LATENT DIRICHLET ALLOCATION: HYPERPARAMETER SELECTION AND APPLICATIONS TO ELECTRONIC DISCOVERY

By

CLINT PAZHAYIDAM GEORGE

A DISSERTATION PRESENTED TO THE GRADUATE SCHOOL
OF THE UNIVERSITY OF FLORIDA IN PARTIAL FULFILLMENT
OF THE REQUIREMENTS FOR THE DEGREE OF
DOCTOR OF PHILOSOPHY

UNIVERSITY OF FLORIDA

2015

© 2015 Clint Pazhayidam George

To
my soul mate Dhanya,
my parents Gracy & George Pazhayidam,
my grandparents Ely & Thomas Pazhayidam, Rosamma & Mathew Kizhakkaalayil,
my great-grandparents Ely & Varkey Pazhayidam

ACKNOWLEDGMENTS

Let me thank all who helped me to complete my Ph.D. journey. First of all, I would like to thank Dr. Joseph N. Wilson for the boundless support, tremendous patience and motivation that he provided for my research from the start of my graduate study at the University of Florida. His valuable comments and critics helped me throughout my research and have improved my writing.

I would like to express my sincere thanks to Dr. Hani Doss for all the insightful comments and advice on my research. He has been a great teacher for me on the principles of statistical learning, Markov chain Monte Carlo methods, and statistical inference. I am extremely grateful for his immense patience in reading my manuscripts, and his valuable ideas for completing this dissertation.

I would like to thank Dr. Daisy Zhe Wang for involving me in the Data Science Research team's weekly meetings and the SurveyMonkey and UF Law E-Discovery project. Her valuable suggestions helped me expand my knowledge in applied machine learning research. I would like to convey my thanks to Dr. Sanjay Ranka, Dr. Anand Rangarajan, and Dr. Rick L. Smith, for being part of my Ph.D. committee, and for their insightful comments and continuous encouragement. I am very fortunate to have those hard questions that helped me to think differently.

I would like to thank Prof. William Hamilton for his valuable support for the UF Law E-Discovery project from the beginning. His suggestions helped me to think from a Lawyer's perspective during the project design. I would like to express my sincere gratitude to Dr. Paul Gader and Dr. George Casella (late) for the lessons on machine learning and statistical inference, and motivations to continue research in machine learning during the early days of my research.

I would like to acknowledge the generous financial contributions from SurveyMonkey and ICAIR (The International Center for Automated Research at the University of Florida Levin College of Law) for my Ph.D. research.

I thank Christan Grant, Peter Dobbins, Zhe Chen, Manu Sethi, Brandon Smock, Taylor Glenn, Claudio Fuentes, Sean Goldberg, Sahil Puri, Srinivas Balaji, Abhiram Jagarlapudi, Chris Jenneisch, and all of my colleagues in the Data Science Research lab, for the fruitful discussions and comments on research. I thank Manu Chandran, Manu Nandan, Asish Skaria, Joseph Thalakkattoor, Paul Thottakkara, Kiran Lukose, Kavya Nair, Jay Nair, and all my friends in the Gainesville, who have made Gainesville a second home for me.

Last but not the least, I am grateful to have Dhanya, who joined my life during the toughest times of my research. I am thankful for all her encouragements to complete this journey. I also would like to thank my parents, Gracy and George, my sisters, Christa and Chris, my grandparents, Thomas (Chachan), Ely (Amma), and Rosamma (Ammachi), my in-laws, Renjith, Albin, Naveen, Evelyn, Rosamma (Amma), Joseph (Acha), and all of my relatives, for their infinite support, encouragement, and patience, during this time.

TABLE OF CONTENTS

page

ACKNOWLEDGMENTS 4

LIST OF TABLES 8

LIST OF FIGURES 9

ABSTRACT 11

CHAPTER

1 DISSERTATION OVERVIEW 13

2 THE LATENT DIRICHLET ALLOCATION MODEL: INTRODUCTION AND IMPORTANCE OF SELECTING HYPERPARAMETERS 19

3 ESTIMATION OF THE MARGINAL LIKELIHOOD UP TO A MULTIPLICATIVE CONSTANT AND ESTIMATION OF POSTERIOR EXPECTATIONS 25

3.1 Estimation of the Marginal Likelihood up to a Multiplicative Constant . . 25
3.2 Estimation of the Family of Posterior Expectations 27
3.3 Serial Tempering 29
3.4 Illustration on Low-Dimensional Examples 34

4 TWO MARKOV CHAINS ON (β, θ, z) 40

4.1 The Conditional Distributions of (β, θ) Given z and of z Given (β, θ) . . . 41
4.2 Comparison of the Full Gibbs Sampler and the Augmented Collapsed Gibbs Sampler 42

5 PERFORMANCE OF THE LDA MODEL BASED ON THE EMPIRICAL BAYES CHOICE OF h 48

5.1 Other Hyperparameter Selection Methods and Criteria for Evaluation . . . 48
5.2 Comparison on Real Datasets 53

6 ELECTRONIC DISCOVERY: INTRODUCTION 67

7 APPLYING TOPIC MODELS TO ELECTRONIC DISCOVERY 73

7.1 System Design and Methods 73
7.2 Experiments and Analysis of Results 80
7.3 Summary and Discussion 91

8 SELECTING THE NUMBER OF TOPICS IN THE LATENT DIRICHLET ALLOCATION MODEL: A SURVEY 97

8.1 Selecting K Based on Marginal Likelihood 97

8.2 Selecting K Based on Predictive Power 99
8.3 Selecting K Based on Human Readability 100
8.4 Hierarchical Dirichlet Processes 103
8.5 Summary 104

9 CONCLUSIONS 106

APPENDIX

A A NOTE ON BLEI ET AL. (2003)'S APPROACH FOR INFERENCE AND PARAMETER ESTIMATION IN THE LDA MODEL 108

B EVALUATION METHODS FOR ELECTRONIC DISCOVERY 112

B.1 Recall and Precision 112
B.2 Receiver Operating Characteristic 112

REFERENCES 116

BIOGRAPHICAL SKETCH 121

LIST OF TABLES

Table page

5-1 Corpora created from the 20Newsgroups dataset and the Wikipedia pages. . . . 55

5-2 Sorted values of the averages of the $\binom{K}{2}$ L_2 distances $\|\beta_j^{\text{true}} - \beta_{j'}^{\text{true}}\|_2$, $j, j' = 1, \ldots, K$, for the nine corpora. 56

5-3 L_2 distances between the default hyperparameter choices h_{DR}, h_{DA}, and h_{DG}, and the empirical Bayes choice $\hat{\hat{h}}$, for the nine corpora. 61

5-4 Estimates of the discrepancy ratios $D(h_{\text{DR}}) := \rho_2(\pi_{\text{DR}}, \delta_{\boldsymbol{\theta}^{\text{true}}})/\rho_2(\pi_{\text{EB}}, \delta_{\boldsymbol{\theta}^{\text{true}}})$, $D(h_{\text{DA}}) := \rho_2(\pi_{\text{DA}}, \delta_{\boldsymbol{\theta}^{\text{true}}})/\rho_2(\pi_{\text{EB}}, \delta_{\boldsymbol{\theta}^{\text{true}}})$, and $D(h_{\text{DG}}) := \rho_2(\pi_{\text{DG}}, \delta_{\boldsymbol{\theta}^{\text{true}}})/\rho_2(\pi_{\text{EB}}, \delta_{\boldsymbol{\theta}^{\text{true}}})$, for all nine corpora, where $h_{\text{DR}} = (1/K, 1/K)$, $h_{\text{DA}} = (.1, .1)$, and $h_{\text{DG}} = (.1, 50/K)$. The discrepancy is smallest for the empirical Bayes model, uniformly across all nine corpora. 62

5-5 Ratios of the estimates of posterior predictive scores of the LDA models indexed by default hyperparameters h_{DR}, h_{DA}, and h_{DG} to the estimate of the posterior predictive score of the empirical Bayes model, for all nine corpora. 63

7-1 Corpora created from the TREC-2010 Legal Track topic datasets. 82

7-2 Corpora created from the 20Newsgroups dataset to evaluate various seed selection methods. 83

7-3 Corpora created from the 20Newsgroups dataset to evaluate various classifiers. . 83

7-4 Performance of various classification models using the features derived from the methods LDA, LSA, and TF-IDF for corpora C-Mideast, C-IBM-PC, C-Motorcycles, and C-Baseball-2. 89

7-5 Running times of various classification models using the features derived from the methods LDA, LSA, and TF-IDF for different corpora. 90

B-1 ROC Dataset: Classification output for 10 data points from two hypothetical classifiers. 114

LIST OF FIGURES

Figure page

2-1 Estimate of the posterior probability that $\|\theta_1 - \theta_2\| \leq 0.07$ for a synthetic corpus of documents. The posterior probability varies considerably with h. 22

3-1 Comparison of the variability of $\hat{I}_\zeta^{\text{st}}$ and $\tilde{I}_\zeta^{\text{st}}$. Each of the top two panels shows two independent estimates of $I(\alpha, \eta)$, using $\hat{I}_\zeta^{\text{st}}(\alpha, \eta)$. For the left panel, $\eta = .35$, and for the right panel, $\eta = .45$. Here, $I(h)$ is the posterior probability that $\|\theta_1 - \theta_2\| < 0.07$ when the prior is ν_h. The bottom two panels use $\tilde{I}_\zeta^{\text{st}}$ instead of $\tilde{I}_\zeta^{\text{st}}$. The superiority of $\hat{I}_\zeta^{\text{st}}$ over $\tilde{I}_\zeta^{\text{st}}$ is striking. 35

3-2 Neighborhood structures for interior, edge, and corner points in a 4×4 grid for the serial tempering chain. 37

3-3 $\widehat{M}(h)$ and MCSE of $\widehat{M}(h)$ for four values of h_{true}. In each case, $\hat{\hat{h}}$ is close to h_{true}. 38

3-4 $\widetilde{M}(h)$ and MCSE of $\widetilde{M}(h)$ for four specifications of h_{true}. 39

4-1 Histograms of the p-values over all the words in all the documents, for each setting of the hyperparameter. 44

4-2 Q-Q plots for the p-values over all the words in all the documents, for four hyperparameter settings. The plots compare the empirical quantiles of the p-values with the quantiles of the uniform distribution on $(0, 1)$. 45

4-3 Log posterior trace plots (top) and autocorrelation function (bottom) plots of the Full Gibbs Sampler and the Augmented Collapsed Gibbs Sampler, for the hyperparameter $h = (3, 3)$. 46

4-4 Autocorrelation functions for selected elements of the $\boldsymbol{\theta}$ and $\boldsymbol{\beta}$ vectors for the Full Gibbs Sampler and the Augmented Collapsed Gibbs Sampler, for the hyperparameter $h = (3, 3)$. 47

5-1 Plots of L_2 norms between the true topic distributions, for all nine corpora. . . . 57

5-2 Plots of $\hat{M}(h)$ for the five 20Newsgroups corpora. 58

5-3 Monte Carlo standard error (MCSE) of $\hat{M}(h)$ for the five 20Newsgroups corpora. 59

5-4 Plots of $\hat{M}(h)$ for corpora C-6, C-7, C-8, and C-9. 60

5-5 Monte Carlo standard error (MCSE) of $\hat{M}(h)$ for corpora C-6, C-7, C-8, and C-9. 61

5-6 Plots of the number of iterations (in units of 100) that the final serial tempering chain spent at each of the hyperparameter values $h_1, \ldots, h_J$ in the subgrid, for corpora C-1–C-5. 65

5-7 Plots of the number of iterations (in units of 100) that the final serial tempering chain spent at each of the hyperparameter values $h_1, \ldots, h_J$ in the subgrid, for corpora C-6–C-9. 66

6-1 Technology Assisted Review Cycle 68

6-2 Computer Assisted Review Model (EDRM, 2009) 71

7-1 SMART e-discovery Retrieval work-flow: Starred numbers represent each step in the work-flow. 74

7-2 ROC curve analysis of various ranking models for corpora C-201 and C-202. . . 85

7-3 ROC curve analysis of various ranking models for corpora C-203 and C-207. . . 86

7-4 Classification performance of various seed selection methods for corpora C-Medicine and C-Baseball. We used the document semantic features (200) generated via the Latent Semantic Analysis algorithm for classifier training and prediction runs 93

7-5 Classification performance of various seed selection methods for corpora C-Medicine and C-Baseball. We used the document topic features (50) generated via the Latent Dirichlet Allocation algorithm for classifier training and prediction runs. 94

7-6 Classification performance of various SVM models (based on document topic mixtures and Whoosh scores) vs. Whoosh retrieval for corpora C-201 and C-202. 95

7-7 Classification performance of various SVM models (based on document topic mixtures and Whoosh scores) vs. Whoosh retrieval for corpora C-203 and C-207. 96

B-1 Recall and Precision 113

B-2 Plots of ROC curves that compares the output of two hypothetical classifiers described in Table B-1. 115

Abstract of Dissertation Presented to the Graduate School
of the University of Florida in Partial Fulfillment of the
Requirements for the Degree of Doctor of Philosophy

LATENT DIRICHLET ALLOCATION: HYPERPARAMETER SELECTION AND APPLICATIONS TO ELECTRONIC DISCOVERY

By

Clint Pazhayidam George

December 2015

Chair: Joseph N. Wilson
Cochair: Hani Doss
Major: Computer Engineering

Keyword-based search is a popular information retrieval scheme to discover relevant documents from a document collection, but it has many shortcomings. Concept or topic search is an alternative to keyword-based search that can address some of these deficiencies, and better categorize documents based on their underlying topics. Latent Dirichlet Allocation (LDA) is a popular topic model that is often used to make inference regarding the properties of a corpus. LDA is a hierarchical Bayesian model that involves a prior distribution on a set of latent topic variables. The prior is indexed by certain hyperparameters which have a considerable impact on inference but are usually chosen either in an ad-hoc manner or by applying an algorithm whose theoretical basis has not been firmly established. We present a method, based on a combination of Markov chain Monte Carlo and importance sampling, for obtaining the maximum likelihood estimate (MLE) of the hyperparameters. We report the results of experiments on both synthetic and real data. These show that when making inference regarding the topics of the documents in a corpus, the LDA model indexed by the MLE of the hyperparameters performs considerably better than LDA models indexed by default choices of the hyperparameters. Topic models such as LDA have many real-world applications such as document clustering, classification, and ranking and summarizing a corpus. In this thesis, we employ various topic models to the electronic discovery (e-discovery) problem, which

refers to the process of identifying, collecting, discovering, and managing electronically stored information (ESI) for a lawsuit. We perform an empirical study comparing the performance of LDA to other topic models in representing ESI and building binary classification models to solve the document discovery problem of e-discovery. We report the performance of this study using several real datasets.

CHAPTER 1
DISSERTATION OVERVIEW

A corpus is a collection of documents. The vocabulary of a corpus is the set of unique words in the corpus. In general, a topic[1] is the subject or theme of a speech, essay, article, or discourse. One can formally define a topic as a distribution on the vocabulary. For example, the topic *sports* has words about sports, e.g., *football*, *soccer*, etc., with high probability. Topic models are often used to make inference regarding the underlying thematic (or topic) structure of a corpus. Latent Dirichlet allocation (LDA, Blei et al. 2003) is a popular topic model that assumes that a topic is a latent (hidden) distribution on the vocabulary and each document in the corpus is described by a latent mixture of topics. LDA is a hierarchical Bayesian model that involves a prior distribution on the latent topic variables. The prior is indexed by certain hyperparameters, which even though they have a major impact on inference, are often chosen in ad-hoc manner. This dissertation presents a principled scheme for selecting the hyperparameters based on a combination of Markov chain Monte Carlo and importance sampling. This dissertation also gives an introduction to the electronic discovery (e-discovery) problem, which is a sub-problem of information retrieval, and describes an empirical study comparing the performance of LDA to several other document modeling schemes that have been employed to model e-discovery corpora. What follows is a general introduction to the dissertation problem, a set of goals, and an overview of our approach to achieving our goals.

Consider a typical information retrieval problem. Suppose we have a system that uses keyword comparisons to find documents in a corpus related to a user's search keywords. Some relevant documents may not contain the exact keywords specified by the user. For example, the keyword *computers* may miss the documents that contain words such as *PC*,

[1] http://dictionary.reference.com/browse/topic

laptop, *desktop*, etc. and do not have the word *computers*. The reason for this keyword search failure can be *synonymy* or *polysemy* of words that appear in a corpus. Synonymy refers to words or phrases that have similar meanings, e.g., *car* and *automobile*, *hood* and *bonnet*. This leads to poor *recall* in information retrieval. Polysemy refers to words that have more than one distinct meaning, e.g., the meaning of the word *chair* in phrases *the chair maker* and *the chair of the department*. Polysemy may lead to poor *precision* in information retrieval. Appendix B.1 gives a formal definition for recall and precision. One popular alternative to keyword-based retrieval is *concept* search or *topic* search, which can overcome some of these issues.

We now give an overview of how a typical information retrieval is performed. The major task is to to represent entities (i.e. search keywords and documents) in an indexing space where each distinct entity lies as far away from each other as possible. Incoming keyword queries are then compared with stored or *indexed* text documents. A vector space model (VSM, Salton et al. 1975) is an indexing generated by a method in which one converts a corpus that consists of D documents and V vocabulary terms into a *term-document* matrix—a.k.a. *term-frequency* (TF) matrix—as follows. For $d = 1, 2, \ldots, D$, document d has a column $\mathbf{c}_d = (\mathrm{tf}_{d1}, \mathrm{tf}_{d2}, \ldots, \mathrm{tf}_{dt}, \ldots, \mathrm{tf}_{dV})$ in the matrix, where tf_{dt} represents the frequency of the vocabulary term t in document d. This matrix is then translated into vectors in a vector space, where one vector is assigned to each document in the corpus. One can then consider a user's keyword query as a document in the corpus and easily map it to a vector in the vector space. Finally, a similarity score, e.g., cosine, between the query vector and document vectors can be used to rank the documents on relevance to a query. The *term-frequency inverse-document-frequency* (TF-IDF, Jones 1972) is a special type of VSM, which has an *inverse-document-frequency* (IDF) for each term t in document d. This IDF term helps to handle commonly occurring words in the corpus. Although these models provide an elegant algebraic framework to represent documents and keyword-queries and perform reasonably quick keyword-based document

retrieval, they suffer from issues such as word synonymy and polysemy. Synonymy can yield a small cosine similarity for two related document vectors. Polysemy can yield a large cosine similarity for two unrelated document vectors. In addition, the dimension of the TF or TF-IDF matrix increases with the size of a corpus, which can cause intractable computational and space complexities.

Latent Semantic Indexing[2] (LSI, see, e.g., Dumais et al. 1995) is another document modeling method, which can handle synonymy. The LSI method typically performs matrix factorization over the TF-IDF matrix of a corpus using the concepts of Singular Value Decomposition, and identifies patterns in the relationships between document terms and concepts. It can group together words and phrases that have similar meanings (George et al., 2012). As an alternative for the TF-IDF or TF approaches, one can use the identified groups or concepts to represent the documents in a corpus and keyword queries. By defining a similarity score on the new representative domain, one can perform a *concept* search to retrieve relevant documents. Hoenkamp (2011) found that LSI can produce inconsistent grouping for independent but identical noise samples. The samples were created by adding random noise (using a uniform distribution) to the TF-IDF matrix of a corpus. In addition, being a linear model, it is unlikely that LSI will identify nonlinear relationships between documents and words in a corpus. LSI also lacks the rewards of a probabilistic model, e.g., generalization ability of the model to include newly encountered documents.

Probabilistic topic modeling (e.g. LDA), the major focus in this dissertation, allows us to represent the properties of a corpus with a small collection of topics, far fewer than the vocabulary size of a corpus. It is also known to be a method to handle both polysemy and synonymy. The popular topic model, LDA, is most easily described by its generative processes, the random process by which the model assumes the documents are created.

[2] It is also known as Latent Semantic Analysis (LSA).

It assumes that the corpus has a vocabulary of words, and each document in the corpus is described by a distribution of topics. A topic is represented by a distribution of words in the vocabulary. A topic (index) is assigned to each word in a document based on the document specific topic distribution, and each observed word in a document is chosen from the word's topic distribution. The parameters (i.e. corpus level topic distributions, document level distribution of topics, and words' topic variables) of the model are latent. One can infer the values of these latent variables via posterior inference, which typically computes the posterior distribution of latent variables conditional on the data (Blei, 2004). Exact posterior inference is intractable in sophisticated models such as LDA and practical data analysis depends on approximate alternatives (Blei et al., 2003; Griffiths and Steyvers, 2004).

LDA is a hierarchical Bayesian model that involves a prior distribution on a set of latent topic variables. The prior is indexed by hyperparameters that, even though they have a large impact on inference, are usually chosen either in an ad-hoc manner or by applying an algorithm whose theoretical basis has not been firmly established. Chapter 2 formally defines the latent Dirichlet allocation model and describes the importance of selecting hyperparameters in the model. In this thesis, we describe a method based on a combination of Markov chain Monte Carlo and importance sampling, to obtain the maximum likelihood estimate (MLE) of the hyperparameters (Chapter 3). The method may be viewed as a computational scheme for implementation of an empirical Bayes analysis. We report the results of experiments on both synthetic and real data (Chapter 5). We also describe two Markov chains whose stationary distribution is the LDA posterior and compare their performance empirically, based on synthetically generated datasets from the LDA model (Chapter 4).

The LDA model is also indexed by a constant K that represents the number of topics in the corpus of interest. It is assumed to be known in advance. In the machine learning

literature, people have looked into the problem of choosing K from the data itself. We give an overview of such methods in Chapter 8.

E-discovery refers to a process in which one identifies, collects, and evaluates electronically stored information (ESI) as part of a legal case or lawsuit. In this dissertation, we are interested in the document discovery or information retrieval part of the e-discovery process. In document discovery, one's goal is to retrieve all documents that are potentially relevant to issues and facts of a legal case, from the ESI identified for that case. This dissertation gives a brief overview of the e-discovery process (Chapter 6) and describes a study comparing the performance of LDA to several other document modeling schemes such as TF-IDF and LSI that have been employed to model ESI (Chapter 7). One approach to relevant document discovery is to build a classifier to classify relevant and non-relevant documents for an e-discovery request and assign class confidence values to individual documents for ranking. We consider representing documents both in the topic space (via LSI or LDA) and in the vocabulary space (via TF-IDF) to define document-document similarities and query-document similarities. One can also look at these document modeling methods as feature engineering schemes to build classifiers. We also describe an iterative ranking and classification work-flow including human-in-the-loop labeling of seed (training) documents and using them to build an iterative document classification model based on Support Vector Machines (Cortes and Vapnik, 1995). To improve this model, we propose several seed selection methods and illustrate the application of these methods using real datasets in the electronic discovery domain.

This dissertation describes three loosely connected topics: (i) principled selection of hyperparameters in the LDA model, (ii) an empirical study of employing LDA to the e-discovery problem, and (iii) a survey of the methods used in the literature to identify the number of topics in a corpus. The topics are organized as follows. Chapters 2, 3, 4, and 5 discuss the first topic (i). Chapters 6 and 7 describe the second topic (ii). In

Chapter 8, we describe the third topic (iii). Finally, Chapter 9 summarizes the results of this dissertation research and points out areas for future research.

CHAPTER 2
THE LATENT DIRICHLET ALLOCATION MODEL: INTRODUCTION AND IMPORTANCE OF SELECTING HYPERPARAMETERS

Latent Dirichlet Allocation (LDA, Blei et al. 2003) is a model that is used to describe high-dimensional sparse count data represented by feature counts. Although the model can be applied to many different kinds of data, for example collections of annotated images and social networks, for the sake of concreteness, here we focus on data consisting of a collection of documents. Suppose we have a corpus of documents, say a collection of news articles, and these span several different topics, such as sports, medicine, politics, etc. We imagine that for each word in each document, there is a latent (i.e. unobserved) variable indicating a topic from which that word is drawn. We have two goals: (i) we want to make inference on the latent topic variables for each document, and (ii) we want to cluster together documents which are similar, i.e. documents which share common topics.

To describe the LDA model, we first set up some terminology and notation. There is a vocabulary $\mathcal{V}$ of V words; typically, this is taken to be the union of all the words in all the documents of the corpus, after removing stop (i.e. uninformative) words. There are D documents in the corpus, and for $d = 1, \ldots, D$, document d has n_d words, $w_{d1}, \ldots, w_{dn_d}$. The order of the words is considered uninformative, and so is neglected. Each word is represented as an index $1 \times V$ vector with a 1 at the s^{th} element, where s denotes the term selected from the vocabulary. Thus, document d is represented by the vector $\boldsymbol{w}_d = (w_{d1}, \ldots, w_{dn_d})$ and the corpus is represented by the vector $\boldsymbol{w} = (\boldsymbol{w}_1, \ldots, \boldsymbol{w}_D)$. The number of topics, K, is finite and known. By definition, a topic is a distribution over $\mathcal{V}$, i.e. a point in $\mathbb{S}_V$, the V-dimensional simplex. For $d = 1, \ldots, D$, for each word w_{di}, z_{di} is an index $1 \times K$ vector which represents the latent variable that denotes the topic from which w_{di} is drawn. The distribution of $z_{d1}, \ldots, z_{dn_d}$ will depend on a document-specific variable θ_d which indicates a distribution on the topics for document d.

We will use $\text{Dir}_L(a_1, \ldots, a_L)$ to denote the finite-dimensional Dirichlet distribution on the L-dimensional simplex. Also, we will use $\text{Mult}_L(b_1, \ldots, b_L)$ to denote the multinomial

distribution with number of trials equal to 1 and probability vector $(b_1, \ldots, b_L)$. We will form a $K \times V$ matrix $\boldsymbol{\beta}$, whose t^{th} row is the t^{th} topic (how $\boldsymbol{\beta}$ is formed will be described shortly). Thus, $\boldsymbol{\beta}$ will consist of vectors $\beta_1, \ldots, \beta_K$, all lying in $\mathbb{S}_V$. Formally, LDA is described by the following hierarchical model, in which $\eta \in (0, \infty)$ and $\boldsymbol{\alpha} \in (0, \infty)^K$ are hyperparameters:

1. $\beta_t \overset{\text{iid}}{\sim} \text{Dir}_V(\eta, \ldots, \eta)$, $t = 1, \ldots, K$.
2. $\theta_d \overset{\text{iid}}{\sim} \text{Dir}_K(\boldsymbol{\alpha})$, $d = 1, \ldots, D$, and the θ_d's are independent of the β_t's.
3. Given $\theta_1, \ldots, \theta_D$, $z_{di} \overset{\text{iid}}{\sim} \text{Mult}_K(\theta_d)$, $i = 1, \ldots, n_d$, $d = 1, \ldots, D$, and the D vectors $(z_{11}, \ldots, z_{1n_1}), \ldots, (z_{D1}, \ldots, z_{Dn_D})$ are independent.
4. Given $\boldsymbol{\beta}$ and the z_{di}'s, w_{di} are independently drawn from the row of $\boldsymbol{\beta}$ indicated by z_{di}, $i = 1, \ldots, n_d$, $d = 1, \ldots, D$.

From the description of the model, we see that there is a latent topic variable for every word that appears in the corpus. Thus it is possible that a document spans several topics. However, because there is a single θ_d for document d, the model encourages different words in the same document to have the same topic. Also note that the hierarchical nature of LDA encourages different documents to share the same topics. This is because $\boldsymbol{\beta}$ is chosen once, at the top of the hierarchy, and is shared among the D documents.

Let $\boldsymbol{\theta} = (\theta_1, \ldots, \theta_D)$, $\boldsymbol{z}_d = (z_{d1}, \ldots, z_{dn_d})$ for $d = 1, \ldots, D$, $\boldsymbol{z} = (\boldsymbol{z}_1, \ldots, \boldsymbol{z}_D)$, and let $\boldsymbol{\psi} = (\boldsymbol{\beta}, \boldsymbol{\theta}, \boldsymbol{z})$. The model is indexed by the hyperparameter vector $h = (\eta, \boldsymbol{\alpha}) \in (0, \infty)^{K+1}$. For any given h, lines 1–3 induce a prior distribution on $\boldsymbol{\psi}$, which we will denote by ν_h. Line 4 gives the likelihood. The words $\boldsymbol{w}$ are observed, and we are interested in $\nu_{h,\boldsymbol{w}}$, the posterior distribution of $\boldsymbol{\psi}$ given $\boldsymbol{w}$ corresponding to ν_h. (Note: In step 1, the distribution of β_t is a symmetric Dirichlet, indexed by a one-dimensional parameter η. We do not use a Dirichlet indexed by an arbitrary vector $\boldsymbol{\eta} \in (0, \infty)^V$ because the resulting high dimension of h would be problematic (Wallach et al., 2009a).)

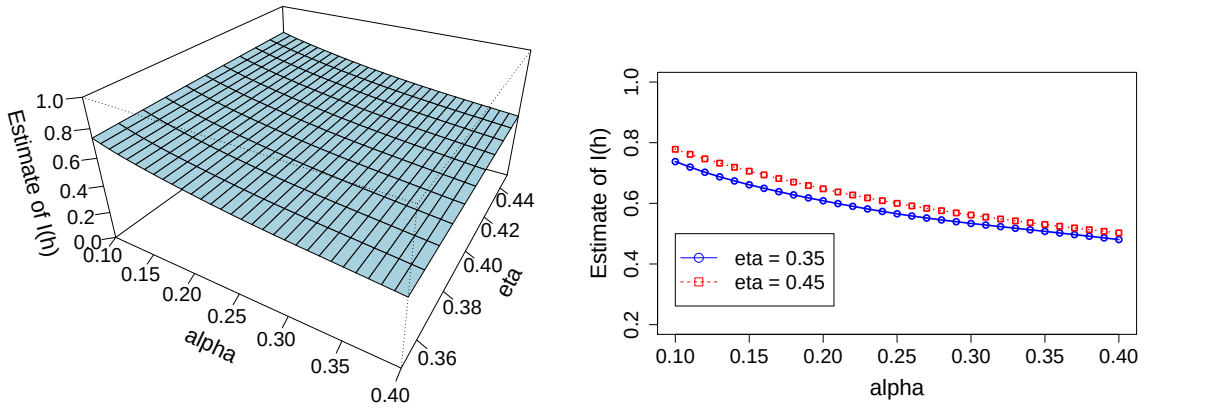
The hyperparameter h is not random, and must be selected in advance. It has a strong effect on the distribution of the parameters of the model. For example, when η

is large, the topics tend to be probability vectors which spread their mass evenly among many words in the vocabulary, whereas when η is small, the topics tend to put most of their mass on only a few words. Also, in the special case where $\boldsymbol{\alpha} = (\alpha, \ldots, \alpha)$, so that $\text{Dir}_K(\boldsymbol{\alpha})$ is a symmetric Dirichlet indexed by the single parameter α, when α is large, each document tends to involve many different topics; on the other hand, in the limiting case where $\alpha \to 0$, each document involves a single topic, and this topic is randomly chosen from the set of all topics.

As indicated above, the hyperparameter h plays a critical role, and its value has an important impact on inference. To demonstrate this empirically, we generated a synthetic corpus of $D = 20$ documents, with document d having $n_d = 200$ words (for $d = 1, \ldots, D$), drawn from a vocabulary of size $V = 40$, using an LDA model with number of topics $K = 5$ and hyperparameter vector $h = (\eta, \alpha) = (0.4, 0.2)$ (we are using a symmetric Dirichlet with a single parameter α in line 2 of the model). A typical question of interest is whether the topics for two given documents are nearly the same. One way to word this question precisely is to ask what is the posterior probability that $\|\theta_i - \theta_j\| \leq \epsilon$, where i and j are the indices of the documents in question and ϵ is some user-specified small number. Here, $\|\cdot\|$ denotes ordinary Euclidean distance. This posterior probability will of course depend on the value of h that is used to fit the LDA model. Let $I(h)$ denote this posterior probability. Figure 2-1A gives a plot of an estimate $\hat{I}(h)$ of $I(h)$ for documents 1 and 2 and $\epsilon = 0.07$, as h varies over the region $(\eta, \alpha) \in (0.35, 0.45) \times (0.1, 0.4)$ in a 11×31 grid of 341 values. (The plot was created by a Markov chain Monte Carlo (MCMC) scheme, described in Section 3, under which it was not necessary to run 341 separate Markov chains to estimate the 341 posterior probabilities.[1]) Figure 2-1B shows line plots of $\hat{I}(h)$

[1] Software for implementation of all algorithms and datasets discussed in chapters two through five is available as an R package at: https://github.com/clintpgeorge/ldamcmc

for the same document pair and ϵ, as α varies over the range $(0.1, 0.4)$ and η is equal to 0.35 and 0.45. As can be seen from the plots, the estimated posterior probability varies considerably as α varies (varying η has little effect on $\hat{I}(h)$): $\hat{I}(h)$ has a maximum value of 0.78, which occurs when α is small, and a minimum value of 0.47, which occurs when α is large.



A Plot of $\hat{I}(h)$ as both α and η vary

B Plot of $\hat{I}(h)$ as α varies and η is fixed at .35 and .45

Figure 2-1. Estimate of the posterior probability that $\|\theta_1 - \theta_2\| \leq 0.07$ for a synthetic corpus of documents. The posterior probability varies considerably with h.

To summarize: The hyperparameter h has a strong effect on the prior distribution of the parameters in the model, and Figure 2-1 shows that it also has a strong effect on the posterior distribution of these parameters; therefore it is important to choose it carefully. Yet in spite of the very widespread use of LDA, there is no method for choosing the hyperparameter that has a firm theoretical basis. In the literature, h is sometimes selected in some ad-hoc or arbitrary manner. A principled way of selecting it is via maximum likelihood: we let $m_{\boldsymbol{w}}(h)$ denote the marginal likelihood of the data as a function of h, and use $\hat{h} = \arg\max_h m_{\boldsymbol{w}}(h)$ which is, by definition, the empirical Bayes choice of h. We will write $m(h)$ instead of $m_{\boldsymbol{w}}(h)$ unless we need to emphasize the dependence on $\boldsymbol{w}$. Unfortunately, the function $m(h)$ is analytically intractable: $m(h)$ is the likelihood of the data with all latent variables integrated or summed out, and from the hierarchical nature of the model, we see that $m(h)$ is a high-dimensional integral of large products of large

sums. Blei et al. (2003) propose estimating $\arg\max_h m(h)$ via a combination of the EM algorithm and "variational inference." Very briefly, $\boldsymbol{w}$ is viewed as "observed data," and $\boldsymbol{\psi}$ is viewed as "missing data." Because the "complete data likelihood" $p_h(\boldsymbol{\psi}, \boldsymbol{w})$ is available, the EM algorithm is a natural candidate for estimating $\arg\max_h m(h)$, since $m(h)$ is the "incomplete data likelihood." But the E-step in the algorithm is infeasible because it requires calculating an expectation with respect to the intractable distribution $\nu_{h,\boldsymbol{w}}$. Blei et al. (2003) substitute an approximation to this expectation. Unfortunately, because there are no useful bounds on the approximation, and because the approximation is used at every iteration of the algorithm, there are no results regarding the theoretical properties of this method. The method and its implementation are discussed further in Section 5.1.

Another approach for dealing with the problem of having to make a choice of the hyperparameters is the fully Bayes approach, in which we simply put a prior on the hyperparameters, that is, add one layer to the hierarchical model. For example, we can either put a flat prior on each of $\alpha_1, \ldots, \alpha_K$ and η, or put a gamma prior instead. While this approach can be useful, there are reasons why one may want to avoid it. On the one hand, if we put a flat prior then one problem is that we are effectively skewing the results towards large values of the hyperparameter. A more serious problem is that the posterior may be improper. In this case, insidiously, if we use Gibbs sampling to estimate the posterior, it is possible that all conditionals needed to implement the sampler are proper; but Hobert and Casella (1996) have shown that the Gibbs sampler output may not give a clue that there is a problem. On the other hand, if we use a gamma prior, then we need to specify the gamma hyperparameters, so we're back to the same problem of having to specify hyperparameters. Another reason to avoid the fully Bayes approach is that, in broad terms, the general interest in empirical Bayes methods arises in part from a desire to select specific values of the hyperparameters because these give a model that is more parsimonious and interpretable. This point is discussed more fully (in a general context) in George and Foster (2000) and Robert (2001, Chapter 7).

In the present thesis we show that while it is not possible to compute $m(h)$ itself, it is nevertheless possible, via MCMC, to estimate the function $m(h)$ up to a single multiplicative constant. Before proceeding, we note that if c is a constant, then the information regarding h given by the two functions $m(h)$ and $cm(h)$ is the same: the same value of h maximizes both functions, and the second derivative matrices of the logarithm of these two functions are identical. In particular, the Hessians of the logarithm of these two functions at the maximum (i.e. the observed Fisher information) are the same and, therefore, the standard point estimates and confidence regions based on $m(h)$ and $cm(h)$ are identical.

As we will see in Chapter 3, our approach for estimating $m(h)$ up to a single multiplicative constant has two requirements: (i) we need a formula for the ratio $\nu_{h_1}(\boldsymbol{\psi})/\nu_{h_2}(\boldsymbol{\psi})$ for any two hyperparameter values h_1 and h_2, and (ii) for any hyperparameter value h, we need an ergodic Markov chain whose invariant distribution is the posterior $\nu_{h,\boldsymbol{w}}$. This thesis is organized as follows. In Chapter 3 we explain our method for estimating the function $m(h)$ up to a single multiplicative constant (and we provide the formula for the ratio $\nu_{h_1}(\boldsymbol{\psi})/\nu_{h_2}(\boldsymbol{\psi})$). Also, we consider synthetic data sets generated from a simple model in which h is low dimensional and known, and we show that our method correctly estimates the true value of h. In Chapter 4 we describe two Markov chains which satisfy requirement (ii) above. In Chapter 5 we first develop criteria for evaluating the performance of the LDA model indexed by any given hyperparameter value. Then we provide empirical evidence that, according to our criteria, the LDA model that uses the empirical Bayes choice of the hyperparameter can significantly outperform LDA models indexed by default choices of the hyperparameter.

CHAPTER 3
ESTIMATION OF THE MARGINAL LIKELIHOOD UP TO A MULTIPLICATIVE CONSTANT AND ESTIMATION OF POSTERIOR EXPECTATIONS

This chapter consists of four parts. In Section 3.1 we show how the marginal likelihood function can be estimated (up to a constant) with a single MCMC run. In Section 3.2 we show how the entire family of posterior expectations $\{I(h),\, h \in \mathcal{H}\}$ can be estimated with a single MCMC run. In Section 3.3 we explain that the simple estimates given in Sections 3.1 and 3.2 can have large variances, and we present estimates which are far more reliable. In Section 3.4 we show empirically that our method for estimating the value of h that maximizes the marginal likelihood works well in practice. Let $\mathcal{H} = (0, \infty)^{K+1}$ be the hyperparameter space. For any $h \in \mathcal{H}$, ν_h and $\nu_{h,\boldsymbol{w}}$ are prior and posterior distributions, respectively, of the vector $\boldsymbol{\psi} = (\boldsymbol{\beta}, \boldsymbol{\theta}, \boldsymbol{z})$, for which some components are continuous and some are discrete. We will use $\ell_{\boldsymbol{w}}(\boldsymbol{\psi})$ to denote the likelihood function (which is given by line 4 of the LDA model).

3.1 Estimation of the Marginal Likelihood up to a Multiplicative Constant

Note that $m(h)$ is the normalizing constant in the statement "the posterior is proportional to the likelihood times the prior," i.e.

$$\nu_{h,\boldsymbol{w}}(\boldsymbol{\psi}) = \frac{\ell_{\boldsymbol{w}}(\boldsymbol{\psi})\nu_h(\boldsymbol{\psi})}{m(h)}.$$

Now suppose that we have a method for constructing a Markov chain on $\boldsymbol{\psi}$ whose invariant distribution is $\nu_{h,\boldsymbol{w}}$ and which is ergodic. Two Markov chains which satisfy these criteria are discussed in Section 4. Let $h_* \in \mathcal{H}$ be fixed but arbitrary, and let $\boldsymbol{\psi}_1, \boldsymbol{\psi}_2, \ldots$ be an ergodic Markov chain with invariant distribution $\nu_{h*,\boldsymbol{w}}$. For any $h \in \mathcal{H}$, as

$n \to \infty$ we have

$$\begin{aligned}\frac{1}{n}\sum_{s=1}^{n}\frac{\nu_h(\boldsymbol{\psi}_s)}{\nu_{h_*}(\boldsymbol{\psi}_s)} &\xrightarrow{\text{a.s.}} \int \frac{\nu_h(\boldsymbol{\psi})}{\nu_{h_*}(\boldsymbol{\psi})}\, d\nu_{h_*,\boldsymbol{w}}(\boldsymbol{\psi}) \\ &= \frac{m(h)}{m(h_*)}\int \frac{\ell_{\boldsymbol{w}}(\boldsymbol{\psi})\nu_h(\boldsymbol{\psi})/m(h)}{\ell_{\boldsymbol{w}}(\boldsymbol{\psi})\nu_{h_*}(\boldsymbol{\psi})/m(h_*)}\, d\nu_{h_*,\boldsymbol{w}}(\boldsymbol{\psi}) \\ &= \frac{m(h)}{m(h_*)}\int \frac{\nu_{h,\boldsymbol{w}}(\boldsymbol{\psi})}{\nu_{h_*,\boldsymbol{w}}(\boldsymbol{\psi})}\, d\nu_{h_*,\boldsymbol{w}}(\boldsymbol{\psi}) \\ &= \frac{m(h)}{m(h_*)}.\end{aligned} \tag{3–1}$$

The almost sure convergence statement in Equation 3–1 follows from ergodicity of the chain. (There is a slight abuse of notation in Equation 3–1 in that we have used $\nu_{h_*,\boldsymbol{w}}$ to denote a probability measure when we write $d\nu_{h_*,\boldsymbol{w}}$, whereas in the integrand, ν_h, ν_{h_*}, and $\nu_{h_*,\boldsymbol{w}}$ refer to probability densities.)

The significance of Equation 3–1 is that this result shows that we can estimate the entire family $\{m(h)/m(h_*),\, h \in \mathcal{H}\}$ with a single Markov chain run. Since $m(h_*)$ is a constant, the remarks made in Section 2 apply, and we can estimate $\arg\max_h m(h)$. Moreover, if we can establish that the chain is geometrically ergodic, then the estimate on the left side of Equation 3–1 even satisfies a central limit theorem under the moment condition $\int(\nu_h/\nu_{h_*})^{2+\epsilon}\, d\nu_{h_*,\boldsymbol{w}} < \infty$ for some $\epsilon > 0$ (Ibragimov and Linnik, 1971, Theorem 18.5.3); in this case, error margins for the estimate can be obtained. The advantage of this approach is that we bypass the need to deal with the posterior distributions: the estimates on the left side of Equation 3–1 involve *only the priors*.

To use Equation 3–1, we need to have a formula for the ratio of densities $\nu_h(\boldsymbol{\psi})/\nu_{h_*}(\boldsymbol{\psi})$. From the hierarchical nature of the LDA model we have

$$\nu_h(\boldsymbol{\psi}) = \nu_h(\boldsymbol{\beta},\boldsymbol{\theta},\boldsymbol{z}) = p^{(h)}_{\boldsymbol{z}\,|\,\boldsymbol{\theta},\boldsymbol{\beta}}(\boldsymbol{z}\,|\,\boldsymbol{\theta},\boldsymbol{\beta})\; p^{(h)}_{\boldsymbol{\theta}}(\boldsymbol{\theta})\; p^{(h)}_{\boldsymbol{\beta}}(\boldsymbol{\beta})$$

in self-explanatory notation, where $p^{(h)}_{\boldsymbol{z}\,|\,\boldsymbol{\theta},\boldsymbol{\beta}}$, $p^{(h)}_{\boldsymbol{\theta}}$, and $p^{(h)}_{\boldsymbol{\beta}}$ are given by lines 3, 2, and 1, respectively, of the LDA model. Let $n_{dj} = \sum_{i=1}^{n_d} z_{dij}$, i.e. n_{dj} is the number of words in document d that are assigned to topic j. Using the Dirichlet and multinomial distributions

specified in lines 1–3 of the model, we obtain

$$\nu_h(\boldsymbol{\psi}) = \left[\prod_{d=1}^{D}\prod_{j=1}^{K}\theta_{dj}^{n_{dj}}\right]\left[\prod_{d=1}^{D}\left(\frac{\Gamma\left(\sum_{j=1}^{K}\alpha_j\right)}{\prod_{j=1}^{K}\Gamma(\alpha_j)}\prod_{j=1}^{K}\theta_{dj}^{\alpha_j-1}\right)\right]\left[\prod_{j=1}^{K}\left(\frac{\Gamma(V\eta)}{\Gamma(\eta)^V}\prod_{t=1}^{V}\beta_{jt}^{\eta-1}\right)\right]. \tag{3–2}$$

Applying Equation 3–2, we see that for $h_* = (\eta^*, \alpha^*)$, we have

$$\frac{\nu_h(\boldsymbol{\psi})}{\nu_{h_*}(\boldsymbol{\psi})} = \left[\prod_{d=1}^{D}\left(\frac{\Gamma\left(\sum_{j=1}^{K}\alpha_j\right)}{\prod_{j=1}^{K}\Gamma(\alpha_j)}\frac{\prod_{j=1}^{K}\Gamma(\alpha_j^*)}{\Gamma\left(\sum_{j=1}^{K}\alpha_j^*\right)}\prod_{j=1}^{K}\theta_{dj}^{\alpha_j-\alpha_j^*}\right)\right]\left[\prod_{j=1}^{K}\left(\frac{\Gamma(V\eta)}{\Gamma(\eta)^V}\frac{\Gamma(\eta^*)^V}{\Gamma(V\eta^*)}\prod_{t=1}^{V}\beta_{jt}^{\eta-\eta^*}\right)\right]. \tag{3–3}$$

Note that the expression in the first set of brackets in Equation 3–2 does not depend on the hyperparameter, and therefore does not appear in Equation 3–3.

To estimate $m(h)/m(h_*)$ via Equation 3–1, we need an ergodic Markov chain whose invariant distribution is $\nu_{h_*,\boldsymbol{w}}$, and as mentioned earlier, in Section 4 we develop such a chain. In that section, we also discuss an alternative approach, which involves the Griffiths and Steyvers (2004) Gibbs sampler, which is a "collapsed Gibbs sampler" whose invariant distribution is the conditional distribution of $\boldsymbol{z}$ given $\boldsymbol{w}$. This Markov chain cannot be used directly, because to apply Equation 3–1 we need a Markov chain on the triple $(\boldsymbol{\beta}, \boldsymbol{\theta}, \boldsymbol{z})$, whose invariant distribution is $\nu_{h_*,\boldsymbol{w}}$. However, in Section 4, as part of our development, we obtain the conditional distribution of $(\boldsymbol{\beta}, \boldsymbol{\theta})$ given $\boldsymbol{z}$ and $\boldsymbol{w}$, and we show how to sample from this distribution. Therefore, given a Markov chain $\boldsymbol{z}^{(1)}, \ldots, \boldsymbol{z}^{(n)}$ generated via the algorithm of Griffiths and Steyvers (2004), we can form triples $(\boldsymbol{z}^{(1)}, \boldsymbol{\beta}^{(1)}, \boldsymbol{\theta}^{(1)}), \ldots, (\boldsymbol{z}^{(n)}, \boldsymbol{\beta}^{(n)}, \boldsymbol{\theta}^{(n)})$, and it is easy to see that this sequence forms a Markov chain with invariant distribution $\nu_{h_*,\boldsymbol{w}}$, and that this chain inherits the ergodicity properties of the $\boldsymbol{z}$-chain. Either of these two Markov chains can be used to form the estimate on the left side of Equation 3–1.

3.2 Estimation of the Family of Posterior Expectations

We now explain how the plots in Figure 2-1 were created, and our explanation is at a general level. Let g be a function of $\boldsymbol{\psi}$, and let $I(h) = \int g(\boldsymbol{\psi})\, d\nu_{h,\boldsymbol{w}}(\boldsymbol{\psi})$ be the posterior expectation of $g(\boldsymbol{\psi})$ when the prior is ν_h. Suppose that we are interested in

estimating $I(h)$ for all $h \in \mathcal{H}$. (For the plots in Figure 2-1, the function g is simply $g(\boldsymbol{\psi}) = \mathbb{I}(\|\theta_1 - \theta_2\| \leq 0.07)$, where $\mathbb{I}$ is the indicator function.) Proceeding as we did for estimation of the family of ratios $\{m(h)/m(h_*),\, h \in \mathcal{H}\}$, let $h_* \in \mathcal{H}$ be fixed but arbitrary, and let $\boldsymbol{\psi}_1, \boldsymbol{\psi}_2, \ldots$ be an ergodic Markov chain with invariant distribution $\nu_{h*,\boldsymbol{w}}$. To estimate $\int g(\boldsymbol{\psi})\, d\nu_{h,\boldsymbol{w}}(\boldsymbol{\psi})$, the obvious approach is to write

$$\int g(\boldsymbol{\psi})\, d\nu_{h,\boldsymbol{w}}(\boldsymbol{\psi}) = \int g(\boldsymbol{\psi}) \frac{\nu_{h,\boldsymbol{w}}(\boldsymbol{\psi})}{\nu_{h_*,\boldsymbol{w}}(\boldsymbol{\psi})}\, d\nu_{h_*,\boldsymbol{w}}(\boldsymbol{\psi}) \tag{3–4}$$

and then use the importance sampling estimate $(1/n)\sum_{i=1}^n g(\boldsymbol{\psi}_i)[\nu_{h,\boldsymbol{w}}(\boldsymbol{\psi}_i)/\nu_{h_*,\boldsymbol{w}}(\boldsymbol{\psi}_i)]$. This doesn't work because we do not know the normalizing constants for $\nu_{h,\boldsymbol{w}}$ and $\nu_{h_*,\boldsymbol{w}}$. This difficulty is handled by rewriting $\int g(\boldsymbol{\psi})\, d\nu_{h,\boldsymbol{w}}(\boldsymbol{\psi})$, via Equation 3–4, as

$$\begin{aligned}
\int g(\boldsymbol{\psi}) \frac{\ell_{\boldsymbol{w}}(\boldsymbol{\psi})\nu_h(\boldsymbol{\psi})/m(h)}{\ell_{\boldsymbol{w}}(\boldsymbol{\psi})\nu_{h_*}(\boldsymbol{\psi})/m(h_*)}\, d\nu_{h_*,\boldsymbol{w}}(\boldsymbol{\psi}) &= \frac{m(h_*)}{m(h)} \int g(\boldsymbol{\psi}) \frac{\nu_h(\boldsymbol{\psi})}{\nu_{h_*}(\boldsymbol{\psi})}\, d\nu_{h_*,\boldsymbol{w}}(\boldsymbol{\psi}) \\
&= \frac{\frac{m(h_*)}{m(h)} \int g(\boldsymbol{\psi}) \frac{\nu_h(\boldsymbol{\psi})}{\nu_{h_*}(\boldsymbol{\psi})}\, d\nu_{h_*,\boldsymbol{w}}(\boldsymbol{\psi})}{\frac{m(h_*)}{m(h)} \int \frac{\nu_h(\boldsymbol{\psi})}{\nu_{h_*}(\boldsymbol{\psi})}\, d\nu_{h_*,\boldsymbol{w}}(\boldsymbol{\psi})} \qquad \text{(3–5a)} \\
&= \frac{\int g(\boldsymbol{\psi}) \frac{\nu_h(\boldsymbol{\psi})}{\nu_{h_*}(\boldsymbol{\psi})}\, d\nu_{h_*,\boldsymbol{w}}(\boldsymbol{\psi})}{\int \frac{\nu_h(\boldsymbol{\psi})}{\nu_{h_*}(\boldsymbol{\psi})}\, d\nu_{h_*,\boldsymbol{w}}(\boldsymbol{\psi})}, \qquad \text{(3–5b)}
\end{aligned}$$

where in (3–5a) we have used the fact that the integral in the denominator is just 1, in order to cancel the unknown constant $m(h_*)/m(h)$ in (3–5b). The idea to express $\int g(\boldsymbol{\psi})\, d\nu_{h,\boldsymbol{w}}(\boldsymbol{\psi})$ in this way was proposed in a different context by Hastings (1970). Expression (3–5b) is the ratio of two integrals with respect to $\nu_{h_*,\boldsymbol{w}}$, each of which may be estimated from the sequence $\boldsymbol{\psi}_1, \boldsymbol{\psi}_2, \ldots, \boldsymbol{\psi}_n$. We may estimate the numerator and the denominator by

$$\frac{1}{n}\sum_{i=1}^n g(\boldsymbol{\psi}_i)[\nu_h(\boldsymbol{\psi}_i)/\nu_{h_*}(\boldsymbol{\psi}_i)] \quad \text{and} \quad \frac{1}{n}\sum_{i=1}^n [\nu_h(\boldsymbol{\psi}_i)/\nu_{h_*}(\boldsymbol{\psi}_i)]$$

respectively. Thus, if we let

$$w_i^{(h)} = \frac{\nu_h(\boldsymbol{\psi}_i)/\nu_{h_*}(\boldsymbol{\psi}_i)}{\sum_{e=1}^n [\nu_h(\boldsymbol{\psi}_e)/\nu_{h_*}(\boldsymbol{\psi}_e)]},$$

then these are weights, and we see that the desired integral may be estimated by the weighted average

$$\hat{I}(h) = \sum_{i=1}^{n} g(\boldsymbol{\psi}_i) w_i^{(h)}. \tag{3–6}$$

The significance of this development is that it shows that with a single Markov chain run, we can estimate the entire family of posterior expectations $\{I(h),\, h \in \mathcal{H}\}$. As was the case for the estimate on the left side of Equation 3–1, the estimate given by Equation 3–6 is remarkable in its simplicity. To compute it, we need to know only the ratio of the *priors*, and not the posteriors.

3.3 Serial Tempering

Unfortunately, Equation 3–6 suffers a serious defect: unless h is close to h_*, ν_h can be nearly singular with respect to ν_{h_*} over the region where the $\boldsymbol{\psi}_i$'s are likely to be, resulting in a very unstable estimate. A similar remark applies to the estimate on the left side of Equation 3–1. In other words, there is effectively a "radius" around h_* within which one can safely move. To state the problem more explicitly: there does not exist a single h_* for which the ratios $\nu_h(\boldsymbol{\psi})/\nu_{h_*}(\boldsymbol{\psi})$ have small variance simultaneously for all $h \in \mathcal{H}$. One way of dealing with this problem is to replace ν_{h_*} in the denominator by $(1/J)\sum_{j=1}^{J} b_i \nu_{h_j}$, for some suitable choice of $h_1, \ldots, h_J \in \mathcal{H}$, and positive constants $b_1, \ldots, b_J$. This approach may be implemented by a methodology called serial tempering, originally developed by Marinari and Parisi (1992) (see also Geyer and Thompson (1995)) for the purpose of improving mixing rates of certain Markov chains that are used to simulate physical systems in statistical mechanics. Here, we use it for a very different purpose, namely to increase the range of values over which importance sampling estimates have small variance. (See Geyer (2011) for a review of various applications of serial tempering.) We now summarize this methodology, in the present context, and show how it can be used to produce estimates that are stable over a wide range of h values. Our explanations are detailed, because the material is not trivial and because we wish to deal with estimates of both marginal likelihood and posterior expectations. To simplify the discussion, suppose

that in line 2 of the LDA model we take $\boldsymbol{\alpha} = (\alpha, \ldots, \alpha)$, i.e. $\mathrm{Dir}_K(\boldsymbol{\alpha})$ is a symmetric Dirichlet, so that $\mathcal{H}$ is effectively two-dimensional, and suppose that we take $\mathcal{H}$ to be a bounded set of the form $\mathcal{H} = [\eta_L, \eta_U] \times [\alpha_L, \alpha_U]$.

Let $h_1, \ldots, h_J \in \mathcal{H}$ be fixed points; these should be taken to "cover" $\mathcal{H}$, in the sense that for every $h \in \mathcal{H}$, ν_h is "close to" at least one of $\nu_{h_1}, \ldots, \nu_{h_J}$. The idea is then to run a Markov chain which has invariant distribution given by the mixture $(1/J)\sum_{j=1}^{J} \nu_{h_j,\boldsymbol{w}}$. The updates will sample different components of this mixture, with jumps from one component to another. We now describe this carefully. Let Ψ denote the state space for $\boldsymbol{\psi}$. Recall that $\boldsymbol{\psi}$ has some continuous components and some discrete components. To proceed rigorously, we will take ν_h and $\nu_{h,\boldsymbol{w}}$ to all be densities with respect to a measure μ on Ψ. Define $\mathcal{L} = \{1, \ldots, J\}$, and for $j \in \mathcal{L}$, suppose that Φ_j is a Markov transition function on Ψ with invariant distribution equal to the posterior $\nu_{h_j,\boldsymbol{w}}$. On occasion we will write ν_j instead of ν_{h_j}. This notation is somewhat inconsistent, but we use it in order to avoid having double and triple subscripts. We have $\nu_{h,\boldsymbol{w}} = \ell_{\boldsymbol{w}}\, \nu_h/m(h)$ and $\nu_{h_j,\boldsymbol{w}} = \ell_{\boldsymbol{w}}\, \nu_j/m(h_j)$, $j = 1, \ldots, J$.

Serial tempering involves considering the state space $\mathcal{L} \times \Psi$, and forming the family of distributions $\{P_\zeta,\, \zeta \in \mathbb{R}^J\}$ on $\mathcal{L} \times \Psi$ with densities

$$p_\zeta(j, \boldsymbol{\psi}) \propto \ell_{\boldsymbol{w}}(\boldsymbol{\psi})\nu_j(\boldsymbol{\psi})/\zeta_j. \tag{3–7}$$

(To be pedantic, these are densities with respect to $\mu \times \sigma$, where σ is counting measure on $\mathcal{L}$.) The vector ζ is a tuning parameter, which we discuss later. Let $\Gamma(j, \cdot)$ be a Markov transition function on $\mathcal{L}$. In our context, we would typically take $\Gamma(j, \cdot)$ to be the uniform distribution on $\mathcal{N}_j$, where $\mathcal{N}_j$ is a set consisting of the indices of the h_l's which are close to h_j. Serial tempering is a Markov chain on $\mathcal{L} \times \Psi$ which can be viewed as a two-block Metropolis-Hastings (i.e. Metropolis-within-Gibbs) algorithm, and is run as follows. Suppose that the current state of the chain is $(L_{t-1}, \boldsymbol{\psi}_{t-1})$.

- A new value $j \sim \Gamma(L_{t-1}, \cdot)$ is proposed. We set $L_t = j$ with the Metropolis probability
$$\min\left\{1, \frac{\Gamma(j, L_{t-1})}{\Gamma(L_{t-1}, j)} \frac{\nu_j(\boldsymbol{\psi})/\zeta_j}{\nu_{L_{t-1}}(\boldsymbol{\psi})/\zeta_{L_{t-1}}}\right\},$$
and with the remaining probability we set $L_t = L_{t-1}$.

- Generate $\boldsymbol{\psi}_t \sim \Phi_{L_t}(\boldsymbol{\psi}_{t-1}, \cdot)$.

By standard arguments, the density in Equation 3–7 is an invariant density for the serial tempering chain. A key observation is that the $\boldsymbol{\psi}$-marginal density of p_ζ is

$$f_\zeta(\boldsymbol{\psi}) = (1/c_\zeta) \sum_{j=1}^{J} \ell_{\boldsymbol{w}}(\boldsymbol{\psi}) \nu_j(\boldsymbol{\psi})/\zeta_j, \qquad \text{where} \qquad c_\zeta = \sum_{j=1}^{J} m(h_j)/\zeta_j. \tag{3–8}$$

Suppose that $(L_1, \boldsymbol{\psi}_1), (L_2, \boldsymbol{\psi}_2), \ldots$ is a serial tempering chain. To estimate $m(h)$, consider

$$\widehat{M}_\zeta(h) = \frac{1}{n} \sum_{i=1}^{n} \frac{\nu_h(\boldsymbol{\psi}_i)}{(1/J) \sum_{j=1}^{J} \nu_j(\boldsymbol{\psi}_i)/\zeta_j}. \tag{3–9}$$

Note that this estimate depends only on the $\boldsymbol{\psi}$-part of the chain. Assuming that we have established that the chain is ergodic, we have

$$\begin{aligned} \widehat{M}_\zeta(h) &\xrightarrow{\text{a.s.}} \int \frac{\nu_h(\boldsymbol{\psi})}{(1/J) \sum_{j=1}^{J} \nu_j(\boldsymbol{\psi})/\zeta_j} \frac{\sum_{j=1}^{J} \ell_{\boldsymbol{w}}(\boldsymbol{\psi}) \nu_j(\boldsymbol{\psi})/\zeta_j}{c_\zeta} \, d\mu(\boldsymbol{\psi}) \\ &= \int \frac{\ell_{\boldsymbol{w}}(\boldsymbol{\psi}) \nu_h(\boldsymbol{\psi})}{c_\zeta/J} \, d\mu(\boldsymbol{\psi}) \\ &= \frac{m(h)}{c_\zeta/J}. \end{aligned} \tag{3–10}$$

This means that for any ζ, the family $\left\{\widehat{M}_\zeta(h),\, h \in \mathcal{H}\right\}$ can be used to estimate the family $\{m(h),\, h \in \mathcal{H}\}$, up to a single multiplicative constant.

To estimate the family of integrals $\left\{\int g(\boldsymbol{\psi})\, d\nu_{h,\boldsymbol{w}}(\boldsymbol{\psi}),\, h \in \mathcal{H}\right\}$, we proceed as follows. Let

$$\widehat{U}_\zeta(h) = \frac{1}{n} \sum_{i=1}^{n} \frac{g(\boldsymbol{\psi}_i) \nu_h(\boldsymbol{\psi}_i)}{(1/J) \sum_{j=1}^{J} \nu_j(\boldsymbol{\psi}_i)/\zeta_j}. \tag{3–11}$$

By ergodicity we have

$$\begin{aligned}\widehat{U}_\zeta(h) &\xrightarrow{\text{a.s.}} \int \frac{g(\boldsymbol{\psi})\nu_h(\boldsymbol{\psi})}{(1/J)\sum_{j=1}^J \nu_j(\boldsymbol{\psi})/\zeta_j} \frac{\sum_{j=1}^J \ell_{\boldsymbol{w}}(\boldsymbol{\psi})\nu_j(\boldsymbol{\psi})/\zeta_j}{c_\zeta}\, d\mu(\boldsymbol{\psi}) \\ &= \int \frac{\ell_{\boldsymbol{w}}(\boldsymbol{\psi})g(\boldsymbol{\psi})\nu_h(\boldsymbol{\psi})}{c_\zeta/J}\, d\mu(\boldsymbol{\psi}) \\ &= \frac{m(h)}{c_\zeta/J} \int g(\boldsymbol{\psi})\, d\nu_{h,\boldsymbol{w}}(\boldsymbol{\psi}).\end{aligned} \tag{3–12}$$

Combining the convergence statements given by Equation 3–12 and Equation 3–10, we see that

$$\hat{I}_\zeta^{\text{st}}(h) := \frac{\widehat{U}_\zeta(h)}{\widehat{M}_\zeta(h)} \xrightarrow{\text{a.s.}} \int g(\boldsymbol{\psi})\, d\nu_{h,\boldsymbol{w}}(\boldsymbol{\psi}).$$

Suppose that for some constant a, we have

$$(\zeta_1, \ldots, \zeta_J) = a(m(h_1), \ldots, m(h_J)). \tag{3–13}$$

Then $c_\zeta = J/a$, and $f_\zeta(\boldsymbol{\psi}) = (1/J)\sum_{j=1}^J \nu_{h_j,\boldsymbol{w}}(\boldsymbol{\psi})$, i.e. the ψ-marginal of p_ζ (see Equation 3–8) gives equal weight to each of the component distributions in the mixture. (Expressing this slightly differently, if Equation 3–13 is true, then the invariant density given by Equation 3–7 becomes $p_\zeta(j, \boldsymbol{\psi}) = (1/J)\nu_{h_j,\boldsymbol{w}}(\boldsymbol{\psi})$, so the L-marginal distribution of p_ζ gives mass $(1/J)$ to each point in $\mathcal{L}$.) Therefore, for large n, the proportions of time spent in the J components of the mixture are about the same, a feature which is essential if serial tempering is to work well. In practice, we cannot arrange for Equation 3–13 to be true, because $m(h_1), \ldots, m(h_J)$ are unknown. However, the vector $(m(h_1), \ldots, m(h_J))$ may be estimated (up to a single multiplicative constant) iteratively as follows. If the current value is $\zeta^{(t)}$, then set

$$\left(\zeta_1^{(t+1)}, \ldots, \zeta_J^{(t+1)}\right) = \left(\widehat{M}_{\zeta^{(t)}}(h_1), \ldots, \widehat{M}_{\zeta^{(t)}}(h_J)\right). \tag{3–14}$$

From the convergence result given in Equation 3–10, we get $\widehat{M}_{\zeta^{(t)}}(h_j) \xrightarrow{\text{a.s.}} m(h_j)/a_{\zeta^{(t)}}$, where $a_{\zeta^{(t)}}$ is a constant, i.e. Equation 3–13 is nearly satisfied by $\left(\zeta_1^{(t+1)}, \ldots, \zeta_J^{(t+1)}\right)$.

To sum up, we estimate the family of marginal likelihoods (up to a constant) and the family of posterior expectations as follows. First, we obtain the vector of tuning parameters ζ via the iterative scheme given by Equation 3–14. To estimate the family of marginal likelihoods (up to a constant) we use $\widehat{M}_\zeta(h)$ defined in Equation 3–9, and to estimate the family of posterior expectations we use $\hat{I}^{\text{st}}_\zeta(h) = \widehat{U}_\zeta(h)/\widehat{M}_\zeta(h)$ (see Equation 3–11 and Equation 3–9).

We point out that it is possible to estimate the family of marginal likelihoods (up to a constant) by

$$\widetilde{M}_\zeta(h) = \frac{1}{n}\sum_{t=1}^{n} \frac{\nu_h(\boldsymbol{\psi}_t)}{\nu_{L_t}(\boldsymbol{\psi}_t)/\zeta_{L_t}}. \tag{3–15}$$

Note that $\widetilde{M}_\zeta(h)$ uses the sequence of pairs $(L_1, \boldsymbol{\psi}_1), (L_2, \boldsymbol{\psi}_2), \ldots$, and not just the sequence $\boldsymbol{\psi}_1, \boldsymbol{\psi}_2, \ldots$. To see why Equation 3–15 is a valid estimator, observe that by ergodicity we have

$$\begin{aligned}
\widetilde{M}_\zeta(h) &\xrightarrow{\text{a.s.}} \iint \frac{\nu_h(\boldsymbol{\psi})}{\nu_L(\boldsymbol{\psi})/\zeta_L} \cdot \left[\frac{1}{c_\zeta}\ell_{\boldsymbol{w}}(\boldsymbol{\psi})\nu_L(\boldsymbol{\psi})/\zeta_L\right] d\mu(\boldsymbol{\psi})\, d\sigma(L) \\
&= \iint \frac{m(h)}{c_\zeta}\nu_{h,\boldsymbol{w}}(\boldsymbol{\psi})\, d\mu(\boldsymbol{\psi})\, d\sigma(L) \\
&= J\frac{m(h)}{c_\zeta}.
\end{aligned} \tag{3–16}$$

(Note that the limit in Equation 3–16 is the same as the limit in Equation 3–10.) Similarly, we may estimate the integral $\int g(\boldsymbol{\psi})\, d\nu_{h,\boldsymbol{w}}(\boldsymbol{\psi})$ by the ratio

$$\tilde{I}^{\text{st}}_\zeta(h) = \sum_{t=1}^{n} \frac{g(\boldsymbol{\psi}_t)\nu_h(\boldsymbol{\psi}_t)}{\nu_{L_t}(\boldsymbol{\psi}_t)/\zeta_{L_t}} \Bigg/ \sum_{t=1}^{n} \frac{\nu_h(\boldsymbol{\psi}_t)}{\nu_{L_t}(\boldsymbol{\psi}_t)/\zeta_{L_t}}.$$

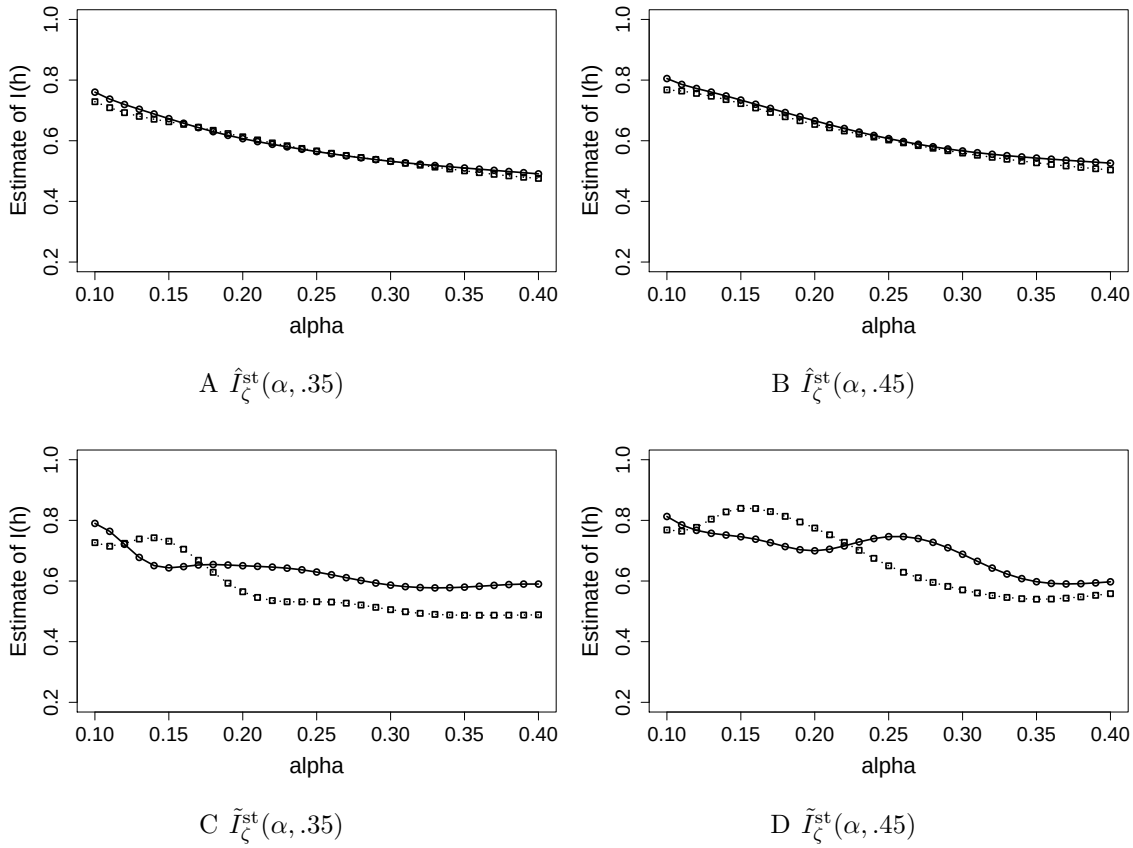
The estimate $\tilde{I}^{\text{st}}_\zeta(h)$ is also based on the pairs $(L_1, \boldsymbol{\psi}_1), (L_2, \boldsymbol{\psi}_2), \ldots$, and it is easy to show that $\tilde{I}^{\text{st}}_\zeta(h) \xrightarrow{\text{a.s.}} \int g(\boldsymbol{\psi})\, d\nu_{h,\boldsymbol{w}}(\boldsymbol{\psi})$.

The estimates $\widetilde{M}_\zeta(h)$ and $\tilde{I}^{\text{st}}_\zeta(h)$ are the ones that are used by Marinari and Parisi (1992) and Geyer and Thompson (1995), but $\widehat{M}_\zeta(h)$ and $\hat{I}^{\text{st}}_\zeta(h)$ appear to significantly outperform $\widetilde{M}_\zeta(h)$ and $\tilde{I}^{\text{st}}_\zeta(h)$ in terms of accuracy. To provide some evidence of this, we reconsidered the corpus described in Section 2 and the family of posterior probabilities

$I(h)$, as h varies, discussed there. We calculated the estimates $\hat{I}_\zeta^{\text{st}}(h)$ twice, using two different seeds, and also calculated $\tilde{I}_\zeta^{\text{st}}(h)$ twice, using two different seeds. The four functions were constructed via four independent serial tempering experiments, each involving three iterations to form the tuning parameter ζ, and one final iteration to form the estimate of $I(h)$. Each serial tempering chain had length 100,000. Figure 3-1A shows the two independent estimates $\hat{I}_\zeta^{\text{st}}(h)$ as α varies over the range $(0.1, 0.4)$ with η fixed at .35, and Figure 3-1B shows the two estimates $\hat{I}_\zeta^{\text{st}}(h)$ as α varies over the same range, but with η fixed at .45. Figures 3-1C and 3-1D are the same as Figures 3-1A and 3-1B, except that $\tilde{I}_\zeta^{\text{st}}(h)$ is used. These plots show clearly that two independent replicates of $\hat{I}_\zeta^{\text{st}}(h)$ are very similar to each other, while two independent replicates of $\tilde{I}_\zeta^{\text{st}}(h)$ are not. Specifically, the maximum deviation between the two independent replicates of $\hat{I}_\zeta^{\text{st}}(h)$ is 0.038 and the maximum deviation between the two independent replicates of $\tilde{I}_\zeta^{\text{st}}(h)$ is 0.132. Here, "maximum deviation" refers to the entire range $(\eta, \alpha) \in (0.35, 0.45) \times (0.1, 0.4)$. Section 3.4 presents the results of some experiments that compare the accuracy of $\widehat{M}_\zeta(h)$ and $\widetilde{M}_\zeta(h)$, and the conclusions are qualitatively the same: the standard deviation of $\widehat{M}_\zeta(h)$ is considerably smaller than that of $\widetilde{M}_\zeta(h)$. Ostensibly, $\widehat{M}_\zeta(h)$ and $\hat{I}_\zeta^{\text{st}}(h)$ require more computation, but the quantities $(1/J)\sum_{j=1}^{J} \nu_j(\boldsymbol{\psi}_i)/\zeta_j$, $i = 1, \ldots, n$ are calculated once, and stored. Doing this essentially offsets the increased computing cost.

3.4 Illustration on Low-Dimensional Examples

Consider the LDA model with a given hyperparameter value, which we will denote by h_{true}, and suppose we carry out steps 1–4 of the model, where in the final step we generate the corpus $\boldsymbol{w}$. The maximum likelihood estimate of h is $\hat{h} = \arg\max_h m(h)$ and, as we mentioned earlier, for any constant a, known or unknown, $\arg\max_h m(h) = \arg\max_h am(h)$. As noted earlier, the family $\{\widehat{M}_\zeta(h),\ h \in \mathcal{H}\}$, where $\widehat{M}_\zeta(h)$ is given by Equation 3–9, may be used to estimate the family $\{m(h),\ h \in \mathcal{H}\}$ up to a multiplicative constant. So we may use $\arg\max_h \widehat{M}_\zeta(h)$ to estimate $\hat{h}$.



A $\hat{I}^{\mathrm{st}}_{\zeta}(\alpha, .35)$ B $\hat{I}^{\mathrm{st}}_{\zeta}(\alpha, .45)$

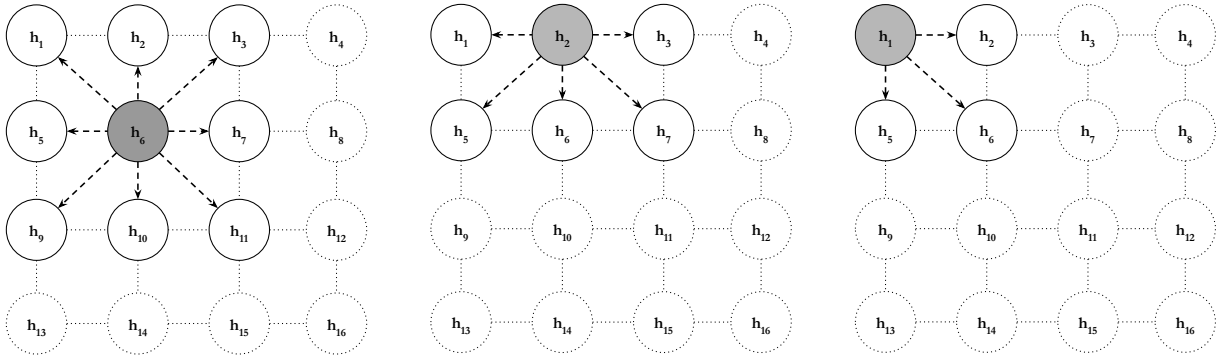
C $\tilde{I}^{\mathrm{st}}_{\zeta}(\alpha, .35)$ D $\tilde{I}^{\mathrm{st}}_{\zeta}(\alpha, .45)$

Figure 3-1. Comparison of the variability of $\hat{I}^{\mathrm{st}}_{\zeta}$ and $\tilde{I}^{\mathrm{st}}_{\zeta}$. Each of the top two panels shows two independent estimates of $I(\alpha, \eta)$, using $\hat{I}^{\mathrm{st}}_{\zeta}(\alpha, \eta)$. For the left panel, $\eta = .35$, and for the right panel, $\eta = .45$. Here, $I(h)$ is the posterior probability that $\|\theta_1 - \theta_2\| < 0.07$ when the prior is ν_h. The bottom two panels use $\tilde{I}^{\mathrm{st}}_{\zeta}$ instead of $\tilde{I}^{\mathrm{st}}_{\zeta}$. The superiority of $\hat{I}^{\mathrm{st}}_{\zeta}$ over $\tilde{I}^{\mathrm{st}}_{\zeta}$ is striking.

Let $\hat{B}(h)$ be the estimate of $m(h)/m(h_*)$ given by the left side of Equation 3–1. In theory, $\arg\max_h \hat{B}(h)$ can also be used. However, as we pointed out earlier, $\hat{B}(h)$ is stable only for h close to h_*—a similar remark applies to $\hat{I}(h)$—and unless the region of hyperparameter values of interest is small, we would not use $\hat{B}(h)$ and $\hat{I}(h)$, and we would use estimates based on serial tempering instead. We have included the derivations of $\hat{B}(h)$ and $\hat{I}(h)$ primarily for motivation, as these makes it easier to understand the development of the serial tempering estimates. In Section 3.3 we presented an experiment which strongly suggested that $\hat{I}^{\mathrm{st}}_{\zeta}(h)$ is significantly better than $\tilde{I}^{\mathrm{st}}_{\zeta}(h)$ in terms of variance.

Here we present the results of an experiment which demonstrates good performance of $\hat{\hat{h}} := \arg\max_h \widehat{M}_\zeta(h)$ as an estimate of h_{true}. We took $\boldsymbol{\alpha} = (\alpha, \ldots, \alpha)$, i.e. $\text{Dir}_K(\boldsymbol{\alpha})$ is a symmetric Dirichlet, so that the hyperparameter in the model reduces to $h = (\eta, \alpha) \in (0, \infty)^2$. We did this solely so that we can visualize the estimate of $m(h)$. Our experiment is set up as follows: the vocabulary size is $V = 20$, the number of documents is $D = 1000$, the document lengths are $n_d = 80$, $d = 1, \ldots, D$, and the number of topics is $K = 2$. We used four settings for the hyperparameter under which we generate the model: h_{true} is taken to be $(2, 2)$, $(2, 5)$, $(5, 2)$, and $(5, 5)$. We estimated the marginal likelihood surface (up to a constant) on the evenly-spaced 41×41 grid of 1681 values over the region $(\eta, \alpha) \in (0.5, 6.5) \times (0.5, 6.5)$ using $\widehat{M}_\zeta(h)$ calculated from a serial tempering chain implemented as follows. We took the sequence $h_1, \ldots, h_J$ to consist of a 9×9 subgrid of 81 evenly-spaced values over the same region. For each hyperparameter value h_j $(j = 1, \ldots, 81)$, we took Φ_j to be the Markov transition function of the full Gibbs sampler alluded to earlier and described in detail in Section 4; this sampler runs over $\boldsymbol{\psi} = (\boldsymbol{\beta}, \boldsymbol{\theta}, \boldsymbol{z})$. We took the Markov transition function $K(j, \cdot)$ on $\mathcal{L} = \{1, \ldots, 81\}$ to be the uniform distribution on $\mathcal{N}_j$ where $\mathcal{N}_j$ is the subset of $\mathcal{L}$ consisting of the indices of the h_l's that are neighbors of the point h_j. (An interior point has eight neighbors, an edge point has five, and a corner point has three.) Figure 3-2 describes the neighborhood structure for interior, edge, and corner points. We obtained the value ζ^{final} via three iterations of the scheme given by Equation 3–14, in which we ran the serial tempering chain in each tuning iteration for 100,000 iterations after a short burn-in period, and we initialized $\zeta^{(0)} = \left(\zeta_1^{(0)}, \ldots, \zeta_{81}^{(0)}\right) = (1, \ldots, 1)$. Using ζ^{final}, we ran the final serial tempering chain for the same number of iterations as in the tuning stage.

Figure 3-3 gives plots of the estimates $\widehat{M}_\zeta(h)$ and also of their Monte Carlo standard errors (MCSE) for the four specifications of h_{true}. We computed these standard error estimates using the method of batch means, which is implemented by the R package `mcmcse` in Flegal and Hughes (2012); the standard errors are valid pointwise, as opposed

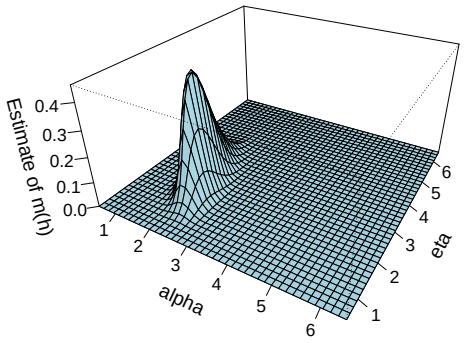


A Transitions from an interior point B Transitions from an edge point C Transitions from a corner point

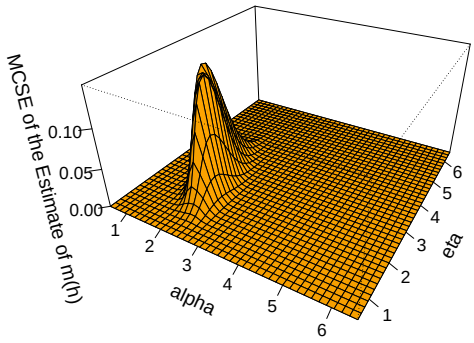
Figure 3-2. Neighborhood structures for interior, edge, and corner points in a 4×4 grid for the serial tempering chain.

to globally, over the h-region of interest. As can be seen from the figure, the location of the point at which the maximum $\widehat{M}_\zeta(h)$ occurs estimates the true value of h reasonably well. In addition, the standard errors of the estimates $\widehat{M}_\zeta(h)$ indicate that the accuracy of these estimates is adequate over the entire h-range for each of the four cases of h_{true}. This experiment involves modest sample sizes; when we increase the document lengths and the number of documents, the surfaces become more peaked, and $\hat{\hat{h}}$ is closer to h_{true} (experiments not shown).

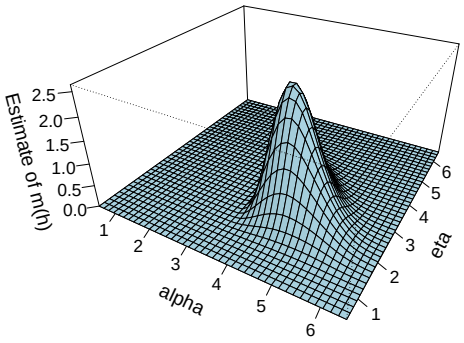
For each specification of h_{true}, we computed the estimates $\widetilde{M}_\zeta(h)$ and also of their Monte Carlo standard errors, and Figure 3-4 shows the plots. As we can see from the figure, while $\arg\max_h \widetilde{M}_\zeta(h)$ provides reasonable estimates of h_{true}, these estimates are typically not better than $\arg\max_h \widehat{M}_\zeta(h)$, and can be much worse. Furthermore, the standard errors of $\widetilde{M}_\zeta(h)$ are always greater than those of $\widehat{M}_\zeta(h)$, and sometimes significantly so. These experiments give results that are analogous to those presented in Section 3.3, and the combined results strongly suggest that $\widehat{M}_\zeta(h)$ and $\hat{I}^{\text{st}}_\zeta(h)$ greatly outperform $\widetilde{M}_\zeta(h)$ and $\tilde{I}^{\text{st}}_\zeta(h)$, respectively.



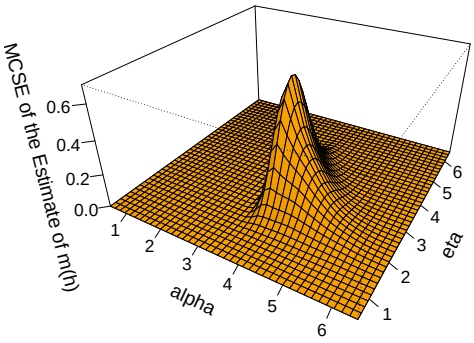
A $\widehat{M}(h)$: $h_{\text{true}} = (2, 2)$, $\hat{\hat{h}} = (2.15, 2.15)$



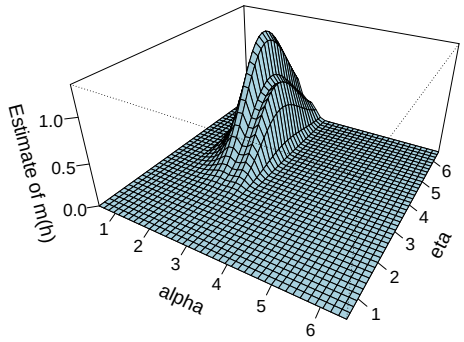
B MCSE of $\widehat{M}(h)$: $h_{\text{true}} = (2, 2)$



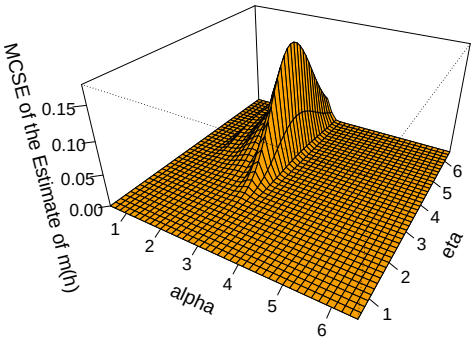
C $\widehat{M}(h)$: $h_{\text{true}} = (2, 5)$, $\hat{\hat{h}} = (2.60, 4.25)$



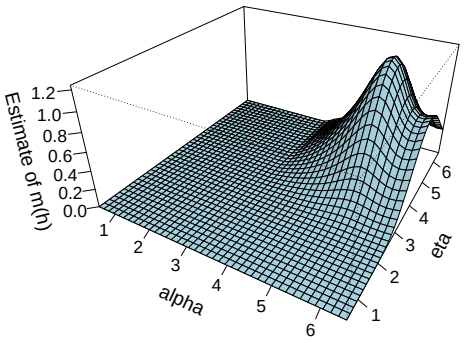
D MCSE of $\widehat{M}(h)$: $h_{\text{true}} = (2, 5)$



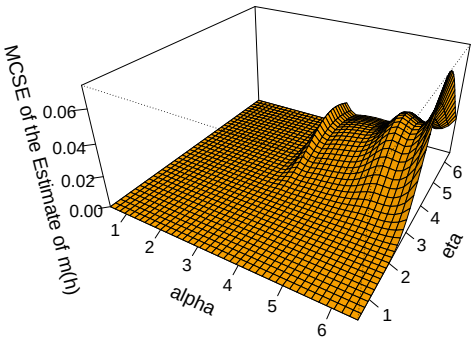
E $\widehat{M}(h)$: $h_{\text{true}} = (5, 2)$, $\hat{\hat{h}} = (4.85, 2.15)$



F MCSE of $\widehat{M}(h)$: $h_{\text{true}} = (5, 2)$

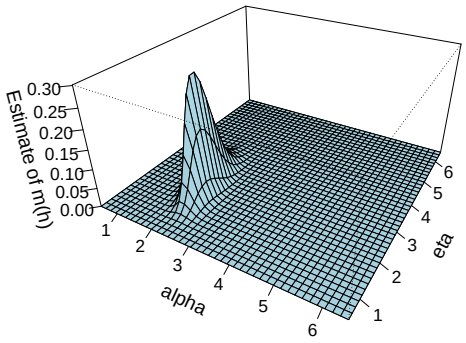


G $\widehat{M}(h)$: $h_{\text{true}} = (5, 5)$, $\hat{\hat{h}} = (5.00, 5.45)$

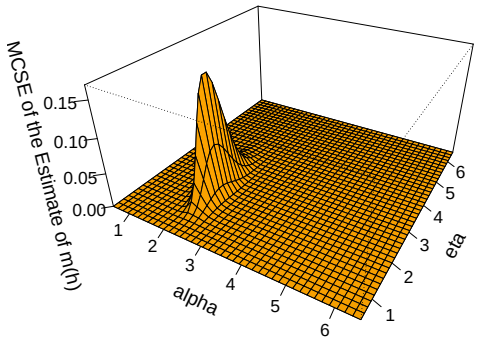


H MCSE of $\widehat{M}(h)$: $h_{\text{true}} = (5, 5)$

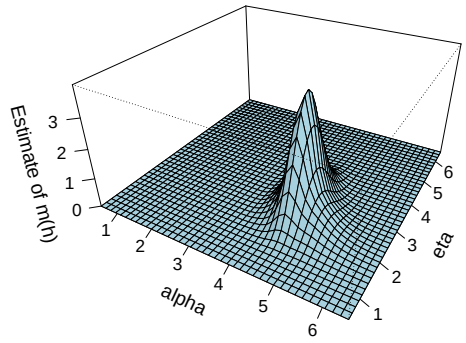
Figure 3-3. $\widehat{M}(h)$ and MCSE of $\widehat{M}(h)$ for four values of h_{true}. In each case, $\hat{\hat{h}}$ is close to h_{true}.



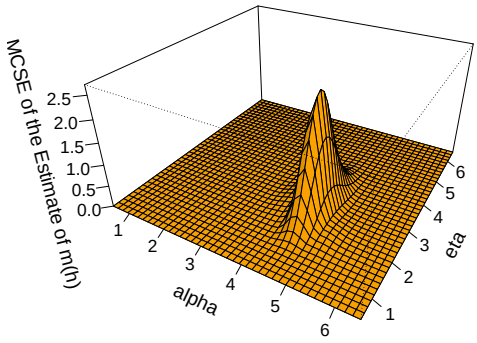
A $\widetilde{M}(h)$: $h_{\text{true}} = (2, 2)$, $\hat{\hat{h}} = (2.15, 2.15)$



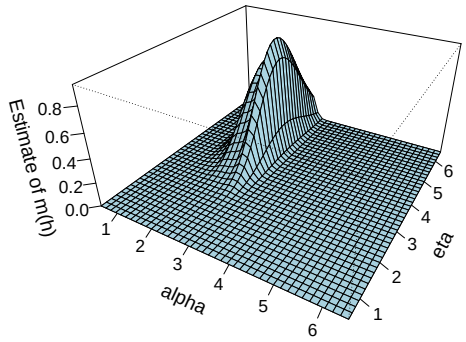
B MCSE of $\widetilde{M}(h)$: $h_{\text{true}} = (2, 2)$



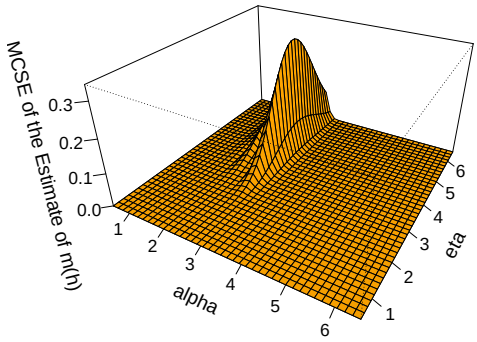
C $\widetilde{M}(h)$: $h_{\text{true}} = (2, 5)$, $\hat{\hat{h}} = (2.60, 4.55)$



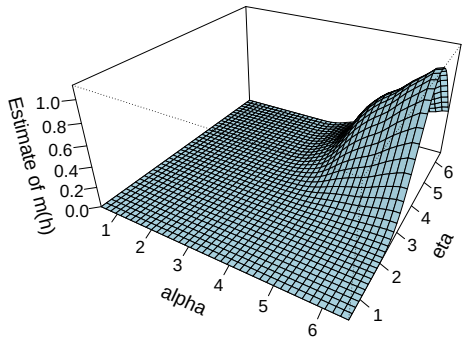
D MCSE of $\widetilde{M}(h)$: $h_{\text{true}} = (2, 5)$



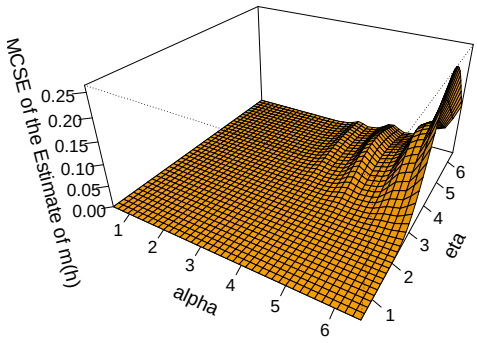
E $\widetilde{M}(h)$: $h_{\text{true}} = (5, 2)$, $\hat{\hat{h}} = (4.70, 2.60)$



F MCSE of $\widetilde{M}(h)$: $h_{\text{true}} = (5, 2)$



G $\widetilde{M}(h)$: $h_{\text{true}} = (5, 5)$, $\hat{\hat{h}} = (5.00, 6.50)$



H MCSE of $\widetilde{M}(h)$: $h_{\text{true}} = (5, 5)$

Figure 3-4. $\widetilde{M}(h)$ and MCSE of $\widetilde{M}(h)$ for four specifications of h_{true}.

CHAPTER 4
TWO MARKOV CHAINS ON $(\boldsymbol{\beta}, \boldsymbol{\theta}, \boldsymbol{Z})$

In order to develop Markov chains on $\boldsymbol{\psi} = (\boldsymbol{\beta}, \boldsymbol{\theta}, \boldsymbol{z})$ whose invariant distribution is the posterior $\nu_{h,\boldsymbol{w}}$, we first express the posterior in a convenient form. We start with the familiar formula

$$\nu_{h,\boldsymbol{w}}(\boldsymbol{\psi}) \propto \ell_{\boldsymbol{w}}(\boldsymbol{\psi})\nu_h(\boldsymbol{\psi}), \tag{4–1}$$

where the likelihood $\ell_{\boldsymbol{w}}(\boldsymbol{\psi}) = p^{(h)}_{\boldsymbol{w}\,|\,\boldsymbol{z},\boldsymbol{\theta},\boldsymbol{\beta}}(\boldsymbol{w}\,|\,\boldsymbol{z},\boldsymbol{\theta},\boldsymbol{\beta})$ is given by line 4 of the LDA model statement. For $d = 1, \ldots, D$ and $j = 1, \ldots, K$, let $S_{dj} = \{i : 1 \le i \le n_d \text{ and } z_{dij} = 1\}$, which is the set of indices of all words in document d whose latent topic variable is j. With this notation, from line 4 of the model statement we have

$$\begin{aligned} p^{(h)}_{\boldsymbol{w}\,|\,\boldsymbol{z},\boldsymbol{\theta},\boldsymbol{\beta}}(\boldsymbol{w}\,|\,\boldsymbol{z},\boldsymbol{\theta},\boldsymbol{\beta}) &= \prod_{d=1}^{D}\prod_{i=1}^{n_d}\prod_{j:z_{dij}=1}\prod_{t=1}^{V}\beta_{jt}^{w_{dit}} \\ &= \prod_{d=1}^{D}\prod_{j=1}^{K}\prod_{t=1}^{V}\prod_{i\in S_{dj}}\beta_{jt}^{w_{dit}} \\ &= \prod_{d=1}^{D}\prod_{j=1}^{K}\prod_{t=1}^{V}\beta_{jt}^{\sum_{i\in S_{dj}} w_{dit}} \\ &= \prod_{d=1}^{D}\prod_{j=1}^{K}\prod_{t=1}^{V}\beta_{jt}^{m_{djt}}, \end{aligned} \tag{4–2}$$

where $m_{djt} = \sum_{i\in S_{dj}} w_{dit}$ counts the number of words in document d for which the latent topic is j and the index of the word in the vocabulary is t. Recalling the definition of n_{dj} given just before Equation 3–2, and noting that $\sum_{i\in S_{dj}} w_{dit} = \sum_{i=1}^{n_d} z_{dij}w_{dit}$, we see that

$$m_{djt} = \sum_{i=1}^{n_d} z_{dij}w_{dit} \quad \text{and} \quad \sum_{t=1}^{V} m_{djt} = n_{dj}. \tag{4–3}$$

Plugging the likelihood given by Equation 4–2 and the prior given by Equation 3–2 into Equation 4–1, and absorbing Dirichlet normalizing constants into an overall constant of proportionality, we have

$$\nu_{h,\boldsymbol{w}}(\boldsymbol{\psi}) \propto \left[\prod_{d=1}^{D}\prod_{j=1}^{K}\prod_{t=1}^{V}\beta_{jt}^{m_{djt}}\right]\left[\prod_{d=1}^{D}\prod_{j=1}^{K}\theta_{dj}^{n_{dj}}\right]\left[\prod_{d=1}^{D}\prod_{j=1}^{K}\theta_{dj}^{\alpha_j-1}\right]\left[\prod_{j=1}^{K}\prod_{t=1}^{V}\beta_{jt}^{\eta-1}\right]. \tag{4–4}$$

The expression for $\nu_{h,\boldsymbol{w}}(\boldsymbol{\psi})$ above also appears in the unpublished report Fuentes et al. (2011).

4.1 The Conditional Distributions of $(\boldsymbol{\beta}, \boldsymbol{\theta})$ Given $\boldsymbol{z}$ and of $\boldsymbol{z}$ Given $(\boldsymbol{\beta}, \boldsymbol{\theta})$

All distributions below are conditional distributions given $\boldsymbol{w}$, which is fixed, and henceforth this conditioning is suppressed in the notation. Note that in Equation 4–4, the terms m_{djt} and n_{dj} depend on $\boldsymbol{z}$. By inspection of Equation 4–4, we see that given $\boldsymbol{z}$,

$$\begin{aligned} &\theta_1, \ldots, \theta_D \text{ and } \beta_1, \ldots, \beta_K \text{ are all independent,} \\ &\theta_d \sim \mathrm{Dir}_K\big(n_{d1} + \alpha_1, \ldots, n_{dK} + \alpha_K\big), \\ &\beta_j \sim \mathrm{Dir}_V\big(\textstyle\sum_{d=1}^{D} m_{dj1} + \eta, \ldots, \sum_{d=1}^{D} m_{djV} + \eta\big). \end{aligned} \tag{4–5}$$

From Equation 4–4 we also see that

$$\begin{aligned} p^{(h)}_{\boldsymbol{z}\,|\,\boldsymbol{\theta},\boldsymbol{\beta}}(\boldsymbol{z}\,|\,\boldsymbol{\theta},\boldsymbol{\beta}) &\propto \prod_{d=1}^{D}\prod_{j=1}^{K}\left(\left[\prod_{t=1}^{V}\beta_{jt}^{m_{djt}}\right]\theta_{dj}^{n_{dj}}\right) \\ &= \prod_{d=1}^{D}\prod_{i=1}^{n_d}\prod_{j=1}^{K}\left[\prod_{t=1}^{V}\beta_{jt}^{z_{dij}w_{dit}}\theta_{dj}^{z_{dij}w_{dit}}\right] \qquad (4–6) \\ &= \prod_{d=1}^{D}\prod_{i=1}^{n_d}\prod_{j=1}^{K}\left[\prod_{t=1}^{V}\big(\beta_{jt}\theta_{dj}\big)^{w_{dit}}\right]^{z_{dij}}, \qquad (4–7) \end{aligned}$$

where Equation 4–6 follows from Equation 4–3. Let $p_{dij} = \prod_{t=1}^{V}\big(\beta_{jt}\theta_{dj}\big)^{w_{dit}}$. By inspection of Equation 4–7 we see immediately that given $(\boldsymbol{\theta}, \boldsymbol{\beta})$,

$$\begin{aligned} &z_{11}, \ldots, z_{1n_1}, z_{21}, \ldots, z_{2n_2}, \ldots, z_{D1}, \ldots, z_{Dn_D} \text{ are all independent,} \\ &z_{di} \sim \mathrm{Mult}_K(p_{di1}, \ldots, p_{diK}). \end{aligned} \tag{4–8}$$

The conditional distribution of $(\boldsymbol{\beta}, \boldsymbol{\theta})$ given by Equation 4–5 can be used, in conjunction with the Griffiths and Steyvers (2004) algorithm, to create a Markov chain on $\boldsymbol{\psi}$ whose invariant distribution is $\nu_{h,\boldsymbol{w}}$: If $\boldsymbol{z}^{(1)}, \boldsymbol{z}^{(2)}, \ldots$ is the Griffiths and Steyvers (2004) chain, then for $l = 1, 2, \ldots$, we generate $(\boldsymbol{\beta}^{(l)}, \boldsymbol{\theta}^{(l)})$ from $p^{(h)}_{\boldsymbol{\theta},\boldsymbol{\beta}\,|\,\boldsymbol{z}}(\cdot\,|\,\boldsymbol{z}^{(l)})$ given by Equation 4–5 and form $(\boldsymbol{z}^{(l)}, \boldsymbol{\beta}^{(l)}, \boldsymbol{\theta}^{(l)})$. We will refer to this Markov chain as the Augmented Collapsed Gibbs Sampler, and use the acronym ACGS. The Griffiths and

Steyvers (2004) chain is uniformly ergodic (Theorem 1 of Chen and Doss (2015)) and an easy argument shows that the resulting ACGS is therefore also uniformly ergodic (and in fact, the rate of convergence of the ACGS is exactly the same as that of the Griffiths and Steyvers (2004) chain; see Diaconis et al. (2008, Lemma 2.4)). The two conditionals given by Equation 4–5 and Equation 4–8 also enable a direct construction of a two-cycle Gibbs sampler that runs on the pair $(\boldsymbol{z}, (\boldsymbol{\beta}, \boldsymbol{\theta}))$. We will refer to this chain as the Full Gibbs Sampler, and use the acronym FGS.

4.2 Comparison of the Full Gibbs Sampler and the Augmented Collapsed Gibbs Sampler

As mentioned earlier, to apply Equation 3–1 we need a Markov chain on the triple $(\boldsymbol{\beta}, \boldsymbol{\theta}, \boldsymbol{z})$, whose invariant distribution is $\nu_{h_*, \boldsymbol{w}}$. The FGS and the ACGS discussed in the last section both have this property. Here we compare their performance.

Before we proceed, we do an empirical check that posterior expectations of certain variables are the same for the two chains. (The purpose of this is to provide an empirical validation that the FGS has the correct invariant distribution.) We do this via the following experiment. For each of the four specifications of the hyperparameter $h = (\eta, \alpha)$ given by $(3, 3)$, $(3, 7)$, $(7, 3)$, and $(7, 7)$, we considered the LDA model for a corpus of 100 documents of 80 words each, drawn from a vocabulary of $V = 20$ words with $K = 2$ topics, and we simulated lines 1–4 of the model. Simulating line 4 gives the data $\boldsymbol{w}$. Using this $\boldsymbol{w}$, for each chain, we ran the chain for 50,000 cycles, deleted the first 10,000 and took every 40^{th} cycle among the remaining 40,000, for a total of 1,000 cycles, which we viewed as effectively independent (this last point is discussed later in this section). For word i in document d, we then have the sequence $z_{di1}^{[\text{FGS},1]}, \ldots, z_{di1}^{[\text{FGS},1000]}$, which records whether the topic from which word i in document d is drawn is topic 1 in the FGS. Similarly, we have the sequence $z_{di1}^{[\text{ACGS},1]}, \ldots, z_{di1}^{[\text{ACGS},1000]}$, in self-explanatory notation. Let p_{di} be the p-value for the two-sample t-test of the null hypothesis that the means of $z_{di1}^{[\text{FGS}]}$ and $z_{di1}^{[\text{ACGS}]}$ are equal. Under the null hypothesis, the distribution of p_{di} is uniform over $(0, 1)$.

Figure 4-1 gives a histogram of these p-values over all the words in all the documents, for each setting of the hyperparameter. Figure 4-2 gives Q-Q plots for the p-values, also for each setting of the hyperparameters. These are plots of the empirical quantiles of the p-values vs. the theoretical quantiles of the uniform distribution. Under the null hypothesis, each plot should be close to a 45° line (this line is also plotted, as a reference). Of course, the plots in Figures 4-1 and 4-2 cannot be the basis for formal inference, since the p-values are dependent; nevertheless, the plots can be useful. For the hyperparameter settings $(3, 3)$ and $(3, 7)$, the histograms and the Q-Q plots are consistent with what we would see for data drawn from a uniform distribution. For the hyperparameter settings $(7, 3)$ and $(7, 7)$, the histograms and Q-Q plots show a deviation from the uniform distribution only in the sense of "granularity." We attribute this to the aforementioned dependence; in particular, p-values for words in the same document are highly correlated. It is not clear why this effect is stronger when η increases. To conclude, we do not believe that the histograms and Q-Q plots provide evidence that the invariant distributions for the FGS and the ACGS are different.

We now wish to compare the mixing rates of the two chains. Diagnostics such as trace plots and auto-correlation functions (ACF's) are often used for this purpose, but unfortunately, the very high dimension of the parameter $\boldsymbol{\psi}$ precludes running the diagnostics for each component in $\boldsymbol{\psi}$. An attractive alternative is to consider, for a chain of length T, the posterior densities $\nu_{h,\boldsymbol{w}}(\boldsymbol{\psi}^{(1)}), \ldots, \nu_{h,\boldsymbol{w}}(\boldsymbol{\psi}^{(T)})$, and run the diagnostics on this sequence (on the log scale); for example, we can compare trace plots of $\log\big(\nu_{h,\boldsymbol{w}}(\boldsymbol{\psi}^{(t)})\big)$, $t = 1, \ldots, T$ for the two chains. The log posterior density is a single univariate quantity, and is known except for a normalizing constant. The fact that we don't know this constant is immaterial, since including this constant would alter the plot only by an additive constant. A trace plot of $\log\big(\nu_{h,\boldsymbol{w}}(\boldsymbol{\psi}^{(t)})\big)$ would reveal whether the chain is spending a considerable amount of time trapped in regions of low posterior probability.

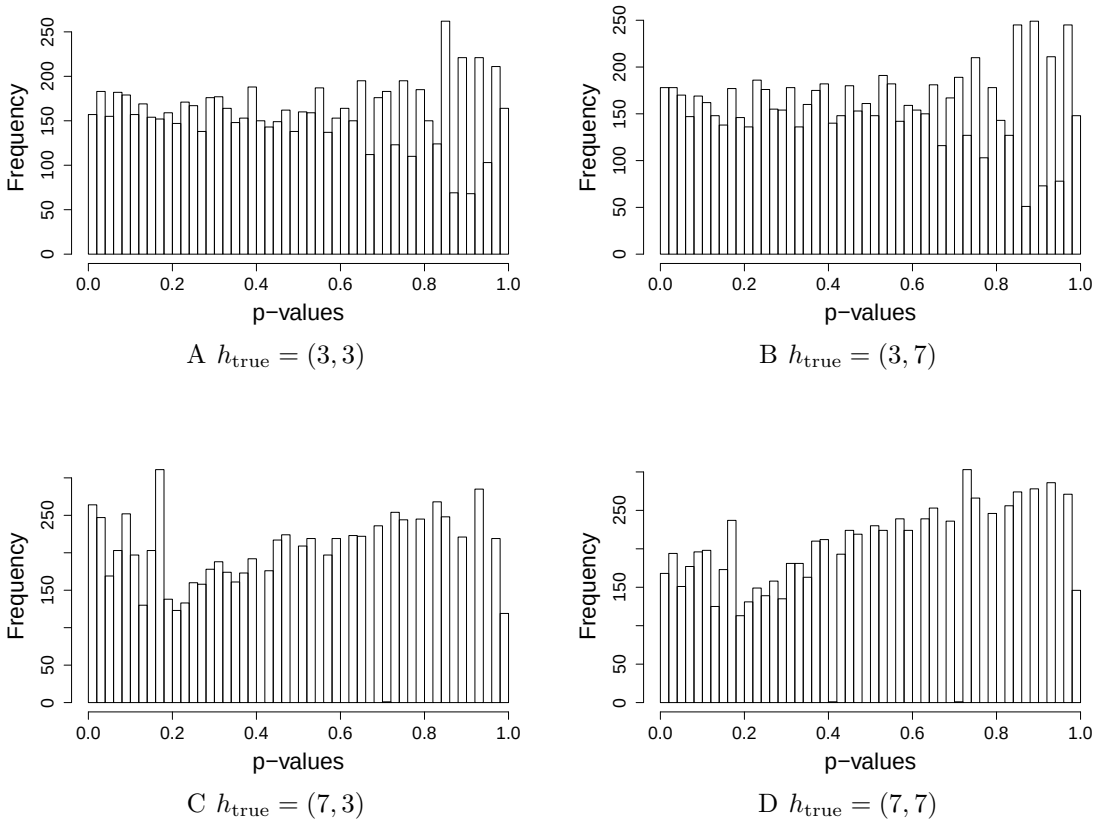
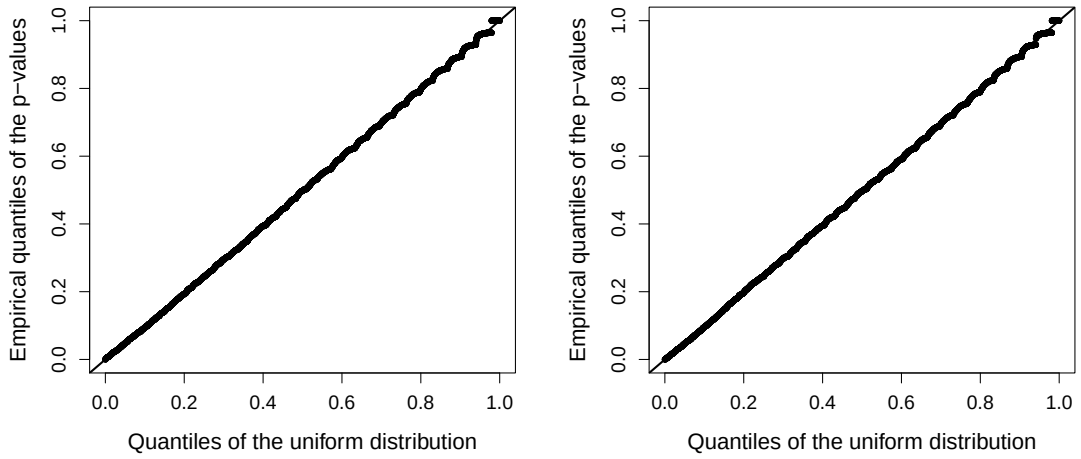
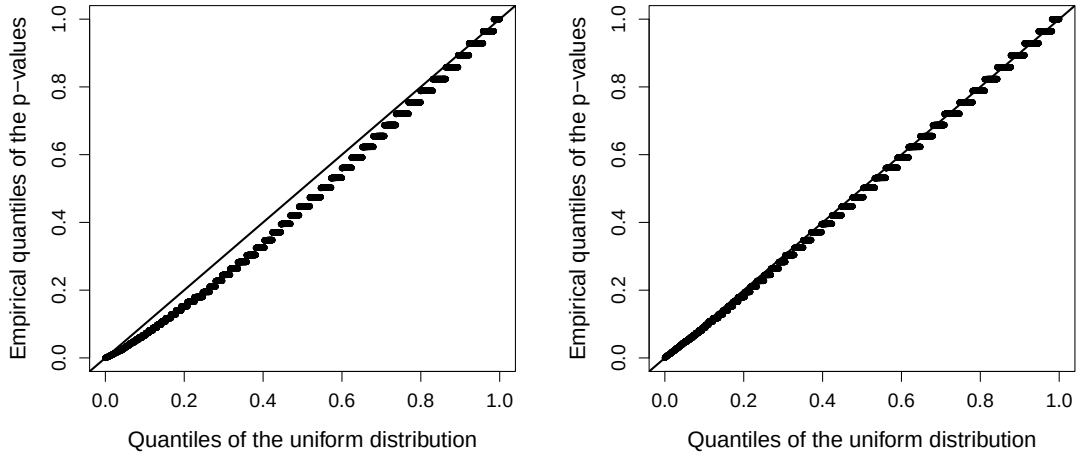


Figure 4-1. Histograms of the p-values over all the words in all the documents, for each setting of the hyperparameter.

Our comparison of the mixing rates of the two chains is conducted as follows. We took the prior on $\boldsymbol{\theta}$ to be a symmetric Dirichlet, so that $\theta_d \overset{\text{iid}}{\sim} \text{Dir}_K(\alpha, \ldots, \alpha)$, fixed $h = (3, 3)$, and considered the LDA model for a corpus of 100 documents of 80 words each, drawn from a vocabulary of $V = 20$ words with $K = 2$ topics; and we simulated lines 1–4 of the model, as before. Using the data $\boldsymbol{w}$, we generated the FGS and the ACGS for 11,000 iterations and deleted the first 1000. The top two panels in Figure 4-3 show trace plots of the log posterior densities for the two chains. The plots suggest that the ACGS mixes faster, although both chains appear to mix adequately. The bottom two panels also suggest that the ACGS mixes faster, although for both chains, iterations separated by a lag of 20 or 30 are essentially uncorrelated. Figure 4-4 shows plots of the ACF's for four variables: θ_{11}, θ_{81}, β_{11}, and β_{17}. There was no particular reason for selecting these two θ's



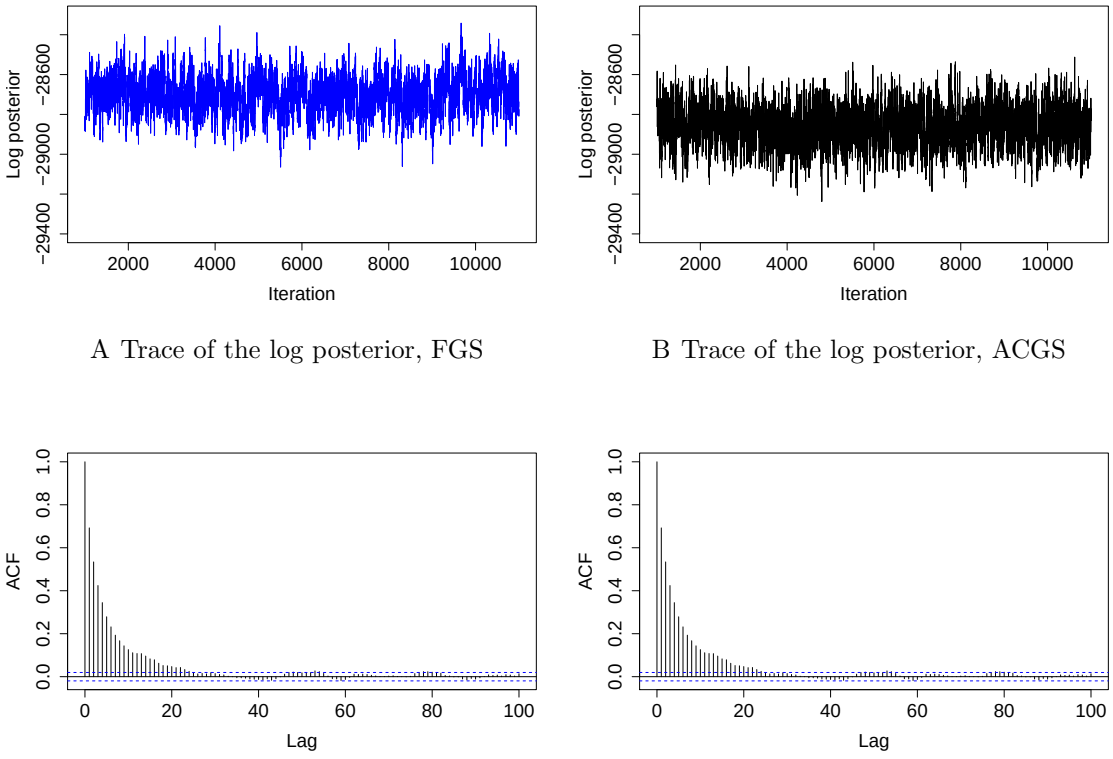
A $h_{\text{true}} = (3, 3)$ B $h_{\text{true}} = (3, 7)$



C $h_{\text{true}} = (7, 3)$ D $h_{\text{true}} = (7, 7)$

Figure 4-2. Q-Q plots for the p-values over all the words in all the documents, for four hyperparameter settings. The plots compare the empirical quantiles of the p-values with the quantiles of the uniform distribution on $(0, 1)$.

and these two β's other than that they are representative of the rest. The figure shows that for the θ's the ACF dies down a bit faster for the ACGS, while for the β's, the ACF's for the two chains die down at about the same rate. To conclude, these limited diagnostics suggest that both chains perform adequately, but that the ACGS has a slight edge.



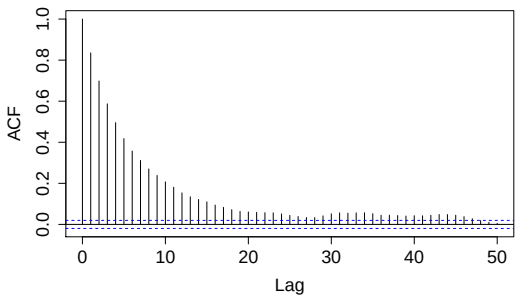
A Trace of the log posterior, FGS

B Trace of the log posterior, ACGS

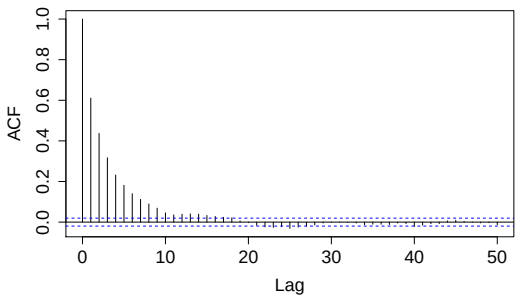
C ACF of the log posterior, FGS

D ACF of the log posterior, ACGS

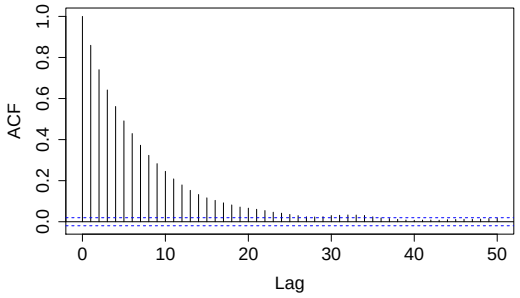
Figure 4-3. Log posterior trace plots (top) and autocorrelation function (bottom) plots of the Full Gibbs Sampler and the Augmented Collapsed Gibbs Sampler, for the hyperparameter $h = (3, 3)$.



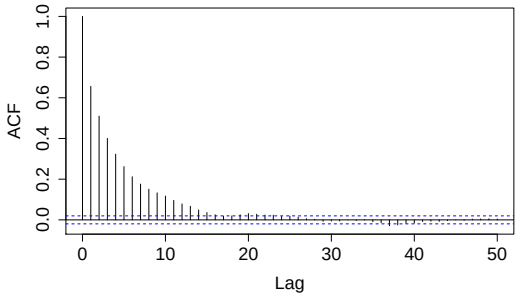
A ACF for θ_{11}, FGS



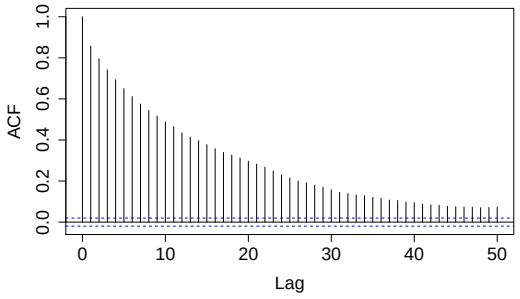
B ACF for θ_{11}, ACGS



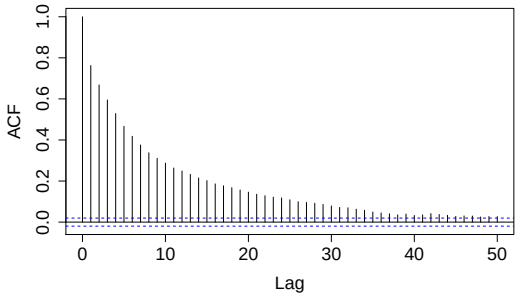
C ACF for θ_{81}, FGS



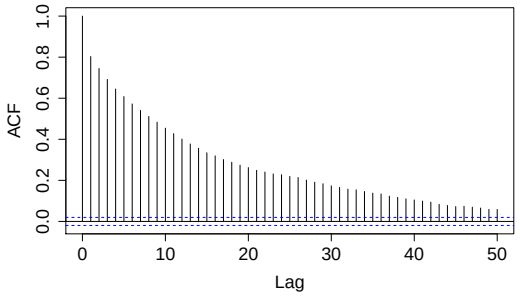
D ACF for θ_{81}, ACGS



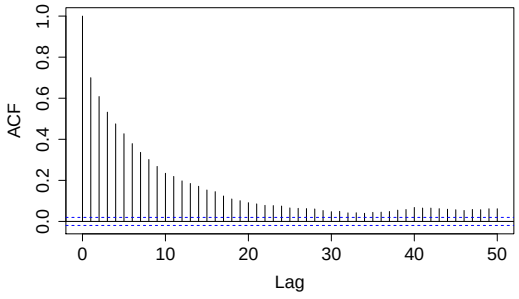
E ACF for β_{11}, FGS



F ACF for β_{11}, ACGS



G ACF for β_{17}, FGS



H ACF for β_{17}, ACGS

Figure 4-4. Autocorrelation functions for selected elements of the $\boldsymbol{\theta}$ and $\boldsymbol{\beta}$ vectors for the Full Gibbs Sampler and the Augmented Collapsed Gibbs Sampler, for the hyperparameter $h = (3, 3)$.

CHAPTER 5
PERFORMANCE OF THE LDA MODEL BASED ON THE EMPIRICAL BAYES CHOICE OF H

We are interested in comparing the performance of the empirical Bayes approach with approaches which use default hyperparameter values. This chapter consists of two parts. In Section 5.1 we first review other methods for choosing the hyperparameter. Then we develop a new criterion for evaluating the performance of the LDA model indexed by a given value of h, and also review an existing criterion. In Section 5.2 we compare, on real data sets, the performance of the LDA model that uses the empirical Bayes choice of h with the performance of LDA models that use other choices of h, using the two criteria discussed in Section 5.1.

5.1 Other Hyperparameter Selection Methods and Criteria for Evaluation

In the literature, the following choices for $h = (\eta, \alpha)$ have been presented: $h_{\mathrm{DG}} = (0.1, 50/K)$, used in Griffiths and Steyvers (2004); $h_{\mathrm{DA}} = (0.1, 0.1)$, used in Asuncion et al. (2009); and $h_{\mathrm{DR}} = (1/K, 1/K)$, used in the `Gensim` topic modeling package (Řehůřek and Sojka, 2010), a well-known package used in the topic modelling community. These choices are ad-hoc, and not based on any particular principle.

Blei et al. (2003) have an approach which deserves special mention. Their goal is to use $\arg\max_h m(h)$, as we do, but their method for doing this is different from ours and, as mentioned in Chapter 2 (Also, see Appendix A for more details), their objective is to estimate $\arg\max_h m(h)$ via the EM algorithm. Very briefly, the general method proceeds as follows. If $h^{(p)}$ is the current estimate of h, the E-step of the EM algorithm is to calculate $E_{h^{(p)}}\left[\log(p_h(\boldsymbol{\psi}, \boldsymbol{w})) \,|\, \boldsymbol{w}\right]$, where $p_h(\boldsymbol{\psi}, \boldsymbol{w})$ is the joint distribution of $(\boldsymbol{\psi}, \boldsymbol{w})$ under the LDA model indexed by h, and the subscript to the expectation indicates that the expectation is taken with respect to $\nu_{h^{(p)},\boldsymbol{w}}$. This step is infeasible because $\nu_{h^{(p)},\boldsymbol{w}}$ is analytically intractable. We consider $\{q_\phi,\ \phi \in \Phi\}$, a (finite-dimensional) parametric family of analytically tractable distributions on $\boldsymbol{\psi}$, and within this family, we find the distribution, say q_{ϕ_*}, which is "closest" to $\nu_{h^{(p)},\boldsymbol{w}}$. Let $Q(h)$

be the expected value of $\log(p_h(\boldsymbol{\psi}, \boldsymbol{w}))$ with respect to q_{ϕ_*}. We view $Q(h)$ as a proxy for $E_{h^{(p)}}\left[\log(p_h(\boldsymbol{\psi}, \boldsymbol{w})) \,|\, \boldsymbol{w}\right]$, and the M-step is then to maximize $Q(h)$ with respect to h, to produce $h^{(p+1)}$. Unfortunately, there are no theoretical results regarding convergence of the sequence $h^{(p)}$ to $\arg\max_h m(h)$.

The implementation of the EM algorithm through variational methods (EM/VM) outlined above describes what Blei et al. (2003) do *conceptually*, but not exactly. Actually, Blei et al. (2003) apply EM/VM to a model that is different from ours. In that model, $\boldsymbol{\beta}$ is viewed as a fixed but unknown parameter, to be estimated, and the latent variable is $\boldsymbol{\vartheta} = (\boldsymbol{\theta}, \boldsymbol{z})$. Thus, the observed and missing data are, respectively, $\boldsymbol{w}$ and $\boldsymbol{\vartheta}$, and the marginal likelihood is a function of two variables, h and $\boldsymbol{\beta}$. Abstractly speaking, the description of EM/VM given above is exactly the same. In principle, EM/VM can be applied to our model also. However, currently there is no algorithm developed for implementing EM/VM on our model, and for this reason we do not compare our method for implementing the empirical Bayes approach with that of Blei et al. (2003). The development of algorithms for implementing EM/VM to our model and the subsequent comparison of our implementation of empirical Bayes with the implementation through EM/VM are clearly of interest, and this is a topic for further work.

Comparison of the Marginal Posterior Distributions of $\boldsymbol{\theta}$ Indexed by Various Choices of h In order to make comparisons, it is necessary to develop a meaningful criterion for evaluating the performance of any given model. Recall that there is a $K \times V$ matrix $\boldsymbol{\beta}$ whose rows, $\beta_1, \ldots, \beta_K$, are each points in $\mathbb{S}_V$; in other words, each of $\beta_1, \ldots, \beta_K$ is a distribution on the vocabulary, i.e. each of $\beta_1, \ldots, \beta_K$ is a topic. Of primary interest are the variables $\theta_1, \ldots, \theta_D$, which are the latent document topic distributions. We imagine that there are K "true" topics for the corpus, $\beta_1^{\text{true}}, \ldots, \beta_K^{\text{true}}$, and that for each document d there is a "true" distribution over the topics, which we will denote θ_d^{true}.

Recall also that $\nu_{h,\boldsymbol{w}}$ is the posterior distribution of $(\boldsymbol{\beta}, \boldsymbol{\theta}, \boldsymbol{z})$ corresponding to the prior ν_h. This posterior distribution induces a $\boldsymbol{\theta}$-marginal distribution on $\boldsymbol{\theta}$ which we will

denote by $\nu_{h,\boldsymbol{w},\boldsymbol{\theta}}$. For a given value of h, we can evaluate the performance of the LDA model indexed by h by calculating a distance between $\nu_{h,\boldsymbol{w},\boldsymbol{\theta}}$ and $\delta_{\boldsymbol{\theta}^{\text{true}}}$, where $\delta_{\boldsymbol{\theta}^{\text{true}}}$ is the point mass at the vector $\boldsymbol{\theta}^{\text{true}} = (\theta_1^{\text{true}}, \ldots, \theta_D^{\text{true}})$. Values of h for which this distance is small are to be preferred.

To lighten the notation, we will use the following: $\pi_{\text{EB}} = \nu_{\hat{h},\boldsymbol{w},\boldsymbol{\theta}}$, the marginal posterior distribution of $\boldsymbol{\theta}$ under our empirical Bayes choice of h; $\pi_{\text{DG}} = \nu_{h_{\text{DG}},\boldsymbol{w},\boldsymbol{\theta}}$, $\pi_{\text{DA}} = \nu_{h_{\text{DA}},\boldsymbol{w},\boldsymbol{\theta}}$, and $\pi_{\text{DR}} = \nu_{h_{\text{DR}},\boldsymbol{w},\boldsymbol{\theta}}$, the marginal posterior distributions of $\boldsymbol{\theta}$ corresponding to the default values h_{DG}, h_{DA}, and h_{DR} respectively. To measure the discrepancy between π_{EB} and $\delta_{\boldsymbol{\theta}^{\text{true}}}$ we may use any of the conventional distances between probability distributions, such as the Kolmogorov-Smirnov distance, or the Cramér-von Mises distance; particularly appropriate is the distance $\rho_1(\pi_{\text{EB}}, \delta_{\boldsymbol{\theta}^{\text{true}}})$ given by an integral as follows:

$$\rho_1(\pi_{\text{EB}}, \delta_{\boldsymbol{\theta}^{\text{true}}}) \coloneqq I_{\text{EB}} \coloneqq \int_{\mathbb{S}_K^D} [\pi_{\text{EB}}(\boldsymbol{\theta}) - \delta_{\boldsymbol{\theta}^{\text{true}}}(\boldsymbol{\theta})]^2 \, dU(\boldsymbol{\theta}), \tag{5–1}$$

where π_{EB} and $\delta_{\boldsymbol{\theta}^{\text{true}}}$ are now viewed as multivariate cumulative distribution functions, and U is the uniform distribution on the product set $\mathbb{S}_K^D$, i.e. U is the product measure $\text{Dir}_K(1, \ldots, 1) \times \cdots \times \text{Dir}_K(1, \ldots, 1)$ (a D-fold product). We define I_{DG}, I_{DA}, and I_{DR} similarly.

If we wish to use integral distances of the type given by Equation 5–1 in order to evaluate the performance of the LDA models indexed by the four hyperparameter choices, we now face two problems, each of which we state and then discuss.

The integrals I_{EB}, I_{DG}, I_{DA}, and I_{DR} are not available in closed form Consider for example I_{EB} given by Equation 5–1. In principle, we could estimate this integral by a double Monte Carlo study: we choose $\boldsymbol{\theta}_1, \ldots, \boldsymbol{\theta}_N \overset{\text{iid}}{\sim} U$, and for each $i = 1, \ldots, N$, we obtain an estimate $\hat{\pi}_{\text{EB}}(\boldsymbol{\theta}_i)$ of $\pi_{\text{EB}}(\boldsymbol{\theta}_i)$ via MCMC. We then estimate I_{EB} via $(1/N)\sum_{i=1}^N [\hat{\pi}_{\text{EB}}(\boldsymbol{\theta}_i) - \delta_{\boldsymbol{\theta}^{\text{true}}}(\boldsymbol{\theta}_i)]^2$. Unfortunately, this is computationally too demanding, and therefore not feasible in practice.

If we take advantage of the fact that we are trying to measure the distance between π_{EB} and a point mass distribution, then there is a sensible alternative. Define

$$\rho_2(\pi_{\text{EB}}, \delta_{\boldsymbol{\theta}^{\text{true}}}) = \int_{\mathbb{S}_K^D} \|\boldsymbol{\theta} - \boldsymbol{\theta}^{\text{true}}\|_1 \, d\pi_{\text{EB}}(\boldsymbol{\theta}),$$

where $\|\cdot\|_1$ is the L_1 norm on $\mathbb{S}_K^D$. Suppose that $\boldsymbol{\psi}_1, \ldots, \boldsymbol{\psi}_S$ is the initial segment of a Markov chain with invariant distribution $\nu_{\hat{h},\boldsymbol{w}}$ (the chain can be either the FGS or the ACGS). Here, $\boldsymbol{\psi}_i = (\boldsymbol{\beta}^{(i)}, \boldsymbol{\theta}^{(i)}, \boldsymbol{z}^{(i)})$. We may estimate $\rho_2(\pi_{\text{EB}}, \delta_{\boldsymbol{\theta}^{\text{true}}})$ simply by

$$\hat{\rho}_2(\pi_{\text{EB}}, \delta_{\boldsymbol{\theta}^{\text{true}}}) = \frac{1}{S} \sum_{s=1}^{S} \|\boldsymbol{\theta}^{(s)} - \boldsymbol{\theta}^{\text{true}}\|_1. \tag{5–2}$$

This quantity is not taxing to compute, and in our experience, the results obtained from using ρ_2 and ρ_1 are approximately the same. Therefore, we will use the measure ρ_2 as our criterion for measuring the distances between each of π_{EB}, π_{DG}, π_{DA}, π_{DR}, and $\delta_{\boldsymbol{\theta}^{\text{true}}}$.

The variables $\boldsymbol{\theta}^{(s)}$ in Equation 5–2 and $\boldsymbol{\theta}^{\text{true}}$ are both points in $\mathbb{S}_K^D$, but have different interpretations Consider any of the choices of h, say h_{DG}, to be specific. The Markov chain with invariant distribution $\nu_{h_{\text{DG}},\boldsymbol{w}}$ gives us a sequence $(\boldsymbol{\beta}^{(1)}, \boldsymbol{\theta}^{(1)}, \boldsymbol{z}^{(1)}), \ldots, (\boldsymbol{\beta}^{(S)}, \boldsymbol{\theta}^{(S)}, \boldsymbol{z}^{(S)})$. Consider the component $\theta_d^{(s)}$ of $\boldsymbol{\theta}^{(s)}$. While both $\theta_d^{(s)}$ and θ_d^{true} are points in $\mathbb{S}_K$, their interpretations are different: $\theta_d^{(s)}$ is a distribution on the K topics $\beta_1^{(s)}, \ldots, \beta_K^{(s)}$, while θ_d^{true} is a distribution on the K topics $\beta_1^{\text{true}}, \ldots, \beta_K^{\text{true}}$, and these are different sets of topics.

Loosely speaking, according to standard statistical principles, if n_d is large so that we have a lot of "information," then with high probability, the topic variables $\beta_1^{(s)}, \ldots, \beta_1^{(s)}$ should be close to $\beta_1^{\text{true}}, \ldots, \beta_K^{\text{true}}$ (possibly after re-ordering). For the distance between $\theta_d^{(s)}$ and θ_d^{true} to be meaningful, it is necessary to "align" these sets of topics.

We do this as follows. We assume that we know the *labels* for the K topics in our corpus. For example, if the corpus is a set of articles from the *New York Times*, the labels might be $L_1 =$ Sports, $L_2 =$ Medicine, $L_3 =$ Politics, $L_4 =$ Health, etc. We also assume that we know the topic labels for each document in the corpus (the corpus on which we will compare the different LDA models could be, for example, all articles over

a certain period of time from the Sports and Medicine sections of the *New York Times*, in which case we automatically know the labels for each document). While the labels might be known, the topics themselves are not known. A standard way to estimate them is through the term frequency matrix defined as follows. Let $\{L_1, \ldots, L_K\}$ be the set of topic labels for the corpus. For each $j = 1, \ldots, K$ and each term t in the vocabulary, we record the term frequency tf_{jt}, which is the number of times term t appears in the group of documents assigned to topic label L_j. We can then form the $K \times V$ term frequency matrix. If we normalize each row to sum to 1, then each row becomes a point in $\mathbb{S}_V$, i.e. a topic. The normalized rows are then taken to be the true topics $\beta_1^{\text{true}}, \ldots, \beta_K^{\text{true}}$. We can now align $\beta_1^{(s)}, \ldots, \beta_K^{(s)}$ and $\beta_1^{\text{true}}, \ldots, \beta_K^{\text{true}}$ as follows. For each $j = 1, \ldots, K$, let

$$j\prime = \underset{l \in \{1,\ldots,K\}}{\arg\min} \|\beta_j^{(s)} - \beta_l^{\text{true}}\|_1. \tag{5–3}$$

For $j = 1, \ldots, K$, topic $\beta_j^{(s)}$ is now aligned with $\beta_{j\prime}^{\text{true}}$. In order to compare the K-vectors $\theta_d^{(s)}$ and θ_d^{true} via the L_1 norm, we first redefine $\theta_d^{(s)}$ as follows. For each $j = 1, \ldots, K$, the mass $\theta_{dj}^{(s)}$ of cell j of the vector $\theta_d^{(s)}$ is assigned to cell $j\prime$, where $j\prime$ is calculated in Equation 5–3. (We note that the map $j \to j\prime$ may or may not be a 1-1 map, but whether or not it is 1-1 is immaterial.) Here is an example. Suppose that $K = 4$, and suppose that originally, i.e. before the alignment, $\theta_d^{(s)} = (p_1, p_2, p_3, p_4)$, where the p's sum to 1. And suppose that $\beta_1^{(s)}$ and $\beta_2^{(s)}$ are aligned with β_2^{true}, and that $\beta_3^{(s)}$ and $\beta_4^{(s)}$ are aligned with β_3^{true}. Then $\theta_d^{(s)}$ should be redefined as $(0, p_1 + p_2, p_3 + p_4, 0)$, and then compared with θ_d^{true}.

Posterior Predictive Checking (PPC) PPC is a Bayesian model checking method which uses a score that is inversely related to the so-called "perplexity" score which is sometimes used in the machine learning literature. When applied to the LDA context, the method is described as follows. For $d = 1, \ldots, D$, let $\boldsymbol{w}_{(-d)}$ denote the corpus consisting of all the documents except for document d. To evaluate a given model (in our case the LDA model indexed by a given h) through posterior predictive checking, in essence we see how well the model based on $\boldsymbol{w}_{(-d)}$ predicts document d, the held-out document. We do this for

$d = 1, \ldots, D$, and take the geometric mean. We formalize this as follows. The predictive likelihood of h for the held-out document is

$$L_d(h) = \int \ell_{\boldsymbol{w}_d}(\boldsymbol{\psi})\, d\nu_{h,\boldsymbol{w}_{(-d)}}(\boldsymbol{\psi}), \tag{5–4}$$

where $\ell_{\boldsymbol{w}_d}(\boldsymbol{\psi})$ is the likelihood of $\boldsymbol{\psi}$ for the held-out document d, and $\nu_{h,\boldsymbol{w}_{(-d)}}$ is the posterior distribution of $\boldsymbol{\psi}$ given $\boldsymbol{w}_{(-d)}$. We form the score $S(h) = \left[\prod_{d=1}^{D} L_d(h)\right]^{1/D}$. Two different values of hyperparameter h are compared via their scores. Unfortunately, calculation of $S(h)$ is computationally extremely demanding. In the machine learning literature, $L_d(h)$ is often estimated by $\ell_{\boldsymbol{w}_d}(\widehat{\boldsymbol{\psi}})$, where $\widehat{\boldsymbol{\psi}}$ is a single point estimate that "summarizes the distribution $\nu_{h,\boldsymbol{w}_{(-d)}}$" in some sense. Approximations of this sort can be woefully inadequate. Conceptually, it is easy to estimate $L_d(h)$ by direct Monte Carlo: let $\boldsymbol{\psi}_1, \boldsymbol{\psi}_2, \ldots$ be an ergodic Markov chain with invariant distribution $\nu_{h,\boldsymbol{w}_{(-d)}}$. We then approximate the integral by $(1/n)\sum_{i=1}^{n} \ell_{\boldsymbol{w}_d}(\boldsymbol{\psi}_i)$. Care needs to be exercised, however, because in Equation 5–4, the variable $\boldsymbol{\psi}$ in the term $\ell_{\boldsymbol{w}_d}(\boldsymbol{\psi})$ has a dimension that is different than that of the variable $\boldsymbol{\psi}$ in the rest of the integral. Chen (2015) gives a careful description of a Monte Carlo scheme for estimating the integral in Equation 5–4.

5.2 Comparison on Real Datasets

Here we compare the performance of LDA models based on various choices of the hyperparameter, on several corpora of real documents. As we will soon see, for the corpora that we use, the true topic distributions are, for practical purposes, known, and this enables us to evaluate the various models. We created two sets of document corpora, one from the 20Newsgroups dataset[1] , and the other from the English Wikipedia. The 20Newsgroups dataset is commonly used in the machine learning literature for experiments on applications of text classification and clustering algorithms. It contains approximately 20,000 articles that are partitioned relatively evenly across 20 different newsgroups or

[1] http://qwone.com/~jason/20Newsgroups

categories. We created the second set of corpora from web articles downloaded from the English Wikipedia, with the help of the MediaWiki API[2] .

We created the 20Newsgroups corpora as follows. We formed five subsets of the 20Newsgroups dataset, which we call C-1–C-5, with the feature that the articles within the subsets are increasingly difficult to distinguish: for corpus C-1 the topics for the different articles are very different, and for corpus C-5 the topics for the different articles are similar. For each article, we took its true topic label to be the newsgroup to which the article is assigned. Thus, for corpora C-1–C-5, it becomes increasingly difficult to place the articles into the correct newsgroup. We built corpus C-1 from a random subset of articles from the 20Newsgroups categories Medicine, Christianity, and Baseball; these three categories are highly unrelated and easily recognizable from article texts. We built corpus C-2 from a random subset of articles from the categories Automobiles, Motorcycles, Baseball, and Hockey (all four of these categories are classified under the super-category Recreation in the 20Newsgroups dataset), and we built corpus C-3 from a random subset of articles from the categories Cryptography, Electronics, Medicine, and Space (all four of these categories are classified under the super-category Science in the 20Newsgroups dataset). Compared to the categories in Corpus C-1, the categories in corpora C-2 and C-3 are moderately related. Lastly, we created corpus C-4 using articles under the categories Autos and Motorcycles, and corpus C-5 using articles under the categories PC Hardware and Mac Hardware. In corpora C-4 and C-5, the corresponding categories are closely related to each other and hard to distinguish from article texts.

We created the Wikipedia corpora as follows. When a Wikipedia article is created, it is typically tagged to one or more categories, one of which is the "primary category." For each article, we took its true topic label to be the primary category label for the article. We created corpus C-6 from a subset of the Wikipedia articles under the categories

[2] http://www.mediawiki.org/wiki/API:Query

Leopardus, *Lynx*, and *Prionailurus* and corpus C-7 from a subset of the Wikipedia articles under the categories *Acinonyx*, *Leopardus*, *Prionailurus*, and *Puma*. All the categories of corpora C-6 and C-7 are part of the Wikipedia super-category *Felines*. We created corpus C-8 from a subset of the Wikipedia articles under the categories *Coyotes*, *Jackals*, and *Wolves*. All the three categories of corpus C-8 are under the Wikipedia super-category *Canis*. Finally, we created Corpus C-9 from a subset of the Wikipedia articles under the categories *Eagles*, *Falco (genus)*, *Falconry*, *Falcons*, *Harriers*, *Hawks*, *Kites*, and *Owls*. All eight categories of corpus C-9 are subcategories of the Wikipedia category *Birds of Prey*. For each of the four Wikipedia corpora that we created, the categories of the articles are closely related to each other, and fairly hard to distinguish from article texts.

Table 5-1 gives some information on the nine corpora we created. In the table, the column labeled V gives the vocabulary size for each corpus, and the column labeled Categories gives newsgroup categories for each 20Newsgroup corpus, and Wikipedia categories for each Wikipedia corpus. The numbers shown in parentheses next to the category names are the number of documents associated with the corresponding categories. For each corpus, we took the number of topics K to be equal to the number of categories for the corpus.

Table 5-1. Corpora created from the 20Newsgroups dataset and the Wikipedia pages.

Corpus	Categories	V
C-1	sci.med (50), soc.religion.christian (50), rec.sport.baseball (50)	807
C-2	rec.autos (50), rec.motorcycles (50), rec.sport.baseball (50), rec.sport.hockey (50)	1,061
C-3	sci.crypt (50), sci.electronics (50), sci.med (50), sci.space (50)	1,033
C-4	rec.autos (50), rec.motorcycles (50)	488
C-5	comp.sys.ibm.pc.hardware (50), comp.sys.mac.hardware (50)	502
C-6	Leopardus (8), Lynx (8), Prionailurus (7)	303
C-7	Acinonyx (6), Leopardus (8), Prionailurus (7), Puma (8)	622
C-8	Coyotes (7), Jackals (7), Wolves (8)	447
C-9	Eagles (62), Falco (genus) (45), Falconry (52), Falcons (10), Harriers (21), Hawks (16), Kites (22), Owls (76)	1,369

We identified "true" topics for each corpus as follows. As described in the previous section, for each corpus, we computed the $K \times V$ term frequency matrix, and normalized each row to sum to 1, thereby obtaining our estimates $\beta_1^{\text{true}}, \ldots, \beta_K^{\text{true}}$ of the true topics, and the resulting $K \times V$ matrix $\boldsymbol{\beta}^{\text{true}}$. For each article d, we took its true distribution on topics $\theta_d^{\text{true}} \in \mathbb{S}_K$ to be the index vector with a 1 at the j^{th} element, where j is the index of article d's category.

Before we proceed, we compare the complexity of the nine created corpora as follows. For each corpus, for $j, j\prime \in \{1, 2, \ldots, K\}$, we computed $\|\beta_j^{\text{true}} - \beta_{j\prime}^{\text{true}}\|_2$, where $\|.\|_2$ is the L_2 norm on $\mathbb{S}_V$. A small value of $\|\beta_j^{\text{true}} - \beta_{j\prime}^{\text{true}}\|_2$ indicates that topics β_j^{true} and $\beta_{j\prime}^{\text{true}}$ are close, and if the norms $\|\beta_j^{\text{true}} - \beta_{j\prime}^{\text{true}}\|_2$ are small for all $j, j\prime \in \{1, 2, \ldots, K\}$, then we consider the corpus to be "complex," in the sense that it is difficult to cluster the documents in the corpus based on the document texts. Table 5-2 gives the average of the $\binom{K}{2}$ L_2 distances for each of the nine corpora. Figure 5-1 gives plots of these L_2 norms on the $K \times K$ grid, for all nine corpora. As can be seen from the plots, corpora C-5 and C-4 are the most complex and C-8 is the least complex, based on the average L_2 norms of the corpus topic pairs.

Table 5-2. Sorted values of the averages of the $\binom{K}{2}$ L_2 distances $\|\beta_j^{\text{true}} - \beta_{j\prime}^{\text{true}}\|_2$, $j, j\prime = 1, \ldots, K$, for the nine corpora.

Corpus	Average inter-topic L_2 distance
C-5	.02561
C-4	.02883
C-3	.03690
C-2	.04116
C-1	.04435
C-9	.07544
C-6	.07557
C-7	.08548
C-8	.11201

For each corpus, we computed (i) the estimate $\widehat{M}_\zeta(h)$ for h over a grid, using the method described in Chapter 3, and (ii) an estimate of the standard error of $\widehat{M}_\zeta(h)$ for

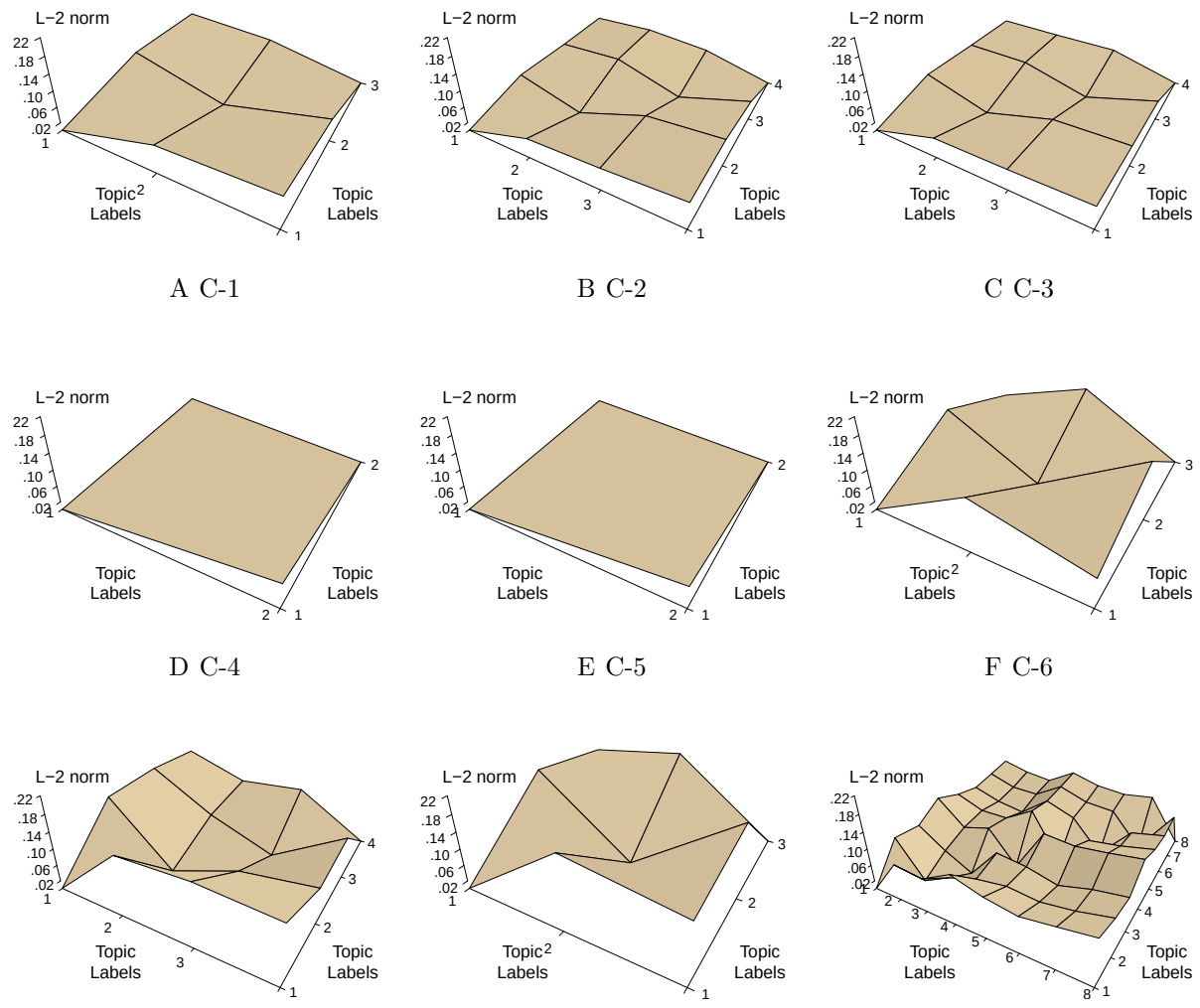
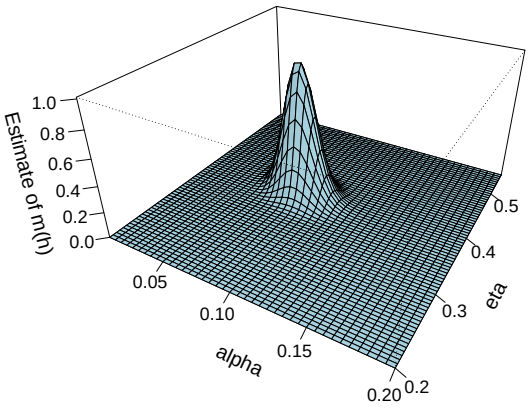


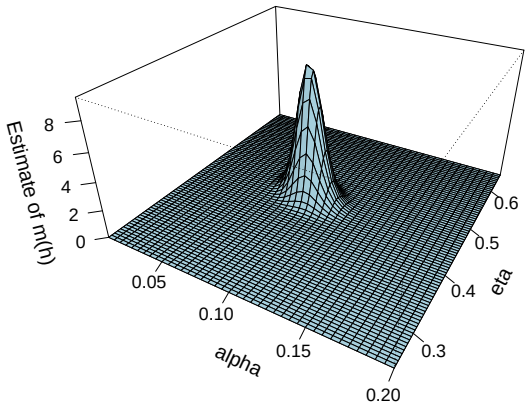
Figure 5-1. Plots of L_2 norms between the true topic distributions, for all nine corpora.

each h in the grid. Details on how these computations were done are given at the end of this section. Figures 5-2 and 5-4 show plots of $\widehat{M}_\zeta(h)$, and also give $\hat{\hat{h}} = \arg\max_h \widehat{M}_\zeta(h)$, for the nine corpora. Figures 5-3 and 5-5 give plots of the standard errors of $\widehat{M}_\zeta(h)$ for the nine corpora, and these indicate that the accuracy of $\widehat{M}_\zeta(h)$ is acceptable over the entire h-range for all nine cases.

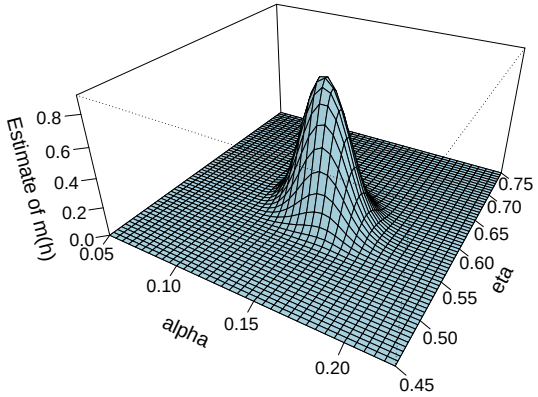
Table 5-3 gives the L_2 distances between the three default hyperparameter choices $h_{\mathrm{DR}} = (1/K, 1/K)$, $h_{\mathrm{DA}} = (.1, .1)$, and $h_{\mathrm{DG}} = (.1, 50/K)$, and the empirical Bayes choice $\hat{\hat{h}}$,



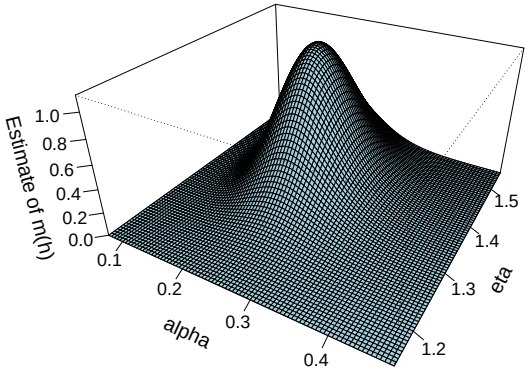
A C-1: $\hat{\hat{h}} = (.385, .085)$



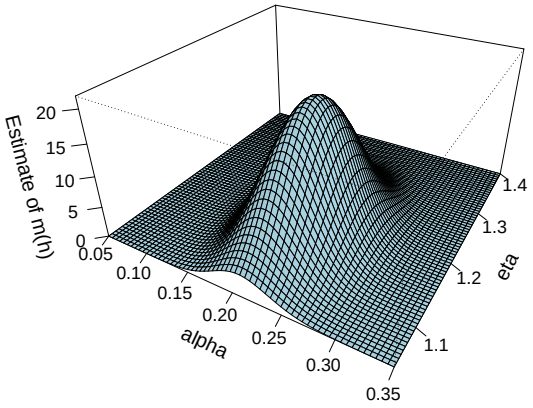
B C-2: $\hat{\hat{h}} = (.460, .090)$



C C-3: $\hat{\hat{h}} = (.585, .145)$

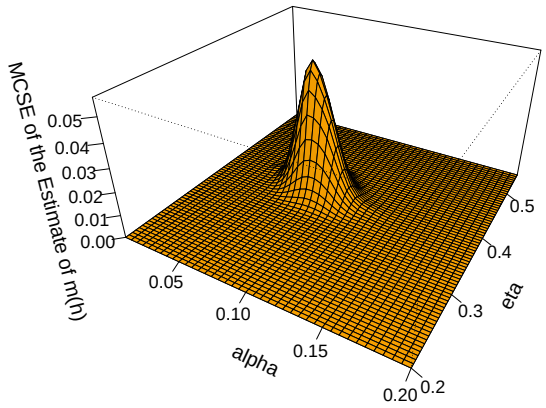


D C-4: $\hat{\hat{h}} = (1.425, .225)$

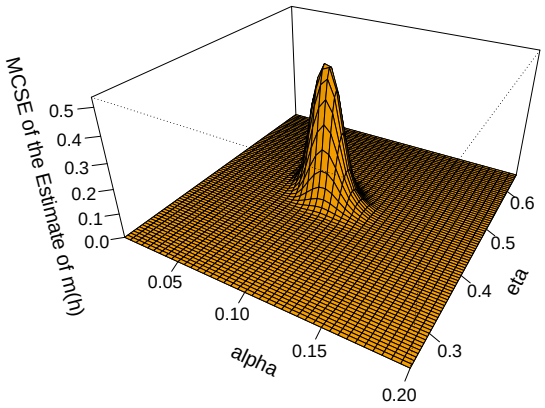


E C-5: $\hat{\hat{h}} = (1.165, .225)$

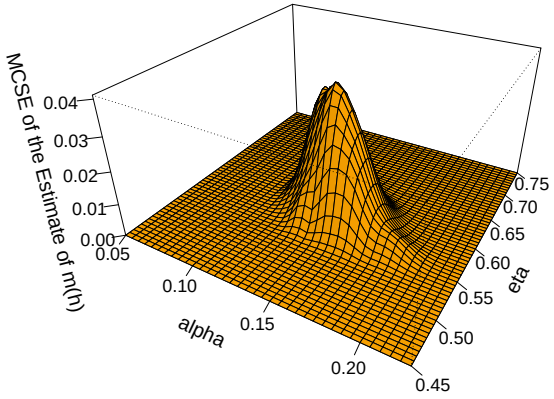
Figure 5-2. Plots of $\hat{M}(h)$ for the five 20Newsgroups corpora.



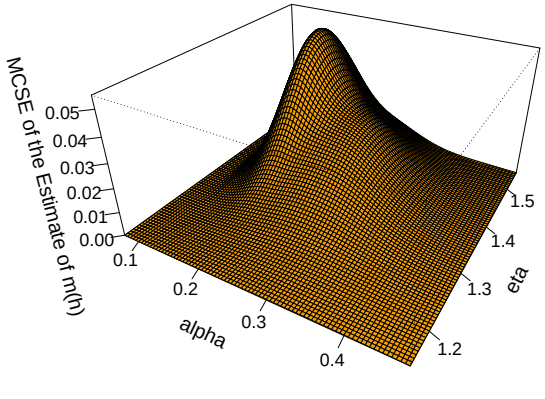
A C-1: $\hat{\hat{h}} = (.385, .085)$



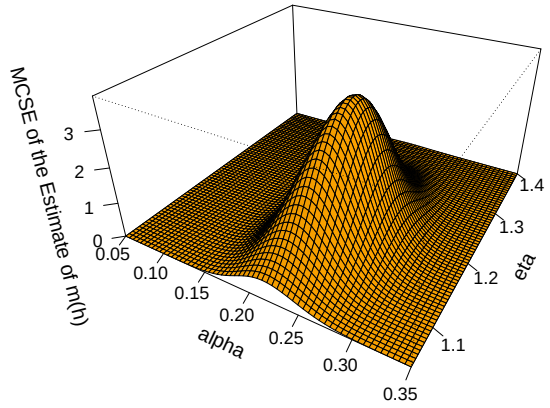
B C-2: $\hat{\hat{h}} = (.460, .090)$



C C-3: $\hat{\hat{h}} = (.585, .145)$

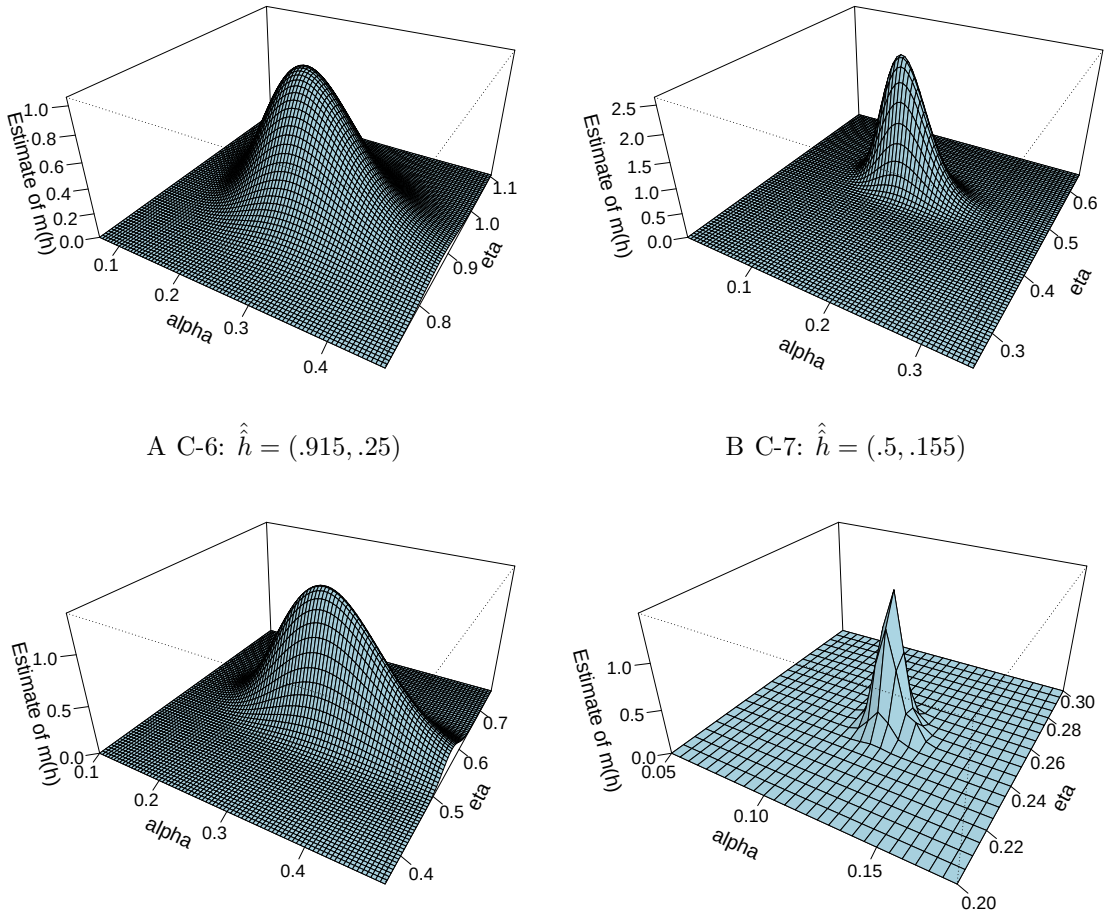


D C-4: $\hat{\hat{h}} = (1.425, .225)$



E C-5: $\hat{\hat{h}} = (1.165, .225)$

Figure 5-3. Monte Carlo standard error (MCSE) of $\hat{M}(h)$ for the five 20Newsgroups corpora.



A C-6: $\hat{\hat{h}} = (.915, .25)$

B C-7: $\hat{\hat{h}} = (.5, .155)$

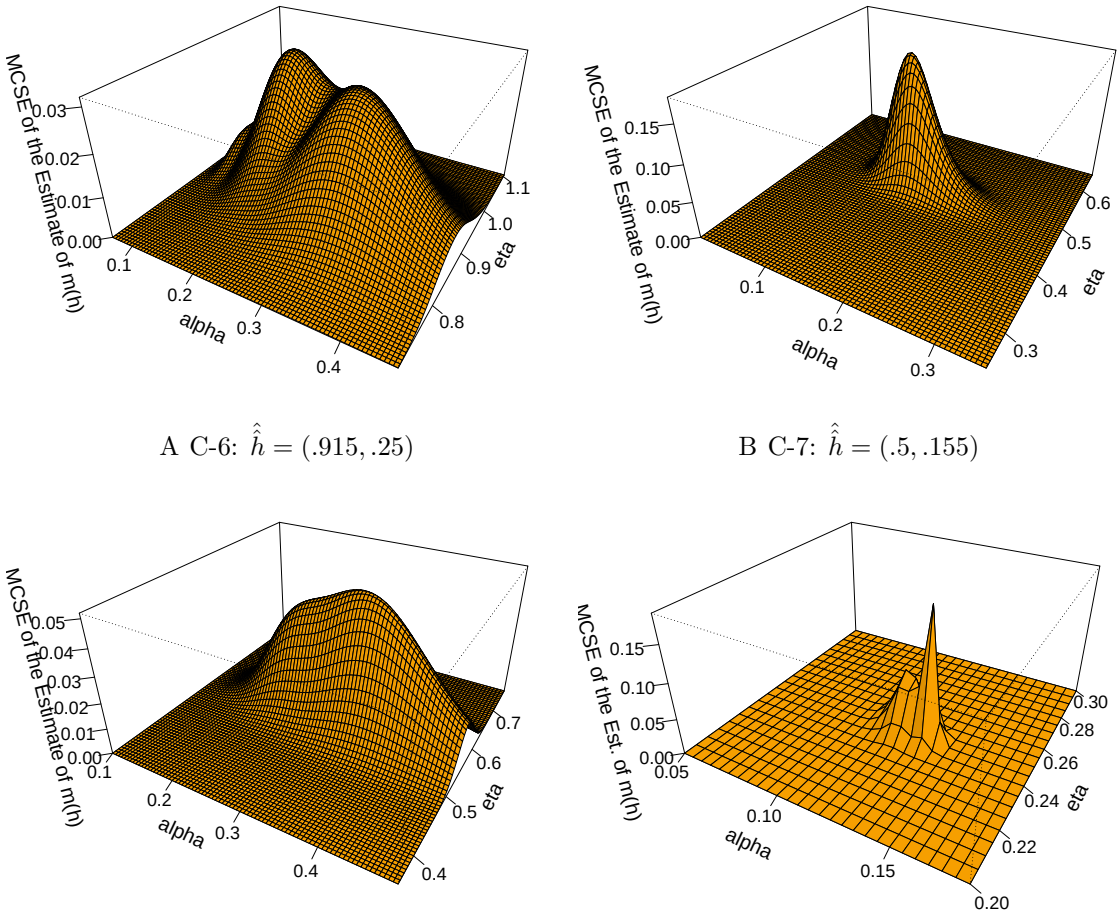
C C-8: $\hat{\hat{h}} = (.57, .31)$

D C-9: $\hat{\hat{h}} = (.250, .120)$

Figure 5-4. Plots of $\hat{M}(h)$ for corpora C-6, C-7, C-8, and C-9.

for all nine corpora (these L_2 distances are on $(0, \infty)^2$). The table shows that the default choice h_{DG} is far from the empirical Bayes choice $\hat{\hat{h}}$ in all cases, and the default choice h_{DR} is fairly close to the empirical Bayes choice $\hat{\hat{h}}$, on average.

Table 5-4 compares the performance of the LDA models indexed by $\hat{\hat{h}}$, h_{DR}, h_{DA}, and h_{DG}, for corpora C-1–C-9, using the evaluation criterion developed in Section 5.1. The table gives the ratios $\hat{\rho}_2(\pi_{\mathrm{DR}}, \delta_{\boldsymbol{\theta}^{\mathrm{true}}})/\hat{\rho}_2(\pi_{\mathrm{EB}}, \delta_{\boldsymbol{\theta}^{\mathrm{true}}})$, $\hat{\rho}_2(\pi_{\mathrm{DA}}, \delta_{\boldsymbol{\theta}^{\mathrm{true}}})/\hat{\rho}_2(\pi_{\mathrm{EB}}, \delta_{\boldsymbol{\theta}^{\mathrm{true}}})$, and $\hat{\rho}_2(\pi_{\mathrm{DG}}, \delta_{\boldsymbol{\theta}^{\mathrm{true}}})/\hat{\rho}_2(\pi_{\mathrm{EB}}, \delta_{\boldsymbol{\theta}^{\mathrm{true}}})$, for all nine corpora. From the table, we see that the empirical Bayes model outperforms all the other models uniformly across all nine corpora.



A C-6: $\hat{\hat{h}} = (.915, .25)$

B C-7: $\hat{\hat{h}} = (.5, .155)$

C C-8: $\hat{\hat{h}} = (.57, .31)$

D C-9: $\hat{\hat{h}} = (.250, .120)$

Figure 5-5. Monte Carlo standard error (MCSE) of $\hat{M}(h)$ for corpora C-6, C-7, C-8, and C-9.

Table 5-3. L_2 distances between the default hyperparameter choices h_{DR}, h_{DA}, and h_{DG}, and the empirical Bayes choice $\hat{\hat{h}}$, for the nine corpora.

Corpus	$\|\hat{\hat{h}} - h_{\mathrm{DR}}\|_2$	$\|\hat{\hat{h}} - h_{\mathrm{DA}}\|_2$	$\|\hat{\hat{h}} - h_{\mathrm{DG}}\|_2$
C-1	.254	.285	16.584
C-2	.264	.360	12.415
C-3	.351	.487	12.360
C-4	.965	1.331	24.810
C-5	.719	1.072	24.797
C-6	.587	.829	16.436
C-7	.267	.403	12.351
C-8	.237	.514	16.363
C-9	.125	.151	6.132

Table 5-4. Estimates of the discrepancy ratios $D(h_{\mathrm{DR}}) \coloneqq \rho_2(\pi_{\mathrm{DR}}, \delta_{\boldsymbol{\theta}^{\mathrm{true}}})/\rho_2(\pi_{\mathrm{EB}}, \delta_{\boldsymbol{\theta}^{\mathrm{true}}})$, $D(h_{\mathrm{DA}}) \coloneqq \rho_2(\pi_{\mathrm{DA}}, \delta_{\boldsymbol{\theta}^{\mathrm{true}}})/\rho_2(\pi_{\mathrm{EB}}, \delta_{\boldsymbol{\theta}^{\mathrm{true}}})$, and $D(h_{\mathrm{DG}}) \coloneqq \rho_2(\pi_{\mathrm{DG}}, \delta_{\boldsymbol{\theta}^{\mathrm{true}}})/\rho_2(\pi_{\mathrm{EB}}, \delta_{\boldsymbol{\theta}^{\mathrm{true}}})$, for all nine corpora, where $h_{\mathrm{DR}} = (1/K, 1/K)$, $h_{\mathrm{DA}} = (.1, .1)$, and $h_{\mathrm{DG}} = (.1, 50/K)$. The discrepancy is smallest for the empirical Bayes model, uniformly across all nine corpora.

Corpus	$D(h_{\mathrm{DR}})$	$D(h_{\mathrm{DA}})$	$D(h_{\mathrm{DG}})$
C-1	1.09	1.19	1.41
C-2	1.21	1.33	1.48
C-3	1.21	1.38	1.84
C-4	1.20	1.40	1.58
C-5	1.09	1.13	1.43
C-6	1.85	2.55	2.46
C-7	1.88	2.18	2.62
C-8	1.18	1.42	2.02
C-9	1.04	1.04	1.21

We now compare the performance of the LDA models indexed by $\hat{\hat{h}}$, h_{DR}, h_{DA}, and h_{DG} for corpora C-1 to C-9, using the estimate of the posterior predictive score $S(h)$, which we denote by $\widehat{S}(h)$, described in Section 5.1. To compute $\widehat{S}(h)$ for a corpus, for every held-out document, we used a full Gibbs sampling chain of length 2,000, after discarding a short burn-in period. Table 5-5 gives the ratios $\widehat{S}(h_{\mathrm{DR}})/\widehat{S}(\hat{\hat{h}})$, $\widehat{S}(h_{\mathrm{DA}})/\widehat{S}(\hat{\hat{h}})$, and $\widehat{S}(h_{\mathrm{DG}})/\widehat{S}(\hat{\hat{h}})$ for all nine corpora. From the table, we see that with only one exception, these ratios are less than 1—typically well below 1, and in some cases strikingly close to 0. The only exception is for corpus C-1, for which the ratio is very slightly above 1. Thus, by this criterion, the LDA model based on the empirical choice of h greatly outperforms LDA models based on the other default choices of h, over a spectrum of corpora, ranging from some for which the documents are unrelated to some for which the documents are highly related.

Prior to carrying out our experiments on these nine corpora, we had conjectured that the magnitude of the gains in using the empirical choice of h would be greater for more complex corpora. In some sense this is true: on the whole, the documents are closer to each other for the Wikipedia corpora than they are for the 20Newsgroup corpora, and the gains in using the empirical choice of h are much greater for the Wikipedia corpora

than for the 20Newsgroup corpora. However, the 20Newsgroup corpora are arranged in order of increasing complexity (for C-1, the documents are very different and for C-5, the documents are similar) and as we go down the three columns on the right in Table 5-5, we do not see any clear pattern of decrease or increase in the entries for the first five rows of the table. Thus, there are other factors, beyond complexity of the corpora, that determine the magnitude of the gains in using the empirical Bayes choice of h, but from the experiments reported here and numerous others, we have not been able to identify a clear relationship between characteristics of the corpora and the gains obtained by using the empirical Bayes choice of h.

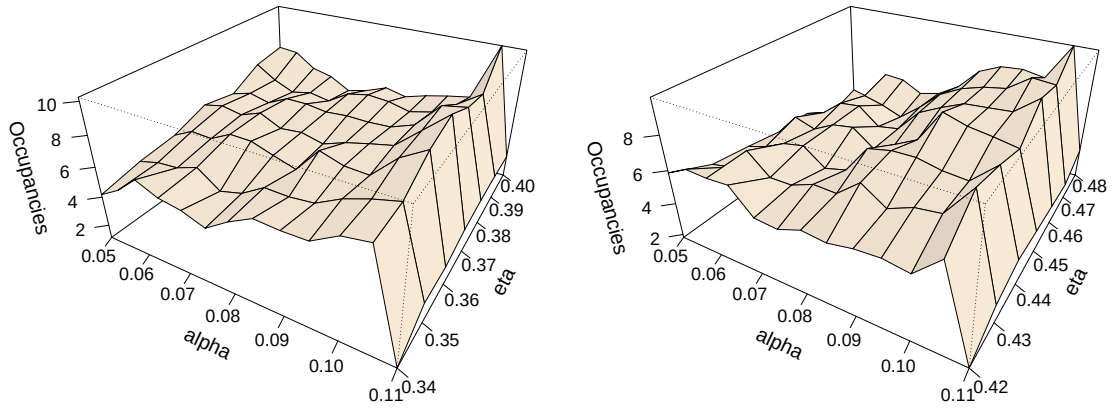
Table 5-5. Ratios of the estimates of posterior predictive scores of the LDA models indexed by default hyperparameters h_{DR}, h_{DA}, and h_{DG} to the estimate of the posterior predictive score of the empirical Bayes model, for all nine corpora.

Corpus	$\widehat{S}(h_{\mathrm{DR}})/\widehat{S}(\hat{\hat{h}})$	$\widehat{S}(h_{\mathrm{DA}})/\widehat{S}(\hat{\hat{h}})$	$\widehat{S}(h_{\mathrm{DG}})/\widehat{S}(\hat{\hat{h}})$
C-1	3.54×10^{-01}	$1.11 \times 10^{+00}$	8.24×10^{-04}
C-2	5.23×10^{-01}	2.52×10^{-02}	7.21×10^{-05}
C-3	2.98×10^{-01}	1.41×10^{-01}	1.33×10^{-02}
C-4	3.48×10^{-01}	1.22×10^{-01}	6.66×10^{-02}
C-5	4.58×10^{-01}	1.61×10^{-01}	9.36×10^{-02}
C-6	7.31×10^{-03}	5.71×10^{-06}	6.57×10^{-08}
C-7	5.34×10^{-03}	1.51×10^{-10}	1.89×10^{-14}
C-8	9.90×10^{-04}	1.77×10^{-09}	3.29×10^{-12}
C-9	2.17×10^{-02}	7.04×10^{-03}	5.56×10^{-09}

We now give details regarding the way the computations were carried out. To compute $\widehat{M}_\zeta(h)$, we implemented the serial tempering scheme described in Chapter 3 as follows. We took the hyperparameter values $h_1, \ldots, h_J$ to be a subgrid of the region of interest, with $J = 7 \times 13 = 91$. We used three iterations of the scheme given by Equation 3–14 to obtain ζ^{final}, with a Markov chain length of 50,000 per iteration (after a short burn-in period). The final run, using ζ^{final}, also used a Markov chain length of 50,000. For each corpus, we determined the h-region of interest by running a small pilot experiment to identify the set of h's having relatively high marginal likelihoods. We note that $\arg\max_h \widehat{M}_\zeta(h)$ can be obtained visually (or through a grid search) from the

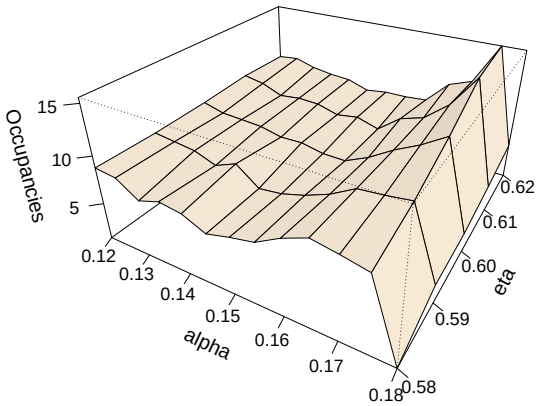
plots in Figures 5-2 and 5-4, but in practice these plots don't need to be generated, and $\arg\max_h \widehat{M}_\zeta(h)$ can be found very quickly through standard optimization algorithms (which are very easy to implement here, since the dimension of h is only 2). These algorithms take very little time because they require calculation of $\widehat{M}_\zeta(\cdot)$ for only a few values of h. To estimate the standard error of $\widehat{M}_\zeta(h)$, we used the method of batch means, which is implemented by the R package `mcmcse` in Flegal and Hughes (2012).

Recall that for the serial tempering chain to work well, it is necessary that the proportions of time spent in the different components of the mixture be approximately equal, and the vector of these proportions is the main diagnostic for assessing convergence of the chain (Geyer, 2011). Figures 5-6 and 5-7 give the distributions of the occupancy times for each of the nine corpora. The figures show that these distributions are acceptably close to the uniform in all cases.

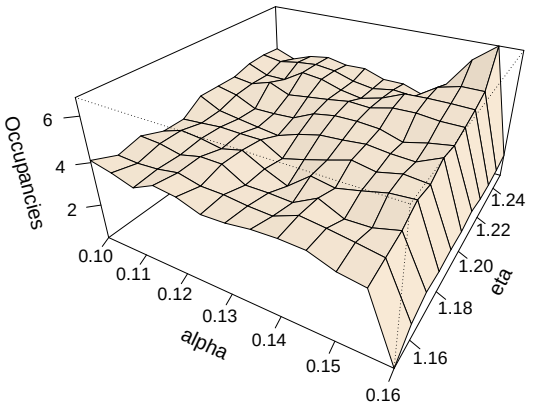


A C-1

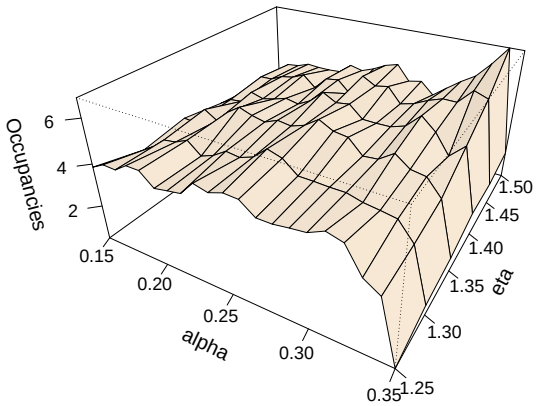
B C-2



C C-3

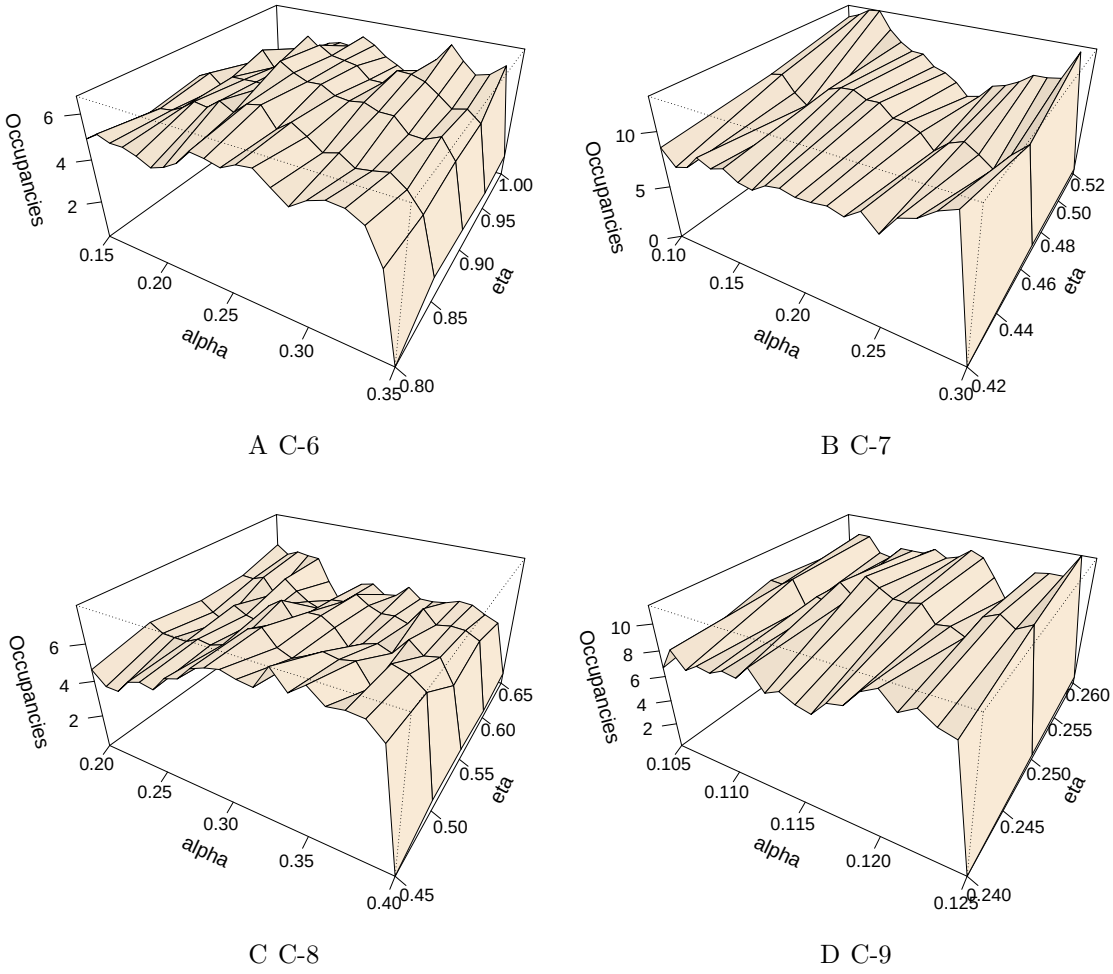


D C-4



E C-5

Figure 5-6. Plots of the number of iterations (in units of 100) that the final serial tempering chain spent at each of the hyperparameter values $h_1, \ldots, h_J$ in the subgrid, for corpora C-1–C-5.



A C-6 B C-7

C C-8 D C-9

Figure 5-7. Plots of the number of iterations (in units of 100) that the final serial tempering chain spent at each of the hyperparameter values $h_1, \ldots, h_J$ in the subgrid, for corpora C-6–C-9.

CHAPTER 6
ELECTRONIC DISCOVERY: INTRODUCTION

Discovery, is a pre-trial procedure in a lawsuit or legal investigation in which each party can obtain evidence from other parties according to the laws of civil procedure in the United States and other countries. This is typically performed via formal request for answers to interrogatories, request for production of documents (RPD), or request for admissions and depositions. By law the responding parties should produce the requested evidence unless such a request is successfully challenged in the court. A requesting party may obtain any information that refers to any tiny matter in the lawsuit, as long as the information is not "privileged" or otherwise protected by any law.

The primary subject of this chapter is document discovery. Computerization of offices and proliferation of smart devices has caused exponential growth in electronically stored information (ESI), i.e., documents either in native format—e.g., emails, attachments, social media messages, etc.—or after conversion into PDF or TIFF form (Casey, 2009). Electronic legal discovery (e-discovery) is the process of collecting, reviewing, and producing ESI to determine its relevance to a request for production. ESI is fundamentally different from paper information because of its form, persistence, and additional information such as document metadata (not available for paper documents). It can play a critical role in identifying evidence. On the other hand, the explosion of ESI to be dealt with in any typical case makes manual review cumbersome and expensive. For example, a study conducted at kCura on the number of documents handled in e-discovery cases (i.e., the median of case sizes), using the 100 largest cases, reported a growth of 2.2 million documents in 2010 to 7.5 million documents in 2011 (kCura, 2013). Some studies show that even with expert reviewers the results of manual review are inconsistent (Lewis, 2011). Both the cost of e-discovery and the error rate of document review pose significant challenges to the litigation process and as a result, are removing the public dispute resolution process from reach of an average citizen or a medium-sized company.

Thus, legal professionals have sought to employ intelligent technology assisted retrieval and review methods to reduce manual labor and increase accuracy.

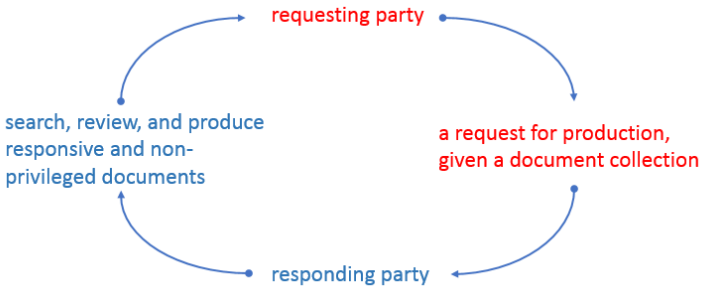


Figure 6-1. Technology Assisted Review Cycle

In a typical e-discovery procedure, ESI that are identified potentially relevant by attorneys on both sides of a lawsuit are placed on a legal hold. They are then searched and reviewed for relevance via a review platform after extracting and analyzing evidence via digital forensic procedures. The process is depicted in Figure 6-1. A popular information retrieval approach for e-discovery is keyword search or Boolean search as described as follows. First, the ESI of interest are processed to extract text within documents and data that describes documents, i.e., metadata. Second, each document along with its extracted data fields, e.g., *from*, *to*, *cc*, *bcc*, and *date*, for emails, are indexed using an indexing engine. One popular choice for this activity is the Apache Lucene indexing engine (Lucene, 2013). The next task is to identify the best keywords to find relevant documents as quickly as possible. Attorneys often derive search keywords from the prior knowledge about the case and the production request. Documents retrieved from this search will contain at least one of the search keywords. Boolean connectives such as AND, OR, NOT can be employed further tailor search results. Indexing schemes such as Lucene also permit Boolean search on different document fields (*faceted search* and search using phrases). Typically, such searching is an iterative procedure in which an attorney refines and validates search terms repeatedly until finding all of the relevant documents. However, finding all of the relevant documents can be burdensome and

expensive. Thus, the parties of a case using e-discovery must find a balance between the projected effort and potential benefits of proposed discovery, considering the facts and value of the case—i.e., the *proportionality* constraints of the case (Losey, 2013). In that sense, it is different from a search performed via search engines such as Google, Bing, Yahoo, which are optimized to produce the best results at the beginning of the list of the returned documents, for any given set of keywords.

Although keyword-based search remains the most popular retrieval scheme for e-discovery, it has many shortcomings. Some relevant documents may not contain the exact keywords specified by a user. Recall the example given in Chapter 1: the search keyword *computer*, may miss the documents that contain the words such as *PC*, *laptop*, *desktop*, and even *computers*, and do not have the word *computer*. *Stemming* and *lemmatization* may help us to solve issues due to different forms of a word, e.g., walk, walks, and walking. Stemming operates on a single word and applies a number of rules, disregarding the knowledge of a word's context, to obtain the *stem* of a word. For example, the stem for the words *fishing*, *fished*, *fish*, and *fisher* is *fish*. Lemmatization uses the meaning of a word (based on a dictionary such WordNet proposed by Miller et al. 1990), the context of a word, or the part-of-speech of a word in a sentence to find the *lemma* of a word. For example, the lemma for the token *better* is *good*. Popular e-discovery tools on the market, e.g., Catalyst[1] , enable stemming for keywords[2] . Even further, they support fuzzy search for keywords that allows a search keyword to match other terms that don't match exactly, but might be different by a letter or two. This helps to address typos. This way, the keyword *Mississippi* could still match word instances *Mississipi* or even *Misissippi*. *Synonymy* or *polysemy* of words that appear in a corpus may also cause poor keyword search performance. In addition, in a keyword-based approach, it's nearly impossible to

[1] http://catalystsecure.com

[2] http://www.edrm.net/resources/guides/edrm-search-guide

perform a *concept* search. The idea of concept search is to find matches for not just exact keywords but concepts that are similar to the keywords entered. For example, the search algorithm should return a document that contains *man's best friend* when one enters the keyword *dog*.

As we discussed before, one way to deal with the keyword search problems is via topic modeling methods such as Latent Semantic Indexing (LSI) (Dumais et al., 1995) and Latent Dirichlet Allocation (LDA, see Chapters 1 and 2). Topic modeling enables us to group co-occuring terms in a corpus and identify the underlying semantic or topic structure of a corpus. We therefore perform an empirical study of (a) comparing the performance of the LDA model to several other document modeling schemes that have been employed to model e-discovery corpora and search keywords (in terms of the underlying topic structure of a corpus) and (b) use documents in the topic representation space to better solve the document discovery sub problem in e-discovery (Chapter 7).

Computer Assisted Review: Background. In a typical Computer Assisted Review (CAR)—a.k.a., Technology Assisted Review (TAR) or predictive coding—for e-discovery one trains a computer to categorize documents based on relevancy to a legal case using a set of training (seed) documents labeled by expert reviewers. CAR has three main components—a domain expert, a categorization engine, and a method for validating results (kCura, 2013). A domain expert is a well trained human reviewer, e.g., a contract attorney, who can identify and label *relevant* and *irrelevant* documents from the document collection available for a legal case. A categorization engine propagates the knowledge of a domain expert to the whole document collection via indexing, relevance-ranking, and document classification. Finally, a validation method such as statistical random sampling (Israel, 1992) is used to validate whether the system's results are the results desired by the review team. For more discussion about the CAR process and e-discovery, one can consult the CAR Reference Model (EDRM, 2009) and Relativity (kCura, 2013).

We follow the EDRM CAR model as depicted in Figure 6-2 as the baseline to build our e-discovery retrieval model.

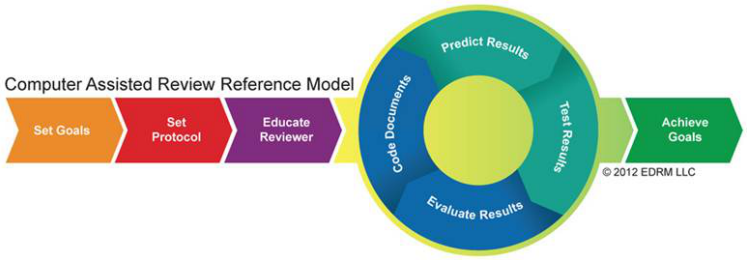


Figure 6-2. Computer Assisted Review Model (EDRM, 2009)

A critical task associated with the categorization of documents is ranking their relevance to a given user query. In a relevance ranking framework, users typically pose topic-specific keywords or phrases. For example, when searching for *computers*, search keywords such as *computer*, *calculator*, *machine*, *integrated circuit*, or *PC* might be formulated. The e-discovery software searches for documents containing the keywords (or variants thereof if the software has more advanced fuzzy logic, stemming and other capabilities), ranks them using a *similarity score*, and displays the results to users. The similarity score identifies how closely a document is related to the query. One example of such a score is cosine similarity. Most of the time, the keyword-based ranking methods are flawed as they are limited by the parameters and search terms employed by the user and the issue described earlier. Typically, when we search for documents we look for their concepts or topics rather than their keywords. This line of thinking leads us to build a hybrid document retrieval algorithm that uses the *topic structure* underneath a corpus along with existing keyword-based search strategies, e.g., Lucene (2013), Whoosh, etc.

Batch-based document classification, which deals with large static document collections, is usually performed using a supervised learning algorithm such as a support vector machine (SVM), neural network, or naïve Bayes classifier. One historical example of this type of system is the DolphinSearch tool (Berry et al., 2012), which supports electronic document discovery solutions. A supervised learning algorithm splits the

data into a training set and test set, and learns a classification function which maps the input documents to the corresponding labels using the training set. Then one analyzes the quality of the classification function by testing it on the test set, and uses the classification model to classify newly encountered unlabeled documents. These methods typically require sufficiently large sets of manually labeled documents for training. Another challenging problem in building a document classifier is the choice of features for documents in the corpus. To overcome these problems, we propose a system (Chapter 7) that uses an iterative classification scheme to discover relevant documents for a production request, where the documents in a corpus are represented in terms of identified topics of the corpus. We also propose several methods to select seed documents, which are typically presented to human experts for review. The system can then build a supervised classification model based on the expert labeled seed documents, for automated relevant document discovery.

CHAPTER 7
APPLYING TOPIC MODELS TO ELECTRONIC DISCOVERY

This chapter is organized as follows. Section 7.1 describes the proposed e-discovery system design and methods. In Section 7.2, we evaluate the proposed methods for our e-discovery model using a set of labeled e-discovery datasets. Section 7.3 summarizes this chapter and talks about future research directions.

7.1 System Design and Methods

This section describes the proposed SMART e-discovery retrieval (SMARTeR) system and implementation. We first give an overview of the SMARTeR work-flow as depicted in Figure 7-1. As noted before, once the ESI is identified by the parties on both sides of a legal case, the likely relevant documents are collected, parsed, and indexed by the SMARTeR data engine. This is an offline component of SMARTeR as depicted by steps (a), (b), and (c) in the system work-flow. The user enters a search query that is one more keywords or combination of multiple keywords on various metadata fields (i.e. *faceted search*) based on a legal request i.e. *Request for Production of Documents* (RPD). The system identifies a set of seed documents and displays it to the *domain experts*. Every seed document is then reviewed based on the document's relevancy to the RPD specifications and tagged to any relevance class such as *relevant* and *irrelevant*. The labeled seeds are used for training a document classifier, which is then used for categorizing the rest of the documents in the corpus. Our retrieval system has two parts—(a) classifying unlabeled documents as relevant and irrelevant given a case, and (b) computing document relevancy ranking scores for each of those classes. We will describe these two parts later in detail.

Once the system (the document classifier) has calculated relevance *ranking scores* and *class labels* for the rest of the documents in a corpus, they are displayed to the user for verification. The amount of data available for each class can be enormous making manual verification of the classification and ranking results intractable. A typical quality control

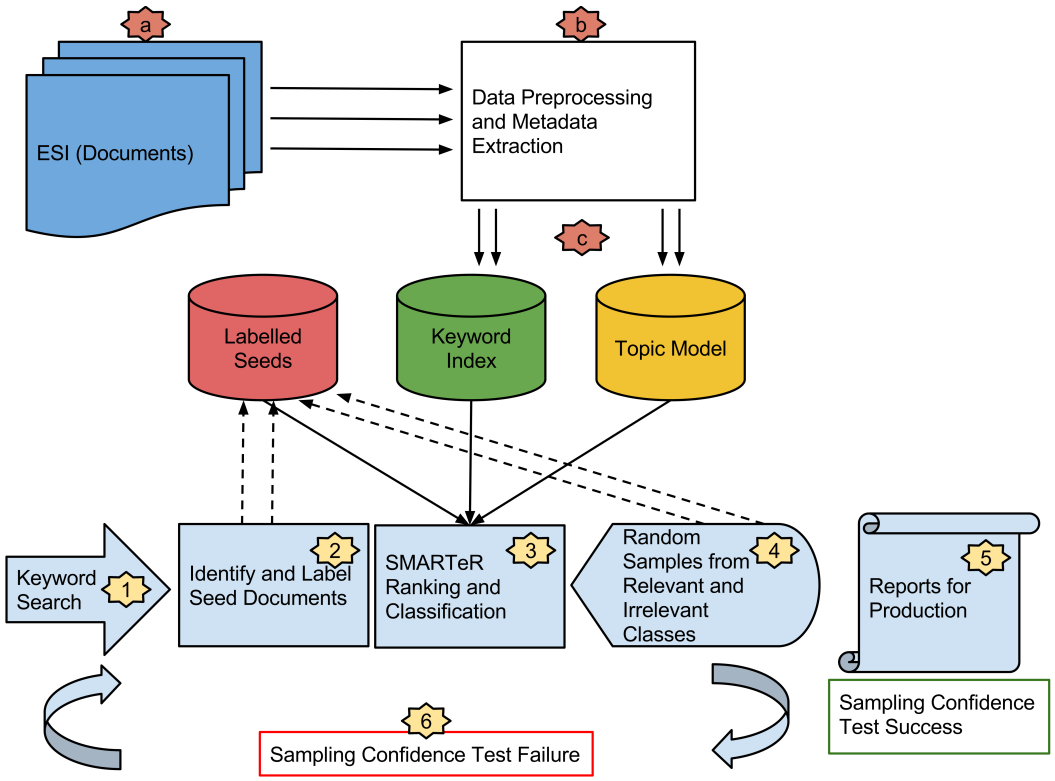


Figure 7-1. SMART e-discovery Retrieval work-flow: Starred numbers represent each step in the work-flow.

method used in the e-discovery community is to generate random samples from the set of relevant and irrelevant documents, and evaluate the quality of both of these sets by manual review of the samples. In this scheme, one typically reviews a random sample of documents from the whole population (i.e. the corpus), classifies them as relevant or non-relevant, and projects the percentage of relevant documents found in the sample onto the whole population. The sample size is influenced by several factors such as the corpus size, sampling error (i.e. the confidence interval), the confidence level, and the degree of variability (i.e. the prevalence of relevant documents in the corpus) (Israel, 1992). For more details about random sampling techniques used in the legal community, the readers may review the article by Losey (2012). For a more in-depth technical description of random sampling and determining sample size, see Cochran (1977); Israel (1992).

If the sampling test is passed the user can proceed to generate reports, otherwise, the user can go back and edit keyword-queries and continue the classification and ranking process in an iterative fashion. Stars 1–6 in Figure 7-1 depict these online iterative retrieval steps of SMARTeR. We now describe the main components of the SMARTeR system.

Data Pre-processing and Metadata Extraction

Electronically stored information for a given case can be represented in any format such as PDF, plain-text, HTML, and emails. The next step after data collection is to extract metadata by parsing ESI. For example, we can extract metadata such as *from*, *to*, *subject*, *date*, and *email body* from emails. Metadata can give additional information for better indexing and efficient meta-field or faceted search (e.g., Whoosh supports fields for a each document in the index). For any bag-of-words models such as TFIDF or LDA, documents are required to be in plain text format, and are converted into word tokens with a tokenizer. We use the python Natural Language Processing Toolkit (NLTK) (Bird et al., 2009) for tokenizing plain text. NLTK supports a number of tokenizers and also regular expressions. The next step is to standardize word tokens by removing noise terms and stop-words, e.g., *a*, *an*, *the*, *I*, *you*, *has*, etc. One can also apply any available *stemming* and *lemmatization* algorithms to normalize tokens. Typically, words that appear only once in a corpus—*hapax legomena*—are also discarded before applying any document modeling. Finally, each document in the corpus is converted into a bag-of-words format (e.g., the LDA-C format[1] , the Matrix Market format[2]) after building a vocabulary for the corpus.

[1] http://www.cs.princeton.edu/~blei/lda-c/

[2] http://math.nist.gov/MatrixMarket/formats.html

Keyword-based and Topic-based Document Indexing

For keyword-based indexing and search, we use algorithms such as *Apache Lucene* (Lucene, 2013), an industry standard in keyword-based indexing and *Whoosh* (Chaput, 2014), a full-text indexing and searching library implemented in pure Python. They enable us to index documents using document metadata, e.g., *file modified date*, and data fields, e.g., *email-subject*, *email-body*, and search for documents using search keywords. Both of these libraries provide methods to rank documents given Boolean search keywords based on similarity scores such as cosine similarity. We use the retrieval results from these libraries as the baseline for analyzing our proposed classification and ranking methods.

A challenging problem in document classification and ranking is the choice of features for documents. Considering relative frequencies of individual words in documents as features as in TF or TF-IDF models may yield a rich but very large feature space (Joachims, 1999) and may cause computational difficulties. A more computationally effective approach would be to analyze documents represented in a reduced topic space extracted by topic models such as LSI and LDA. For example, words such as *football*, *quarterback*, *dead ball*, *free kick*, *NFL*, *touchdown*, etc., are representative of the single topic *football*. Topic models are used to identify these topic structures automatically from document collections.

We now give some details regarding the implementation of topic modeling for a corpus. We use the scalable implementation of LSI and LDA algorithms by Řehůřek and Sojka (2010) in our experiments. The LSI implementation is based on Halko et al. (2011) that performs a scalable singular value decomposition (SVD) of the TF-IDF matrix for a corpus, and projects documents represented in the TF-IDF matrix into the LSI (semantic) space. The LDA implementation is based on the online variational Bayes (VB) algorithm (Hoffman et al., 2010) that reduces any document in the corpus to a fixed set of real valued features—the variational posterior Dirichlet parameters θ_d^* associated with each document d in the corpus. Henceforth, we denote θ_d^* as the estimate of θ_d, i.e., document

d's distribution on the topics (see the hierarchical model of LDA given in Section 2). For the LSA-based methods, we also use θ_d^* to denote the projected document d in the LSI space for notational simplicity. One can then consider each keyword search as a document in the corpus and identify its representation θ_{query}^* in the topic or semantic space using a pre-identified LDA or LSA model of the corpus, for the topic modeling-based document retrieval.

Seed Document Selection

In the CAR process, expert labeled seed documents are crucial for building the classification and ranking models alluded to earlier and described later in detail. One of the goals in e-discovery is to reduce manual labor for review, and also to increase the accuracy of relevant document retrieval. Typically, seed documents are chosen randomly or from the initial ranking results from a keyword-based search engine. Here, we propose four principled seed selection strategies:

- **k-means (a)**: This method emerges from the concept of "stratified sampling" from the whole population. We first employ a distance-based clustering algorithm such as k-means clustering (see, e.g., Bishop et al. (2006) for more details) on documents that are represented as feature vectors (θ_d^*, $d = 1, \ldots, D$), to identify their membership clusters—*strata*. We then take a sample from each learned stratum via random sampling and aggregate them to form a set of seed documents. The size of the sample for each stratum is chosen in proportion to the size of the stratum.

- **k-means (b)**: As in **k-means (a)**, we first cluster documents using k-means. We then select documents which are far away from the cluster centers and aggregate them to form a seed set of documents. The number of documents being selected from each stratum is chosen in proportion to the size of the stratum.

- **whoosh (a)**: We first form *search keywords* based on the request for production of document for the case of interest. We then perform a keyword-based search for relevant documents using the search keywords and the Whoosh full-text index created for the document collection as discussed before. In principle, we can use any full-text indexing method for this purpose . We then consider the documents retrieved from the Whoosh index given the search query as the class of *relevant* documents and the rest of the documents in the corpus as the class of *irrelevant* documents. Finally, to form a seed set, we sample documents from both of these sets proportionally. The ratio of the number of *relevant* documents and *irrelevant* documents in the seed set follow the same ratio of the number of documents in

the *relevant* class and the number of documents in the *irrelevant* class. This seed selection method also can be considered as a variation of **k-means (a)**: both **k-means (a)** and **whoosh (a)** are based on the principles of "stratified sampling"; **k-means (a)** uses the k-means algorithm to stratify of documents, but **whoosh (a)** uses the Whoosh search to stratify documents.

- **whoosh (b)**: As in **whoosh (a)**, we first define the class of *relevant* documents and the class of *irrelevant* documents. We then *evenly* sample documents from each of these classes to create the seed set.

Along with these four seed selection methods, we will also evaluate classification models that are built using randomly selected seed documents from each corpus. The results of this comparative study are given in Section 7.2.

Document Ranking

Recognizing how relevant a document is to a legal case (in terms of a relevancy score) is crucial in any e-discovery process, as it may help lawyers to decide the review budget and the cut off on the number of documents to be reviewed. Here, we consider a number of methods to identify the optimal ranking for documents given a keyword-search:

- **whoosh**: We present the search keywords to the Whoosh search algorithm (Chaput, 2014), and use its relevance response for each document as the document's relevance index. This method is essentially the type of keyword search done in any of the keyword-based e-discovery software.

- **keyword-lsa**: We first compute the LSI model of a corpus and for $d = 1, 2, \ldots, D$, we identify bag-of-words formatted document d's projection θ_d^* in the LSI semantic space. We then consider each keyword query as a document in the corpus and identify its representation θ_{query}^* by projecting it into the same LSI space. Finally, for document d, we compute the document relevancy score as the cosine similarity between the semantic vectors θ_{query}^* and θ_d^*.

- **keyword-lda**: We first compute the LDA model of a corpus and for $d = 1, 2, \ldots, D$, we identify bag-of-words formatted document d's θ_d^* in the LDA topic space. We then consider each keyword query as a document in the corpus and identify the estimate of the query topic distribution θ_{query}^* using the learned LDA model. Finally, we compute cosine similarity between θ_{query}^* and each document's θ_d^* as document d's relevancy score.

- **topic-lda**: As in **keyword-lda**, for $d = 1, 2, \ldots, D$, we first estimate θ_d^* for document d in the corpus, and $\theta_{\text{query}}^* = (\theta_1^*, \theta_2^*, \ldots, \theta_K^*)$ for a keyword query. Second, we then identify k most relevant topics given the search keywords as follows. From

the distribution on topics θ^*_{query} for the search keywords, we select the most probable topics by sorting the corresponding probabilities $\theta^*_1, \theta^*_2, \ldots, \theta^*_K$. Lastly, we compute the combined relevancy score of k most relevant topics for each document d based on θ^*_d as document d's relevance index as follows. Let $\mathcal{K}$ represent the indices of topics in the corpus and $\mathcal{T} \subset \mathcal{K}$ represents the indices of k most relevant topics given the query topic distribution θ^*_{query}. For each document $d = 1, 2, \ldots, D$ in the corpus, we can calculate the score (George et al., 2012):

$$\text{sim}(d) = \sum_{j \in \mathcal{T}} \ln \theta^*_{dj} + \sum_{j \notin \mathcal{T}} \ln(1 - \theta^*_{dj}) \tag{7–1}$$

Note that a high value of $\text{sim}(d)$ indicates the topics indexed in $\mathcal{T}$ are prominent in document d.

Document Classification

To learn the document classifiers mentioned in the e-discovery workflow, we employ the Support Vector Machines (SVM) (Vapnik, 1995), a popular algorithm used for text classification (Joachims, 1998). SVM classifiers require a training set that consists of data points (i.e. feature vectors) and their desired output (i.e. class labels) for training. We build the training set combining the feature vector x_d (described below) and expert annotated label y_d (i.e. the desired class) for each seed document. The learned SVM models are used for classifying the rest of unlabeled documents in the collection. We consider a number of possible approaches to build the feature vector x_d, for document $d = 1, 2, \ldots, D$ in the corpus:

- **lda**: For $d = 1, 2, \ldots, D$, we take the vector $x_d \in (0, 1)^K$ as the K-dimensional distribution on topics θ^*_d for document d, from the LDA model for a corpus.
- **lda+whoosh**: For $d = 1, 2, \ldots, D$, we build the vector $x_d \in (0, 1)^{K+1}$ as the aggregated vector of K-dimensional distribution on topics θ^*_d for document d from the LDA model of a corpus, and the ranking score for document d computed by the Whoosh search engine given a keyword search. We normalize the document ranking scores to the range of $(0, 1)$ for the SVM algorithm.
- **lsa**: For document $d = 1, 2, \ldots, D$, we consider the vector x_d as the K-dimensional document representation θ^*_d in the LSI semantic space.
- **lsa+whoosh**: For document $d = 1, 2, \ldots, D$, we build the K-dimensional vector x_d as an aggregated vector of the projected document into the LSI semantic space and the document ranking score computed by a keyword search engine given a

keyword-query. We also normalize the document ranking scores to the range of $(0, 1)$ for the SVM algorithm.

7.2 Experiments and Analysis of Results

Here we describe a set of experiments based on an e-discovery dataset that was employed in the TREC[3] 2010 Legal Learning Track (Cormack et al., 2010), and also the 20Newsgroups dataset[4] , a popular dataset used in the machine learning literature for experiments in applications of text classification and clustering algorithms. The TREC dataset contains emails and their attachments from the well-known Enron dataset. TREC has annotated a subset of this dataset against eight sample topics as *relevant*, *irrelevant*, and *not assessed*. We use these annotated topics after removing non-assessed documents. Table 7-1 describes the four created corpora from the annotated topics. The column RPD gives the *Request for Production of Documents* to produce relevant and irrelevant items from the Enron collection of 685,592 e-mail messages and attachments for each corpus. In a typical keyword search for e-discovery, one builds a Boolean query using the search keywords derived from an RPD. The column Search Keywords gives the corresponding search keywords used in our analysis for each corpus.

The 20Newsgroups dataset contains approximately 20,000 articles that are partitioned relatively even by across 20 different newsgroups or categories. We created two sets of corpora from this dataset as described in Table 7-2 and Table 7-3. Corpora C-Medicine and C-Baseball were built for evaluating various seed selection methods described in Section 7.1. In corpus C-Medicine, the relevant class consisted of all the documents (990) under the newsgroup sci.med and the irrelevant class consisted of the rest of the documents (17,856) in the 20Newsgroups document collection. In corpus C-Baseball, the relevant class consisted of all the documents (994) under the newsgroup rec.sport.baseball

[3] http://trec.nist.gov

[4] http://qwone.com/~jason/20Newsgroups

and the irrelevant class consisted of the rest of the documents (17,852) in the 20Newsgroups document collection. To suit the real-world situations we have observed for e-discovery, we made these two corpora unbalanced in terms of class population, with small proportions of positive classes (5% of each corpus). We built corpora C-Mideast, C-IBM-PC, C-Motorcycles, and C-Baseball-2 to evaluate the performance of various document classifiers. For each corpus, the relevant class included documents under a single relevant group and the irrelevant class included documents under a set of irrelevant groups from the 20Newsgroups dataset. In Table 7-3, the column Relevant Group gives the relevant newsgroup and the column Irrelevant Groups gives the set of irrelevant newsgroups used for each corpus. The column Rel./Irrel. gives the number of documents in the relevant class vs the number of documents in the irrelevant class, for each created corpora.

Comparing Document Ranking Methods

As discussed in Section 7.1, we consider a number of different methods to identify the optimal ranking for documents given an RPD, based on their ability to classify documents—using document ranking scores—as *relevant* or *irrelevant*. Each ranking method is evaluated by employing the Receiver Operating Characteristic (ROC) curve analysis on the ranking scores produced for all documents in the corpus given an RPD. Appendix B.2 gives a brief introduction to the ROC curve analysis. Our experimental results using topic-learning methods provide the evidence that topic-learning may be able to improve automatic detection of relevant documents and can be employed to rank documents by their relevance to a topic.

We now give some details regarding the implementation of various ranking methods. We used the four corpora described in Table 7-1 for our analysis. We set both the number of topics K for the LDA algorithm and the number of components for the LSA algorithm as 50 for each corpus. In our analysis, for each corpus, we considered two versions of the text data: (a) one using raw word tokens and (b) the other using normalized word tokens. To perform Whoosh search, we built whoosh queries in the format `all_fields:( ... )`

Table 7-1. Corpora created from the TREC-2010 Legal Track topic datasets.

Corpus	Request for production of documents (RPD)⋆	Search keywords†	Rel./Irrel.‡
C-201	"All documents or communications that describe, discuss, refer to, report on, or relate to the Company's engagement in structured commodity transactions known as *prepay transactions*."	pre-pay, swap	168 / 520
C-202	"All documents or communications that describe, discuss, refer to, report on, or relate to the Company's engagement in transactions that the Company characterized as compliant with *FAS 140 (or itspredecessor FAS 125)*."	FAS, transaction, swap, trust, Transferor, Transferee	994 / 400
C-203	"All documents or communications that describe, discuss, referto, reporton, or relate to whether the Company had met, or could, would, or might meet its *financial forecasts models, projections, or plans* at any time after January 1, 1999."	forecast, earnings, profit, quarter, balance sheet	64 / 878
C-207	"All documents or communications that describe, discuss, refer to, report on, or relate to *fantasy football, gambling on football, and related activities*, including but not limited to, football teams, football players, football games, football statistics, and football performance."	football, Eric Bass	80 / 492

⋆The RPDs are taken from the TREC-2010 Legal Track description.
†The search keywords are adapted from Tomlinson (2010).
‡This column shows the number of relevant documents vs. the number of irrelevant documents for a corpus.

that will search the keywords ... in all fields of the Whoosh index for a corpus. For both versions of corpora (a) and (b), we converted the search keywords specified in Table 7-1 to lower case before ranking. For (b), we also normalized the search keywords for each corpus.

Figure 7-2A, Figure 7-2C, Figure 7-3A, and Figure 7-3C show the performance of various ranking methods based on raw word tokens of corpora C-201, C-202, C-203, and C-207. Figure 7-2B, Figure 7-2D, Figure 7-3B, and Figure 7-3D show the performance of

Table 7-2. Corpora created from the 20Newsgroups dataset to evaluate various seed selection methods.

Corpus	Relevant group	Search keywords	Rel./Irrel.†
C-Medicine	sci.med	medicine science hospital patient capsule diabetes hypertension cholesterol dyslipidemia pain fever rash ECG EKG x-ray MRI CT scan	990 / 17856
C-Baseball	rec.sport.baseball	baseball pitching batting pitcher batsman ground ball national league playoff fielding inning	994 / 17852

†This column shows the number of relevant documents vs. the number of irrelevant documents for a corpus. Irrelevant documents are taken from all 20-news groups except the relevant group.

Table 7-3. Corpora created from the 20Newsgroups dataset to evaluate various classifiers.

Corpus	Relevant group	Irrelevant groups	Rel./Irrel.†
C-Mideast	talk.politics.mideast	rec.sport.hockey, rec.autos, rec.sport.baseball, rec.motorcycles, comp.sys.ibm.pc.hardware	940 / 4,942
C-IBM-PC	comp.sys.ibm.pc.hardware	rec.sport.hockey, rec.autos, rec.sport.baseball, rec.motorcycles, talk.politics.mideast	982 / 4,900
C-Motorcycles	rec.motorcycles	rec.sport.hockey, rec.autos, rec.sport.baseball, talk.politics.mideast, comp.sys.ibm.pc.hardware	996 / 4,886
C-Baseball-2	rec.sport.baseball	rec.sport.hockey, rec.autos, rec.motorcycles, talk.politics.mideast, comp.sys.ibm.pc.hardware	994 / 4,888

†This column shows the number of relevant documents vs. the number of irrelevant documents for a corpus.

various ranking methods based on normalized word tokens of corpora C-201, C-202, C-203, and C-207. It is clear that topic modeling-based ranking methods outperforms **whoosh** uniformly in all cases except for corpus C-202. For the raw text version corpus C-202, **whoosh** outperforms all the three methods and the normalized version of corpus C-202 and **whoosh** performs marginally over **keyword-lda** and **keyword-lsa**. For instance,

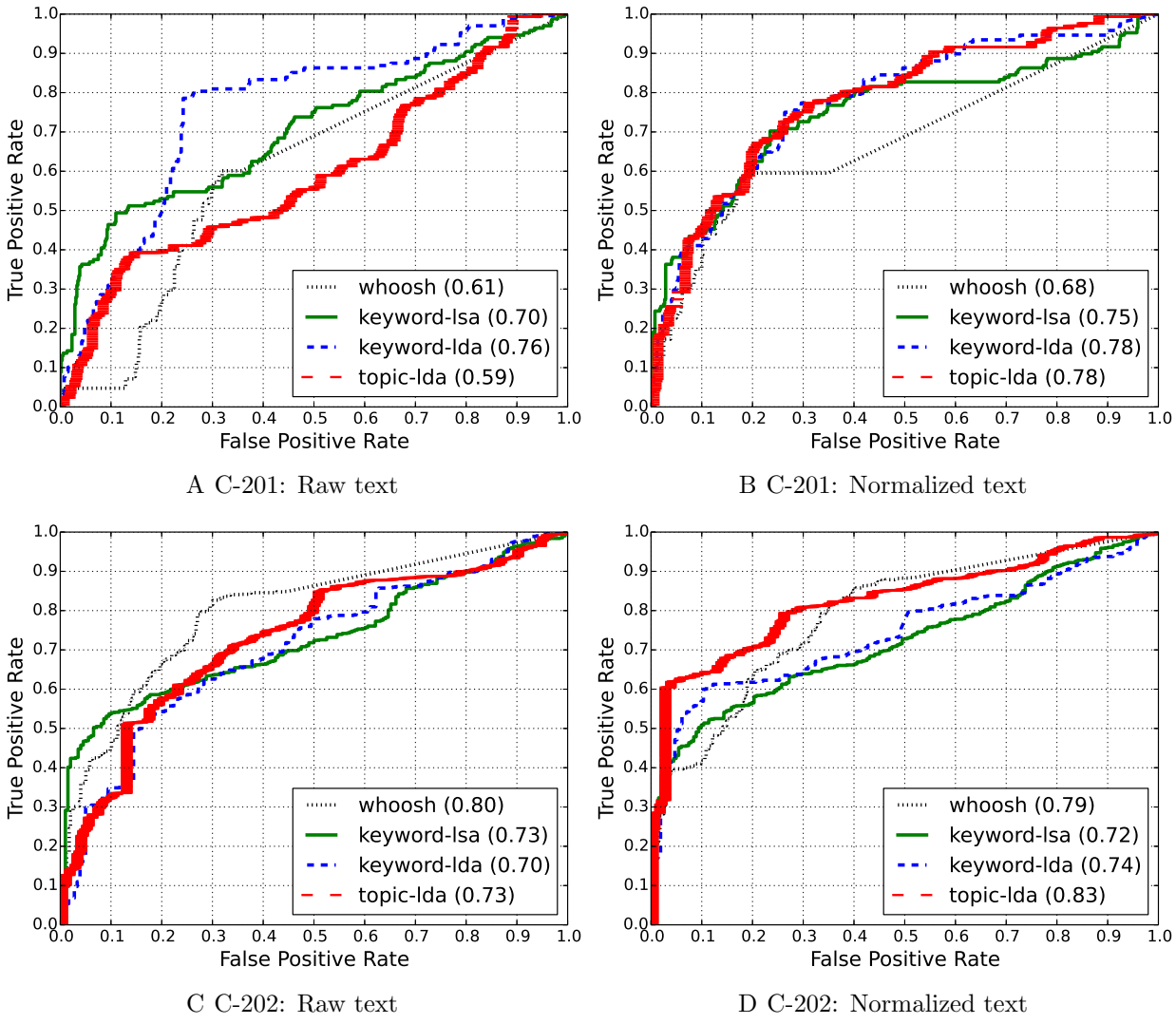
in Figure 7-3A, Whoosh search achieves about a 35% True Positive Rate (TPR) with a very small False Positive Rate (FPR) of about 4%, but learns very little after that. It does not exceed about 35% TPR before it labels every succeeding document with an identical confidence (as shown by the diagonal dotted line leading to the upper right corner). The **keyword-lda** and **topic-lda** perform reasonably well, achieving a TPR of around 88% with a 30% FPR. The approach **keyword-lda** is marginally better than **topic-lda** after that.

In our experience, the right keyword combination is critical for the Whoosh search algorithm. In the experiments not shown, we found that small changes in the whoosh Boolean query combinations for the same set of keywords may produce large performance differences for retrieval. On the other hand, the methods **keyword-lda**, **keyword-lsa**, and **topic-lda**, considered bag-of-words formatted search terms, and performed consistently. On average, normalization of tokens helped topic modeling-based ranking methods, but it did not make much difference in **whoosh** ranking. Overall, **keyword-lsa** performs reasonably well, but the LDA-based methods have an edge in nearly all cases. To conclude, unless we have the right keyword combination for **whoosh**, **keyword-lda** is reasonable choice for ranking documents given an RPD.

In addition, in our experience (experiments are not shown) fusing keyword-based ranking scores with topic-modeling-based ranking scores gives better performances in certain cases.

Comparing Seed Document Selection Methods

Here we compare the performance of various seed selection methods proposed earlier in Section 7.1 using corpora C-Medicine and C-Baseball as follows. Seeds generated via various seed selection schemes along with their expert annotated labels are used to build document classifiers for each corpus. We then evaluate each seed selection scheme by the predictive performance of the corresponding classifier. We used the implementation of Support Vector Machines based on linear kernels by Pedregosa et al. (2011) as classifiers.



A C-201: Raw text

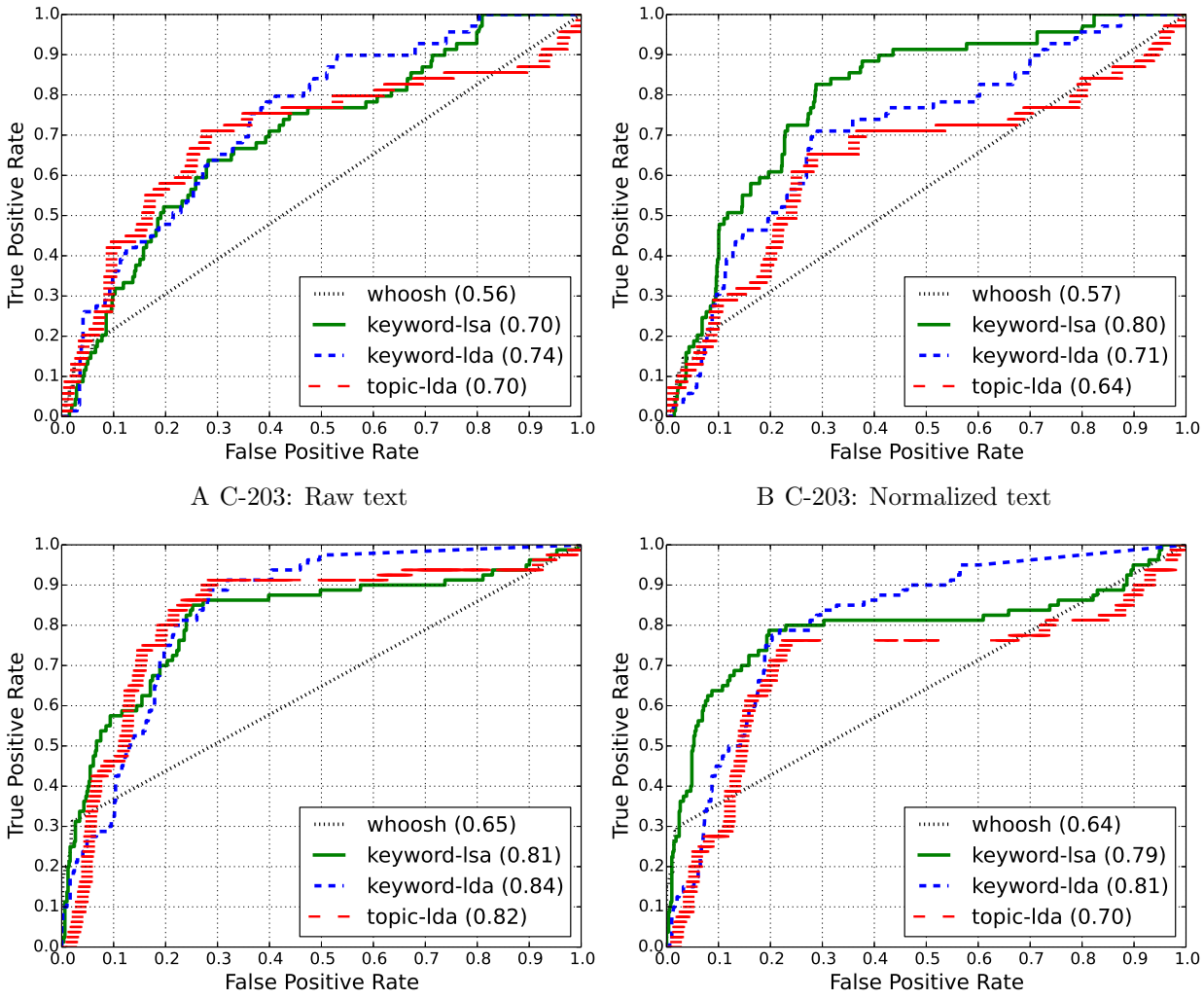
B C-201: Normalized text

C C-202: Raw text

D C-202: Normalized text

Figure 7-2. ROC curve analysis of various ranking models for corpora C-201 and C-202.

In a typical e-discovery setting, where labeled training data is scarce, selecting parameters for SVM via cross-validation may not be ideal, as it may cause over-fitting (Cormack and Grossman, 2015). So, we used the default parameter configurations given by Pedregosa et al. (2011) while training SVM models. The number of seeds n_{seeds} is an input to every seed selection method. To evaluate the performance of a learned SVM classifier, we compared the predicted labels of the rest of the documents in each corpus with the true document labels using AUC, Recall, and Precision scores.



A C-203: Raw text

B C-203: Normalized text

C C-207: Raw text

D C-207: Normalized text

Figure 7-3. ROC curve analysis of various ranking models for corpora C-203 and C-207.

Figure 7-4 gives the plots of AUC, Recall and Precision for the results of various SVM classifiers based on different seed selection methods, for corpora C-Medicine and C-Baseball. Here, to train and test the classifiers, we used semantic features (total 200 semantic features) generated by the Latent Semantic Analysis algorithm for each document. To study the impact of the number of seeds n_{seeds} in classification, we ran each seed selection method for $n_{\text{seeds}} = 100, 200, \ldots, 1{,}000, 1{,}500, \ldots, 5{,}000$. For ***k*-means (a)** and ***k*-means (b)**, we set $k = 4$ as it gave us superior results in our experiments (Choosing the right k is an interesting problem to deal with, but we leave this problem to

future work.). From Figure 7-4, after $n_{\text{seeds}} = 500$, the methods **whoosh (b)**, k**-means (a)**, and **random** perform reasonably well in terms of AUC scores. In the n_{seeds} range 100-500, **whoosh (b)** has an edge over k**-means (a)** and **random** . As we can see, **whoosh (a)** performs slightly lower than **whoosh (b)**, k**-means (a)**, and **random**, and k**-means (b)** performs poorly compared to other methods, especially with small numbers of seeds. In terms of Recall, k**-means (a)** outperforms all other methods marginally well and k**-means (b)** performs poorly for these two datasets. Lastly, in terms of Precision, Whoosh-based approaches perform reasonably well for corpus C-Baseball, but for C-Medicine k**-means (b)** performs exceptionally well. It's surprising to see that random selection of seed documents performs reasonably well in terms of AUC and Recall for these two datasets. Our guess is that the richness of relevant documents played a role in these two datasets. To conclude, to select seeds we can either use k**-means (a)** or **whoosh (b)**, but we think k**-means (a)** is superior because it does not require much supervision compared to **whoosh (b)** (i.e. **whoosh (b)** requires us to specify the right keywords given an RPD to perform Boolean search).

To perform further analysis, we repeated the same set of experiments using topic features (total 50 topics) generated by the LDA algorithm for each document in corpora C-Medicine and C-Baseball. Figure 7-5 gives the plots of AUC, Recall, and Precision for the results generated by the corresponding SVM classifiers based on various seed selection methods. As can be seen, topic features produce comparable results for all five seed selection methods, but on average, the methods k**-means (a)** and **whoosh (b)** have an edge. Lastly, LSA-based models outperform LDA-based models reasonably well in terms of classification performance for these two corpora.

In the experiments not shown here, we noticed that custom regular expression generated tokens turned out to be better features for the SVM-based classifiers, compared to raw word tokens (generated via a token-ization scheme based on white space or

non-alphanumeric characters). On the other hand, normalization based on stemming did not improve the performance of SVM-based classifiers in our experience.

Comparing Various Classifiers based on Topic Modeling and Keyword-based Features

Here, we compare the performance of a set of classifiers built based on different feature types, using the corpora described in Table 7-3. The details of our experiments are as follows. Document texts were token-ized with the help of a *regular expression*-based token-izer. We used document features derived from document modeling methods such TF-IDF, LSA, and LDA to build various classifiers. The number of topics set for the LDA model was 50 and the number of semantic features set for the LSA model was 200. For classification, we considered popular classification algorithms such as *Logistic Regression* (LR), *SVM (RBF)* (SVM-R), *SVM (Linear)* (SVM-L), and k-*Nearest Neighbor* (k-NN). We used the implementations of these classification algorithms (along with the default tuning parameters) provided in the `scikit-learn` package for our experiments.

Table 7-4 gives AUC, Precision, and Recall scores of the various classification results for corpora C-Mideast, C-IBM-PC, C-Motorcycles, and C-Baseball-2. Table 7-5 gives the run time performance for the same set of experiments. The classification models are evaluated using a stratified 5-fold cross-validation scheme on all four corpora. This cross-validation scheme is a variation of k-fold cross-validation, in which, the folds—configurations of the test and training sets created from the original dataset—are made by preserving the percentage of documents for each class in a dataset. We now compare various classification models in terms of AUC performance. Precision and Recall scores are included as a reference for readers. All classification methods performed reasonably well for all features types in terms of AUC, except for k-*Nearest Neighbor* classifiers, which performed poorly for all feature types. It is surprising to note that *Logistic Regression* and *SVM (Linear)* methods gave similar AUC scores for all feature types (and Precision and Recall scores are comparable). We believe this is due to the

similarity of the algorithms used in the `scikit-learn` package to find optimal solutions, and the choice of penalties. Similarly, *SVM (Linear)* is superior to *SVM (RBF)* uniformly in all cases except for corpus C-Baseball-2, for which *SVM (RBF)* is marginally better. In addition, the training and test times of the *SVM (RBF)*-based models is too high (See Table 7-5), which is a drawback. We believe selecting the *SVM (RBF)* kernel parameters and slack variable will further improve the *SVM (RBF)*-based models. Another interesting observation is that for classification, simpler document models such as LSA and TF-IDF outperforms LDA-based models for all the four corpora. Our guess is that selecting hyperparameters and the number of topics for the LDA model of a corpus may make a difference in the classification performance (this is part of our future work). One issue with the TF-IDF-based models were the computational challenges of handling huge vocabularies (e.g., we did limited experiments for corpus C-Mideast).

Table 7-4. Performance of various classification models using the features derived from the methods LDA, LSA, and TF-IDF for corpora C-Mideast, C-IBM-PC, C-Motorcycles, and C-Baseball-2.

Corpus	Classifier	AUC			Precision			Recall		
		lda	lsa	tfidf	lda	lsa	tfidf	lda	lsa	tfidf
C-Mideast	LR	0.95	0.99	-	0.63	0.92	-	0.84	0.83	-
	SVM-R	0.95	0.99	0.83	0.64	0.86	0.16	0.85	0.88	0.20
	SVM-L	0.95	0.99	0.99	0.64	0.83	0.98	0.85	0.90	0.83
	k-NN	0.17	0.38	0.48	0.84	1.00	0.00	0.51	0.38	0.00
C-IBM-PC	LR	0.96	0.99	0.99	0.81	0.95	0.97	0.87	0.90	0.89
	SVM-R	0.96	0.99	0.83	0.85	0.95	0.17	0.81	0.92	0.80
	SVM-L	0.96	0.99	0.99	0.80	0.93	0.97	0.87	0.93	0.89
	k-NN	0.28	0.24	0.49	0.90	1.00	0.00	0.73	0.29	0.00
C-Motorcycles	LR	0.84	0.96	0.97	0.39	0.71	0.88	0.78	0.81	0.78
	SVM-R	0.85	0.96	0.76	0.40	0.72	0.17	0.79	0.81	0.20
	SVM-L	0.84	0.96	0.97	0.37	0.70	0.91	0.80	0.83	0.76
	k-NN	0.19	0.22	0.47	0.62	0.98	0.00	0.16	0.13	0.00
C-Baseball-2	LR	0.91	0.97	0.98	0.55	0.78	0.88	0.77	0.83	0.81
	SVM-R	0.92	0.98	0.71	0.59	0.74	0.17	0.77	0.86	0.40
	SVM-L	0.91	0.98	0.98	0.55	0.72	0.91	0.78	0.88	0.82
	k-NN	0.24	0.23	0.51	0.84	0.97	0.00	0.46	0.16	0.00

Table 7-5. Running times of various classification models using the features derived from the methods LDA, LSA, and TF-IDF for different corpora.

Corpus	Classifier	Train-time			Test-time		
		lda	lsa	tfidf	lda	lsa	tfidf
C-Mideast	LR	0.05	0.81	-	0.00	0.01	-
	SVM-R	5.13	15.65	5822.48	0.81	2.32	1385.64
	SVM-L	0.25	2.42	9.71	0.00	0.02	1.09
	k-NN	0.26	0.42	62.16	3.31	18.87	1883.26
C-IBM-PC	LR	0.07	0.50	3.48	0.00	0.01	0.80
	SVM-R	5.88	6.52	4217.18	1.13	1.22	1016.66
	SVM-L	0.11	0.61	3.72	0.00	0.01	0.81
	k-NN	0.25	0.23	53.30	3.17	14.28	1604.86
C-Motorcycles	LR	0.08	0.52	3.59	0.00	0.01	0.81
	SVM-R	9.14	12.22	4190.50	1.43	2.20	1016.77
	SVM-L	0.11	0.73	3.81	0.00	0.01	0.81
	k-NN	0.25	0.23	53.09	3.13	14.39	1609.67
C-Baseball-2	LR	0.07	0.54	3.65	0.00	0.01	0.85
	SVM-R	6.58	15.05	4297.01	1.07	2.95	1039.10
	SVM-L	1.05	5.68	3.84	0.00	0.01	0.83
	k-NN	0.25	0.23	53.91	3.19	14.44	1645.57

We now compare the performance of various SVM classifiers based on document features derived from LDA and LSA and their combinations with Whoosh retrieval score 7.1. For inference, we built both LDA and LSA models based on the number of features or topics k from the sequence $5, 10, 15, 20, 30, \ldots, 80$. To compare the SVM classification performance with keyword-based classification, for a given corpus and a keyword query, we took documents retrieved by Whoosh as relevant documents and the rest of the documents in a corpus as irrelevant documents. The SVM model parameters parameters are selected via grid-search. Figure 7-6 and Figure 7-7 give the plots of *AUC*, *Precision*, and *Recall* scores for the results of the SVM classifiers, evaluated in cross validation, for corpora C-201, C-202, C-203, and C-207. The evaluation scores of the Whoosh retrieval for the respective search keywords (see Table 7-1) are also plotted in these figures. We now analyze the performance of various classifiers for all four corpora.

As can be seen, variants of LDA and LSA feature selection methods outperform Whoosh retrieval in all four corpora of interest in terms of *AUC*.

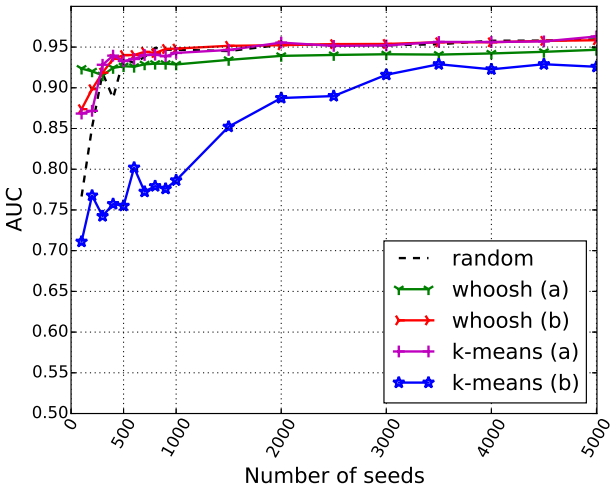
In terms of *Recall*, topic modeling-based classifiers outperforms Whoosh classification for corpora C-201 and C-202 (see Figure 7-6C and Figure 7-6D), but not for corpora C-203 and C-207 (see Figure 7-7C and Figure 7-7C). In terms of Recall, LSA-based classifiers are marginally or reasonably better than LDA-based classifiers for all four corpora. Variants of LSA-based classifiers have an edge over the variants of LDA-based classifiers in all cases. We believe this is due to the impact of the size of the documents (mostly emails) used for topic modeling, as it might adversely affect the learned topics and document topic feature. In addition, appending Whoosh ranking scores as a feature to topic modeling feature vectors (i.e. **lsa-whoosh** and **lda-whoosh**) for documents helps marginally in some cases.

7.3 Summary and Discussion

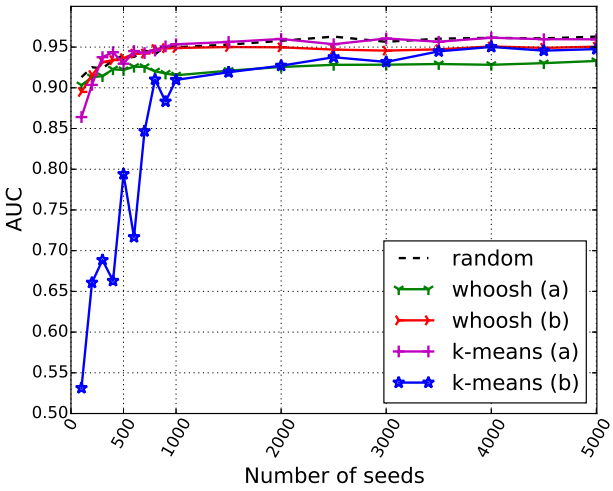
This chapter proposed a Computer Assisted Review (CAR) work-flow for e-discovery based on various document modeling methods and supervised classification. We employed the popular topic model Latent Dirichlet Allocation (LDA) along with other document modeling schemes such as TF-IDF and Latent Semantic Analysis (LSA) to model documents in an e-discovery process. We considered the document discovery problem to be a document classification problem and applied well-known classification algorithms such as Support Vector Machines (SVM), Logistic Regression, and k-Nearest Neighbor Classifiers. We found that ranking models developed using documents that are represented in a topic space (created via the LDA algorithm) gives better ranking scores than using the typical keyword-based ranking method (e.g., Whoosh) alone in a study conducted on several labeled e-discovery datasets deployed in TREC. We also compared the performance of classifiers built on LDA to those based on different document modeling methods such as TF-IDF and LSA. It was surprising to note that we can achieve reasonable classification performance by using less complex models (with low computational cost) such as LSA

and TF-IDF. (The TF-IDF scheme may not be ideal for large datasets as it can encounter computational difficulties in training a classifier.)

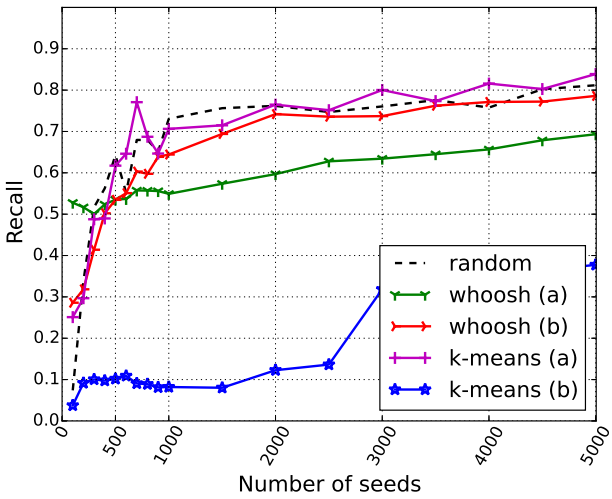
In our experience, different classification methods such as SVM (RBF kernel), SVM (Linear kernel), and logistic regression show mixed classification performance for different datasets as well. This suggests that having identified the right features for documents in a corpus the choice of algorithms to build the optimal classifier is relatively insignificant. It is arguable that the selection of hyperparameters in the LDA model (see Chapters 2–5) might give a better performance for the classifiers (built based on LDA features) employed in this chapter. We performed a preliminary experiment to compare the performance of the LDA models using the empirical Bayes choice of hyperparameters with approaches which use popular default hyperparameter values (Chapter 5) to generate document features for various classifiers. We also considered the number of topics K as a configurable parameter for feature selection. For evaluation, we used two corpora created from the 20Newsgroup dataset. Each corpus consists of documents from two news groups. One of the corpus built was hard to distinguish and the other was easy to distinguish. In our experience, selecting parameters helped improving the classification performance in certain cases, especially when the corpus was hard to distinguish. We cannot make any conclusive remarks unless we perform more experiments. We leave this study to future work.



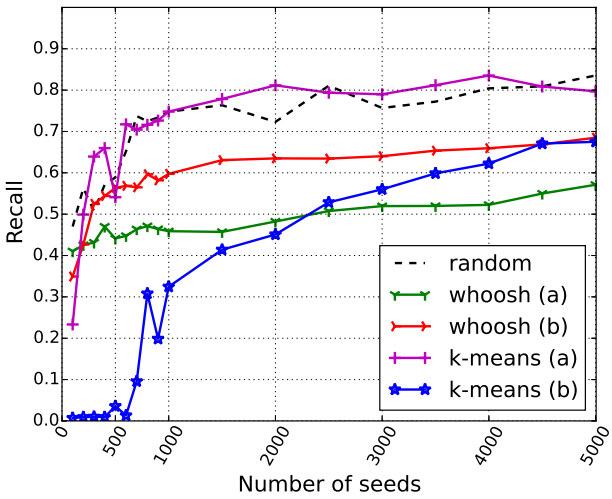
A C-Medicine: *AUC*



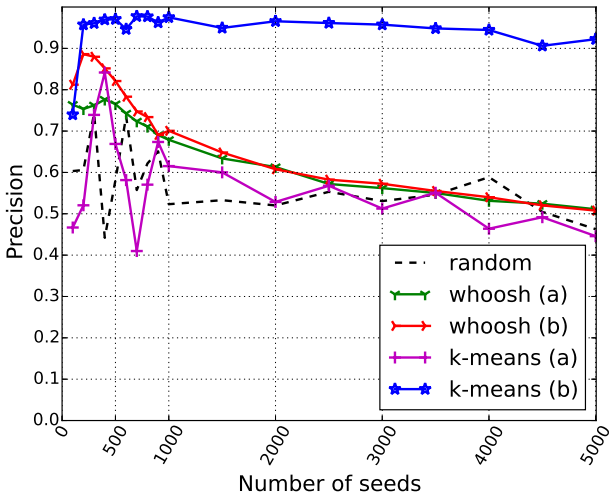
B C-Baseball: *AUC*



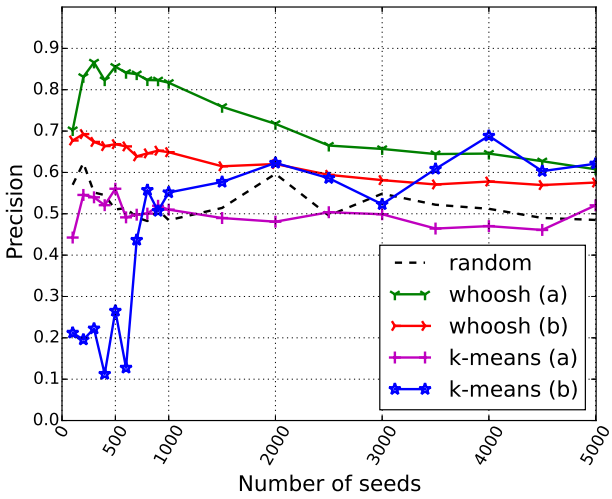
C C-Medicine: *Recall*



D C-Baseball: *Recall*

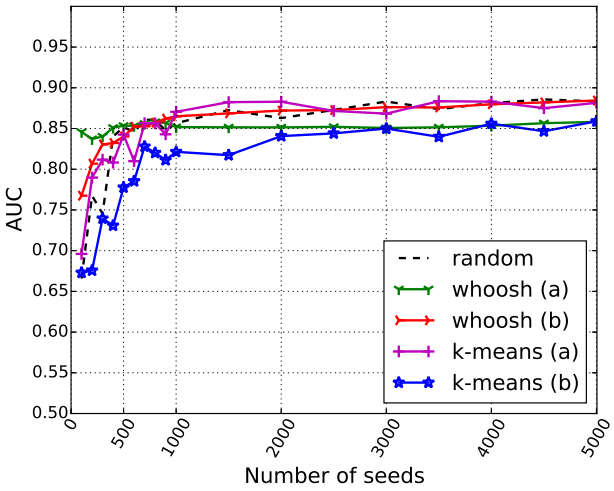


E C-Medicine: *Precision*

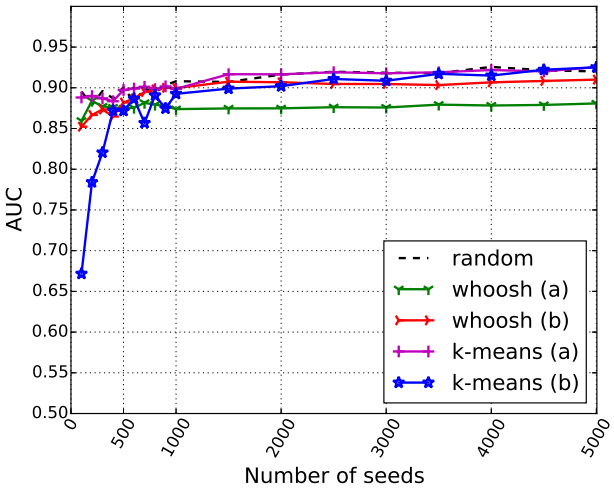


F C-Baseball: *Precision*

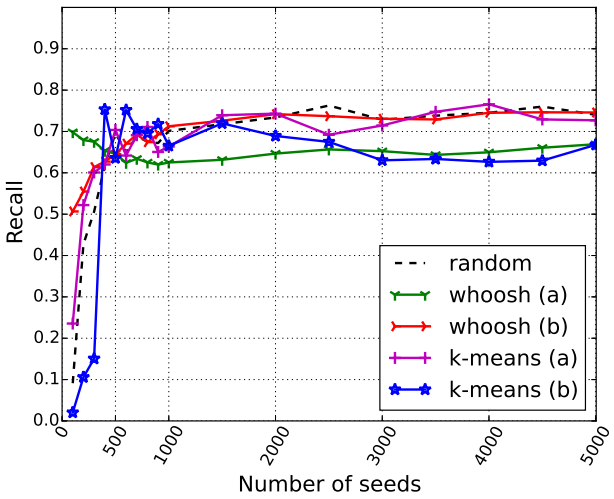
Figure 7-4. Classification performance of various seed selection methods for corpora C-Medicine and C-Baseball. We used the document semantic features (200) generated via the Latent Semantic Analysis algorithm for classifier training and prediction runs



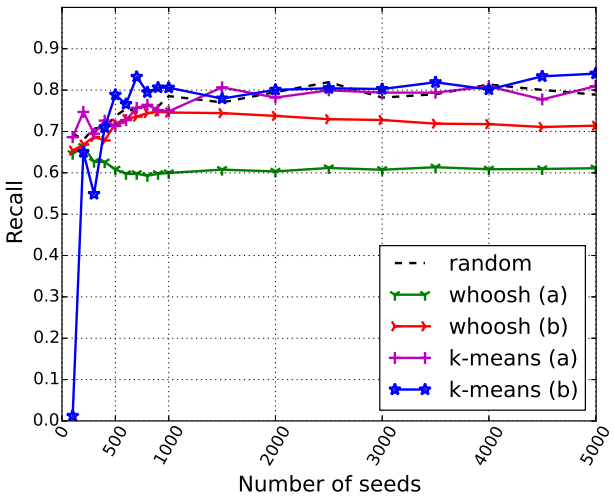
A C-Medicine: *AUC*



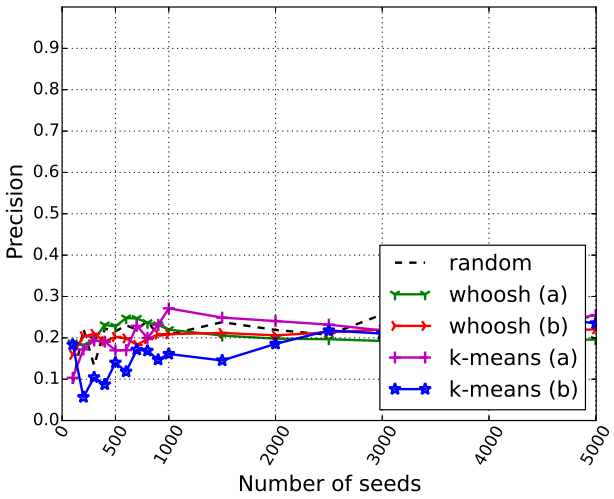
B C-Baseball: *AUC*



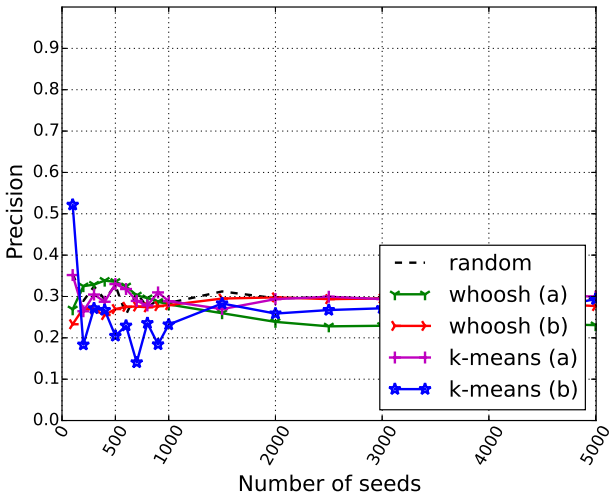
C C-Medicine: *Recall*



D C-Baseball: *Recall*

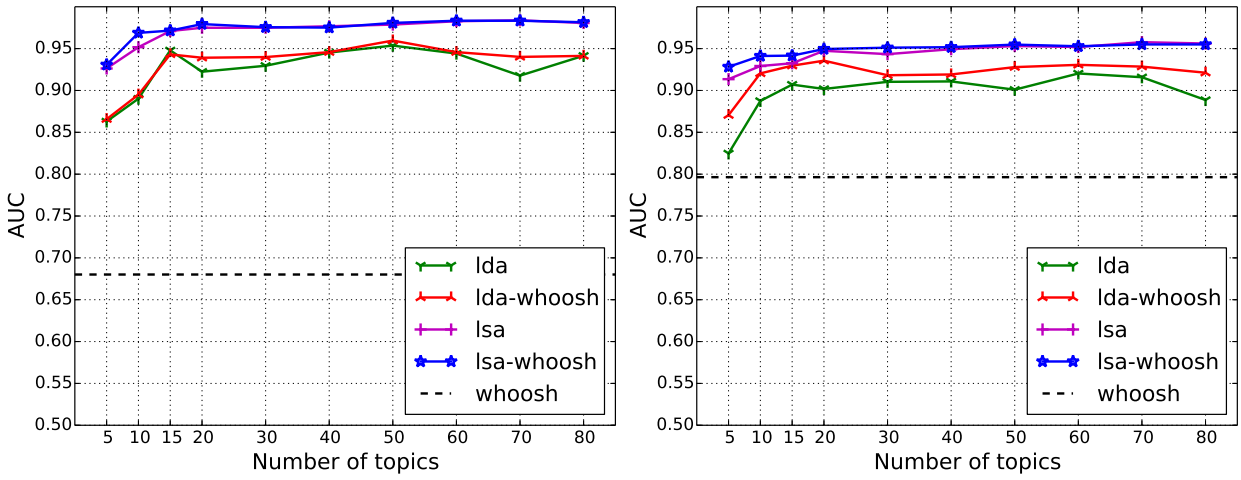


E C-Medicine: *Precision*



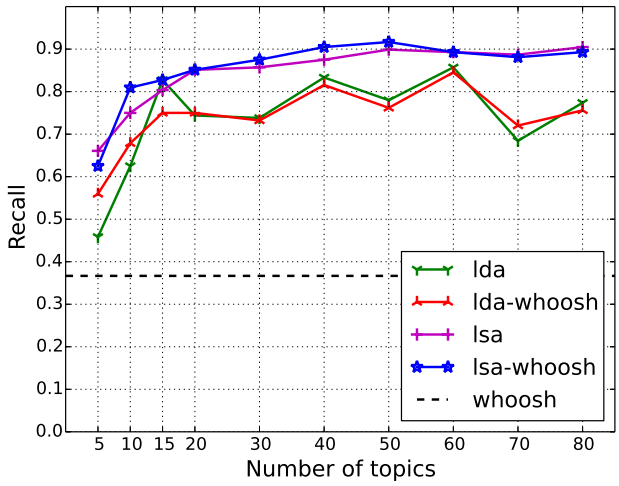
F C-Baseball: *Precision*

Figure 7-5. Classification performance of various seed selection methods for corpora C-Medicine and C-Baseball. We used the document topic features (50) generated via the Latent Dirichlet Allocation algorithm for classifier training and prediction runs.

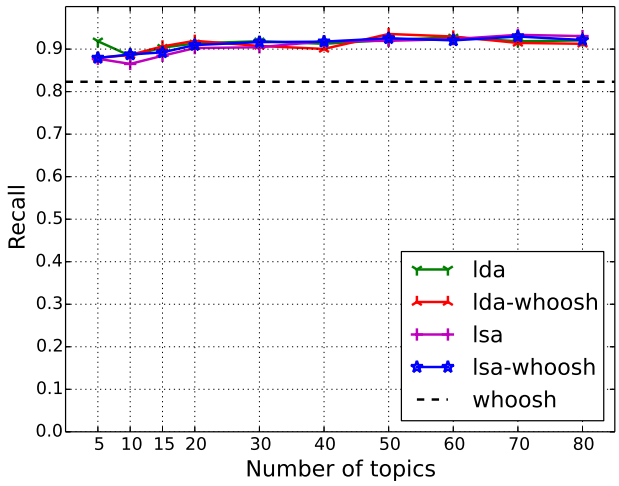


A C-201: *AUC*

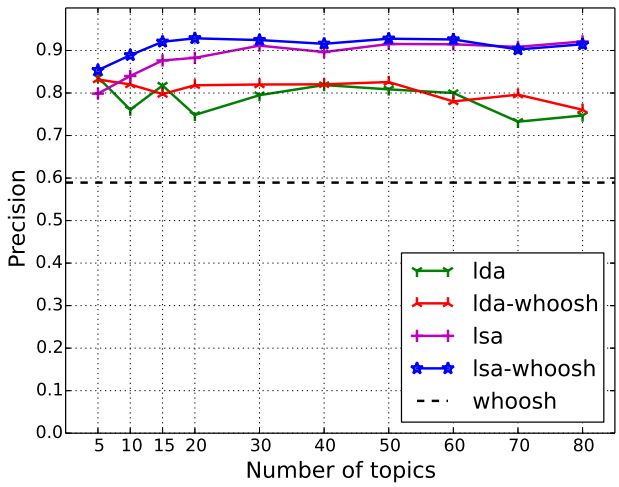
B C-202: *AUC*



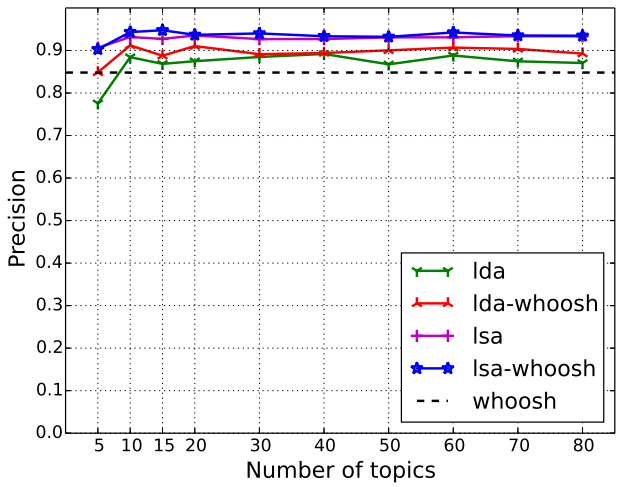
C C-201: *Recall*



D C-202: *Recall*

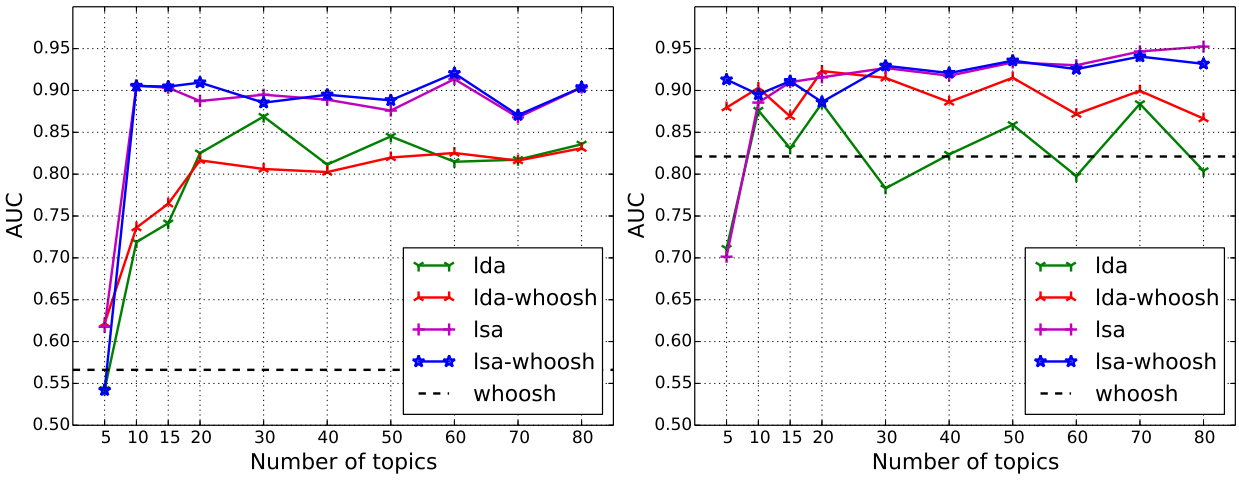


E C-201: *Precision*



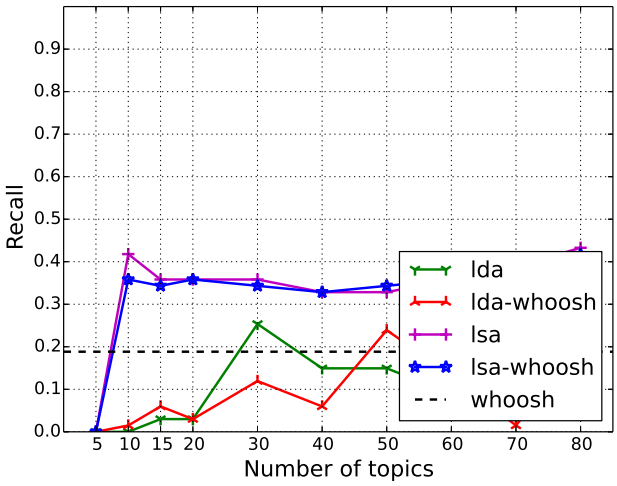
F C-202: *Precision*

Figure 7-6. Classification performance of various SVM models (based on document topic mixtures and Whoosh scores) vs. Whoosh retrieval for corpora C-201 and C-202.

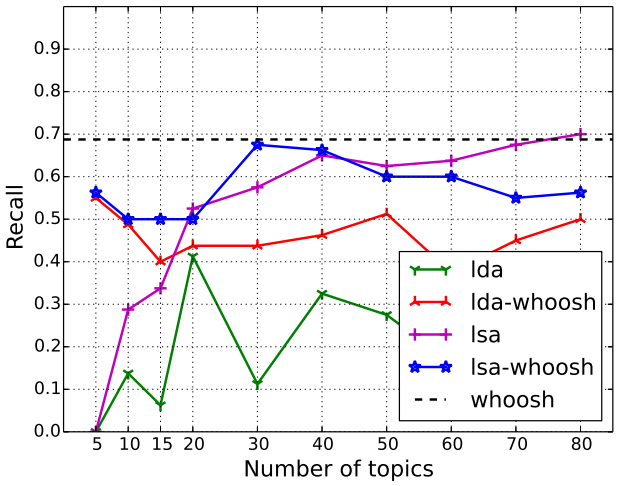


A C-203: *AUC*

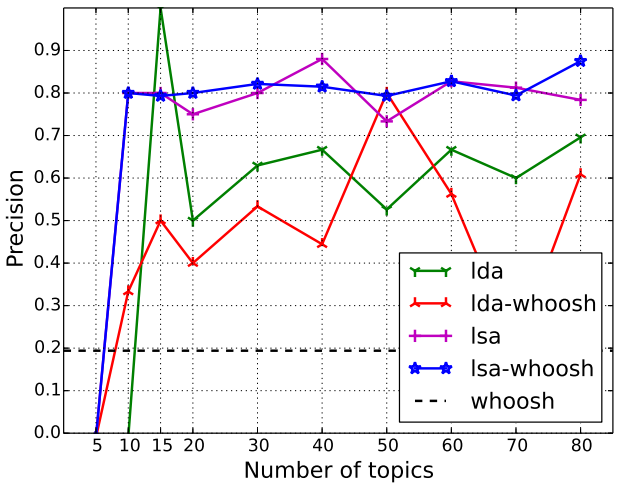
B C-207: *AUC*



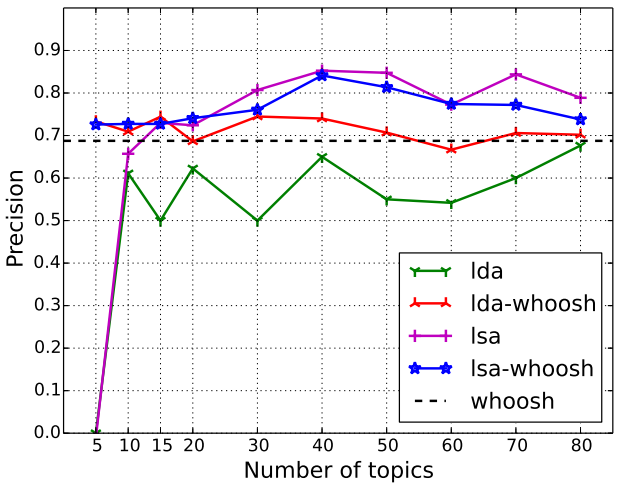
C C-203: *Recall*



D C-207: *Recall*



E C-203: *Precision*



F C-207: *Precision*

Figure 7-7. Classification performance of various SVM models (based on document topic mixtures and Whoosh scores) vs. Whoosh retrieval for corpora C-203 and C-207.

CHAPTER 8
SELECTING THE NUMBER OF TOPICS IN THE LATENT DIRICHLET ALLOCATION MODEL: A SURVEY

The hierarchical model of Latent Dirichlet Allocation is indexed by the number of topics K and the hyperparameter h (i.e. $(\eta, \boldsymbol{\alpha}) \in (0, \infty)^{K+1}$, see Chapter 2). In Chapter 2, we have suppressed the role of K in the model by assuming it to be known for a given corpus. We have then seen the role of the hyperparameter h in inference and an described an efficient method for selecting h. The choice of K can have an impact on inference: for example, if we use a K that is larger than the optimal number of topics in the corpus for inference from the LDA model, we may end up getting duplicate or meaningless topics. In addition, the hyperparameters and the number of topics in the model are interconnected: for example, changing η can be expected to reduce or increase the number of topics in the model, due to η's impact on sparsity in the LDA posterior (Griffiths and Steyvers, 2004). This chapter gives a literature survey of methods to identify the number of topics in the LDA model for a given dataset, and discusses possible improvements to some of these methods.

8.1 Selecting K Based on Marginal Likelihood

In Bayesian statistics, one way to identify the most suitable model for a given dataset from a set of models is to select the model that has the highest *marginal likelihood*. The marginal likelihood or *evidence* of a model is the probability that the model gives to the observed data (i.e., the observed words $\boldsymbol{w}$ in a corpus) (Neal, 2008). From the LDA hierarchical model, the marginal likelihood, $m_{\boldsymbol{w}}(h, K) = p^{(h,K)}(\boldsymbol{w})$, is a function of h and K, after integrating out all of the latent variables of the model. Griffiths and Steyvers (2004) took selecting the number of topics K for the LDA model, given a corpus and fixed h, as the problem of model selection. We now give an overview of the approach here. The hyperparameter h is fixed and is suppressed in the notation henceforth. As we all know, the computation of the marginal likelihood $m_{\boldsymbol{w}}(K)$ for the LDA model is intractable due to the requirement of higher dimensional integration. Griffiths and Steyvers suggested

the use of the harmonic mean of the likelihood evaluated at the posterior distribution $p^{(K)}(\boldsymbol{z} \mid \boldsymbol{w})$ as an approximate of $m_{\boldsymbol{w}}(K)$. One can compute the harmonic mean by (Newton and Raftery, 1994; Wallach et al., 2009b)

$$\frac{1}{p^{(K)}(\boldsymbol{w})} = \sum_{\boldsymbol{z}} \frac{p^{(K)}(\boldsymbol{z} \mid \boldsymbol{w})}{p^{(K)}(\boldsymbol{w} \mid \boldsymbol{z})}. \tag{8–1}$$

Let $\boldsymbol{z}^{(1)}, \boldsymbol{z}^{(2)}, \boldsymbol{z}^{(3)}, \ldots$ be the samples from the posterior $p^{(K)}(\boldsymbol{z} \mid \boldsymbol{w})$, then we can approximate the right hand side by

$$\sum_{\boldsymbol{z}} \frac{p^{(K)}(\boldsymbol{z} \mid \boldsymbol{w})}{p^{(K)}(\boldsymbol{w} \mid \boldsymbol{z})} \approx \frac{1}{S} \sum_{s=1}^{S} \frac{1}{p^{(K)}(\boldsymbol{w} \mid \boldsymbol{z}^{(s)})} \tag{8–2}$$

For example, one can utilize the samples generated from the collapsed Gibbs sampling (CGS, Griffiths and Steyvers, 2004) chain of LDA, which is a Markov chain on $\boldsymbol{z}$, to compute this expectation. The error in this approximation can be low with an ample number of $\boldsymbol{z}$ samples from $p^{(K)}(\boldsymbol{z} \mid \boldsymbol{w})$. Once we have the estimate of $p^{(K)}(\boldsymbol{w})$ via the harmonic mean method, we can find K by:

$$\hat{K} = \arg\max_{K} p^{(K)}(\boldsymbol{w}) \tag{8–3}$$

To evaluate this method Griffiths and Steyvers used a dataset that consists of 28,154 abstracts of the PNAS publications from 1991 to 2001. For various choices of K, they ran the CGS chain to sample $\boldsymbol{z}$'s from the posterior distribution $p^{(K)}(\boldsymbol{z} \mid \boldsymbol{w})$ for a constant hyperparameter $h = (\eta, \alpha) = (.1, 50/K)$, i.e., symmetric Dirichlet priors for the LDA models. The study found that the marginal likelihood peaked at $K = 300$ for this dataset.

Even though the study showed reasonable results, this approach discarded the choice of h, which might affect the number of topics K in the model (We leave this to the future research). In addition, using the harmonic mean to estimate the marginal likelihood of the data given a model is suboptimal because (a) the harmonic mean estimator is very likely unable to measure the effects of the prior in a Bayesian model (Neal, 2008), and (b) the

estimator is based on the inverse likelihood which often has infinite variance (Chib, 1995; Neal, 2008).

8.2 Selecting K Based on Predictive Power

An issue with using the marginal likelihood of the training data for model selection is *over-fitting*, which is a well-known problem in machine learning. In general, over-fitted models will have poor predictive performance. One solution to deal with this problem is to evaluate an LDA model fitted on a set of training documents by checking the predictive probability of unobserved, held-out (or test) documents (Blei et al., 2003; Wallach et al., 2009b) given by the model. The intuition behind this approach is that a better model will yield high probability for the documents in the test set. We now give a very brief explanation for this method here.

Let $\boldsymbol{w}'$ be the set of training documents and $\boldsymbol{w}$ be the set of test documents. From the hierarchical model of LDA (Chapter 2), recall $\nu_{h,K,\boldsymbol{w}'}(\boldsymbol{\psi}')$ represents the posterior distribution of $\boldsymbol{\psi}' = (\boldsymbol{\beta}', \boldsymbol{\theta}', \boldsymbol{z}')$ given the observed data $\boldsymbol{w}'$ and the number of topics K corresponding to ν_h, a prior distribution on $\boldsymbol{\psi}'$. One can write the probability of the set of test documents $\boldsymbol{w}$ given the posterior $\nu_{h,K,\boldsymbol{w}'}(\boldsymbol{\psi}')$ as:

$$p^{(h,K)}(\boldsymbol{w} \mid \boldsymbol{w}') = \int p^{(h,K)}(\boldsymbol{w}|\boldsymbol{\psi}')\nu_{h,K,\boldsymbol{w}'}(\boldsymbol{\psi}')d\nu_{h,K,\boldsymbol{w}'}(\boldsymbol{\psi}') \tag{8–4}$$

This integral is computationally intractable for most datasets. Wallach et al. (2009b) suggested to approximate this integral via evaluating at a single point estimate, $\hat{\boldsymbol{\psi}}' = (\hat{\boldsymbol{\beta}}', \hat{\boldsymbol{\theta}}', \hat{\boldsymbol{z}}')$, as follows. In the hierarchical model of LDA, the topic assignments for words in a document are independent of the topic assignments for words in all other documents in the corpus. That means we can compute $p^{(h,K)}(\boldsymbol{w}_d \mid \hat{\boldsymbol{\psi}}')$ individually. We can then write:

$$p^{(h,K)}(\boldsymbol{w} \mid \hat{\boldsymbol{\psi}}') = \prod_{d=1}^{D} p^{(h,K)}(\boldsymbol{w}_d \mid \hat{\boldsymbol{\psi}}') \tag{8–5}$$

In addition, these probabilities are only depended on the single point estimate $\hat{\boldsymbol{\beta}}'$ in $\hat{\boldsymbol{\psi}}'$, which is shared among all documents in the corpus, i.e., $\boldsymbol{w} \cup \boldsymbol{w}'$.

To compute the predictive probability of the held-out documents, we need to estimate the likelihood $p^{(h,K)}(\boldsymbol{w}_d \,|\, \hat{\boldsymbol{\beta}}')$, which is an intractable integral as follows

$$p^{(h,K)}(\boldsymbol{w}_d \,|\, \hat{\boldsymbol{\beta}}') = \int \sum_{\boldsymbol{z}_d} p^{(h,K)}(\boldsymbol{w}_d, \boldsymbol{z}_d, \boldsymbol{\theta}_d \,|\, \hat{\boldsymbol{\beta}}') d\boldsymbol{\theta}_d, \tag{8–6}$$

where $\boldsymbol{z}_d$ represents the vector of latent topic assignments and $\boldsymbol{\theta}_d$ represents the document specific topic distribution, for the held-out document $\boldsymbol{w}_d$. A popular alternative to solving this problem is to estimate the normalizing constant in the formulation (Wallach et al., 2009b):

$$p^{(h,K)}(\boldsymbol{z}_d \,|\, \boldsymbol{w}_d, \hat{\boldsymbol{\beta}}') = \frac{p^{(h,K)}(\boldsymbol{z}_d, \boldsymbol{w}_d \,|\, \hat{\boldsymbol{\beta}}')}{p^{(h,K)}(\boldsymbol{w}_d \,|\, \hat{\boldsymbol{\beta}}')}. \tag{8–7}$$

Wallach et al. (2009b) reported several methods to estimate the normalizing constant, which include the harmonic mean method and importance sampling. Given a training data $\boldsymbol{w}'$ and a specified hyperparameter h, one can use any such method to compute the predictive probability of test documents given an LDA model. Since the predictive probability of held-out documents is a function of both K and h, we can use

$$\hat{K} = \arg\max_{K} p^{(h,K)}(\boldsymbol{w} \,|\, \hat{\boldsymbol{\beta}}')$$

to find K, for a given h. To get an estimate of $\hat{K}$ that is general to the whole corpus, one can consider the use of cross validation in selecting test and training sets. This approach can possibly solve the issue of over-fitting. On the other hand, evaluating the integral in Equation 8–4 using a single point estimate, $\hat{\boldsymbol{\psi}}'$, can cause serious inconsistencies. Chen (2015) gives an alternative Monte Carlo scheme to estimate this integral.

8.3 Selecting K Based on Human Readability

Another interesting option to explore in finding the right number of topics in the LDA model given a corpus is to consider only the topics that are sensible to humans. This method needs an evaluation metric that can capture the human perception of topics, which are probability distributions over the terms in a vocabulary, in the LDA model. One can use that score to prune low-quality topics from the whole set of topics identified

for the corpus. In the literature, topic models are typically evaluated by (a) checking an external classification or information retrieval task that uses the topics from a fitted LDA model (Wei and Croft, 2006) or (b) checking the predictive probability of held-out documents given by the fitted model (Blei et al., 2003; Wallach et al., 2009b) as described in the previous section. Recent studies (Chang et al., 2009) on topic models such as Latent Dirichlet Allocation showed that the latter approach (b) may not give a good measure of human perception of topics. In addition, the main focus of the methods (a) and (b) is to evaluate the whole topic model of interest rather than individual topics in the topic model.

To identify semantically incoherent topics, Mimno et al. (2011) explored several evaluation methods based on human coherence judgments of topics in a fitted LDA model. The first method uses the size of a topic to compare various topics. To compute the size of a topic in the LDA model for a corpus, they used samples generated from the posterior of the latent variable $\boldsymbol{z}$ given $\boldsymbol{w}$, e.g., samples from the CGS chain. They then estimated the size of topic as the number of words assigned to each topic in the CGS chain for a corpus. To evaluate this method, they did a user study that confirmed the utility of this approach. But, specific and fine grained topics in a corpus can have relatively few words assigned to them. In this scenario, topic size may not be the right choice to evaluate topics.

The second method for comparing topics utilizes the *coherence score* for a topic in the LDA model of a corpus, based on the most probable words in the topic. The most probable words for a topic are determined by sorting the vocabulary words assigned to each topic in the descending order of topic specific probabilities. The topic specific probabilities, i.e., the elements in each β_j row, are typically inferred via Gibbs sampling or variational methods. The most probable words for a topic are typically presented to end users to label a topic in the LDA model. Let $v_1^{(j)}, v_2^{(j)}, \ldots, v_M^{(j)}$ be the list of M most probable terms in the corpus vocabulary for topic j, and let $\mathrm{df}(v_t)$ be the *document frequency* of term v_t, i.e., the number of documents in the corpus which have the term

v_t. Let $\text{df}(v_m, v_l)$ be the *co-document frequency* of the terms v_m and v_l, i.e., the number of documents in the corpus which have both of the terms v_m and v_l. For each topic $j = 1, 2, \ldots, K$ in the corpus, the *coherence score* is defined as (Mimno et al., 2011):

$$\text{topic-coherence}_j = \sum_{m=2}^{M} \sum_{l=1}^{m} \log \frac{\text{df}(v_m^{(j)}, v_l^{(j)}) + 1}{\text{df}(v_t^{(j)})} \tag{8–8}$$

The intuition behind computing this score is that there are chances that group of words belonging to a single topic will co-occur with in a document in the corpus, but it is unlikely that words belonging to different topics will appear in a document together. Note that it is not a probabilistic score, rather a score based on the relative frequency of the most probable words for a topic in the corpus. Mimno et al. employed the score in Equation 8–8 to evaluate an LDA model fitted on a National Institute of Health (NIH) dataset. The *coherence score* demonstrated good qualitative behavior in terms of human perception of topics, when it was compared with human judgments of observed coherence (measured on a 3-point scale based on the most probable words of topics), for the fitted topics in the LDA model.

Lau et al. (2014) considered the same problem, i.e., measuring human interpretability of individual topic distributions identified for the LDA model of a corpus. This work was an extension of Chang et al. (2009)'s work on evaluating semantic coherence of topics by *word intrusion*. Intruder words are the words with very low probability in a topic of interest. Lau et al. (2014) inserted intruder words into the set of most probable words for a topic arbitrarily, and human evaluators were asked to identify the intruder words. They then defined a score based on the number of intruder words to compare various topics. The intuition behind this method was that the intruder words are more easily recognizable in semantically coherent topics than in incoherent topics. Lau et al. automated the human involvement in identifying intruder words proposed a better model for topic modeling. But there is no study of the robustness of this scheme in a real-world scenario is available.

8.4 Hierarchical Dirichlet Processes

Teh et al. (2006) introduced the Hierarchical Dirichlet processes (HDP) for the purpose of Bayesian nonparametric modeling of several distributions believed to be related. Suppose we have q populations, and that for population l, $l = 1, \ldots, q$, there are observations $Y_{lj} \overset{\text{indep}}{\sim} F_{\psi_{lj},\sigma_{lj}}$, $j = 1, \ldots, n_l$. Here, $F_{\psi_{lj},\sigma_{lj}}$ is a distribution depending on some unobserved (latent) variable ψ_{lj} and possibly also on some other known parameter σ_{lj} particular to the lj-th individual. We assume that $\psi_{lj} \overset{\text{iid}}{\sim} G_l$, $j = 1, \ldots, n_l$, and that for $l = 1, \ldots, q$, $G_l \overset{\text{iid}}{\sim} \mathcal{D}_{G_0,\alpha}$, the Dirichlet process with base probability measure G_0 and precision parameter $\alpha > 0$ (Ferguson, 1973, 1974). As is well known (and is discussed below), for each l, the latent variables ψ_{lj}, $j = 1, \ldots, n_l$ form clusters, with the ψ_{lj}'s in the same cluster being equal. This can be seen most transparently through the Sethuraman (1994) construction of the Dirichlet process, which says that we may represent G_l as $G_l = \sum_{s=1}^{\infty} \beta_{ls}\delta_{\phi_{ls}}$, where $\phi_{l1}, \phi_{l2}, \ldots$ are independent random variables distributed according to G_0, and $\beta_{l1}, \beta_{l2}, \ldots$ are also random, with a distribution depending on α. Since $\psi_{lj} \overset{\text{iid}}{\sim} G_l$, and G_l is discrete, there will be groups of ψ_{lj}'s that are drawn from the same atom, and hence the clustering property.

Teh et al. (2006) discuss a number of applications, including genomics, hidden Markov models, and topic modeling, in which it is desirable to model the distributions of the Y_{lj}'s as mixtures, and to have mixture components shared among the distributions of the Y_{lj}'s in different populations. They note that this property is obtained if we take G_0 itself to have a Dirichlet process prior, $G_0 \sim \mathcal{D}_{\mathcal{K},\gamma}$, where $\mathcal{K}$ is a probability distribution and $\gamma > 0$. This is because G_0 is then discrete, $G_0 = \sum_{s=1}^{\infty} \beta_{0s}\delta_{\phi_{0s}}$, and so the atoms of the G_l's are all drawn from the atoms of G_0. In the case of topic modeling, we have a corpus of q documents, with document l containing n_l words. These words come from a vocabulary $\mathcal{V}$ of size V. For word j of document l, Y_{lj}, we imagine that there exists a *topic* ψ_{lj}, from which the word is drawn. Here, a topic is by definition a distribution on $\mathcal{V}$, i.e. a topic is

a point in the V-dimensional simplex $\mathbb{S}_V$. Typically, the distribution $\mathcal{K}$ is a member of a known parametric family $\{\mathcal{K}_\omega, \omega \in \Omega\}$, and choosing it reduces to choosing ω.

The hyperparameter specifying the hierarchical Dirichlet processes is the three-dimensional vector $h = (\omega, \gamma, \alpha)$, which we now discuss. The hyperparameters γ and α play important roles, among other things determining the extent to which mixture components or topics are shared within and across groups. The role of ω is problem specific. For topic models, we take $\mathcal{K}_\omega = D_V(\omega, \ldots, \omega)$, a symmetric Dirichlet distribution on $\mathbb{S}_V$, so the parametric family is $\{\mathcal{K}_\omega, \omega > 0\}$, the set of all symmetric Dirichlet distributions on $\mathbb{S}_V$. When ω is large, the topics tend to be probability vectors which spread their mass evenly among many words in the vocabulary, whereas when ω is small, the topics tend to put most of their mass on only a few words. It is clear that the hyperparameter h plays a critical role in this model, and that its value has an important impact on inference and the number of topics in the corpus. Currently, there does not exist a method for choosing h that has a rigorous mathematical basis.

One can consider HDP as a model-based alternative to infer the number of topics K from data. It formulates each document's topic distribution (i.e. the distribution of the Y_{lj} for document l) as a probability vector of infinite length. That means one doesn't have to specify K for the HDP model. But, estimation by doing finite truncation to the prior Dirichlet processes can be sensitive, and often ends up doing inference about high-dimensional term-topic membership vectors (Taddy, 2011).

8.5 Summary

This chapter describes three methods from the machine learning literature to select the number of topics K in the latent Dirichlet allocation model (LDA) for a given corpus. All three methods are based on the output of the LDA model. Lastly, we described a model-based approach to find the number of topics from the data based on the concept of infinite mixture models. In summary, none of these methods have a clear lead on finding the number of topics K in a corpus, and their shortcomings are mainly: (a) the

computational cost for these procedures can be huge, especially for the first three methods (based on model selection and pruning topics), (b) the selection of hyperparameters in the model can play a role the number of topics, and (c) some of these methods are designed with a specific problem in mind, e.g., the method for pruning topics is tested only on the NIH datasets and based on some predefined human evaluation schemes. Chapters two through five discuss a principled way of selecting the hyperparameters in the LDA model, but selecting the hyperparameters in the HDP model is a challenging problem for which no solution has been presented.

CHAPTER 9
CONCLUSIONS

This chapter concludes this dissertation and describes potential future work directions.

Chapters two through five gave an overview of the hierarchical model of the Latent Dirichlet Allocation (LDA) model and an analysis of the importance of choosing hyperparameters in the model, using a set of synthetic corpora. We presented a method based on a combination of Markov chain Monte Carlo and importance sampling to get the maximum likelihood estimate of the hyperparameters. This can be viewed as a method for empirical Bayes analysis in which, the prior of the model is estimated from the data. Our empirical study, using both synthetic and real datasets, showed that the LDA models indexed by the empirical Bayes choice of hyperparameters outperform the LDA models that are indexed by the default choices of hyperparameters employed in the literature. The case study of various models, using the two evaluation schemes that we described, also suggests that some of the default choices of hyperparameters should not be used in practice.

In Chapter 7, we compared various document modeling methods such as TF-IDF, Latent Semantic Analysis (LSA) with LDA to represent e-discovery documents. We then formulated the problem of discovering relevant documents as the problem of binary document classification in the representation space. We used popular document classifiers such as SVM and logistic regression for training. The experimental results suggest that we can achieve reasonable classification performance by using simpler models such as LSA, with low computational cost. In addition, we noticed that the classification models based on TF-IDF, LSA, and LDA produce mixed classification performance on datasets created from the Enron dataset and the 20Newsgroups dataset. It is possible that there is no single classifier that is suitable for solving all classification problems. One can consider combining decision statistics from multiple classifier models to yield more robust results.

In the future, we are also interested in the following problems.

Stochastic Search Algorithms for Estimating $\arg\max_h B(h)$. We are interested in estimating $\hat{h} = \arg\max_h m(h)$, or equivalently, $\arg\max_h B(h)$. Empirical Bayes inference then uses the posterior distribution corresponding to the prior $\nu_{\hat{h}}$. The approach described in Chapters two through five is to form estimates $\hat{B}(h)$ of $B(h)$ as h varies over a fine grid and then find $\arg\max_h \hat{B}(h)$ via grid search. This approach works only when the dimension of h is very low ($\dim(h)$ is 1 or 2, possibly 3). A useful alternative approach is stochastic search, recently proposed by Atchadé (2011).

Finding the Number of Topics in a Corpus. Chapter 8 gave an overview of three popular approaches in the literature to select the number of topics K in the LDA model of a given corpus. We also mentioned a model-based approach to find the number of topics from the data, i.e., Hierarchical Dirichlet Processes (HDP), based on the concept of infinite mixture models. But, none of these methods have a clear lead on finding the number of topics K in a corpus.

Selecting hyperparameters has an effect on the number of topics to be selected for the LDA model. In the future, we would like to study whether finding optimal hyperparameters for the model can help in finding the number of topics for a corpus.

APPENDIX A
A NOTE ON BLEI ET AL. (2003)'S APPROACH FOR INFERENCE AND PARAMETER ESTIMATION IN THE LDA MODEL

We first describe the hierarchical model of latent Dirichlet allocation (LDA) used in Blei et al. (2003) in terms of our notation. We then discuss the variational method for inference and the empirical Bayes method for parameter estimation in LDA using the variational method output.

The hierarchical model discussed in Blei et al. (2003, Section 5) differs from the model described in Chapter 2 in line 1. Blei et al. (2003) assume the $K \times V$ topic matrix $\boldsymbol{\beta}$ as a fixed quantity (i.e., it is not random) which is to be estimated. Based on this reduced hierarchical model, the probabilities of interest are the posterior of the latent variables θ_d and $\boldsymbol{z}_d$ given document d (useful for inference) and the marginal likelihood of the data (useful for empirical Bayes methods). Let $\mathcal{A} = (0, \infty)^K$ be the hyperparameter space. For any $\boldsymbol{\alpha} = (\alpha_1, \ldots, \alpha_K) \in \mathcal{A}$, $\nu_{\boldsymbol{\alpha}}$ and $\nu_{\boldsymbol{\alpha},\boldsymbol{\beta},\boldsymbol{w}_d}$ are distributions on a vector for which some components are continuous and some are discrete. We use $\ell_{\boldsymbol{w}_d}(\theta_d, \boldsymbol{z}_d, \boldsymbol{\beta})$ to denote the likelihood function for document d (which is given by line 4 of the LDA model). Then, the posterior of θ_d and $\boldsymbol{z}_d$ given the observed words $\boldsymbol{w}_d$ is given by (using the Bayes rule)

$$\nu_{\boldsymbol{\alpha},\boldsymbol{\beta},\boldsymbol{w}_d}(\theta_d, \boldsymbol{z}_d) = \frac{\ell_{\boldsymbol{w}_d}(\theta_d, \boldsymbol{z}_d, \boldsymbol{\beta})\nu_{\boldsymbol{\alpha}}(\theta_d, \boldsymbol{z}_d)}{m_d(\boldsymbol{\alpha}, \boldsymbol{\beta})}, \tag{A–1}$$

where the normalization constant $m_d(\boldsymbol{\alpha}, \boldsymbol{\beta})$ is the marginal likelihood of the observed data $\boldsymbol{w}_d$, which is a function of $\boldsymbol{\alpha}$ and $\boldsymbol{\beta}$. From the hierarchical model, the prior $\nu_{\boldsymbol{\alpha}}$ is given by (by lines 2–3 of the LDA model)

$$\nu_{\boldsymbol{\alpha}}(\theta_d, \boldsymbol{z}_d) = p^{(\boldsymbol{\alpha})}_{\boldsymbol{z}_d \,|\, \theta_d}(\boldsymbol{z}_d \,|\, \theta_d) p^{(\boldsymbol{\alpha})}_{\theta_d}(\theta_d) \tag{A–2}$$

In general, Equation A–1 is intractable to compute due to the high dimensionality of the latent variable space. Therefore, Blei et al. (2003) looked at variational methods for finding deterministic approximations to the posterior distribution of latent variables and the marginal likelihood of the data.

Variational methods (Bishop et al., 2006, Chapter 10) are based on the concept of *functional derivatives* in the field of *calculus of variations*. A functional is a mapping function that takes a function as the input and returns a scalar output. The functional derivative describes how the output value varies as we make minute changes to the input function that the functional depends on. In variational methods the quantity being optimized is a functional, but one usually restricts the range of functions over which the optimization is performed.

We now describe how variational methods help us to identify approximations for the posterior $\nu_{\boldsymbol{\alpha},\boldsymbol{\beta},\boldsymbol{w}_d}(\theta_d, \boldsymbol{z}_d)$ and the marginal likelihood of the data $m_d(\boldsymbol{\alpha}, \boldsymbol{\beta})$. Let $q(\theta_d, \boldsymbol{z}_d)$ be any distribution over the latent variables θ_d and $\boldsymbol{z}_d$ (We will describe more about this distribution later), and let $\nu_{\boldsymbol{\alpha},\boldsymbol{\beta}}(\theta_d, \boldsymbol{z}_d, \boldsymbol{w}_d)$ be the joint probability of θ_d, $\boldsymbol{z}_d$, and $\boldsymbol{w}_d$ based on the hierarchical model. We can then break up the log marginal probability of the data $\boldsymbol{w}_d$ as (Bishop et al., 2006)

$$\log m_d(\boldsymbol{\alpha}, \boldsymbol{\beta}) = \mathcal{L}_d(q, \nu_{\boldsymbol{\alpha},\boldsymbol{\beta}}) + \mathrm{KL}_d(q, \nu_{\boldsymbol{\alpha},\boldsymbol{w}_d,\boldsymbol{\beta}}) \tag{A–3}$$

where[1]

$$\mathcal{L}_d(q, \nu_{\boldsymbol{\alpha},\boldsymbol{\beta}}) = \int \sum_{\boldsymbol{z}_d} q(\theta_d, \boldsymbol{z}_d) \log \left\{ \frac{\nu_{\boldsymbol{\alpha},\boldsymbol{\beta}}(\theta_d, \boldsymbol{z}_d, \boldsymbol{w}_d)}{q(\theta_d, \boldsymbol{z}_d)} \right\} d\theta_d \tag{A–4}$$

and

$$\mathrm{KL}_d(q, \nu_{\boldsymbol{\alpha},\boldsymbol{w}_d,\boldsymbol{\beta}}) = -\int \sum_{\boldsymbol{z}_d} q(\theta_d, \boldsymbol{z}_d) \log \left\{ \frac{\nu_{\boldsymbol{\alpha},\boldsymbol{\beta},\boldsymbol{w}}(\theta_d, \boldsymbol{z}_d)}{q(\theta_d, \boldsymbol{z}_d)} \right\} d\theta_d. \tag{A–5}$$

From Equation A–4, $\mathcal{L}_d(q, \nu_{\boldsymbol{\alpha},\boldsymbol{\beta}})$ is a functional of the distribution $q(\theta_d, \boldsymbol{z}_d)$ and a function of the parameters $\boldsymbol{\alpha}$ and $\boldsymbol{\beta}$. The Kullback-Leibler (KL) divergence specified in Equation A–5 satisfies $\mathrm{KL}_d(q, \nu_{\boldsymbol{\alpha},\boldsymbol{w}_d,\boldsymbol{\beta}}) \geq 0$ (by the positivity of the KL divergence), with equality if, and only if, $q(\theta_d, \boldsymbol{z}_d)$ equals the posterior $\nu_{\boldsymbol{\alpha},\boldsymbol{\beta},\boldsymbol{w}}(\theta_d, \boldsymbol{z}_d)$. It therefore

[1] The summation $\sum_{\boldsymbol{z}_d}$ represents the summation over all z_{di}s for document d. We use summation instead of an integral because z_{di}s are discrete.

follows from Equation A–3 that $\mathcal{L}_d(q, \nu_{\boldsymbol{\alpha},\boldsymbol{\beta}})$ is a lower-bound for the log marginal probability. We can maximize the lower-bound $\mathcal{L}_d(q, \nu_{\boldsymbol{\alpha},\boldsymbol{\beta}})$ with respect to $q(\theta_d, \boldsymbol{z}_d)$, which is also equivalent to minimizing the KL-divergence $\mathrm{KL}_d(q, \nu_{\boldsymbol{\alpha},\boldsymbol{w}_d,\boldsymbol{\beta}})$. The tightest lower-bound occurs when the KL divergence vanishes, i.e., when $q(\theta_d, \boldsymbol{z}_d)$ equals the posterior distribution (but it is intractable to work with). Thus, in variational methods, one considers a restricted family of distributions $q(\theta_d, \boldsymbol{z}_d)$ instead of working on the intractable posterior, and then seeks the member of the family for which the lower-bound $\mathcal{L}_d(q, \nu_{\boldsymbol{\alpha},\boldsymbol{\beta}})$ is maximized. One way to restrict the family of approximating distributions is to use a parametric distribution (i.e., variational distribution) that is governed by a set of parameters (i.e., variational parameters). Usually, this parametric distribution is much simpler to work with than the original posterior by assuming independence between respective variables. The goal is then to identify the parameters which give the tightest lower-bound with in the family. For example, Blei et al. (2003) proposed to use a parametric distribution on θ_d and $\boldsymbol{z}_d$

$$q_{\boldsymbol{\gamma},\boldsymbol{\phi}_d}(\theta_d, \boldsymbol{z}_d) = q^{\gamma}_{\theta_d}(\theta_d) \prod_{i=1}^{n_d} q_{z_{di} \,|\, \phi_{di}}(z_{di} \,|\, \phi_{di}) \tag{A–6}$$

in which $q^{\gamma}_{\theta_d}(\theta_d)$ is a Dirichlet probability governed by hyperparameter $\boldsymbol{\gamma} \in (0, \infty)^K$, and $q_{z_{di} \,|\, \phi_{di}}(z_{di} \,|\, \phi_{di})$ is a multinomial probability governed by parameter $\phi_{di} \in \mathbb{S}_K$, i.e., a point in the K-dimensional simplex. The lower-bound $\mathcal{L}_d(q, \nu_{\boldsymbol{\alpha},\boldsymbol{\beta}})$ then becomes a function of $\boldsymbol{\gamma}$ and $\boldsymbol{\phi}_d = (\phi_{d1}, \phi_{d2}, \ldots, \phi_{dn_d})$. We can then apply any standard nonlinear optimization techniques to determine the optimal values for $\boldsymbol{\gamma}$ and $\boldsymbol{\phi}_d$ that maximizes the lower-bound on the marginal likelihood. Blei et al. (2003) employed an *iterative fixed point method* for finding the optimal values $\boldsymbol{\gamma}^*$ and $\boldsymbol{\phi}^*_d$ and used the resulting variational distribution $q_{\boldsymbol{\gamma}^*,\boldsymbol{\phi}^*_d}(\theta_d, \boldsymbol{z}_d)$ as an approximation for the posterior $\nu_{\boldsymbol{\alpha},\boldsymbol{\beta},\boldsymbol{w}_d}(\theta_d, \boldsymbol{z}_d)$ for inference. In addition, they used the optimal lower-bound $\mathcal{L}_d(q^*, \nu_{\boldsymbol{\alpha},\boldsymbol{\beta}})$, a function of $q_{\boldsymbol{\gamma}^*,\boldsymbol{\phi}^*_d}(\theta_d, \boldsymbol{z}_d)$, as the tractable approximation for the log marginal likelihood $\log m_d(\boldsymbol{\alpha}, \boldsymbol{\beta})$.

We now describe the empirical Bayes method employed in Blei et al. (2003) to estimate the parameters $\boldsymbol{\alpha}$ and $\boldsymbol{\beta}$ in the hierarchical model. Given the corpus $\boldsymbol{w} = (\boldsymbol{w}_1, \ldots, \boldsymbol{w}_D)$, we are interested in finding the parameters $\boldsymbol{\alpha}$ and $\boldsymbol{\beta}$ that maximize the log marginal likelihood of the data, i.e.,

$$\log m(\boldsymbol{\alpha}, \boldsymbol{\beta}) = \sum_{d=1}^{D} \log m_d(\boldsymbol{\alpha}, \boldsymbol{\beta}), \tag{A–7}$$

given by the LDA hierarchical model. For each document, one can replace the intractable log marginal likelihood $\log m_d(\boldsymbol{\alpha}, \boldsymbol{\beta})$ by the optimal lower-bound $\mathcal{L}_d(q^*, \nu_{\boldsymbol{\alpha},\boldsymbol{\beta}})$ obtained from the variational method described above. It will then become a lower-bound on the log marginal likelihood of the corpus given by Equation A–7. One can exploit this lower-bound for maximum likelihood parameter estimation via a tractable approximation of the EM algorithm (Neal and Hinton, 1998). The EM algorithm (Dempster et al., 1977) is a two stage, iterative optimization method to find maximum likelihood estimates of parameters in probabilistic models having latent variables. Each iteration of the EM algorithm alternates between (1) an expectation (E) step, which computes the expected value of the log likelihood function, with respect to the posterior distribution of latent variables given the observed data under the current estimate of the parameters in the model (In our case, the expectation is based on the posterior $\nu_{\boldsymbol{\alpha},\boldsymbol{\beta},\boldsymbol{w}_d}(\theta_d, \boldsymbol{z}_d)$) and (2) a maximization (M) step, which computes parameters (In our case, the parameters are $\boldsymbol{\alpha}$, $\boldsymbol{\beta}$) maximizing the expectation computed on the E step. In LDA, the expected value in E-step is intractable to compute to perform exact EM. But, we can replace it with the lower-bound on the log marginal likelihood from the variation method and perform an approximate EM for parameter estimation as follows (Blei et al., 2003):

- **E-step**: For fixed values of $\boldsymbol{\alpha}$ and $\boldsymbol{\beta}$, for each document in the corpus, compute the optimal lower-bound, which is indexed by the optimal variational distribution $q_{\boldsymbol{\gamma}^*,\boldsymbol{\phi}_d^*}(\theta_d, \boldsymbol{z}_d)$, based on the variational optimization method described above.
- **M-step**: Maximize the resulting lower-bound on the log marginal likelihood given by Equation A–7 with respect to parameters $\boldsymbol{\alpha}$ and $\boldsymbol{\beta}$, after fixing $\boldsymbol{\gamma}^*$, $\boldsymbol{\phi}_d^*$.

APPENDIX B
EVALUATION METHODS FOR ELECTRONIC DISCOVERY

B.1 Recall and Precision

Two popular evaluation scores used in information retrieval (IR) to assess the effectiveness of a search or document categorization are *Recall* and *Precision*. Figure B-1[1] shows a graphical representation for these two scores. The outer rectangle represents all documents in a corpus. The inner circle represents documents retrieved by an IR method given a search query. Filled circles represent expert labeled relevant documents and empty circles represent expert labeled non-relevant documents.

We now formally define Recall and Precision. Let TP be the number of *true* positives, FP be the number of *false* positives, and FN be the number of *false* negatives in the retrieved items. Recall—a.k.a. True Positive Rate—"is the fraction of all relevant documents that are retrieved" (Manning et al., 2008), which we can write as:

$$\text{Recall} = \frac{\text{TP}}{\text{TP} + \text{FN}}. \tag{B–1}$$

Precision "is the fraction of retrieved documents that are relevant" (Manning et al., 2008), which we can write as:

$$\text{Precision} = \frac{\text{TP}}{\text{TP} + \text{FP}}. \tag{B–2}$$

B.2 Receiver Operating Characteristic

Receiver Operating Characteristic (ROC, Swets 1996) curve illustrates the performance of a binary (two-class) classifier as its discrimination threshold or *decision value* is varied from its greatest to least value. It is typically drawn using a set of data points (e.g., documents), some of which are positive data points displaying a property of interest (e.g., relevance to the query terms) while others are negative data points. Each data point is

[1] The image is reproduced from https://commons.wikimedia.org/wiki/File:Precisionrecall.svg

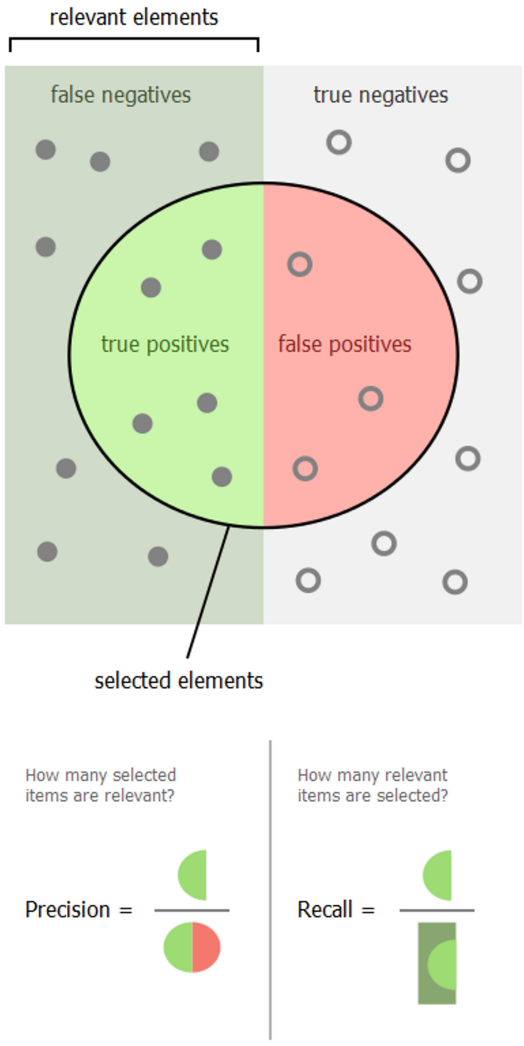


Figure B-1. Recall and Precision

assigned a scalar valued confidence—i.e., *decision value*—that it is positive. In the IR setting, one can use the ranking score of a document given a query from a ranking model as the confidence value. See Table B-1 for a sample dataset. A curve is constructed by varying the confidence value c from its greatest to least value and plotting a curve showing the fraction of true positives (TP) out of the total actual positives, i.e., True Positive Rate (Equation B–1) and the fraction of false positives (FP) out of the total actual negatives,

i.e., False Positive Rate—a.k.a. False Alarm Rate. We can define False Alarm Rate as:

$$\text{False Alarm Rate} = \frac{\text{FP}}{\text{TN} + \text{FP}}, \tag{B–3}$$

where TN represents the number of *true* negatives in the classification output. Figure B-2 shows the ROC curves created for the classification results given in Table B-1. A *Perfect Classifier* will yield a curve that goes from the bottom left corner $(0, 0)$ to the top left $(0, 1)$, then to the top right $(1, 1)$ (Gray line). The worst case of detection referred to as the chance diagonal—*Random Guess* line (Gray dotted line), a straight line from the bottom left corner to the top right. One can use the area under a ROC curve, i.e., AUC, as an estimate of the relative performance of classifiers.

Table B-1. ROC Dataset: Classification output for 10 data points from two hypothetical classifiers.

#	Class labels (Truth)	Decision values	
		Classifier I	Classifier II
1	1	0.98	0.98
2	1	0.89	0.45
3	2	0.81	0.88
4	1	0.79	0.85
5	1	0.70	0.20
6	1	0.69	0.40
7	2	0.50	0.90
8	2	0.49	0.49
9	1	0.45	0.77
10	2	0.20	0.40

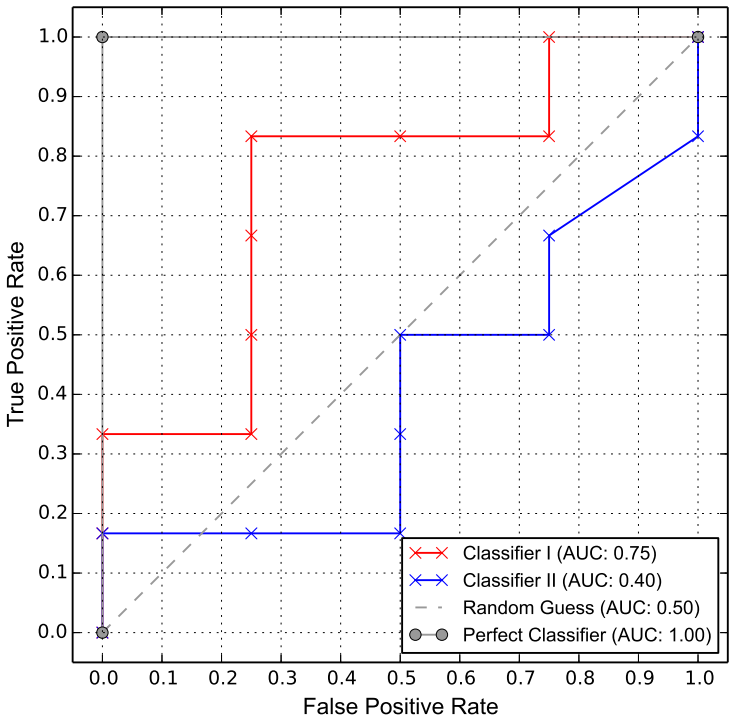


Figure B-2. Plots of ROC curves that compares the output of two hypothetical classifiers described in Table B-1.

Asuncion, A., Welling, M., Smyth, P. and Teh, Y. W. (2009). On smoothing and inference for topic models. In *Proceedings of the Twenty-Fifth Conference on Uncertainty in Artificial Intelligence*. UAI '09, AUAI Press, Arlington, Virginia, United States. 48

Atchadé, Y. F. (2011). A computational framework for empirical Bayes inference. *Statistics and Computing* **21** 463–473. 107

Berry, M. W., Esau, R. and Keifer, B. (2012). *The Use of Text Mining Techniques in Electronic Discovery for Legal Matters*, chap. 8. IGI Global, 174–190. 71

Bird, S., Klein, E. and Loper, E. (2009). *Natural Language Processing with Python*. O'Reilly Media.

URL http://books.google.com/books?id=KGIbfiiP1i4C 75

Bishop, C. M. et al. (2006). *Pattern Recognition and Machine Learning*, vol. 1. Springer, New York. 77, 109

Blei, D. M. (2004). *Probabilistic Models of Text and Images*. Ph.D. thesis, University of California, Berkeley. 16

Blei, D. M., Ng, A. Y. and Jordan, M. I. (2003). Latent Dirichlet allocation. *Journal of Machine Learning Research* **3** 993–1022. 13, 16, 19, 23, 48, 49, 99, 101, 108, 110, 111

Casey, E. (2009). *Handbook of Digital Forensics and Investigation*. Academic Press. 67

Chang, J., Gerrish, S., Wang, C., Boyd-graber, J. L. and Blei, D. M. (2009). Reading tea leaves: How humans interpret topic models. In *Advances in neural information processing systems*. 101, 102

Chaput, M. (2014). Whoosh:fast, pure-python full text indexing, search, and spell checking library.

URL http://pythonhosted.org//Whoosh 76, 78

Chen, Z. (2015). *Inference for the Number of Topics in the Latent Dirichlet Allocation Model via Bayesian Mixture Modelling*. Ph.D. thesis, University of Florida. 53, 100

Chen, Z. and Doss, H. (2015). Inference for the number of topics in the latent Dirichlet allocation model via Bayesian mixture modelling. Tech. rep., Department of Statistics, University of Florida. 42

Chib, S. (1995). Marginal likelihood from the gibbs output. *Journal of American Statistical Association* . 99

Cochran, W. G. (1977). *Sampling Techniques*. 2nd ed. John Wiley and Sons, New York, NY. 74

Cormack, G. V. and Grossman, M. R. (2015). Autonomy and reliability of continuous active learning for technology-assisted review. *CoRR* **1504.06868**.

URL http://arxiv.org/abs/1504.06868 85

Cormack, G. V., Grossman, M. R., Hedin, B. and Oard, D. W. (2010). Overview of the TREC 2010 legal track. In *TREC 2010 Notebook*. TREC 2010, TREC, USA. 80

Cortes, C. and Vapnik, V. (1995). Support-vector networks. *Machine learning* **20** 273–297. 17

Dempster, A. P., Laird, N. M. and Rubin, D. B. (1977). Maximum likelihood from incomplete data via the EM algorithm (C/R: p22–37). *Journal of the Royal Statistical Society,* Series B **39** 1–22. 111

Diaconis, P., Khare, K. and Saloff-Coste, L. (2008). Gibbs sampling, exponential families and orthogonal polynomials (with discussion). *Statistical Science* **23** 151–178. 42

Dumais, S., Furnas, G., Landauer, T., Deerwester, S., Deerwester, S. et al. (1995). Latent semantic indexing. In *Proceedings of the Text Retrieval Conference*. 15, 70

EDRM (2009). The Electronic Discovery Reference Model. Online.

URL http://www.edrm.net 10, 70, 71

Ferguson, T. S. (1973). A Bayesian analysis of some nonparametric problems. *The Annals of Statistics* **1** 209–230. 103

Ferguson, T. S. (1974). Prior distributions on spaces of probability measures. *The Annals of Statistics* **2** 615–629. 103

Flegal, J. M. and Hughes, J. (2012). *mcmcse: Monte Carlo Standard Errors for MCMC*. Riverside, CA and Minneapolis, MN. R package version 1.0-1.

URL https://cran.r-project.org/web/packages/mcmcse/index.html 36, 64

Fuentes, C., Gopal, V., Casella, G., George, C. P., Glenn, T. C., Wilson, J. N. and Gader, P. D. (2011). Product partition models for Dirichlet allocation. Tech. Rep. 519, University of Florida. Department of Computer and Information Science and Engineering. 41

George, C. P., Wang, D. Z., Wilson, J. N., Epstein, L. M., Garland, P. and Suh, A. (2012). A machine learning based topic exploration and categorization on surveys. In *Machine Learning and Applications (ICMLA), 2012 11th International Conference on*, vol. 2. IEEE. 15, 79

George, E. I. and Foster, D. P. (2000). Calibration and empirical Bayes variable selection. *Biometrika* **87** 731–747. 23

Geyer, C. J. (2011). Importance sampling, simulated tempering, and umbrella sampling. In *Handbook of Markov Chain Monte Carlo* (S. P. Brooks, A. E. Gelman, G. L. Jones and X. L. Meng, eds.). Chapman & Hall/CRC, Boca Raton, 295–311. 29, 64

Geyer, C. J. and Thompson, E. A. (1995). Annealing Markov chain Monte Carlo with applications to ancestral inference. *Journal of the American Statistical Association* **90** 909–920. 29, 33

Griffiths, T. L. and Steyvers, M. (2004). Finding scientific topics. *Proceedings of the National Academy of Sciences* **101** 5228–5235. 16, 27, 41, 42, 48, 97, 98

Halko, N., Martinsson, P.-G. and Tropp, J. A. (2011). Finding structure with randomness: Probabilistic algorithms for constructing approximate matrix decompositions. *SIAM review* **53** 217–288. 76

Hastings, W. K. (1970). Monte Carlo sampling methods using Markov chains and their applications. *Biometrika* **57** 97–109. 28

Hobert, J. P. and Casella, G. (1996). The effect of improper priors on Gibbs sampling in hierarchical linear mixed models. *Journal of the American Statistical Association* **91** 1461–1473. 23

Hoenkamp, E. (2011). Trading spaces: On the lore and limitations of latent semantic analysis. In *Advances in Information Retrieval Theory* (G. Amati and F. Crestani, eds.), vol. 6931 of *Lecture Notes in Computer Science*. Springer Berlin Heidelberg, 40–51. 15

Hoffman, M., Bach, F. R. and Blei, D. M. (2010). Online learning for latent dirichlet allocation. In *advances in neural information processing systems*. 76

Ibragimov, I. A. and Linnik, Y. V. (1971). *Independent and Stationary Sequences of Random Variables*. Wolters-Noordhoff, Groningen. 26

Israel, G. D. (1992). *Determining Sample Size*. University of Florida Cooperative Extension Service, Institute of Food and Agriculture Sciences, EDIS. 70, 74

Joachims, T. (1998). Text categorization with suport vector machines: Learning with many relevant features. In *Proceedings of the 10th European Conference on Machine Learning*. ECML '98, Springer-Verlag, London, UK, UK.

URL http://dl.acm.org/citation.cfm?id=645326.649721 79

Joachims, T. (1999). Making Large-scale Support Vector Machine Learning Practical. In *Advances in Kernel Methods* (B. Schölkopf, C. J. C. Burges and A. J. Smola, eds.). MIT Press, Cambridge, MA, USA, 169–184.

URL http://dl.acm.org/citation.cfm?id=299094.299104 76

Jones, K. S. (1972). A statistical interpretation of term specificity and its application in retrieval. *Journal of Documentation* **28** 11–21. 14

kCura (2013). Workflow for Computer-Assisted Review in relativity. In *EDRM: White Paper Series*. EDRM, –. 67, 70

Lau, J. H., Newman, D. and Baldwin, T. (2014). Machine reading tea leaves: Automatically evaluating topic coherence and topic model quality. In *Proceedings of the European Chapter of the Association for Computational Linguistics*. 102

Lewis, D. D. (2011). Machine learning for discovery in legal cases.

URL http://www.youtube.com/watch?v=k-bocleGfok 67

Losey, R. (2012). Random sample calculations and my prediction that 300,000 lawyers will be using random sampling by 2022. Online.

URL http://e-discoveryteam.com 74

Losey, R. (2013). Predictive coding and proportionality: A marriage made in heaven. In *Regent University Law Review*, vol. 26. Regent University Law, 1–70. 69

Lucene, A. (2013). The Lucene search engine.

URL http://lucene.apache.org 68, 71, 76

Manning, C. D., Raghavan, P. and Schütze, H. (2008). *Introduction to Information Retrieval*. Cambridge University Press, New York, NY, USA. 112

Marinari, E. and Parisi, G. (1992). Simulated tempering: A new Monte Carlo scheme. *Europhysics Letters* **19** 451–458. 29, 33

Miller, G. A., Beckwith, R., Fellbaum, C., Gross, D. and Miller, K. (1990). Wordnet: An on-line lexical database. *International Journal of Lexicography* **3** 235–244. 69

Mimno, D., Wallach, H. M., Talley, E., Leenders, M. and McCallum, A. (2011). Optimizing semantic coherence in topic models. In *Proceedings of the Conference on Empirical Methods in Natural Language Processing*. Association for Computational Linguistics. 101, 102

Neal, R. (2008). The harmonic mean of the likelihood: Worst monte carlo method ever. Online.

URL http://radfordneal.wordpress.com 97, 98, 99

Neal, R. M. and Hinton, G. E. (1998). A view of the EM algorithm that justifies incremental, sparse, and other variants. In *Learning in Graphical Models* (M. I. Jordan, ed.), vol. 89 of *NATO ASI Series*. MIT Press, Cambridge, MA, USA, 355–368. 111

Newton, M. A. and Raftery, A. E. (1994). Approximate bayesian inference with the weighted likelihood bootstrap. *Journal of the Royal Statistical Society. Series B (Methodological)* 3–48. 98

Pedregosa, F., Varoquaux, G., Gramfort, A., Michel, V., Thirion, B., Grisel, O., Blondel, M., Prettenhofer, P., Weiss, R., Dubourg, V., Vanderplas, J., Passos, A., Cournapeau, D., Brucher, M., Perrot, M. and Duchesnay, E. (2011). Scikit-learn: Machine learning in Python. *Journal of Machine Learning Research* **12** 2825–2830. 84, 85

Řehůřek, R. and Sojka, P. (2010). Software framework for topic modelling with large corpora. In *Proceedings of the LREC 2010 Workshop on New Challenges for NLP Frameworks*. ELRA, Valletta, Malta. 48, 76

Robert, C. P. (2001). *The Bayesian Choice: From Decision-Theoretic Foundations to Computational Implementation*. Springer-Verlag, New York. 23

Salton, G., Wong, A. and Yang, C. S. (1975). A vector space model for automatic indexing. *Commun. ACM* **18** 613–620.

URL http://doi.acm.org/10.1145/361219.361220 14

Sethuraman, J. (1994). A constructive definition of Dirichlet priors. *Statistica Sinica* **4** 639–650. 103

Swets, J. A. (1996). *Signal Detection Theory and ROC Analysis in Psychology and Diagnostics: Collected Papers.* Lawrence Erlbaum Associates, Inc. 112

Taddy, M. A. (2011). On estimation and selection for topic models. *arXiv preprint arXiv:1109.4518* . 104

Teh, Y. W., Jordan, M. I., Beal, M. J. and Blei, D. M. (2006). Hierarchical Dirichlet processes. *Journal of the American Statistical Association* **101** 1566–1581. 103

Tomlinson, S. (2010). Learning task experiments in the TREC 2010 legal track. In *TREC*. 82

Vapnik, V. (1995). *The Nature of Statistical Learning Theory*. springer. 79

Wallach, H. M., Mimno, D. and McCallum, A. (2009a). Rethinking LDA: Why priors matter. *Advances in Neural Information Processing Systems* **22** 1973–1981. 20

Wallach, H. M., Murray, I., Salakhutdinov, R. and Mimno, D. (2009b). Evaluation methods for topic models. In *Proceedings of the 26th Annual International Conference on Machine Learning*. ACM. 98, 99, 100, 101

Wei, X. and Croft, W. B. (2006). Lda-based document models for ad-hoc retrieval. In *Proceedings of the 29th annual international ACM SIGIR conference on Research and development in information retrieval*. ACM. 101

BIOGRAPHICAL SKETCH

Clint received the Bachelor of Technology in computer science and engineering from the Department of Computer Science and Engineering, the University of Kerala, India, in 2004. After graduation, he worked over four years in industry for the software companies ENVESTNET and TATA Consultancy Services, as a software engineer. Clint joined the Department of Computer and Information Science and Engineering at the University of Florida, Gainesville, Florida, USA as a master's student in 2008. He received the Master of Science in computer engineering in 2010. He continued his research in the area of empirical Bayes methods, Markov chain Monte Carlo methods, electronic discovery, and large scale machine learning algorithms as a doctorate student. Clint graduated from the University of Florida with a Doctor of Philosophy in computer engineering in 2015.