2012-11-15

Bayesian Test Analytics for Document Collections

Daniel David Walker
Brigham Young University - Provo

Bayesian Text Analytics for Document Collections

Daniel D. Walker IV

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

Doctor of Philosophy

Eric K. Ringger, Chair
Kevin Seppi
C. Shane Reese
Kent E. Seamons
Tony Martinez

Department of Computer Science

Brigham Young University

November 2012

ABSTRACT

Bayesian Text Analytics for Document Collections

Daniel D. Walker IV
Department of Computer Science, BYU
Doctor of Philosophy

Modern document collections are too large to annotate and curate manually. As increasingly large amounts of data become available, historians, librarians and other scholars increasingly need to rely on automated systems to efficiently and accurately analyze the contents of their collections and to find new and interesting patterns therein. Modern techniques in Bayesian text analytics are becoming wide spread and have the potential to revolutionize the way that research is conducted. Much work has been done in the document modeling community towards this end, though most of it is focussed on modern, relatively clean text data.

We present research for improved modeling of document collections that may contain textual noise or that may include real-valued metadata associated with the documents. This class of documents includes many historical document collections. Indeed, our specific motivation for this work is to help improve the modeling of historical documents, which are often noisy and/or have historical context represented by metadata. Many historical documents are digitized by means of Optical Character Recognition (OCR) from document images of old and degraded original documents. Historical documents also often include associated metadata, such as timestamps, which can be incorporated in an analysis of their topical content. Many techniques, such as topic models, have been developed to automatically discover patterns of meaning in large collections of text. While these methods are useful, they can break down in the presence of OCR errors. We show the extent to which this performance breakdown occurs.

The specific types of analyses covered in this dissertation are document clustering, feature selection, unsupervised and supervised topic modeling for documents with and without OCR errors and a new supervised topic model that uses Bayesian nonparametrics to improve the modeling of document metadata. We present results in each of these areas, with an emphasis on studying the effects of noise on the performance of the algorithms and on modeling the metadata associated with the documents. In this research we effectively: improve the state of the art in both document clustering and topic modeling; introduce a useful synthetic dataset for historical document researchers; and present analyses that empirically show how existing algorithms break down in the presence of OCR errors.

Keywords: document modeling, document clustering, topic modeling, noisy data, OCR

ACKNOWLEDGMENTS

I would first like to thank my advisor, Dr. Eric Ringger for all his help in formulating the ideas, developing the methodology, analyzing the data and writing up the results presented throughout this dissertation. He has been patient with me during my many years as a PhD student and has helped me in countless ways with my research and in providing support and coordination to make sure that I always had the resources to get my research done. He has made himself available as a mentor and a friend and I am grateful that I have been able to have him as my guide through the world of academia. Dr. Seppi also deserves my deepest gratitude for stepping into a much more active role than a second committee member would typically have in order to help me finish this dissertation before I had to leave the state.

My lab mates have also been very helpful as sounding boards and proof readers over the years and I would like to thank them all. I would especially like to thank Robbie Haertel for his excellent feedback and math skills, Bill Lund for his reliability and willingness to share resources during our collaboration, Andrew McNabb for his openness in sharing his encyclopedic knowledge of Linux and Python, and Paul Felt for proof reading papers while professors were away and doing my legwork for me when I was too far from Provo to do it myself.

I would also like to thank Dr. C. Shane Reese for his feedback and help related to the techniques and mathematical derivations presented throughout the dissertation. Dr. Tony Martinez and Dr. Kent Seamons were great fourth and fifth committee members. I appreciate all of their help in making themselves available and in offering helpful suggestions that improved the quality of the dissertation.

Thanks also to Charles Elkan for his feedback on Chapter 2 and for pointing us to his work on deterministic annealing to improve document clustering with EM.

I would also like to acknowledge and thank the Fulton Supercomputing Center for providing the computing resources and maintenance required to run the experiments reported here.

The BYU Computer Science Department also deserves my thanks for supplying me with equipment, tuition, travel expenses and a stipend to support me during my time here. I also received support in the form of mentoring grants from the College of Physical and Mathematical Sciences, for which I am grateful.

There are many others who have been helpful, and I thank them all. My parents Daniel and Sherry Walker have helped in countless ways throughout my life. They gave me many opportunities to develop my skills and knowledge of computers during my childhood and through their encouragement gave me the confidence to set high goals and to believe that I could achieve them. My siblings, grandparents, in-laws and extended family have all been supportive and positive. I have been so blessed to be surrounded by such wonderful and giving people in my life. Thank you to all of you for everything you have done.

My children have also been amazingly supportive and understanding, from Zayaa who has been along for the whole journey through Danny, and Ariunaa who came along a little later and who do not know what it is like to not have a daddy in graduate school all the way to Ulzii, who won’t even remember what it was like to have a daddy in graduate school. I thank them for their love and their prayers. I hope that they will grow to appreciate this time and remember the importance of setting and achieving goals and of pursuing education and knowledge.

Of course, the most important help and support I have received during my time in school came from my beautiful and supportive wife, Choka. She has been a truly amazing companion for this journey and I can’t imagine how I could have done it without her. She has been very patient and very supportive beyond what the average person could endure. She has sacrificed so much to get us to this point and so I dedicate this work to her, knowing that I will never be able to fully repay what she has given to make it possible.

Table of Contents

Chapter 1

Introduction

The amount of available digital text data is ballooning at an incredible pace. For example, in 2010 Twitter, Inc. reported that its users were creating 50 million “tweets" per day[1], in 2006 the WikiMedia foundation announced that its users were creating new English articles at the rate of 1,700 daily[2], also in 2006 it was reported that Google Books was scanning books at a rate of 3,000 per day[3], and the Internet Archive scans 1,000 books each day[4]. Other documents are being made available through whistle-blower sites like wikileaks[5] or through the discovery process during civil and criminal trials.

Many of these documents have historical value and can and will be studied and analyzed in order to improve our understanding of past events and trends in order to enrich our understanding of history. To aid in the automation of the analysis of these mountains of data, algorithms can be used to create models of the latent (i.e., un-observed, or un-specified) semantic information present in a given corpus of text. One example of this type of analysis is document clustering, in which documents are grouped such that documents which discuss the same topics are clustered together. Another type of topic analysis attempts to discover the actual sub-topics being discussed in each document and determine which portions of the text correspond to each of the topics discussed therein. This type of analysis, commonly referred to as “topic modeling", has gained in popularity recently as inference on models containing large numbers of latent variables has become feasible due to advances in algorithms and computing capacity.

[1] http://blog.twitter.com/2010/02/measuring-tweets.html

[2] http://wikimediafoundation.org/wiki/Press_releases/English_Wikipedia_Publishes_Millionth_Article

[3] http://www.washingtonpost.com/wp-dyn/content/article/2006/12/20/AR2006122000213_pf.html

[4] http://www.archive.org/scanning

[5] http://wikileaks.org/

This dissertation focuses on document modeling and text analytics with the specific goal of establishing models and techniques that will be useful in the analysis of historical documents. There are two special considerations that we account for in our research which are endemic to the study of historical documents: noise and context. While historical provided the motivation for this work, the research presented in this dissertation is applicable to any dataset for which these are important characteristics.

Noise Historical documents often originate from a time when document creation and storage technologies were relatively primitive, and so we can often only obtain digital text versions of these documents through the use of Optical Character Recognition (OCR) conducted on images of documents with varying levels of degradation. Digitization of degraded documents results in text which contains OCR errors. Topic models rely on counts of the occurrences and co-occurrences of words. Errors such as those introduced by OCR distort counts and so are likely to interfere with the discovery of patterns in the data. Until recently it has not been well understood how the quality of these models degrade as the level of OCR noise increases. Recent results indicate that, as expected, both clustering and topic models degrade in quality rapidly as the word error rate (WER) increases towards 50%. My goal is to measure the extent to which OCR errors degrade the performance of document models and algorithms and to identify those cases in which modeling problems can be solved through preprocessing of the data to remove noisy tokens. Part of my research focuses on unsupervised models of latent document semantics including document clustering, topic modeling, and so-called “supervised" topic modeling.

Context Most document collections have metadata associated with the documents that can help put the topics in the document into a wider historical context. For example, the metadata could represent document creation dates, and a common task would be to track the evolution of topics discussed in the corpus over time. Context is especially important when analyzing historical document collections as the focus of the analysis of these documents often hinges on establishing a chronology, or discovering trends that can be synthesized into a better understanding of the events surrounding the creation of the documents. Recent research in later chapters studies the use of a class of topic models known as “supervised" topic models which simultaneously model topics and metadata.

Note that, though the specific motivation of this dissertation is to study and create models and algorithms that will be useful for the analysis of historical documents, there is nothing about this work which would preclude its application to any text document collection that may include noise in the form of OCR, handwriting recognition, voice recognition or other errors. Likewise, the work aimed at improving our understanding of historical documents in context can be applied to any document collection, clean or noisy, for which real-valued metadata exists and for which it is desirable to include that metadata in the modeling of the documents. It just so happens that many historical documents have these properties so, in the context of this dissertation, a historical document is simply a potentially noisy document that may be accompanied by real-valued metadata.

The research presented here furthers the goal of making text analytics useful for historical document analysis by: contributing positive results in model-based document clustering; creating a fast and effective feature selection technique to help account for noise; quantifying the effects of OCR errors on various text analytics tasks along with the ability of feature selection to compensate for these effects; introducing a new synthetic OCR dataset; and developing a new supervised topic model which possesses several desirable properties.

The remainder of this introduction is organized as follows: Section 1.1 outlines the basic concepts of Bayesian graphical models, introducing the various discrete distributions that are used as the building blocks of many Bayesian document models, including various results relating to the specific properties of these distributions, how they combine, and how inference is conducted with them. Section 1.2 presents a family of models that model the content of documents using a single latent variable per document and are used for document classification and clustering. Section 1.3 outlines a family of word-level document models that model documents using a distinct latent variable per word in each document. These models are called topic models. Section 1.4 introduces a special class of topic models that contain random variables related to real-valued metadata associated with each document. These models are called supervised topic models. Finally, Sections 1.6 and 1.7 present the thesis of this dissertation and a brief description of the contents of the remaining chapters, respectively.

1.1 Bayesian Modeling of Discrete Events

Many modern document analysis systems and algorithms rely heavily on statistical methods, specifically Bayesian statistical analysis. This section gives a brief overview of mathematical and graphical elements that will be helpful in understanding the graphical Bayesian models that are found throughout the dissertation.

1.1.1 Probabilistic Graphical Models

It is common for statistical models to be presented in graphical form. Figure 1.1 shows a very simple directed graphical model. Each node represents a random variable or parameter of the model. Directed edges are used to indicate dependence. In this example, we see that there are two random variables x and y. We indicate that y is an observed value by shading the corresponding node and that x is latent or unobserved by leaving that node unshaded. The way in which y depends on x is not shown in the graph, but can be stated explicitly using the “distributed as" syntax, for example:

$$y|x \sim \textit{Normal}(x, 0.5)$$

In English this statement is read “y given x is distributed normally with mean x and variance 0.5".

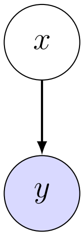
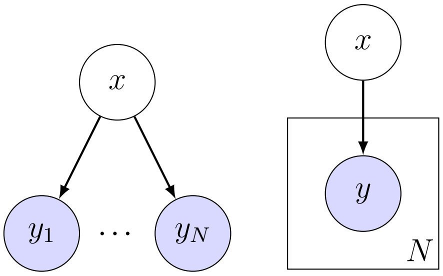


Figure 1.1: A very simple graphical model.



(a) Without plate notation.

(b) With plate notation.

Figure 1.2: Another simple graphical model without (a) and with (b) plate notation.

In document modeling, it is often the case that portions of the model graph are repeated many times. For example, most document models have at least one random variable for each word and several for each document resulting in potentially millions of random variables and thus nodes in the graph. To simplify diagrams of this size, a visual shorthand known as plate notation is used. Plates are indicated in a graph by a rectangle enclosing one or more nodes and by an extra variable at the bottom of the rectangle that indicates the number of times the plate is replicated. Figure 1.2 shows an example of a model in which there are N nodes labeled y, $(y_1, \ldots, y_N)$, drawn without (a) and with (b) a plate.

1.1.2 Conjugacy

Conjugacy is a desirable property to have in hierarchical Bayesian models. Intuitively, it is the idea of mathematical compatibility between distributions. When a random variable y depends on a random variable x in such a way that the distribution of x is the conjugate prior for the distribution of y, then the posterior distribution of x given y, $p(x|y)$ will have the same mathematical form as the original prior. The use of conjugacy can greatly simplify graphical models and the inference used to estimate values and distributions for their parameters and latent variables. A specific example of conjugacy is explained in the next section and in Section 1.1.4.

1.1.3 Discrete Distributions and Their Conjugate Priors

Unlike many other problems studied by statisticians, most of the phenomena related to language and document modeling are not real-valued. This means that, in place of gamma, beta, and normal distributions (the work-horses of statistics over continuous random variables), we instead use discrete distributions and their conjugate priors. This is because, though words are typically represented by numerical indices, they are nominal and not truly numerical in nature. So, though the word type “the" may be represented with the number 5 and the word type “ant" by the number 15 in a given model, it does not make sense to say that the sum of three occurrences of the word “the" is equal to one occurrence of the word “ant". Cluster and topic assignments similarly represent the category of an outcome and not numeric quantities. Nominal random variables that represent category assignments are called categorical random variables. Other examples of categorical random variables are the result of a die roll, the suit of a card drawn from a deck of cards, the color of a ball drawn from an urn, etc.

Categorical variables are discrete and typically finite in nature. That is, for categorical random variable $c = j, j \in (1, \ldots, K)$ is an index that arbitrarily assigns possible outcomes to a numerical index.

There are three event models or sampling distributions that are typically used to model the values of categorical random variables. These distributions rely on the discrete and finite nature of categorical random variables to fully specify the p.m.f. Let $\mathbf{c} = \{c_1, \ldots c_n\}$ be a series of n categorical random variables. All of the distributions below have the same vector-valued multinomial parameter θ where $p(c_i = j) = \theta$. The first two distributions deals directly with the individual categorical random variables, and as such will be referred to as Categorical distributions. The last treats the entire series at once, proposing a distribution over count vectors $C = (n_1, \ldots, n_K)$ where n_k is the number of random variables having outcome k. This last distribution is known as the multinomial distribution.

Categorical - Nominal Encoding

This is the most basic of the event models and is configured exactly as discussed above, **c** is a series of independent categorical random variables with $p(c_i = j) = \theta$. This yields likelihood:

$$\begin{aligned} l_c(\theta|\mathbf{c}) &= \prod_{i=1}^{n} \theta_{c_i} \\ &= \prod_{k=1}^{K} \theta_k^{n_k} \end{aligned}$$

Categorical - 1-of-K Encoding

This event model replaces the encoding of a single categorical outcome as a numerical index with a vector of length K consisting of all zeros except at the element corresponding to the numerical index of the outcome. So, the event $c_i = j$ leads to encoding vector $C = (C_1 = 0, \ldots C_j = 1, \ldots C_K = 0)$. One way of thinking about the 1-of-K encoding is that it is just a categorical where the outcome is a 1-of-K vector instead of 1 scalar integer. Another way is to interpret each vector as being the event count vector of a multinomial distribution with $n = 1$. The second interpretation is the most useful because, while nominal category indices cannot be manipulated by arithmetic, counts can be and so are able to be summed, averaged, etc.

For n 1-of-K encoded categorical variables $\mathbf{C} = C_1, \ldots, C_n$ we have $C_i \sim \textit{Mult}(1, \theta)$ which yields likelihood:

$$\begin{aligned}
{}_k(\theta|\mathbf{C}) &= \prod_{i=1}^{n} \frac{1!}{1^K} \prod_{k=1}^{K} \theta_k^{\mathbf{C}_{ik}} \\
&= \prod_{i=1}^{n} \prod_{k=1}^{K} \theta_k^{\mathbf{C}_{ik}} \\
&= \prod_{k=1}^{K} \prod_{i=1}^{n} \theta_k^{\mathbf{C}_{ik}} \\
&= \prod_{k=1}^{K} \theta_k^{\mathbf{C}_{1k}} \cdot \theta_k^{\mathbf{C}_{2k}} \cdots \theta_k^{\mathbf{C}_{nk}} \\
&= \prod_{k=1}^{K} \theta_k^{n_k}
\end{aligned}$$

Multinomial

As mentioned above, the multinomial distribution can be thought of as a distribution over categorical event outcome count vectors. More generally, the multinomial distribution is a distribution over vectors with integer valued elements greater than or equal to zero that are of a specific dimension and parameterized $\mathbf{L^1}$ norm n. The multinomial distribution explicitly takes into account all of the ways that n trials could occur to produce the count vector. For a count vector $C = n_1, \ldots, n_K$ as defined above, we say that $C \sim \textit{Mult}(n, \theta)$ with likelihood defined explicitly to be:

$$l_m(\theta|C) = \frac{n!}{\prod_{k=1}^{K} n_k!} \prod_{k=1}^{K} \theta_k^{n_k}$$

Since (as functions of θ) $l_c = l_k \propto l_m$, all three event models produce the same inference about θ under any inferential scheme that obeys the Likelihood Principal (e.g. Bayesian methods and many well behaved classical methodologies) [78]. This close relationship explains why it is common in the literature to call all of these distributions the multinomial[6]. We may often switch between the various event models, even in the same analysis, depending on the specific mathematical properties that are required at the moment.

The Dirichlet Distribution

The conjugate prior for the θ parameter for all three of the above event models is the Dirichlet distribution. The Dirichlet is a distribution over elements of a k dimensional simplex with a parameter vector of positive reals α and with probability density function:

$$p(\theta|\alpha) = \frac{1}{B(\alpha)} \prod_{i=1}^{k} \theta_i^{\alpha_i - 1} \tag{1.1}$$

where B is the Euler beta function:

$$B(\alpha) = \frac{\prod_{i=1}^{K} \Gamma(\alpha_i)}{\Gamma(\sum_{i-1}^{k} \alpha_i)} \tag{1.2}$$

An intuitive interpretation of the α parameter is as a set of prior event counts plus one. A common choice for the parameter of a Dirichlet prior is the so-called “uninformative" prior in which each element of α is set to one. This yields a uniform distribution over the entire simplex.

Because of its simplicity and because it serves as a model for many of the manipulations that are necessary to perform inference efficiently using these distributions, the proof of conjugacy between the Dirichlet and multinomial distributions is given below in Section 1.1.4.

1.1.4 Dirichlet/Multinomial Conjugacy

Assume that a vector of observations $\mathbf{y}$ have been drawn from a multinomial distribution with parameters θ and n, Also, assume that θ is unknown, but is distributed Dirichlet with parameter

[6] See the article at `http://en.wikipedia.org/wiki/Categorical_distribution` for a more in-depth description of the Categorical distribution and its relation to the multinomial distribution.

vector α, then the posterior distribution over θ can be expressed as follows:

$$\begin{aligned}
p(\theta|\mathbf{x},\alpha) &= \frac{p(\mathbf{x}|\theta)p(\theta|\alpha)}{p(\mathbf{x})} \\
&\propto p(\mathbf{x}|\theta)p(\theta|\alpha) \\
&\propto \frac{n!}{\prod_i \mathbf{x}_i!}\prod_i \theta_i^{\mathbf{x}_i} \cdot \frac{1}{B(\alpha)}\prod_{i=1}^{k}\theta_i^{\alpha_i-1} \\
&\propto \prod_{i=1}^{k}\theta_i^{\mathbf{x}_i}\prod_{i=1}^{k}\theta_i^{\alpha_i-1} \\
&\propto \prod_{i=1}^{k}\theta_i^{\mathbf{x}_i+\alpha_i-1}
\end{aligned}$$

At this point we note the form of this un-normalized distribution and its similarity to the Dirichlet distribution, from which we know that:

$$\begin{aligned}
\int_\theta \prod_{i=1}^{k}\theta^{\mathbf{x}_i+\alpha_i-1} =& B(\alpha_*) \\
\frac{1}{B(\alpha_*)}\int_\theta \prod_{i=1}^{k}\theta^{\mathbf{x}_i+\alpha_i-1} =& 1
\end{aligned}$$

where $\alpha_{*i} = x_i + \alpha_i$, so the posterior of θ is

$$p(\theta|\alpha,\mathbf{x}) = \frac{1}{B(\alpha_*)}\prod_{i=1}^{k}\theta_i^{\alpha_*-1}$$

and therefore

$$\theta|\alpha,\mathbf{x} \sim \textit{Dir}(\alpha_*)$$

This demonstrates the conjugacy of the Dirichlet and multinomial distributions and shows how the compatibility between the two distributions relies on the common base in the exponentiation present in the mass/density functions.

A common outcome in Dirichlet-multinomial models is to arrive at complete conditionals that are the ratio of two Euler Beta functions where the vector argument of one of the Betas is equal to the other except with some integral amount added to each element. These ratios often simplify nicely, as discussed in Appendix A.

1.1.5 Mixture Modeling

Statistical model-based clustering methods are often based on the assumption that the entire sample of instances was actually produced from a combination of k distinct distributions. A model structured in this way is called a mixture model. Each of the individual distributions in the mixture is called a component of the mixture. It is common for each component i of a mixture model to be from the same family of distributions with unique parameters Θ_i. In addition to the mixture components, mixture models also include a mixing distribution over the component distributions. A typical mixture model has the following form:

$$x \sim \sum_{i=1}^{k} p(x|\Theta_i)p(\Theta_i|\pi)$$

where $p(\Theta_i|\pi)$ is the mixing distribution over the mixture components and $p(x|\Theta_i)$ is the ith component distribution.

Mixture models can also be equivalently expressed as hierarchical models by adding categorically distributed indicator variables that “assign" observations to a single component:

$$x_d \sim \sum_{i=1}^{k} p(x|\Theta_i)p(c_d = i|\pi) \text{ and}$$

$$x_d|c_d \sim p(x|\Theta_{c_d})$$

So, to sample instance d from a mixture model, one first samples a topic assignment $c_d = j$ with probability π_j and then an instance from the distribution for component j with parameters Θ_j.

1.1.6 Bayesian Nonparametrics

The distributions in the statistical models discussed up to this point have a common characteristic: Their probability density and mass functions are expressible as functions parameterized with a finite number of real-valued or real vector-valued parameters. Distributions that have this property are referred to as parametric distributions. Most common distributions, including the members of the exponential family, are parametric.

Another class of distributions is defined over random variables in infinite dimensions. These distributions are typically called nonparametric distributions since there do not exist a finite number of parameters nor sufficient statistics that can describe them.

One example of a nonparametric distribution is the Dirichlet Process (DP). The DP is a distribution over probability measures. Let G be a distribution-valued probability measure. We say that $G \sim \textit{DP}(G_0, m)$ if G is a probability measure on state space Ω and for any finite partition $A_1, \ldots, A_k$ of Ω it is the case that:

$$(G(A_1), \ldots, G(A_k)) \sim \textit{Dir}(mG_0(A_1), \ldots, mG_0(A_k)),$$

where G_0 is a base measure and m is called the total mass parameter.

The DP is a point process and it can be shown that the G drawn from a DP distribution almost surely have the form:

$$G = \sum_{i}^{\infty} \delta_i \omega_i$$

where the $\omega_i \in \Omega$ are the *locations* of the point masses in the state space and δ_i are the *weights*, or amount of mass at the corresponding locations. Figure 1.3 shows a finite approximation of a draw from a DP with $G_0 = \textit{Normal}(3, 1)$ and $m = 1$

There is a large body of work on the DP, its theoretical properties, specific use cases, and inference for models that use it as a component [13, 31, 33, 67].

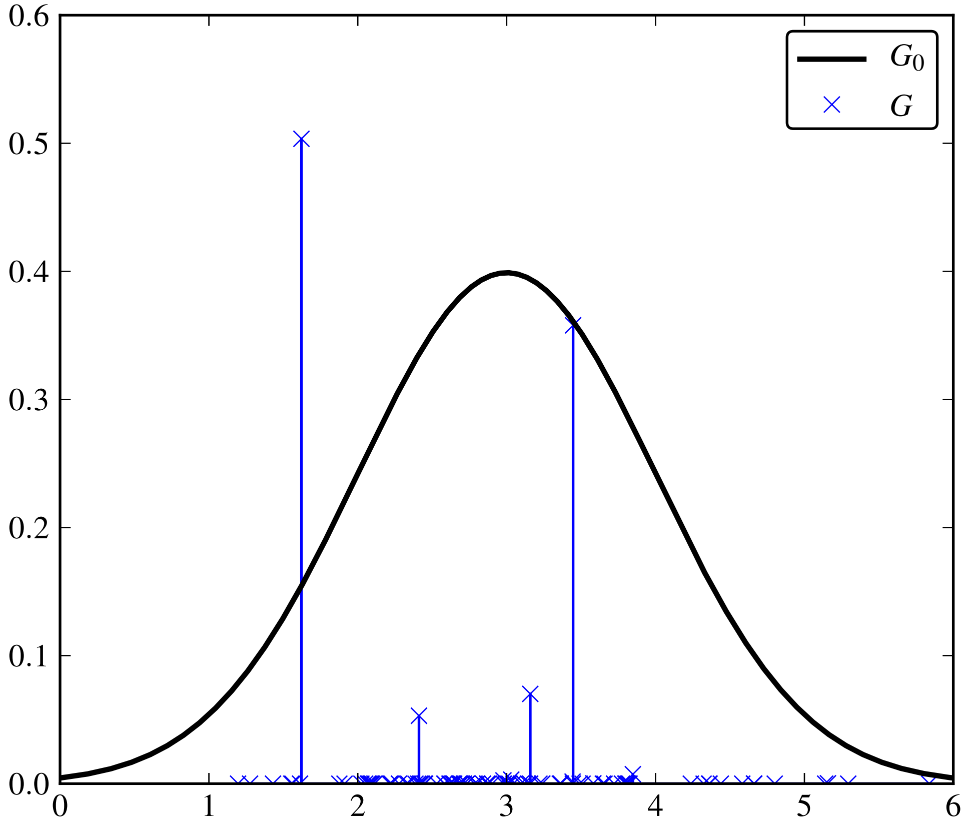


Figure 1.3: A finite approximation of a probability measure G drawn from a Dirichlet Process with $G_0 = \textit{Normal}(3, 1)$ and $m = 1$

1.1.7 Dirichlet Process Mixtures

As discussed above, many phenomena can be effectively modeled using a mixture of distributions. One issue that arises when modeling data with a mixture model is that, in a classic mixture model, one must choose the number of components in the mixture up front. Mixtures consisting of a fixed number of pre-specified components are called finite mixture models.

In cases where the appropriate number of components is not known in advance, it is possible to model the data as an infinite mixture model with a DP prior on the mixing distribution of the components. This is equivalent to convolving the DP with a continuous density. A distribution of this form is known as a Dirichlet Process Mixture (DPM).

Values for the parameters are drawn according to G_0 and the mixing distribution over the components is proportional to the number of data point that have been assigned to that component. In addition to “existing" components (defined as components that have data points assigned to them), new data points can also be assigned to a “new" component, with probability proportional to m. Although the DPM is described as an infinite mixture model, in reality it is not possible to estimate an infinite-dimensional model given finite data. So, in practice only finite approximations are inferred, and the components that have no data points assigned to them are marginalized out of the likelihood.

1.1.8 Inference

For simple statistical models, it is sometimes possible to find closed-form solutions for maximum likelihood estimates of model parameters, or for the posterior distributions for all variables of interest given data. However, even when conjugacy is used to simplify the forms of joint and posterior distributions, it often becomes impossible to do inference exactly or in closed form. In these cases, it is necessary to use approximate inference techniques.

One common approximate inference algorithm is the Gibbs sampler [34, 75]. The essential component of a Gibbs sampler is the complete conditional. Given a set of observed data $\mathbf{y}$ and a set of un-observed parameters and other latent variables $\mathbf{\Theta} = (\theta_1, \ldots, \theta_m)$, the complete conditional

```
input  : A set of data points $D$, a set of unknown variables $\mathbf{\Theta}$ and the desired number of
         samples $l$
output: A vector of sampled values for $\mathbf{\Theta}$, $(\mathbf{\Theta}^{(1)}, \ldots, \mathbf{\Theta}^{(l)})$
Randomly select values for $\mathbf{\Theta}^{(0)}$ $i \leftarrow 1$
while $i \leq l$ do
    $j \leftarrow 1$
    while $j \leq m$ do
        $\theta_j^{(i)} = draw(p(\theta_j | \mathbf{y}, \mathbf{\Theta}_{-\theta_j}))$
        $j \leftarrow j + 1$
    end
    $i \leftarrow i + 1$
end
```

Algorithm 1: Collapsed Gibbs sampling algorithm.

of random variable $\theta \in \mathbf{\Theta}$ is the posterior distribution of θ given all other random variables $p(\theta|\mathbf{y}, \mathbf{\Theta}_{-\theta})$, where $\mathbf{\Theta}_{-\theta}$ is the set of all variables in $\mathbf{\Theta}$ except θ. When one or more variables is marginalized or integrated out before finding the complete conditionals, the resulting sampler is called a collapsed Gibbs sampler.

Gibbs samplers are guaranteed to converge in distribution to the true joint posterior distribution of the random variables. However, Gibbs samplers are approximate in practice because true convergence in distribution only occurs as the number of samples approaches infinity. In addition, as models become more complex, they typically require an increasing number of samples in order to suitably approximate the distribution of interest. As a result, samplers for most document models, with their huge numbers of random variables and parameters, usually do not produce very close approximations of the true joint posterior.

Another method that is commonly used for inference in document models is Variational Bayesian inference (c.f. [15, 16]). Variational methods are a family of methods that replace complex statistical models with slightly less complex models that can be tractably dealt with in close form. Gradient descent methods are then used to find a parameterization of the simple model that minimizes the Kullback-Leibler divergence between the simplified model and the “true" document model. Variational methods are often difficult to set up, as the math required to find the model update formulas can be quite complex, and different for each model. However, since variational

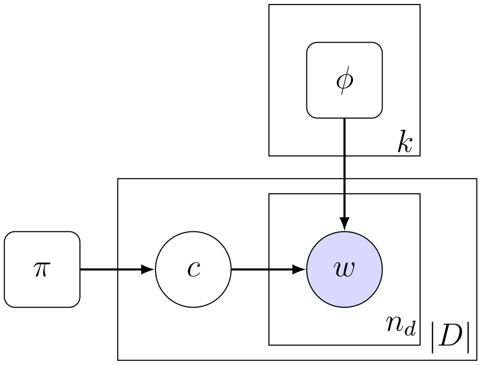


Figure 1.4: The Naive Bayes model.

methods are not stochastic, they are often much faster than sampler based inference methods. In addition, their deterministic nature can be a desirable trait.

1.2 Models of Latent Document-Level Semantics

This section presents some of the most influential and relevant models that have been studied by researchers in the field for document clustering and both supervised and unsupervised topic modeling.

1.2.1 Naive Bayes Model and Family

One of the simplest document models is the Naive Bayes (NB) model, which was introduced for document classification in the mid-1990s, and formally specified and analyzed by McCallum and Nigam in 1998 [60]. The NB model as traditionally used is shown in Figure 1.4: it assumes that each document $d \in D$ was generated by first choosing a class c_d for that document according to a distribution parameterized by π and then by choosing type values for the n_d words $w_{d1}, \ldots, w_{dn_d}$ in that document independently, according to a distribution parameterized by a class-specific parameter ϕ_{c_d}, the parameter vector for the c_d-th categorical distribution. More formally, $\forall d \in D$

and $\forall(d, i)$ *s.t.* $i \leq n_d$:

$$c_d|\pi \sim \textit{Categorical}(\pi)$$

$$w_{di}|\mathbf{c}, \phi \sim \textit{Categorical}(\phi_{c_d})$$

Note that no priors are specified for π or ϕ. The model assumes that these variables are not random quantities, but are instead fixed (but unknown) quantities, which we indicate in our graphical notation with the use of rounded squares for the corresponding nodes.

The NB model is naive in that it assumes that all of the words in a document are independent from one another given the document's class. This is clearly an incorrect assumption for natural language text as there are correlations between the words used together in documents. For example, an article that contains the word “York" is more likely to contain the capitalized word “New" than not.

Despite the fact that it is based on a faulty assumption, the NB model is popular because it is simple to implement and also because the parameters of a trained model yield interpretable insights about patterns in the data. Further, in practice the model performs well in document classification tasks, because occurrences of words in isolation provide sufficient evidence for topical categorization. For example, imagine the task of dividing a set of documents into two classes, one that contains documents about theoretical physics and another about animal husbandry. If a document is presented in this scenario that contains 20 occurrences of the word “quark", that is typically sufficient evidence to place it in the first category, despite what any of the other words in that document might be.

When used for supervised classification the class labels (values of the elements of c) are given for a training set, and point-valued estimates $\hat{\pi}$ and $\hat{\phi}$ are found for the model parameters using the complete (labeled) data. Then, for the prediction of a class label for an incomplete datum $\mathbf{w}_d$, these point estimates are treated as known values for the parameters, and Bayes Law is applied to determine the posterior class probability of label j, $p(c_d = j|\mathbf{w}_d, \hat{\pi}, \hat{\phi})$, for each j, and the j that maximizes this value is selected as the output label prediction for the classifier. Note that this is not a fully Bayesian model, as the primary parameters π and ϕ are not treated as random variables, but as fixed but unknown quantities to be estimated.

Meilă and Heckerman also explored the use of the NB model for unsupervised document clustering [63]. In the case of clustering, values for π and ϕ can be initialized randomly, and then the Expectation Maximization (EM) algorithm is used to iteratively improve the estimates for these parameters. EM has the desirable property that it is a non-decreasing optimization of the likelihood of the data. However, most interesting document likelihood distributions are multi-modal, and so EM is only locally optimal, and only converges to the global maximum given a “lucky" initialization.

One technique that mitigates this problem is deterministic annealing. Deterministic annealing smooths the likelihood surface during early stages of EM, giving the algorithm a better chance of reaching a point nearer to the global optimum [106].

Another way to avoid local maxima is to employ an inference algorithm that naturally explores a larger portion of the likelihood surface. In Chapter 2, we show that a clustering algorithm based on a mixture model with multinomial component distributions over words together with a collapsed Gibbs sampler outperforms standard EM significantly. In order to use the Gibbs sampler we must expand the NB model to be a fully Bayesian model by specifying prior distributions (with their accompanying parameters) for the π and ϕ vectors. A model which incorporates the priors is shown in Figure 1.5. The model uses Dirichlet distributions as priors to take advantage of Dirichlet-multinomial conjugacy:

$$\pi|\alpha \sim \textit{Dirichlet}(\alpha) \qquad \phi|\beta \sim \textit{Dirichlet}(\beta)$$

$$z_d|\pi \sim \textit{Categorical}(\pi) \qquad w_{di}|z_d, \phi \sim \textit{Categorical}(\phi_{z_d})$$

This model is called the Mixture of Multinomials document model. Note that the cluster membership node is now labeled z to indicate that it is a latent cluster and not a human-specified class.

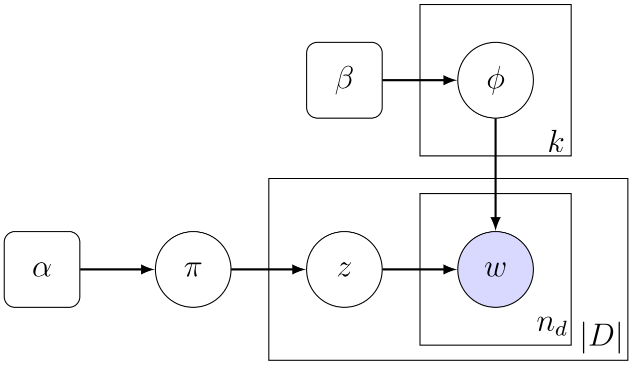


Figure 1.5: The Mixture of Multinomials model.

A Gibbs sampler is theoretically guaranteed to converge in distribution to the distribution of interest. As a result, the sampler will visit all of the supported regions of the posterior manifold in proportion to each region's posterior density. So, it should be possible to approximate the posterior distribution over the model parameters given the data. In practice, however, the sampler often becomes “stuck" due to spikes in the complete conditional distributions and so is still prone to become trapped by local maxima once converged.

1.3 Topic Models: Models of Word-Level Latent Semantics

The Naive Bayes and the more general Mixture of Multinomials document models treat documents as the basic unit of analysis. They have a single latent topic random variable per document. Another class of models works at a more fine-grained level, attempting to model all of the dominant topics that comprise each document by labeling individual words and phrases with topical assignments. These models are called *topic models*.

1.3.1 Latent Dirichlet Allocation

The most famous and well-studied example of a topic model is the Latent Dirichlet Allocation model (LDA) [16]. LDA is shown as a graphical model in Figure 1.6. The random variables of the

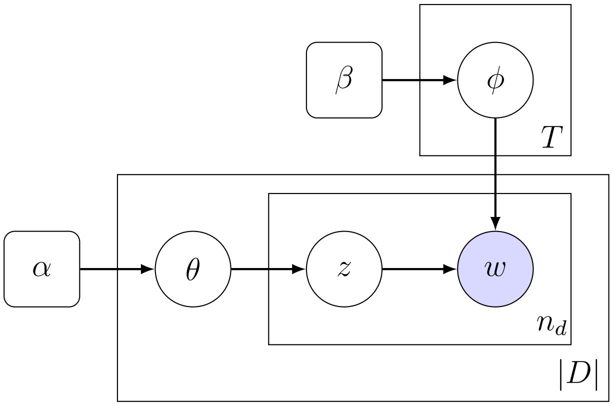


Figure 1.6: The LDA topic model.

model are distributed as follows:

$$\theta_d|\alpha \sim \textit{Dirichlet}(\alpha) \qquad \phi_t|\beta \sim \textit{Dirichlet}(\beta)$$

$$z_{di}|\theta \sim \textit{Categorical}(\theta_d) \qquad w_{di}|z \sim \textit{Categorical}(\phi_{z_{di}})$$

Although the graphical models for the Mixture of Multinomials and LDA models as shown in Figures 1.5 and 1.6 appear similar and even have the same number of nodes as depicted, there are some important difference that make parameter estimation more difficult for the LDA model than the Mixture of Multinomials. First, though it is obscured somewhat by the plate notation, there are many more random variables in LDA, as each individual token in the corpus has a topic label. Also, the per-document random variable in LDA (θ) is a vector of reals, of dimensionality T, instead of a single discrete value. These factors and others lead to the result that there is no closed-form solution for the maximum likelihood estimate of the parameters of an LDA model.

The two main methods that have been used to conduct inference with the LDA model are Variational inference and Markov chain Monte Carlo algorithms, such as the Gibbs sampler (see Section 1.1.8).

One of the most important properties of LDA is its relative simplicity both mathematically and conceptually. This simplicity increases the model's usefulness in that the trained model parameters can easily be interpreted through simple visualizations, which are typically included

Topic 0	**Topic 30**	**Topic 36**	**Topic 66**	**Topic 122**
printer	game	god	water	power
fonts	team	religion	energy	circuit
print	goal	christian	nuclear	output
postscript	play	atheist	wpi	voltage
font	leafs	faith	plants	chip
hp	games	christianity	cooling	current
printing	win	people	power	input
laser	blues	belief	air	led
printers	detroit	beliefs	heat	light
laser	player	religions	temperature	ca

Figure 1.7: A visualization designed to show the quality of the topics discovered using a run of LDA on the publicly available corpus of Enron e-mails with 200 topics.

in the evaluation sections of papers on topic modeling to provide qualitative verification of model quality. Figure 1.7 shows an example of this type of visualization, in which a few topics are chosen by the author and for each topic t the top N words are displayed, sorted by $p(w|t)$, the conditional probability of a word w given the topic label. When topics are of high quality, the words in the list for each topic should be recognizably related to one another and to an identifiable concept or topic.

Another benefit of LDA's simplicity is that the model is quite easily expanded and adapted. As a result, LDA has been used for many purposes and has been extended in a variety of ways. For example, Blei, et al. created a hierarchical version referred to as hLDA [17]. This idea was later expanded to allow for more complex relationships between topics as defined by an arbitrary directed acyclic graph in the Pachinko Allocation Model [53].

In addition, the basic “vanilla" LDA model has been extended in many ways to accomplish more complex tasks in addition to simple topic modeling, including topical n-grams [100], joint topic and author modeling [80], topics over time [99], joint semantic and syntactic topic modeling [40], and many more variations, including ones that use the DCM rather than the categorical distribution as in LDA [27, 64].

1.4 Supervised Topic Models: Joint Models of Word-Level Latent Semantics and Metadata

The models described to this point take into account only the words that make up a document collection. In many cases, however, real-world datasets include extra information together with the documents in the form of real-valued metadata. Supervised topic models are a class of topic models which include this metadata as part of the model in order to jointly model the text and the metadata.

There are two main reasons for including metadata in a topic model:

Prediction Given a trained model and a new document with missing metadata, a supervised topic model allows one to predict the value of the metadata variable for that document.

Analysis In order to understand a document collection better, it is often helpful to understand how the metadata and topics are related. For example, one might want to analyze the development of a topic over time, or investigate what the presence of a particular topic means in terms of the sentiment being expressed by the author. One may, for example, plot the distribution of the value that the metadata takes on given a topic from a trained model.

The main models that can be used to accomplish both of these tasks easily are the Topics Over Time model (TOT) [99] and the Supervised LDA model (sLDA) [15]. TOT was first proposed as a way to include temporal information in a topic model. In the generative model for TOT, after generating the topic for each token, both a token type and a time-stamp are generated. The word type is drawn according to a topic-specific categorical distribution just as in vanilla LDA, and the time-stamp is drawn from a topic-specific beta distribution.

sLDA was proposed as a general-purpose supervised topic model. sLDA uses a generalized linear model to model a single document metadata value which could be a time-stamp, numeric product rating, or any real-valued metadata.

Another model that has been proposed in the supervised topic model family is the Dirichlet Multinomial Regression model (DMR) [65]. The DMR has been shown to model data quite well as measured by perplexity, but it models the metadata in a conditional way (as opposed to generatively), which makes it much less convenient for the prediction and analysis usages listed above.

1.5 Feature Engineering

Feature engineering is the process of choosing features for a set of data points in order to achieve the best possible results at a machine learning task. In some cases, feature engineering means computing functions of observed features, or making new measurements about the instances in a dataset. We will not deal directly with these kinds of feature engineering operations. Another type of feature engineering starts with a large set of candidate features, and then selects only those which are useful for achieving good results on the given learning task. This type of feature engineering is known as *feature selection*.

Feature selection is a natural fit for document modeling because, in a machine learning sense, the words that make up a document in the above mentioned text analytics tasks are the features of the document. In the context of a topical analysis of a document collection, many of these features will be noisy, in the sense that they do not correlate directly with the topical content of the documents. For example, many words in natural language serve purely syntactical or functional roles in the language; the words “the", “of", “an", and “or", for example, are used with nearly equivalent frequency in documents treating any topic. Any apparent correlations between these words and the topics present in the document collection is most likely random and spurious. Because of this, it is usually possible to simply remove these function words from consideration when modeling documents and achieve results as good as, or better than those achieved when leaving them in.

In the case of supervised learning tasks, it is often possible for the learning algorithm to distinguish between features that are and features that are not useful for distinguishing topical classes, and so feature selection becomes important mostly in achieving faster learning times or more space-efficient models [103]. In the case of unsupervised learning tasks, such as document clustering and topic modeling, however, there are no labels to help the learning algorithms distinguish between significant and spurious patterns in the text, and so modeling text collections without first conducting feature selection can lead to relatively poor results. So, feature selection becomes extremely important when learning document clustering or topic models. At the same time, feature selection in these cases becomes more difficult because there are no labels with which to correlate the features in order to assess their value. So, in many ways, unsupervised feature selection itself is a very interesting and difficult problem. Chapter 3 presents more details about unsupervised feature selection and shows how it can improve results for document clustering. Chapters 4 and 7 show how unsupervised feature selection can improve results for topic modeling and supervised topic modeling, even in the presence of OCR errors.

1.6 Thesis Statement

Clustering and topic modeling of documents, which may contain noise and may include metadata, can be improved through feature engineering and Bayesian methods such as collapsed Gibbs sampling and Bayesian nonparametrics.

1.7 Dissertation Organization

The following list describes the chapters which comprise the remainder of this dissertation and demonstrate the claimed thesis.

Chapter 2 explores the use of a collapsed Gibbs sampler to find document clusters using a mixture of multinomials document model. The majority of the work in this chapter was published in KDD 2008 [91]. Since publication, several sections of follow-up work have been completed and appear in an addendum. This chapter demonstrates the effectiveness of a fully Bayesian approach, together with a stochastic sampling algorithm to produce high quality document clusters which are better than can be produced with expectation maximization or variational methods.

Chapter 3 introduces Top-N Per Document (TNPD), an unsupervised feature selection algorithm that chooses features on a per-document basis to be included in a global feature vocabulary. Words that do not make it into the vocabulary are culled from all documents. The technique is very fast and competitive with slower, more complex, algorithms in performance. The work in this chapter was published as technical report by the BYU Natural Language Processing Lab in July 2010 [92]. We found TNPD to be quite effective at removing both the natural noise that is a part of all human language (in the form of function words and other language phenomena that are not directly related to topical contents) and noise introduced through OCR errors and it is used throughout the dissertation whenever unsupervised feature selection is required.

Chapter 4 establishes that OCR errors significantly impact the quality of document models. Our findings shed significant light on the robustness of clustering and topic models in the context of OCR data as well as on the effectiveness of standard pre-processing techniques to correct for model deficiencies with this type of data. The work in this chapter was published in the proceedings of the 2010 Conference on Empirical Methods in Natural Language Processing [93].

Chapter 5 introduces a corpus of synthetic document datasets to correct the significant lack of datasets suitable for many kinds of research that require text data digitized through OCR. Our datasets were created with the intent that they be useful in a variety of research scenarios. We reached this goal and demonstrate several desirable properties and the usefulness of the datasets in several ways. The work in this chapter was published in the proceedings of Document Recognition and Retrieval XIX [94].

Chapter 6 introduces the Topics Over Nonparametric Time (TONPT) model, together with the necessary equations for inference using a collapsed Gibbs sampler. The model is compared in performance against the existing Topics Over Time and Supervised LDA supervised topic models, as well as against a baseline consisting of a vanilla LDA analysis of the data that has been augmented with a linear regression model that relates topic proportions to metadata values post-hoc. This work was published in the proceedings of the 9th Bayesian Modelling Applications Workshop [95].

Having introduced TONPT, Chapter 7 applies that model as well as other existing supervised topic models to the task of modeling noisy OCR text data. We investigate the robustness (both absolute and relative to other models in the same family) of these models to OCR errors and evaluate to what extent degradations in performance can be corrected through feature selection or other data pre-processing techniques. At the time of this writing, Chapter 7 has been accepted for publication in the proceedings of Document Recognition and Retrieval XX [96].

Chapter 2

Model-Based Document Clustering with a Collapsed Gibbs Sampler

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. Additional results show that the Gibbs and EM algorithms are sensitive to hyperparemeter settings. Also, a comparison is made to a variational Bayesian inference algorithm, to a random restart version of EM and to deterministically annealed versions of the base inference algorithms. It was found that annealing almost always improved performance. Additionally, results indicate that EM and the variational Bayesian algorithms perform similarly both in the standard and annealed versions, though the collapsed Gibbs sampler with annealing usually performs significantly better than either.

2.1 Introduction

As discussed in Chapter 1 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 [9, 41, 83, 104, 105]. 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 [83], 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 chapter empirically investigates many of these questions and hopefully serves 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.

The empirical effects on cluster quality of using various sample summarization methods will be shown, and it will be demonstrated 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.

Additionally, an addendum has been added to this version of the work which explores the effect of alternative parameterizations of the model on the quality of the produced clusterings. Also in the addendum, the concept of deterministic annealing is discussed and experimental results are presented which demonstrate how annealing can improve the clusterings produced using a collapsed Gibbs sampler, EM and variational Bayesian inference.

2.2 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 [63, 77], 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 2.3.

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 to be superior to some vector-space solutions, such as agglomerative clustering [63]. However, as a hill-climbing algorithm, EM is subject to becoming trapped by local maxima in the likelihood surface [16]. 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 [83, 104]. 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 [37]. 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. Although MCMC-based document clustering techniques have been compared with EM-based techniques in the past by Banerjee and Basu [9], 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.

2.3 Model

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

$$\begin{aligned}
\boldsymbol{\phi}_j \sim & \quad Dirichlet(\boldsymbol{\beta}) \\
\boldsymbol{\pi} \sim & \quad Dirichlet(\boldsymbol{\alpha}) \\
\boldsymbol{z}_d \sim & \quad Categorical(\boldsymbol{\pi}) \\
\boldsymbol{w}_d | \boldsymbol{z}_d \sim & \quad Multinomial(\boldsymbol{\phi}_{j=\boldsymbol{z}_d})
\end{aligned}$$

This model is illustrated in Figure 2.1. Here, $\boldsymbol{w}$ is a matrix representing the words in all M documents in the data. Document d from cluster j consists of a row vector of N words $\boldsymbol{w}_d$,

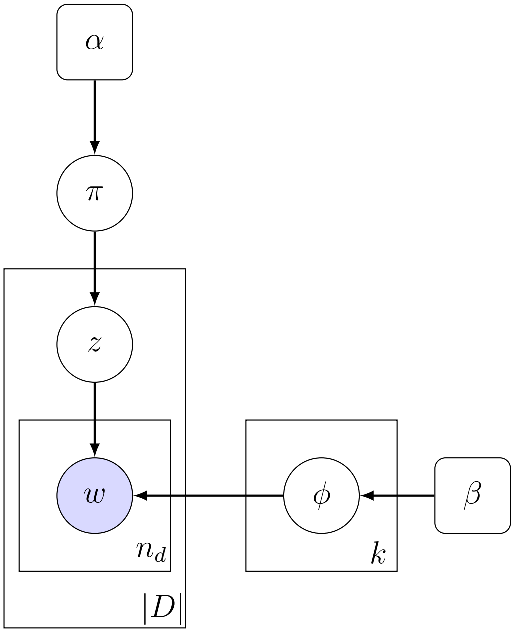


Figure 2.1: Graphical representation of the generative document model used in this paper for clustering. Square nodes are constants, shaded nodes are observed values.

distributed according to a multinomial distribution parameterized by the vector $\boldsymbol{\phi}_j$. Assuming that $\boldsymbol{z}_d = j$, $\boldsymbol{\phi}_{j,v}$ is the probability that word i of document d is the word v in the vocabulary V:

$$p(\boldsymbol{w}_{d,i} = v | \boldsymbol{z}_d = j) = \boldsymbol{\phi}_{j,v}$$

Likewise, each cluster label $\boldsymbol{z}_d$ is distributed according to a categorical distribution with parameter vector $\boldsymbol{\pi}$. This means that the prior distribution on the value of the cluster label $\boldsymbol{z}_d$ is given by:

$$p(\boldsymbol{z}_d = j) = \boldsymbol{\pi}_j$$

Since both $\boldsymbol{\pi}$ and $\boldsymbol{\phi}$ 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 $\boldsymbol{\alpha}$ and $\boldsymbol{\beta}$, respectively) to all ones. The uninformative distribution was chosen because we assume that *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 categorical and multinomial distributions.

Finally, it should be noted that the model includes an unexpressed parameter K, the number of mixture components. Other work has explored model selection [63] and automatic estimation of this parameter using a Dirichlet process prior [67]. 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.

2.4 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 $\boldsymbol{z}$. There are valid reasons to sample the values for the $\boldsymbol{\phi}$ matrix as well, but this is not strictly part of the clustering task. Furthermore, sampling Dirichlets in the number of dimensions required in this case presents complications as machine precision can result in a large number of samples in which some components of the sampled vector are zero, which leads to degenerate sampling distributions.

Perhaps more compelling than 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 [22, 39, 88].

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 2.1.

After marginalizing out $\boldsymbol{\phi}$ and $\boldsymbol{\pi}$, the complete conditional over the label z_d for document d is:

$$p(z_d = j|\boldsymbol{w}, \boldsymbol{z}_{-d}) = \frac{p\left(\boldsymbol{z}, \boldsymbol{w}\right)}{p\left(\boldsymbol{z}_{-d}, \boldsymbol{w}\right)} \tag{2.1}$$

where $\boldsymbol{z}_{-d} = [\boldsymbol{z}_1, ..., \boldsymbol{z}_{d-1}, \boldsymbol{z}_{d+1}, ...\boldsymbol{z}_M]$. We also have omitted $\boldsymbol{\alpha}$ and $\boldsymbol{\beta}$ from the conditions in each term for the sake of brevity. The right-hand-side of Equation (2.1) can also be expanded by explicitly showing that $\boldsymbol{z}_d = j$ giving:

$$p(z_d = j|\boldsymbol{w}, \boldsymbol{z}_{-d}) = \frac{p\left(\boldsymbol{z}_d = j, \boldsymbol{z}_{-d}, \boldsymbol{w}\right)}{p\left(\boldsymbol{z}_{-d}, \boldsymbol{w}\right)} \tag{2.2}$$

In order to continue the derivation, it is necessary to have the marginalized joint distribution $p(\boldsymbol{z}, \boldsymbol{w})$.

The derivation that produces the marginal joint uses the following variables and is similar to the derivation for a related model, as presented by Safiei and Milios [83].

M	Number of documents in data set
N_d	Length of document d
$\boldsymbol{\phi}_x$	The x^{th} column of $\boldsymbol{\phi}$
$\boldsymbol{w}_{dn}$	The n^{th} word of document d
V	the vocabulary
K	The number of clusters/mixture components
n_k	$\sum_{d}^{M} \delta\left(\boldsymbol{z}_d, k\right)$, the number of documents labeled k.
n_{kv}	$\sum_{d}^{M} \sum_{n}^{N_d} \left(\delta\left(\boldsymbol{z}_d, k\right) \cdot \delta\left(\boldsymbol{w}_{dn}, v\right)\right)$, the number of times word v occurs in documents with the label k.

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

$$p\left(\boldsymbol{\pi}, \boldsymbol{z}, \boldsymbol{w}, \boldsymbol{\phi} | \boldsymbol{\alpha}, \boldsymbol{\beta}\right) = p\left(\boldsymbol{\pi} | \boldsymbol{\alpha}\right) \prod_{k=1}^{K} p\left(\boldsymbol{\phi}_k | \boldsymbol{\beta}\right) \prod_{d=1}^{M} \left(p\left(\boldsymbol{z}_d | \boldsymbol{\pi}\right) \prod_{n=1}^{N_d} p\left(\boldsymbol{w}_{dn} | \boldsymbol{z}_d, \boldsymbol{\phi} \boldsymbol{z}_d\right) \right) \tag{2.3}$$

Next, integrate out $\boldsymbol{\pi}$ and $\boldsymbol{\phi}$.

$$p\left(\boldsymbol{z}, \boldsymbol{w} | \boldsymbol{\alpha}, \boldsymbol{\beta}\right) = \int \ldots \int p\left(\boldsymbol{\pi} | \boldsymbol{\alpha}\right) \prod_{k=1}^{K} p\left(\boldsymbol{\phi}_k | \boldsymbol{\beta}\right) \prod_{d=1}^{M} \left(p\left(\boldsymbol{z}_d | \boldsymbol{\pi}\right) \prod_{n=1}^{N_d} p\left(\boldsymbol{w}_{dn} | \boldsymbol{z}_d, \boldsymbol{\phi} \boldsymbol{z}_d\right) \right) d\boldsymbol{\pi}, d\boldsymbol{\phi}_1, \ldots, d\boldsymbol{\phi}_K \tag{2.4}$$

Expand the nested product

$$= \int \ldots \int p\left(\boldsymbol{\pi} | \boldsymbol{\alpha}\right) \prod_{k=1}^{K} p\left(\boldsymbol{\phi}_k | \boldsymbol{\beta}\right) \left(\prod_{d=1}^{M} p\left(\boldsymbol{z}_d | \boldsymbol{\pi}\right) \right) \prod_{d=1}^{M} \prod_{n=1}^{N_d} p\left(\boldsymbol{w}_{dn} | \boldsymbol{z}_d, \boldsymbol{\phi} \boldsymbol{z}_d\right) d\boldsymbol{\pi}, d\boldsymbol{\phi}_1, \ldots, d\boldsymbol{\phi}_K \tag{2.5}$$

and then group like terms

$$= \int p\left(\boldsymbol{\pi} | \boldsymbol{\alpha}\right) \prod_{d=1}^{M} p\left(\boldsymbol{z}_d | \boldsymbol{\pi}\right) d\boldsymbol{\pi} \int \ldots \int \prod_{k=1}^{K} p\left(\boldsymbol{\phi}_k | \boldsymbol{\beta}\right) \prod_{d=1}^{M} \prod_{n=1}^{N_d} p\left(\boldsymbol{w}_{dn} | \boldsymbol{z}_d, \boldsymbol{\phi} \boldsymbol{z}_d\right) d\boldsymbol{\phi}_1, \ldots, d\boldsymbol{\phi}_K \tag{2.6}$$

Expand the multinomial and Dirichlet distributions

$$= \int B^{-1}(\boldsymbol{\alpha}) \prod_{j=1}^{K} \boldsymbol{\pi}_j^{\boldsymbol{\alpha}_j - 1} \prod_{d=1}^{M} \boldsymbol{\pi}_{\boldsymbol{z}_d} d\boldsymbol{\pi} \int \ldots \int \prod_{k=1}^{K} \left(B^{-1}(\boldsymbol{\beta}) \prod_{v=1}^{|V|} \boldsymbol{\phi}_{kv}^{\boldsymbol{\beta}_v - 1} \right) \prod_{d=1}^{M} \prod_{n=1}^{N_d} \phi_{\boldsymbol{z}_d, \boldsymbol{w}_{dn}} d\boldsymbol{\phi}_1, \ldots, d\boldsymbol{\phi}_K \tag{2.7}$$

Now, change the product indices from products over documents (d) and word sequences (n), to products over cluster labels (k) and word types (v)

$$= \int B^{-1}(\boldsymbol{\alpha}) \prod_{j=1}^{K} \boldsymbol{\pi}_j^{\boldsymbol{\alpha}_j - 1} \prod_{k=1}^{K} \boldsymbol{\pi}_k^{n_k} d\boldsymbol{\pi} \int \ldots \int \prod_{k=1}^{K} \left(B^{-1}(\boldsymbol{\beta}) \prod_{v=1}^{|V|} \boldsymbol{\phi}_{kv}^{\boldsymbol{\beta}_v - 1} \right) \prod_{k=1}^{K} \prod_{v=1}^{|V|} \phi_{kv}^{n_{kv}} d\boldsymbol{\phi}_1, \ldots, d\boldsymbol{\phi}_K \tag{2.8}$$

this is possible because $\prod_{d=1}^{M} \boldsymbol{\pi}_{\boldsymbol{z}_d} = \prod_{k=1}^{K} \boldsymbol{\pi}_k^{n_k}$ and $\prod_{n=1}^{N_d} \phi_{\boldsymbol{z}_d, \boldsymbol{w}_{dn}} = \prod_{v=1}^{|V|} \phi_{kv}^{n_{kv}}$, 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.

$$= B^{-1}(\boldsymbol{\alpha}) \int \prod_{j=1}^{K} \boldsymbol{\pi}_j^{\boldsymbol{\alpha}_j + n_k - 1} d\boldsymbol{\pi} \cdot B^{-K}(\boldsymbol{\beta}) \int \cdots \int \prod_{k=1}^{K} \prod_{v=1}^{|V|} \phi_{kv}^{\boldsymbol{\beta} + n_{kv} - 1} d\boldsymbol{\phi}_1, \ldots, d\boldsymbol{\phi}_K \tag{2.9}$$

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

$$p\left(\boldsymbol{z}, \boldsymbol{w} | \boldsymbol{\alpha}, \boldsymbol{\beta}\right) = B^{-1}(\boldsymbol{\alpha}) B(\boldsymbol{\alpha}_*) \cdot B^{-K}(\boldsymbol{\beta}) \prod_{k=1}^{K} B(\boldsymbol{\beta}_{k*}) \tag{2.10}$$

where we assume, for the purpose of computing counts, that $z_d = j$ and that,

$$\boldsymbol{\alpha}_* = \left[\boldsymbol{\alpha}_1 + n_1, ..., \boldsymbol{\alpha}_K + n_K\right],$$

$$\boldsymbol{\beta}_{k*} = \left[\boldsymbol{\beta}_1 + n_{1k}, ..., \boldsymbol{\beta}_{|V|} + n_{|V|k}\right],$$

$$\boldsymbol{\alpha}_{-d*} = \left[\boldsymbol{\alpha}_1 + n_1, ..., \boldsymbol{\alpha}_j + n_j - 1, ..., \boldsymbol{\alpha}_K + n_K\right],$$

also,

$$\boldsymbol{\beta}_{-dj*} = \left[\boldsymbol{\beta}_1 + n_{1j} - d_1, ..., \boldsymbol{\beta}_{|V|} + n_{|V|j} - d_{|V|}\right],$$

and $\boldsymbol{\beta}_{-dk*} = \boldsymbol{\beta}_{k*}$ 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.

```
input  : A set of documents $D$, and the number of desired clusters $K$
output: A sample matrix $\boldsymbol{Z}$
for $d \leftarrow 1$ to $M$ do
    $z_{d,0} \leftarrow$ draw(Uniform([1,...,K]))
end
$i \leftarrow 1$
while More samples are needed do
    for $d \leftarrow 1$ to $M$ do
        $z_{d,i} = draw(p(z|\boldsymbol{w}, z_{1,i}, ..., z_{d-1,i}, z_{d+1,i-1}, ..., z_{M,i-1}))$
    end
    $i \leftarrow i + 1$
end
```

Algorithm 2: Collapsed Gibbs sampling algorithm.

Given this marginal joint, we can now finish the derivation of the complete conditional by substituting the result from Equation 2.10 into Equation 2.1 yielding:

$$p(z_d = j|\boldsymbol{w}, \boldsymbol{z}_{-d}) \propto \frac{B\left(\alpha_*\right) \prod_{k=1}^{K} B\left(\boldsymbol{\beta}_{k*}\right)}{B\left(\alpha_{-d*}\right) \prod_{k=1}^{K} B\left(\boldsymbol{\beta}_{-dk*}\right)} \tag{2.11}$$

After simplifying we arrive at the following expression for the complete conditional for the label of document d:

$$p(z_d = j|\boldsymbol{w}, \boldsymbol{z}_{-d}) \propto (\alpha_j + n_j - 1) \frac{B\left(\boldsymbol{\beta}_j *\right)}{B\left(\boldsymbol{\beta}_{-dj*}\right)} \tag{2.12}$$

Using Equation 2.12, one may sample from $p(z_d = j|\boldsymbol{w}, \boldsymbol{z}_{-d})$, given the labelings 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 2.

2.5 Evaluation

Using the Gibbs sampling algorithm described in Section 2.4 yields a matrix $\boldsymbol{Z}$ of samples for the documents in D, such that $z_{d,i}$ is the i^{th} label sampled for the d^{th} document in D.

In this section, we will discuss three issues as they relate to using $\boldsymbol{Z}$ to choose a clustering for D.

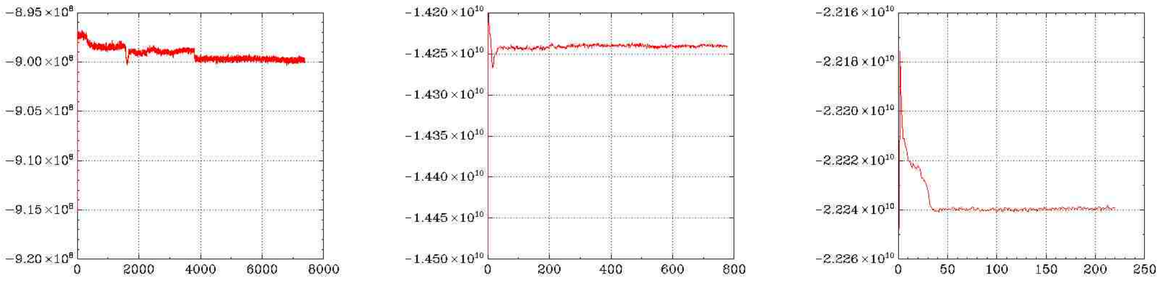
1. How long does it take to converge?
2. How many samples should be taken?
3. How should the collected samples be summarized?

2.5.1 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 Newsgroups set which has become a standard in classification and clustering research [48].

The second data set is a portion of the Enron e-mail corpus which has been annotated and made available through the LDC [11]. 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* [90]. 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".



(a) Enron likelihood time-series. (b) Newsgroups likelihood time-series. (c) Del.icio.us likelihood time-series.

Figure 2.2: Example time-series likelihood plots. The x axis is the sample number, the y axis is the joint likelihood $p(\boldsymbol{w}, \boldsymbol{z})$ for that sample.

We designed a feature selection process to facilitate topical clustering. Before clustering, each document undergoes feature extraction and vectorization. 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 (see Chapter 3 for a more detailed description of this method).

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 [84], Variation of Information (VI) [62], the Adjusted Rand Index (ARI) [45], the V-Measure [81], and the Q_2 metric [26]. These metrics will be used below in any experiments where the quality of various partitionings of the same data set are compared.

2.5.2 Convergence

MCMC sampling techniques are guaranteed to converge in the limit to the target distribution, given some reasonable assumptions [12]. 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 recommends burning up to 50% of the collected samples [34], 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 [75]. 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 could not employ 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. [39]). Several examples of these plots are shown in Figure 2.2. 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.

2.5.3 Identifiability

The model proposed in Section 2.3 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 and all of the dog related documents are labeled 2 from the case where all of the space exploration documents are labeled 2 and the dog documents are labeled 1. This is known as *non-identifiability* and can result in *label switching* during sampling.

Label switching is often a problem when clustering using MCMC techniques on mixture models. This is because it is possible for label switching to occur mid-chain. In this event, averaging across multiple samples can be worse than taking any one individual sample as the chain summary, because the meaning of each label can change across multiple samples. Solutions to the non-identifiability problem have been proposed. These often take the form of a re-labeling scheme [76, 85], and often feel slightly *ad hoc*.

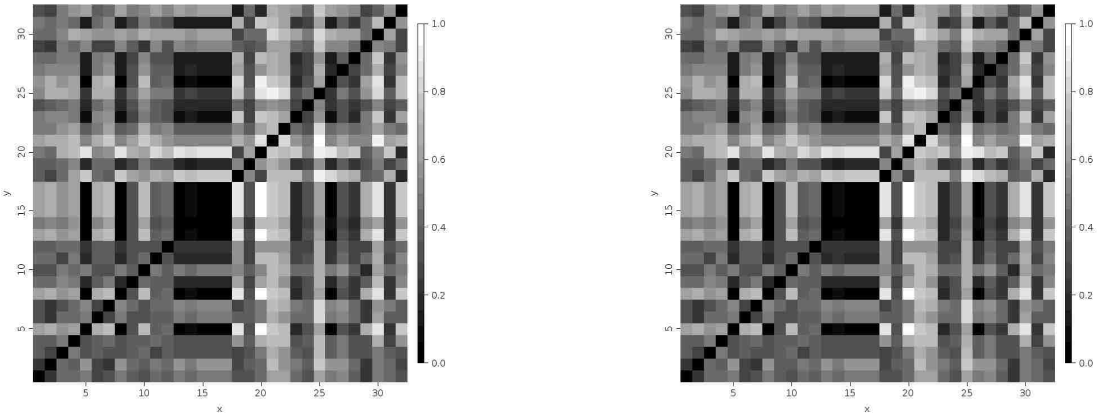
We wanted to determine the extent to which label switching occurs using the collapsed sampler with our model. If within-chain label switching occurs often, then any summarization technique that attempts to use more than one sample will require steps for label switching diagnosis and compensation to be 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:

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

where n_{vj} 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 $p_0(w|z = h)$ and $p_i(w|z = j)$ 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 $p_i(w|z = j)$.

A plot produced in this way 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



(a) Plot showing the distribution of words given the first sample after burn, p_0, against itself.

(b) Plot showing p_0 against p_{5000}.

Figure 2.3: Plots the KL-divergence of the distribution of $p(w|z)$ for the first sample after burn with the same distribution for the i^{th} sample after burn. Values closer to zero are darker, values approaching 1 are lighter.

cells. If label switching occurs in a chain, however, we would expect that for larger values of i, the plot of p_0 against p_i 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 2.3. 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, only 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, so we omit presenting others as they are uninformative. This result is likely the result of the high dimensionality of the data, together with the relatively high correlation between variables.

2.5.4 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, $\hat{\boldsymbol{z}}_d$, is chosen to be the label which was most frequently assigned to that document during sampling:

$$\hat{z}_d = \operatorname*{argmax}_j \sum_i \delta\left(z_{d,i}, j\right)$$

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 $\boldsymbol{Z}_{\cdot,i}$ from $\boldsymbol{Z}$) 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:

$$\hat{\boldsymbol{z}} = \operatorname*{argmax}_{\boldsymbol{Z}_{\cdot,i}} p(\boldsymbol{Z}_{\cdot,i}, \boldsymbol{w})$$

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. In a real clustering application, the number of natural classes would not be known and some sort of model selection would be required in order to determine the appropriate value for K. Model selection is outside the scope of this work, however, and so we sidestep the issue by making use of this information even though it will not typically be readily available to practitioners. The metrics for the 100 chains, across each data set and summarization method pair were averaged. The results are shown in Tables 2.1- 2.3 and indicate that, even though the Marginal and MAP summarizations tend to do slightly better, the summarization methods all performed very similarly. Especially in the cases of the Marginal and MAP summarizations, the results are so close that it is difficult to assert with confidence in any given case that the difference is statistically significant.

Metric	Marginal	MAP	Rand
F-Measure	**0.38591**	0.38571	0.38586
VI	**3.74822**	3.74847	3.74871
ARI	**0.22893**	0.22860	0.228840
V-Measure	**0.47867**	0.47850	0.47859
Q_2	**0.72949**	0.72938	0.72945

Table 2.1: Results comparing the performance of the 3 summarization techniques on 100 chains of 200 samples each, produced using the Social Bookmarking data set.

Metric	Marginal	MAP	Rand
F-Measure	**0.42174**	0.42017	0.42118
VI	**2.14728**	2.16088	2.15931
ARI	**0.27730**	0.27507	0.27692
V-Measure	**0.55637**	0.55330	0.55405
Q_2	**0.73058**	0.72918	0.72969

Table 2.2: Results comparing the performance of the 3 summarization techniques on 140 chains of 750 samples each, produced using the 20 Newsgroups data set.

2.5.5 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 with add-one smoothing was used to cluster each of the three data

Metric	**Marginal**	**MAP**	**Rand**
F-Measure	0.35023	$\mathbf{0.35100}$	0.34989
VI	3.50433	$\mathbf{3.49994}$	3.50624
ARI	0.13499	$\mathbf{0.13564}$	0.13461
V-Measure	$\mathbf{0.32366}$	0.32348	0.32322
Q_2	$\mathbf{0.71328}$	0.71315	0.71309

Table 2.3: Results comparing the performance of the 3 summarization techniques on 100 chains of 7300 samples each, produced using the Enron data set.

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 2.4-2.6. It can be seen that the metrics consistently indicate that the Gibbs sampling algorithm produces better clusterings than EM.

Metric	**Gibbs**	**EM**
F-Measure	$\mathbf{0.38591}$	0.30556
VI	$\mathbf{3.74822}$	4.08036
ARI	$\mathbf{0.22893}$	0.16453
V-Measure	$\mathbf{0.47867}$	0.41225
Q_2	$\mathbf{0.72949}$	0.69472

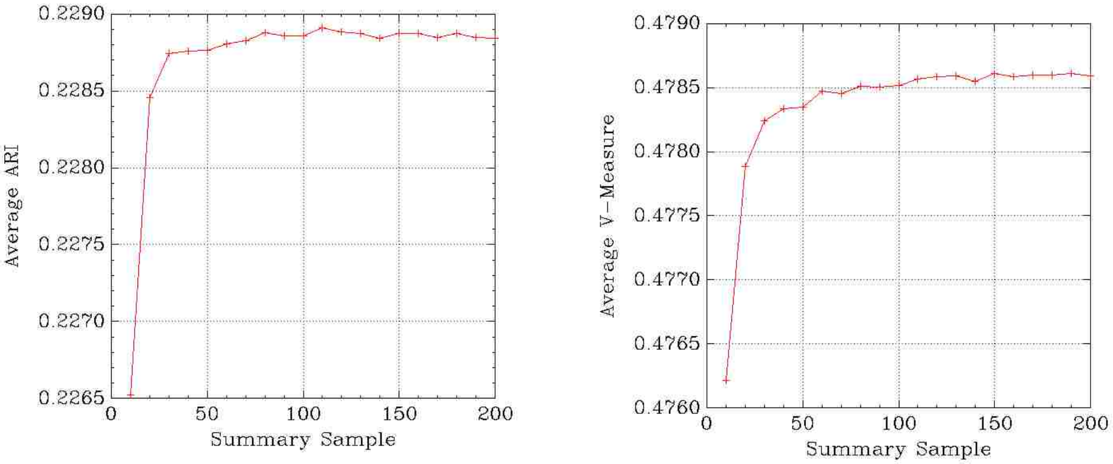
Table 2.4: Results comparing the performance of the best configuration of the Gibbs sampling clustering algorithm to the performance of an EM clustering algorithm on the del.icio.us data.

Metric	**Gibbs**	**EM**
F-Measure	$\mathbf{0.42174}$	0.33012
VI	$\mathbf{2.14728}$	2.60149
ARI	$\mathbf{0.27730}$	0.20405
V-Measure	$\mathbf{0.55637}$	0.44627
Q_2	$\mathbf{0.73058}$	0.68064

Table 2.5: Results comparing the Gibbs clustering algorithm to the EM clustering algorithm on the 20 Newsgroups data.

Metric	Gibbs	EM
GF-Measure	**0.35100**	0.30351
VI	**3.49994**	3.86434
ARI	**0.13564**	0.09739
V-Measure	**0.32366**	0.27483
Q_2	**0.71328**	0.68704

Table 2.6: Results comparing the Gibbs sampling clustering algorithm to the EM clustering algorithm on the Enron data.



(a) Average ARI scores.

(b) Average V-Measures.

Figure 2.4: Results for summarizers that take the sample at a fixed location in the chain as the clustering of the data on the del.icio.us data set.

2.5.6 Summaries Using Few Samples

The experiments from Section 2.5.4 and 2.5.5 show that, in general, the differences between the performance of the various summarization techniques is small compared to the performance of Gibbs versus that of EM.

This fact, coupled with the relatively short times needed for the sampler to reach steady-state, suggests a simplified clustering strategy. Instead of letting the sampler converge and collecting many samples thereafter, it may possible to use the first sample after burn (a prespecified number of samples) as the clustering of the data. The experiments thus far suggest that this approach will likely yield good results, while requiring significantly less time and fewer resources.

To evaluate the potential of this simplified clustering algorithm, a new set of summarizations was generated for the samples collected for the del.icio.us dataset in the previous experiments. These were single-sample summarizations that used samples at regular intervals as the clustering for the data set. This process was repeated starting at the 10th sample, in increments of 10 samples up to the 200th sample. For each data set, the metrics produced for the summarizations based on the i^{th} sample of each chain were averaged together. The results of this process are shown in Figure 2.4.

These results support two points. First, they appear to indicate that the sampler really does converge to steady-state quite quickly, although perhaps not quite as quickly as the likelihood time series plots suggested (Figure 2.2). Second, they show that, after a certain point, choosing a later sample will yield diminishing returns. For this particular data set, it appears that choosing the 200th sample will not necessarily yield better results than choosing the 100th. So, an acceptable algorithm for this data set would be to run the sampler long enough to collect 100 samples, and then use the last sample as the selected clustering of the data.

When the absolute best clustering is desired, longer chains should be run and either the marginal posterior, or MAP summarization method should be used. However, since the Random summarization tends to do nearly as well as the Marginal and MAP summarizations (Tables 2.1-2.3) and since the Random summarization achieves diminishing returns with longer chains (Figure 2.4), the short-circuited strategy suggested here will require less time and yield competitive results since the Random summarization was competitive with the other summarizations and we have now shown that collecting more samples for a Random summarization yields diminishing returns.

2.5.7 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

	00	01	02	03	04	05	08	09	12	16
alt.atheism	1	0	1	10	14	0	5	13	7	948
comp.graphics	1	0	17	20	3	1	927	16	0	13
comp.os.ms-windows.misc	1	0	1	10	4	0	940	5	2	9
comp.sys.ibm.pc.hardware	0	0	3	59	3	0	932	1	0	0
comp.sys.mac.hardware	0	0	1	95	2	0	891	2	0	3
comp.windows.x	0	0	11	10	3	0	966	3	0	3
misc.forsale	0	0	21	500	15	4	429	10	7	3
rec.autos	0	0	2	941	34	0	14	3	0	5
rec.motorcycles	0	0	0	972	19	0	3	2	0	1
rec.sport.baseball	9	0	0	35	929	0	23	1	0	2
rec.sport.hockey	56	0	1	8	910	0	12	0	1	2
sci.crypt	0	0	890	11	30	0	47	4	2	13
sci.electronics	0	0	26	614	6	0	343	5	1	3
sci.med	0	0	4	78	24	0	32	14	790	55
sci.space	1	0	1	41	7	0	62	875	4	7
soc.religion.christian	0	0	1	3	16	0	18	5	4	949
talk.politics.guns	0	0	27	17	888	0	6	15	2	41
talk.politics.mideast	0	0	3	5	290	0	4	588	1	98
talk.politics.misc	0	0	59	6	529	0	5	87	5	305
talk.religion.misc	0	0	42	15	183	0	7	13	9	729

Table 2.7: 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.

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 2.7. 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, close to one that a human annotator might settle on, if the newsgroup labels were withheld (though some topics have been conflated).

2.6 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 2.3. 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 (see Section 2.5.6).
2. Little or no label-switching occurs with our model and data. The results in Section 2.5.3 verify the absence of label switching.
3. Once samples have been collected, there is little difference in the performance of reasonable summarization methods (see Section 2.5.4).
4. The Gibbs sampling algorithm presented here consistently produces better clusterings than EM on the same model (see Section 2.5.5).
5. Longer chains, together with either the marginal posterior, or MAP summarization methods produce the best results. However, much simpler strategies yield competitive results (see Section 2.5.6).

2.7 Future Work

We compared 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 [25]) 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 is likely to be one of the higher hills to climb.

2.8 Addendum

As originally published the paper ended with Section 2.7. However, after publication we ran more experiments which address some of the proposed future work. Though they were not published separately, they are interesting and worth including here because they provide more insight into the performance of the collapsed Gibbs sampler on the mixture of multinomials model. In this addendum we first investigate the issue of model parameterization and show that the choice of the values for the $\boldsymbol{\alpha}$ and $\boldsymbol{\beta}$ hyperparameters can have a significant impact on the quality of the clusters produced by both the collapsed Gibbs sampler and the EM algorithm. Next, we introduce a variational inference algorithm for the mixture of multinomials model and two potential ways of improving the performance of the EM algorithm: deterministic annealing and a brute-force “random restart" algorithm. We also apply deterministic annealing to the collapsed Gibbs sampler and to the variational inference algorithm. Finally, experimental results are presented that compare the three base inference algorithms and the three annealed variants to each other and to the random restart version of EM. Note that the data parsing used in this method is different from that used in the previous sections of this paper, so the absolute values of the metrics presented here are not directly comparable to those found earlier in the chapter.

2.8.1 Hyper Parameterization

Parameters	**Metric**	**del.icio.us**	**20 Newsgroups**	**Enron**
$\boldsymbol{\alpha} = 1.0, \boldsymbol{\beta} = 1.0$	F-Measure	0.41358	0.38467	0.32371
	VI	3.53101	**2.45504**	3.75398
	ARI	0.25834	0.23862	0.11723
	V-Measure	0.50861	0.49402	0.27386
	Q_2	0.73693	0.70336	0.67737
$\boldsymbol{\alpha} = 2.0, \boldsymbol{\beta} = 2.0$	F-Measure	0.36300	0.25709	0.34160
	VI	**3.50239**	2.70451	**3.47321**
	ARI	0.21107	0.13505	**0.14280**
	V-Measure	0.48735	0.36862	0.24747
	Q_2	0.72205	0.63740	0.66992
$\boldsymbol{\alpha} = 2.0, \boldsymbol{\beta} = 0.01$	F-Measure	**0.44027**	**0.49897**	0.32404
	VI	3.56996	2.53061	3.92758
	ARI	**0.27365**	0.34719	0.09856
	V-Measure	**0.51590**	**0.54484**	**0.30006**
	Q_2	**0.74290**	**0.75107**	**0.68112**
$\boldsymbol{\alpha} = 2.0, \boldsymbol{\beta} = 0.001$	F-Measure	0.42564	0.44426	**0.34287**
	VI	3.56015	2.65876	3.78742
	ARI	0.25481	0.29015	0.12279
	V-Measure	0.51084	0.50533	0.29094
	Q_2	0.73859	0.72535	0.67615
$\boldsymbol{\alpha} = 10.0, \boldsymbol{\beta} = 0.01$	F-Measure	0.43262	0.49714	0.31481
	VI	3.58395	2.53247	3.93892
	ARI	0.27095	**0.34892**	0.09477
	V-Measure	0.51361	0.54442	0.29892
	Q_2	0.74162	0.75087	0.68078

Table 2.8: Results of 30 runs of the collapsed Gibbs sampler using alternative parameterizations on the three datasets with marginal summarization. The for each (dataset, metric) pair, the best value produced by any of the parameterizations is highlighted. Recall that for the variation of information metric lower values are better.

As noted above, in our initial experiments we employed several collapsed Gibbs clustering algorithms with $\boldsymbol{\alpha}$ and $\boldsymbol{\beta}$ set to symmetric vectors with all values set to 1 and compared them to EM with add-one smoothing. These are not equivalent parameterizations of the model as, mathematically, add-one smoothing is equivalent to the MAP estimate of the $\boldsymbol{\pi}$ and $\boldsymbol{\phi}$ variables with symmetric $\boldsymbol{\alpha}$ and $\boldsymbol{\beta}$ parameters set to 2 [6]. Another problem with the uninformative prior is that it ignores the fact that we do have prior knowledge about how the clusters should look. Specifically, since we are clustering documents topically, we can suspect that the word distributions for the various clusters should be sparse. For example, we expect the cluster about sports to have high probability for words like “ball", “player" and “team", and low probability for words like “shuttle", “astronaut", and “space". If clusters are topically coherent, then they should all have this characteristic of giving high probability to certain sets of words in the vocabulary that are related to one another and to some external concept. In addition they should give low probability to words that are not related to that concept. This expectation for sparsity can be encoded by selecting a symmetric Dirichlet parameter vector with components less than one. The closer to zero the components are, the more “spiky" or sparse the resulting multinomial parameter vectors will be.

With this in mind, we repeated the experiments from Section 2.5.1 for the collapsed Gibbs sampler using symmetric $\boldsymbol{\alpha}$ and $\boldsymbol{\beta}$ with various values in order to determine to what extent parameterization matters for this task. We used 30 chains for each dataset of the same length as those used above: 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. Only the Marginal summarization technique was used. Burn was also the same as above, 1000 samples for the Enron data, 100 samples for the 20 Newsgroups data, and 50 samples for the del.icio.us data. The data parsing and processing were slightly different for the results in this addendum; so, the results in this section cannot be compared directly to those above. The relative trends between algorithms are similar, though the absolute values of the measured metrics are different.

The results are shown in Table 2.8. It appears that the choice of parameterization can have an affect on the outcomes of the experiments, though not necessarily in a predictable way. For example, the $\boldsymbol{\alpha} = 2.0, \boldsymbol{\beta} = 0.01$ parameterization seems to have significantly better performance on the del.icio.us dataset than the options with $\boldsymbol{\beta} \geq 1.0$, but this trend does not extend as strongly to the other datasets, where the outcomes are less clear. For the 20 Newsgroups dataset, three metrics show that the $\boldsymbol{\alpha} = 2.0, \boldsymbol{\beta} = 0.01$ parameterization is the best, one metric shows that the $\boldsymbol{\alpha} = 1.0, \boldsymbol{\beta} = 1.0$ parameterization is best and, for the ARI, the $\boldsymbol{\alpha} = 10.0, \boldsymbol{\beta} = 0.01$ parameterization is superior. For the Enron dataset the $\boldsymbol{\alpha} = 2.0, \boldsymbol{\beta} = 2.0$ and $\boldsymbol{\alpha} = 2.0, \boldsymbol{\beta} = 0.01$ parameterizations are evenly split with two metrics each coming out on top, with the $\boldsymbol{\alpha} = 2.0, \boldsymbol{\beta} = 0.01$ parameterization taking the last one. In general, though, the $\boldsymbol{\alpha} = 2.0, \boldsymbol{\beta} = 0.01$ appears to be preferable.

We conducted a similar study over alternative parameterizations for the EM algorithm as well. Recall that a hyperparameter value of n is equivalent to add $n - 1$ smoothing over the corresponding counts during EM, which means that values for $\boldsymbol{\alpha}$, and $\boldsymbol{\beta}$ parameters can not be less than 1. As a result, we substitute 1.01 and 1.001 for 0.01 and 0.001, respectively. The results of these trials are shown in Table 2.9. Here the story appears much more simple, with the $\boldsymbol{\alpha} = 2.0, \boldsymbol{\beta} = 2.0$ parameterization having the best results for all of the metrics for the del.icio.us and Enron datasets and for the variation of information and V-Measure metrics in the case of the 20 Newsgroups data.

2.8.2 Comparison to Other Methods

As mentioned in Section 2.7, there are other methods that can be used to infer document clusters with the Mixture of Multinomials model. Here, we briefly explain a few of these methods and then present experimental results that show how their performance compares to each other and to the performance of the collapsed Gibbs sampler.

Variational Bayesian Inference

An alternative inference procedure that is common in the Bayesian document modeling literature is the Variational Bayes approach [12]. Variational Bayes is a functional optimization algorithm that attempts to find the member q of a family of functions that minimizes $KL(q||p)$, the KL-divergence between q and the proposed Bayesian model p. It is sometimes referred to as variational EM (VEM) because of its close relationship to the EM algorithm. The family of candidate functions is typically closely related to the proposed distribution, but simplified in some way. The most common simplification that is made in order to conduct variational inference is the "mean field

Parameters	Metric	del.icio.us	20 Newsgroups	Enron
$\boldsymbol{\alpha} = 1.0, \boldsymbol{\beta} = 1.0$	F-Measure	0.1390	0.1291	0.1995
	VI	6.5082	4.6567	4.5903
	ARI	0.0393	0.0178	0.0207
	V-Measure	0.1410	0.0602	0.1118
	Q_2	0.5535	0.5213	0.5965
$\boldsymbol{\alpha} = 1.0, \boldsymbol{\beta} = 1.01$	F-Measure	0.3657	0.2905	0.2764
	VI	4.2189	3.6681	4.2205
	ARI	0.2129	0.1450	0.0748
	V-Measure	0.4319	0.2978	0.2109
	Q_2	0.7005	0.6272	0.6407
$\boldsymbol{\alpha} = 2.0, \boldsymbol{\beta} = 2.0$	F-Measure	**0.3870**	0.2466	**0.3080**
	VI	**3.7734**	**2.9070**	**3.6872**
	ARI	**0.2254**	0.1344	**0.1130**
	V-Measure	**0.4775**	**0.3397**	**0.2220**
	Q_2	**0.7212**	0.6300	**0.6610**
$\boldsymbol{\alpha} = 2.0, \boldsymbol{\beta} = 1.01$	F-Measure	0.3612	**0.2988**	0.2607
	VI	4.2264	3.6831	4.2771
	ARI	0.2107	**0.1525**	0.0671
	V-Measure	0.4308	0.3038	0.2018
	Q_2	0.6999	**0.6315**	0.6367
$\boldsymbol{\alpha} = 2.0, \boldsymbol{\beta} = 1.001$	F-Measure	0.3493	0.2221	0.2495
	VI	4.3895	4.0412	4.3168
	ARI	0.1962	0.0907	0.0566
	V-Measure	0.4081	0.2055	0.1803
	Q_2	0.6882	0.5844	0.6270

Table 2.9: Results of 30 runs of the EM algorithm using alternative parameterizations on the three datasets. The for each (dataset, metric) pair, the best value produced by any of the parameterizations is highlighted. Recall that for the variation of information metric lower values are better.

approximation". To apply the mean field approximation, one first divides the latent variables of the model Z into a finite set of disjoint groups Z_i, for $i = 1, \ldots, M$, and then assumes that q factorizes as the product of individual distributions for each group:

$$q(Z) = \prod_{i=1}^{M} q_i(Z_i).$$

During the process of factorizing the model, variational variables are introduced and these variables are optimized over in order to produce a clustering. The optimization is a maximization of a lower bound which simultaneously minimizes the KL divergence of the mean field model and the true model. Once the mean field approximation has been applied to the mixture of multinomials document model and variational calculus has been used to find the resulting q families, the following variational variables result in the specified update equations for the coordinate-wise gradient ascent in the variational inference [1]:

$$\mathbf{a}_j^{(l)} = \boldsymbol{\alpha}_j + \sum_{d=1}^{|D|} \hat{\boldsymbol{\phi}}_{dj}^{(l)} \tag{2.13}$$

$$\mathbf{b}_{jv}^{(l)} = \boldsymbol{\beta}_{jv} + \sum_{d=1}^{|D|} \mathbf{n}_{dv} \hat{\boldsymbol{\phi}}_{dj}^{(l)} \tag{2.14}$$

$$\hat{\boldsymbol{\phi}}_{dj}^{(l+1)} \propto \exp\left\{ \psi\left(\mathbf{a}_j^{(l)}\right) - \psi\left(\sum_{j'}^{K} \mathbf{a}_{j'}^{(l)}\right) \right\} \prod_{v=1}^{V} \exp\left\{ \psi\left(\mathbf{b}_{jv}^{(l)}\right) - \psi\left(\sum_{v'}^{V} \mathbf{b}_{jv'}^{(l)}\right) \right\}^{\mathbf{n}_{dv}} \tag{2.15}$$

where $\mathbf{a}$, $\mathbf{b}$, and $\hat{\boldsymbol{\phi}}$ are the variational parameters.

By iteratively calculating first the $\mathbf{a}$ and $\mathbf{b}$ followed by the $\hat{\boldsymbol{\phi}}$s, the following lower bound is maximized, minimizing the KL divergence between the variational distribution and the true distribution:

$$\mathcal{L}(q) = \log B(\mathbf{a}) - \log B(\boldsymbol{\alpha}) + \sum_{k=1}^{K} (\log B(\mathbf{b}_k) - \log B(\boldsymbol{\beta}_k)) - \sum_{i=1}^{N} \sum_{k=1}^{K} \boldsymbol{\phi}_{ik} \cdot \log \boldsymbol{\phi}_{ik}$$

When this bound converges, optimization stops.

We conducted a parameter evaluation for this variational inference algorithm using the same methods that were used for the collapsed Gibbs and EM algorithms in Section 2.8.1. The results of this evaluation are found in Table 2.10. In general, the best parameterization for the variational algorithm was found to be $\boldsymbol{\alpha} = 2.0, \boldsymbol{\beta} = 1.01$, although on the Enron dataset three of the metrics had slightly better outcomes with the $\boldsymbol{\alpha} = 2.0, \boldsymbol{\beta} = 2.0$ parameterization.

[1] See `https://facwiki.cs.byu.edu/cs779/index.php/Variational_equations_for_mixture_of_multinomials` courtesy of Robbie Haertel for the details of the derivation

Parameters	Metric	del.icio.us	20 Newsgroups	Enron
$\boldsymbol{\alpha} = 1.0, \boldsymbol{\beta} = 1.0$	F-Measure	0.3722	0.2432	0.2991
	VI	**2.9364**	2.9364	3.6633
	ARI	0.2139	**0.1263**	0.1002
	V-Measure	0.3304	0.3304	0.2117
	Q_2	0.6256	0.6256	0.6566
$\boldsymbol{\alpha} = 1.0, \boldsymbol{\beta} = 0.01$	F-Measure	0.1151	0.1151	0.1831
	VI	4.7153	4.7153	4.5712
	ARI	0.0094	0.0368	0.0080
	V-Measure	0.0368	0.0094	0.0870
	Q_2	0.5119	0.5119	0.5904
$\boldsymbol{\alpha} = 2.0, \boldsymbol{\beta} = 2.0$	F-Measure	0.3528	0.1876	**0.3037**
	VI	3.6606	2.9586	**3.5252**
	ARI	0.1949	0.0844	**0.1137**
	V-Measure	0.4684	0.2609	0.1995
	Q_2	0.7129	0.5930	0.6548
$\boldsymbol{\alpha} = 2.0, \boldsymbol{\beta} = 0.01$	F-Measure	0.1111	0.1125	0.1948
	VI	6.7568	4.7619	4.5482
	ARI	0.0287	0.0084	0.0130
	V-Measure	0.1098	0.0336	0.0895
	Q_2	0.5375	0.5106	0.5900
$\boldsymbol{\alpha} = 2.0, \boldsymbol{\beta} = 1.01$	F-Measure	**0.3760**	**0.2462**	0.3035
	VI	3.7832	**2.9024**	3.6712
	ARI	**0.2179**	0.1257	0.1075
	V-Measure	**0.4715**	**0.3344**	**0.2148**
	Q_2	**0.7174**	**0.6267**	**0.6573**

Table 2.10: Results of 30 runs of the variational algorithm using alternative parameterizations on the three datasets. The for each (dataset, metric) pair, the best value produced by any of the parameterizations is highlighted. Recall that for the variation of information metric lower values are better.

Random Restarts and Deterministic Annealing

Another set of approaches seek to improve the performance of the EM algorithm in order to make it more competitive with the Bayesian approaches. Recall the penchant for the EM algorithm to become trapped in local maxima as one of the main reasons for its inferior performance. One possible way to overcome this problem is to initiate many random restarts of the algorithm, and then use some internal metric (such as the log likelihood of the data given the learned parameters) to choose the best parameters from the random restarts and to use those to produce a final clustering. This method takes advantage of the much shorter run times of the EM algorithm to achieve better results using a somewhat brute-force strategy with the idea that many runs of an inferior optimization algorithm can still complete faster than a single run of a better one with hopefully similar results.

Another approach to improve EM results is through the use of deterministic annealing [29, 89]. Recall that one of the steps of the EM algorithm is to compute expected cluster-word co-occurrence counts by multiplying word counts in each document by the conditional probability of that document belonging to each cluster given the current point estimates for the parameters and the words in that document:

$$\mathbb{E}[n_{vj}] = \sum_d n_{dv} \cdot p(z_d = j | \boldsymbol{w}_d, \pi, \boldsymbol{\phi})$$

where n_{vj} is the count of the number of times that word tokens of type v occurred in documents with cluster label j, and n_{dv} is the number of tokens in document d of type v. Also,

$$p(z_d = j | \boldsymbol{w}_d) \propto p(z_d = j | \pi) \prod_i p(w_d i | z_d = j, \phi_j).$$

In deterministic annealing, we modify this distribution by taking the t^{th} root of one or both of the $p(z_d = j|\pi)$ and/or $\prod_i p(w_d i|z_d = j, \phi_j)$ terms, where t is referred to as the temperature. This has two desired effects. First, it alleviates the problem of overconfidence caused by the incorrect model assumption of independence between word tokens. With annealing, multiple mentions of the same word have less of an effect on the conditional probability of the cluster label. Second, it smooths the objective function manifold, such that the manifold is very smooth for large values of t (thus having fewer local maxima) and is equal to the original objective function when $t = 1$ [89]. Deterministic annealing repeats the annealing procedure using an annealing schedule taking the t-th root of the distributions, starting with a relatively large value and then successively decreasing t until it is 1. At each value of the temperature parameter, the algorithm is allowed to run with the modified distributions until convergence. We apply annealing to both terms of the conditional probability,

Metric	**Gibbs**	**EM**	**VEM**	**DA Gibbs**	**DA EM**	**DA VEM**	RR EM
F-Measure	0.4380	0.3910	0.3793	**0.4422**	0.4119	0.3979	0.3879
VI	3.5748	3.7688	3.7856	**3.5684**	3.6579	3.6665	3.7778
ARI	0.2722	0.2310	0.2171	**0.2738**	0.2506	0.2395	0.2291
V-Measure	0.5155	0.4796	0.4724	**0.5175**	0.4968	0.4912	0.4780
Q_2	0.7426	0.7225	0.7180	**0.7440**	0.7317	0.7280	0.7216
Time(s)	13585	**294.43**	457.39	22188	562.70	749.39	5233.7

Table 2.11: Results comparing the Gibbs clustering algorithm to various other inference algorithms on the Social Bookmarking data.

Metric	**Gibbs**	**EM**	**VEM**	**DA Gibbs**	**DA EM**	**DA VEM**	RR EM
F-Measure	0.4986	0.2478	0.2375	**0.6053**	0.3642	0.3261	0.2501
VI	2.5281	2.9006	2.9280	**2.1243**	2.5850	2.6421	2.8806
ARI	0.3471	0.1351	0.1212	**0.4582**	0.2459	0.2117	0.1342
V-Measure	0.5448	0.3422	0.3226	**0.6284**	0.4728	0.4424	0.3445
Q_2	0.7509	0.6305	0.6219	**0.7983**	0.6968	0.6788	0.6311
Time(s)	1211.3	**38.840**	74.340	2351.8	71.130	136.98	2183.8

Table 2.12: Results comparing the Gibbs clustering algorithm to various other inference algorithms on the 20 Newsgroups data.

which is the same as taking the t-th root of $p(z_d = j|, \boldsymbol{w}_d, \pi, \boldsymbol{\phi})$. We found that this generally yields superior results to annealing just the second term, the product over word probabilities. Following in this line we apply the same smoothing factor to both the topic-word variational parameters (i.e., the $\hat{\phi}$s) of VEM and the complete conditional of the cluster labels in the Gibbs sampler in order to determine whether annealing might similarly improve these other methods as well.

Experimental Results

In order to assess the quality of the clusterings learned with the various algorithms, we ran a set of experiments similar to those described in Section 2.5. In order to provide the fairest point of comparison possible for the performance of each of the algorithms, we used the findings in Tables 2.8 through 2.10. For the Gibbs clustering algorithm we chose $\boldsymbol{\alpha} = 2.0$, and $\boldsymbol{\beta} = 2.0$ because that parameterization has the most metrics across the three datasets for which it is the best (although it is not the best on the Enron data for any of the metrics) also, the sum of the F-Measure, V-Measure and

Metric	**Gibbs**	**EM**	**VEM**	**DA Gibbs**	**DA EM**	**DA VEM**	RR EM
F-Measure	0.3244	0.2993	0.2985	**0.3270**	0.3083	0.3140	0.3040
VI	3.9231	3.7211	3.6880	3.8828	3.7239	**3.6658**	3.7156
ARI	0.1004	0.1021	0.0999	0.0991	0.1095	**0.1129**	0.1104
V-Measure	0.2996	0.2214	0.2124	**0.3188**	0.2538	0.2483	0.2246
Q_2	0.6810	0.6599	0.6573	**0.6908**	0.6737	0.6718	0.6612
Time(s)	4095.1	**12.440**	25.250	14045	27.690	50.030	652.67

Table 2.13: Results comparing the Gibbs clustering algorithm to various other inference algorithms on the Enron data.

Q_2 scores across the three datasets are the highest for this parameterization. For the EM experiments we continued to use add-one smoothing. For VEM we used $\boldsymbol{\alpha} = 2.0$ and $\boldsymbol{\beta} = 1.01$. With the random restart EM algorithm, each run of the clustering algorithm consisted of 50 runs of standard EM, with the final clustering being generated using the model with the highest log-likelihood of the training data. For all experiments involving annealing, the temperature schedule was 25,10,5,1. In the case of the annealed Gibbs sampler, for each temperature greater than 1 we ran the sampler for the same number of iterations as the value chosen for the burn parameter for that dataset, then we collected burn + length samples with the temperature set to 1. Based on the findings of Section 2.5.6, we used the same values for burn, but only collected 200 samples after burn for each dataset for use with the Marginal summarization method. Each algorithm was used to produce 100 clusterings of the data and the results for each algorithm on each dataset were averaged. The means of the experimental results are in Tables 2.11- 2.13, the random restart variant of EM is indicted by the name RREM and the deterministically annealed variants are indicated by prepending “DA" to the short name of the base algorithm.

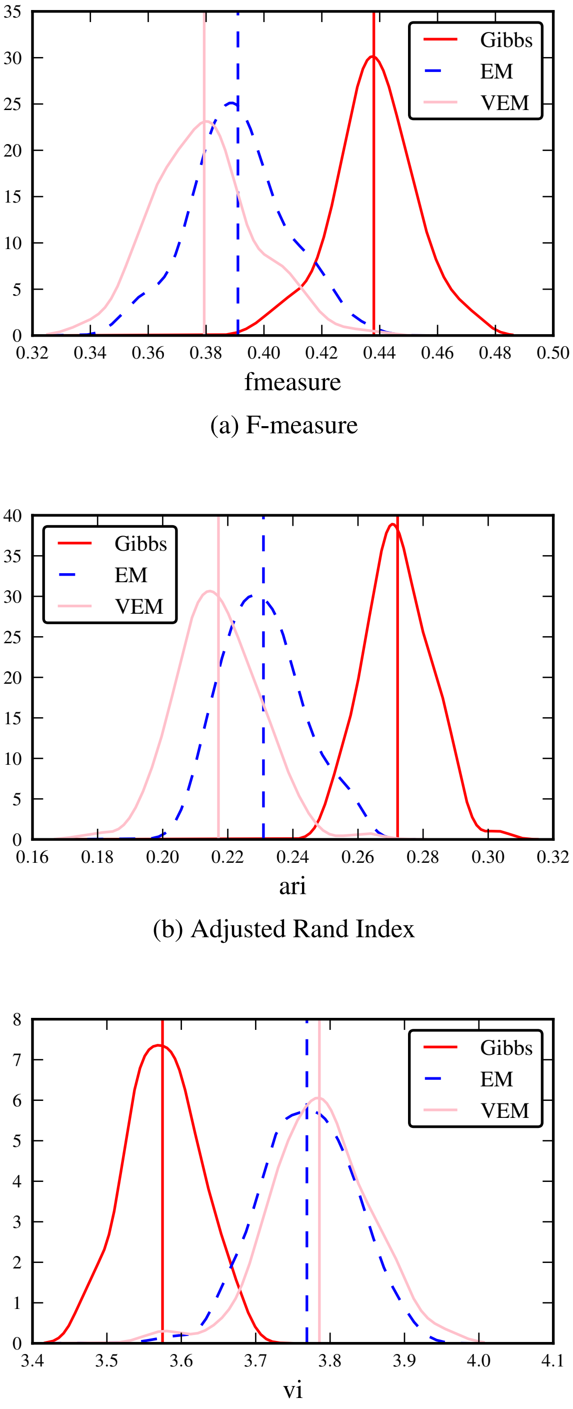
The results suggest that the deterministically annealed versions of the algorithms do better. The deterministically annealed collapsed Gibbs sampler tended to produce the best clusterings according to most of the metrics used. The dominance of the deterministically annealed Gibbs algorithm was especially striking in the case of the 20 Newsgroups data where the highest average F-measure of any of the experiments in this paper was observed at 0.6053. The improvement in performance came at a price, however. On average, the deterministically annealed version of the

Gibbs sampler took between 61 and 1129 times longer to produce clusterings than the basic EM algorithm and between 1.6 and 3.4 times longer than collapsed Gibbs without annealing. This time difference seems severe, but it could most likely be drastically reduced by using better initialization, fewer numbers of samples, and a more optimized implementation of the sampler. One point to note is that deterministic annealing is really just a smart initialization scheme as, with a temperature of 1, the final round is equivalent to running the base model with the output of the penultimate round of annealing as its initial state. These potential improvements in runtime are beyond the scope of this work, which is mainly concerned with the quality of the clusters being produced.

In order to better visualize the distribution over the metrics produced by the various algorithms, we also used the output of these experiments to produce kernel density estimates (KDEs) from these metrics, several of which we will show and discuss here. These plots show how the metrics computed for a given algorithm on a dataset are distributed across many random runs and are helpful in showing not just the mean values of the metrics but also how the metrics are skewed and to what extent differences between the computed means are significant.

Figures 2.5 through 2.7 show how the three base algorithms (Gibbs, EM, and VEM) compare to one another. In most cases, the EM and VEM appear roughly equivalent in performance. Furthermore, the collapsed Gibbs algorithm consistently performs better than EM and VEM in the majority of the cases shown here. This may be due to the fact that VEM, as another hill-climbing algorithm is subject to the same potential of becoming caught in local maxima as EM. Our findings suggest, therefore, that it is not just the fully Bayesian nature of the Gibbs sampler that gives it the advantage over EM, as VEM is considered a Bayesian method as well. The fact that the sampler is stochastic and able to more fully explore the sample space seems to play a role in its superior performance.

Figures 2.8 through 2.10 show how each of the three base algorithms compare to their deterministically annealed versions and to the Random Restart version (in the case of EM). We show plots for only the F-measure metric, as the plots for the other metrics were quite similar to these results. These plots indicate that the deterministically annealed versions of the algorithms

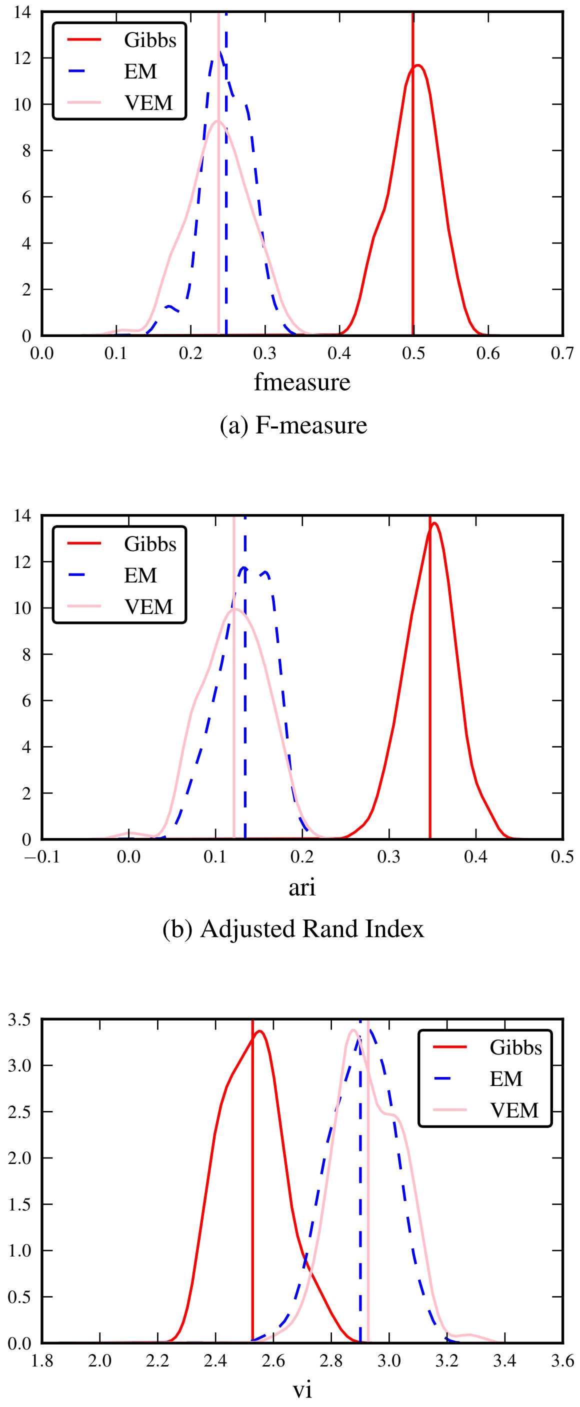


(a) F-measure

(b) Adjusted Rand Index

(c) Variation of Information (lower is better)

Figure 2.5: A comparison of the metric distributions of the various base inference algorithms on the Social Bookmarking dataset. Gibbs is significantly better than EM and VEM for the three metrics shown. EM appears to be slightly better than VEM, though the difference is less significant than the dominance of Gibbs.

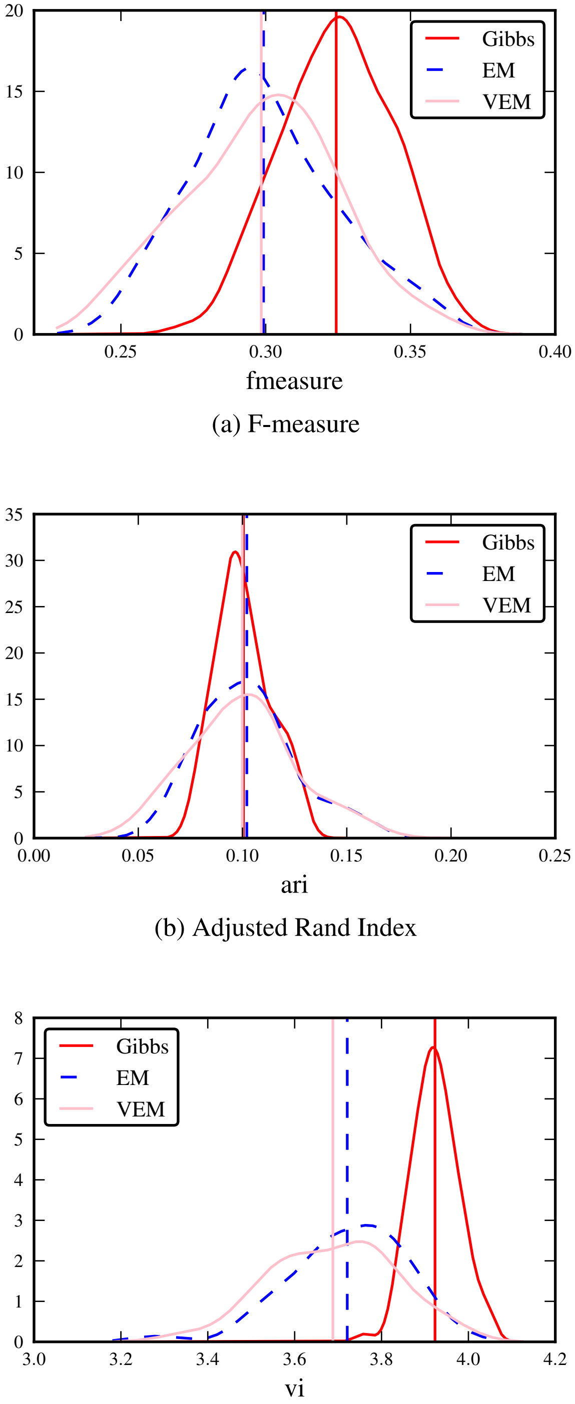


(a) F-measure

(b) Adjusted Rand Index

(c) Variation of Information (lower is better)

Figure 2.6: A comparison of the metric distributions of the various base inference algorithms on the 20 Newsgroups dataset. Again, Gibbs dominates significantly while EM and VEM achieve nearly identical results.



(a) F-measure

(b) Adjusted Rand Index

(c) Variation of Information (lower is better)

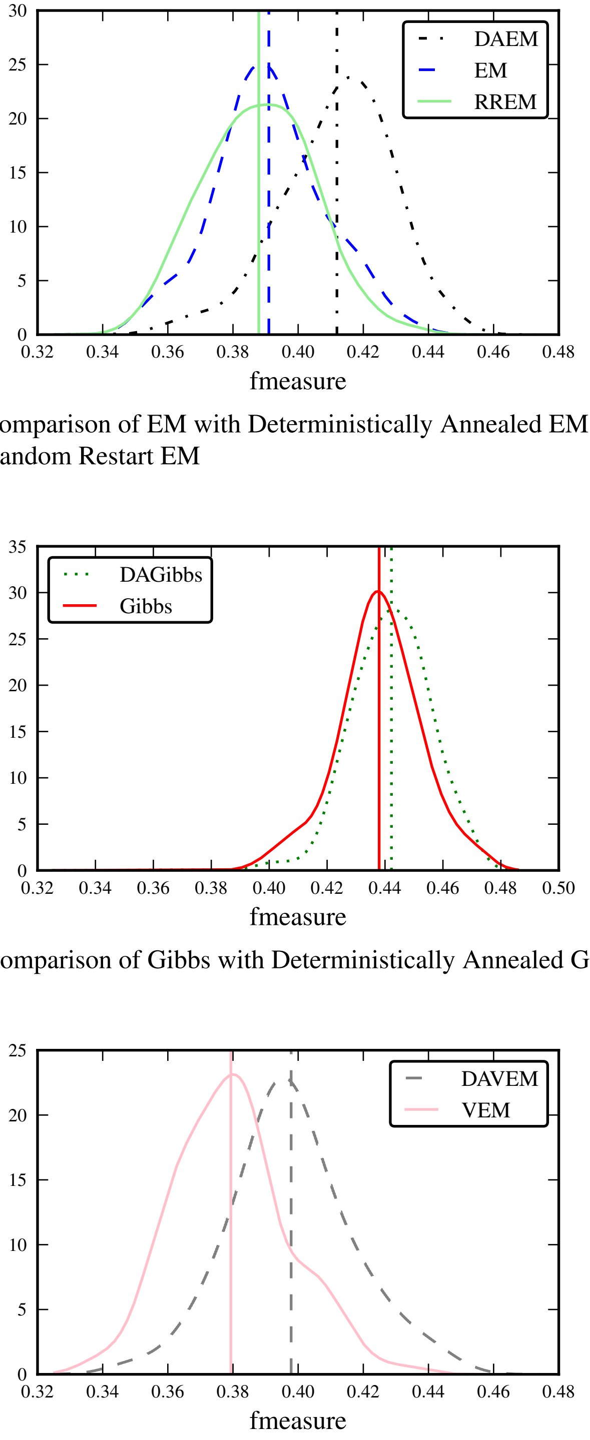
Figure 2.7: A comparison of the metric distributions of the various base inference algorithms on the Enron dataset. In this case the F-measure indicates that Gibbs is significantly better than the other methods. In contrast, the Adjusted Rand Index shows them performing very similarly, and the Variation of Information appears shows Gibbs performing the worse of the three.

tend to do no worse than the base algorithms (e.g., as shown in Figures 2.8b and 2.10), and often perform significantly better. Contrary to intuition, the random restart version of EM does not show a significant improvement over standard EM.

Figures 2.11 through 2.13 directly compare the deterministically annealed versions of the three base algorithms. These graphs show that the deterministically annealed version of EM appears to be slightly better than the deterministically annealed version of the variational algorithm, although there is a lot of overlap between the distributions over outcomes for the two models. This difference does not appear to be significant the majority of the time. The deterministically annealed version of the Gibbs sampler tends to produce better outcomes than either of the other two algorithms and the results appear to almost always be significant.

2.8.3 Addendum Conclusion

We have shown that for the mixture of multinomials document model the EM, collapsed Gibbs clustering and variational inference algorithms are all somewhat sensitive to the setting of the α and β hyperparameters. In addition, the experimental results imply that the collapsed Gibbs sampler tends to outperform both EM and Variational EM, while standard EM and variational EM tend to produce fairly similar results. Finally, we have also shown how deterministic annealing can be used to improve the performance of not only the EM algorithm (this has been known for some time [29]), but also the performance of variational inference and the collapsed Gibbs sampler, with the deterministically annealed collapsed Gibbs sampler generally dominating the other algorithms used here in terms of the majority of the metrics we used. The improvement comes at the cost of much longer running times. Future work could focus on scaling the deterministically annealed Gibbs sampler by optimizing the code, by proposing more efficient sampling algorithms such as has been done for LDA [73], or by turning to parallel or online algorithms, again using techniques that have been developed for LDA as a basis [5, 42, 101].

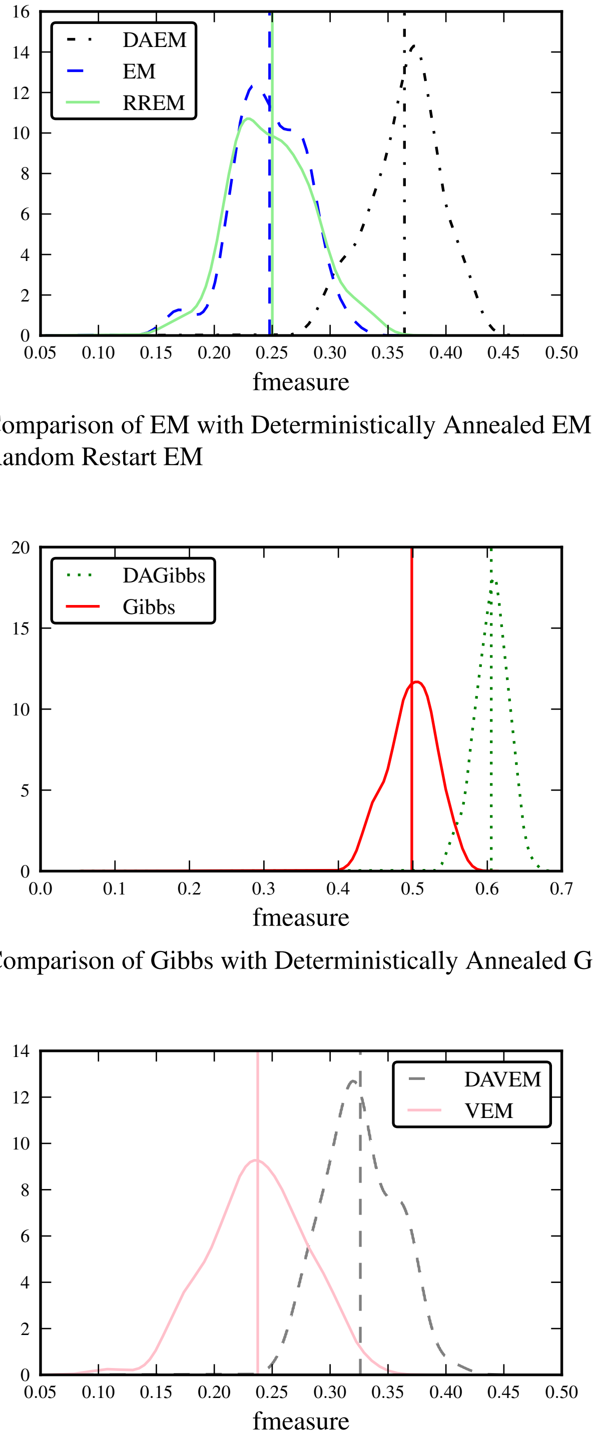


(a) Comparison of EM with Deterministically Annealed EM and Random Restart EM

(b) Comparison of Gibbs with Deterministically Annealed Gibbs

(c) Comparison of Variational with Deterministically Annealed Variational

Figure 2.8: A comparison of the F-measure distributions of the various inference algorithms with their annealed and random restart variants on the Social Bookmarking dataset. The deterministically annealed variants do as well or better than the base algorithms. The Random Restart EM variant is no better than EM.

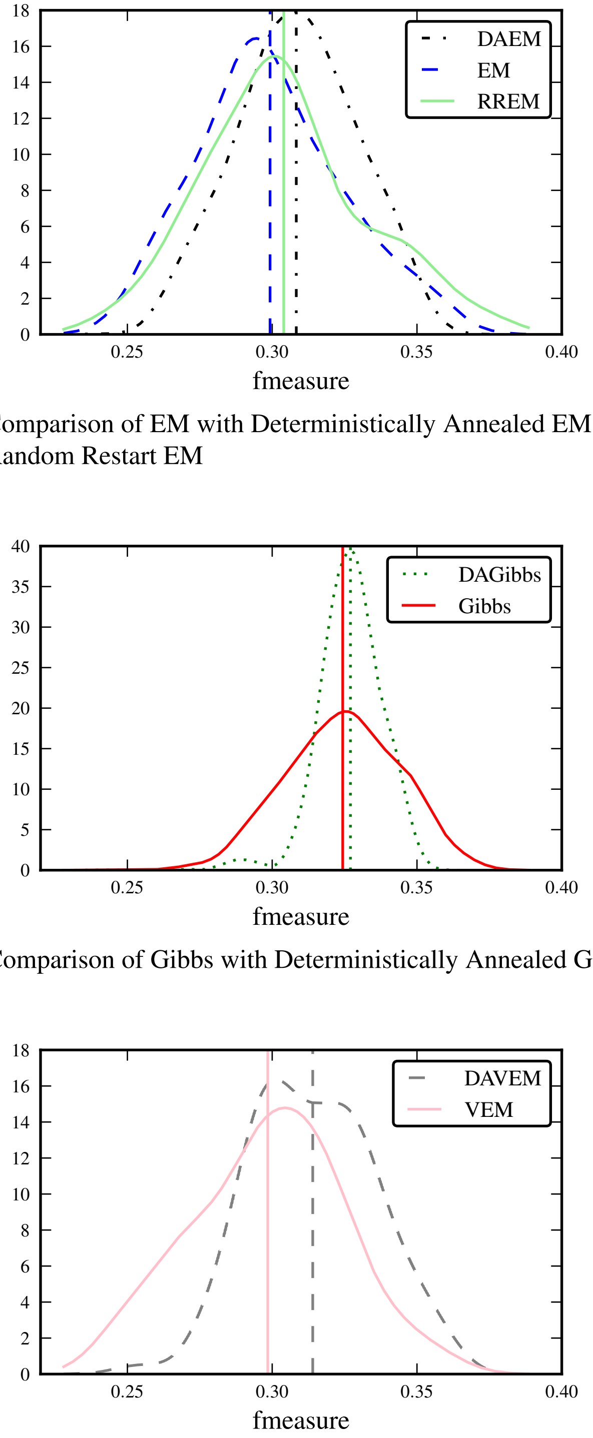


(a) Comparison of EM with Deterministically Annealed EM and Random Restart EM

(b) Comparison of Gibbs with Deterministically Annealed Gibbs

(c) Comparison of Variational with Deterministically Annealed Variational

Figure 2.9: A comparison of the F-measure distributions of the various inference algorithms with their annealed and random restart variants on the 20 Newsgroups dataset. The deterministically annealed variant do as well or better than the base algorithms. The Random Restart EM variant is no better than EM.

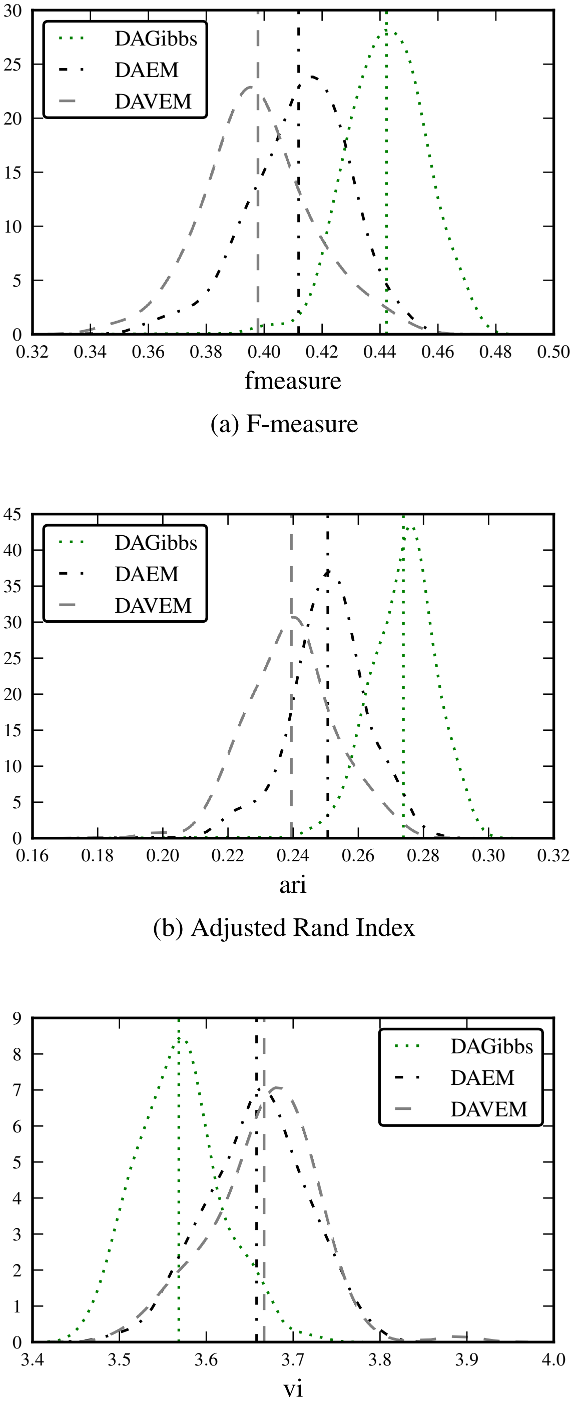


(a) Comparison of EM with Deterministically Annealed EM and Random Restart EM

(b) Comparison of Gibbs with Deterministically Annealed Gibbs

(c) Comparison of Variational with Deterministically Annealed Variational

Figure 2.10: A comparison of the F-measure distributions of the various inference algorithms with their annealed and random restart variants on the Enron dataset. The deterministically annealed variant do better than the base algorithms. The Random Restart EM variant is no better than EM.

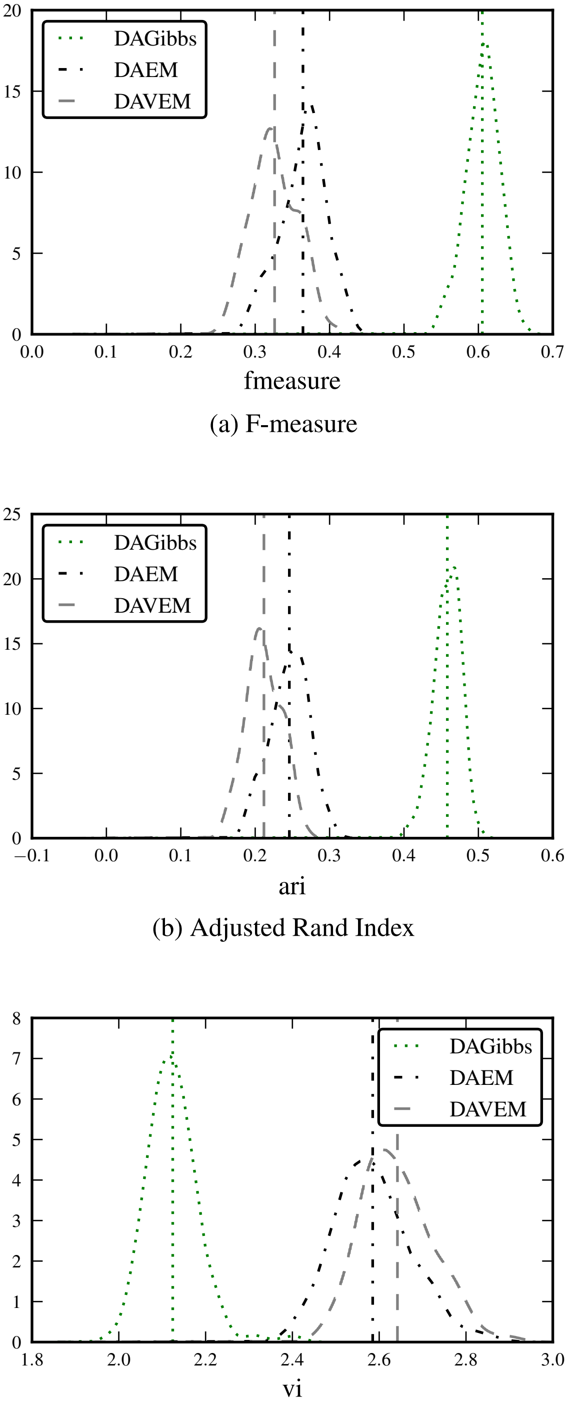


(a) F-measure

(b) Adjusted Rand Index

(c) Variation of Information (lower is better)

Figure 2.11: A comparison of the metric distributions of the various deterministically annealed inference algorithms on the Social Bookmarking dataset. The Gibbs variant dominates the other two methods. The EM variant tends to be slightly better than the VEM variant.

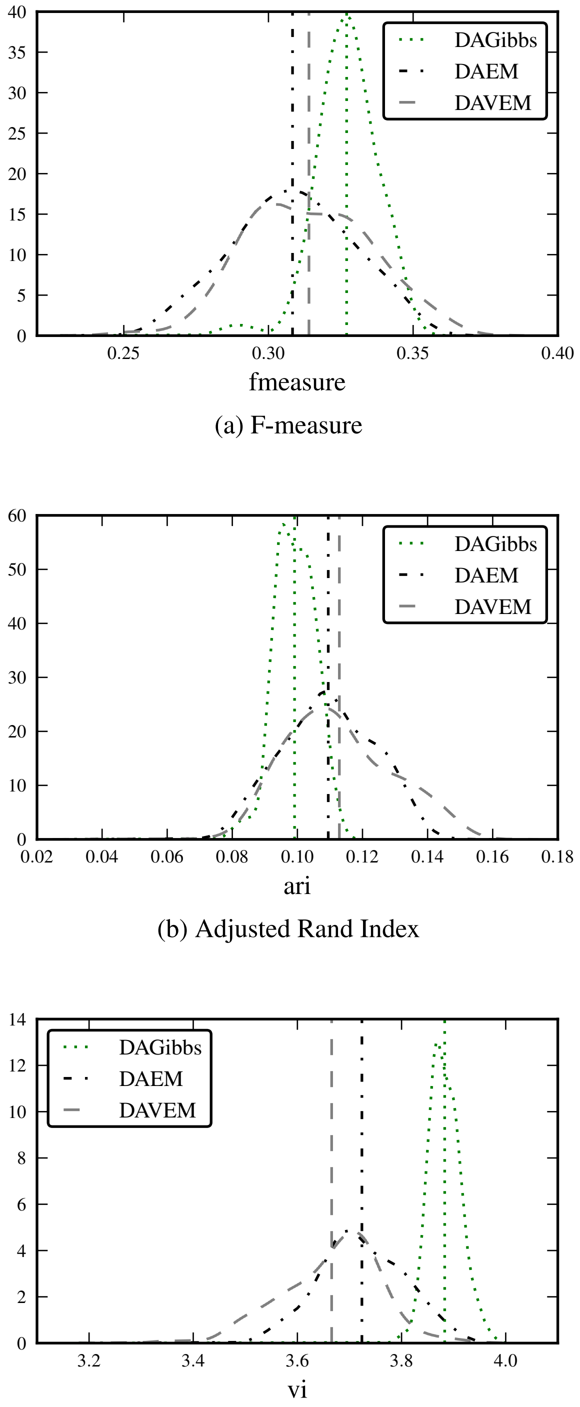


(a) F-measure

(b) Adjusted Rand Index

(c) Variation of Information (lower is better)

Figure 2.12: A comparison of the metric distributions of the various deterministically annealed inference algorithms on the 20 Newsgroups dataset. The Gibbs variant dominates the other two methods. The EM variant tends to be slightly better than the VEM variant.



(a) F-measure

(b) Adjusted Rand Index

(c) Variation of Information (lower is better)

Figure 2.13: A comparison of the metric distributions of the various deterministically annealed inference algorithms on the Enron dataset. Again, this dataset yields anomalous results, with the Gibbs variant doing better for F-measure, and worse for the other two metrics.

Chapter 3

Top N Per Document: Fast and Effective Unsupervised Feature Selection for Document Clustering

Abstract

Feature selection has been shown to be an effective way to both simplify a clustering task and to improve the performance of clustering algorithms. Filter feature selection can be computationally efficient, since it involves a pre-process of the data and only invokes the clustering algorithm once. We present Top N Per Document, a feature selection method that selects features at the document level, rather than at the corpus level. This helps to ensure that the defining characteristics of each document are preserved. This approach is computationally efficient compared to other filtering feature selectors and is simple to implement since it is able to make use of off-the-shelf feature weighting functions as its core component. Furthermore, empirical evidence suggests that Top N Per Document produces clustering results that are competitive with other filter feature selectors.

3.1 Document Clustering and Feature Selection

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 should be dissimilar. This is sometimes referred to as “hard" clustering. 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.

Feature selection is a pre-processing step that is often applied to reduce the dimensionality of the feature space in the context of document clustering as it helps alleviate some of the problems related to high dimensionality. In the classification setting, feature selection is primarily useful in that it simplifies the task and makes learning faster. Intuitively, it would seem that feature selection is even more important in the unsupervised case. In a supervised setting, it is possible for a learning algorithm to deal with noisy data or data that arises from spurious signals, because that noise will be un-correlated with the features supplied at training time. In the case of clustering, however, there are no such clues to help separate noisy signals from the “true" or “good" signals. As a result, random noise in the feature space can lead clustering algorithms to cluster according to these noisy signals, producing poor quality clusterings. This intuition is borne out by empirical evidence that shows that clustering results can be significantly improved by drastically reducing the number of features, while classification results are generally better with relatively few features removed. This evidence will presented later in the paper.

Formally, we say that feature selection is a function

$$f : \mathbb{R}^m \rightarrow \mathbb{R}^n$$

which maps documents represented as vectors in an m-dimensional feature space to vectors in an n-dimensional feature space and, usually, $n \ll m$. Also, feature selection is distinguished from other types of dimensionality reduction in that the features in the new space are a subset of the features in the candidate space. That is, there is a candidate vocabulary $C = \{c_1, \ldots, c_m\}$ and a selected vocabulary $V = \{v_1, \ldots, v_n\}$, and it is the case that $V \subset C$.

There are three main types of feature selection: filter methods, wrapper methods, and embedded methods [47]. Filter methods are applied as a pre-process on the data, before any clustering or classification has begun. They usually involve the computation of a weight function for each term in the candidate vocabulary that is global to the entire corpus of documents being processed. This global weighting function is used to rank the terms, and some proportion of the highest ranking terms are typically chosen for inclusion in the selected vocabulary.

Wrapper selection methods treat feature selection as an iterative process that is interleaved with runs of the clustering algorithm. Thus, the feature selection logic becomes a “wrapper" for the clustering process. These techniques have been shown to produce excellent results but are significantly more expensive than common filter methods, because they involve multiple runs of the clustering algorithm.

Embedded methods are feature selection methods that are incorporated into the learning algorithm. While they can be fast and produce good results, they also preclude the use of most off-the-shelf clustering algorithms, and it may thus be difficult to modify existing algorithms to use embedded methods.

Top N Per Document is a filter method that relies on weighting functions that are defined at the document level, instead of at the global or corpus level. Because it is a filter method, it is useful specifically in the cases where filter methods are more appropriate, such as when one is using an off-the-shelf clustering algorithm, or when it is prohibitive to run the clustering algorithm more than once (e.g., the data is large or the clustering algorithm is particularly costly). We will therefore constrain our comparison to state-of-the-art filter method going forward. More details of Top N Per Document are given in Section 3.3.

3.2 Related Work

A significant body of literature has explored the issue of dimensionality reduction for clustering. Yang and Pederson conducted an in-depth analysis of feature selection for document classification [103]. They were able to systematically compare the performance of several effective techniques for supervised feature selection. Techniques that work in the supervised case, however, are not particularly suited for the unsupervised case because they rely on finding correlations and relationships between feature occurrences and the class labels of training instances; and, by definition, labels are not available in an unsupervised setting.

Liu, et al. in 2003, conducted a similar study for unsupervised feature selection [54]. They proposed an effective unsupervised filtering method called Term Contribution, which we use as a baseline in this work. Term Contribution calculates the degree to which each term contributes to the cosine similarity between pairs of documents (represented as TF-IDF vectors), in the dataset. The authors also implement an EM-inspired wrapper method in which they alternate clustering with the use of supervised feature selection algorithms using the cluster labels instead of class labels. Their results indicate that the wrapper methods are superior but are expensive as they require multiple runs of the clustering algorithm, in order to reach the superior performance level. Also in 2003, Dhillon, et al. published a book chapter on feature selection and document clustering in which they propose two global feature weighting functions. The first is called “term variance quality" that measures the variance in the frequency of occurrences of each term across all documents. The second relies on “term profiles" that evaluate the contexts in which each term occurs. These term profiles are used to calculate a term profile quality according to a “somewhat contrived" formula that attempts to penalize words with undesirable characteristics [24].

More recently, Jashki. et al. proposed an iterative filter-wrapper hybrid using the same basic structure as in [54], but substituting a cluster-specific document frequency for the supervised feature selection methods used in that earlier work [47]. Wrapper methods have also been combined with search techniques and ensemble clustering to yield compelling results on non-document data [44].

In addition to feature selection, other dimensionality reduction techniques have been quite successful. These methods, such as principal component analysis (PCA) [72], latent semantic analysis (LSA) [23], probabilistic latent semantic indexing (PLSI) [43], and latent Dirichlet allocation (LDA) [16] embed documents in a lower dimensional feature space. The new features are derived from the old features, but often in non-obvious ways. Even in the case of LDA, which is a topic modeling algorithm and is meant to produce meaningful groupings of features, it is often difficult to interpret the new features [21], and models built based on these features will likely also be difficult to interpret.

Top N Per Document was briefly mentioned previously in the context of a paper on document clustering methods [91]. The earlier mention did not include the details of the algorithm. Any discussion of its properties and performance relative to other methods were also omitted in that work.

3.3 Top N Per Document Filtering

The Top N Per Document (TNPD) method attempts to preserve the semantic content of each document visible to the clustering algorithm during feature selection by ensuring that each document contributes those features that best characterize its meaning to the selected feature set. The features of each document are weighted according to some function and then the top N features according to this weighting are added to the overall selected feature vocabulary. The details are given in Algorithm 3. It can be seen that this is a simple technique that converts a per-document feature weighting function into a global feature selector. By avoiding many calculations that span multiple documents for each feature, we still capture important information about which features best represent the dataset (by ensuring that each document is well represented) while greatly increasing

```
input  : A set of documents $D$, a set of candidate feature types $C$, the number of features to
         keep per document $N$, and a function $W : D \times C \rightarrow \mathbb{R}$ that maps a document and a
         feature to a real-valued weight
output: A set of selected features $V$
for $d \leftarrow 1$ to $|D|$ do
    // Weight each feature type in $d$
    for $i \leftarrow 1$ to $|C|$ do
        $w_i \leftarrow W(d, i)$
    end
    //Sort features by weight
    $s \leftarrow (s_1, \ldots, s_M)$ such that $w_{s_i} \leq w_{s_{i+1}}$
    //Select top N features
    $V \leftarrow V \cup \{C_{s_i}, \ldots, C_{s_N}\}$
end
```

Algorithm 3: The Top N Per Document feature selection algorithm

the efficiency of the process. Note that even though decisions to add features to the vocabulary occur on a per-document basis, the result is a global set of features. After feature selection, this set is used to filter from each document the words that were not included in this vocabulary.

3.3.1 Choice of Weight Function

Several off-the-shelf feature weighting functions exist for text documents. The purpose of these functions is to measure the extent to which each feature of the document is representative of the meaning of the document.

The simplest weight function is the Term Frequency (TF), method which weights each feature in the document by the number of times the feature occurs in the given document:

$$\mathit{TF}(d, i) = \frac{d_i}{\displaystyle\sum_{j=1}^{M} d_j} \tag{3.1}$$

where d_i is the number of times feature i occurs in document d normalized by the total number of tokens in the document. The intuition behind Term Frequency is that words that occur frequently in a document are important to its meaning. While extremely simple, TF is actually a poor indicator of a feature's importance in a document, as a consequence of the Zipfian distribution of words; there are few words that occur very often across all documents and many words that occur rarely across all documents. The words that occur often across all documents are the most likely words to occur often in any individual document and are precisely the ones that carry the least information about the meaning of the document in question, as they tend to be function words and carry little semantic value.

Perhaps the most popular weighting function is Term Frequency-Inverse Document Frequency (TF-IDF). TF-IDF begins with Term Frequency and penalizes words that occur frequently across all documents by an inverse document frequency term:

$$\textit{IDF}(i) = \log \frac{|D|}{\sum\limits_{d \in D} \mathbb{1}_i(d)} \tag{3.2}$$

where $\mathbb{1}_i(d) = 1$ if $d_i > 0$ and $\mathbb{1}_i(d) = 0$ otherwise, yielding:

$$\textit{TF-IDF}(d, i) = \textit{TF}(d, i) \cdot \textit{IDF}(i) \tag{3.3}$$

TF-IDF is very popular in practice, although its justification is mostly based on intuition.

We also experiment with a third weighting function based on the KL-divergence that has both an intuitive interpretation as well as a principled one. The KL-divergence between two probability mass functions is given by the following equation:

$$D_{KL}(p||q) = \sum_i p(i) \log \frac{p(i)}{q(i)} \tag{3.4}$$

The KL-divergence is an information theoretic measure of the expected number of extra bits (alternatively nats or bans, depending on the base of the log function used) required to encode samples drawn using the distribution p when encoded with an encoding optimized on q. Based on that interpretation we let $p = p_d$ be a document specific categorical distribution over words and $q = q_D$ to a corpus-level categorical distribution over words and observe that the KL-divergence between p_d and q_D gives an indication of the difference between this document in particular and the corpus in general. Each feature in the candidate set has a definable contribution to this measure, meaning that not only can we tell how different a given document is from the “average" document, but we can also identify which features contribute the most to this difference. Based on this reasoning, we define a third weight function which is the individual component of the KL divergence sum for a feature, which we call point-wise KL divergence (PKL):

$$\mathit{PKL}(d,i) = p_d(i) \log \frac{p_d(i)}{q_D(i)} \tag{3.5}$$

PKL and TF-IDF are similar in form and function. Both techniques give positive weight to words that occur frequently in the current document, and penalize words that occur frequently in the corpus in general. As can be seen in Figure 3.1, they appear to be strongly positively correlated.

3.4 Experimental Results

In this section, we present two sets of experiments conducted in order to evaluate the performance of TNPD in relation to other filtering feature selection methods. For the sake of readability, we denote all TNPD methods by the underlying weight function used. Thus, results reported with the label PKL are results obtained by TNPD using the PKL feature weighting function from Equation 3.5.

3.4.1 Experimental Design

Three datasets were chosen for evaluation. Table 3.1 summarizes key statistics for these datasets. The first is the 20 Newsgroups dataset [51], which was collected from Usenet newsgroups in 1995, and has a roughly uniform class distribution (i.e., 1,000 documents per class). The second dataset is the Reuters-21578 dataset [52]. Only documents with at least one topic label were included in the dataset. In the case where an individual document was labeled with more than one topic, the first topic listed was assigned to the document for the purposes of evaluation. This dataset has a

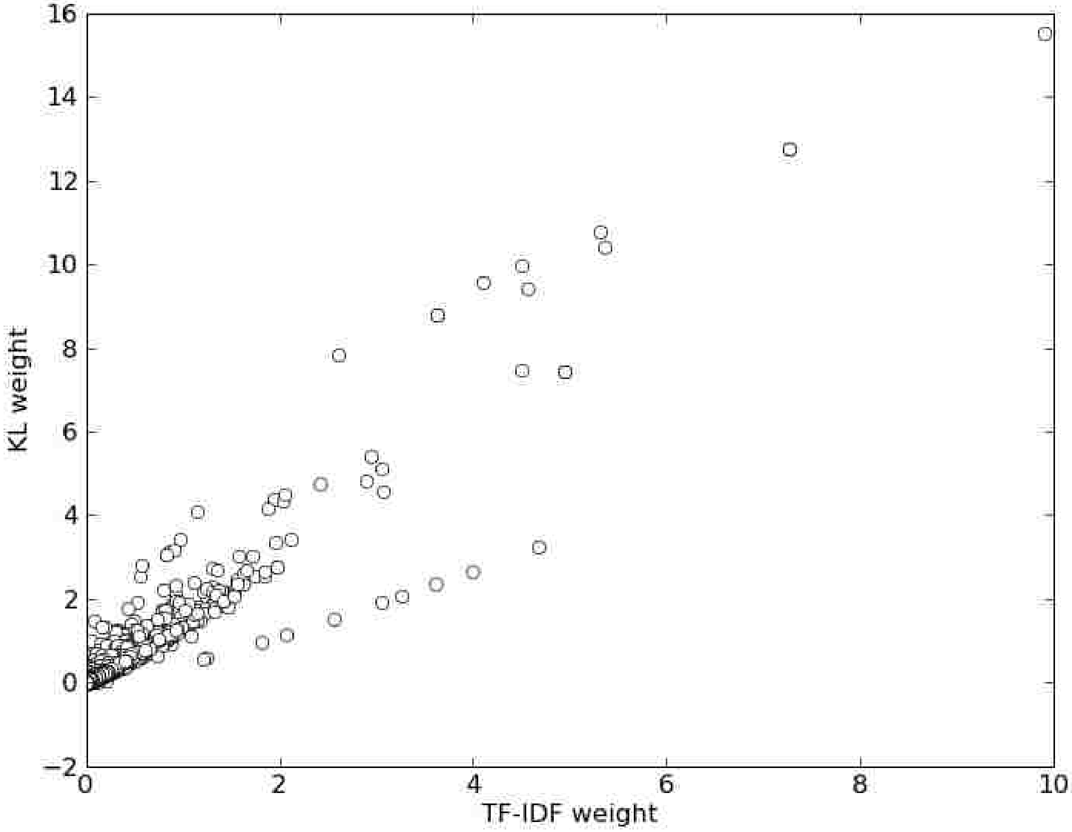


Figure 3.1: The relationship between the TF-IDF and PKL weighting functions as computed on features in the documents belonging to the 20 Newsgroups dataset.

much less uniform class distribution, with the largest class containing 3,972 documents and the 10 smallest classes containing only 1 document each. The last dataset used is the LDC-annotated portion of the Enron e-mail corpus.

To evaluate the performance of TNPD, we compare against the Term Contribution (TC) method described in [54]. TC is a global weight function (as opposed to a per-document weight function like TF, TF-IDF, and PKL) defined as follows:

$$TC(i) = \sum_{d,d' \in D, d \neq d'} \textit{TF-IDF}(d,i) \cdot \textit{TF-IDF}(d',i) \tag{3.6}$$

Dataset	|D|	**# Classes**	M
20 News	19997	20	112429
Reuters	11367	81	32603
Enron	4935	32	63136

Table 3.1: Summary of test dataset characteristics. M is the number of features in the un-filtered vocabulary of the dataset, as tokenized in our experiments.

The TC feature selection algorithm simply computes $TC(i)$ for features $i = 1, \ldots, m$ and then selects the top n features with the highest weight.

In addition, we also compared against a non-feature selecting dimensionality reduction method: LDA with a collapsed Gibbs sampler. LDA was chosen as it is perhaps one of the more principled methods available and is a general case of PLSI [36]. To use LDA as a dimensionality reducer we ran the algorithm over each of the datasets, using the MALLET implementation [61], and replaced the original tokens with their topic assignments in the last sample of the MALLET Gibbs sampler. Memory constraints prevented us from running this method for larger numbers of topics (12 gigabytes of RAM were insufficient), and so we chose to run only with topic numbers $50 \cdot 2^i$ for $i \in \{0, \ldots, 11\}$.

To evaluate the relative quality of the features selected by each method, we ran feature selections for the TNPD feature selectors for all values of N from 1 to 50. In the case of the clustering experiments, no other pre-processing steps were applied, as we wished to avoid biases imposed by using other arbitrary criteria such as frequency cutoffs or arbitrary lists of stop words. After each run of a TNPD selector, we recorded the number of features that were selected by each and then ran the TC feature selector parameterized by that same number. For the evaluation we then ran the clustering algorithm on each of the resulting feature-selected datasets 10 times. As a clustering algorithm, we chose the simple EM with a mixture of multinomials method [63] which is very popular amongst practitioners and is closely related to the vector-based model commonly used in K-means clustering. Since selection of the optimal number of clusters is beyond the scope of this work, we simply selected the number of natural classes, as labeled in the data.

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 [84], Variation of Information (VI) [62], the Adjusted Rand Index (ARI) [45], the V-Measure (along with its components: homogeneity and completeness) [81], average entropy [54] and the Q_0 and Q_2 metrics [26].

Finally, we also ran TNPD in the context of a classification task. The available cluster quality metrics are useful for determining how well the unsupervised feature selectors performed in comparison to other methods on a particular dataset, but it is difficult to tell based solely on these metrics whether the results are of reasonably high quality. In order to verify that TNPD is selecting a reasonable set of features, we used TNPD to conduct feature selection as a pre-process to a supervised document classification task. For comparison, we implemented two supervised feature selectors based on the χ^2 and mutual information (MI) global feature weighting functions. Note, that the MI used here is distributional MI (referred to as information gain in some parts of the literature), instead of point-wise MI. These methods have been shown to be effective for supervised feature selection in the literature [54, 103]. Since χ^2 can be inaccurate for low frequency features [103], we also eliminated all features occurring in fewer than 2 documents before conducting feature selection.

To determine the number of features to select, the PKL TNPD method was first run for $N \in \{1, \ldots, 50\}$ on the 20 Newsgroups dataset. Then, features in the candidate set were assigned a weight using either χ^2 or MI, and then, again for each N the top n_N features are selected, where n_N is the number of features remaining after running TNPD with parameter N. The quality of each feature selector was measured by conducting classification with a Naive Bayes classifier on each of the feature-selected datasets.

3.4.2 Clustering Results

Figures 3.2 - 3.4, show the results of the clustering experiments. For the sake of space only ARI, F-Measure, VI, V-Measure, Average entropy, and Q_2 are shown, with error bars to show standard error. The plots for the other metrics match the trends shown in these graphs. In order to increase readability, the data has been thinned to show only 15 data points per trend line.

These results indicate that, not only does using TNPD instead of TC not hurt clustering performance, it can actually improve it when the greatest possible number of features is removed from the candidate set (i.e. when $N = 1$). This occurs in the 20 Newsgroups results. It is important to note that, though TNPD degrades in relation to TC for some of the datasets as large numbers of features are being selected, the area in which TNPD is most competitive is the preferred scenario, as it represents both a greater reduction in the number of features as well as the highest cluster quality metrics (typically).

TNPD with TF-IDF appears to be the clear winner in this regard among the feature selection methods for the 20 Newsgroups dataset. Results on the other datasets are less conclusive, however. For the Reuters-21578 data , the TF-IDF weighted TNPD method is actually the worse performer for the lower levels of N. The TNPD methods recover toward the center of the graph, but then degrade to be significantly worse than TC for the highest values of N. In the case of the Enron data, the TF-IDF method starts out worse according to the VI and ARI metrics, but better according to V-Measure and Average entropy. In contrast to TF-IDF, the PKL-weighted TNPD is more consistently competitive with TC at the lowest values of N. In addition, despite its simplicity, the TF method tracks TC quite well, though it appears to do so in a consistently inferior way.

The LDA dimensionality reduction method performed very well on the 20 Newsgroups dataset and competitively well on the other two datasets for high levels of dimensionality reduction, and yields competitive results with only dozens of features. Performance drops off quickly, however, as the number of topics increases, and reaches levels of performance that are significantly worse than the pre-processing feature selectors at similar numbers of features, though it appears to stabilize eventually. The drop off is likely due to the clustering behavior of the LDA dimensionality reducer. For small numbers of topics, most instances of a single feature type, and many instances of related features, will have the same label on any given iteration of the Gibbs sampler. This essentially clusters many features into a single feature type, giving the clustering algorithm information

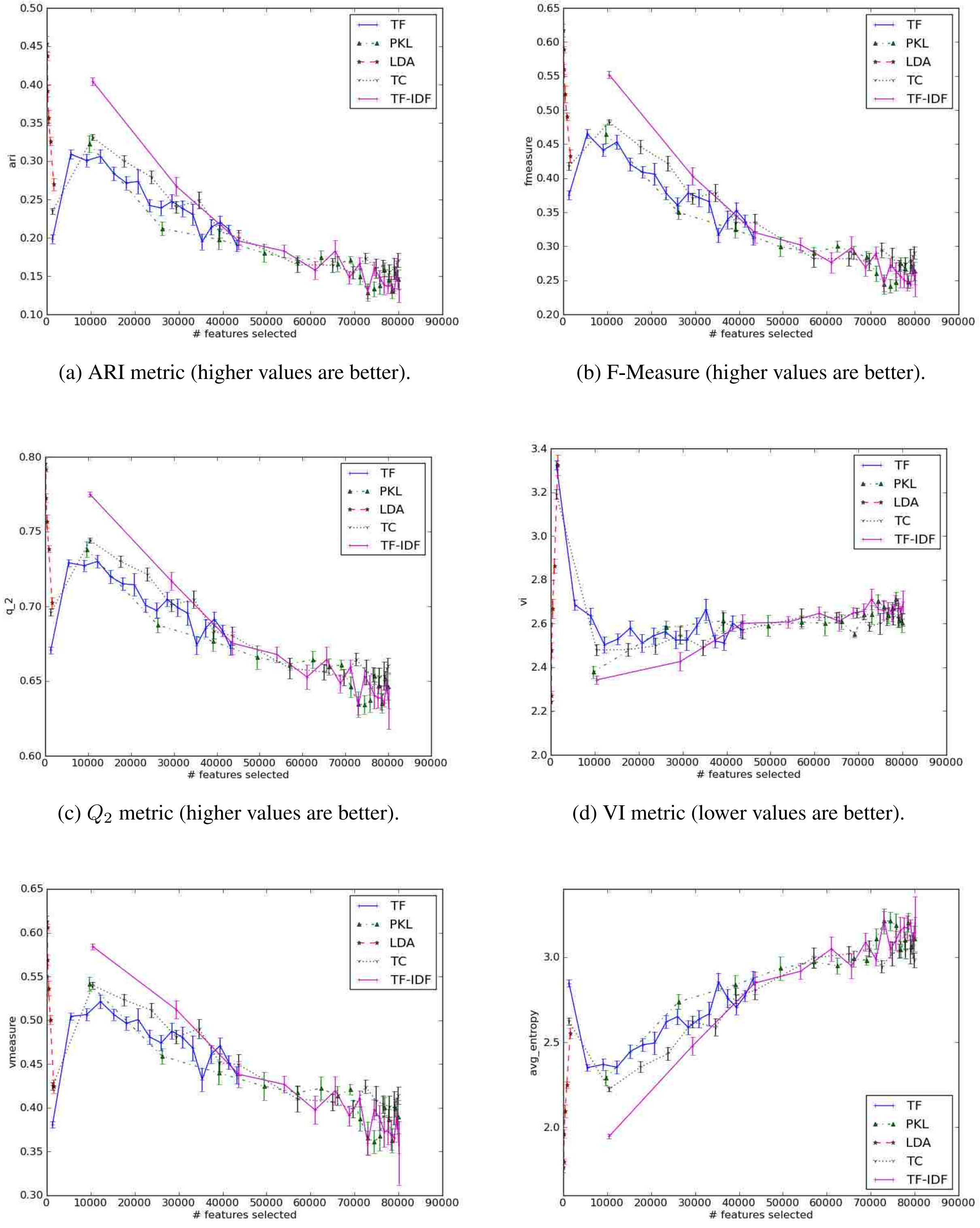
Dataset	TNPD TFIDF(s)	TNPD PKL(s)	TNPD TF(s)	TC(s)
20 News	23.1	6.9	2.7	1076.3
Reuters	5.7	2.5	1.0	237.4
Enron	35.2	4.1	1.5	176.6

Table 3.2: The amount of time required to run each feature-selection algorithm.

about document similarities that are not apparent without information about how feature types are correlated.

As the number of topics increases, tokens of different types become less and less likely to be labeled with the same topic. Even worse, tokens of the same type become more likely to be labelled differently in the same iteration. This has the effect of fracturing feature types and the mixing of feature tokens into types become more random, leading to losses in the amount of information about document similarity available to the algorithm. When the number of features increases even more, many topics begin to be “empty", with no tokens being assigned to these empty topics. Increasing the number of topics past this point seems to mostly increase the number of empty topics, which explains the flat-lining of the LDA dimensionality reducer. While this is not a problem per-se, since higher levels of reduction are generally more desirable, it does make finding the “correct" number of dimensions to which to reduce more important than for the feature selecting methods.

These results may not appear very impressive in and of themselves, as they do not indicate whether any of the methods in question are clearly superior to any of the others in terms of quality. They become more impressive, though, in light of the difference in complexity and time usage between the TNPD and TC. Table 3.2 shows the amount of time that was used to perform feature selection using TNPD and TC. All of the TNPD methods complete feature selection in orders of magnitude less time than TC. A comparison of the complexity of these methods reveals the source of this difference. When using an inverse document index, TC is $O(m\overline{N}^2)$, where $\overline{N}$ is the average number of documents that the features in the candidate set occur in and m is the number of features in the candidate set. This does not include the complexity of building the index. The complexity of TNPD with the PKL weight function is $O(\overline{|d|}|D|)$, where $\overline{|d|}$ is the average number of feature



(a) ARI metric (higher values are better).

(b) F-Measure (higher values are better).

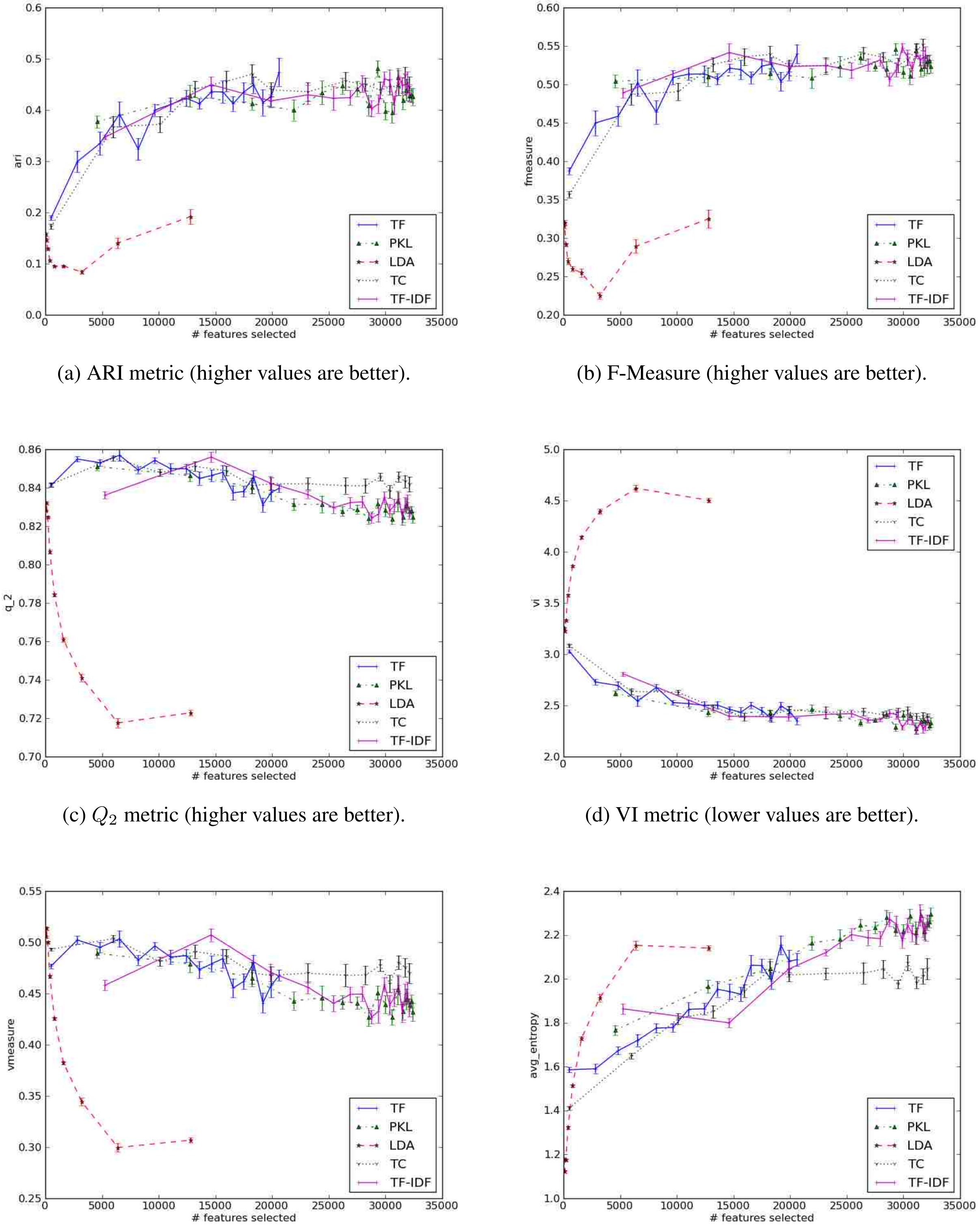
(c) Q_2 metric (higher values are better).

(d) VI metric (lower values are better).

(e) V-Measure (higher values are better).

(f) Average entropy (lower values are better).

Figure 3.2: Results for the six metrics on the 20 Newsgroups dataset.



(a) ARI metric (higher values are better).

(b) F-Measure (higher values are better).

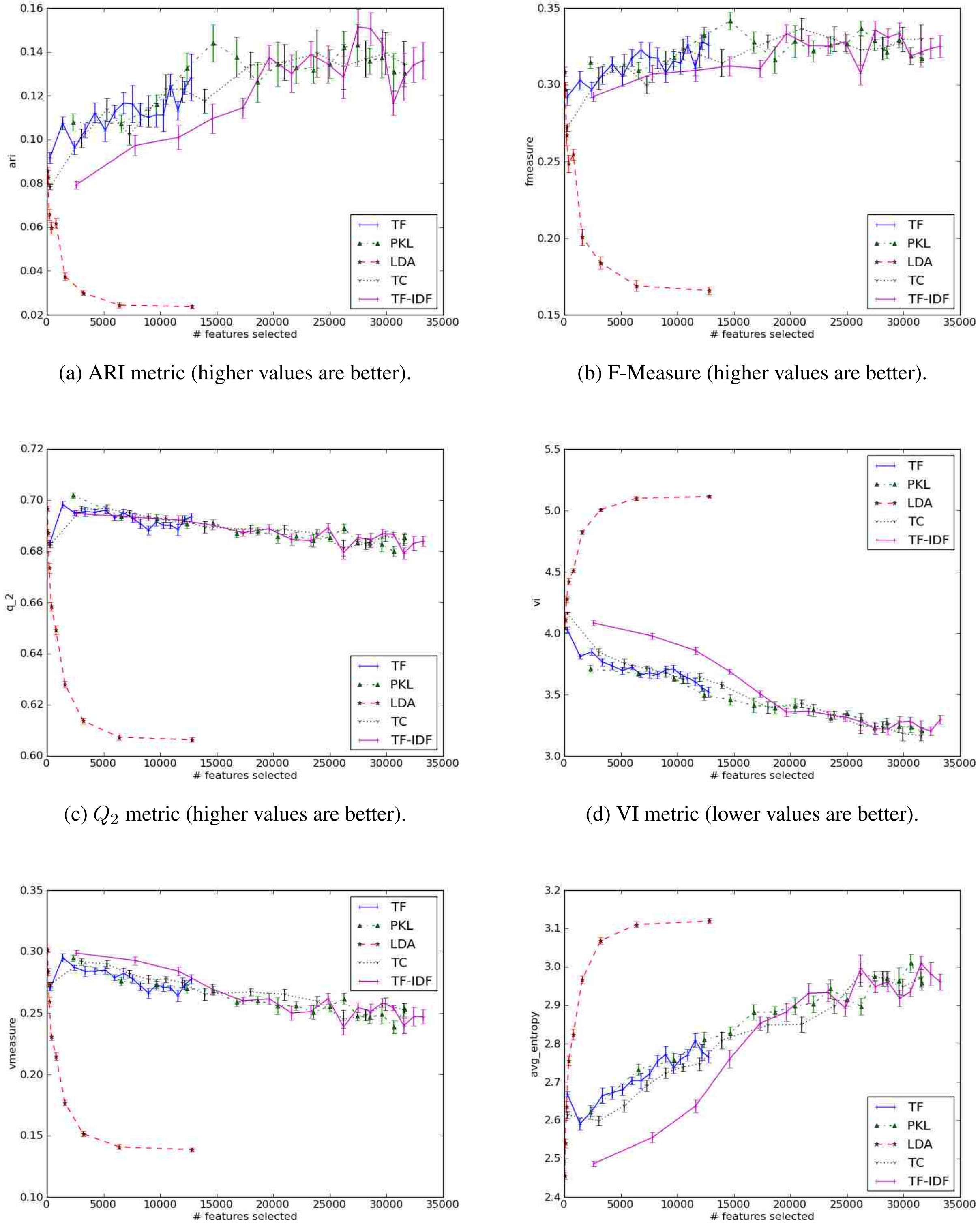
(c) Q_2 metric (higher values are better).

(d) VI metric (lower values are better).

(e) V-Measure (higher values are better).

(f) Average entropy (lower values are better).

Figure 3.3: Results for the six metrics on the Reuters-21578 dataset.



(a) ARI metric (higher values are better).

(b) F-Measure (higher values are better).

(c) Q_2 metric (higher values are better).

(d) VI metric (lower values are better).

(e) V-Measure (higher values are better).

(f) Average entropy (lower values are better).

Figure 3.4: Results for the six metrics on the Enron dataset

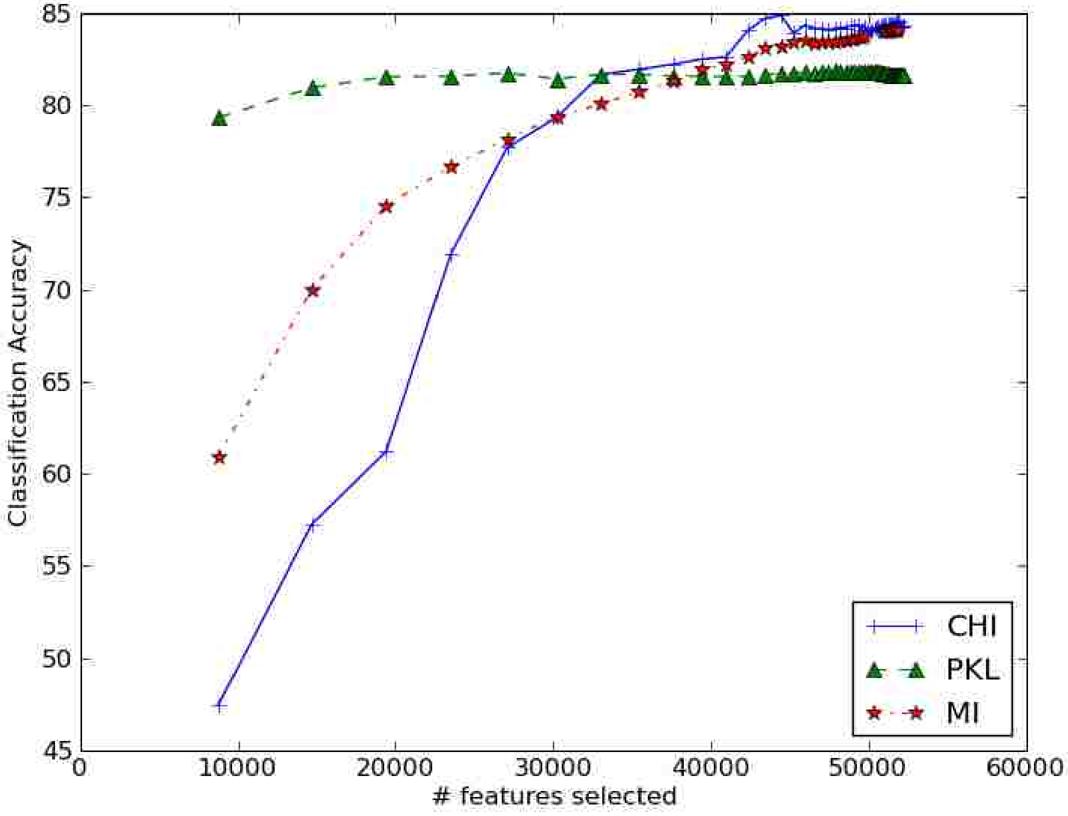


Figure 3.5: A comparison of the performance of PKL TNPD with χ^2 (CHI), and mutual information (MI) on a supervised learning task using the 20 Newsgroups dataset.

types occurring in documents in the collection. Since, typically, $\overline{|d|} \ll M$ (document lengths in our datasets were typically in the hundreds) the TNPD method is doing much less work.

3.4.3 Classification Results

Figure 3.5 shows the results of the classification experiment.

Although the MI and χ^2 methods do produce significantly higher accuracies for some values of N, TNPD is actually very competitive for the smallest values. The elimination of low frequency features accounts for the difference in the shape of the TNPD curves here, compared to those in the previous section. Also, the range of the x axis has changed, as in this task the number of features in the candidate feature set was 59,192.

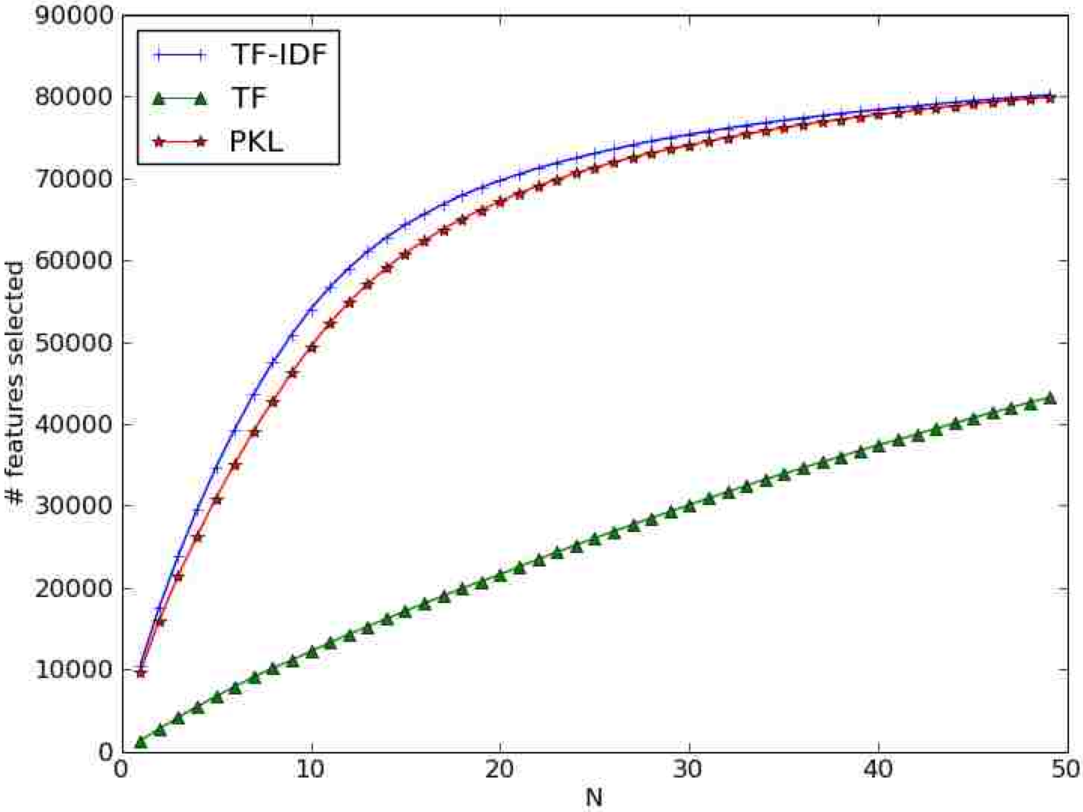


Figure 3.6: The relationship between the N parameter in Top N per Document and the actual number of features remaining in the vocabulary after feature selection.

3.5 Caveats and Considerations

One deficiency of the TNPD method is that it does not give fine grained control over the number of features that will remain after feature selection. Figure 3.6 shows how, on the 20 Newsgroup dataset, the N parameter affects n_N, the final number of features. This graph also illustrates why the TF trend lines in the clustering results cover such a narrow range. At $N = 50$ TF selects only about 45,000 features, while TF-IDF and PKL select about twice that many. Thus, when fined grained control over feature set sizes is important, or when it is desirable to reduce features further than the reduction possible with TNPD with $N = 1$ for a given weight function, another feature selection method may be a better fit.

3.6 Conclusions

We have shown that Top N Per Document is an effective method of unsupervised feature selection across several datasets. This was demonstrated both in the context of classification and clustering, where it performed as well as the more computationally intensive state of the art Term Contribution feature selector, while running in 2 orders of magnitude less time than Term Contribution, even with relatively expensive local term weighting functions. It appears as though any of the three document-local feature weighting functions discussed here perform reasonably well, although the point-wise KL divergence function might be slightly more consistent in its ability to match and occasionally exceed the performance of Term Contribution. TNPD has also been shown to be very efficient compared to Term Contribution.

Thus, the Top N Per Document feature selection algorithm has the following benefits

- Model interpretability is preserved, as the reduced feature set is a subset of the original.
- It achieves state-of-the-art performance in the class of filtering feature selection algorithms.
- It is very simple to implement.
- It is very fast, both asymptotically and in practice.

The first point is shared by all reasonable feature selectors, in contrast to other dimensionality reduction techniques, such as PLSI or the LDA method used here. While the LDA results may be more interpretable than LSI or PLSI, it has its own interpretability and evaluation related problems [21]. The remaining three points together summarize the reasons that a practitioner might prefer TNPD to other feature selection algorithms.

Chapter 4

Evaluating Models of Latent Document Semantics in the Presence of OCR Errors

Abstract

Models of latent document semantics such as the mixture of multinomials model and Latent Dirichlet Allocation have received substantial attention for their ability to discover topical semantics in large collections of text. In an effort to apply such models to noisy optical character recognition (OCR) text output, we endeavor to understand the effect that character-level noise can have on unsupervised topic modeling. We show the effects both with document-level topic analysis (document clustering) and with word-level topic analysis (LDA) on both synthetic and real-world OCR data. As expected, experimental results show that performance declines as word error rates increase. Common techniques for alleviating these problems, such as filtering low-frequency words, are successful in enhancing model quality, but exhibit failure trends similar to models trained on unprocessed OCR output in the case of LDA. To our knowledge, this study is the first of its kind.

4.1 Introduction

As text data becomes available in massive quantities, it becomes increasingly difficult for human curators to manually catalog and index modern document collections. To aid in the automation of such tasks, algorithms can be used to create models of the latent semantics present in a given corpus. One example of this type of analysis is document clustering, in which documents are grouped into clusters by topic. Another type of topic analysis attempts to discover finer-grained topics—labeling individual words in a document as belonging to a particular topic. This type of analysis has grown in popularity recently as inference on models containing large numbers of latent variables has become feasible.

The modern explosion of data includes vast amounts of historical documents, made available by means of Optical Character Recognition (OCR), which can introduce significant numbers of errors. Undertakings to produce such data include the Google Books, Internet Archive, and HathiTrust projects. In addition, researchers are having increasing levels of success in digitizing hand-written manuscripts [18], though error rates remain much higher than for OCR. Due to their nature, these collections often lack helpful meta-data or labels. In the absence of such labels, unsupervised machine learning methods can reveal patterns in the data.

Finding good estimates for the parameters of models such as the mixture of multinomials document model [91] and the Latent Dirichlet Allocation (LDA) model [16] requires accurate counts of the occurrences and co-occurrences of words. Depending on the age of a document and the way in which it was created, the OCR process results in text containing many types of noise, including character-level errors, which distort these counts. It is obvious, therefore, that model quality must suffer, especially since unsupervised methods are typically much more sensitive to noise than supervised methods. Good supervised learning algorithms are substantially more immune to spurious patterns in the data created by noise for the following reason: under the mostly reasonable assumption that the process contributing the noise operates independently from the class labels, the noise in the features will not correlate well with the class labels, and the algorithm will learn to ignore those features arising from noise. Unsupervised models, in contrast, have no grounding in labels to prevent them from confusing patterns that emerge by chance in the noise with the “true" patterns of potential interest. For example, even on clean data, LDA will often do poorly if the very simple feature selection step of removing stop-words is not performed first. Though we expect model quality to decrease, it is not well understood how sensitive these models are to OCR errors, or how quality deteriorates as the level of OCR noise increases.

In this work we show how the performance of unsupervised topic modeling algorithms degrades as character-level noise is introduced. We demonstrate the effect using both artificially corrupted data and an existing real-world OCR corpus. The results are promising, especially in the case of relatively low word error rates (e.g. less than 20%). Though model quality declines as errors increase, simple feature selection techniques enable the learning of relatively high quality models even as word error rates approach 50%. This result is particularly interesting in that even humans find it difficult to make sense of documents with error rates of that magnitude [66].

Because of the difficulties in evaluating topic models, even on clean data, these results should not be interpreted as definitive answers, but they do offer insight into prominent trends. For example, properties of the OCR data suggest measures that can be taken to improve performance in future work. It is our hope that this work will lead to an increase in the usefulness of collections of OCRed texts, as document clustering and topic modeling expose useful patterns to historians and other interested parties.

The remainder of the paper is outlined as follows. After an overview of related work in Section 4.2, Section 4.3 introduces the data used in our experiments, including an explanation of how the synthetic data were created and of some of their properties. Section 4.4 describes how the experiments were designed and carried out, and gives an analysis of the results both empirically and qualitatively. Finally, conclusions and future work are presented in Section 4.5.

4.2 Related Work

Topic models have been used previously to process documents digitized by OCR, including eighteenth-century American newspapers [68], OCRed editions of *Science* [14], OCRed NIPS papers [99], and books digitized by the Open Content Alliance [64]. Most of this previous work ignores the presence of OCR errors or attempts to remove corrupted tokens with special pre-processing such as stop-word removal and frequency cutoffs. Also, there are at least two instances of using topic modeling to improve the results of an OCR algorithm [32, 102].

Similar evaluations to ours have been conducted to assess the effect of OCR errors on supervised document classification [3, 87], information retrieval [10, 86], and a more general set of natural language processing tasks [57]. Results suggest that in these supervised tasks OCR errors have a minimal impact on the performance of the methods employed, though it has remained unclear how well these results transfer to unsupervised methods.

4.3 Data

We conducted experiments on synthetic and real OCR data. As a real-world dataset, we used a corpus consisting of 604 of the Eisenhower World War II communiqués [49, 58]. These documents relate the daily progress of the Allied campaign from D-Day until the German surrender. They were originally produced as telegrams and were distributed as mimeographed copies. The quality of the originals is often quite poor, making them a challenging case for OCR engines. The communiqués have been OCRed using three popular OCR engines: ABBYY FineReader [1], OmniPage Pro [70], and Tesseract [38]. In addition, the curator of the collection has created a “gold standard" transcription, from which it is possible to obtain accurate measures of average document word error rates (WER) for each engine, which are: 19.9%, 30.4%, and 50.1% respectively.

While the real-world data is attractive as an example of just the sort of data that the questions addressed here apply to, it is limited in size and scope. All of the documents in the Eisenhower corpus discuss the fairly narrow topic of troop movements and battle developments taking place at the end of World War II. Neither the subject matter nor the means of conveyance allowed for a large or diverse vocabulary of discourse.

In an attempt to generalize our results to larger and more diverse data, we also ran experiments using synthetic OCR data. This synthetic data was created by corrupting “clean" datasets, adding character-level noise. The synthetic data was created by building a noise model based on mistakes made by the worst performing OCR engine on the Eisenhower dataset, Tesseract.

To construct the noise model, a character-level alignment between the human transcribed Eisenhower documents and the OCR output was first computed. From this alignment, the contingency table $\mathbf{M}^d$ was generated such that $\mathbf{M}^d_{x,y}$ was the count of the instances in which a character x in the transcript was aligned with a y in the OCR output. The rows in this matrix were then normalized so that each represented the parameters of a categorical distribution, conditioned on x. To parameterize the amount of noise being generated, the $\mathbf{M}^d$ matrix was interpolated with an identity matrix $\mathbf{I}$ using a parameter γ so that the final interpolated parameters $\mathbf{M}^i$ were calculated with the formula $\mathbf{M}^i = \gamma\mathbf{M}^d + (1-\gamma)\mathbf{I}$. So that at $\gamma = 0$, $\mathbf{M}^i = \mathbf{I}$ and no errors were introduced. At $\gamma = 1.0$, $\mathbf{M}^i = \mathbf{M}^d$, and we would expect to see characters corrupted at the same rate as in the output of the OCR engine.

We then iterated over each document, choosing a new (possibly the same) character y_l for each original character x_l according to the probability distribution $p(y_l = w'|x_l = w) = M^i_{w,w'}$. Our process was a one-substitution algorithm, as we did not include instances of insertions or deletions, consequently words were changed but not split or deleted. This allowed for a more straightforward calculation of word error rate. Segmentation errors can still occur in the learning stage, however, as the noise model sometimes replaced alphabet characters with punctuation characters, which were treated as delimiters by our tokenizer.

We chose three datasets to corrupt: 20 Newsgroups [51], Reuters 21578 [52], and the LDC-annotated portion of the Enron e-mail archive [11]. Each of these datasets were corrupted at values $\gamma = i * 0.01$ for $i \in (0, 13)$. At this point, the word error rate of the corrupted data was near 50% and, since this was approximately the WER observed for the worst OCR engine on the real-world data, we chose to stop there. The word error rate was calculated during the corruption process. Here is an example sentence corrupted at two γ values:

$\gamma = 0.000$ I am also attaching the RFP itself.

$\gamma = 0.02$ I am also attachEng the RFP itself.

Dataset	|D|	K	**# Types**	**# Tokens**
20 News	19997	20	107211	2261805
Reuters	11367	81	29034	747458
Enron	4935	32	60495	2063667
Eisenhower	604	N/A	8039	76674

Table 4.1: Summary of test dataset characteristics. $|D|$ is the number of documents in the dataset. K is the number of human-labeled classes provided with the dataset.

$\gamma = 0.10$ I Jm alAo attaching the RFP itself.

Table 4.1 shows some basic statistics for the datasets. The values shown are for the “clean" versions of the data. For an example of how noise and pre-processing techniques affect these counts see Section 4.4.1.

It is interesting to note that the word error rates produced by the noise model appear to be significantly higher than first expected. One might assume that the WER should increase fairly steadily from 0% at $\gamma = 0$ to about 50% (the error rate achieved by the Tesseract OCR engine on the Eisenhower dataset) at $\gamma = 1$. There are at least two sources for the discrepancy. First, the vocabulary of the Eisenhower dataset does not match well with that of any of the source datasets from which the synthetic data were generated. This means that the word and character distributions are different and so the error rates will be as well. Secondly, whereas our technique gives the same probability of corruption to all instances of a given character, errors in true OCR output are bursty and more likely to be concentrated in specific tokens, or regions, of a document. This is because most sources of noise do not affect document images uniformly. Also, modern OCR engines do not operate at just the character level. They incorporate dictionaries and language models to prevent them from positing words that are highly unlikely. As a consequence, an OCR engine is much more likely to either get a whole word correct, or to miss it completely, concentrating multiple errors in a single word. This is the difference between 10 errors in a single word, which only contributes 1 to the numerator of the WER formula and 10 errors spread across 10 different words, which contributes 10 to the numerator. Furthermore, because content bearing words tend to be relatively rare, language models are poorer for them than for frequent function words, meaning that the words most correlated with semantics are also the most likely to be corrupted by an OCR engine.

An example of this phenomenon is easy to find. In the Enron corpus, there are 165,871 instances of the word “the" and 102 instances of the string “thc". Since “c” has a high rate of confusion with “e”, we would expect at least some instances of “the" to be corrupted to “thc" by the error model. At $\gamma = 0.03$, there are 156,663 instances of the word “the" and 513 instances of “thc". So, the noise model converts “the" to “thc" roughly 0.3% of the time. In contrast, there are no instances of “thc" in the Tesseract OCR output even though there are 5186 instances of “the" in the transcription text, and so we would expect approximately 16 occurrences of “thc" if the errors introduced by the noise model were truly representative of the errors in the actual OCR output.

Another interesting property of the noise introduced by actual OCR engines and our synthetic noise model is the way in which this noise affects words distributions. This is very important, since word occurrence and co-occurrence counts are the basis for model inference in both clustering and topic modeling. As mentioned previously, one common way of lessening the impact of OCR noise when training topic models over OCRed data is to apply a frequency cutoff filter to cull words that occur fewer than a certain number of times. Figures 4.1 and 4.2 show the number of word types that are culled from the synthetic 20 Newsgroups OCR data and the Eisenhower OCR data, respectively, at various levels of noise. Note that the cutoff filters use a strict “less than", so a frequency cutoff of 2 eliminates only words that occur once in the entire dataset. Also, these series are additive, as the words culled with a frequency cutoff of 2 are a subset of those culled with a frequency cutoff of $j > 2$.

In both cases, it is apparent that by far the largest impact that noise has is in the creation of singletons. It seems that the most common corruptions in these scenarios is the creation of one-off word types through a unique corruption of a (most likely rare) word. This means that it is unlikely that enough evidence will be available to associate, through similar contexts, the original word and its corrupted forms.

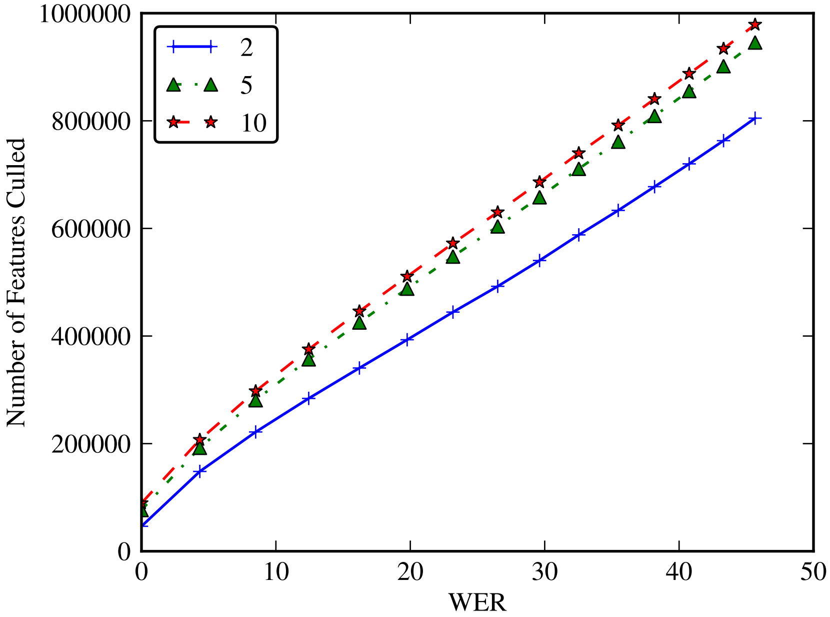


Figure 4.1: The number of word types culled with frequency cutoff filters applied to the 20 Newsgroups data with various levels of errors introduced. The line labeled 2 represents the number of types occurring only once (singletons), the line labeled 5 represents the number of types occurring 4 or fewer times, and the line labeled 10 represents the number of types occurring 9 or fewer times in the corpus.

Due to the fact that most clustering and topic models ignore the forms of word tokens (the characters that make them up), and only take into account word identities, we believe that the similarity in the way in which real OCR engines and our synthetic OCR noise model distort word distributions is sufficient evidence to support the use of the synthetic data until larger and better real-world OCR datasets can be made available. Though the actual errors will take a different form, the character-level details of the errors are less relevant than the word distribution alterations for the models in question.

4.4 Experimental Results

We ran experiments on both the real and synthetic OCR data. In this section we explain our experimental methodology and present both empirical and qualitative analyses of the results.

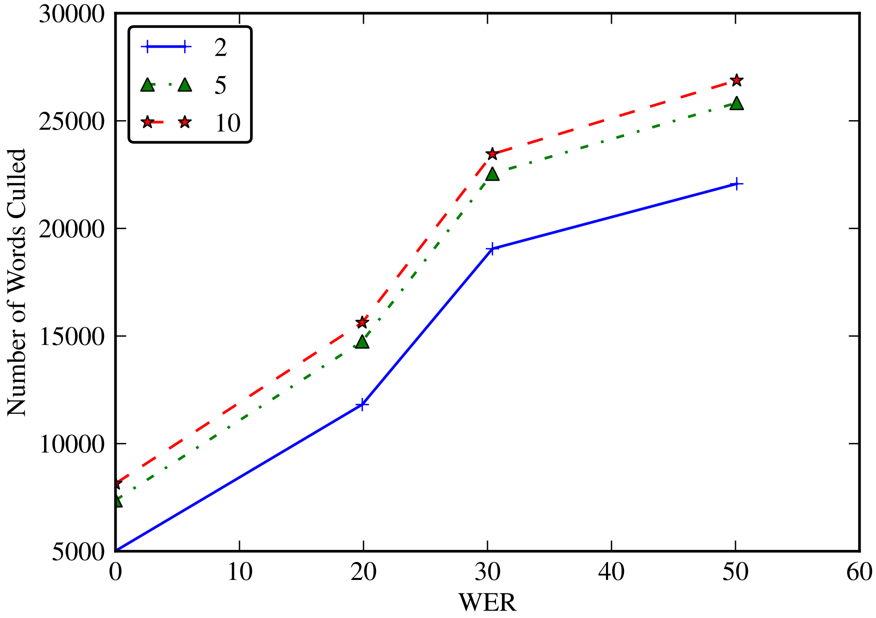


Figure 4.2: The number of word types culled with frequency cutoff filters applied to the transcript and three OCR engine outputs for the Eisenhower data.

4.4.1 Methodology

For the synthetic OCR datasets, we ran clustering experiments using EM on a mixture of multinomials (c.f. [91]). We specified the number of clusters to be the same as the number of classes provided with the data. Clusters were evaluated using several external cluster quality metrics which compare "gold standard" labels to those created through clustering. The metrics used were Variation of Information (VI) [62], and the Adjusted Rand Index (ARI) [45]. Other metrics were also calculated (e.g. the V-Measure [81], and Average Entropy [54]), but these results were excluded due to space constraints and the fact that their plots are similar to those shown. We did not cluster the Eisenhower data because of the absence of the class labels necessary for evaluation.

For both the synthetic and non-synthetic data we also trained LDA topic models [16] using Gibbs sampling. We used the implementation found in the MALLET software package [61] with the option enabled to learn the priors during sampling as discussed by Wallach et al. [97]. Each LDA model was trained on 90% of the documents in each dataset. The trained model was used to calculate an estimate of the marginal log-likelihood of the remaining 10% of the documents using the left-to-right algorithm [98]. The number of topics used for each dataset was adjusted *a priori* according to the number of documents it contained. We used 100 topics for Enron and Eisenhower, 150 for Reuters, and 200 for 20 Newsgroups.

In addition to running experiments on the “raw" synthetic data, we also applied simple unsupervised feature selectors before training in order to evaluate the effectiveness of such measures in mitigating problems caused by OCR errors. For the topic modeling (LDA) experiments three feature selectors were used. The first method employed was a simple term frequency cutoff filter (TFCF), with a cutoff of 5 as in [99]. The next method employed was Term Contribution (TC), a feature selection algorithm developed for document clustering [54]. Term contribution is parameterized by the number of word types that are to remain after selection. We attempted three values for this parameter, 10,000, 20,000, and 50,000. The final method we employed was a method called Top-N per Document (TNPD) [92], which is a simple feature selection algorithm that first assigns each type in every document a document-specific score (e.g. its TF-IDF weight), and then selects words to include in the final vocabulary by choosing the N words with the highest score from each document in the corpus. We found that $N = 1$ gave the best results at the greatest reduction in word types. After the vocabulary is built, all words not in the vocabulary are culled from the documents. This does not mean that all documents contain only one word after feature selection, as the top word in one document may occur in many other documents, even if it is not the top word in those documents. Likewise, if two documents would both contribute the same word, then the second document makes no contribution to the vocabulary. This process can result in vocabularies with thousands of words, leaving sufficient words in each document for analysis. For the clustering experiments, initial tests showed little difference in the performance of the feature selectors, so only the TNPD selector was used. Figures 4.3a and 4.3b show how the various pre-processing methods affect word type and token counts, respectively, for the 20 Newsgroups dataset. In contrast, without pre-processing the number of types scales from 107,211 to 892,983 and the number of tokens from 2,261,805 to 3,073,208.

Because all of these procedures alter the number of words and tokens in the final data, log-likelihood measured on a held-out set cannot be used to accurately compare the quality of topic models trained on pre-processed data, as the held-out data will contain many unknown words. If the held-out data is also pre-processed to only include known words, then the likelihood will be greater for those procedures that remove the most tokens, as the product that dominates the calculation will have fewer terms. If unknown words are allowed to remain, even a smoothed model will assign them very little probability and so models will be heavily penalized.

We use an alternative method for evaluating the topic models, discussed in [40], to avoid the aforementioned problems with an evaluation based on log-likelihood. Since the synthetic data is derived from datasets that have topical document labels, we are able to use the output from LDA in a classification problem with the word vectors for each document being replaced by the assigned topic vectors. This is equivalent to using LDA as a dimensionality reduction pre-process for document classification. A naive Bayes learner is trained on a portion of the topic vectors, labeled with the original document label, and then the classification accuracy on a held-out portion of the data is computed. Ten-fold cross-validation is used to control for sampling issues. The rationale behind this evaluation is that, even though the topics discovered by LDA will not correspond directly to the labels, there should at least be a high degree of correlation. Models that discover topical semantics that correlate well with the labels applied by humans will yield higher classification accuracies and be considered better models according to this metric.

To compensate for the randomness inherent in the algorithms, each experiment was replicated ten times. The results of these runs were averaged to produce the values reported here.

4.4.2 Empirical Analysis

Both the mixture of multinomials document model and LDA appear to be fairly resilient to character-level noise. Figures 4.4 and 4.5 show the results of the document clustering experiments with and without feature selection, respectively. Memory issues prevented the collection of results for the highest error rates on the Enron and Reuters data without feature selection.

With no pre-processing, the results are somewhat mixed. The Enron dataset experiences almost no quality degradation as the WER increases, yielding remarkably constant results according to the metrics. However, this may be an artifact of the relatively poor starting performance for this dataset, a result of the fact that the gold standard labels do not align well with automatically discovered patterns because they correspond to external events. In contrast, the Reuters data appears to experience drastic degradation in performance. Once feature selection occurs, however, performance remains much more stable as error rates increase.

Figure 4.6a shows the results of running LDA on the transcript and digitized versions of the Eisenhower dataset. Log-likelihoods of the held-out set are plotted with respect to the WER observed for each OCR engine. The results support the finding that the WER of the OCR engine that produced the data has a significant negative correlation with model quality. Unfortunately, it was not possible to compare the performance of the pre-processing methods on this dataset, due to a lack of document topic labels and the deficiencies of log-likelihood mentioned previously.

Figure 4.6b shows the results of the LDA topic-modeling experiments on the three “raw" synthetic datasets. Similar trends are observed in this graph. LDA experiences a marked degree of performance degradation, with all of the trend lines indicating a linear decrease in log-likelihood.

Figures 4.7 through 4.9 show the results of evaluating the various proposed pre-processing procedures in the context of topic modeling. In the graph “noop.0" represents no pre-processing, “tc.N" are the Term Contribution method parameterized to select N word types, “tfcf.5" is the term frequency cutoff filter with a cutoff of 5 and “tnpd.1" is the Top N per Document method with $N = 1$. The y-axis is the average of the results for 10 distinct trials, where the output for each trial is the average of the ten accuracies achieved using ten-fold cross-validation as described in Section 4.4.1.

Here, the cross-validation accuracy metric reveals a slightly different story. These results show that topic quality on both the raw and pre-processed noisy data degrades at a rate relative to the amount of errors in the data. That is, the difference in performance between two relatively low word error rates (e.g. 5% and 7% on the Reuters data) is small, whereas the differences between two high error rates (e.g. 30% and 32% on the Reuters data) can be relatively quite large.

While pre-processing does improve model quality, in the case of LDA this improvement amounts to a nearly constant boost; at high error rates quality is improved the same amount as at low error rates. Degradations in model quality, therefore, follow the same trends, occurring at mostly the same rate in pre-processed data as in the raw noisy data. This is in contrast to the clustering experiments where pre-processing virtually eliminates failure trends.

4.4.3 Qualitative Analysis

Higher values measured with automated metrics such as log-likelihood on a held-out set and the cross-validation classification metric discussed here do not necessarily indicate superior topics according to human judgement [21]. In order to provide a more thorough discussion of the relative quality of the topic models induced on the OCR data versus those induced on clean data, we sampled the results of several of the runs of the LDA algorithm. In Tables 4.2 and 4.3 we show the top words for the five topics with the highest learned topic prior (α in the LDA literature) learned during Gibbs sampling. This information is shown for the Reuters data first with no corruption and then at the highest error rate for which we have results for that data of 45% WER.

In general, there appears to be a surprisingly good correlation between the topics learned on the clean data and those learned on the corrupted data, given the high level of noise involved. The topics are generally cohesive, containing mostly terms relating to financial market news. However, the topics trained on the clean data, though all related to financial markets, are fairly distinctive. Topic 3, for example seems to be about fluctuations in stock prices, and Topics 106 and 34 about business acquisitions and mergers. The topics trained on the noisy data are fairly homogeneous and the differences between them are more difficult to identify.

In addition, it appears as though the first topic (topic 93) is not very coherent at all. This topic is significantly larger, in terms of the number of tokens assigned to it than the other topics shown in either table. In addition, the most probable words listed for the topic seem less cohesive than for the other topics. It contains many two-letter words that are likely a mixture of valid terms (e.g., stock exchange and ticker symbols, and parts of place names like “Rio de Janeiro") and corruptions of real words. For example, even though there are no instances of “ts" as a distinct token in the clean Reuters data, it is in the list of the top 19 words for topic 93. This is perhaps due to the fact that “is" can easily be converted to “ts" by mistaking “t" for “i".

It is also the case that, for most topics learned on the corrupted data, the most probable words for those topics tend to be shorter, on average, than for topics learned on clean data. We believe this is due to the fact that the processes used to add noise to the data (both real OCR engines and our synthetic noise model) are more likely to corrupt long words, especially in the case of the synthetic data which was created using a character-level noise model.

Examination of the data tends to corroborate this hypothesis, though even long words usually contain only a few errors. For example, in the 20 Newsgroups data there are 379 instances of the word “yesterday", a long word that is not close to other English words in edit distance. When the data has been corrupted to a WER of 47.9%, however, there are only 109 instances of “yesterday" and 132 tokens that are within an edit distance of 1 from “yesterday".

To some extent, we would expect to observe similar trends in real-world data. However, most OCR engines employ language models and dictionaries to attempt to mitigate OCR errors. As a result, given that a word recognition error has occurred in true OCR output, it is more likely to be an error that lies at an edit distance greater than one from the true word, or else it would have been corrected internally. For example, there are 349 instances of the word “yesterday" in the Eisenhower transcripts, and 284 instances in the Tesserect OCR output and only 5 tokens within an edit distance of one, meaning that 60 corruptions of this word contained more than one error, making up 90% of the errors for that word. However, many of these errors still contain most of the letters from the original word (e.g. “yesterdj.", and “yestjkday"). In all cases, the corrupted versions of a given word are very rare, occurring usually only once or twice in the noisy output, making them useless features for informing a model.

Topic	Words	Tokens
64	told, market, reuters, reuter, added, time, year, major, years, president, make, made, march, world, today, officials, industry, government, move	67159
3	year, pct, prices, expected, rise, lower, higher, demand, increase, due, fall, decline, current, high, end, added, level, drop, market	32907
106	reuter, corp, company, unit, sale, march, dlrs, mln, sell, subsidiary, acquisition, terms, group, april, purchase, acquired, products, division, business	22167
34	shares, dlrs, company, mln, stock, pct, share, common, reuter, corp, agreement, march, shareholders, buy, cash, outstanding, merger, acquire, acquisition	22668
7	mln, cts, net, shr, qtr, revs, reuter, avg, shrs, march, mths, dlrs, sales, st, corp, oct, note, year, april	18511

Table 4.2: Top words for the five topics with the highest α prior values found using MALLET for one run of LDA on the uncorrupted Reuters data.

4.5 Conclusions and Future Work

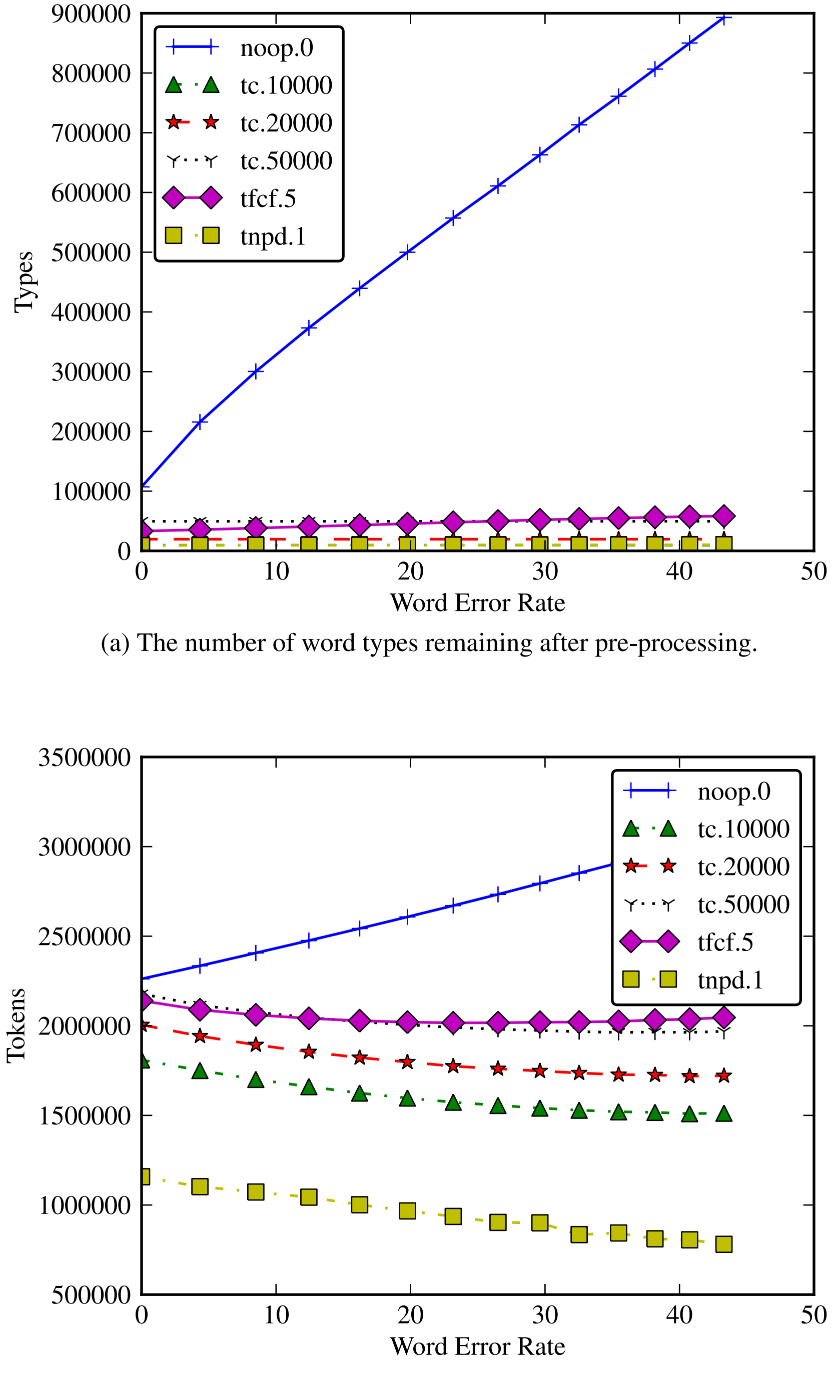
The primary outcome of these experiments is an understanding regarding when clustering and LDA topic models can be expected to function well on noisy OCR data. Our results imply that clustering methods should perform almost as well on OCR data as they do on clean data, provided that a reasonable feature selection algorithm is employed. The LDA topic model degraded less gracefully in performance with the addition of character level errors to its input, with higher error rates impacting model quality in a way that was apparent empirically in the log-likelihood and ten-fold cross-validation metrics as well as through human inspection of the produced topics. Pre-processing the data also helps model quality for LDA, yet still yields failure trends similar to those observed on unprocessed data.

We found it to be the case that even in data with high word error rates, corrupted words often share many characters in common with their uncorrupted form. This suggests an approach in which

Topic	Words	Tokens
93	reuter, march, pct, year, april, ed, market, er, told, es, st, end, ts, al, de, ng, id, sa, added	258932
99	company, pct, corp, shares, stock, dlrs, share, offer, group, reuter, mln, march, unit, stake, buy, cash, bid, sale, board	50377
96	mln, cts, net, shr, qtr, dlrs, revs, reuter, note, oper, avg, march, shrs, year, mths, st, sales, corp, oct	54659
141	mln, dlrs, year, net, quarter, share, company, billion, tax, sales, earnings, dlr, profit, march, income, ln, results, sale, corp	40475
53	pct, year, rose, rise, january, february, fell, march, index, december, month, figures, compared, reuter, rate, earlier, show, ago, base	22556

Table 4.3: Top words for the five topics with the highest α prior values found using MALLET for one run of LDA on the Reuters data corrupted with the data-derived noise model to a WER of 45%.

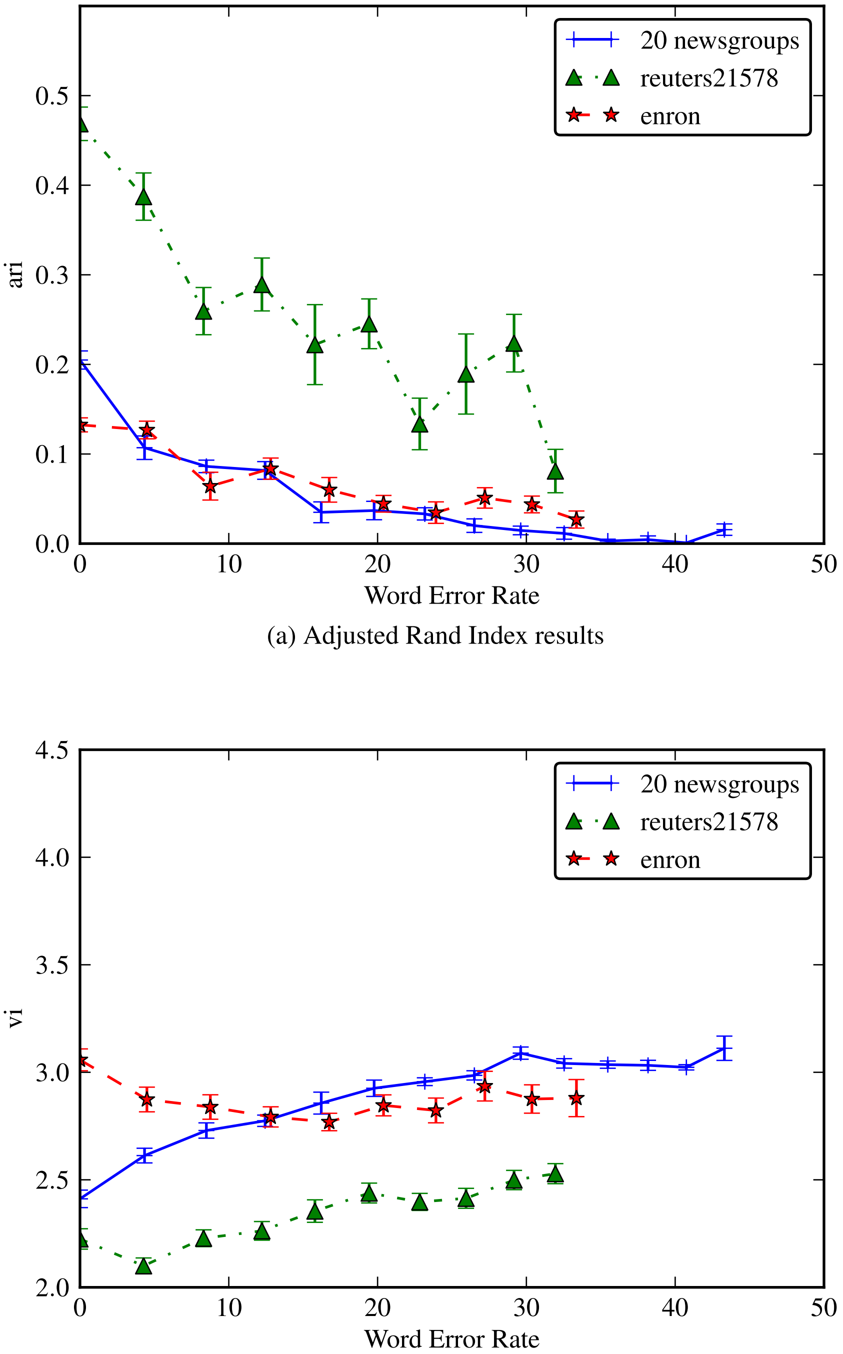
word similarities are used to cluster the unique corrupted versions of a word in order to increase the evidence available to the topic model during training time and improve model quality. As the quality of models increases on these noisy datasets, we anticipate a consequent rise in their usefulness to researchers and historians as browsing the data and mining it for useful patterns becomes more efficient and profitable.



(a) The number of word types remaining after pre-processing.

(b) The number of word tokens remaining after pre-processing.

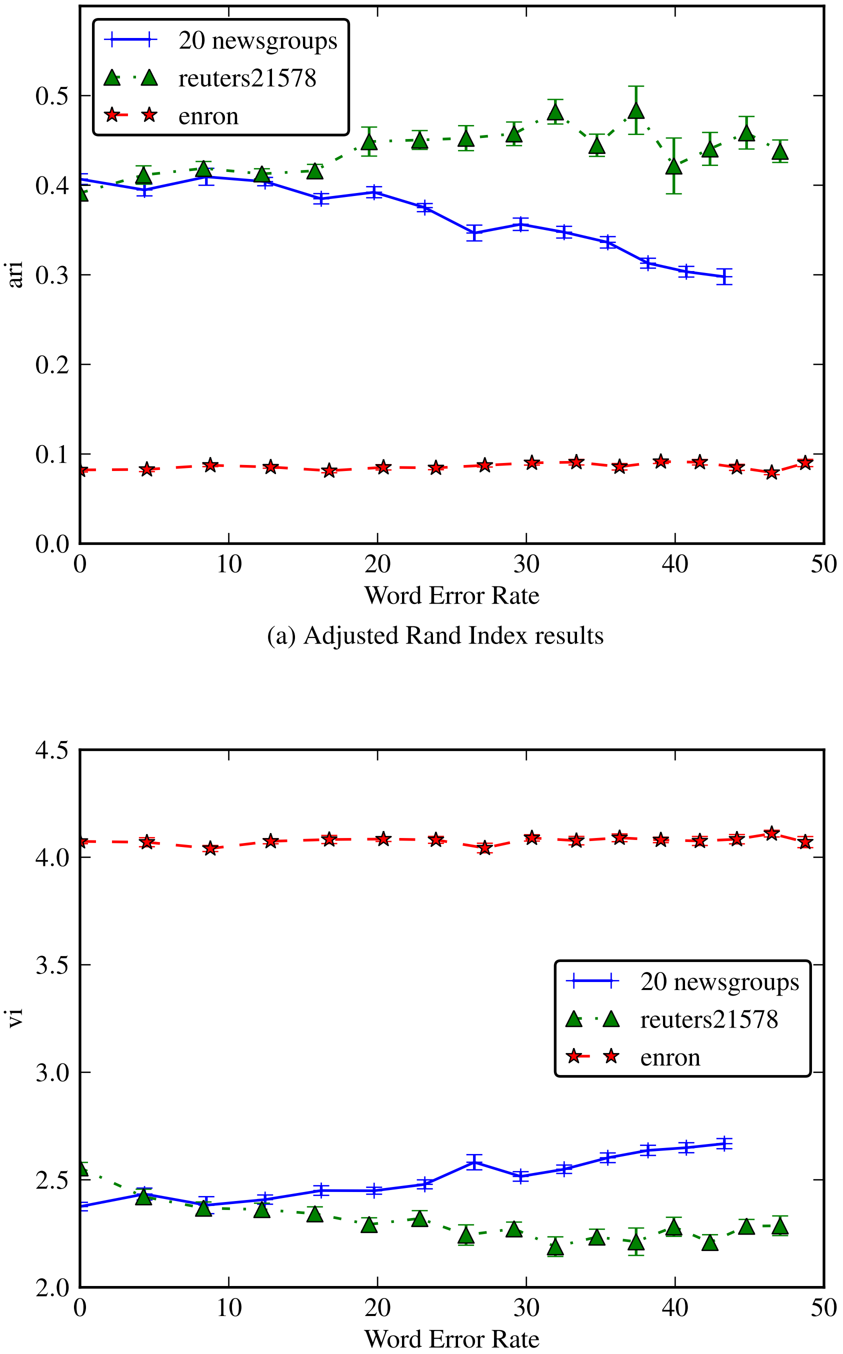
Figure 4.3: The effect of pre-processing on token and type counts for the 20 Newsgroups dataset at various error rates.



(a) Adjusted Rand Index results

(b) Variation of Information results (lower is better)

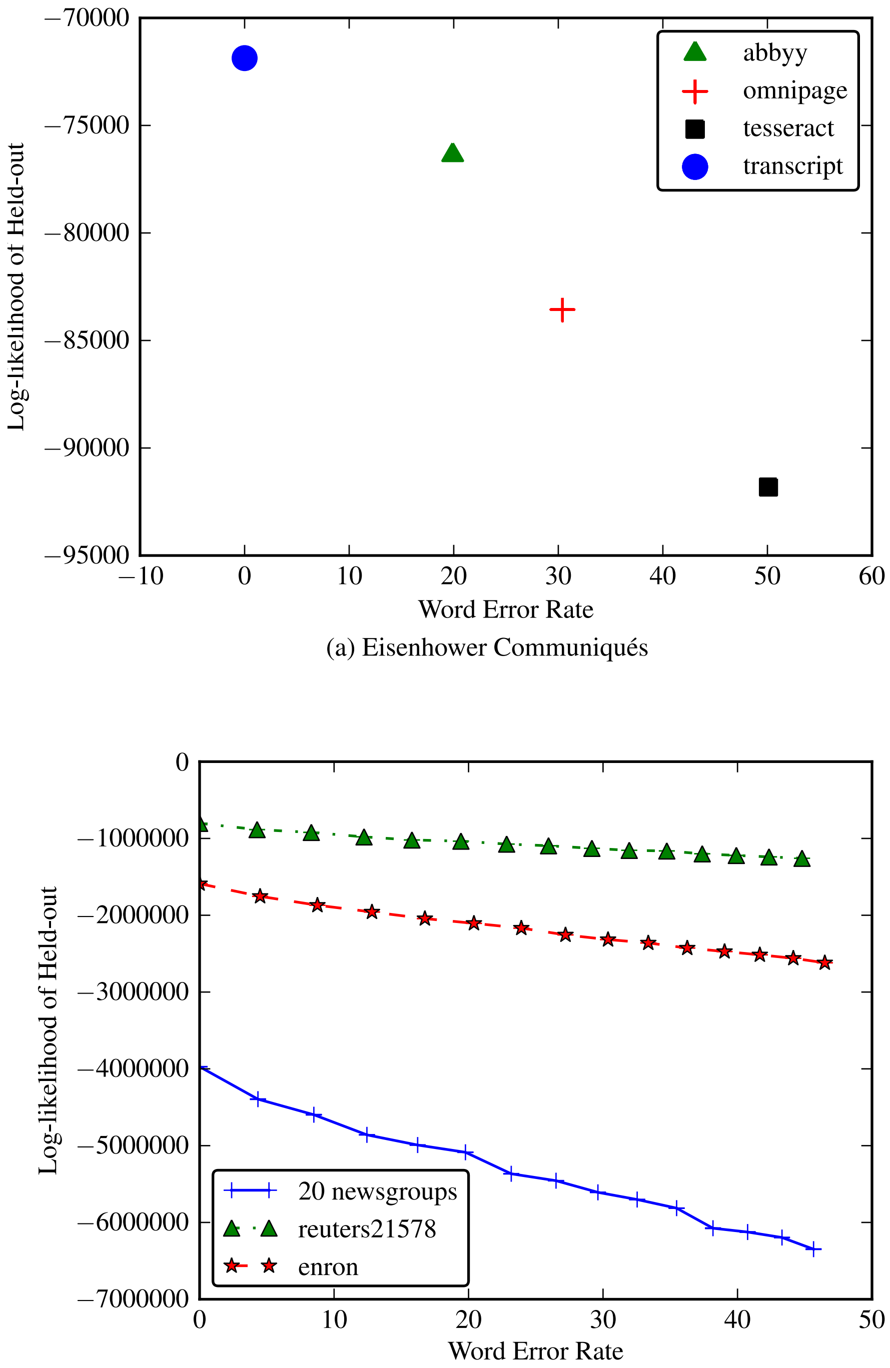
Figure 4.4: Results for the clustering experiments on the three synthetic datasets *without* feature selection.



(a) Adjusted Rand Index results

(b) Variation of Information results (lower is better)

Figure 4.5: Results for the clustering experiments on the three synthetic OCR datasets *with* TNPD feature selection.



(a) Eisenhower Communiqués

(b) Synthetic Data

Figure 4.6: Log-likelihood of heldout data for the LDA experiments.

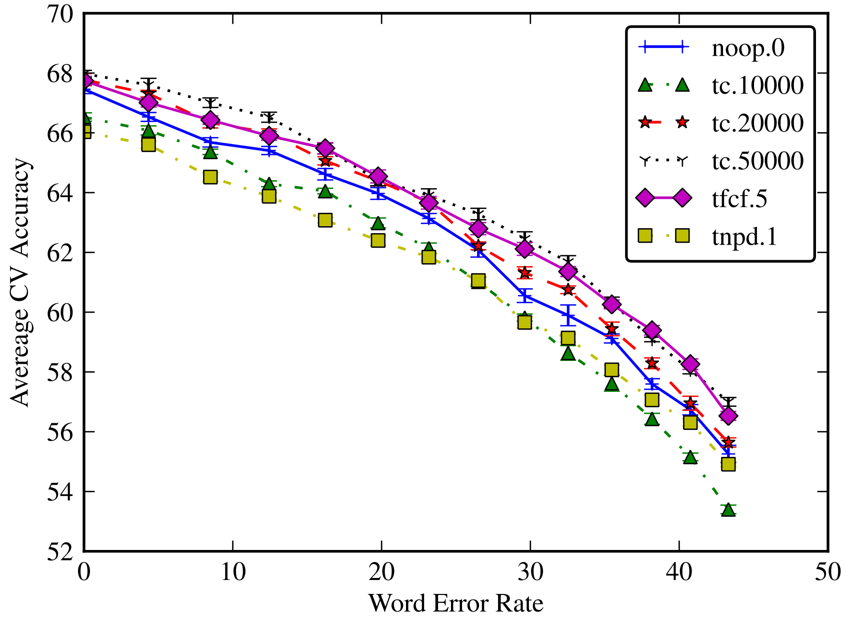


Figure 4.7: Average ten-fold cross-validation accuracy for the LDA pre-processing experiments on the 20 Newsgroups synthetic OCR data.

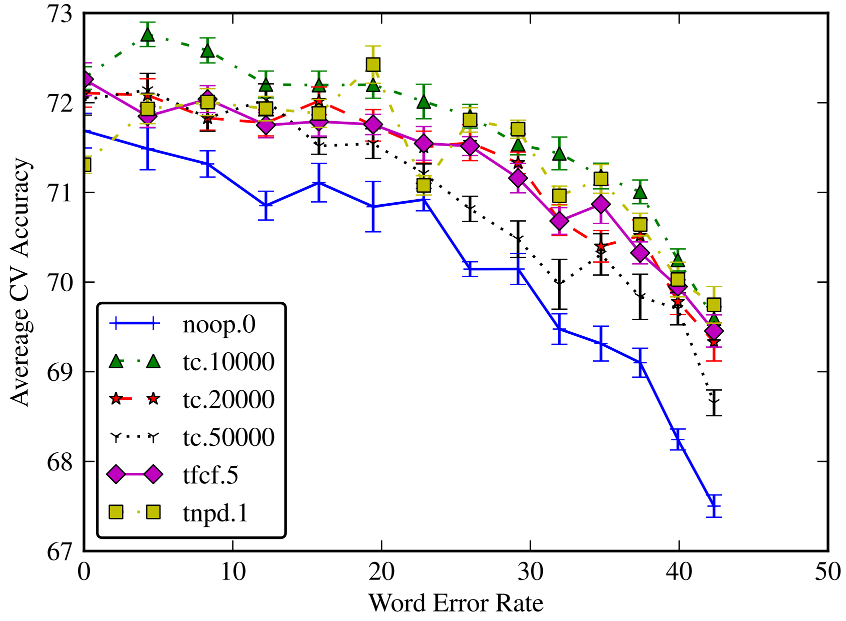


Figure 4.8: Average ten-fold cross-validation accuracy for the LDA pre-processing experiments on the Reuters synthetic OCR data.

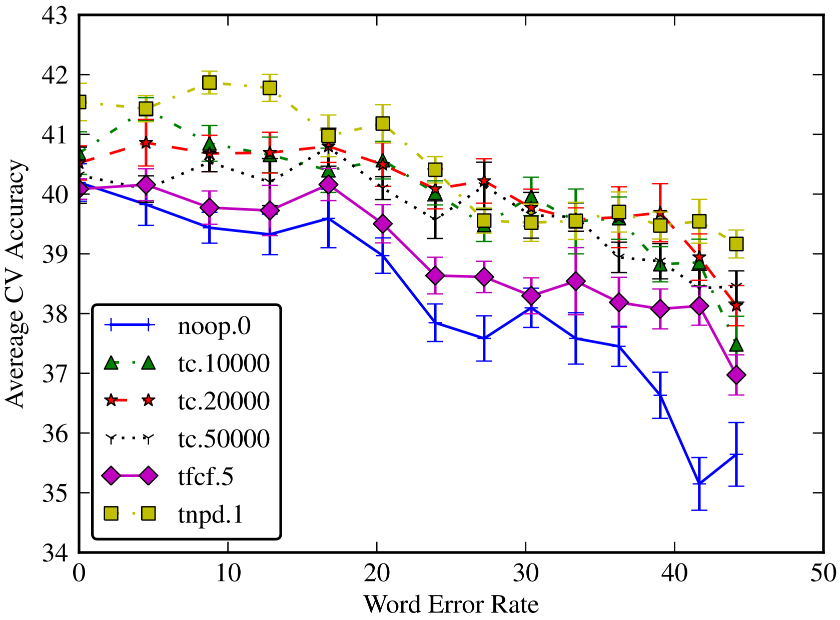


Figure 4.9: Average ten-fold cross-validation accuracy for the LDA pre-processing experiments on the Enron synthetic OCR data.

Chapter 5

A Synthetic Document Image Dataset for Developing and Evaluating Historical Document Processing Methods

Abstract

Document images accompanied by OCR output text and ground truth transcriptions are useful for developing and evaluating document processing methods, especially on historical document images. Certainly many document processing tasks are more difficult on text data produced by OCR. Furthermore, research into improving the performance of such tasks can benefit from further annotation of the data; for example, topical labels on documents can facilitate such research. However, transcribing and labeling historical documents to obtain such data is expensive. As a result, existing real-world document image datasets with such accompanying resources are rare and often relatively small. We introduce document image datasets of varying quality that have been synthetically created from standard (English) text corpora using an existing document degradation model applied in a novel way. Included in the datasets, for convenience, is OCR output from real OCR engines, including a proprietary engine (Abbyy FineReader) and the open-source Tesseract engine. These datasets are designed to exhibit some of the characteristics of an example real-world document image dataset, the Eisenhower Communiqués. The new datasets also benefit from additional metadata that exist due to nature of their collection and prior labeling efforts. In addition, we demonstrate the usefulness of these synthetic datasets by training a multi-engine OCR correction model on the synthetic data and then applying the model to reduce the word error rate on the historical document dataset. This data is to be made available for use by other researchers.

5.1 Introduction

A document image dataset that includes gold-standard transcriptions and optical character recognition (OCR) output can be useful in several types of document processing research. The most obvious work that can benefit from such a dataset is the investigation of improved OCR systems or OCR error correction algorithms. In this case, a gold-standard transcription is required in order to assess the effectiveness of the methods being evaluated by the researcher. If the labeled dataset is similar to real-world datasets then these types of evaluations on transcribed datasets can help practitioners choose the best algorithms to use when OCRing new document images for non-transcribed data.

Other types of research may not be concerned with correcting errors but with the impact that these errors have on performance in text processing and retrieval tasks [56, 57, 93]. Once performance has been assessed, the data may then be used to help build document indexing, summarization, topic modeling or parsing algorithms that are robust to the presence of these errors. In these cases, the researcher would like to be able to evaluate the methods in question against various datasets with varying amounts of noise, in order to assess the degree to which OCR errors degrade performance on the target task.

In all cases, it is desirable to have common datasets that researchers can use as benchmarks in their research. A dataset that includes images, OCR output and gold-standard images together allows researchers to have a common starting point for their methodology. If any of these components is not present, then it becomes difficult to compare results produced by distinct groups. For example, if reference OCR text for the images is not provided, it is difficult to know whether differences in final results arise due to the methods under investigation, or to differences in the OCR engine versions or configurations.

There are a small number of historical document image datasets for which reference OCR text and gold-standard transcriptions exist. These are produced at relatively high cost, and are typically quite small in scale compared to datasets that a practitioner would encounter in real-world scenarios. One example of a historical document dataset is the Eisenhower Communiqués [49, 58], a collection of 610 facsimiles of typewritten documents issued by the Supreme Headquarters Allied

Expeditionary Force (SHAEF) during the last years of World War II. Having been typewritten and duplicated using carbon paper, the quality of the print is poor, making them a challenging case for OCR engines. The communiqués have been OCRed using five OCR engines: ABBYY FineReader for Windows version 10 [1], OmniPage Pro X for Mac OS X [70], Adobe Acrobat Pro for Max OS X [2], ReadIris Pro for Mac OS X [46], and Tesseract version 1.03 [38]. A manual transcription of these documents serves as the gold standard from which it is possible to obtain accurate measures of average document word error rates (WER) for each engine, where a single document WER is defined to be:

$$WER = 100 * \frac{insertions + deletions + substitutions}{total\ tokens\ in\ document}$$

which can exceed 100% when the numbers of insertions are high. See Table 5.1 for the average document WERs for the Eisenhower Communiqués data.

OCR Word Error Rates					Average WER
ABBYY	Omnipage	Adobe	ReadIris	Tesseract	
18.2%	30.0%	51.8%	54.6%	67.8%	44.5%

Table 5.1: Word Error Rates of the five OCR engines used on the Eisenhower Communiqués

Although this dataset has many desirable features, it is small. This means that there is not enough data to build useful training and test sets for many tasks. The text is also very homogeneous in vocabulary and subject matter and is thus not well suited for research involving topic modeling or document clustering on noisy historical documents. Nonetheless, the image quality and OCR error rates are representative of the problem one encounters when working with faccimiles of historical documents and illustrate the need for robust text processing and retrieval techniques when workiing with historical data.

We set out to create a set of synthetic datasets that would be useful in as many of the above mentioned research scenarios as possible. Our goal was to produce a set of datasets that satisfies the following requirements, each inspired by research needs:

1. The datasets should contain the same data at various levels of degradation.
2. The datasets should be reasonably large, containing thousands of documents each.

3. Each document should have a human-provided topical label, to enable research in noisy text analytics.
4. The datasets should be heterogeneous in the amounts and kinds of noise they contain, so as to be consistent with trends observed in real-world historical documents.
5. Most importantly, the errors in the synthetic data should be as like the errors produced by an actual OCR engine on actual degraded documents as possible.

Requirements 1-3 were met by selecting as the source data three datasets commonly used in the document classification and clustering research literature: 20 Newsgroups [51], Reuters 21578 [52], and the LDC-annotated portion of the Enron e-mail archive [11] [1]. Requirements 3 and 4 were met by rendering the digital text documents to images, and then corrupting the images using a parameterizable document degradation model (see Section 5.3) from the literature with stochastically chosen parameters (see Section 5.4), and then OCRing the resulting images using the ABBYY and Tesseract OCR engines.

Our contributions include: 1) the synthetic datasets themselves, which we will be releasing and distributing, 2) the methodology and code used in the creation of the datasets, which will also be released with the data, and 3) the evaluation and verification of the datasets which we conducted in order to verify that it is useful for OCR error correction research.

The remainder of the paper is organized as follows. In Section 5.2 we describe related work to this research. In Section 5.3 we give an overview of Baird's document degradation model, including its parameters and the effect they have on the degraded image. Section 5.4 describes how the document degradation model was used to create datasets with increasing average word error rates consisting of documents with heterogeneous word error rates. Section 5.5 gives relevant statistics for the synthetic datasets, with comparisons to the statistics of the Eisenhower Communiqués. Section 5.6 provides a practical example of how the data can be used for model training in an existing OCR error correction model [59]. Finally, Section 5.7 summarizes the paper and presents our conclusions.

[1] Due to our agreement with the LDC, only the raw corrupted data, and not the topic annotations from the LDC are distributed with our Enron datasets.

5.2 Related Work

Document degradation models have been studied extensively in the literature [8]. Perhaps the best known and well regarded being the work done by Henry Baird and his colleagues [7]. We use a two-parameter version of the model described in Sakar, Baird, and Zhang [82], which will be explained in more detail in Section 5.3.

Other synthetic document image datasets exist as well. In 2008, Daniel Lopresti introduced a dataset that he produced from the Reuters21578 dataset by physically printing the digital documents with a Ricoh Aficio digital photocopier and then scanning these printouts with the same machine [57]. Each document was scanned five times. The first was a scan of the original print of the document, another scan was produced by setting the machine to its darkest contrast setting, and another by scanning that output. This procedure was then repeated on the lightest contrast setting to produce the other two scans. The scanned images were then OCRed using the Tesseract OCR engine.

The Lopresti dataset is a valuable resource for researchers in need of noisy text data and can be used to demonstrate the effect of noise on the performance of NLP algorithms for many tasks [57]. It has the advantage that, although it is synthetic, it makes use of actual physical, optical and mechanical systems in order to degrade documents. However, the dataset does not meet some needs. For example, the dataset consists of 3,305 total pages. However, because each original document is duplicated in the dataset 5 times, there are only 661 unique documents. Consequently, the dataset is too small for realistic research for text analytics and it therefore does not satisfy Requirement 2 from Section 5.1. Also, it turns out that even for the most degraded documents in the dataset—the second-generation copies made with extreme contrast settings—the documents were not sufficiently degraded to produce a wide range in average word error rates to match those found on real-world historical documents. This shortcoming means that the Lopresti dataset does not satisfy our Requirements 1 and 4 either.

5.3 Degradation Model

The model we chose for document degradation is a simplified version [82] of Henry Baird's full optical degradation model [7]. The model is controlled by two parameters, the blur standard deviation b, and the binarization threshold t. The degradation of a document is conducted as follows, given a document that is originally created or sampled at a higher resolution:

1. Translate the image randomly uniformly in x and y between 0 and 1 equivalent output pixels. That is, if the image has been rendered at 5 times the output resolution, then translate between 0 and 5 original pixels. This introduces variation due to pixel alignment variations.
2. Use a Gaussian convolution kernel with standard deviation b to blur the image.
3. Subsample the image to the output resolution.
4. Model image sensor pixel sensitivity error by adding a random value to the intensity at each pixel according to a Gaussian distribution centered at 0.0 with a standard deviation of 0.025.
5. Threshold the image to produce a bi-level output image such that all pixels with intensity less than t become 0.0 (black) and all pixels with an intensity greater than or equal to t become 1.0 (white).

5.4 Methodology

The first step in producing a degraded dataset is rendering the digital text from the source data to images to be used as input for the OCR process. We chose to use the LaTeX document typesetting system for rendering text to images. This choice was motivated by the fact that, with a relatively small amount of cleaning and marking up the source data, LaTeX is able to render text using a complex set of rules that results in high quality text layouts, including ligatures, kerning, and automatic pagination when needed.

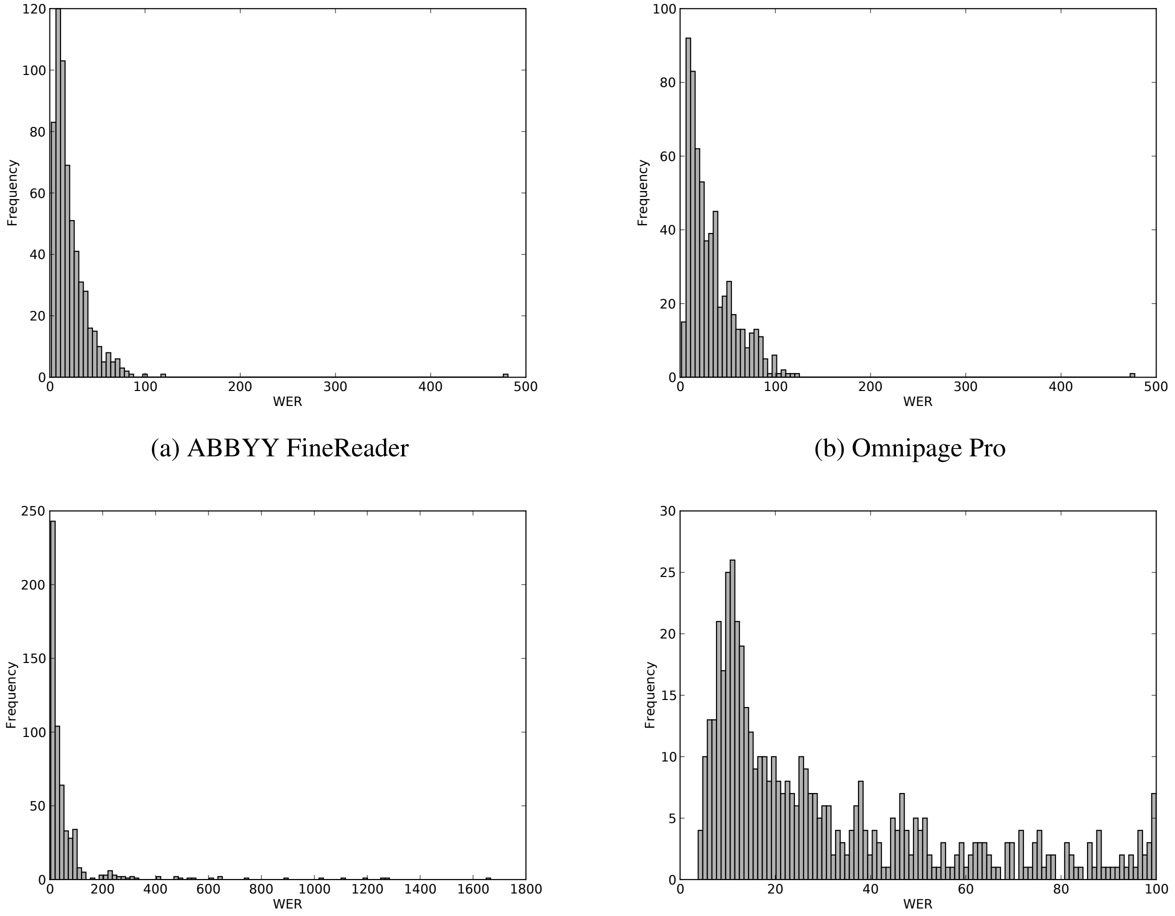
The output from LaTeX is a document in the Adobe Portable Document Format (PDF). In order to obtain an image from this file, we use Ghostscript, a program available on most Linux distributions, to render the PDF to a bitonal Tagged Image Format (TIFF) image at 1500 dpi.

The TIFF images are then passed through the document degradation model to produce the noisy document images.

Creating datasets with one setting for the document degradation model, whether that be a single contrast setting on a copier or a single choice of b and t parameters using the model described in Section 5.3, produces documents with very uniform characteristics. Doing so is at odds with observations taken on the Eisenhower dataset, which reveals that the documents in real-world historical data collections can vary significantly in terms of quality from one to another. For example, some documents may have been reproduced under less than ideal circumstances. Others may have experienced physical damage. Variations can even occur during the original production of the documents. For example, some of the Eisenhower Communiqués seem to have been typed on machines with fading ink ribbons or were typed by inferior typists whose only options for correcting typing mistakes was to type over mistaken characters with the correct ones.

These differences are reflected in the variation in WERs that are observed on the Eisenhower data, as shown in Figure 5.1. It can be seen in 5.1c and 5.1d that even the Tesseract engine, which had an average WER of 50.1% on this data contained many documents that had relatively low WERs. In fact, for the Tesseract engine, only 29% of the documents had a WER greater than 50% and 43% of the documents had WERs less than 20%. This serves to illustrate the high amount of variation in the quality of the OCR output for individual documents in real-world historical document collections.

It would be difficult or impossible to model all of the ways that an historical document can be degraded. However, we decided that, to be true to our real-world data the synthetic OCR documents should consist of documents at various levels of degradation. Thus the distinction between a dataset with low average WER and one with high WER is not that all documents in the high WER dataset have higher WERs. Both datasets consist of a mixture of documents with low, medium, and high amounts of degradation, but the high average WER dataset consists mostly of medium and high WER documents, and the low WER dataset consists mostly of low and medium WER documents. In order to achieve this hetorgeneity, we decided that documents should be degraded with randomized



(a) ABBYY FineReader

(b) Omnipage Pro

(c) Tesseract

(d) Tesseract (truncated)

Figure 5.1: (a)- (c) show histograms of the WERs observed on the Eisenhower data using three OCR engines. (d) shows the Tesseract WERs, ommitting those greater than 100, in order to show more detail in the 0-100% range.

choices for the b and t parameters. We chose to parameterize this process with a single parameter α such that higher values of α would produce datasets with documents that are more degraded on average than documents in datasets with lower α values.

Sakar et al. [82] show the relationship between the t and b parameters and the resulting image quality. In that work, they show how, for a fixed value of t increasing b results in degraded text for which OCR results contain more errors. In addition, for a fixed value of b, there is a “sweet spot" t_s for the t parameter at which documents are degraded in such a way that minimizes OCR error rate, and error rates increase as $|t - t_s|$ increases. As t gets smaller, the lines that form the letter glyphs grow fatter, filling in the gaps in letters such as “a", “o", and “R", and joining letters. As t gets larger, the lines grow thinner, causing single letters to spit, as their thinnest parts are eliminated.

Figure 5.2 shows how word error rates vary over values of the b and t parameters. This graph was produced by first randomly choosing 10 random documents and then sweeping the 10x10 grid of values for b between 1.3 and 2.65 in increments of .13 and values of t between .1 and .4 in increments of .03. The documents were degraded with each pair of b and t in this grid and the degraded documents were OCRed using ABBYY FineReader. For the documents degraded with the same parameter values the resulting word error rates were averaged, and that average word error rate is plotted in the z direction. It can be seen that, in general, as b increases, average WER also increases. In addition, for any given b value, average WER increases as $|.20 - t|$ increases.

As mentioned earlier, the degradation model parameter generation process was controlled with a single parameter α. Using α, we draw the t and b parameters for a document according to the following distributions:

$$b \sim (2.6 - 1.3)\textit{Beta}(\alpha, 1.0) + 1.3$$

$$t \sim \textit{Norm}(.225, ((\alpha/11.0) * (.25 - .001)) + .001)$$

The first distribution is a Beta distribution, scaled and translated to the interval $(1.3, 2.6)$. The second distribution is a Normal distribution with mean .225 and standard deviation that increases linearly from .001 to .25 as α ranges from 0 to 11.0. Figure 5.3 plots (b, t) points drawn from these distributions at four values of α in order to illustrate how the distributions of b and t vary jointly as α changes. As α increases, sampled b values tend to be greater. Also, though the distribution in t values remains centered at .225, the variance increases, increasing the number of points with t values far from the mean.

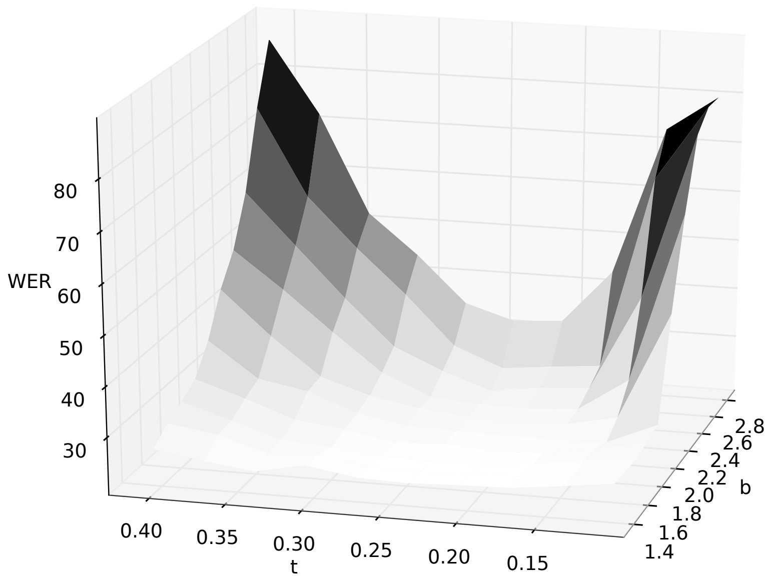


Figure 5.2: The relationship of the t and b parameters on average word error rate of OCRed documents.

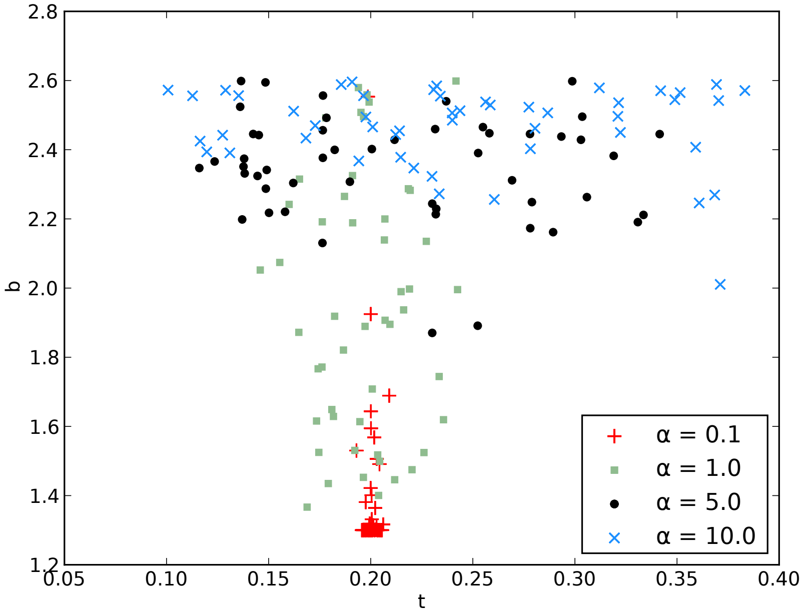


Figure 5.3: Samples of (b, t) parameter pairs at four different α values. Fifty points were drawn for each α value.

Using this process, for each source dataset and each α value in $0.1, 1.0, 2.0, \ldots, 10.0$ we drew a random parametrization (b, t) for each document based on the current α and then degraded that document. This resulted in 33 datasets, one for each (α, source dataset) pair.

5.5 Statistics

The procedure discussed above produced 452,726 degraded image pages: 239,326 for 20 Newsgroups, 123,748 for Reuters and 89,652 for Enron. Figure 5.4 shows how the ABBYY FineReader OCR engine performed on these images. This outcome demonstrates that we have created datasets with varying levels of noise, as measured by the average word error rates of the constituent documents. This outcome also verifies that our synthetic data satisfy Requirement 1 from Section 5.1. Figure 5.5 shows histograms of word error rates for the 20 Newsgroups dataset, degraded using four different values of α. Although the histograms appear to be much more regular than the Eisenhower histograms, they do exhibit the desired trend that many documents have low error rates, a noticeable difference being that there is a sharp drop-off at 100%, which does not occur in the Eisenhower data. Despite these small differences, these graphs give some evidence that we have also met Requirements 4 and 5. Further evidence to this effect will be given in the form of an OCR correction task in Section 5.6.

5.6 OCR Error Correction Task

In order to further validate the utility of the synthetic datasets, we used the data to train an existing approach to multiple-engine OCR error correction, as described by Lund et al. [59]. The method is motivated by the reasoning that OCR engines have different strengths and weaknesses. Thus, if one OCR engine outputs an incorrect hypothesis for a word token in the source image, another engine might output the correct hypothesis. The problem then becomes, given the output from multiple OCR engines based on the same source image, to choose between the hypotheses at each word in such a way so as to minimize the word error rate of the final output.

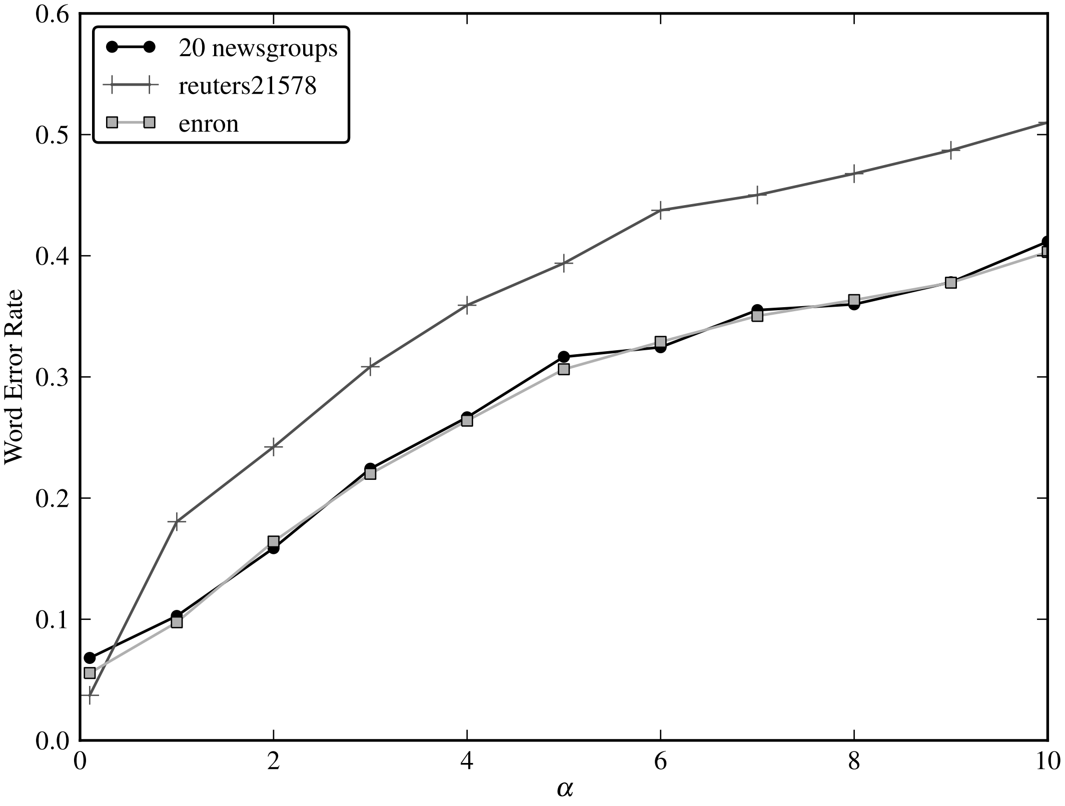
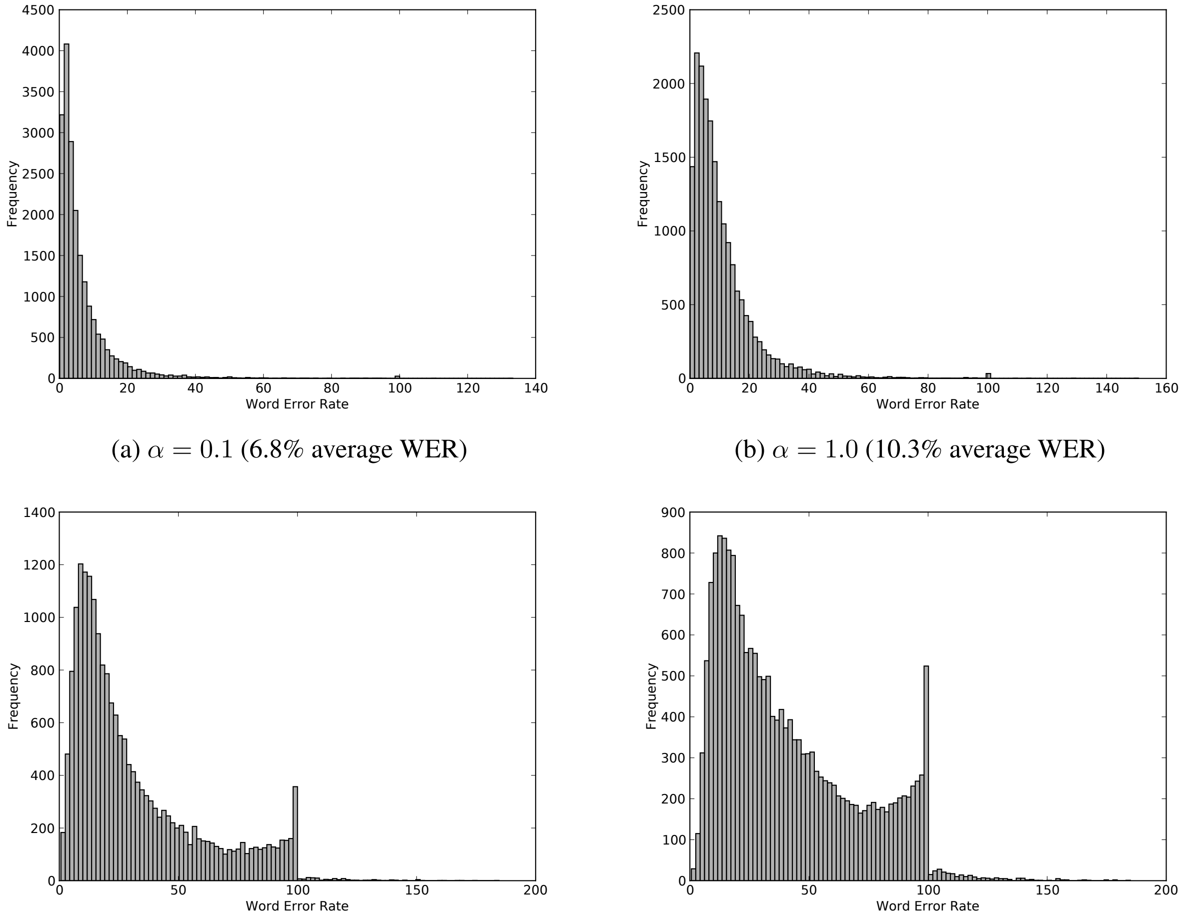


Figure 5.4: Correlation between α corruption levels and the resulting average word error rates from ABBYY FineReader.



(a) $\alpha = 0.1$ (6.8% average WER)

(b) $\alpha = 1.0$ (10.3% average WER)

(c) $\alpha = 5.0$ (31.7% average WER)

(d) $\alpha = 10.0$ (41.2% average WER)

Figure 5.5: Histograms of the word error rates of ABBYY FineReader on the 20 Newsgroups synthetic datasets at four α levels. The x-axis has been truncated to the (0,200) range.

Engine	Aligned Column Hypothesis
ABBYY	Precipitation
Tesseract	Precipitation:

Figure 5.6: An aligned column of the OCR outputs from the two OCR engines from the Eisenhower Communiqués. Note that the Tesseract engine has erroneously proposed an extra colon at the end of the word.

We employ a Maximum Entropy multi-class classifier [69] as implemented in the MALLET toolkit [61], trained on a subset of the synthetic data introduced here, consisting of 785 documents from the Enron data at three different α levels: 1.0, 5.0, and 9.0. We call these three datasets the *calibration sets*. We prepared training data from the synthetic document image datasets by first aligning the output of the OCR engines in order to produce aligned columns where each column roughly corresponds to hypotheses for the same word in the source image. We then extracted the following features from each column:

Voting: indicates when multiple hypotheses in a column match exactly,

Number: binary indicators for whether each hypothesis is a cardinal number,

Dictionary: binary indicators for whether each hypothesis appears in the Linux dictionary,

Gazetteer: binary indicators for whether each hypothesis appears in a gazetteer of place names, and

Spell Checker: an additional hypothesis generated by Aspell [50] from words that do not appear in the dictionary or in the gazetteer.

For each training case (an aligned column), the label indicates which OCR engine provided the correct hypothesis. Ties were resolved by selecting the output from the OCR engine with the lowest overall WER on the training data (ABBYY FineReader). Figure 5.6 shows an example column of proposals from the Enron, $\alpha = 5.0$ calibration dataset.

We trained individually on the three calibration sets, and used the resulting classifiers to choose the transcription hypotheses for the Eisenhower data, given the output for that dataset from the two OCR engines, for which OCR output has been included in the datasets. The results are shown in Table 5.2. Recall from Table 5.1 that the best single OCR engine (ABBYY) achieved an 18.2% WER. The improvements shown in rows 1 and 2 indicate that these datasets are useful for training the correction model in a way that improves accuracy on the Eisenhower data. This result is significant because it tells us that this synthetic data can be used to train models for improving OCR output on un-transcribed historical document images. If the synthetic data consisted of errors that were unlike those seen in real-world data, then the patterns learned during training on the calibration sets would not have generalized to the Eisenhower data, precluding this level of improvement. We believe this is additional evidence that the synthetic datasets are sufficiently like real-world historical OCR data to be useful for OCR research. (Our previously published experiments showed the value of this multi-engine error correction method, reducing word error rates to 13.29% – a 26% relative reduction in WER – by adding the output from three additional OCR engines [59].)

Callibration Set (α) Used for Training	Resulting Average WER on the Eisenhower dataset
1.0	16.73%
5.0	15.56%
9.0	22.87%

Table 5.2: Average word error rates of the final output of multi-engine error correction algorithm trained on the Enron calibration sets Eisenhower Communiqués

5.7 Conclusions

We have introduced a set of synthetic datasets of degraded document images together with gold standard text and the output from two OCR engines. The data have nice statistical properties in terms of the word error rates observed with the OCR engines. In addition, we have presented evidence that the data are sufficiently like real-world OCR data.

Though the document degradation model we used has been introduced before, our method for stochastically choosing parameters per image in order to simulate differences observed in real historical data is novel, and the methodology could be applied to other source data.

For instructions on how to download the synthetic datasets, the code used to produce it, the Eisenhower Communiqués, and document image samples please visit:

`https://facwiki.cs.byu.edu/nlp/index.php/Synthetic_OCR_Data`.

Chapter 6

Topics Over Nonparametric Time: A Supervised Topic Model Using Bayesian Nonparametric Density Estimation

Abstract

We present a new supervised topic model that uses a nonparametric density estimator to model the distribution of real-valued metadata given a topic. The model is similar to Topics Over Time, but replaces the beta distributions used in that model with a Dirichlet process mixture of normals. The use of a nonparametric density estimator allows for the fitting of a greater class of metadata densities. We compare our model with existing supervised topic models in terms of prediction and show that it is capable of discovering complex metadata distributions in both synthetic and real data.

6.1 Introduction

Supervised topic models are a class of topic models that, in addition to modeling documents as mixtures of topics, each with a distribution over words, also model metadata associated with each document. Document collections often include such metadata. For example, timestamps are commonly associated with documents that represent the time of the document's creation. In the case of online product reviews, “star" ratings frequently accompany written reviews to quantify the sentiment of the review's author.

There are three basic reasons that make supervised topic models attractive tools for use with document collections that include metadata. *Better Topics*: one assumption that is often true for document collections is that the topics being discussed are correlated with information that is not necessarily directly encoded in the text. Using the metadata in the inference of topics provides an extra source of information, which could lead to an improvement in modeling the topics that are found. *Prediction*: given a trained supervised topic model and a new document with missing metadata, one can predict the value of the metadata variable for that document. Even though timestamps are typically included in modern, natively digital, documents they may be unavailable or wrong for historical documents that have been digitized using OCR. Also, even relatively modern documents can have missing or incorrect timestamps due to user error or system mis-configuration. For example, in the full Enron e-mail corpus[1], there are 793 email messages with a timestamp before 1985, the year Enron was founded. Of these messages 271 have a timestamp before the year 100. *Analysis*: in order to understand a document collection better, it is often helpful to understand how the metadata and topics are related. For example, one might want to analyze the development of a topic over time, or investigate what the presence of a particular topic means in terms of the sentiment being expressed by the author. One may, for example, plot the distribution of the metadata given a topic from a trained model.

Several supervised topic models can be found in the literature and will be discussed in more detail in Section 6.3. These models make assumptions about the way in which the metadata

[1] `http://www.cs.cmu.edu/~enron`

are distributed given the topic or require the user to specify their own assumptions. Usually, this approach involves using a unimodal distribution, and the same distribution family is used to model the metadata across all topics. These modeling assumptions are problematic. First, it is easy to imagine metadata and topics that have complex, multi-modal relationships. For example, the U.S. has been involved in two large conflicts with Iraq over the last 20 years. A good topic model trained on news text for that period should ideally discover an Iraq topic and successfully capture the bimodal distribution of that topic in time. Existing supervised topic models, however, will either group both modes into a single mode, or split the two modes into two separate topics. Second, it seems incorrect to assume that the metadata will be distributed similarly across all topics. Some topics may remain fairly uniform over a long period of time, others appear quickly and then fade out over long periods of time (e.g., terrorism after 9/11), others enter the discourse gradually over time (e.g., healthcare reform), still others appear and disappear in a relatively short period of time (e.g., many political scandals).

To address these issues, we introduce a new supervised topic model, Topics Over Nonparametric Time (TONPT), based on the Topics Over Time (TOT) model [99]. Where TOT uses a per-topic beta distribution to model topic-conditional metadata distributions, TONPT uses a nonparametric density estimator, a Dirichlet process mixture (DPM) of normals.

The remainder of the paper is organized as follows: in Section 6.2 we provide a brief discussion of the Dirichlet process and show how a DPM of normals can be used to approximate a wide variety of densities. Section 6.3 outlines related work. In Section 6.4 we introduce the TONPT model and describe the collapsed Gibbs sampler we used to efficiently conduct inference in the model on a given dataset. Section 6.5 describes experiments that were run in order to compare TONPT with two other supervised topic models and a baseline. Finally, in Section 6.6 we summarize our results and contributions.

6.2 Estimating Densities with Dirichlet Process Mixtures

Significant work has been done in the document modeling community to make use of Dirichlet process (DP) mixtures with the goal of eliminating the need to specify the number of components in a mixture model. For example, it is possible to cluster documents without specifying a-priori the number of clusters by replacing the Dirichlet-multinomial mixing distribution in the Mixture of Multinomials document model with a Chinese Restaurant Process. The CRP is the distribution over partitions created by the clustering effect of the Dirichlet process [4]. So, one way of using the Dirichlet process is in model-based clustering applications where it is desirable to let the number of clusters be determined dynamically by the data, instead of being specified by the user.

The DP is a distribution over probability measures G with two parameters: a base measure G_0 and a total mass parameter m. Random probability measures drawn from a DP are generally not suitable as likelihoods for continuous random variates because they are discrete. This complication can be overcome by convolving the G with a continuous kernel density f [30, 31, 55]:

$$\begin{aligned} G &\sim \mathit{DP}(m, G_0) \\ x_i|G &\sim \int f(x_i|\theta)dG(\theta) \end{aligned}$$

This model is equivalent to an infinite mixture of f distributions with hierarchical formulation:

$$\begin{aligned} G &\sim \mathit{DP}(m, G_0) \\ \theta_i|G &\sim G \\ x_i|\theta_i &\sim f(x_i|\theta) \end{aligned}$$

In our work we use the normal distribution for f. The normal distribution has many advantages that make it a useful choice here. First, the parameters map intuitively to the idea that the θ parameters in the DPM are the “locations" of the point masses of G and so are a natural fit for the mean parameter of the normal distribution. Second, because the normal is conjugate to the mean of a

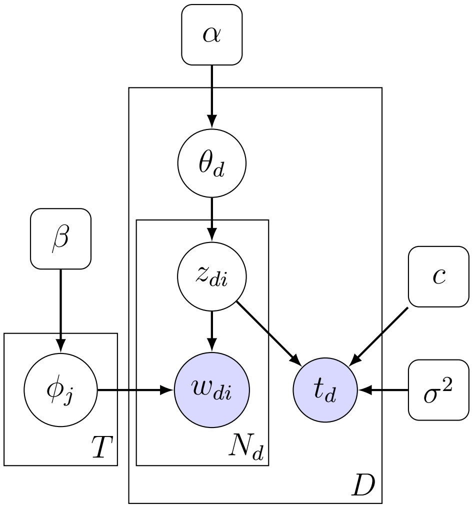


Figure 6.1: The Supervised LDA model.

normal with known variance, we can also choose a conjugate G_0 that has intuitive parameters and simple posterior and marginal forms. Third, the normal is almost trivially extensible to multivariate cases. Fourth, the normal can be centered anywhere on the positive or negative side of the origin which is not true, for example, of the gamma and beta distributions. Finally, just as any 1-D signal can be approximated with a sum of sine waves, almost any probability distribution can be approximated with a weighted sum of normal densities. Hence, the DP mixture of normals is an excellent candidate for modeling arbitrary densities of metadata.

6.3 Related Work

In this section we will describe the three models which are most closely related to our work. In particular, we focus on the issues of prediction and the posterior analysis of metadata distributions in order to highlight the strengths and weaknesses of each model.

The most closely related models to TONPT are Supervised LDA (sLDA) [15] and Topics Over Time [99]. sLDA uses a generalized linear model (GLM) to regress the metadata given the topic proportions of each document. GLMs are flexible in that they allow for the specification of a link and a dispersion function that can change the behavior of the regression model. In practice, however, making such a change to the model requires non-trivial modifications to the inference procedure used to learn the topics and regression co-efficients. In the original sLDA paper, an identity link function and normal dispersion distribution were used. The model, shown in Figure 6.1, has per-document timestamp variables $t_d \sim \mathit{Normal}(c \cdot \overline{z_d}, \sigma^2)$, where c is the vector of linear model coefficients and $\overline{z_d}$ is a topic proportion vector for document d (See Table 6.1 for a description of the other variables in the models shown here). This configuration leads to a stochastic EM inference procedure in which one alternately samples from the complete conditional for each topic assignment, given the current values of all the other variables, and then finds the regression coefficients that minimize the sum squared residual of the linear prediction model. Variations of sLDA have been used successfully in several applications including modeling the voting patterns of U.S. legislators [35] and links between documents [20].

Prediction in sLDA is very straightforward, as the latent metadata variable for a document can be marginalized out to produce the LDA complete conditional distribution for the topic assignments. The procedure for prediction can thus be as simple as first sampling the topic assignments for each word in an unseen document given the assignments in the training set, and then taking the dot product between the estimated topic proportions for the document and the GLM coefficients. In terms of the representation of the distribution of metadata given topics, however, the model is somewhat lacking. The coefficients learned during inference convey only one-dimensional information about the correlation between topics and the metadata. A large positive coefficient for a given topic indicates that documents with a higher proportion of that topic tend to have higher metadata values, and a large negative coefficient means that documents with a higher proportion of that topic tend to have lower metadata values. A coefficient close to zero indicates a low correlation between the corresponding topic and the metadata.

In TOT, metadata are treated as per-word observations, instead of as a single per-document observation. The model, shown in Figure 6.2, assumes that each per-word metadatum t_{di} is drawn from a per-topic beta distribution: $t_{di} \sim \mathit{Beta}(\psi_{z_{di}1}, \psi_{z_{di}2})$. The inference procedure for TOT is a stochastic EM algorithm, where the topic assignments for each word are first sampled with a collapsed Gibbs sampler and then the shape parameters for the per-topic beta distributions are point

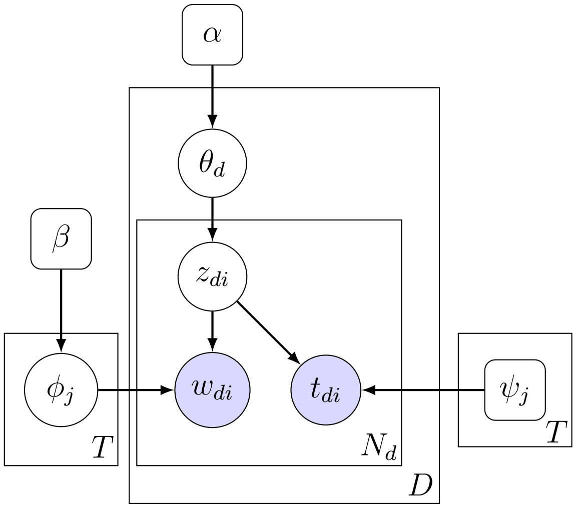


Figure 6.2: The Topics Over Time model.

estimated using the Method of Moments based on the mean and variance of the metadata values for the words assigned to each topic.

Prediction in TOT is not as straightforward as for sLDA. Like sLDA, it is possible to integrate out the random variables directly related to the metadata and estimate a topic distribution for a held-out document using vanilla LDA inference. However, because the model does not include a document-level metadata variable, there is no obvious way to predict a single metadata value for held-out documents. We describe a prediction procedure in Section 6.5, based on work by Wang and McCallum, that yields acceptable results in practice.

Despite having a more complicated prediction procedure, TOT yields a much richer picture of the trends present in the data. It is possible with TOT, for example, to get an idea of not only whether the metadata are correlated with a topic, but also to see the mean and variance of the per-topic metadata distributions and even to show whether the distribution is skewed or symmetric.

Another related model is the Dirichlet Multinomial Regression (DMR) model [65]. Whereas the sLDA and TOT models both model the metadata generatively, i.e., as random variables conditioned on the topic assignments for a document, the DMR forgoes modeling the metadata explicitly, putting the metadata variables at the “root" of the graphical model and conditioning the document distributions over topics on the metadata values. By forgoing a direct modeling of the metadata,

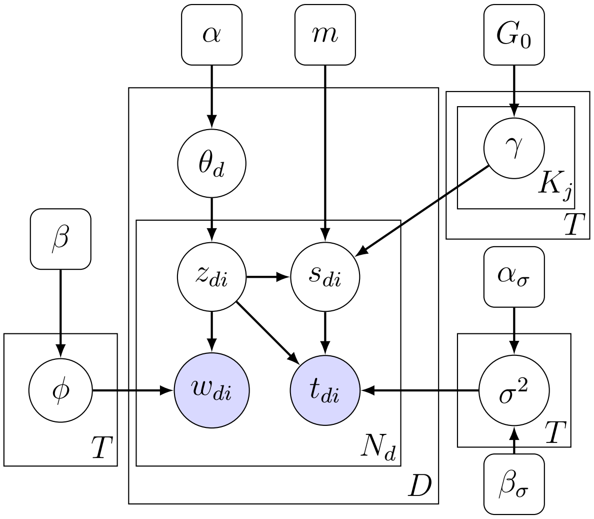


Figure 6.3: TONPT as used in sampling.

the DMR is able to take advantage of a wide range of metadata types and even to include multiple metadata measurements (or “features") per document. The authors show how, conditioning on the metadata, the DMR is able to outperform other supervised topic models in terms of its ability to fit the observed words of held-out documents, yielding lower perplexity values. The DMR is thus able to accomplish one of the goals of supervised topic modeling very well (the increase in topic quality). However, because it does not propose any distribution over metadata values, it is difficult to conduct the types of analyses or missing metadata predictions possible in TOT and sLDA without resorting to ad-hoc procedures. Because of these deficiencies, we leave the DMR out of the remaining discussion of supervised topic models.

6.4 Topics Over Nonparametric Time

TONPT models metadata variables associated with each word in the corpus as being drawn from a topic-specific Dirichlet process mixture of normals. In addition, TONPT employs a common base measure G_0 for all of the per-topic DPMs, for which we use a normal with mean μ_0 and variance σ_0^2.

Symbol	**Meaning**
Common Supervised Topic Modeling Variables	
α	Prior parameter for document-topic distributions
θ_d	Parameter for topic mixing distribution for document d
β	Prior parameter for the topic-word distributions
ϕ_j	Parameter for the jth topic-word distribution
z_{di}	Topic label for word i in document d
$\mathbf{z}_{-di}$	All topic assignments except that for z_{di}
$\mathbf{w}$	Vector of all word token types
w_{di}	Type of word token i in document d
t_{di}	Timestamp for word i in document d
t_d	Timestamp for document d
$\mathbf{t}$	Vector of all metadata variable values
$\hat{t}$	A predicted value for the metadata variable
D	The number of documents
T	The number of topics
V	The number of word types
N_d	The number of tokens in document d
TONPT Specific Variables	
m	Total mass parameter for DP mixtures
s_{di}	DP component membership for word i in document d
$\mathbf{s}_{-di}$	All DP component assignments except that for s_{di}
G_0	The base measure of the DP mixtures
μ_0	The mean of the base measure
σ_0^2	The variance of the base measure
γ_{jk}	The mean of the kth mixture component for topic j
$\boldsymbol{\gamma}$	A vector of all the γ values
$\boldsymbol{\gamma}_{-jk}$	$\boldsymbol{\gamma}$ without γ_{jk}
σ_j^2	The variance of the components of the jth DP mixture
$\boldsymbol{\sigma}^2$	A vector of all the DPM σ^2s
α_σ, β_σ	Shape and scale parameters for prior on topic σs
K_j	The number of unique observed γs for topic j
n_j	The number of tokens assigned to topic j
n_{jk}	The number of tokens assigned to the kth component of topic j
n_{dj}	The number of tokens in document d assigned to topic j
n_{jv}	The number of tokens of type v assigned to topic j
$K_{z_{di}}^{<di}$	The number of unique γs observed for topic z_{di} before the ith token of document d
$\tau^{(jk)}$	The set of all t_{di} s.t. $z_{di} = j$ and $s_{di} = k$
$f(y; \mu, \sigma^2)$	The normal p.d.f. at y with mean μ and variance σ^2

Table 6.1: Mathematical symbols used in the models and derivations of this chapter. The “common" symbols are shared by TONPT, sLDA, and TOT.

The random variables are distributed as follows:

$$
\begin{aligned}
\theta_d|\alpha &\sim \textit{Dirichlet}(\alpha) \\
\phi_t|\beta &\sim \textit{Dirichlet}(\beta) \\
z_{di}|\theta_d &\sim \textit{Categorical}(\theta_d) \\
w_{di}|z_{di}, \phi &\sim \textit{Categorical}(\phi_{z_{di}}) \\
\sigma_j^2|\alpha_\sigma, \beta_\sigma &\sim \textit{InverseGamma}(\alpha_\sigma, \beta_\sigma) \\
G_j|G_0, m &\sim \textit{DP}(G_0, m) \\
t_{di}|G_{z_{di}}, \sigma_{z_{di}}^2 &\sim \int f(t_{di}; \gamma, \sigma_{z_{di}}^2) dG_{z_{di}}(\gamma)
\end{aligned}
$$

where $f(\cdot; \gamma, \sigma^2)$ is the normal p.d.f. with mean γ and variance σ^2. Also, $j \in \{1, \ldots, T\}$, $d \in \{1, \ldots, D\}$, and, given a value for d, $i \in \{1, \ldots, N_d\}$. We note that, as in TOT, the fact that the metadata variable is repeated per-word leads to a deficient generative model, because the metadata are typically observed at a document level and the assumed constraint that all of the metadata values for the words in a document be equivalent is not modeled. The advantage of this approach is that this configuration simplifies inference and also naturally balances the plurality of the word variables with the singularity of the metadata variable, allowing the metadata to exert a similarly scaled influence on the topic assignments during inference. That is, by having the same number of variables in the model to represent the metadata as there are representing the topic assignments, the various products in the complete conditionals, posteriors and likelihoods are affected as much by the metadata variables as by the topic assignment variables, whereas a single metadata variable per document would mean that the metadata distributions would be represented by only a single term in these products and the topic assignments by many more leading to an imbalance in the model. In addition, this modeling choice allows for a more fine-grained labeling of documents (e.g., at the word, phrase, or paragraph level) and for finer grained prediction. For example, individual words in a document are not all written simultaneously and were most likely written at some time before the document timestamp (depending on whether the timestamp represents the creation of the document or the time of the last edit). Also, in terms of sentiment, there are often positive comments even in very negative reviews.

This model does not lend itself well to inference and sampling because of the integral in the distribution over t_{di}. A typical modification that is made to facilitate sampling in mixture models is to use an equivalent hierarchical model. Another modification that is typically made when sampling in mixture models is to separate the “clustering," or mixing, portion of the distribution from the prior over mixture component parameters. The mixing distribution in a DPM is the distribution known as the Chinese Restaurant Process. The Chinese Restaurant Process is used to select an assignment to one of the points that makes up the DP point process for each data observation drawn from G. The locations of these points are independently drawn from G_0.

Figure 6.3 shows the model that results from decomposing the Dirichlet process into these two component pieces. The K_j unique γ values that have been sampled so far for each topic j are drawn from G_0. The s_{di} variables are indicator variables that take on values in $1, \ldots, K_j$ and represent which of the DPM components each t_{di} is drawn from. This model has the following changes to the variable distributions:

$$\begin{aligned}
s_{di}|z_{di}, \mathbf{s}^{<di}, m &\sim \begin{cases} = k \text{ with prob } \propto n_{z_{di},k}^{<di} \text{ for } k = 1, \ldots, K_{z_{di}}^{<di} \\ = K_{z_{di}}^{<di} + 1 \text{ with prob } \propto m \end{cases} \\
\gamma_{jk}|G_0 &\sim G_0 \\
t_{di}|z_{di}, s_{z_{di}}, \boldsymbol{\gamma}, \boldsymbol{\sigma}^2 &\sim f\left(t_{di}; \gamma_{z_{di}s_{di}}, \sigma_{z_{di}}^2\right)
\end{aligned}$$

Where $\mathbf{s}^{<di}$ refers to all the $s_{d'i'}$ that came “before" s_{di} and before is defined to mean all (d', i') such that $(d' < d)$ or $(d' = d$ and $i' < i)$. Likewise, $n_{z_{di},k}^{<di}$ is the count of the number of times that $s_{d'i'} = k$ for all d', i' before s_{di} and $K_{z_{di}}^{<di}$ is the highest value of any $s_{d'i'}$ (number of unique observed γs) before s_{di}. So, conditioned on z_{di} the s_{di} are distributed according to a Chinese Restaurant Process with mass parameter m.

The θ and ϕ variables in the model are nuisance variables: they are not necessary for the assignment of tokens to topics or for the estimation of the distributions of the response variables so, as is typical when conducting Gibbs sampling on these models, we integrate them out before sampling.

6.4.1 Gibbs Sampler Conditionals

Now we derive the complete conditionals for the collapsed Gibbs sampler used for inference in the model. There are four groups of variables that must be sampled during inference: the per-word topic labels z, the per word DPM component assignment variables s, the DPM component means $\boldsymbol{\gamma}$, and the per-topic DPM component variances $\boldsymbol{\sigma}^2$. Note that, because of normal-normal conjugacy, it would be possible to collapse the γ variables from the model. We choose to sample values for γ anyway because the parameters of the DPM are useful artifacts in their own right, as they enable rich posterior analyses of the per-topic metadata distributions.

Complete Conditional for z and s

We choose to sample z_{di} and s_{di} in a block, since the calculations necessary to sample z_{di} include those sufficient to sample both variables jointly.

$$
\begin{aligned}
[z_{di}, s_{di}] &= p(z_{di} = j, s_{di} = k | \mathbf{z}_{-di}, \mathbf{s}_{-di}, \boldsymbol{\sigma}^2, \mathbf{w}, \mathbf{t}, m, \boldsymbol{\gamma}, \alpha_\sigma, \beta_\sigma, G_0, \alpha, \beta) \\
&\propto \alpha_{\star dj} \frac{\beta_{\star j w_{di}}}{\sum_{v=1}^{V} \beta_{\star jv}} \cdot \begin{cases} \frac{n_{jk}}{n_j+m} f(t_{di}; \gamma_{jk}, \sigma_j^2) \text{ if } k \leq |\gamma_j|, \\ \frac{m}{n_j+m} f(t_{di}; \mu_0, \sigma_0^2 + \sigma_j^2) \text{ if } k = |\gamma_j| + 1 \end{cases}
\end{aligned}
$$

where $\alpha_{\star dj} = \alpha_j + n_{dj}$, $\beta_{\star jv} = \beta_v + n_{jv}$.

Complete Conditional for γ

When sampling a z_{di}, s_{di} pair if $s_{di} = K_{z_{di}} + 1$ (i.e., we are creating a new component for the DPM for that topic), then we need to draw a new γ for the z_{di}th DPM. Also, each γ_{jk} needs to be resampled each iteration of the Gibbs sampler.

Let $\tau^{(jk)} = \{t_{di} : z_{di} = j \text{ and } s_{di} = k\}$ ordered arbitrarily, which groups the t_{di} by the topic and DPM component that they are associated with. The complete conditional for each γ is:

$$\begin{aligned}[\gamma_{jk}] &= p(\gamma_{jk}|\mathbf{s}, \mathbf{t}, \mathbf{w}, \mathbf{z}, \boldsymbol{\gamma}_{-jk}, \boldsymbol{\sigma}^2, \alpha_\sigma, \beta_\sigma, m, \alpha, \beta, \mu_0, \sigma_0^2) \\ &= f\left(\gamma_{jk}; \mu_{jk\star}, \sigma_{jk\star}^2\right) \qquad (6.1)\end{aligned}$$

where $\sigma_\star^2 = \left(\frac{1}{\sigma_0^2} + \frac{|\tau^{(jk)}|}{\sigma_j^2}\right)^{-1}$ and $\mu_\star = \left(\frac{\mu_0}{\sigma_0^2} + \frac{\sum_{i=1}^{|\tau^{(jk)}|} \tau_i^{(jk)}}{\sigma_j^2}\right) \cdot \sigma_\star^2$

Complete Conditional for σ^2

The complete conditional is a common result for gamma-normal conjugacy. In this case, the likelihood is restricted to those t_{di} for which $z_{di} = j$:

$$\left[\sigma_j^2\right] = \mathit{InverseGamma}\left(\alpha_{\sigma_\star}, \beta_{\sigma_\star}\right) \qquad (6.2)$$

where $\alpha_{\sigma_\star} = \alpha_\sigma + \frac{n_j}{2}$,

$$\beta_{\sigma_\star} = \beta_\sigma + \frac{\sum_{d=1}^{D} \sum_{i=1}^{N_d} [\mathbf{1}_j(z_{di})(\gamma_{z_{di}, s_{di}} - t_{di})^2]}{2},$$

and $\mathbf{1}_j(x)$ is the Kronecker delta.

6.5 Experiments

We inferred topic assignments and metadata distributions for several real-world datasets using sLDA, TOT, TONPT, and two baseline methods that we will refer to as PostHocLinear and PostHocTONPT, in which a vanilla LDA model is inferred over the dataset and then either a linear model is fit to the metadata using the document topic proportions as predictors, or a DPM of Normals is fit to the metadata for the words assigned to each topic. For sLDA, we used the stochastic EM procedure suggested by Chang [19] which consists of collapsed Gibbs sampling of the topic assignments followed by an optimization of the linear model coefficients and variance.

Because it is difficult to know a-priori what form the distributions over metadata given topics will take in real-world data, we also ran one experiment with synthetic data, where the metadata distributions were pre-specified. Synthetic data was used in order to determine whether TONPT can accurately recover complex metadata distributions in conjunction with topic distributions.

Our experiments focus on the quantitative measurement of how well each model can predict metadata values on unseen data and to assess qualitatively (e.g., via inspection) whether the trained models capture human intuition and domain knowledge with respect to the correlations between topics and metadata values.

6.5.1 Data

We ran our experiments on five real-world datasets. The first dataset is the Pang and Lee movie review dataset[2] [71]. The Movie Review dataset consists of 5006 movie reviews written by four authors. Each review is associated with a numerical rating, in the interval $[0, 1]$. To be consistent with Blei and McAuliffe, the Movie Review ratings were centered around 0 [15]. However, we do not use the logarithm of the data as in that work because we found that it did not make a significant difference.

The second dataset consists of the State of the Union Addresses (SotU) delivered by U.S. Presidents of the United States from the first address by George Washington in 1790 to the second address by Barack Obama in 2010. The data was prepared in the manner similar to that of Wang and McCallum [99], in which addresses are subdivided into individual documents by paragraph, resulting in 7507 (three-paragraph) documents. The metadata values for this dataset are the timestamps of

[2] We used: “original reviews for scale dataset v1.0"

the addresses, normalized to the interval $[0, 1]$ to match the processing described by Wang and McCallum [99].

The third dataset is a set of video game reviews collected from a major online gaming news site. This dataset consists of the text of the reviews as well as the difference between the rating given by the website and the average rating given by all such review sites as reported by the website Metacritic[3]. The task in this case is not just to regress sentiment, but to predict from the text alone whether the critic will over- or under-rate a game. This dataset consists of 1901 reviews.

The fourth dataset is the LDC-annotated portion of the Enron corpus [11]. The creators of this dataset chose 4,935 emails with timestamps covering approximately one year of time from January 2001 through December 2001 from the set of internal Enron e-mails made public as part of the investigation of illegal activities by the Enron Corporation. The metadata for this dataset are the timestamps present in the e-mail headers.

The final dataset is the Reuters 21578 corpus [52]. We used the subset of the articles for which topical tags are available, which consists of 11,367 documents. The articles were written during a time interval that spans most of the year 1987. The documents were processed using the same feature selection as for the other two datasets with an additional step in which variants of the names of the months were removed. These words are especially common in this corpus (e.g., in datelines) and provide a strong signal that is not based on the topical content of the articles (i.e., they allowed the models to “cheat"). After feature selection there are 10,230 non-empty documents in the final dataset.

For all real-world datasets, stopwords were removed using the MALLET stopwords list [61]. Words that occurring in more than a half (or a third, in the case of the Movie Review dataset) of the documents in a dataset and those that occurring in fewer than 1% were culled. Words were converted to lowercase, and documents that were empty after pre-processing were removed. Finally, for TOT the metadata were all normalized to the $(0, 1)$ interval to accommodate usage of the Beta.

[3] http://www.metacritic.com

6.5.2 Procedure

In our prediction experiments, models were trained on 90% of the documents and then were used to predict the metadata values for the remaining 10%. This was repeated in a cross-validation scheme ten times, with the training and evaluation sets being randomly sampled each time.

The quality of metadata predictions were measured with two metrics from the literature. The first metric is the formulation for the R^2 metric as used by Blei and McAuliff [15]

$$R^2(\mathbf{t}, \hat{\mathbf{t}}) = 1 - \frac{\sum_d (t_d - \hat{t}_d)^2}{\sum_d (t_d - \bar{t})^2},$$

where t_d is the actual metadatum for document d, $\hat{t}_d$ is the prediction and $\bar{t}$ is the mean of the observed t_ds. For certain linear models this metric measures the proportion of the variability in the data that is accounted for by the model. More generally, it is one minus the relative efficiency of the supervised topic model predictor to a predictor that always predicts the mean of the observed data points. Because we are defining R^2 in this way, the value can be negative (contrary to the implications of the square in the notation) in cases where the model being evaluated performs worse than the mean predictor. This metric is useful in cases where minimizing the sum squared error is desirable, but can be problematic when the true distribution of the metadata is skewed or multimodal, as one can achieve relatively high R^2 scores in these cases by predicting values with very low likelihood. For example, choosing a point with near-zero density halfway between two modes of equal height can lead to a high R^2, even though the probability of the true value being close to that point is near zero. Because of this deficiency of R^2 we also use a second metric based on the generalized 0-1 loss [74]. It is the proportion of test instances that are within a distance of Δ from the true value:

$$\textit{Zero-One}(\mathbf{t}, \hat{\mathbf{t}}; \Delta) = \frac{1}{N} \sum_d \begin{cases} 1 \text{ if } |t_d - \hat{t}_d| < \Delta \\ 0 \text{ otherwise} \end{cases}$$

where N is the number of test instances, $\Delta = 0.01 \cdot (t_{max} - t_{min})$ and t_{max} and t_{min} are the maximal and minimal observed metadata values respectively. When it is important that predictions are very close to the true values at least some of the time the 0-1 loss is an appropriate metric.

In order to assess the statistical significance of the results, a one-sided stochastic permutation test with 10,000 permutations was used to calculate p-values for the hypothesis that the mean R^2 for the model with the highest mean R^2 is greater than the mean R^2 for each of the other models being tested. P-values less than 0.05 were considered significant.

For the TONPT runs, G_0 was chosen to be a Normal with mean and variance equal to the sample mean and variance for the observed metadata, the total mass parameter $m = 10$, α_σ was 2.0 and β_σ was 1.0. For all runs, the document-topic parameter $\alpha = 0.1$, and the topic-word parameter $\beta = 0.01$.

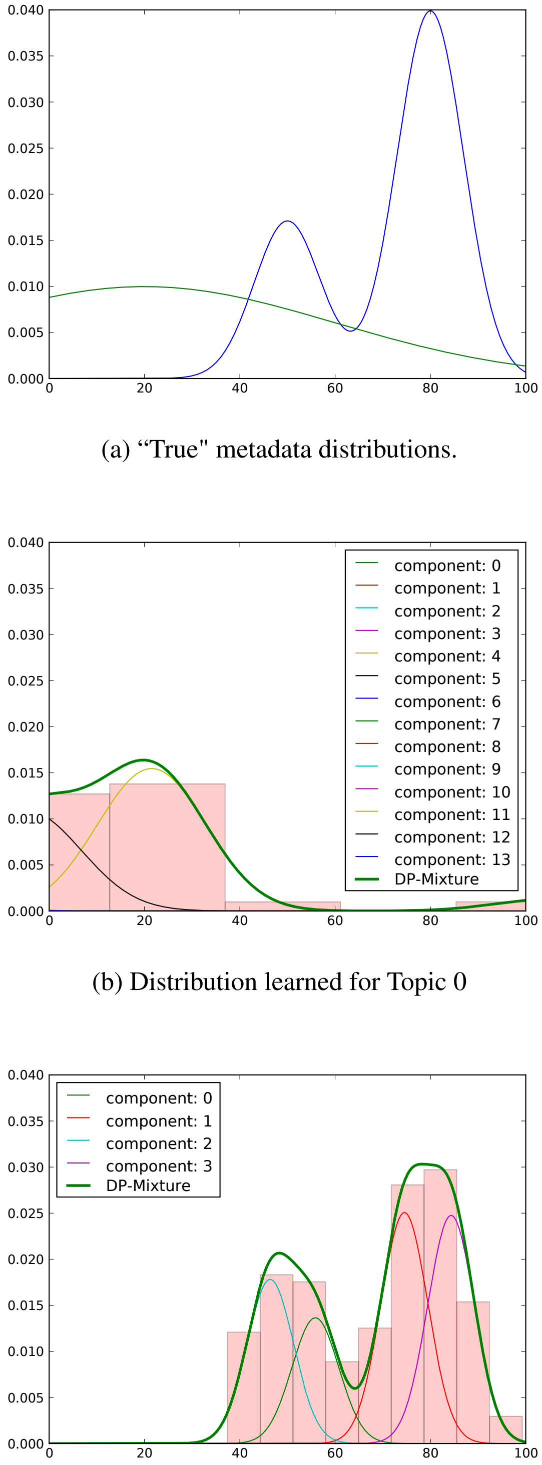
6.5.3 Synthetic Data Results

The synthetic dataset was created such that there are 2 topics and a vocabulary of 5 words: “common", “semicommon1", “semicommon2", “rare1" and “rare2". The “common" word occurs with 0.6 probability in both topics, “semicommon1" is slightly more likely than “semicommon2" in the first topic, and slightly less likely in the second topic. The “rare1" word is much more likely in the first topic than the second and “rare2" is much more likely in the second topic than the first, but both are much less common in general than the “semicommon"s.

Each topic was given a fixed metadata distribution:

$$t_0 \sim 0.3 \cdot f(50, 7) + 0.7 \cdot f(80, 7)$$
$$t_1 \sim f(20, 40)$$

Figure 6.4 shows that (for at least one run of the inference procedure) the model was able to separate the two topics and recreate the original metadata distributions. Note that the names of the topics do not align, due to non-identifiability in the model. Some runs result in slightly better approximations,



(a) “True" metadata distributions.

(b) Distribution learned for Topic 0

(c) Distribution learned for Topic 1

Figure 6.4: The estimated metadata distributions discovered for the synthetic dataset.

while others do worse, but these plots seem to be representative of TONPT's performance on this task.

6.5.4 Prediction Results

As discussed above, prediction in the case of sLDA is quite simple. For TOT and TONPT, prediction is complicated by the fact that these models have per-word metadata variables, and not per-document variables. Also, they do not produce a prediction using a simple dot product, but instead they provide a distribution over predicted values given a topic assignment. Luckily, in both TOT and TONPT, the posterior predictive distribution over topic assignments for a new document with observed words but unobserved metadata (i.e., the metadata variable is marginalized or expected over) is the same as that for vanilla LDA, thus a topic assignment vector $\mathbf{z}_{test\mathrm{d}}$ can be easily sampled for each test instance. Given point estimates $\mathbf{z}_{test\mathrm{d}}$ for the topic assignments for a test document d, there are two procedures for predicting a single metadata value t for the test document. First, the posterior probability of assigning the metadata variable for all of the words in that document the value of t is $p(t|\mathbf{z}_{test}) = \prod_i p(t|\mathbf{z}_{test,i})$, and so we might predict the mode of this distribution $\arg\max_t p(t|\mathbf{z}_{test})$. We call this *mode prediction*. Another approach would be to use $\mathbf{z}_{test}$ to obtain a point estimate for θ_{test}. Recognizing that in the generative model $p(t|\theta)$ is a mixture model with mixing parameter θ, the expected value of t is the θ_{test}-weighted average of the expected values of the per-topic metadata distributions. We call this *mean prediction*.

Figure 6.5 shows the log posterior distribution of t for four test documents in a test run of TONPT on the movie review dataset. This figure illustrates common performance characteristics of the mean and mode predictors. The first row shows examples where the mode predictor is better than the mean predictor. In these cases the mode predictor almost exactly matches the true value of the metadata for that document. The bottom row illustrates how, when the true value is at a non-maximal mode or at an area with lower posterior density, the mean predictor is typically closer to the true value (although it is almost never “close" to the true value in the way the mode predictor is often close). In evaluations where the sum squared error is part of the performance metric, then,

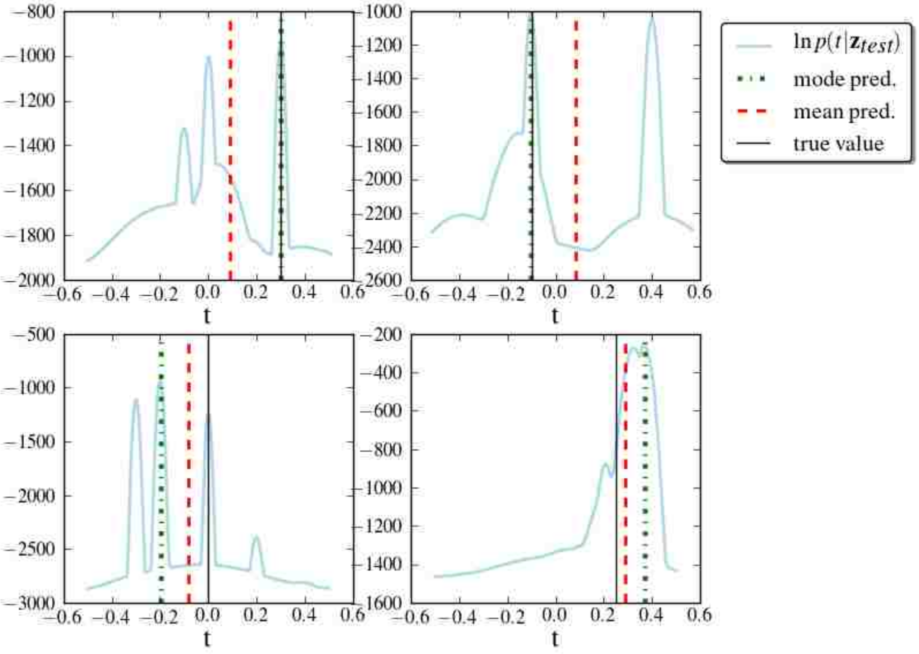


Figure 6.5: Log posterior TONPT predictive distributions for the metadata t on four Movie Review test documents.

we expect the mean prediction to perform better. In evaluations based on a generalized zero-one loss we expect the mode prediction to perform better.

Figure 6.6 shows the performance of the various models for the prediction task with 40 topics, which we found to be a number of topics at which peak performance occurs for most of the models. For each dataset and metric combination, the result for the top-performing model is highlighted, together with the results from other models that were not found to be statistically significant from the top-performing model's results. Note that this table represents a fairly large number of statistical tests, and so it is likely that some of the significance findings are erroneous, though in many cases the p-values were very small. The overall trends represented in this table are more important than individual significance test outcomes. These results show that TONPT with mean prediction is usually competitive on prediction tasks when performance is measured using R^2. With respect to R^2, TONPT with mean prediction is typically either the top-performing model, or its results are not statistically significantly worse than the best model's. It can also be seen that TONPT with mode prediction is significantly superior in terms of zero-one loss on the SotU and

Movie data, but fails on the game review bias prediction task. Figure 6.7 shows that the dominance of TONPT in zero-one loss is marked and consistent across various numbers of topics on most of the datasets. A somewhat surprising find in these results is that TOT actually does well at the two sentiment related tasks compared to sLDA, while sLDA is better than TOT at predicting timestamps in the SotU dataset. This is surprising since TOT was specifically formulated for temporal modeling and was originally evaluated on the SotU dataset, and so one might expect it to perform well on that task and poorly on the sentiment tasks.

6.5.5 Posterior Analysis

Figures 6.8 through 6.10 show the distribution over time for three topics found during runs of TONPT and approximately similar topics found during runs of PostHocTONPT on the SotU dataset. The topics shown in 6.8a and 6.9a are typical of the majority of the distributions we find with TONPT (simple symmetric distributions). The first topic deals with health care and health insurance, the second with the Mexican-American War.

Figure 6.10a shows an example of a more complex distribution. This particular topic appears to capture several conflicts the United States was involved in during the early 1800s, including The War of 1812 and several conflicts related to the Seminole Wars in Florida (a Spanish territory until 1821).

One potential concern for TONPT is that, because the DPM is so flexible it might not exert enough influence over the topics that are inferred. Figures 6.8 through 6.10 show that this concern is not valid. It can easily be seen that the joint model has learned distributions that more accurately and narrowly place events in history (the Mexican Amercan War topic shown in 6.9 is an especially good example of a topic that is very well historically placed by TONPT, but spread too wide by the PostHoc algorithm), while the post-hoc method spreads the topics throughout time, sometime combining topics as a result (e.g., healthcare versus public schools and other social programs). DPMs learned in the joint TONPT model also tend to have fewer components as shown in Figure 6.11. The reason this occurs is that, while it is true that the DPM can fit a wide variety of

Dataset	**Model**	**Mean** R^2	**Std Err**	**Mean** 0-1	**Std Err**	**Mean Time (s)**	**Std Err**
SotU	PostHocLinear	0.7936	0.0036	0.0820	0.0018	*209*	0.7652
	PostHocTONPT	0.5288	0.0044	0.0386	0.0014	8509	64.71
	sLDA	**0.8083**	0.0052	0.0911	0.0039	3804	69.44
	TOT_{mean}	0.7554	0.0219	0.0781	0.0027	**3413**	107.0
	TOT_{mode}	0.6999	0.0062	0.0907	0.0046	**3377**	58.49
	$\text{TONPT}_{\text{mean}}$	**0.8244**	0.0039	0.0856	0.0026	4244	69.66
	$\text{TONPT}_{\text{mode}}$	**0.8145**	0.0056	**0.2584**	0.0061	4078	40.09
Movie	PostHocLinear	0.2808	0.0168	0.0511	0.0029	*449*	0.6674
	PostHocTONPT	0.0549	0.0031	0.0818	0.0037	18843	216.8
	sLDA	**0.3383**	0.0133	0.0547	0.0062	**5646**	30.03
	TOT_{mean}	**0.3589**	0.0064	0.0513	0.0035	7775	91.06
	TOT_{mode}	0.3276	0.0074	0.0537	0.0036	7318	105.63
	$\text{TONPT}_{\text{mean}}$	**0.3505**	0.0188	0.0587	0.0022	7603	128.3
	$\text{TONPT}_{\text{mode}}$	0.0410	0.0298	**0.2096**	0.0061	7852	117.1
Game Bias	PostHocLinear	0.0622	0.0128	0.1005	0.0067	*283*	0.7721
	PostHocTONPT	-0.0068	0.0077	0.1126	0.0051	9576	333.7
	sLDA	0.0584	0.0117	0.1047	0.0067	3802	20.14
	TOT_{mean}	**0.2248**	0.0250	**0.1304**	0.0074	5539	96.54
	TOT_{mode}	**0.2219**	0.0176	**0.1225**	0.0068	5309	40.20
	$\text{TONPT}_{\text{mean}}$	0.0096	0.0522	0.1079	0.0071	**2837**	35.67
	$\text{TONPT}_{\text{mode}}$	-0.0870	0.0373	0.0738	0.0059	**2816**	35.67
Enron	PostHocLinear	0.2501	0.0099	0.0368	0.0028	*718.03*	1.554
	PostHocTONPT	0.1227	0.0065	0.0318	0.0035	26957	741.6
	sLDA	0.2390	0.0122	0.0415	0.0029	**7046.7**	27.67
	TOT_{mean}	0.0649	0.1048	0.0515	0.0042	11888	210.2
	TOT_{mode}	**0.3385**	0.0144	0.0669	0.0031	12126	107.6
	$\text{TONPT}_{\text{mean}}$	**0.3370**	0.0183	0.0659	0.0032	16506	226.1
	$\text{TONPT}_{\text{mode}}$	-0.0431	0.0296	**0.1472**	0.0035	16386	197.5
Reuters	PostHocLinear	**0.1031**	0.0072	0.0523	0.0013	*209.49*	0.4379
	PostHocTONPT	0.0639	0.0033	0.0442	0.0019	3777.3	17.35
	sLDA	0.0775	0.0132	0.0588	0.0052	**2689.73**	5.8156
	TOT_{mean}	-1.9861	0.2911	0.0117	0.0025	4096.88	53.092
	TOT_{mode}	-0.7873	0.0447	0.0704	0.0031	4405.37	133.04
	$\text{TONPT}_{\text{mean}}$	**0.1445**	0.0311	0.0709	0.0031	4170.68	21.801
	$\text{TONPT}_{\text{mode}}$	-0.0840	0.0270	**0.1246**	0.0039	4429.68	23.664

Figure 6.6: Prediction results for the 5 real-world datasets. Values that are not statistically significantly different from the best in each column are highlighted, except in the time category where the best value that is not PostHocLinear is highlighted.

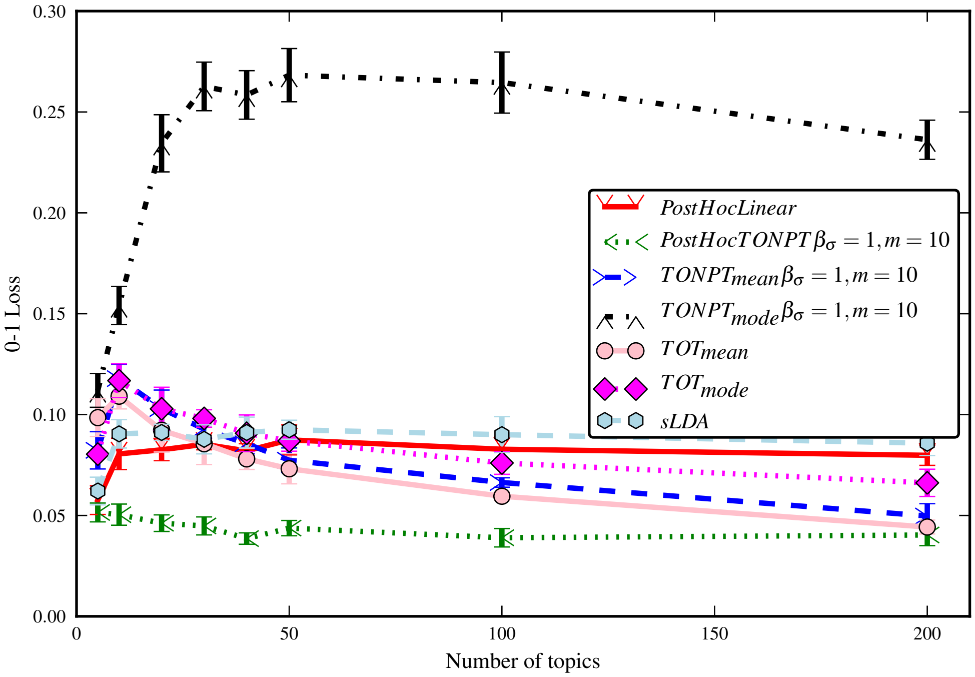
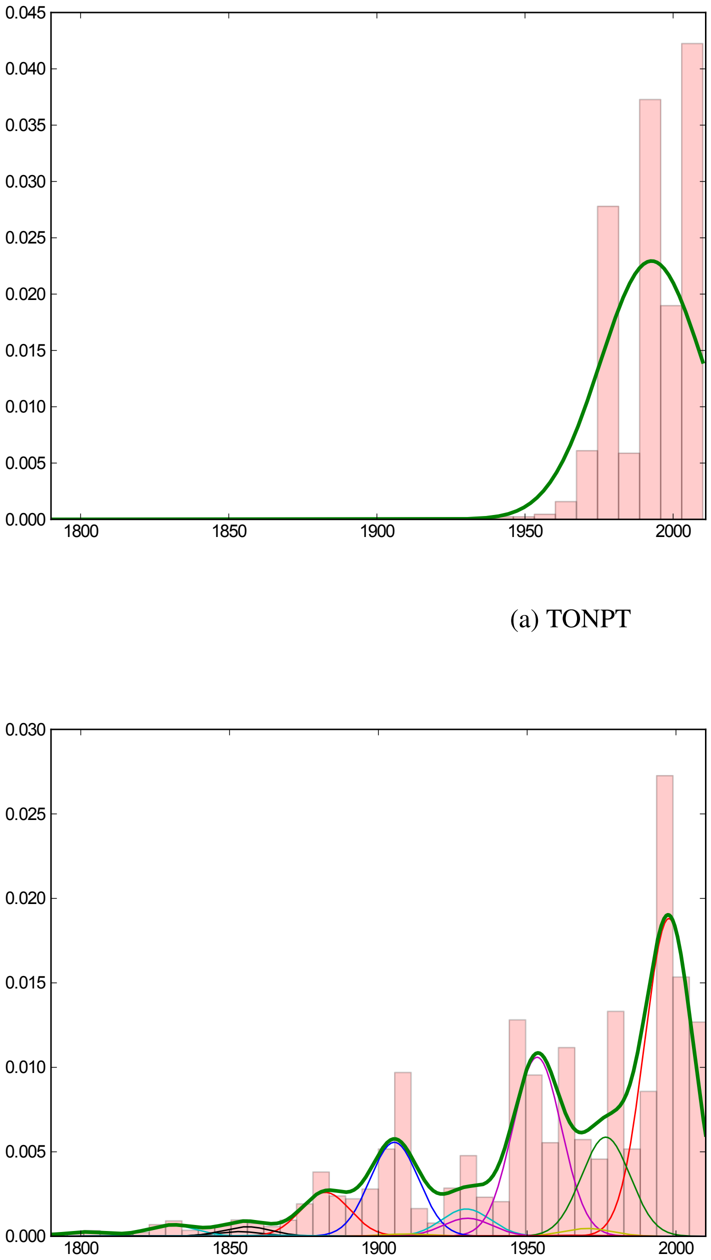


Figure 6.7: Zero-one loss of the various models on the SotU dataset.

distributions, in order for a token to be assigned to a topic in TONPT it must not only be a good fit for the topic's distribution over words and the document's distribution over topics but must also be a good fit for the topic's current metadata distribution. If a word is not a good fit to the metadata distribution the DPM has the capacity to add a new component to accommodate the token but, new components are created with low probability and so the other evidence must be very strong for this to occur.

6.6 Conclusion and Future Work

We have presented TONPT, a supervised topic model that models metadata using a nonparametric density estimator. The model accomplishes the goal of accommodating a wider range of metadata distributions and, in the case of the datasets that we evaluated against, prediction performance remains competitive with previous models when measured using mean predictions and evaluating with R^2 and is often superior when using mode predictions and evaluating with generalized zero-one



Top 10 Words in topic

health care
insurance coverage
cost medical
medicare quality
cover delivery

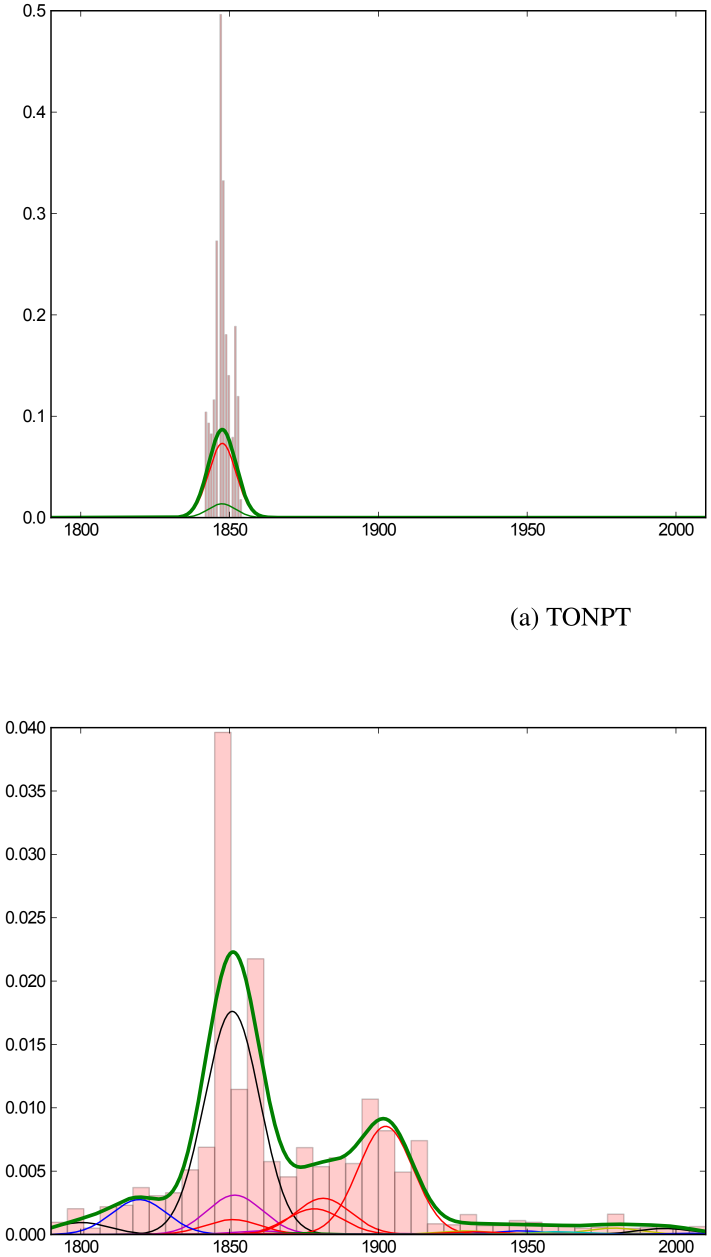
(a) TONPT

Top 10 Words in topic

health children
care education
schools school
security social
system insurance

(b) PostHoc TONPT

Figure 6.8: An example of health care topics. TONPT learns a topic that is focused on modern health care issues while the PostHoc model not historically specific and incorporates other social issues.



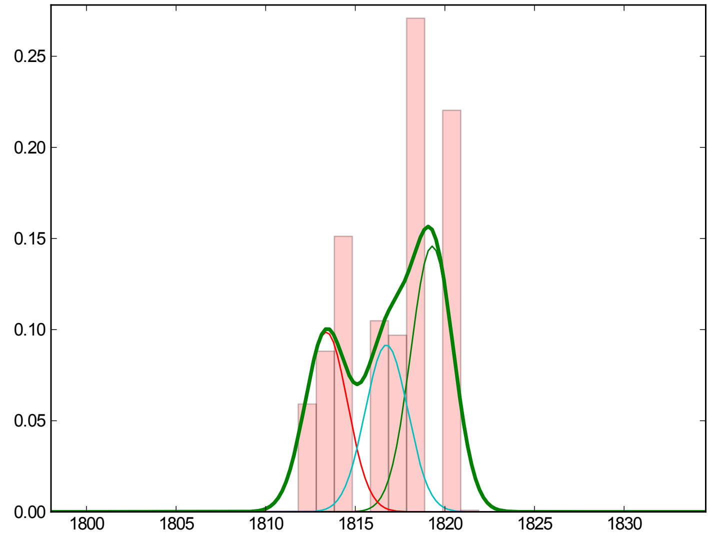
Top 10 Words in topic	
mexico	united
states	texas
treaty	territory
government	mexican
army	war

(a) TONPT

Top 10 Words in topic	
mexico	government
war	citizens
mexican	texas
territory	peace
military	american

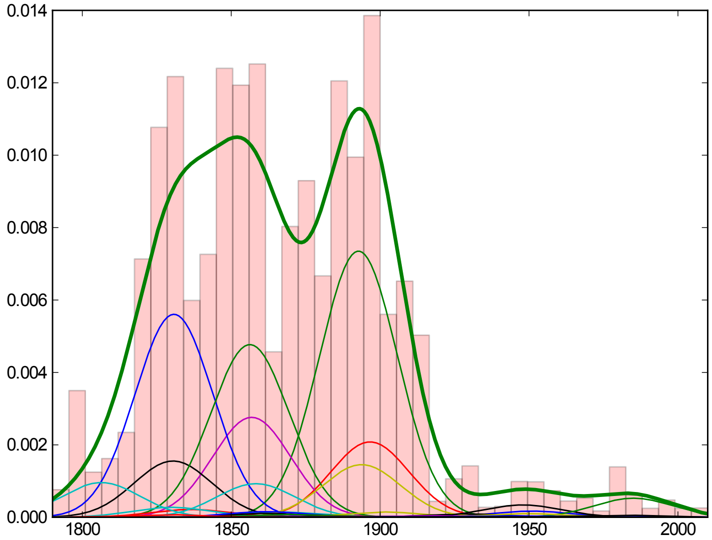
(b) PostHoc TONPT

Figure 6.9: An example of Mexican American War topics. TONPT learns a topic that is very specifically about the Mexican American War, and the words in the topic are all in a narrow range close to time range during which the war occurred (1946-1948). The PostHoc model has similar list of top words, but the tokens in the topic are spread out through time.



Top 10 Words in topic	
war	great
public	made
british	spain
country	state
force	citizens

(a) TONPT



Top 10 Words in topic	
treaty	united
great	britain
convention	british
claims	made
relations	subject

(b) PostHoc TONPT

Figure 6.10: Examples of topics dealing with early American wars. TONPT discovers a topic dealing with the War of 1812 and a series of conflicts related to the Seminole wars. Each of the two sets of conflicts is represented by a mode in the DPM. In the case of TONPT, there closest topics were much less historically distinct and combined a wide range of tokens about many different wars in American history.

Dataset	Model	m	Mean	Max
SotU	PostHocTONPT	1	12.52	23
		10	55.80	77
	TONPT	1	3.88	11
		10	8.28	37
Movie	PostHocTONPT	1	13.95	29
		10	70.63	107
	TONPT	1	3.45	11
		10	7.93	27
Game Bias	PostHocTONPT	1	38.03	54
		10	50.15	122
	TONPT	1	1.78	15
		10	2.13	19

Figure 6.11: Mean and max numbers of components in per-topic DPMs for TONPT and PostHoc-TONPT for two values of the total mass parameter m for three of the datasets.

loss. Future work could extend the model to multivariate metadata, such as temporal-spatial data including both timestamps and geolocation information. A multidimensional version of TONPT could be used to capture the development of trends in Twitter data, identifying geographic areas where topics originate and how they spread across the country over time.

6.7 Recent Developments

After the publication of this work, it came to light that Dubey, et. al. are working on a similar line of research which proposes a model they call non-parametric Topics over Time (npTOT) [28]. Our work differs from theirs in that we chose to focus more on metadata prediction, and various types of metadata variables, while they focus mostly on achieving lower perplexity and negative log likelihood values and focus on timestamp metadata. The npTOT model has the potential to model different types of phenomena, as they use dependent Dirichlet processes to achieve nonparametric distributions both over the topics and the metadata which are hierarchical and allow for the sharing of strength between the individual Dirichlet process mixtures.

Chapter 7

Evaluating Supervised Topic Models in the Presence of OCR Errors

Abstract

Supervised topic models are promising tools for text analytics that simultaneously model topical patterns in document collections and relationships between those topics and document metadata, such as timestamps. We examine empirically the effect of OCR noise on the ability of supervised topic models to produce high quality output through a series of experiments in which we evaluate three supervised topic models and a naive baseline on synthetic OCR data with various levels of degradation and on real OCR data from two different decades. The evaluation includes experiments with and without feature selection. Our results suggest that supervised topic models are no better, or at least not much better in terms of their robustness to OCR errors, than unsupervised topic models and that feature selection has the mixed result of improving topic quality while harming metadata prediction quality. For users of topic modeling methods on OCR data, supervised topic models do not yet solve the problem of finding better topics than the original unsupervised topic models.

7.1 Introduction

As text data becomes available in massive quantities, it becomes increasingly difficult for human curators to manually catalog and index modern document collections. Topic models, such as LDA [16], have emerged as one method of automatically discovering the topics discussed in a given document corpus. These models typically contain a latent topic label for each token. Tools from Bayesian statistics (such as Gibbs sampling and variational inference) are often used to infer labels in an unsupervised fashion given a set of documents. The topics discovered using topic models have the potential to be useful for facilitating browsing and discovering topical patterns and trends. This type of analysis has grown in popularity recently as inference on models containing large numbers of latent variables has become feasible due to computational advances.

Building on topic models, another class of document models called supervised topic models not only discovers topical labelings of words but also jointly models continuous metadata associated with the documents. For example, many documents have a known creation date or, in the case of product reviews, often have a numeric rating attached to them which summarizes the sentiment of the review's author. These models are especially promising for scholars of historical document collections because they allow for the discovery of patterns in the correlations between topics and metadata. For example, they can be used to trace the evolution of topics over time and to explore what the mention of a particular topic means in terms of the sentiment being expressed by an author. Many examples of supervised topic models exist in the literature. Examples of supervised topic models include sLDA [15], Topics Over Time [99], Dirichlet Multinomial Regression [65] and Topics Over Nonparametric Time [95].

These models have typically been trained and evaluated on relatively clean datasets but the modern explosion of text data includes vast amounts of historical documents, made available by means of Optical Character Recognition (OCR), which can introduce significant numbers of errors. Undertakings to produce such data include the Google Books, Internet Archive, and HathiTrust projects. In addition, researchers are having increasing levels of success in digitizing hand-written manuscripts [18], though error rates remain much higher than for OCR. Due to their nature, these collections often lack topical annotations and may contain documents with missing or incorrect metadata as evidenced by the many publicized problems with the metadata associated with Google Books documents. [1]

Depending on the age of a document and the way in which it was created, the OCR process results in text containing many types of noise, including character-level errors, which distort the counts of words and co-occurrences of words. However, finding good estimates for the parameters of supervised topic models requires that these counts be accurate. It is ostensible, therefore, that model quality must suffer, especially since the performance of completely unsupervised topic models are known to degrade in the presence of OCR errors [93].

Good supervised learning algorithms are substantially more immune to spurious patterns in the data created by noise for the following reason: under the mostly reasonable assumption that the process contributing the noise operates independently from the class labels, the noise in the features will not correlate well with the class labels, and the algorithm will learn to ignore those patterns arising from noise. Unsupervised models, in contrast, have no grounding in labels to prevent them from confusing patterns that emerge by chance in the noise with the “true" patterns of potential interest. For example, even on clean data, LDA will often do poorly if the very simple feature selection step of removing stop-words is not performed first.

Though we call the models discussed here “supervised" topic models, it should be clarified that these models are only supervised in terms of the real-valued metadata. This supervision is not supervision in the classical sense, where training data is supplied in the form of complete data, with values given for the latent variables in the target or evaluation data. The supervision in supervised topic models is much more akin to observing an additional feature for each document, a feature which may or may not be known for test data. The way in which words cluster by co-occurrence and/or in correlation with the metadata is completely unsupervised. One hope and expectation is that this extra feature will contribute information and improve the quality of the topics found by the model. Another advantage for supervised topic models is their ability to be used to predict

[1] http://chronicle.com/article/Googles-Book-Search-A/48245/

the values of the metadata for documents missing this value. Though we expect model quality to decrease, it is not well understood how sensitive supervised topic models are to OCR errors, or how quality deteriorates as the level of OCR noise increases.

In this paper we show how the performance of supervised topic models degrades as character-level noise is introduced. We demonstrate the effect using both artificially corrupted data (which has several desirable properties due to our ability to choose the source data and the amount of corruption) and an existing real-world OCR corpus (with subsets from two separate decades) and measure model performance both in terms of the ability of the models to effectively predict missing metadata values and in terms of the quality of the topics discovered by the model.

Because of the difficulties in evaluating topic models, even on clean data, these results should not be interpreted as definitive answers, but they do offer insight into prominent trends. It is our hope that this work will lead to an increase in the usefulness of collections of OCRed texts, as supervised topic models expose useful patterns to historians and other scholars.

7.2 Related Work

Both supervised and unsupervised topic models have been used previously to process documents digitized by OCR, including eighteenth-century American newspapers [68], OCRed editions of *Science* [14], OCRed NIPS papers [99], and books digitized by the Open Content Alliance [64]. Most of this previous work ignores the presence of OCR errors or attempts to remove corrupted tokens with special pre-processing such as stop-word removal and frequency cut-offs. Also, there are at least two instances of using topic modeling to improve the results of an OCR algorithm [32, 102].

Similar evaluations to ours have been conducted to assess the effect of OCR errors on supervised document classification [3, 87], information retrieval [10, 86], and a more general set of natural language processing tasks [57]. Results suggest that in these supervised tasks OCR errors have a minimal impact on the performance of the methods employed, though it has remained unclear how well these results transfer to unsupervised methods.

More directly related, unsupervised document clustering and topic modeling have been evaluated in the presence of OCR errors [93]. It was found that both clustering and topic modeling suffer increasing performance degradation as word error rates (WER) increase. In the case of document clustering, simple feature selection can almost completely ameliorate the deleterious effects of the noise, not only increasing performance evaluations at each noise level, but also allowing the performance achieved at high word error rates to be fairly similar to the performance at low ones. In the case of topic modeling, a performance increase can be achieved through feature selection, though the increase is additive in nature, and topic quality at high word error rates is significantly worse than at low ones. That is, feature selection improved performance but did not alter the shape of the quality degredation curve.

7.3 Models

We chose to compare the performance of three different supervised topic models as part of our evaluation: Supervised LDA (sLDA) [15], Topics Over Time (TOT) [99], and Topics Over Nonparametric Time (TONPT) [95]. The Dirichlet Multinomial Regression model [65] is another interesting supervised topic model but was not chosen because it is completely conditional on the metadata (e.g., no distribution is proposed for the metadata, but other variables depend on the values of the metadata) and is therefore not easily usable for metadata prediction. All of these models share a common LDA core, modeling documents as mixtures over topics and topics as mixtures over words. This LDA base has been extended in each case in order to jointly model document metadata with the topics and words. In the case of sLDA, metadata are modeled as per-document variables using a generalized linear model on the topic proportion vectors for each document together with a vector of linear coefficients and a variance parameter. TOT models metadata as per-word random variables (if there is only one metadata label for the document it is repeated for every word in the document) distributed according to per-topic Beta distributions. TONPT is based on TOT, but replaces the Beta distributions with Dirichlet Process Mixtures of Normals, which are Bayesian nonparametric density estimators. In addition, we use a baseline that is equivalent to unsupervised LDA during training; for prediction, a linear model is fit to the document topic proportions; we refer to this baseline as the PostHoc model.

7.4 Data

For our evaluations we used a real OCR dataset and two synthetic OCR datasets derived from natively digital datasets commonly used in the document modeling and text analytics literature. In all cases, we used timestamps associated with the documents as the metadata in the supervised topic models. The real OCR dataset is based on the Legacy Tobacco Documents Library [2] which was derived for the legal track of the 2006-2009 Text Retrieval Conferences. This real dataset was created from documents made public as part of various court cases against US tobacco companies. The documents were OCRed by the University of California at San Francisco. The documents in the collection span a wide range of time from the early 1900s to the early 2000s and are found in a similarly wide range of quality as to the fidelity of the OCR output. For our experiments we created two subsets of the data: one consisting of 5000 documents created from the year 1970 through the end of 1979 (Tobacco 70s)and another subset of 5000 documents created from the year 1990 through the end of 1999 (Tobacco 90s). Though gold standard transcriptions of the documents are not available against which word error rates could be calculated, we use these two datasets to represent relatively high and low word error rates (respectively) with the assumption that documents produced in the 1990s will have been produced using higher quality printing technologies and preserved for shorter periods of time, yielding higher quality document images and higher quality OCR output than those produced in the 1970s.

In addition to the real OCR data, we also used 2 synthetic OCR datasets that were created by rendering common text analytics datasets to images, stochastically degrading the images to various levels and then OCRing the degraded images [94]. The datasets we used consisted of the LDC annotated portion of the Enron e-mail corpus and the Reuters21578 corpus, both in uncorrupted form and at 5 levels of increasing average degradation. The synthetic datasets are useful for three

[2] `http://legacy.library.ucsf.edu`

reasons: first, because we know the source text, it is possible to calculate the average word error rate exactly; second, the same data are available at varying levels of degradation with word error rates (WER) ranging from very close to 0 to close to 50%, and so the effects of increasing degradation on prediction and topic quality can be assessed independently from the underlying data; finally, the synthetic data have both timestamps and topical class labels which can be correlated with the topics found by the various models in order to assess topic quality.

Both the Tobacco and synthetic datasets were lower-cased, had stopwords removed, and were also variously processed with a set of feature selection algorithms described in Section 7.5.

7.5 Experiments

In order to assess the effects of OCR errors on the models studied here we conducted a series of experiments. In these experiments, supervised topic models were estimated, with 100 topics, on the real and synthetic datasets described above. A 20-round cross-validation scheme was used in which the model was trained on 90% of the data, sampled randomly without replacement each round, and 10% of the data were withheld for the evaluation. The ability of the model in question to predict the metadata values for the held-out documents given their text was calculated and recorded each round.

7.5.1 Metrics

In order to assess the quality of the predictions, we used the formulation for the R^2 metric as used by Blei and McAuliff [15]

$$R^2(\mathbf{t}, \hat{\mathbf{t}}) = 1 - \frac{\sum_d (t_d - \hat{t}_d)^2}{\sum_d (t_d - \bar{t})^2},$$

where t_d is the actual metadatum for document d, $\hat{t}_d$ is the prediction and $\bar{t}$ is the mean of the observed t_ds. For linear models this metric measures the proportion of the variability in the data that is accounted for by the model. More generally, it is one minus the relative efficiency of the supervised topic model predictor to a predictor that always predicts the mean of the observed data points. This value can be negative in cases where the model being evaluated performs worse than the mean predictor. This metric is useful in cases where minimizing the sum squared error is desirable, but can be problematic when the true distribution of the metadata is skewed or multimodal, as one can achieve relatively high R^2 scores in these cases by predicting values with very low likelihood. For example, choosing a point with near zero density halfway between two modes of equal height can lead to a high R^2, even though the probability of the true value being close to that point is near zero.

Because of this deficiency of R^2 a second metric was used based on the generalized 0-1 loss [74]. It is the proportion of test instances that are within a distance of Δ from the true value:

$$\textit{Zero-One}(\mathbf{t}, \hat{\mathbf{t}}; \Delta) = \frac{1}{N} \sum_{d} \begin{cases} 1 \text{ if } |t_d - \hat{t}_d| < \Delta \\ 0 \text{ otherwise} \end{cases}$$

where N is the number of test instances, $\Delta = 0.01 \cdot (t_{max} - t_{min})$ and t_{max} and t_{min} are the maximal and minimal observed metadata values respectively. When it is important that predictions are very close to the true values at least some of the time the 0-1 loss is an appropriate metric.

Topic quality was assessed using two metrics: half-document perplexity and an N-fold cross-validation metric (CV Accuracy). The half-document perplexity was calculated using a procedure similar to the one described by Rosen-Zvi et. al. [80] in which point estimates for the topic-word and document-topic categoricals (for the test documents) were generated using the training data together with the metadata for the test documents and half of the words in each test document. Using these point estimates, the perplexity for the held-out words of the test documents was calculated.

The cross-validation metric is based on one first described by Griffiths et. al. [40]. To compute this metric, the learned topic assignments are used as features in 10-fold cross-validation classification of the documents with the average accuracy across the folds defining the value of the metric. This evaluation mechanism avoids a potential problem that arises when evaluating topic models using a likelihood-based measure, such as perplexity, on noisy data where feature selection can significantly change the number of word types and tokens remaining in documents as the word error rate increases, giving the false impression that topic quality is greater than it really is [93]. The CV Accuracy metric is computed for each fold of the experiment (to be clear, that is 10 folds of cross-validated classification for each of the 20 folds of the larger topic modeling experiment). The means and standard errors of these four metrics across all twenty folds were recorded.

To generate metadata predictions in the case of TOT and TONPT, we used two procedures discussed in Chapter 6. The first technique is based on the original prediction schema used by Wang and McCallum [99], in which the posterior density for assigning a single value to all the metadata variables of a test document (given the words in the test document and a proposed assignment of topic labels to those words) is calculated for a finite set of real-valued candidate points and the candidate with maximal posterior density is chosen as the prediction for that document. The topic assignments were made using Gibbs sampling, treating the assignments made to the training data previously as given. The candidates were chosen by sampling a value from the topic-conditional metadata distribution for each word given that word's topic assignment. We refer to this technique as mode prediction, since it aims to predict the value at the maximal mode of the joint posterior of the metadata given the words in the test document.

The second prediction procedure also made use of sampled topic assignments. The assignments were used to estimate the document-specific distribution over topics and this distribution was used to calculate the expected value of the document metadata variables as a weighted average of the expected value of the topic-specific metadata distributions. We refer to this technique as mean prediction, since it aims to predict the mean or expected value of the posterior metadata distribution.

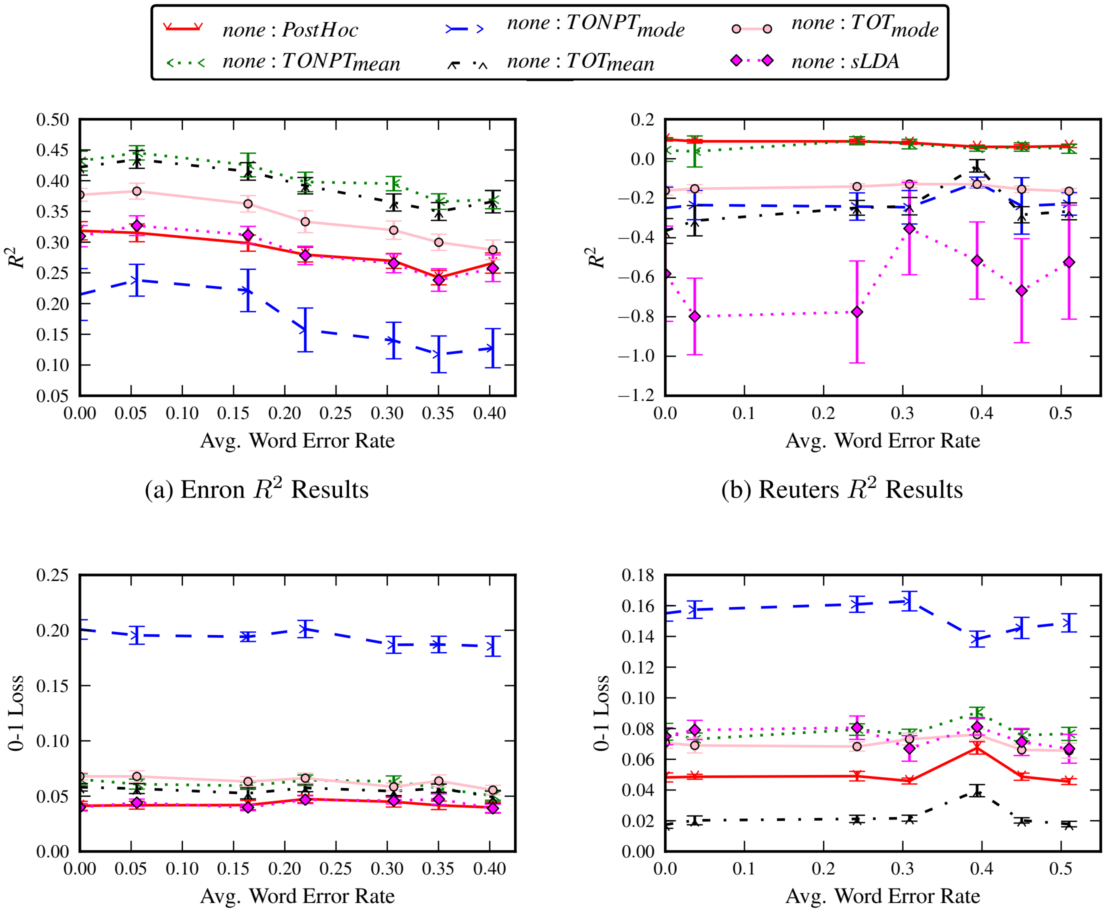
In order to determine to what degree the effects of the OCR errors could be mitigated, we ran our experiments both on the “raw" documents and on documents that had been processed using various unsupervised feature selection algorithms. The first method employed was a simple term frequency cut-off filter (TFCF), with a cut-off of 5 as employed by Wang and McCallum [99] (indicated with tfcf.5 in the results plots). The next method was a proportion filter (proportion) which eliminates any word occurring in fewer than 1% of the documents or in more than 50%. The next method employed was Term Contribution (TC), a feature selection algorithm developed for document clustering [54]. Term contribution is parameterized by the number of word types that are to remain after selection. We attempted two values for this parameter: 10,000 and 50,000 (tc.10000, and tc.50000). The final method we employed was Top-N per Document (TNPD) [92] (see Chapter 3), which is a simple feature selection algorithm that first assigns each type in every document a document-specific score (e.g., its TF-IDF weight), and then selects words to include in the final vocabulary by choosing the N words with the highest score from each document in the corpus. For $N = 1$ we abbreviate this as *tnpd.1*. In many of the plots and graphs that follow, the specific feature selector that was used with the dataset is specified before the name of the algorithm for which the results are being presented, with the feature selector name and the algorithm name being separated by a colon.

7.5.2 Synthetic Dataset Results

Here we discuss the results of the experiments on the raw (without feature selection) synthetic data in terms of both metadata prediction accuracy and topic quality.

Metadata Prediction Quality

Figure 7.1 shows the prediction results for the raw data. In terms of metadata prediction, Figure 7.1 shows that the algorithms appear to produce a fairly wide range of outcomes according to the R^2 metric. For the Enron data the TONPT and TOT models have a slight edge when used together with the mean prediction algorithm, although all of the curves appear to trend downward. In the case of the Reuters data, TONPT is mostly tied with the PostHoc baseline, with the other models performing mostly worse than the TONPT mean predictor. As the text error rate increases, there also appears to be a downward trend for the Reuters data, though the variance in the results is greater and the trend is difficult to discern. With respect to the 0-1 Loss metric, the TONPT model with the mode prediction algorithm is clearly superior to the other models.



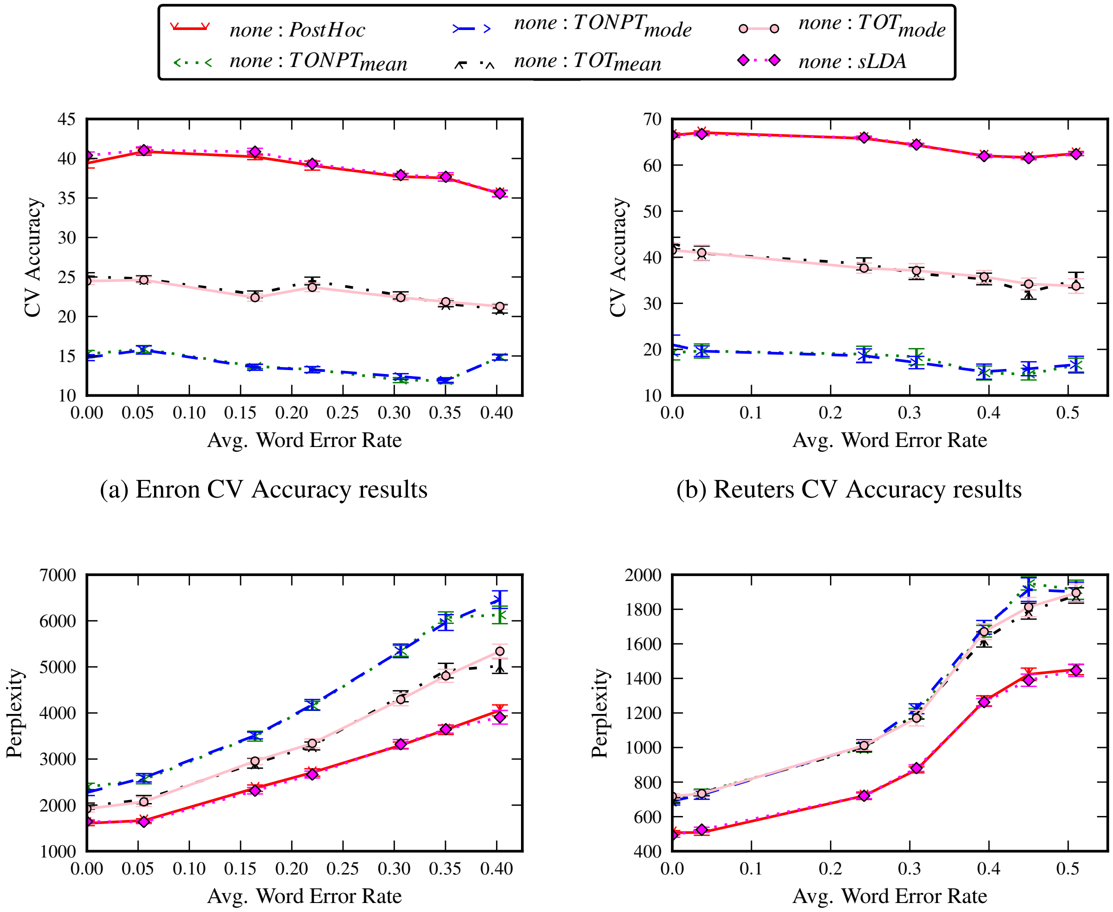
(a) Enron R^2 Results

(b) Reuters R^2 Results

(c) Enron 0-1 Loss Results

(d) Reuters 0-1 Loss Results

Figure 7.1: Timestamp prediction results for the two synthetic datasets without feature selection



(a) Enron CV Accuracy results

(b) Reuters CV Accuracy results

(c) Enron Half-document Perplexity results

(d) Reuters Half-document Perplexity results

Figure 7.2: Topic quality results for the two synthetic datasets without feature selection. Note that higher perplexity indicates worse topic quality.

Topic Quality

According to the two topic quality metrics, shown in Figure 7.2 the methods fare quite differently. Both the PostHoc baseline and sLDA are clearly superior in this regard, though their respective performance is indistinguishable in these graphs. The TOT model fares significantly worse and the TONPT model worse still. In addition, the graphs seem to suggest that all of the models degrade in performance at roughly the same rate, as the shapes of the curves in the graphs are fairly similar. This is especially obvious in the Reuters results, where the distinct difference in performance from the 0.3 to 0.4 word error rates is mirrored in similar upticks (in the case of perplexity) and dips (for the cross-validation metric) across all of the models tested.

Significance of Trends

In order to analyze the effects of noise more objectively, we attempted to quantify the effects of noise on performance by testing whether the results at each WER level were statistically significantly worse than results on the clean un-degraded data. This was determined using a one-sided stochastic permutation test of the hypothesis that the mean of the results at a given word error rate was worse (greater than in the case of perplexity, less than in the case of the other metrics) than the mean results on the clean data. P-values less than .05 were considered significant. Because of the large number of tests conducted here, we control for likely Type I errors by only considering a result significant if the results at all higher WERs are also significant or, in the case of the highest WER, if the the P-value is less than 0.001. Figure 7.3 shows, for each dataset and each metric, at what lowest WER a significantly worse outcome was found. These results suggest that all of the algorithms experience significant degradation in performance as the WER increases, often even at relatively low WERs. In some cases, the true supervised topic models appear to be slightly more robust, degrading at higher WERs than the PostHoc baseline, although that is not always true: several of the supervised topic models experience degradation at lower error rates in term of half-document perplexity and CV Accuracy on the Reuters dataset than the PostHoc baseline. In other words, supervised topic models may have an advantage in terms of how topic quality degrades as word error rates increase, but it is

		Lowest WER with Statistically Significantly Worse:			
Dataset	**Model**	$\mathbf{R^2}$	**0-1**	**Perplexity**	**CV Accuracy**
Enron	PostHoc	0.1641	-	0.0556	0.3063
	sLDA	0.2200	-	0.1641	0.2200
	TOT_{mean}	0.2200	-	0.0556	0.1641
	TOT_{mode}	0.2200	0.4031	0.0556	0.1641
	$TONPT_{mean}$	0.2200	0.3503	0.0556	0.3063
	$TONPT_{mode}$	0.2200	0.3063	0.0556	-
Reuters	PostHoc	0.3084	-	0.2422	0.2422
	sLDA	-	-	0.0372	0.2422
	TOT_{mean}	-	-	0.0372	0.0372
	TOT_{mode}	-	-	0.2422	0.2422
	$TONPT_{mean}$	-	-	0.0372	0.3938
	$TONPT_{mode}$	-	-	0.0372	0.2422

Figure 7.3: The lowest word error rate at which each model had significantly worse performance than the same model on the “clean" data and all following WERs were also significantly worse. A dash indicates that none of results at higher WERs were worse than the results on the clean data, or if they were, there was an insignificant difference at a higher WER.

not a large one, and it appears inconsistently across datasets. It should also be noted that, though the dashes indicate that no significant degradation in performance was found, the dashes usually occur in combinations that have very poor performance to begin with. For example, the PostHoc and sLDA methods both did not experience significant degradation for the Enron dataset and the 0-1 Loss metric but that is most likely because their performance in that category is approaching random performance (See Figure 7.1c).

Feature Selection

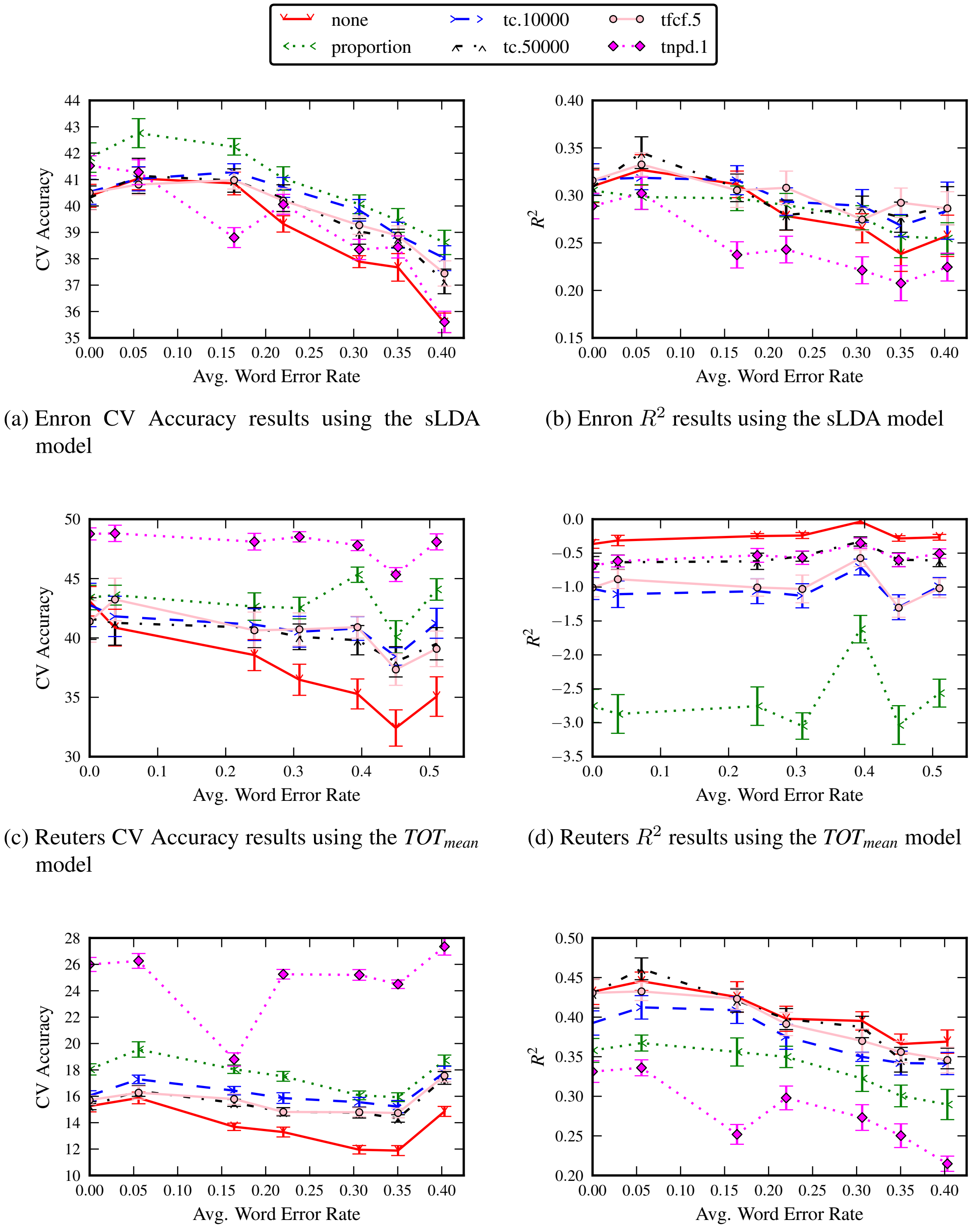
The next question that we wished to examine was whether and to what extent unsupervised feature selection algorithms can ameliorate the deleterious effects of OCR noise on supervised topic models. We repeated the above experiments on each of the feature-selected versions of the synthetic data discussed above. Figure 7.4 shows a few representative examples of the types of trends that we observed in the results across all models and feature selection algorithms. Figure 7.5 shows how the models compare given a single feature selection algorithm (tnpd.1). We found that it was typical for feature selection to improve CV Accuracy values, but at the same time hurt the R^2 and 0-1 Loss values. Furthermore, the feature selectors that improve CV Accuracy the most (typically the TNPD and proportion methods) were the ones that hurt R^2 and 0-1 Loss performance the most.

These outcomes have a fairly intuitive explanation if we consider the supervised topic modeling task as being composed of two somewhat orthogonal tasks: learning topic word clusterings and learning word metadata clustering. The learning of the topic clusters is an unsupervised task while the learning of the metadata clusters is supervised (since the metadata are observed for the training data). With this in mind, the results start to make sense. It has long been known that for supervised learning tasks, such as text classification, feature selection often hurts the learner's performance [60]. As discussed in Section 7.1, this is because a supervised algorithm is able to learn the correspondence between the features and the labels and gain information even from features that appear less correlated with the classifications a-priori. In contrast, feature selection is often essential in unsupervised learning tasks, such as document clustering and topic modeling [93]. This is because unsupervised learning algorithms have no frame of reference to distinguish extraneous patterns in the data from those that matter to the human conducting the analysis. So, in the case of supervised topic models, feature selection has the natural consequence of helping the performance of the unsupervised learning facet of the task and harming the performance of the supervised facet.

7.5.3 Real Dataset Results

Here we examine the prediction quality and topic quality outcomes for the real world datasets. As in the case of the synthetic datasets, we show results without and with feature selection. The results for the Tobacco 70s and Tobacco 90s datasets without feature selection are shown in Figure 7.6. CV Accuracy results were not calculated for the Tobacco data because it lacks topical labels for the documents.

The results match the findings on the synthetic dataset without feature selection in terms of the trends across models and across noise levels. In the case of each of the models across the evaluations used, the performance was better for the data from the 1990s. It is possible that some of the difference in performance could be attributed to variables other than increased noise; for



(a) Enron CV Accuracy results using the sLDA model

(b) Enron R^2 results using the sLDA model

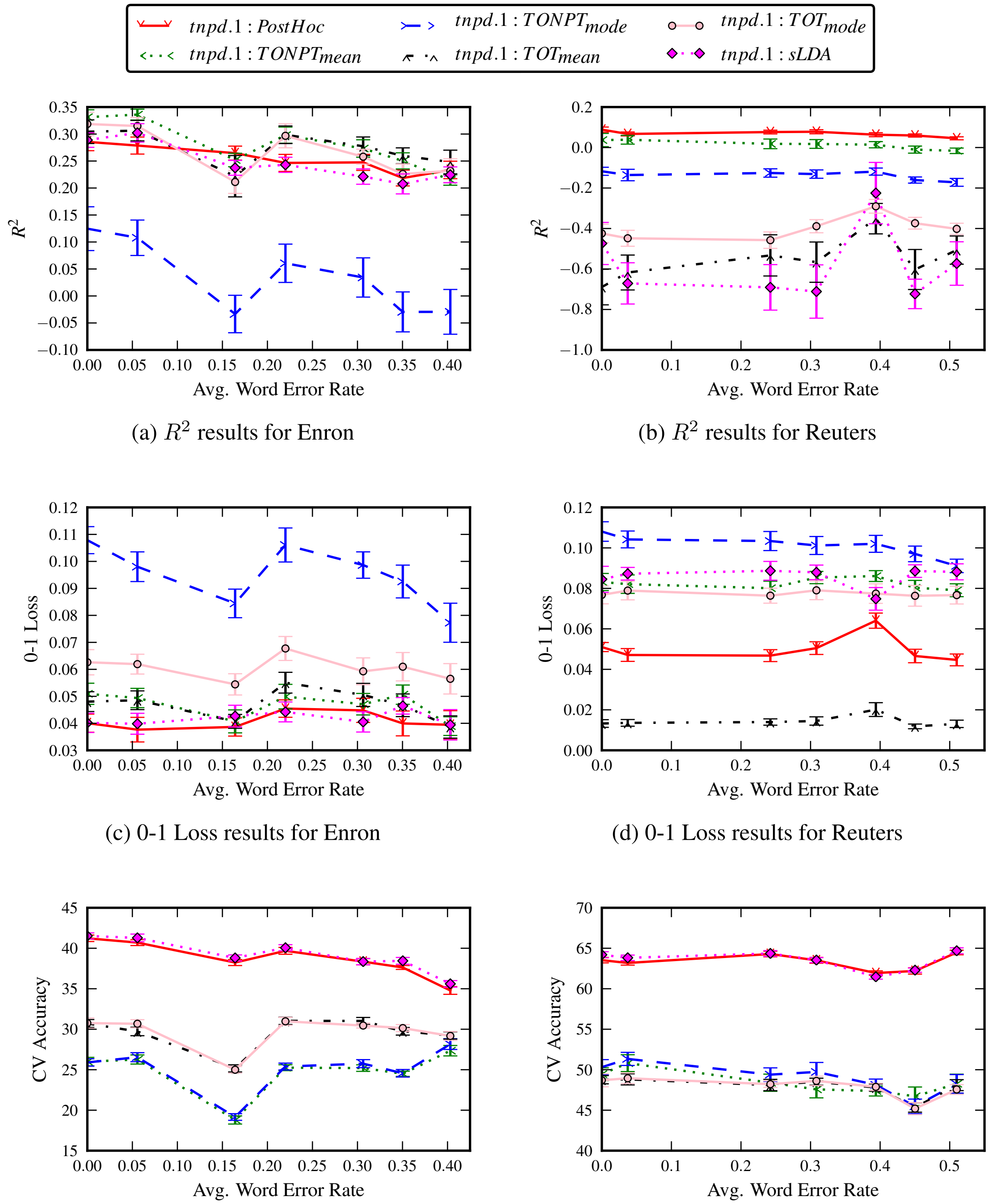
(c) Reuters CV Accuracy results using the TOT_{mean} model

(d) Reuters R^2 results using the TOT_{mean} model

(e) Enron CV Accuracy results using the $TONPT_{mean}$ model

(f) Enron R^2 results using the $TONPT_{mean}$ model

Figure 7.4: Example results for three (dataset, selector) pairs showing the effects of the feature selectors on CV Accuracy and R^2 scores. (a) and (b) show results on Enron with sLDA. The proportion and TFCF selectors improve both metrics, especially at higher error rates. (c) and (d) show the results on Reuters using TOT_{mean}. TNPD and proportion improve CV Accuracy, but all of the selectors hurt R^2 scores. (f) and (e) show the results on Enron using $TONPT_{mean}$. TNPD and proportion improve CV Accuracy, but all of the selectors hurt R^2 scores.



(a) R^2 results for Enron

(b) R^2 results for Reuters

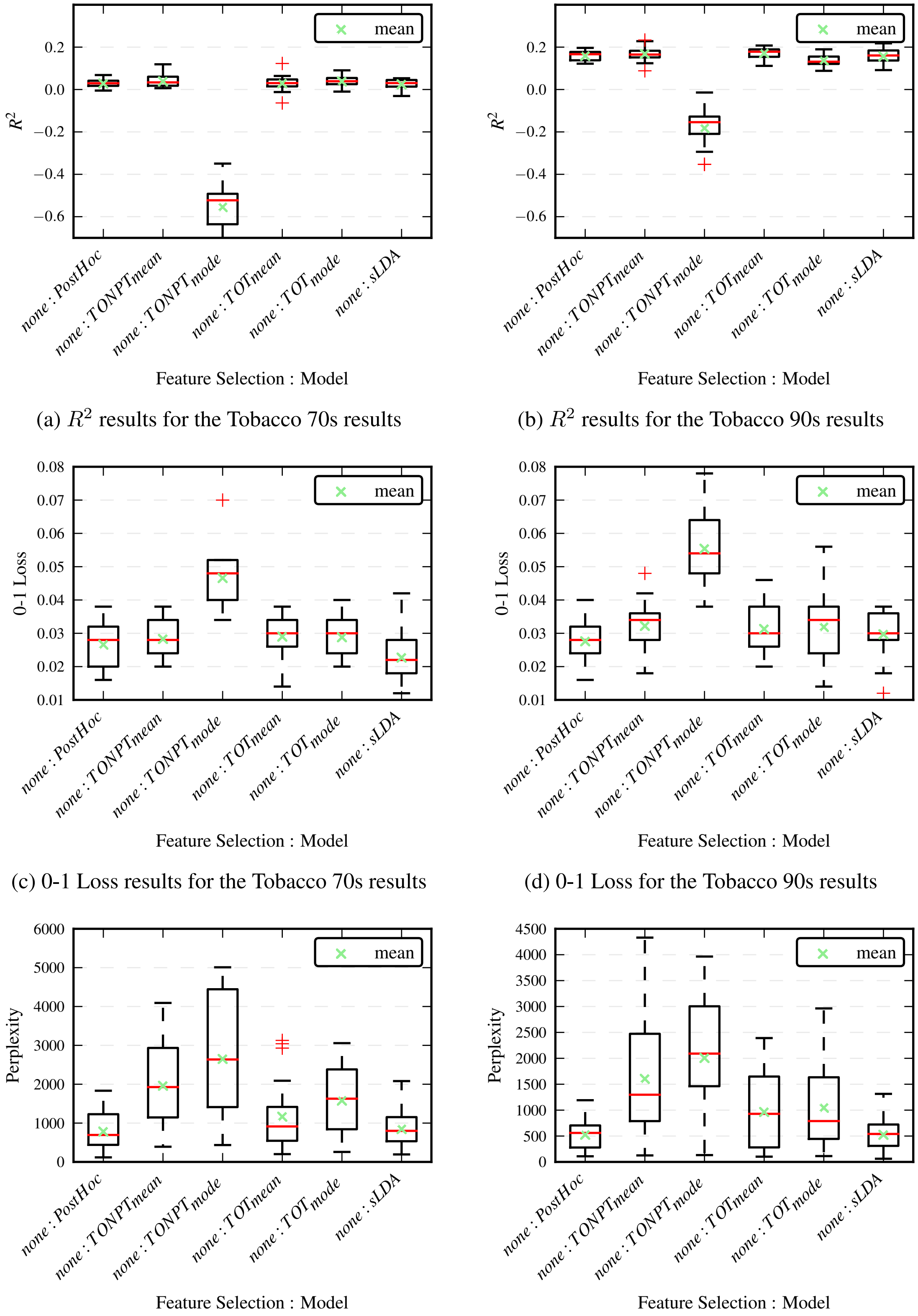
(c) 0-1 Loss results for Enron

(d) 0-1 Loss results for Reuters

(e) Cross Validation Accuracy results for Enron

(f) Cross Validation Accuracy results for Reuters

Figure 7.5: Results for the two synthetic datasets with TNPD feature selection. A comparison of these plots to those in Figures 7.1 and 7.2 show trends in the effect that feature selection has on the performance of the models. Specifically, while topic quality is improved (as evidenced by generally higher CV Accuracies), metadata prediction performance is actually hurt (according to both the R^2 and 0-1 Loss metrics). Half-document perplexity is not shown because feature selection skews that metric making it unreliable.



(a) R^2 results for the Tobacco 70s results

(b) R^2 results for the Tobacco 90s results

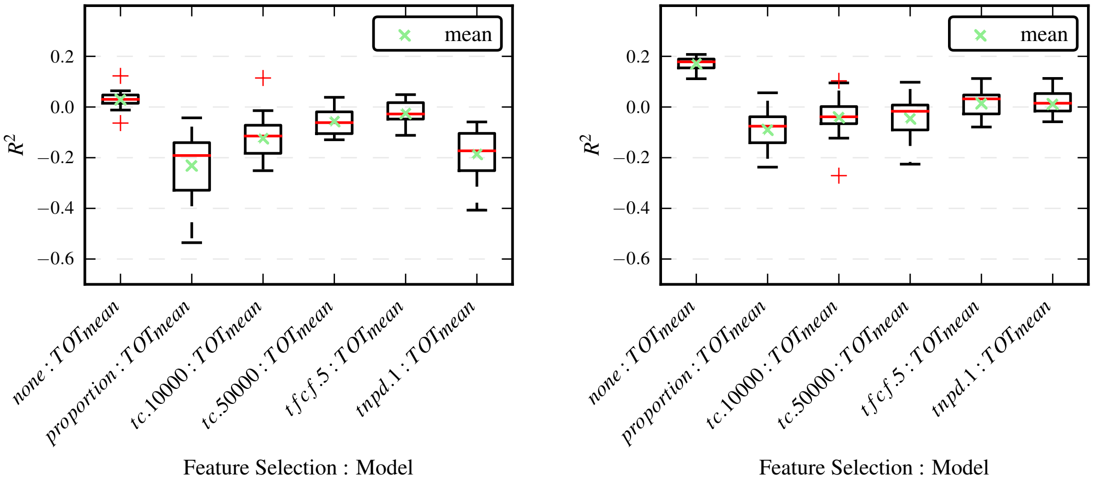
(c) 0-1 Loss results for the Tobacco 70s results

(d) 0-1 Loss for the Tobacco 90s results

(e) Perplexity for the Tobacco 70s results (lower is better)

(f) Perplexity for the Tobacco 90s results (lower is better)

Figure 7.6: Results for the two real world datasets without feature selection.



(a) R^2 results for the Tobacco 70s results

(b) R^2 results for the Tobacco 90s results

Figure 7.7: R^2 results for the two real world datasets with feature selection using the TOT_{mean} model. All of the feature selectors result in worse performance than is achieved without feature selection (none).

example, it could be the case that the environment of the 1970s was more static and that less changed from year to year, making timestamp prediction based solely on words difficult. While we do not rule out the possibility of such factors having an impact, it seems unlikely since not only the predictions, but also the topic quality (as measured with half-document perplexity) are impacted.

Again, we repeated the experiments after processing the Tobacco datasets with the feature selection algorithms. Figure 7.7 shows a result that was typical across the various supervised topic models. On this data, like on the synthetic data, feature selection appeared to mostly harm prediction results.

It is much more difficult to assess the impact of feature selection on topic quality for the Tobacco datasets, since there are no human-supplied topic labels for the documents. A visual inspection of the most probable words in topics produced with and without feature selection did not reveal any substantial differences in the quality of the topics. Figure 7.8 shows similar topic pairs found on two separate runs of the sLDA algorithm on the Tobacco 70s dataset, one run with no feature selection and the other with TNPD feature selection. The top 10 most probable words for

F. Selection	none	tnpd.1	none	tnpd.1
Words	alveolar	alveolar	cholesterol	blood
	cells	lung	patients	cholesterol
	pulmonary	cells	subjects	patients
	bacteria	macrophages	blood	heart
	lung	tissue	ldl	coronary
	macrophages	bacteria	serum	regan
	bacterial	pgnbr	disease	serum
	lymphatic	blood	cv	disease
	tissue	particles	age	uptake
	blood	cell	healthy	plasma

F. Selection	none	tnpd.1	none	tnpd.1
Words	paper	filter	sales	coupon
	mm	mm	market	promotion
	filter	paper	brand	sales
	tipping	plug	promotion	carton
	plug	tipping	total	pack
	weight	weight	year	display
	wrap	dilution	advertising	coupons
	length	acetate	share	purchase
	pressure	wrap	coupon	retail
	cigarette	drop	media	brand

Figure 7.8: The top 10 most probable words from four pairs of similar topics produced by the sLDA algorithm on the Tobacco 70s dataset with no feature selection and with TNPD feature selection.

each topic are listed. Although there are differences between the similar topic pairs, it is difficult to claim that any are significantly better than their counterparts.

7.6 Conclusion and Future Work

We assessed the impact of OCR errors on the performance of supervised topic models, specifically the Supervised LDA, Topics Over Time, and Topics Over Nonparametric Time models. Our results suggest that, despite having an extra source of information about the underlying data, the supervised topic models do not seem to have greater robustness to noise than traditional topic models, at least on these datasets. As document quality decreased, topic quality decreased at roughly the same pace for the supervised topic models that we examined as for the traditional topic model (PostHoc) baseline. Prediction quality also degraded significantly with increases in word error rates in the majority of cases. Feature selection did help improve the performance of the models in terms of topic quality, but at the same time hurt the performance of the models in terms of metadata prediction quality. It is possible that a feature selection algorithm tailored to this task might help alleviate this problem. Instead of using completely unsupervised feature selection algorithms, it could be possible to take into account the degree to which each feature correlates to training metadata values in a way that would allow for feature selection that would increase topic quality and not greatly decrease prediction quality (e.g., information gain or distributional mutual information).

Chapter 8

Conclusion

Throughout this dissertation we have presented a set of findings and tools that have the potential to improve the ability of scholars, historians, and analysts to conduct more effective analyses of their textual data.

In Chapter 2 we learned about the effectiveness of collapsed Gibbs sampling with a fully Bayesian mixture of multinomials document model to generally produce better quality document clusterings than can be produced with the same model using either classic expectation maximization or a Bayesian approach with variational Bayes inference and the mean field approximation. Furthermore, we were able to show that deterministic annealing could be applied to any of the algorithms to obtain a significant improvement over the classic versions for the document clustering problem.

Chapter 3 saw the introduction of an unsupervised feature selection algorithm, TNPD, that makes use of local feature weighting functions that can be computed very quickly in order to build a global vocabulary for a given document collection. While this technique did not necessarily produce better clusterings than other unsupervised feature selection algorithms, it was very competitive with state of the art results and orders of magnitude faster than methods that compute global scores for all features. We were able to use TNPD throughout the dissertation on both relatively “clean" text data and on data with large amounts of noise caused by OCR errors to improve the quality of both document clustering and unsupervised and supervised topic modeling results.

In Chapter 5 we presented a new language resource in the form of a synthetic OCR dataset that can effectively be used in various ways by scholars conducting research on problems relating to OCR error correction or in the area of text analytics in the presence of OCR errors. We showed that this dataset has desirable properties as a tool for researchers and demonstrated how it could be used to train an OCR error correction model to significantly reduce the number of errors in a real-world historical OCR dataset.

Chapters 4 and 7 examined the effects of noise in the form of real and synthesized OCR errors on the quality of the output of both clustering algorithms and unsupervised and supervised topic models. We found that all of these methods were significantly impacted by increasing levels of noise. In the case of document clustering, it was possible to almost completely overcome these effects. However, in the case of topic modeling, both with and without metadata, it was found that quality suffers and that feature selection has only a limited ability to alleviate the drop in performance.

Finally, in Chapter 6 we introduced Topics Over Nonparametric Time (TONPT), a supervised topic model, based on the Topics Over Time model that uses Bayesian nonparametric density estimation to model the topic-conditional distribution over document metadata. We showed that TONPT is competitive in terms of its ability to produce high quality predictions. Specifically, we showed that TONPT models can produce predictions using the mean or the mode of the posterior metadata distribution given the words in an unseen document and that the mean prediction method produces predictions that are competitive with other supervised topic models with respect to a metric based on sum squared error. We also showed that the mode prediction method produces predictions that are almost always significantly superior to other models with respect to a metric based on generalized 0-1 Loss. In terms of topic quality, the results shown in Chapter 7 suggest that TONPT produces inferior topics to sLDA and TOT, though the difference is less severe when feature selection is used to pre-process the data. The difference between the ability of TONPT to model the metadata distributions very well but to produce inferior topics suggests that there is a natural trade-off in terms of modeling the two modalities of the text. It is possible to have high quality topics or high quality metadata distributions, but improvements to the one appear to lead to the worsening of the other.

These results have shown how Bayesian text analytics can improve the performance of clustering and topic modeling of historical documents or of any documents that may contain noise and/or include metadata.

8.1 Future Work

The research presented here opens several avenues of future work. To begin with, the promising performance of deterministic annealing in Chapter 2 suggests that applications of Gibbs sampling in the context of document models, where the sampler often becomes “stuck" (as evidenced by a lack of label switching), could benefit from methods to widen the state space explored by the sampler. Deterministic annealing could be applied to many other models in the literature, including topic models such as LDA and TONPT.

Future work related to the modeling of noisy documents could focus on using the findings of these results to formulate topic models that are more robust to noise. It may be possible to incorporate error models into existing topic models in order to more effectively recover the original contents of noisy document during topic model inference.

Another potential line of research would seek to find more specific methods that could be used to explicitly control the trade-off between good models of topics and good models of metadata for supervised topic models.

In addition, the TONPT model has many potential future improvements that could be made. One improvement would be to provide a better theoretical motivation for the choice of the various hyper-parameter values used in the model. A further promising improvement would extend TONPT to multi-dimensional metadata (e.g., spatio-temporal data) through the use of Dirichlet process mixtures of multi-dimensional normals. There is already a growing body of work related to geolocated and spatio-temporal document modeling (e.g., [79]), though most of this work discretizes the spatial domain, whereas a multi-dimensional version of TONPT would be able to treat both the time and space domains as continuous.

Appendix A

Ratio of Beta Functions

For Dirichlet-multinomial models a common result is to arrive at complete conditionals that are the ratio of two Euler Beta functions where the vector argument of one of the Betas is equal to the other except with some integral amount added to each element. In these cases, it is possible to simplify the results further, though this simplification is not always necessarily more efficient, because of the existence of fast methods for estimating Gamma function calculations. In this section we illustrate the math that leads to the simplification.

Let $\beta = (\beta_1, \ldots, \beta_n)$, $\delta = (\delta_1, \ldots, \delta_n)$, and $\beta_* = (\beta_1 + \delta_1, \ldots, \beta_n + \delta_n)$, where each $\delta_1 \in \mathbb{N}^0$. Also, let $\mathrm{B} = \sum_{i=1}^{n} \beta_i$, and $\Delta = \sum_{i=1}^{n} \delta_i$. Finally, to account for potential sparsity in δ we divide the indices into two disjoint subsets $z = \{i : \delta_i = 0\}$ and $\underline{z} = \{i : \delta_i > 0\}$. With these variables defined we proceed as follows:

$$\begin{aligned}
\frac{B(\beta_*)}{\beta} =& \frac{\prod_{i=1}^{n}\Gamma(\beta_{*i})}{\Gamma(\sum_{i=1}^{n}\beta_{*i})}\frac{\Gamma(\sum_{i=1}^{n}\beta_i)}{\prod_{i=1}^{n}\Gamma(\beta_i)} \\
=& \frac{\prod_{i=1}^{n}\Gamma(\beta_{*i})}{\prod_{i=1}^{n}\Gamma(\beta_i)}\frac{\Gamma(\sum_{i=1}^{n}\beta_i)}{\Gamma(\sum_{i=1}^{n}\beta_{*i})} \\
=& \frac{\prod_{i\in z}\Gamma(\beta_i)}{\prod_{i\in z}\Gamma(\beta_i)}\frac{\prod_{i\in \underline{z}}\Gamma(\beta_{*i})}{\prod_{i\in \underline{z}}\Gamma(\beta_i)}\frac{\Gamma(\sum_{i=1}^{n}\beta_i)}{\Gamma(\sum_{i=1}^{n}\beta_{*i})} \\
=& \frac{\prod_{i\in \underline{z}}\Gamma(\beta_i+\delta_i)}{\prod_{i\in \underline{z}}\Gamma(\beta_i)}\frac{\Gamma(\mathrm{B})}{\Gamma(\mathrm{B}+\Delta)} \\
=& \prod_{i\in \underline{z}}\frac{\Gamma(\beta_i+\delta_i)}{\Gamma(\beta_i)}\frac{\Gamma(\mathrm{B})}{\Gamma(\mathrm{B}+\Delta)}
\end{aligned}$$

at this point we use the recurrence relation $\Gamma(x+1) = x\Gamma(x)$ which leads to:

$$\begin{aligned}
\Gamma(x+n) =& (x+n-1)\Gamma(x+n-1) \\
=& (x+n-2)(x+n-1)\Gamma(x+n-2) \\
\dots& \\
=& (\prod_{=0}^{n-1} x+i)\Gamma(x) \\
=& x^{(n)}\Gamma(x)
\end{aligned}$$

where $x^{(n)} = \prod_{=0}^{n-1} x + i$ is the rising factorial, also known as the Pochammer symbol. Continuing from above:

$$\begin{aligned}
\frac{B(\beta_*)}{\beta} &= \prod_{i\in\underline{z}} \frac{\Gamma(\beta_i+\delta_i)}{\Gamma(\beta_i)} \frac{\Gamma(\mathrm{B})}{\Gamma(\mathrm{B}+\Delta)} \\
&= \prod_{i\in\underline{z}} \frac{\beta_i^{(\delta_i)}\Gamma(\beta_i)}{\Gamma(\beta_i)} \frac{\Gamma(\mathrm{B})}{\mathrm{B}^{(\Delta)}\Gamma(\mathrm{B}} \\
&= \prod_{i\in\underline{z}} \beta_i^{(\delta_i)} \frac{1}{\mathrm{B}^{(\Delta)}} \\
&= \frac{\prod_{i\in\underline{z}} \beta_i^{(\delta_i)}}{\frac{1}{\mathrm{B}^{(\Delta)}}}
\end{aligned}$$

As mentioned previously, the ascending factorial can be problematic when it becomes more expensive than efficient approximations of the Γ function which can happen when Δ or δ_i becomes large. Also, when the base of the ascending factorial (x in $x^{(n)}$) is large, it can be possible to run into machine precision problems for relatively small values of the power of the factorial (n in $x^{(n)}$).

References

[1] ABBYY. ABBYY finereader. `http://finereader.abbyy.com`, 2010.

[2] Adobe Systems Inc. Acrobat pro. `http://www.adobe.com/products/acrobatpro.html`, 2010.

[3] Sumeet Agarwal, Shantanu Godbole, Diwakar Punjani, and Shourya Roy. How much noise is too much: A study in automatic text classification. In *Proceedings of the 7th IEEE International Conference on Data Mining*, pages 3–12, Omaha, NE, 2007.

[4] David Aldous, Illdar Ibragimov, Jean Jacod, and David Aldous. Exchangeability and related topics. In *ÃL'cole d'ÃL'tÃl' de ProbabilitÃl's de Saint-Flour XIII âĂŤ 1983*, volume 1117 of *Lecture Notes in Mathematics*, pages 1–198. Springer Berlin / Heidelberg, 1985.

[5] Arthur Asuncion, P. Smyth, and M. Welling. Asynchronous distributed learning of topic models. In D. Koller, D. Schuurmans, Y. Bengio, and L. Bottou, editors, *Advances in Neural Information Processing Systems 21*, volume 21, pages 81–88. MIT Press, 2009.

[6] Arthur Asuncion, Max Welling, Padhraic Smyth, and Yee Whye Teh. On smoothing and inference for topic models. In *Proceedings of the 25th Conference on Uncertainty in Artificial Intelligence*, pages 27–34, Montreal, Quebec, 2009. AUAI Press.

[7] Henry S. Baird. Document image defect models. In *Proceedings of the IAPR Workshop on Syntactic and Structural Pattern Recognition*, pages 38–46, Los Alamitos, CA, 1990. IEEE Computer Society Press.

[8] Henry S. Baird. The state of the art of document image degradation modelling. In Bidyut B. Chaudhuri, editor, *Digital Document Processing*, Advances in Pattern Recognition, pages 261–279. Springer London, 2007.

[9] Arindam Banerjee and Sugato Basu. Topic models over text streams: A study of batch and online unsupervised learning. In *Proceedings of the SIAM International Conference on Data Mining*, pages 431–436, Minneapolis, MN, April 2007.

[10] Steven M. Beitzel, Eric C. Jensen, and David A. Grossman. A survey of retrieval strategies for OCR text collections. In *In Proceedings of the Symposium on Document Image Understanding Technologies*, pages 145–152, Greenbelt, MD, 2003.

[11] Michael W. Berry, Murray Brown, and Ben Signer. 2001 Topic annotated Enron email data set, 2007.

[12] Christopher M. Bishop. *Pattern Recognition and Machine Learning*, volume 4. Springer New York, 2006.

[13] David Blackwell and James B. MacQueen. Ferguson distributions via Pólya urn schemes. *The Annals of Statistics*, 1(2):353–355, 1973.

[14] David M. Blei and John D. Lafferty. Dynamic topic models. In *Proceedings of the 23rd International Conference on Machine Learning*, pages 113–120, Pittsburgh, PA, June 2006.

[15] David M. Blei and Jon D. McAuliffe. Supervised topic models. *arXiv:1003.0783*, March 2010.

[16] David M. Blei, Andrew Y. Ng, and Michael I. Jordan. Latent Dirichlet allocation. *Journal of Machine Learning Research*, 3:993–1022, 2003.

[17] David M. Blei, Thomas L. Griffiths, Michael I. Jordan, and Joshua B. Tenenbaum. Hierarchical topic models and the nested chinese restaurant process. In Sebastian Thrun, Lawrence Saul, and Bernhard Schölkopf, editors, *Advances in Neural Information Processing Systems 16*, volume 16, pages 17–25. MIT Press, 2004.

[18] Horst Bunke. Recognition of cursive Roman handwriting- past, present and future. In *Proceedings of the 7th International Conference on Document Analysis and Recognition*, pages 448–459, Edinburgh, Scotland, Aug 2003.

[19] Jonathan Chang. *Uncovering, Understanding, and Predicting Links*. PhD thesis, Princeton University, November 2011. `http://gradworks.umi.com/34/81/3481564.html`.

[20] Jonathan Chang and David M. Blei. Relational topic models for document networks. In *Proceedings of the 12th International Conference on Artificial Intelligence and Statistics*, pages 81–88, Clearwater Beach, FL, 2009.

[21] Jonathan Chang, Jordan Boyd-Graber, Sean Gerrish, Chong Wang, and David M. Blei. Reading tea leaves: How humans interpret topic models. In Y. Bengio, D. Schuurmans,

J. Lafferty, C. K. I. Williams, and A. Culotta, editors, *Advances in Neural Information Processing Systems 22*, volume 22, pages 288–296. MIT Press, 2009.

[22] Ming-Hui Chen, Qi-Man Shao, and Joseph George Ibrahim. *Monte Carlo Methods in Bayesian Computation*. Springer, 2000.

[23] Scott Deerwester, Susan T. Dumais, George W. Furnas, Thomas K. Landauer, and Richard Harshman. Indexing by latent semantic analysis. *Journal of the American Society for Information Science*, 41(6):391–407, 1990.

[24] Inderjit Dhillon, Jacob Kogan, and Charles Nicholas. *Feature Selection and Document Clustering*, chapter 4, pages 73–100. Springer-Verlag, 2003.

[25] Inderjit S. Dhillon, Yuqiang Guan, and J. Kogan. Iterative clustering of high dimensional text data augmented by local search. In *Proceedings of the 2002 IEEE International Conference on Data Mining*, pages 131–138, Los Alamitos, CA, 2002. IEEE Computer Society.

[26] Byron Dom. An information-theoretic external cluster-validity measure. Technical Report RJ10219, IBM, October 2001. `http://citeseer.ist.psu.edu/dom01informationtheoretic.html`.

[27] Gabriel Doyle and Charles Elkan. Accounting for burstiness in topic models. In *Proceedings of the 26th Annual International Conference on Machine Learning*, pages 281–288, Montreal, Quebec, June 2009.

[28] Avinava Dubey, Ahmed Hefny, Sinead Williamson, and Eric P. Xing. A non-parametric mixture model for topic modeling over time. *arXiv:1208.4411*, August 2012.

[29] Charles Elkan. Mixture models. Technical report, The University of California, San Diego, 2010. `http://cseweb.ucsd.edu/users/elkan/250Bwinter2011/mixturemodels.pdf`.

[30] Michael D. Escobar. Estimating normal means with a Dirichlet process prior. *Journal of the American Statistical Association*, 89(425):268–277, 1994.

[31] Michael D. Escobar and Mike West. Computing nonparametric hierarchical models. volume 133 of *Lecture Notes in Statistics*, pages 1–22. Springer New York, 1998.

[32] Faisal Farooq, Anurag Bhardwaj, and Venu Govindaraju. Using topic models for OCR correction. *International Journal on Document Analysis and Recognition*, 12(3):153–164, September 2009.

[33] T.S. Ferguson. A Bayesian analysis of some nonparametric problems. *The Annals of Statistics*, pages 209–230, 1973.

[34] Andrew Gelman, John B. Carlin, Hal S. Stern, and Donald B Rubin. *Bayesian Data Analysis*. Chapman & Hall/CRC, second edition, 2004.

[35] Sean M. Gerrish and David M. Blei. Predicting legislative roll calls from text. In Lise Getoor and Tobias Scheffer, editors, *Proceedings of the 28th International Conference on Machine Learning*, pages 489–496, Bellevue, WA, June 2011.

[36] Mark Girolami and Ata Kabán. On an equivalence between PLSI and LDA. In *Proceedings of the 26th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*, pages 433–434, Toronto, Canada, 2003. ACM.

[37] Sharon Goldwater and Thomas L. Griffiths. A fully Bayesian approach to unsupervised part-of-speech tagging. In *Proceedings of the 45th Annual Meeting of the Association of Computational Linguistics*, pages 744–751, Prague, Czech Republic, June 2007. Association for Computational Linguistics.

[38] Google, Inc. Tesseract. `http://code.google.com/p/tesseract-ocr`, 2010.

[39] Thomas L. Griffiths and Mark Steyvers. Finding scientific topics. In *Proceedings of the National Academy of Sciences*, volume 101 (suppl. 1), pages 5228–5235, 2004.

[40] Thomas L. Griffiths, Mark Steyvers, David M. Blei, and Joshua B. Tenenbaum. Integrating topics and syntax. In Lawrence K. Saul, Yair Weiss, and Léon Bottou, editors, *Advances in Neural Information Processing Systems 17*, volume 17, pages 537–544. MIT Press, 2005.

[41] Aria Haghighi and Dan Klein. Unsupervised coreference resolution in a nonparametric Bayesian model. In *Proceedings of the 45th Annual Meeting of the Association of Computational Linguistics*, pages 848–855, Prague, Czech Republic, June 2007. Association for Computational Linguistics.

[42] Matthew Hoffman, David M. Blei, and Fancis Bach. Online learning for latent Dirichlet allocation. In J. Lafferty, C. K. I. Williams, J. Shawe-Taylor, R.S. Zemel, and A. Culotta, editors, *Advances in Neural Information Processing Systems 23*, volume 23, pages 856–864. MIT Press, 2010.

[43] Thomas Hofmann. Probabilistic latent semantic indexing. In *Proceedings of the 22nd Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*, pages 50–57, Berkeley, CA, 1999. ACM.

[44] Yi Hong, Sam Kwong, Yuchou Chang, and Qingsheng Ren. Unsupervised feature selection using clustering ensembles and population based incremental learning algorithm. *Pattern Recognition*, 41(9):2742–2756, 2008.

[45] Lawrence Hubert and Phipps Arabie. Comparing partitions. *Journal of Classification*, 2(1): 193–218, December 1985.

[46] I.R.I.S. s.a. Readiris pro. `http://www.irislink.com`, 2010.

[47] Mohammad-Amin Jashki, Majid Makki, Ebrahim Bagheri, and Ali A. Ghorbani. An iterative hybrid filter-wrapper approach to feature selection for document clustering. In *Proceedings of the 22nd Canadian Conference on AI*, pages 74–85, 2009.

[48] Thorsten Joachims. A probabilistic analysis of the Rocchio algorithm with TFIDF for text categorization. In *Proceedings of the 14th International Conference on Machine Learning*, Nashville, TN, July 1997.

[49] David Reed Jordan. Daily battle communiques, 1944-1945. Harold B. Lee Library, L. Tom Perry Special Collections, MSS 2766, `http://lib.byu.edu/digital/spc/eisenhower`, 1945.

[50] Kevin Atkinson. GNU Aspell. `http://aspell.net`, 2011.

[51] Ken Lang. NewsWeeder: Learning to filter netnews. In *Proceedings of the 12th International Conference on Machine Learning*, pages 331–339, Tahoe City, CA, July 1995.

[52] D. Lewis. Reuters-21578 text categorization test collection. *`http://www.research.att.com/~lewis`*, 1997.

[53] Wei Li and Andrew Mccallum. Pachinko allocation: DAG-structured mixture models of topic correlations. In *Proceedings of the 23rd International Conference on Machine Learning*, pages 577–584, Pittsburgh, PA, June 2006.

[54] Tao Liu, Shengping Liu, Zheng Chen, and Wei-Ying Ma. An evaluation on feature selection for text clustering. In *Proceedings of the 20th International Conference on Machine Learning*, pages 488–496, Washington D.C., August 2003.

[55] A.Y. Lo. On a class of Bayesian nonparametric estimates: I. density estimates. *The Annals of Statistics*, 12(1):351–357, 1984.

[56] Daniel Lopresti. Performance evaluation for text processing of noisy inputs. In *Proceedings of the 20th Annual ACM Symposium on Applied Computing*, pages 759–763, Santa Fe, NM, March 2005.

[57] Daniel Lopresti. Optical character recognition errors and their effects on natural language processing. In *Proceedings of the 2nd Workshop on Analytics for Noisy Unstructured Text Data*, pages 9–16, Singapore, 2008. ACM.

[58] William B. Lund and Eric. K Ringger. Improving optical character recognition through efficient multiple system alignment. In *Proceedings of the 9th ACM/IEEE-CS Joint Conference on Digital Libraries*, pages 231–240, Champaign, IL, June 2009. ACM.

[59] William B. Lund, Daniel D. Walker, and Eric K. Ringger. Progressive alignment and discriminative error correction for multiple OCR engines. In *Proceedings of the 11th International Conference on Document Analysis and Recognition*, pages 764–768, Beijing, China, September 2011.

[60] Andrew McCallum and Kamal Nigam. A comparison of event models for naive Bayes text classification. In *Proceedings of the AAAI-98 Workshop for Text Categorization*, Madison, WI, July 1998.

[61] Andrew Kachites McCallum. MALLET: A machine learning for language toolkit. `http://mallet.cs.umass.edu`, 2002.

[62] Marina Meilă. Comparing clusterings—an information based distance. *Journal of Multivariate Analysis*, 98(5):873–895, 2007.

[63] Marina Meilă and David Heckerman. An experimental comparison of model-based clustering methods. *Machine Learning*, 42(1–2):9–29, January 2001.

[64] David Mimno and Andrew McCallum. Organizing the OCA: learning faceted subjects from a library of digital books. In *Proceedings of the 7th ACM/IEEE-CS Joint Conference on Digital Libraries*, pages 376–385, Vancouver, Canada, 2007. ACM Press.

[65] David Mimno and Andrew McCallum. Topic models conditioned on arbitrary features with Dirichlet-multinomial regression. In *Proceedings of the 24th Conference on Uncertainty in Artificial Intelligence*, pages 411–418, Helsinki, Finland, 2008. AUAI Press.

[66] Cosmin Munteanu, Ronald Baecker, Gerald Penn, Elaine Toms, and David James. The effect of speech recognition accuracy rates on the usefulness and usability of webcast archives. In

Proceedings of the SIGCHI Conference on Human Factors in Computing Systems, pages 493–502, Montreal, Quebec, Canada, 2006. ACM.

[67] Radford M. Neal. Markov chain sampling methods for Dirichlet process mixture models. *Journal of Computational and Graphical Statistics*, 9(2):249–265, June 2000.

[68] David J. Newmann and Sharon Block. Probabilistic topic decomposition of an eighteenth-century American newspaper. *Journal of the American Society for Information Sciences and Technology*, 57(6):753–767, February 2006.

[69] K. Nigam, J. Lafferty, and A. McCallum. Using maximum entropy for text classification. In *Proceedings of the IJCAI-99 Workshop on Machine Learning for Information Filtering*, volume 1, pages 61–67, 1999.

[70] Nuance Communications, Inc. OmniPage Pro. `http://www.nuance.com/imaging/products/omnipage.asp`, 2010.

[71] Bo Pang and Lillian Lee. Seeing stars: Exploiting class relationships for sentiment categorization with respect to rating scales. In *Proceedings of the 43rd Annual meeting of the Association for Computational Linguistics*, pages 115–124, Ann Arbor, MI, June 2005. `http://www.cs.cornell.edu/People/pabo/movie-review-data`.

[72] K. Pearson. On lines and planes of closest fit to systems of points in space. *Philosophical Magazine*, 2(6):559–572, 1901.

[73] Ian Porteous, David Newman, Alexander Ihler, Arthur Asuncion, Padhraic Smyth, and Max Welling. Fast collapsed Gibbs sampling for latent Dirichlet allocation. In *Proceedings of the 14th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, pages 569–577, Las Vegas, NV, 2008. ACM.

[74] John W. Pratt, Howard Raïffa, and Robert Schlaifer. *Introduction to Statistical Decision Theory*. MIT Press, 1995. ISBN 9780262161442.

[75] Adrian E. Raftery and Steven M. Lewis. *Implementing MCMC*, chapter 7, pages 115–130. Chapman & Hall, 1996.

[76] Sylvia Richardson and Peter J. Green. On Bayesian analysis of mixtures with an unknown number of components. *Journal of the Royal Statistical Society. Series B*, 59(4):731–792, 1997.

[77] Loïs Rigouste, Olivier Cappé, and François Yvon. Evaluation of a probabilistic method for unsupervised text clustering. In *Proceedings of the International Symposium on Applied Stochastic Models and Data Analysis*, pages 114–123, Best, France, May 2005.

[78] Christian P. Robert. *The Bayesian Choice: from Decision-Theoretic Foundations to Computational Implementation*. Springer Verlag, 2007.

[79] Stephen Roller, Michael Speriosu, Sarat Rallapalli, Benjamin Wing, and Jason Baldridge. Supervised text-based geolocation using language models on an adaptive grid. In *Proceedings of the Joint Conference on Empirical Methods in Natural Language Processing and Computational Language Learning*, pages 1500–1510, Jeju, Korea, July 2012.

[80] Michal Rosen-Zvi, Thomas L. Griffiths, Mark Steyvers, and Padhraic Smyth. The author-topic model for authors and documents. In *Proceedings of the 20th Conference in Uncertainty in Artificial Intelligence*, pages 487–494, Banff, Canada, 2004. AUAI Press.

[81] Andrew Rosenberg and Julia Hirschberg. V-measure: A conditional entropy-based external cluster evaluation measure. In *Proceedings of the Joint Conference on Empirical Methods in Natural Language Processing and Computational Language Learning*, pages 410–420, Prague, June 2007.

[82] Prateek Sarkar, Henry S. Baird, and Xiaoju Zhang. Training on severely degraded text-line images. In *Proceedings of the 7th International Conference on Document Analysis and Recognition*, pages 38–43, Edinburgh, Scotland, August 2003.

[83] Mahdi Shafiei and Evangelos E. Milios. Latent Dirichlet co-clustering. In *Proceedings of the 6th International Conference on Data Mining*, pages 542–551, Washington, DC, 2006. IEEE Computer Society.

[84] Michael Steinbach, George Karypis, and Bipin Kumar. A comparison of document clustering techniques. Technical report, University of Minnesota, May 2000.

[85] Matthew Stephens. Dealing with label switching in mixture models. *Journal of the Royal Statistical Society: Series B*, 62(4):795–809, 2000.

[86] Kazem Taghva, Julie Borsack, and Allen Condit. Results of applying probabilistic IR to OCR text. In *Proceedings of the 17th International ACM/SIGIR Conference on Research and Development in Information Retrieval*, pages 202–211, 1994.

[87] Kazem Taghva, Tom Nartker, Julie Borsack, Steve Lumos, Allen Condit, and Ron Young. Evaluating text categorization in the presence of OCR errors. In *Proceedings of the*

IS&T/SPIE 2001 International Symposium on Electronic Imaging Science and Technology, pages 68–74. SPIE, 2001.

[88] Yee Whye Teh, David Newman, and Max Welling. A collapsed variational Bayesian inference algorithm for latent Dirichlet allocation. In B. Schölkopf, J. Platt, and T. Hoffman, editors, *Advances in Neural Information Processing Systems 19*, volume 19, pages 1353–1360. MIT Press, 2007.

[89] Naonori Ueda and Ryohei Nakano. Deterministic annealing EM algorithm. *Neural Networks*, 11(2):271–282, 1998.

[90] Daniel D. Walker and Eric K. Ringger. New social bookmarking data set. `https://facwiki.cs.byu.edu/nlp/index.php/Data#New_Social_Bookmarking`, October 2007.

[91] Daniel D. Walker and Eric K. Ringger. Model-based document clustering with a collapsed Gibbs sampler. In *Proceedings of the 14th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, pages 704–712, Las Vegas, NV, August 2008.

[92] Daniel D. Walker and Eric K. Ringger. Top N per document: Fast and effective unsupervised feature selection for document clustering. Technical Report 6, Brigham Young University, July 2010. `http://nlp.cs.byu.edu/techreports/BYUNLP-TR6.pdf`.

[93] Daniel D. Walker, William B. Lund, and Eric K. Ringger. Evaluating models of latent document semantics in the presence of OCR errors. In *Proceedings of the Conference on Empirical Methods in Natural Language Processing*, pages 240–250, Cambridge, MA, 2010.

[94] Daniel D. Walker, William B. Lund, and Eric K. Ringger. A synthetic document image dataset for developing and evaluating historical document processing methods. In *Proceedings of Document Recognition and Retrieval XIX*, San Francisco, CA, January 2012.

[95] Daniel D. Walker, Kevin Seppi, and Eric K. Ringger. Topics over nonparametric time: A supervised topic model using Bayesian nonparametric density estimation. In *Proceedings of the 9th Bayesian Modelling Applications Workshop*, Catalina Island, CA, August 2012.

[96] Daniel D. Walker, Kevin Seppi, and Eric K. Ringger. Evaluating supervised topic models in the presence of OCR errors. In *Proceedings of Document Recognition and Retrieval XX, to appear*, San Fancisco, CA, February 2013.

[97] Hanna M. Wallach, David Mimno, and Andrew McCallum. Rethinking LDA: Why priors matter. In Y. Bengio, D. Schuurmans, J. Lafferty, C. K. I. Williams, and A. Culotta, editors,

Advances in Neural Information Processing Systems 22, volume 22, pages 1973–1981. MIT Press, 2009.

[98] Hanna M. Wallach, Iain Murray, Ruslan Salakhutdinov, and David Mimno. Evaluation methods for topic models. In *Proceedings of the 26th Annual International Conference on Machine Learning*, pages 1105–1112, Montreal, Canada, June 2009.

[99] Xuerui Wang and Andrew McCallum. Topics over time: A non-Markov continuous-time model of topical trends. In *Proceedings of the 12th ACM SIGKDD International Conference*, pages 424–433, Philadelphia, PA, August 2006.

[100] Xuerui Wang, Andrew McCallum, and Xing Wei. Topical n-grams: Phrase and topic discovery, with an application to information retrieval. In *Proceedings of the 7th IEEE International Conference on Data Mining*, pages 697–702, Omaha, NE, 2007. IEEE Computer Society.

[101] Yi Wang, Hongjie Bai, Matt Stanton, Wen-Yen Chen, and Edward Y. Chang. PLDA: Parallel latent Dirichlet allocation for large-scale applications. In Andrew Goldber and Yunhong Zhou, editors, *Algorithmic Aspects in Information and Management*, volume 5564 of *Lecture Notes in Computer Science*, pages 301–314. Springer Berlin / Heidelberg, 2009.

[102] Michael L. Wick, Michael G. Ross, and Erik G. Learned-Miller. Context-sensitive error correction: Using topic models to improve OCR. In *Proceedings of the 9th International Conference on Document Analysis and Recognition*, pages 1168–1172, Curitiba, Brazil, 2007.

[103] Yiming Yang and Jan O. Pedersen. A comparative study on feature selection in text categorization. In *Proceedings of the 14th International Conference on Machine Learning*, pages 412–420, Nashville, TN, July 1997.

[104] Shipeng Yu. *Advanced Probabilistic Models for Clustering and Projection*. PhD thesis, Fakultät für Mathematik, Informatik und Statistik der Ludwig-Maximilians-Universität München, 2006.

[105] Jian Zhang, Zoubin Ghahramani, and Yiming Yang. A probabilistic model for online document clustering with application to novelty detection. In Lawrence K. Saul, Yair Weiss, and Léon Bottou, editors, *Advances in Neural Information Processing Systems 17*, volume 17, pages 1617–1624. MIT Press, Cambridge, MA, 2005.

[106] Shi Zhong and Joydeep Ghosh. Generative model-based document clustering: a comparative study. *Knowledge and Information Systems*, 8(3):374–384, 2005.