiments were performed on each of three data collections: 20 Newsgroups (Joachims, 1997), Annotated Enron Email (Berry *et al.*, 2007) and del.icio.us (Walker & Ringger, 2008). These data collections have one class label per document and are used as a gold standard for evaluation by external metrics. The class labels are not consulted during the clustering phase in order to properly execute unsupervised learning. We have included Table 5.1 which describes the characteristics of each of the data sets.

5.2 Bisecting EM

In order to begin an evaluation of Bisecting EM we must evaluate the effect of increasing the number of split attempts. We begin with two split attempts and increase to 150 split attempts for each

Data Set	Documents	Tokens	Summary	Classes
Enron	4,936	29,812	Subpoenaed e-mail from Enron employees	32
20 Newsgroups	19,997	43,627	Online newsgroup postings	20
del.icio.us	21,627	71,548	Web pages bookmarked by users of social bookmarking tool del.icio.us	50

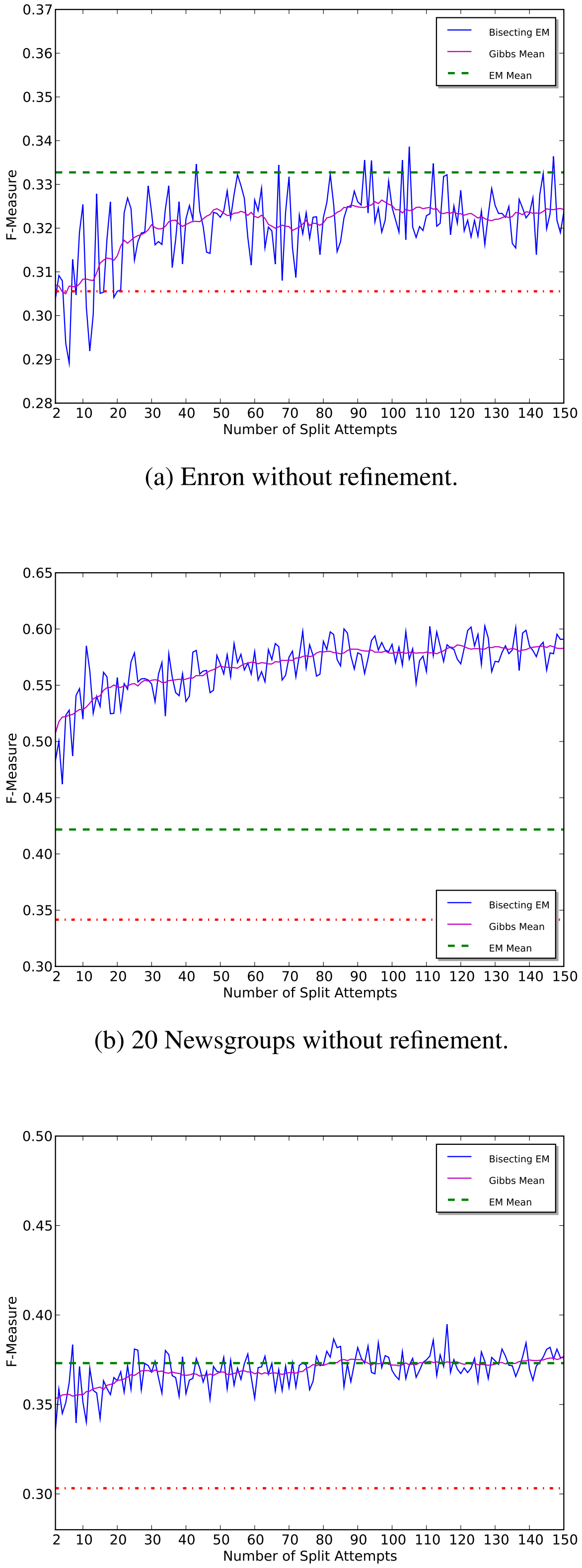
Table 5.1: Descriptions of each data set used for empirical evaluation.

of the three data collections. While simple to run each split attempt in parallel, we run each split attempt serially and accumulate the time it takes to complete the specified number of split attempts. Total clustering time for Bisecting EM will be used when determining how many iterations of regular EM to run. This gives regular EM an opportunity to overcome a poor initialization just as multiple split attempts do for Bisecting EM. Figure 5.1 shows the effect of increasing the number of split attempts on F-Measure for each of the three data sets.

It is important to note that F-Measure is not used to determine which split attempt to keep. This would unfairly skew the final results and be considered “cheating”. The smoothed average shown in Figure 5.1 is computed by a moving average of width nine across the scatter plot for Bisecting EM.

There are two additional lines in Figure 5.1 . These represent the mean of multiple runs of EM as specified in the “EM iter” column of Table 5.2 and five runs of Gibbs Sampling. They are included as a point of reference to see how Bisecting EM performs against the base model-based algorithms.

We examine the smoothed average to select the number of split attempts which is most likely to produce a good result. This is done by looking for a “knee” in the smoothed average. By searching for the point where additional split attempts is least likely to increase F-Measure, we reduce the processing power and run time of Bisecting EM. Table 5.2 gives the selected number of split attempts.



(a) Enron without refinement.

(b) 20 Newsgroups without refinement.

(c) del.icio.us without refinement.

Figure 5.1: Sweeping the number of split attempts using Bisecting EM

Data Set	Split Attempts	Run Time (seconds)	EM iter
Enron	90	3,298	16
20 Newsgroups	82	11,456	34
del.icio.us	86	52,769	54

Table 5.2: Selected number of split attempts for each data set using Bisecting EM, average run time for the reported number of split attempts and Number of EM iterations equal to the run time of five runs of Bisecting EM using reported Split Attempts.

Using the selected number of split attempts we ran Bisecting EM five times. The run time was summed for the five runs and used to determine the appropriate number of times to run EM on the same data set. Run time was considerably faster for smaller data sets. Table 5.3 shows the results of clustering with Bisecting EM and EM according to the chosen external and internal metrics.

Each metric in the table is accompanied by an arrow to indicate whether higher or lower values are better. The first line for each metric is the mean and the second line contains the standard deviation. In the first line for each metric we have indicated in bold the better result between the two clustering algorithms. Internal metrics are informational and are separated from external metrics by a double line.

Bisecting EM had better results for each of the metrics. With many of the metrics the standard deviation for Bisecting EM seems to be less than EM. However, there are clear cases where the standard deviation of Bisecting EM is much higher for specific metrics and data sets. No clear pattern emerges from the standard deviation data.

For EM refinement, we use the output from Bisecting EM as the input to EM in a refining step. To do this we use the cluster assignments made in the last iteration of bisection to estimate parameters for a run of EM. This begins the familiar process of estimating and maximizing until convergence. The results are included in Table 5.4 with a comparison to unrefined Bisecting EM. In every case EM refinement improved the results of Bisecting EM using external metrics. The standard deviation was again mixed.

Metric	Bisecting EM	EM
F-Measure ↑	**0.328** 0.006	0.306 0.008
VI ↓	**3.748** 0.005	4.080 0.007
ARI ↑	**0.194** 0.006	0.165 0.012
V-Measure ↑	**0.429** 0.009	0.412 0.014
Q_2 ↑	**0.730** 0.010	0.695 0.012
TD ↑	190.649 5.895	567.833 7.739
TIAD ↓	8.353 0.367	14.522 1.13
LD ↑	-1.653×10^7 1.461×10^5	-1.657×10^7 1.525×10^5

(a) Enron data set using 90 split attempts for Bisecting EM.

Metric	Bisecting EM	EM
F-Measure ↑	**0.543** 0.00007	0.306 0.00003
VI ↓	**2.826** 0.008	4.080 0.009
ARI ↑	**0.365** 0.011	0.165 0.011
V-Measure ↑	**0.524** 0.013	0.412 0.012
Q_2 ↑	**0.758** 0.009	0.695 0.008
TD ↑	271.878 3.727	320.888 4.829
TIAD ↓	22.212 1.518	10.732 1.356
LD ↑	-2.851×10^7 1.547×10^5	-3.142×10^7 2.245×10^5

(b) 20 Newsgroups data set using 82 split attempts for Bisecting EM.

Metric	Bisecting EM	EM
F-Measure ↑	**0.377** 0.005	0.303 0.012
VI ↓	**3.937** 0.007	4.080 0.009
ARI ↑	**0.201** 0.005	0.165 0.010
V-Measure ↑	**0.422** 0.012	0.412 0.008
Q_2 ↑	**0.727** 0.007	0.695 0.012
TD ↓	2008.804 2.491	2226.456 3.361
TIAD ↑	17.758 1.336	36.548 2.005
LD ↑	-2.014×10^8 4.798×10^5	-2.020×10^8 4.683×10^5

(c) del.icio.us data set using 86 split attempts for Bisecting EM.

Table 5.3: Comparing the performance of Bisecting EM to EM clustering. Each row contains the mean followed by the standard deviation for the corresponding metric

5.3 Bisecting Gibbs

The first experiment required to run Bisecting Gibbs is to run multiple chains of Gibbs sampling with the number of clusters equal to two for each data set. By running these experiments we were

Metric	Bisecting EM	BEM/EMR
F-Measure ↑	0.328 0.011	**0.333** 0.012
VI ↓	3.748 0.005	**3.559** 0.007
ARI ↑	0.194 0.011	**0.207** 0.012
V-Measure ↑	0.429 0.012	**0.441** 0.010
Q_2 ↑	0.730 0.010	**0.731** 0.008
TD ↑	190.649 5.895	183.444 4.562
TIAD ↓	8.353 0.367	13.138 0.623
LD ↑	-1.653×10^7 1.460×10^5	-1.672×10^7 1.415×10^5

(a) Enron data set using 90 split attempts for each.

Metric	Bisecting EM	BEM/EMR
F-Measure ↑	0.54282 0.007	**0.614927** 0.009
VI ↓	2.826 0.009	**2.470** 0.011
ARI ↑	0.365 0.009	**0.386** 0.008
V-Measure ↑	0.524 0.012	**0.555** 0.011
Q_2 ↑	0.758 0.005	**0.769** 0.005
TD ↑	271.877 3.727	262.978 2.331
TIAD ↓	22.212 1.517	19.871 1.419
LD ↑	-2.851×10^7 1.547×10^5	-2.741×10^7 1.497×10^5

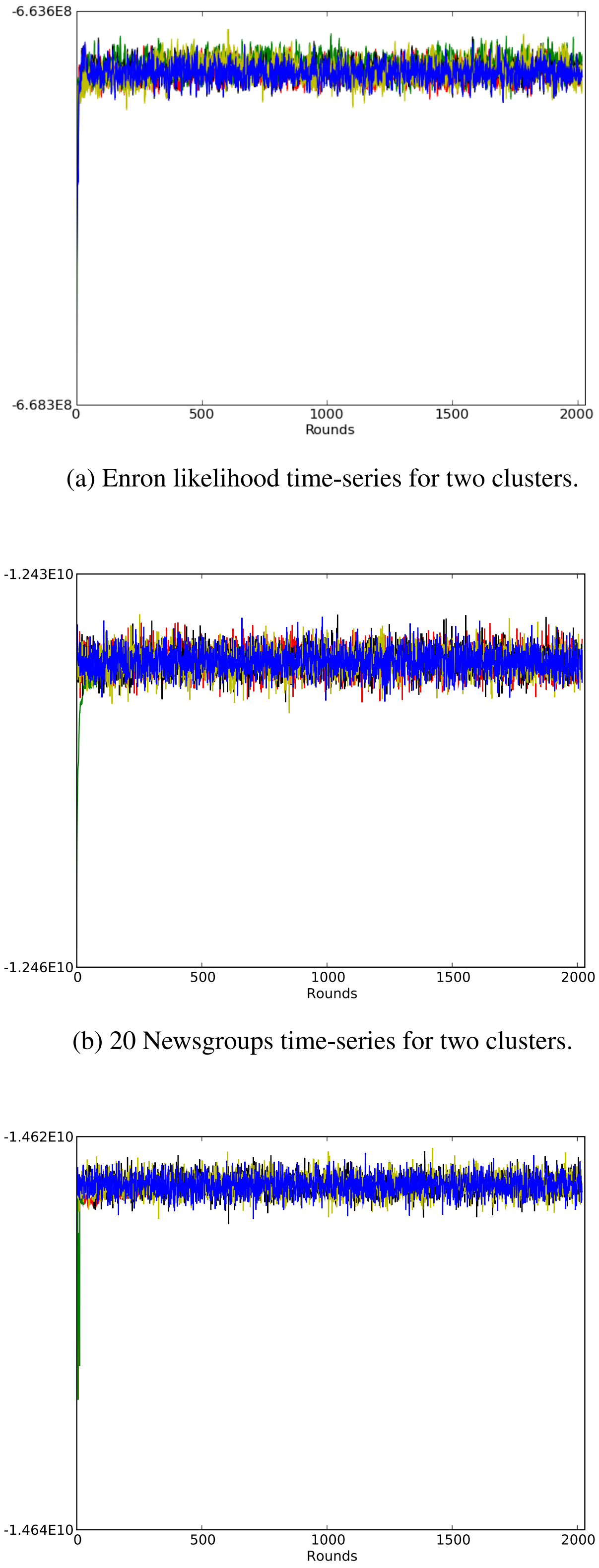
(b) 20 Newsgroups data set using 82 split attempts for each.

Metric	Bisecting EM	BEM/EMR
F-Measure ↑	0.377 0.009	**0.415** 0.00010
VI ↓	3.937 0.006	**3.862** 0.008
ARI ↑	0.201 0.010	**0.245** 0.009
V-Measure ↑	0.422 0.013	**0.429** 0.011
Q_2 ↑	0.727 0.008	**0.754** 0.009
TD ↑	2008.804 2.491	2004.527 2.524
TIAD ↓	17.758 1.336	17.463 0.938
LD ↑	-2.014×10^8 4.817×10^5	-2.009×10^8 4.598×10^5

(c) del.icio.us data set using 86 split attempts for each.

Table 5.4: Results comparing the performance of Bisecting EM without refinement to Bisecting EM with EM refinement (BEM/EMR). Each row contains the mean followed by the standard deviation for the corresponding metric

able to choose a reasonable number of sampling rounds to take at each bisection during Bisecting Gibbs. Results for these experiments are shown in Figure 5.2 .



(a) Enron likelihood time-series for two clusters.

(b) 20 Newsgroups time-series for two clusters.

(c) del.icio.us time-series for two clusters.

Figure 5.2: Gibbs sampling with k=2. Five chains are shown superimposed.

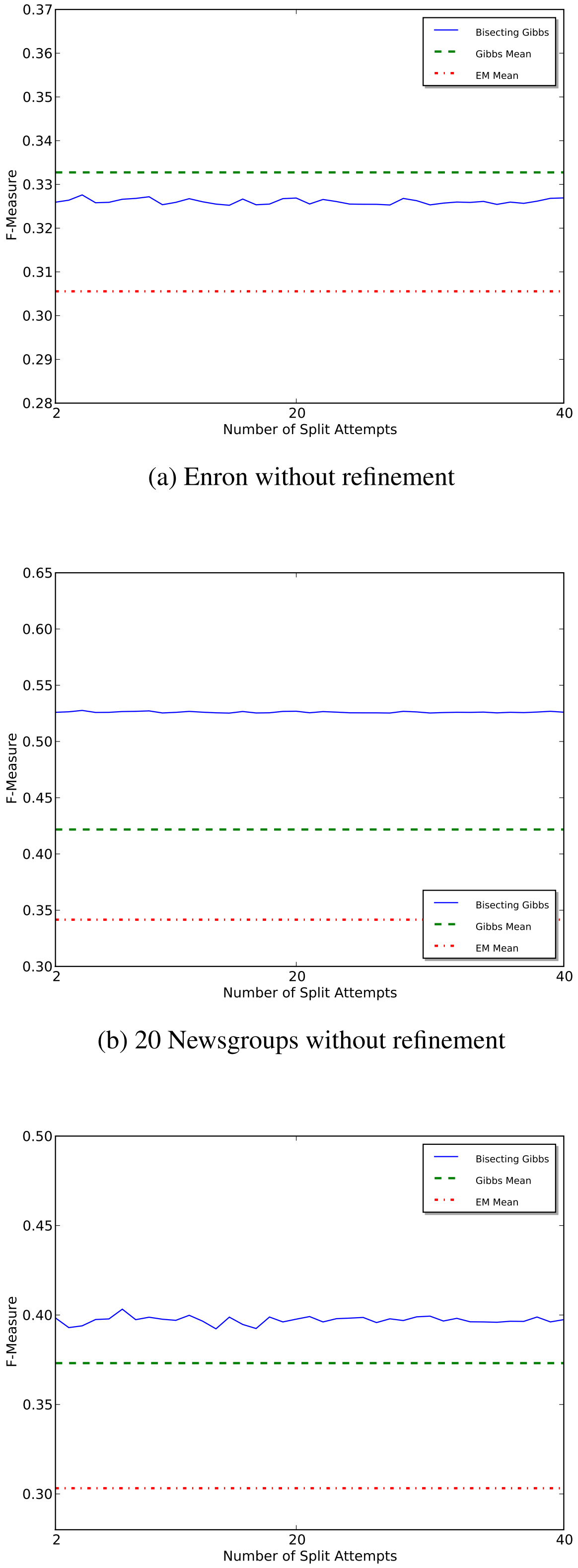
The time-series data show that with five random initializations the sample likelihood settles into a common range for each chain. We chose 800 rounds with a burn of 20 rounds for each bisection. Our criteria for choosing the number of rounds is to examine the time-series data from each of the mixing plots and approximate the lowest number of rounds before each of the chains appeared to settle. 800 rounds could likely be reduced using the time-series data in Figure 5.2 but we decided to use a conservative number to account for variations from random initialization and a different data likelihood surface for later bisections. Table 5.5 contains the results of running Bisecting Gibbs and Gibbs using 800 rounds.

In order to determine the necessity of increasing split attempts with Bisecting Gibbs we again swept the number of split attempts and compared F-Measure. Figure 5.3 shows the results of these experiments. Due to the excessive time and resources running multiple chains of Gibbs sampling, we limited the number of split attempts to 40. We found that split attempts with Gibbs Sampling are unnecessary given a sufficient number of sampling rounds. This is most likely due to the fact that the global maximum of the data likelihood is generally reached after relatively few samples in spite of the initialization of the model parameters (Walker & Ringger, 2008).

After establishing the number of sampling rounds and split attempts to use for Bisecting Gibbs, we continue with our experiments comparing Bisecting Gibbs to Gibbs. The results of these experiments are included in Table 5.5 .

Again we use the final cluster output in a refinement step. In this case there are a few differences. First we use EM to refine Bisecting Gibbs. While it is possible to use Gibbs to refine Bisecting Gibbs this was not included in the scope of this work due to the amount of computing and programming time required to run these experiments. In either case, improvement from a refinement step shows that the final clustering from Bisecting Gibbs can be improved, even if it outperforms the best result from other algorithms on the same data set.

Additionally, we do not run multiple iterations of Bisecting Gibbs and EM refinement. As we have shown previously, Bisecting Gibbs tends to output the same clustering when sufficient sampling rounds are run. Multiple runs of Bisecting Gibbs and EM refinement equates to multiple



(a) Enron without refinement

(b) 20 Newsgroups without refinement

(c) del.icio.us without refinement

Figure 5.3: Sweeping split attempts using Bisecting Gibbs

Metric	Bisecting Gibbs	Gibbs
F-Measure ↑	0.332	**0.333**
VI ↓	**3.369**	3.574
ARI ↑	0.128	**0.136**
V-Measure ↑	**0.330**	0.324
Q_2 ↑	**0.722**	0.713
TD ↑	902.235	902.502
TIAD ↓	12.588	11.901
LD ↑	-1.618×10^7	-1.541×10^7

(a) Enron data set.

Metric	Bisecting Gibbs	Gibbs
F-Measure ↑	**0.396**	0.373
VI ↓	**3.525**	3.615
ARI ↑	**3.215**	3.127
V-Measure ↑	**0.422**	0.412
Q_2 ↑	**0.722**	0.710
TD ↑	442.248	445.089
TIAD ↓	17.231	19.874
LD ↑	-2.512×10^7	-2.556×10^7

(b) 20 Newsgroups data set.

Metric	Bisecting Gibbs	Gibbs
F-Measure ↑	**0.509**	0.422
VI ↓	**2.318**	2.525
ARI ↑	**0.289**	0.235
V-Measure ↑	**0.569**	0.412
Q_2 ↑	**0.738**	0.712
TD ↑	2471.946	2478.805
TIAD ↓	19.635	20.805
LD ↑	-1.974×10^8	-1.943×10^8

(c) del.icio.us data set.

Table 5.5: Results comparing the performance of Bisecting Gibbs to the performance of Gibbs clustering.

runs of EM with the same initialization and consequently the same output. The results for these experiments are included in Table 5.6 .

5.4 Comparison

To summarize our results, Figure 5.4 is included showing the results of all four algorithms run on each of the three data sets. Also included are the results from refinement using EM on both Bisecting EM and Bisecting Gibbs.

In a practical implementation involving multiple runs of EM or Bisecting EM, we would retain the clustering that produces the highest likelihood value. However, in Figure 5.4 we report the *expected* value of the F-measure for EM and Bisecting EM after multiple random restarts. In addition we report the maximum and minimum F-measures to show the range of possible results. It is important to note that had we picked the best result (rather than the expected result) according to likelihood, it would have done no better than the maximum value shown; consequently, the

Metric	Bisecting Gibbs	BG/EMR
F-Measure ↑	0.332	**0.337**
VI ↓	3.440	**3.316**
ARI ↑	0.126	**0.168**
V-Measure ↑	**0.350**	0.320
Q_2 ↑	0.716	**0.753**
TD ↑	902.235	977.044
TIAD ↓	12.588	13.504
LD ↑	-1.618×10^7	-1.647×10^7

(a) Enron data set.

Metric	Bisecting Gibbs	BG/EMR
F-Measure ↑	0.400	**0.400**
VI ↓	3.673	**3.260**
ARI ↑	3.297	**3.569**
V-Measure ↑	0.416	**0.431**
Q_2 ↑	0.755	**0.787**
TD ↑	442.248	423.256
TIAD ↓	17.231	16.992
LD ↑	-2.512×10^7	-2.464×10^7

(b) 20 Newsgroups data set.

Metric	Bisecting Gibbs	BG/EMR
F-Measure ↑	0.509	**0.554**
VI ↓	2.187	**1.973**
ARI ↑	0.582	**0.774**
V-Measure ↑	0.723	**0.948**
Q_2 ↑	0.862	**0.880**
TD ↑	2471.946	2468.697
TIAD ↓	19.635	18.394
LD ↑	-1.974×10^8	-1.964×10^8

(c) del.icio.us data set.

Table 5.6: Results comparing the performance of Bisecting Gibbs without refinement to the performance of Bisecting Gibbs with EM refinement(BG/EMR).

results from this graph and the relative ranking of the various algorithms would not change in a substantive way.

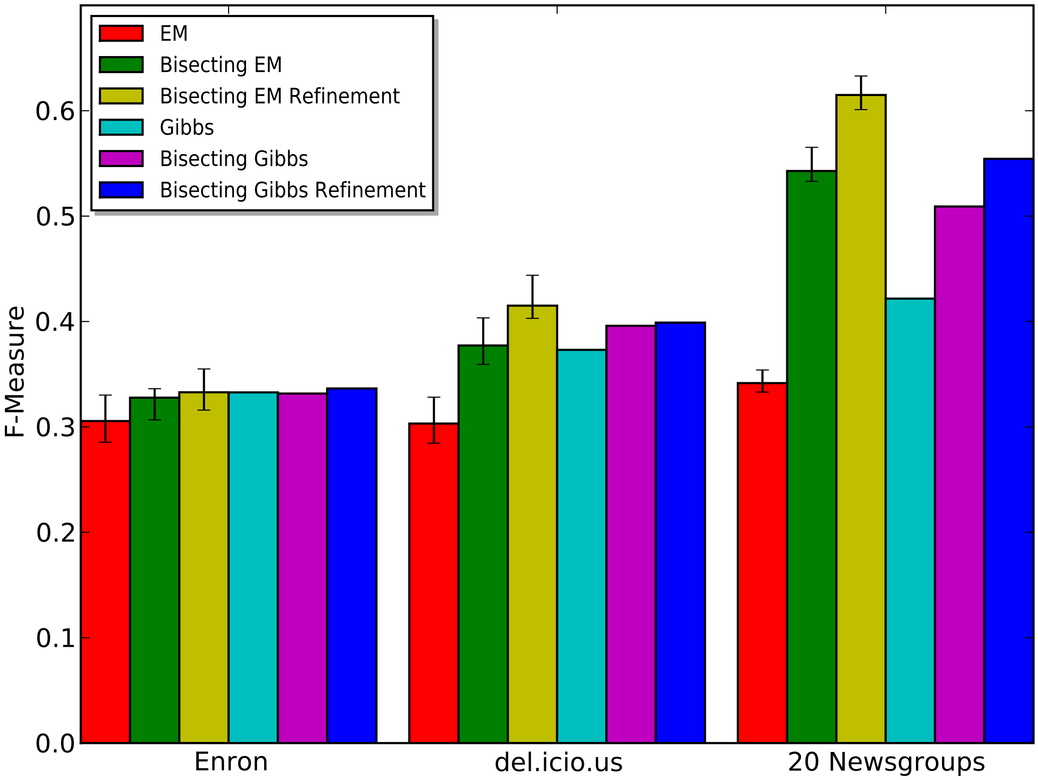


Figure 5.4: Model-based clustering and bisecting variants with max and min shown for EM and Bisecting EM

Chapter 6

Qualitative Analysis

In this chapter we assess the quality of our clusters based on the hidden class labels on a more granular level. While metrics provide a succinct number to assess quality we add qualitative analysis in this chapter. Additionally we examine the characteristics of the data sets and how that may affect the performance of the algorithm. There are many contingency tables which we could show but we have chosen to focus on the best run of Bisecting EM. While this may sound biased, the difference in one run of Bisecting EM and another is found in the initialization of the parameters. The initialization is not an area we are examining in our analysis which makes choosing any run of Bisecting EM interesting so long as we are consistent in picking the corresponding best, worst or middle for each of the data sets. We therefore present data from the best run to show how the best performance of Bisecting EM succeeds or fails to reproduce the original class labels.

In this chapter we present a heatmap for each data set. These heatmaps are increasingly dark when a particular cluster included more documents from a specific class. The purpose of including these heatmaps is to provide a visual representation of the entire dataset with the ability to quickly see how specific clusters map to specific classes. A dark set of boxes along the diagonal indicates strong agreement between document clusters and the labeled data. Data used to generate the "big picture" heatmaps is contained in Appendix A. Because much of this data would be unreadable if we shrunk it to fit on the page, we have removed some of the less interesting clusters in Appendix A. We focus on the best and worst clusters to see where we are getting things right and where we are getting things wrong. The mapping of clusters to classes is one-to-one as we have chosen to use the number of classes as the target number of clusters.

Finally, we acknowledge the potential value that internal metrics offer in analysis of clusters as well. However, because the metrics are more open to interpretation we do not use them. In addition, the purpose of document clustering can go beyond reproducing labeled classes: clustering can also be a means of mining documents for interesting patterns which people may not be able to see otherwise. Each of these points could lead to lengthy subjective debate and are not the ultimate purpose of analyzing bisecting techniques in the context of this thesis. Therefore, we have chosen to exclude internal metrics or new patterns as part of our qualitative analysis.

6.1 Enron

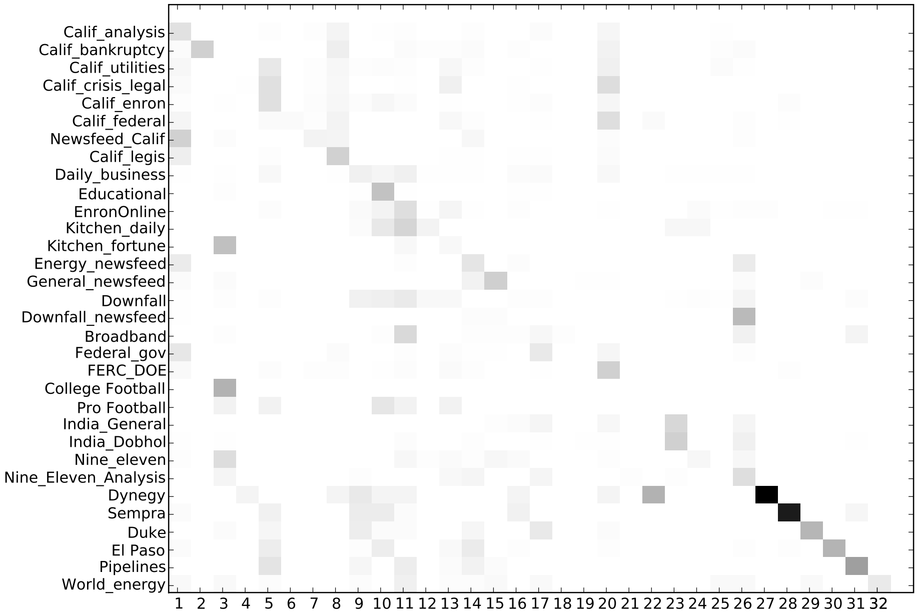


Figure 6.1: A contingency heatmap showing topic on the y-axis and clusters on the x-axis for the Enron data set

The data used to generate Figure 6.1 is contained in Table A.1a and Table A.2 in Appendix A. One of the first things we noticed in examining documents to come up with class labels for each of the classes is that extra information about Enron must have been known in order to separate some of the classes. For example, many of the classes talk about the California energy crisis and how that effects legislation and the connection of both of those to creative accounting practices. With only the words in a document it would be difficult to pick distinctive topics for each of the 32 classes.

Many of the smaller classes that had distinct terms in them were more concentrated around a possible corresponding class. For example, “Dynergy”, “Sempra”, and “College Football”. Each of these have a small set of distinguishing words such as the class labels we have given them show.

Another interesting characteristic of this data set which is different from the others is the rather non-uniform distribution of the documents into class labels. The largest class, “Daily_business”, makes up 30% of the entire dataset. It is natural to expect a more uniform distribution of documents from bisecting techniques where the largest cluster is always being divided. However, the non-uniform distribution of the Enron data is not a clear reason for why bisecting techniques fail to perform significantly better than the base algorithms. Also, the final number of documents in each of the generated clusters show that bisecting methods do adapt to non-uniform distributions of documents. Therefore, we conclude that the non-uniform document distribution of Enron data is not an issue for bisecting techniques.

6.2 del.icio.us

The data used to generate Figure 6.2 is contained in Table A.1b and Table A.3 . The del.icio.us is a considerably larger data set than the other two. As with Enron, we see that the del.icio.us data set has a number of areas of overlap. There are cases such as “games” and “final fantasy” were the topic implies a hierarchical relationship. In this particular case “games” is a super-set which contains “final fantasy.” However, “final fantasy” can contain documents about much more than the video game. In general there appear to be a wider array of topics in the del.icio.us data set even if some of them are closely related.

Another interesting feature of the del.icio.us data set is the relatively uniform size of the document classes. Most of the classes have 500 documents but there are enough significant exceptions that this data set cannot be said to have a uniform distribution of documents. Again we see

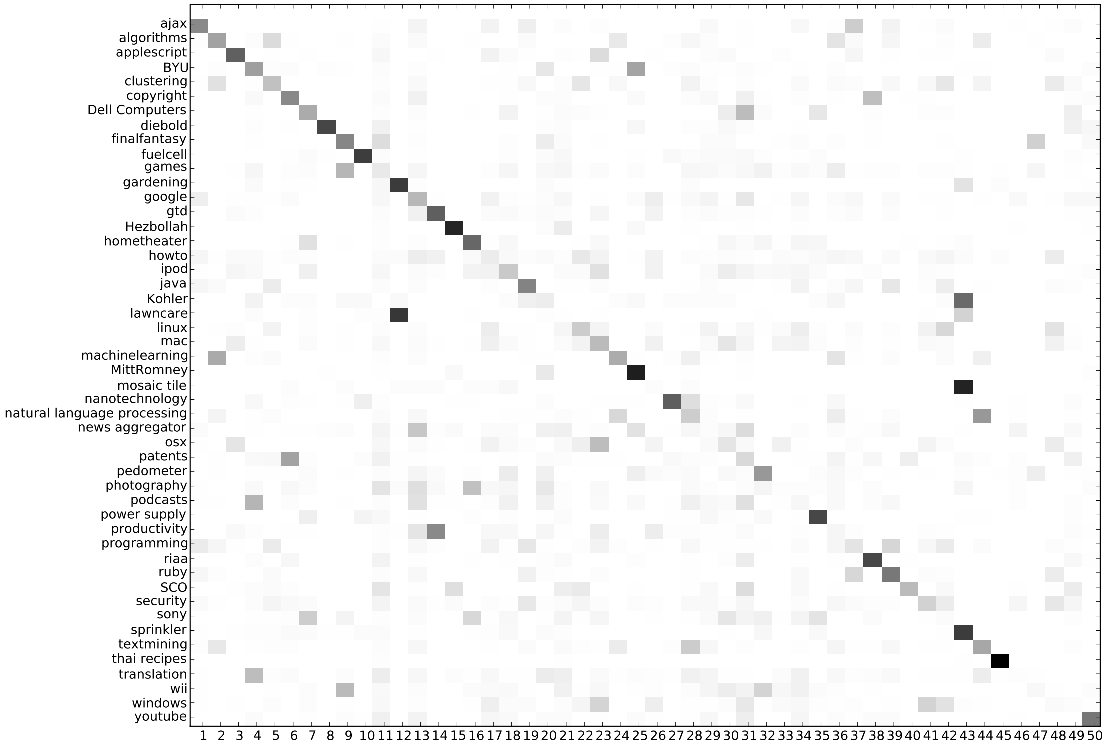


Figure 6.2: A contingency heatmap showing topic on the y-axis and clusters on the x-axis for the del.icio.us data set

that bisecting methods produce roughly the same amount of significantly smaller clusters as there are document classes. However, in the case of the del.icio.us data set we see a significant performance improvement with regard to external metrics. This is a strong confirmation that bisecting methods do not require a uniform data set to closely match human labeled data.

A unique trend in the clustering data for the del.icio.us data set is something we call the “omnicluster.” These are clusters that contain documents from almost every document class. At the same time no one document class stands out significantly from another in document count. Cluster 20, which is represented in the x-axis by 20 in Figure 6.2 and Table A.3 , is an example of such a cluster.

6.3 20 Newsgroups

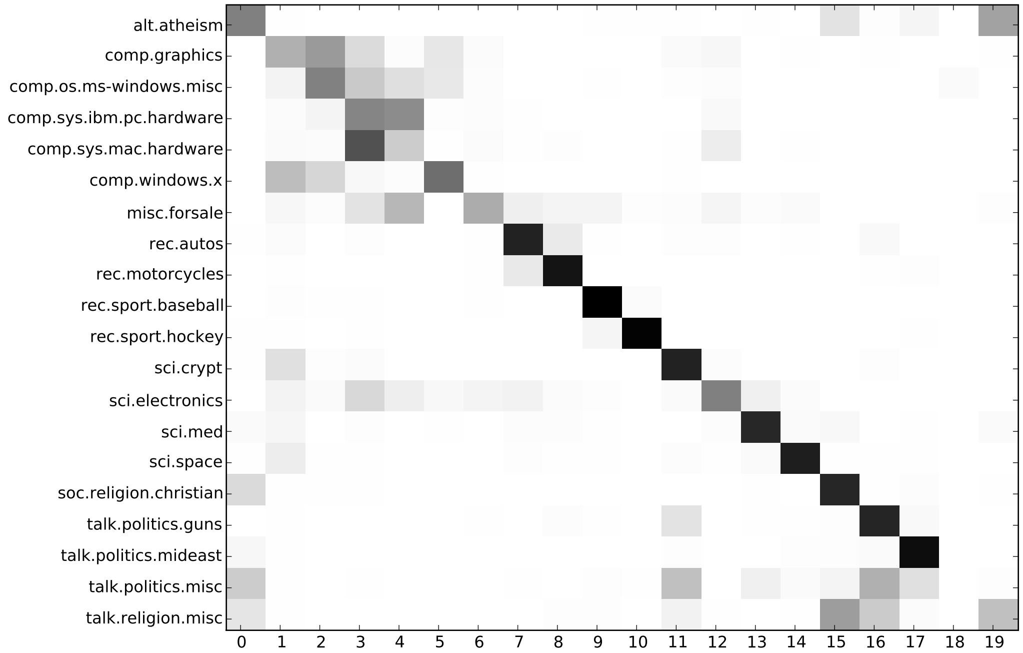


Figure 6.3: A contingency heatmap showing topic on the y-axis and clusters on the x-axis for the 20 Newsgroups data set

The data used to generate Figure 6.3 is contained in Table A.4 . We did not include a table with a break down of the classes sizes for the 20 Newsgroups data set as this data set contains a uniform distribution of documents for each of the classes. Each of the classes contains 1,000 documents.

Like Enron and del.icio.us, the 20 Newsgroups data set has some document classes with obvious overlap. These prove to be the most difficult classes to recreate. It is important to note the manner in which the class labels were applied to the document class. The 20 Newsgroups data set contains newsgroups posting by individuals to a particular newsgroup. Humans did not examine a completed document and classify it. It is likely the case that users posted messages that could have been included in other newsgroups and at times they may have been given a recommendation to post it elsewhere. As a result, the class labels for the 20 Newsgroups data set are more subjective and it is possible that a post-hoc labeling of this data set would categorize newsgroup postings differently.

Overall the quality of the clusters for the 20 Newsgroups data set show a high correlation with the human-labeled classes. Cluster 19 seems to be the only cluster that is not clearly attached to any document class. We examined several of the files and found that they contained mime-encoded files.

Finally, the 20 Newsgroups data set seems to perform much better due to the style of writing. This is the oldest data set of the three data sets and it was written in a period where less online communication was occurring. There are natural changes to language and style over time and depending on the forum. We speculate that the authors of the 20 Newsgroups data set may have been more cautious in their communication using distinctive terms and phrases to ensure their message was passed on correctly without as much ambiguity.

Chapter 7

Conclusions

As we have seen with K-means, altering the approach to model-based document clustering by using bisecting techniques improves the ability to recreate the document categories created by humans in most cases. We have also seen that refinement of the bisecting variant improves the clustering.

In each of our selected data sets we see improvement with Bisecting EM and Bisecting EM with refinement over EM and Gibbs (see Figure 5.4). Bisecting Gibbs and Bisecting Gibbs with refinement show improvement on the del.icio.us and 20 Newsgroups data set and are essentially equal to the results from Gibbs on the Enron data set. The improvement was most dramatic when using EM as a base algorithm for bisecting and bisecting with refinement. In every case where we examined the effect of a refinement step to bisecting we noticed there was improvement over simple bisecting using a model-based algorithm.

The only case where bisecting methods did not improve the results of the corresponding model-based method was using Gibbs and Bisecting Gibbs on the Enron data set. While the results for Enron using Gibbs and Bisecting Gibbs were equal to Gibbs, this pattern was atypical of results from other data sets. Characteristics for each of the data sets such as size, number of clusters and content, may affect the effectiveness of bisecting methods. Further evidence that bisecting methods may be particularly tuned to specific data sets can be seen from the 20 Newsgroups data set which has the fewest number of clusters but a rather large number of documents.

In addition, the 20 Newsgroups data set shows another interesting result. The results of Bisecting EM are better than Bisecting Gibbs. This is particularly interesting given the fact that

EM does not perform better than Gibbs on the same data set. If we only look at the maximum F-Measure for Bisecting EM compared to Bisecting Gibbs we see that Bisecting EM is better than Bisecting Gibbs on all three data sets.

This work represents a step forward in model-based clustering by augmenting such methods with bisecting. In so doing, we have surpassed previously known algorithms' ability to recreate the work done by human experts in organizing document sets. This is an important contribution to the field of text mining and specifically document clustering.

7.1 Future Work

The results of these experiments suggest further study of bisecting techniques to understand why this approach improves on existing clustering techniques especially when fewer clusters are desirable. Deeper investigation into the results of each bisection may yield better understanding about how bisecting works. This includes examining what the clusters look like qualitatively at each bisection and examining the possibility of reducing the number of split attempts as the cluster we are splitting becomes smaller from earlier iterations.

Further study on selecting the best bisection is another area where continued study may be useful. With many different internal metrics, we can explore the outcomes from using each of them on different data sets. In addition to understanding the ideal way to bisect a cluster, this may yield additional insight into what each of the internal metrics tells us about the data.

Efficient clustering of large data sets is another future area of experimentation. Bisecting methods easily extend to grid computing. Split attempts could be run on a number of nodes in parallel. Additional data sets could be used to understand how well bisecting methods scale with additional nodes. Before undertaking this change we would like to experiment with deterministic annealing EM to see if split attempts could be reduced or eliminated. By using deterministic annealing EM, EM is less likely to converge on a local maximum which is the basis for multiple split attempts using random initialization (Ueda & Nakano, 1998; Elkan, 2006).

We employed basic ad hoc feature selection in our experiments by using the top ten terms using TFIDF and a stop word list. While our qualitative analysis points out some possible areas of feature selection we did not re-experiment using additional feature selection. In the future, we could expand the feature selection we have done by examining the effects of increasing the number of features to keep and decreasing that number. Additionally, we may discover obvious stop words that are not being eliminated. It is important to note that feature selection must be carefully undertaken as not to upset the nature of unsupervised learning which can be applied on data sets that do not have pre-annotated classes.

Finally, two models which have outperformed the mixture of multinomials model on most data sets should be explored in future experiments. The first is the von Mises-Fisher model which is a model-based adaptation of the spherical k-means algorithm using the cosine similarity metric (Zhong & Ghosh, 2005; Mardia & El-Atoum, 1976). The second is the EDCM model proposed by Elkan which is an approximation to the DCM model using an exponential family of distributions (Elkan, 2006). In addition to using each of these models, the metric used by each of these papers (NMI) has the ability to evaluate clusterings which do not match the natural number of classes. Adding this external metric would be helpful in evaluating which internal metrics appear to be doing the best at predicting the bisection to keep.

References

Berkhin, P. (2006). A survey of clustering data mining techniques. In J. Kogan, C. Nicholas, & M. Teboulle (Eds.), *Grouping Multidimensional Data: Recent Advances in Clustering*, Springer. 25–71.

Berry, M. W., Brown, M., & Signer, B. (2007). 2001 topic annotated enron email data set.

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

Boley, D., Gini, M., Gross, R., Han, E.-H. S., Hastings, K., Karypis, G., Kumar, V., Mobasher, B., & Moore, J. (1999). Partitioning-based clustering for web document categorization. *Decision Support Systems*, *27*(3), 329–341.

Dempster, A., Laird, N., & Rubin, D. (1977). Maximum likelihood from incomplete data via the EM algorithm. *Journal of the Royal Statistical Society Series B*, *39*, 1–38.

Dom, B. (2001). An information-theoretic external cluster-validity measure. Tech. Rep. RJ10219, IBM. URL `http://citeseer.ist.psu.edu/dom01informationtheoretic.html`.

Elkan, C. (2006). Clustering documents with an exponential-family approximation of the dirichlet compound multinomial distribution. In *ICML '06: Proceedings of the 23rd International Conference on Machine learning*. New York, NY: ACM, 289–296.

Grossman, D. (2004). *Information retrieval : algorithms and heuristics*. Dordrecht, Great Britain: Springer, 2nd ed.

Hearst, M. A. (1999). Untangling text data mining. Tech. rep., University of Maryland.

Hearst, M. A. & Pedersen, J. O. (1996). Reexamining the cluster hypothesis: scatter/gather on retrieval results. In *SIGIR '96: Proceedings of the 19th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*. New York, NY: ACM, 76–84.

Jain, A. K. & Dubes, R. C. (1988). *Algorithms for clustering data*. Upper Saddle River, NJ: Prentice-Hall, Inc.

Joachims, T. (1997). A probabilistic analysis of the rocchio algorithm with tfidf for text categorization. In *Machine Learning: Proceedings of the Fourteenth International Conference (ICML'97)*. 143–151.

Jones, K. S. (1972). A statistical interpretation of term specificity and its application in retrieval. *Journal of Documentation*, *28*, 11–21.

King, B. (1967). Step-wise clustering procedures. *Journal of the American Statistical Association*, *69*, 86–101.

Larsen, B. & Aone, C. (1999). Fast and effective text mining using linear-time document clustering. In *KDD '99: Proceedings of the Fifth ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*. New York, NY: ACM, 16–22.

Manning, C. D. & Schutze, H. (1999). *Foundations of Statistical Natural Language Processing*. The MIT Press.

Mardia, K. V. & El-Atoum, S. A. M. (1976). Bayesian inference for the von Mises-Fisher distribution. *Biometrika*, *63*(1), 203–206. URL `http://www.jstor.org/stable/2335106`.

McCallum, A. & Nigam, K. (1998). A comparison of event models for naive bayes text classification. In *AAAI-98 Workshop on Learning for Text Categorization*. Madison, WI, 792 – 799. URL `http://citeseer.ist.psu.edu/mccallum98comparison.html`.

Meilă, M. (2002). Comparing clusterings—an information based distance. *Journal of Multivariate Analysis*, *98*(5), 873–895.

Meilă, M. & Heckerman, D. (1998). An experimental comparison of several clustering and initialization methods. In *Proceedings of the Fourteenth Conference on Uncertainty in Artificial Intelligence*. San Francisco, CA, 386–395. URL `http://citeseer.ist.psu.edu/meila98experimental.html`.

Nigam, K., McCallum, A. K., Thrun, S., & Mitchell, T. M. (2000). Text classification from labeled and unlabeled documents using EM. *Machine Learning*, *39*(2/3), 103–134. URL `http://citeseer.ist.psu.edu/nigam99text.html`.

Rosenberg, A. & Hirschberg, J. (2007). V-measure: A conditional entropy-based external cluster evaluation measure. In *Joint Conference on Empirical Methods in Natural Language Processing and Computational Language Learning*. Prague, Czech Republic, 410–420.

Salton, G. & Buckley, C. (1988). Term-weighting approaches in automatic text retrieval. *Information Processing and Management*, *24*, 513–523. URL `http://citeseerx.ist.psu.edu/viewdoc/summary?doi=10.1.1.101.9086`.

Savaresi, S., Boley, D. L., Bittanti, S., & Gazzaniga, G. (2000). Choosing the cluster to split in bisecting divisive clustering algorithms. In *Proceedings of the 2002 SIAM International Conference on Data Mining*. Arlington, VA, 299–314.

Sneath, P. & Sokal, R. (1973). *Numerical Taxonomy*. London, UK: Freeman.

Steinbach, M., Karypis, G., & Kumar, B. (2000). A comparison of document clustering techniques. Tech. rep., University of Minnesota.

Ueda, N. & Nakano, R. (1998). Deterministic annealing EM algorithm. *Neural Networks*, *11*(2), 271–282.

Walker, D. & Ringger, E. (2008). Model-based document clustering with a collapsed gibbs sampler. In *Proceedings of the Fourteenth ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*. Las Vegas, NV, 704–712.

Wasserman, L. (2003). *All of Statistics*. New York, NY: Springer.

Yeung, K. Y., Yeung, K. Y., Ruzzo, W. L., & Ruzzo, W. L. (2001). An empirical study on principal component analysis for clustering gene expression data. *Bioinformatics*, *17*, 763–774. URL `http://citeseer.ist.psu.edu/article/yeung01empirical.html`.

Zhao, Y. & Karypis, G. (2001). Criterion functions for document clustering: Experiments and analysis. Tech. rep., University of Minnesota.

Zhao, Y., Karypis, G., & Fayyad, U. (2005). Hierarchical clustering algorithms for document datasets. *Data Mining and Knowledge Discovery*, *10*(2), 141–168.

Zhong, S. & Ghosh, J. (2005). Generative model-based document clustering: a comparative study. *Knowledge and Information Systems*, *8*(3), 374–384.

Appendix A

Contingency Tables

Topic	Size
Calif_analysis	304
Calif_bankruptcy	36
Calif_utilities	115
Calif_cris_legal	87
Calif_enron	699
Calif_federal	61
Newsfeed_Calif	190
Calif_legis	181
Daily_business	1516
Educational	92
EnronOnline	271
Kitchen_daily	37
Kitchen_fortune	11
Energy_newsfeed	331
General_newsfeed	48
Downfall	144
Downfall_newsfeed	48
Broadband	26
Federal_gov	81
FERC_DOE	219
College Football	100
Pro football	6
India_General	38
India_Dobhol	79
Nine_eleven	29
Nine_Eleven_Analysis	30
Dynegy	7
Sempra	16
Duke	17
El Paso	34
Pipelines	17
World_energy	25

(a) Enron

Topic	Size
algorithms	498
applescript	500
BYU	398
clustering	499
copyright	500
Dell+Computers	380
diebold	500
finalfantasy	500
fuelcell	500
games	500
gardening	500
google	499
gtd	500
Hezbollah	500
hometheater	500
howto	500
ipod	500
java	500
Kohler	84
lawncare	238
linux	500
machinelearning	500
mac	500
MittRomney	273
mosaic+tile	69
nanotechnology	500
natural+language+processing	167
news+aggregator	500
osx	500
patents	500
pedometer	267
photography	346
podcasts	390
power+supply	498
productivity	500
programming	500
riaa	500
ruby	451
SCO	500
security	500
sony	403
sprinkler	189
textmining	500
thai+recipes	500
translation	500
wii	500
windows	500
youtube	478

(b) del.icio.us

Table A.1: Topics and number of documents.

	01	05	06	08	09	11	12	14	15	20	21	22	23	24	25	26	28	29	30	31
Calif_analysis	**120**	11	0	45	0	9	0	10	2	39	0	0	2	0	6	3	2	1	0	1
Calif_bankruptcy	2	0	0	8	0	2	1	2	0	6	0	0	0	0	1	2	0	0	0	0
Calif_utilities	13	**34**	0	13	3	3	0	4	0	23	0	0	0	0	6	2	0	0	0	0
Calif_cris_legal	6	35	0	8	0	2	0	0	0	**39**	0	0	0	0	1	0	0	0	0	0
Calif_enron	19	**283**	0	75	28	38	1	4	0	89	0	0	3	2	4	3	32	0	1	3
Calif_federal	8	4	4	10	0	0	0	2	0	**26**	0	4	0	0	0	2	1	0	0	0
Newsfeed_Calif	**114**	0	0	27	0	1	0	20	2	9	0	0	0	0	0	5	0	0	0	0
Calif_legis	41	6	0	**109**	0	2	0	3	1	10	0	0	0	0	1	0	0	0	0	0
Daily_business	34	155	0	20	**309**	306	31	28	6	136	3	0	21	34	23	28	10	1	4	10
Educational	0	0	0	0	2	2	0	1	1	0	0	0	0	1	0	0	0	0	0	0
EnronOnline	0	12	0	0	21	**117**	5	5	1	5	0	0	1	15	5	9	0	1	4	5
Kitchen_daily	0	0	0	0	2	**20**	6	0	0	0	0	0	4	4	0	0	0	0	0	0
Kitchen_fortune	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Energy_newsfeed	87	3	0	4	0	10	0	**112**	2	3	1	0	0	0	3	86	0	0	0	0
General_newsfeed	3	0	0	0	0	0	0	8	**30**	1	0	0	0	0	0	3	0	2	0	0
Downfall	2	5	1	0	26	**39**	14	1	1	1	0	0	2	4	2	22	0	0	0	6
Downfall_newsfeed	1	0	0	0	0	0	0	2	2	0	0	0	0	0	0	43	0	0	0	0
Broadband	0	0	0	0	0	**13**	0	1	1	0	0	0	0	0	0	5	0	0	0	4
Federal_gov	**25**	1	0	5	0	3	0	2	2	10	0	0	0	0	0	3	0	0	0	0
FERC_DOE	18	9	0	7	0	10	0	6	0	**133**	0	0	1	0	2	2	6	0	0	0
College Football	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Pro football	0	1	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0
India_General	0	0	0	0	0	0	0	0	1	4	0	0	**20**	0	0	5	0	0	0	0
India_Dobhol	1	0	0	0	0	4	0	3	0	1	0	0	**48**	0	0	16	0	0	0	2
Nine_eleven	1	0	0	0	0	3	1	1	3	1	0	0	0	3	0	3	0	0	0	0
Nine_Eleven_Analysis	0	0	0	0	1	0	0	4	0	0	1	0	1	0	0	**13**	0	0	0	0
Dynegy	0	0	0	1	2	1	0	0	0	1	0	7	0	0	0	0	0	0	0	0
Sempra	1	3	0	0	4	1	0	0	0	0	0	0	0	0	0	0	**47**	0	0	2
Duke	0	2	0	0	4	1	0	2	0	1	0	0	0	0	0	0	0	**16**	0	0
El Paso	2	8	0	0	1	0	0	9	0	1	0	0	0	0	0	1	1	0	**33**	0
Pipelines	0	6	0	0	2	4	0	3	1	0	0	0	0	0	0	0	0	0	0	**21**
World_energy	2	1	0	0	1	5	0	1	2	0	0	0	0	0	2	2	0	2	0	0

Table A.2: Sample contingency table showing the relationship of natural classes to clusters produced by BEM with refinement for the Enron data.

	06	08	09	10	12	15	19	20	22	25	27	32	33	35	41	42	44	45	48	50
ajax	0	0	0	0	0	0	32	0	0	0	0	1	1	0	2	9	0	0	3	0
algorithms	3	0	2	0	0	1	8	1	1	0	4	0	1	0	0	5	37	0	0	1
applescript	0	0	0	0	0	0	2	0	3	0	0	0	4	0	3	3	0	0	8	0
BYU	6	3	2	0	0	1	0	38	0	138	3	2	0	0	0	0	2	1	0	5
clustering	0	0	0	0	0	0	15	0	48	1	0	0	0	1	14	47	21	0	39	0
copyright	**225**	1	0	0	0	0	0	7	0	10	0	2	0	0	2	0	0	0	0	6
Dell Computers	0	1	0	1	0	0	0	2	5	0	0	7	0	36	4	3	0	0	1	1
diebold	3	**353**	1	0	0	1	0	4	1	16	0	1	0	0	0	1	0	0	0	16
finalfantasy	0	0	**231**	0	0	0	0	36	0	0	0	1	3	0	1	2	0	0	0	12
fuelcell	3	2	0	**371**	0	0	0	8	0	2	13	5	0	6	0	0	0	0	0	2
games	5	1	**141**	1	0	0	1	14	2	7	0	26	5	6	4	4	0	3	0	1
gardening	3	0	0	0	**375**	1	0	9	0	1	0	1	0	1	0	0	0	6	0	2
google	8	0	0	0	0	0	2	7	0	8	2	1	3	0	2	3	3	0	2	14
gtd	0	0	0	0	0	0	4	9	0	1	0	3	0	0	0	2	0	0	1	2
Hezbollah	0	1	0	1	0	**417**	0	7	0	13	0	1	0	0	0	1	0	0	0	3
hometheater	2	0	3	1	0	0	0	0	2	0	0	18	2	16	0	2	0	0	1	0
howto	7	0	1	1	9	0	7	11	**44**	2	1	1	9	12	12	20	1	6	38	1
ipod	1	0	0	0	0	0	1	1	8	1	0	12	14	9	6	5	0	0	1	6
java	0	0	1	0	0	0	**237**	0	3	0	0	0	2	1	5	33	3	0	6	1
Kohler	1	0	2	2	2	0	5	6	0	2	1	0	0	1	0	1	0	0	0	0
lawncare	0	0	0	2	**182**	0	0	2	0	0	0	1	0	0	1	0	0	0	0	0
linux	0	0	2	0	0	0	2	0	**98**	0	0	2	8	4	25	75	0	0	55	1
mac	1	0	3	0	1	0	2	0	23	0	0	0	22	7	11	9	0	0	25	0
machinelearning	0	0	0	0	0	0	2	2	1	2	5	1	0	0	0	8	30	0	0	0
MittRomney	0	1	0	0	0	7	0	23	0	**235**	0	0	0	0	0	0	0	0	0	3
mosaic tile	1	0	1	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
nanotechnology	4	0	2	31	0	2	1	7	0	2	**307**	1	0	0	1	0	0	0	0	0
natural language processing	0	0	0	0	0	0	1	1	1	0	0	0	0	0	0	0	**66**	0	0	0
news aggregator	3	5	7	1	2	10	1	27	2	56	3	1	0	0	1	1	0	0	0	0
osx	1	0	1	0	0	0	5	0	32	0	0	0	27	1	21	6	0	0	38	0
patents	**176**	0	1	6	2	1	0	9	2	0	12	0	0	3	1	0	4	0	0	0
pedometer	2	0	2	0	1	0	0	16	0	17	4	**105**	0	4	1	0	0	1	0	1
photography	10	0	1	1	6	2	0	31	0	2	1	14	3	3	0	1	0	0	0	4
podcasts	5	0	2	0	1	2	0	22	0	9	2	1	2	1	0	0	1	0	0	1
power supply	1	0	0	21	0	0	0	2	1	0	2	13	1	**351**	0	1	2	0	0	0
productivity	2	0	0	0	2	0	0	18	1	1	4	0	1	0	5	2	1	0	1	2
programming	0	0	1	0	0	0	47	3	5	0	0	3	1	1	17	39	8	0	6	0
riaa	17	0	0	0	0	1	0	5	0	4	0	0	0	0	1	0	0	0	1	2
ruby	0	0	0	0	0	0	25	4	1	0	0	1	0	0	6	5	0	0	33	0
SCO	4	0	1	0	0	62	0	1	41	3	0	0	0	0	1	13	0	0	2	0
security	12	1	0	0	0	8	44	5	20	6	0	0	1	0	**87**	37	0	0	46	1
sony	4	0	17	0	0	0	0	2	2	0	0	24	2	42	2	7	0	0	0	1
sprinkler	4	1	1	3	0	0	0	3	0	0	0	6	0	5	0	1	0	0	0	0
textmining	4	0	0	0	0	1	8	0	4	1	4	0	0	0	2	3	**168**	0	0	1
thai recipes	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	**488**	0	1
translation	9	0	6	0	0	6	3	38	0	6	2	0	8	0	2	2	35	3	0	0
wii	0	0	**132**	0	0	0	2	0	2	2	0	82	16	8	0	0	1	0	2	6
windows	1	0	2	0	0	0	8	0	20	0	0	2	10	5	**85**	57	3	0	11	2
youtube	0	0	7	2	0	0	0	8	1	8	0	1	0	0	0	0	0	0	0	**249**

Table A.3: Sample contingency table showing the relationship of natural classes to clusters produced by BEM with refinement for the del.icio.us data.

	00	01	02	03	04	05	06	07	08	09	10	11	12	13	14	15	16	17	18	19
alt.atheism	**472**	5	0	0	0	0	0	1	2	6	6	7	1	4	1	104	8	39	0	343
comp.graphics	2	292	**374**	12	135	89	15	1	1	1	0	20	31	2	7	2	5	3	0	6
comp.os.ms-windows.misc	1	46	**465**	121	201	87	13	1	1	4	1	9	13	0	2	2	1	0	21	3
comp.sys.ibm.pc.hardware	0	18	41	426	**452**	11	12	5	1	2	0	1	23	1	2	1	1	1	0	0
comp.sys.mac.hardware	2	19	18	189	**642**	5	17	5	9	0	2	6	69	2	6	0	1	2	1	1
comp.windows.x	1	244	153	12	26	**537**	5	1	1	2	1	4	3	0	1	1	3	1	2	0
misc.forsale	3	33	14	269	106	2	**309**	62	43	43	8	12	37	13	21	2	1	3	0	10
rec.autos	7	16	1	2	11	0	7	**815**	81	6	0	8	10	1	6	1	25	2	0	0
rec.motorcycles	1	5	1	1	3	0	6	83	**870**	2	1	1	2	3	2	3	5	8	0	0
rec.sport.baseball	0	9	4	0	7	0	4	2	3	**944**	17	0	0	1	2	2	2	2	0	0
rec.sport.hockey	5	4	1	2	4	0	0	1	1	40	**930**	1	1	1	0	0	1	4	3	0
sci.crypt	4	116	10	0	17	3	1	0	2	2	0	**816**	14	2	0	1	8	1	2	0
sci.electronics	3	45	21	65	145	29	43	50	18	8	1	22	**469**	59	18	1	0	2	0	0
sci.med	22	35	1	2	10	4	3	14	13	5	3	3	13	**798**	19	26	2	5	0	20
sci.space	7	67	4	2	5	1	3	10	7	6	2	14	4	19	**831**	4	6	4	0	3
soc.religion.christian	140	6	5	1	5	0	2	0	0	3	2	2	0	6	0	**803**	4	13	0	5
talk.politics.guns	3	4	1	1	0	0	5	3	14	4	2	106	2	6	7	11	**806**	23	1	1
talk.politics.mideast	32	7	2	0	0	0	2	2	1	2	2	11	0	1	9	9	22	**895**	1	2
talk.politics.misc	191	5	1	0	5	0	0	7	1	8	4	234	3	59	20	41	**292**	118	0	11
talk.religion.misc	96	6	0	3	1	0	0	2	9	10	3	49	4	3	5	**363**	195	16	0	235

Table A.4: Sample contingency table showing the relationship of natural classes to clusters produced by BEM with refinement for the 20 Newsgroups data.

Appendix B

Stop Word List

The following is a list of stop words used to reduce the number of words in each document. This list was obtained from the BYU NLP lab code base.

(,), :,   ., /, ¡, ¿, $, =, ?, ;, &, +, %, *,a, an, and, are, as, at, be, but, by, do, for, if, in, it, would, mm, the, to, you, was, of, or, I, his, he, my, www, com, have, http, is, this, with, what, we, were, your, your, any, br, ff, that, the, to, you, was, of, I, his, he, my, www, com, http, a, A, b, B, c, C, d, D, e, E, f, F, g, G, h, H, i, I, j, J, k, K, l, L, m, M, n, N, o, O, p, P, q, Q, r, R, s, S, t, T, u, U, v, V, w, W, x, X, y, Y, z, Z, An, is, this, would, your, any, would, br