Learning Author Topic Models from Text Corpora*

Michal Rosen-Zvi
School of Computer Science and Engineering
The Hebrew University of Jerusalem
91904 Jerusalem, Israel

Thomas Griffiths
Department of Cognitive and Linguistic Sciences
Brown University
Providence, RI 02912, USA

Padhraic Smyth
Department of Computer Science
University of California, Irvine
Irvine, CA 92697-3425, USA

Mark Steyvers
Department of Cognitive Sciences
University of California, Irvine
Irvine, CA 92697-5100, USA

November 4, 2005

Abstract

We propose a new unsupervised learning technique for extracting information about authors and topics from large text collections. We model documents as if they were generated by a two-stage stochastic process. An author is represented by a probability distribution over topics, and each topic is represented as a probability distribution over words. The probability distribution over topics in a multi-author paper is a mixture of the distributions associated with the authors. The topic-word and author-topic distributions are learned from data in an unsupervised manner using a Markov chain Monte Carlo algorithm. We apply the methodology to three large text corpora: 150,000 abstracts from the CiteSeer digital library, 1,740 papers from the Neural Information Processing Systems Conference (NIPS), and 121,000 emails from a large corporation. We discuss in detail the interpretation of the results discovered by the system including specific topic and author models, ranking of authors by topic and topics by author, parsing of abstracts by topics and authors, and detection of unusual papers by specific authors. Experiments based on perplexity scores for test documents are used to illustrate systematic differences between the proposed author topic model and a number of alternatives. Extensions to the model, allowing (for example) generalizations of the notion of an author, are also briefly discussed.

Keywords: topic models, Gibbs sampling, unsupervised learning, author models, perplexity.

1 Introduction

With the advent of the Web and specialized digital text collections, automated extraction of useful information from text has become an increasingly important research area in information retrieval,

*The material in this paper was presented in part at the 2004 Uncertainty in AI Conference and the 2004 ACM SIGKDD Conference.

statistical natural language processing, and machine learning. Applications include document annotation, database organization, query answering, and automated summarization of text collections. Statistical approaches based upon generative models have proven effective in addressing these problems, providing efficient methods for extracting structured representations from large document collections.

In this paper we describe a generative model for document collections, the author topic (AT) model, that simultaneously models the content of documents and the interests of authors. This generative model represents each document as a mixture of probabilistic topics, in a manner similar to Latent Dirichlet Allocation [Blei et al., 2003]. It extends previous work using probabilistic topics to author modeling by allowing the mixture weights for different topics to be determined by the authors of the document. By learning the parameters of the model, we obtain the set of topics that appear in a corpus and their relevance to different documents, and identify which topics are used by which authors. Figure 1 shows an example of several such topics (with associated authors and words) learned by the algorithm from a collection of papers from the NIPS conference (these will be discussed in more detail later in the paper). Both the words and the authors associated with each topic are quite focused and reflect a variety of different and quite specific research areas associated with the NIPS conference. The model used in Figure 1 also produces a topic distribution for each author—Figure 2 shows the likely topics for a set of well-known NIPS authors from this model. By modeling the interests of authors, we can answer a range of important queries about the content of document collections, including (for example) which subjects an author writes about, which authors are likely to have written documents similar to an observed document, and which authors produce similar work.

The generative model at the heart of our approach is based upon the idea that a document can be represented as a mixture of topics. This idea has motivated several different approaches in machine learning and statistical natural language processing [Hofmann, 1999, Blei et al., 2003, Minka and Lafferty, 2002, Griffiths and Steyvers, 2004, Buntine and Jakulin, 2004]. Topic models have three major advantages over other approaches to document modeling: the topics are extracted in a completely unsupervised fashion, requiring no document labels and no special initialization; each topic is individually interpretable, providing a representation that can be understood by the user; and each document can express multiple topics, capturing the topic combinations that arise in text documents.

Supervised learning techniques for automated categorization of documents into known classes or topics have received considerable attention in recent years [e.g., Yang, 1999]. However, unsupervised methods are often necessary for addressing the challenges of modeling large document collections. For many document collections, neither predefined topics nor labeled documents may be available. Furthermore, there is considerable motivation to uncover hidden topic structure in large corpora, particularly in rapidly changing fields such as computer science and biology, where predefined topic categories may not reflect dynamically evolving content.

Topic models provide an unsupervised method for extracting an interpretable representation from a collection of documents. Prior work on automatic extraction of representations from text has used a number of different approaches. One general approach, in the context of the general "bag of words" framework, is to represent high-dimensional term vectors in a lower-dimensional space. Local regions in the lower-dimensional space can then be associated with specific topics. For example, the WEBSOM system [Lagus et al., 1999] uses non-linear dimensionality reduction via self-organizing maps to represent term vectors in a two-dimensional layout. Linear projection techniques, such as latent semantic indexing (LSI), are also widely used (e.g., Berry et al. [1994]). Deerwester et al. [1990], while not using the term "topics" per se, state:

TOPIC 4	
WORD	**PROB.**
LIGHT	.0306
RESPONSE	.0282
INTENSITY	.0252
RETINA	.0241
OPTICAL	.0233
KOCH	.0190
BACKGROUND	.0162
CONTRAST	.0145
CENTER	.0124
FEEDBACK	.0118
AUTHOR	**PROB.**
Koch_C	.0903
Boahen_K	.0320
Skrzypek_J	.0283
Liu_S	.0250
Delbruck_T	.0232
Etienne-C._R	.0210
Bair_W	.0178
Bialek_W	.0133
Yasui_S	.0106
Hsu_K	.0103

TOPIC 13	
WORD	**PROB.**
RECOGNITION	.0500
CHARACTER	.0334
TANGENT	.0246
CHARACTERS	.0232
DISTANCE	.0197
HANDWRITTEN	.0166
DIGITS	.0154
SEGMENTATION	.0142
DIGIT	.0124
IMAGE	.0111
AUTHOR	**PROB.**
Simard_P	.0602
Martin_G	.0340
LeCun_Y	.0339
Henderson_D	.0289
Denker_J	.0245
Revow_M	.0206
Rashid_M	.0205
Rumelhart_D	.0185
Sackinger_E	.0181
Flann_N	.0142

TOPIC 28	
WORD	**PROB.**
KERNEL	.0547
VECTOR	.0293
SUPPORT	.0293
MARGIN	.0239
SVM	.0196
DATA	.0165
SPACE	.0161
KERNELS	.0160
SET	.0146
MACHINES	.0132
AUTHOR	**PROB.**
Scholkopf_B	.0774
Smola_A	.0685
Vapnik_V	.0487
Burges_C	.0411
Ratsch_G	.0296
Mason_L	.0232
Platt_J	.0225
Cristianini_N	.0179
Laskov_P	.0160
Chapelle_O	.0152

TOPIC 9	
WORD	**PROB.**
SOURCE	.0389
INDEPENDENT	.0376
SOURCES	.0344
SEPARATION	.0322
INFORMATION	.0319
ICA	.0276
BLIND	.0227
COMPONENT	.0226
SEJNOWSKI	.0224
NATURAL	.0183
AUTHOR	**PROB.**
Sejnowski_T	.0627
Bell_A	.0378
Yang_H	.0349
Lee_T	.0348
Attias_H	.0290
Parra_L	.0271
Cichocki_A	.0262
Hyvarinen_A	.0242
Amari_S	.0160
Oja_E	.0143

TOPIC 82	
WORD	**PROB.**
BAYESIAN	.0437
POSTERIOR	.0377
PRIOR	.0333
PARAMETERS	.0228
GAUSSIAN	.0183
DATA	.0183
EVIDENCE	.0144
LIKELIHOOD	.0142
MACKAY	.0127
COVARIANCE	.0126
AUTHOR	**PROB.**
Williams_C	.0854
Bishop_C	.0504
Barber_D	.0370
Rasmussen_C	.0351
MacKay_D	.0281
Tipping_M	.0225
Opper_M	.0191
Sollich_P	.0160
Sykacek_P	.0153
Wolpert_D	.0141

TOPIC 7	
WORD	**PROB.**
STATE	.0715
POLICY	.0367
ACTION	.0301
REINFORCEMENT	.0283
STATES	.0244
FUNCTION	.0190
ACTIONS	.0179
OPTIMAL	.0155
REWARD	.0154
AGENT	.0129
AUTHOR	**PROB.**
Singh_S	.1293
Barto_A	.0554
Sutton_R	.0482
Parr_R	.0385
Hansen_E	.0300
Dayan_P	.0249
Thrun_S	.0223
Tsitsiklis_J	.0222
Dietterich_T	.0206
Loch_J	.0163

TOPIC 62	
WORD	**PROB.**
METHOD	.0497
METHODS	.0349
RESULTS	.0314
APPROACH	.0270
BASED	.0239
TECHNIQUES	.0182
APPLIED	.0167
SINGLE	.0158
NUMBER	.0149
PROBLEMS	.0135
AUTHOR	**PROB.**
Sejnowski_T	.0121
Baluja_S	.0101
Thrun_S	.0097
Moody_J	.0095
Hinton_G	.0084
Moore_A	.0081
Barto_A	.0079
Bengio_Y	.0079
Singh_S	.0078
Dietterich_T	.0068

TOPIC 16	
WORD	**PROB.**
HINTON	.0243
SET	.0131
WEIGHTS	.0126
COST	.0118
SPACE	.0106
UNSUPERVISED	.0102
PROCEDURE	.0100
SINGLE	.0097
ENERGY	.0092
VISIBLE	.0088
AUTHOR	**PROB.**
Hinton_G	.1959
Mozer_M	.0915
Zemel_R	.0771
Becker_S	.0285
Dayan_P	.0200
Seung_H	.0169
Sejnowski_T	.0127
Ghahramani_Z	.0113
Nowlan_S	.0102
Schraudolph_N	.0098

Figure 1: 8 examples of topics (out of 100 topics in total) from a model fit to NIPS papers from 1987 to 1999—shown are the 10 most likely words and 10 most likely authors per topic.

AUTHOR = Jordan_M		
PROB.	TOPIC	WORDS
.1389	37	MIXTURE, EM, LIKELIHOOD, EXPERTS, MIXTURES, EXPERT, GATING, PARAMETERS, LOG, JORDAN
.1221	60	BELIEF, FIELD, STATE, APPROXIMATION, MODELS, VARIABLES, FACTOR, JORDAN, NETWORKS, PARAMETERS
.0598	52	ALGORITHM, ALGORITHMS, PROBLEM, STEP, PROBLEMS, LINEAR, UPDATE, FIND, LINE, ITERATIONS
.0449	77	MOTOR, TRAJECTORY, ARM, INVERSE, HAND, CONTROL, MOVEMENT, JOINT, DYNAMICS, FORWARD

AUTHOR = Koch_C		
PROB.	TOPIC	WORDS
.2518	4	LIGHT, RESPONSE, INTENSITY, RETINA, OPTICAL, KOCH, BACKGROUND, CONTRAST, CENTER, FEEDBACK
.0992	45	VISUAL, STIMULUS, CORTEX, SPATIAL, ORIENTATION, RESPONSE, CORTICAL, RECEPTIVE, TUNING, STIMULI
.0882	84	SPIKE, FIRING, SYNAPTIC, SYNAPSES, MEMBRANE, POTENTIAL, CURRENT, SPIKES, RATE, SYNAPSE
.0504	64	CIRCUIT, CURRENT, VOLTAGE, ANALOG, CHIP, VLSI, CIRCUITS, SILICON, PULSE, MEAD

AUTHOR = LeCun_Y		
PROB.	TOPIC	WORDS
.2298	13	RECOGNITION, CHARACTER, TANGENT, CHARACTERS, DISTANCE, HANDWRITTEN, DIGITS, SEGMENTATION, DIGIT, IMAGE
.0930	53	GRADIENT, FUNCTION, DESCENT, ERROR, VECTOR, DERIVATIVE, DERIVATIVES, OPTIMIZATION, PARAMETERS, LOCAL
.0930	69	LAYER, WEIGHTS, PROPAGATION, BACK, OUTPUT, LAYERS, INPUT, NUMBER, WEIGHT, FORWARD
.0762	36	INPUT, OUTPUT, INPUTS, OUTPUTS, VALUES, ARCHITECTURE, SUM, ADAPTIVE, PREVIOUS, PROCESSING

AUTHOR = Sejnowski_T		
PROB.	TOPIC	WORDS
.0927	9	SOURCE, INDEPENDENT, SOURCES, SEPARATION, INFORMATION, ICA, BLIND, COMPONENT, SEJNOWSKI, NATURAL
.0852	45	VISUAL, STIMULUS, CORTEX, SPATIAL, ORIENTATION, RESPONSE, CORTICAL, RECEPTIVE, TUNING, STIMULI
.0495	36	INPUT, OUTPUT, INPUTS, OUTPUTS, VALUES, ARCHITECTURE, SUM, ADAPTIVE, PREVIOUS, PROCESSING
.0439	74	MOTION, FIELD, DIRECTION, RECEPTIVE, FIELDS, VELOCITY, MOVING, FLOW, DIRECTIONS, ORDER

AUTHOR = Vapnik_V		
PROB.	TOPIC	WORDS
.3374	28	KERNEL, VECTOR, SUPPORT, MARGIN, SVM, DATA, SPACE, KERNELS, SET, MACHINES
.1243	44	LOSS, ESTIMATION, METHOD, ESTIMATE, PARAMETER, INFORMATION, ENTROPY, BASED, LOG, NEURAL
.0943	72	BOUND, BOUNDS, THEOREM, EXAMPLES, DIMENSION, FUNCTIONS, CLASS, PROBABILITY, NUMBER, RESULTS
.0669	92	ERROR, TRAINING, GENERALIZATION, EXAMPLES, SET, ENSEMBLE, TEST, FUNCTION, LINEAR, ERRORS

Figure 2: Selected authors from the NIPS corpus, and four high-probability topics for each author from the author topic model. Topics unrelated to technical content (such as topics containing words such as *results, methods, experiments*, etc.) were excluded.

> In various problems, we have approximated the original term-document matrix using 50-100 orthogonal factors or derived dimensions. Roughly speaking, these factors may be thought of as artificial concepts; they represent extracted common meaning components of many different words and documents.

A well-known drawback of the LSI approach is that the resulting representation is often hard to interpret. The derived dimensions indicate axes of a space, but there is no guarantee that such dimensions will make sense to the user of the method. Another limitation of LSI is that it implicitly assumes a Gaussian (squared-error) noise model for the word-count data, which can lead to implausible results such as predictions of negative counts, although more recent work has generalized these LSI approaches by projecting word counts to a continuous latent space [Globerson and Tishby, 2003, Welling et al., 2005].

A different approach to unsupervised topic extraction relies on clustering documents into groups containing (presumably) similar semantic content. A variety of well-known document clustering techniques have been used for this purpose [e.g., Cutting et al., 1992, McCallum et al., 2000, Popescul et al., 2000, Dhillon and Modha, 2001]. Each cluster of documents can then be associated with a latent topic as represented (for example) by the mean term vector for documents in the cluster. While clustering can provide useful broad information about topics, clusters are inherently limited by the fact that each document is (typically) only associated with one cluster. This is often at odds with the multi-topic nature of text documents in many contexts—combinations of diverse topics within a single document are difficult to represent. For example, the present paper contains at least two significantly different topics: document modeling and Bayesian estimation. For this reason, other representations that allow documents to be composed of multiple topics generally provide better models for sets of documents [e.g., better out of sample predictions, Blei et al., 2003].

There are several generative models for document collections that model individual documents as mixtures of topics. Hofmann [1999] introduced the aspect model (also referred to as probabilistic LSI, or pLSI) as a probabilistic alternative to projection and clustering methods. In pLSI, topics are modeled as multinomial probability distributions over words, and documents are assumed to be generated by the activation of multiple topics. While the pLSI model produced impressive results on a number of text document problems such as information retrieval, the parameterization of the model was susceptible to overfitting and did not provide a straightforward way to make inferences about documents not seen in the training data. Blei et al. [2003] addressed these limitations by proposing a more general Bayesian probabilistic topic model called latent Dirichlet allocation (LDA). The parameters of the LDA model (the topic-word and document-topic distributions) are estimated using an approximation technique known as variational EM, since standard estimation methods are intractable. Griffiths and Steyvers [2004] further showed how Gibbs sampling, a Markov chain Monte Carlo technique, could be applied to the problem of parameter estimation for this model with relatively large data sets. Other approximate inference methods have been explored by Minka and Lafferty [2002] and Buntine and Jakulin [2004] in document modeling and Pritchard et al. [2000] in genetics.

More recent research on topic models in information retrieval has focused on including additional sources of information to constrain the learned topics. For example, Cohn and Hofmann [2001] proposed an extension of pLSI to model both the document content as well as citations or hyperlinks between documents. Similarly, Erosheva et al. [2004] extended the LDA model to model both text and citations and applied their model to scientific papers from the Proceedings of the National Academy of Sciences.

Our aim here is to extend the probabilistic topic models to include authorship information.

Joint author-topic modeling has received little or no attention as far as we are aware. The areas of stylometry, authorship attribution, and forensic linguistics focus on the related but different problem of identifying which author (among a set of possible authors) wrote a particular piece of text [Holmes, 1998]. For example, Mosteller and Wallace [1964] used Bayesian techniques to infer whether Hamilton or Madison was the more likely author of disputed Federalist papers. More recent work of a similar nature includes authorship analysis of a purported poem by Shakespeare [Thisted and Efron, 1987], identifying authors of software programs [Gray et al., 1997], and the use of techniques such as neural networks [Kjell, 1994] and support vector machines [Diederich et al., 2003] for author identification.

These author identification methods emphasize the use of distinctive stylistic features (such as sentence length) that characterize a specific author. In contrast, the models we present here focus on extracting the general semantic content of a document, rather than the stylistic details of how it was written. For example, in our model we omit common "stop" words since they are generally irrelevant to the topic of the document—however, the distributions of stop words can be quite useful in stylometry. While topic information could be usefully combined with stylistic features for author classification we do not pursue this idea in this particular paper.

Graph-based and network-based models are also frequently used as a basis for representation and analysis of relations among scientific authors. For example, McCain [1990], Newman [2001], Mutschke [2003] and Erten et al. [2003] use a variety of methods from bibliometrics, social networks, and graph theory to analyze and visualize co-author and citation relations in the scientific literature. Kautz et al. [1997] developed the interactive ReferralWeb system for exploring networks of computer scientists working in artificial intelligence and information retrieval, and White and Smyth [2003] used PageRank-style ranking algorithms to analyze co-author graphs. In all of this work only the network connectivity information is used—the text information from the underlying documents is not used in modeling. Thus, while the grouping of authors via these network models can implicitly provide indications of latent topics, there is no explicit representation of the topics in terms of the content (the words) of the documents.

The novelty of the work described in this paper lies in the proposal of a probabilistic model that represents both authors and topics. This approach goes beyond existing work on topic models by using a set of topics to simultaneously model both authors and documents, and goes beyond existing approaches to author modeling by making it possible to capture the semantic content of the contributions associated with a given author. As we will show later in the paper, the model provides a general framework for exploration, discovery, and query-answering in the context of the relationships of author and topics for large document collections.

The outline of the paper is as follows: Section 2 describes the author topic model and Section 3 outlines how the parameters of the model (the topic-word distributions and author-topic distributions) can be learned from training data consisting of documents with known authors. Section 4 discusses the application of the model to three different document collections: papers from the NIPS conference, abstracts from the CiteSeer collection, and emails from Enron. The section includes a general discussion of convergence and stability in learning, and examples of specific topics and specific author models that are learned by the algorithm. In Section 5 we describe illustrative applications of the model, including detecting unusual papers for selected authors and detecting which parts of a text were written by different authors. Section 6 compares and contrasts the proposed author topic model with a number of related models, including the LDA model, a simple author model (with no topics), and a model allowing "fictitious authors." Section 7 contains a brief discussion and concluding comments.

2 The Author Topic (AT) Model

In this section we introduce the author topic model. The author topic model belongs to a family of generative models for text where words are viewed as discrete random variables, a document contains a fixed number of words, and each word takes one value from a predefined vocabulary. We will use integers to denote the entries in the vocabulary, with each word w taking a value from $1, \ldots, W$ where W is the number of unique words in the vocabulary. A document d is represented as a vector of words, $\mathbf{w}_d$, with N_d entries. A corpus with D documents is represented as a concatenation of the document vectors, which we will denote $\mathbf{w}$, having $N = \sum_{d=1}^{D} N_d$ entries. In addition to these words, we have information about the authors of each document. We define $\mathbf{a}_d$ to be the set of authors of document d. $\mathbf{a}_d$ consists of elements that are integers from $1, \ldots, A$, where A is the number of authors who generated the documents in the corpus. A_d will be used to denote the number of authors of document d.

To illustrate this notation, consider a simple example. Say we have $D = 3$ documents in the corpus, written by $A = 2$ authors that use a vocabulary with $W = 1000$ unique words. The first author (author 1) wrote paper 1, author 2 wrote paper 2, and they co-authored paper 3. According to our notation, $\mathbf{a}_1 = (1)$, $\mathbf{a}_2 = (2)$ and $\mathbf{a}_3 = (1, 2)$, and $A_1 = 1$, $A_2 = 1$, and $A_3 = 2$. Say the first document contains a single line, *Machine learning has an abundance of interesting research problems*. We can remove stop words such as *has*, *an*, and *of*, to leave a document with 6 words. If *machine* is the 8th entry in the vocabulary, *learning* is the 12th, and *abundance* is the 115th, then $w_1 = 8$, $w_2 = 12$, $w_3 = 115$, and so on.

The author topic model is a hierarchical generative model in which each word w in a document is associated with two latent variables: an author, x and a topic, z. These latent variables augment the N-dimensional vector $\mathbf{w}$ (indicating the values of all words in the corpus) with two additional N-dimensional vectors $\mathbf{z}$ and $\mathbf{x}$, indicating topic and author assignments for the N words.

For the purposes of estimation, we assume that the set of authors of each document is observed. This leaves unresolved the issue of having unobserved authors, and avoids the need to define a prior on authors, which is outside of the scope of this paper. Each author is associated with a multinomial distribution over topics. Conditioned on the set of authors and their distributions over topics, the process by which a document is generated can be summarized as follows: first, an author is chosen uniformly at random for each word that will appear in the document; next, a topic is sampled for each word from the distribution over topics associated with the author of that word; finally, the words themselves are sampled from the distribution over words associated with each topic.

This generative process can be expressed more formally by defining some of the other variables in the model. Assume we have T topics. We can parameterize the multinomial distribution over topics for each author using a matrix Θ of size $T \times A$, with elements θ_{ta} that stand for the probability of assigning topic t to a word generated by author a. Thus $\sum_{t=1}^{T} \theta_{ta} = 1$, and for simplicity of notation we will drop the index t when convenient and use θ_a to stand for the ath column of the matrix. The multinomial distributions over words associated with each topic are parameterized by a matrix Φ of size $W \times T$, with elements ϕ_{wt} that stand for the probability of generating word w from topic t. Again, $\sum_{w=1}^{W} \phi_{wt} = 1$, and ϕ_t stands for the tth column of the matrix. These multinomial distributions are assumed to be generated from symmetric Dirichlet priors with hyperparameters α and β respectively. In the results in this paper we assume that these hyperparameters are fixed. Table 1 summarizes this notation.

The sequential procedure of first picking an author followed by picking a topic then generating a word according to the probability distributions above leads to the following generative process:

Table 1: Symbols associated with the author topic model, as used in this paper.

Authors of the corpus	$\mathcal{A}$	Set
Authors of the dth document	$\mathbf{a}_d$	A_d-dimensional vector
Number of authors of the dth document	A_d	Scalar
Number of words assigned to author and topic	C^{TA}	$T \times A$ matrix
Number of words assigned to topic and word	C^{WT}	$W \times T$ matrix
Set of authors and words in the training data	$\mathcal{D}^{\text{train}}$	Set
Number of authors	A	Scalar
Number of documents	D	Scalar
Number of words in the dth document	N_d	Scalar
Number of words in the corpus	N	Scalar
Number of topics	T	Scalar
Vocabulary Size	W	Scalar
Words in the dth document	$\mathbf{w}_d$	N_d-dimensional vector
Words in the corpus	$\mathbf{w}$	N-dimensional vector
ith word in the corpus	w_i	ith component
Author assignments	$\mathbf{x}$	N Dimensional vector
Author assignment for the ith word	x_i	ith Component
Topic assignments	$\mathbf{z}$	N Dimensional vector
Topic assignment for the ith word	z_i	ith Component
Dirichlet prior	α	Scalar
Dirichlet prior	β	Scalar
Probabilities of words given topics	Φ	$W \times T$ matrix
Probabilities of words given topic t	ϕ_t	W-dimensional vector
Probabilities of topics given authors	Θ	$T \times A$ matrix
Probabilities of topics given author a	θ_a	T-dimensional vector

1. For each author $a = 1, ..., A$ choose $\theta_a \sim \text{Dirichlet}(\alpha)$
   For each topic $t = 1, ..., T$ choose $\phi_t \sim \text{Dirichlet}(\beta)$
2. For each document $d = 1, ..., D$
   Given the vector of authors $\mathbf{a}_d$
   For each word w_i, indexed by $i = 1, ..N_d$
   Conditioned on $\mathbf{a}_d$ choose an author $x_i \sim \text{Uniform}(\mathbf{a}_d)$
   Conditioned on x_i choose a topic $z_i \sim \text{Discrete}(\theta_{x_i})$
   Conditioned on z_i choose a word $w_i \sim \text{Discrete}(\phi_{z_i})$

The graphical model corresponding to this process is shown in Figure 3. Note that by defining the model we fix the number of possible topics to T. In circumstances where the number of topics is not determined by the application, methods such as comparison of Bayes factors (e.g., Griffiths and Steyvers [2004]) or non-parametric Bayesian statistics (e.g., Teh et al. [2005]) can be used to infer T from a dataset. In this paper, we will deal with the case where T is fixed.

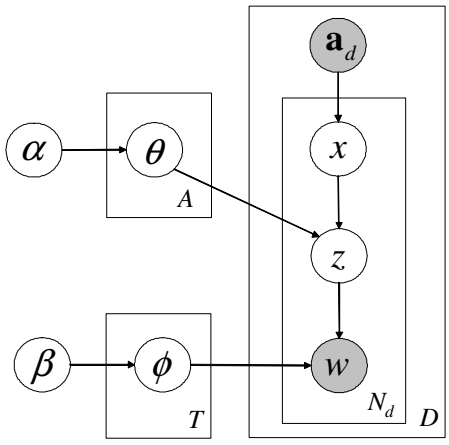

Figure 3: Graphical model for the author topic model.

Under this generative process, each topic is drawn independently when conditioned on Θ, and each word is drawn independently when conditioned on Φ and $\mathbf{z}$. The probability of the corpus $\mathbf{w}$, conditioned on Θ and Φ (and implicitly on a fixed number of topics T), is

$$P(\mathbf{w}|\Theta, \Phi, \mathcal{A}) = \prod_{d=1}^{D} P(\mathbf{w}_d|\Theta, \Phi, \mathbf{a}_d). \tag{1}$$

We can obtain the probability of the words in each document, $\mathbf{w}_d$ by summing over the latent variables x and z, to give

$$\begin{aligned} P(\mathbf{w}_d|\Theta, \Phi, \mathcal{A}) &= \prod_{i=1}^{N_d} P(w_i|\Theta, \Phi, \mathbf{a}_d) \\ &= \prod_{i=1}^{N_d} \sum_{a=1}^{A} \sum_{t=1}^{T} P(w_i, z_i = t, x_i = a|\Theta, \Phi, \mathbf{a}_d)) \\ &= \prod_{i=1}^{N_d} \sum_{a=1}^{A} \sum_{t=1}^{T} P(w_i|z_i = t, \phi_t) P(z_i = t|x_i = a, \theta_a) P(x_i = a|\mathbf{a}_d) \\ &= \prod_{i=1}^{N_d} \frac{1}{A_d} \sum_{a \in \mathbf{a}_d} \sum_{t=1}^{T} \phi_{w_i t} \theta_{ta}, \end{aligned} \tag{2}$$

where the factorization in the third line makes use of the independence assumptions of the model. The last line in the equations above expresses the probability of the words $\mathbf{w}$ in terms the entries of the parameter matrices Φ and Θ introduced earlier. The probability distribution over author assignments, $P(x_i = a|\mathbf{a}_d)$, is assumed to be uniform over the elements of $\mathbf{a}_d$, and deterministic if $A_d = 1$. The probability distribution over topic assignments, $P(z_i = t|x_i = a, \Theta)$ is the multinomial distribution θ_a in Θ that corresponds to author a, and the probability of word given a topic assignment, $P(w_i|z_i = t)$ is the multinomial distribution ϕ_t in Φ that corresponds to topic t.

Equations 1 and 2 can be used to compute the probability of a corpus $\mathbf{w}$ conditioned on Θ and Φ, i.e., the likelihood of a corpus. If Θ and Φ are treated as parameters of the model, this likelihood can be used in maximum-likelihood or maximum-a-posteriori estimation. Another strategy is to treat Θ and Φ as random variables, and compute the marginal probability of a corpus by integrating

them out. Under this strategy, the probability of $\mathbf{w}$ becomes

$$\begin{aligned} P(\mathbf{w}|\mathcal{A},\alpha,\beta) &= \int\int P(\mathbf{w}|\mathcal{A},\Theta,\Phi)p(\Theta,\Phi|\alpha,\beta)d\Theta d\Phi \\ &= \int\int \left[\prod_{d=1}^{D}\prod_{i=1}^{N_d}\frac{1}{A_d}\sum_{a\in\mathbf{a}_d}\sum_{t=1}^{T}\phi_{w_i t}\theta_{ta}\right] p(\Theta,\Phi|\alpha,\beta)d\Theta d\Phi, \end{aligned} \tag{3}$$

where $p(\Theta,\Phi|\alpha,\beta) = p(\Theta|\alpha)p(\Phi|\beta)$ are the Dirichlet priors on Θ and Φ defined earlier.

3 Learning the Author Topic Model from Data

The author topic model contains two continuous random variables, Θ and Φ. Various approximate inference methods have recently been employed for estimating the posterior distribution for continuous random variables in hierarchical Bayesian models. These approximate inference algorithms range from variational inference [Blei et al., 2003] and expectation propagation [Minka and Lafferty, 2002] to MCMC schemes [Pritchard et al., 2000, Griffiths and Steyvers, 2004, Buntine and Jakulin, 2004]. Inference in these models is hard: if Θ is treated as a random variable, the expectation step in an EM algorithm that learns the parameters, Φ, cannot be performed in a closed form. The inference scheme used in this paper is based upon a Markov chain Monte Carlo (MCMC) algorithm. While MCMC is not as computationally efficient as approximation schemes such as variational inference and expectation propagation, it is unbiased and has been successfully used in several recent large scale applications of topic models [Buntine and Jakulin, 2004, Griffiths and Steyvers, 2004].

Our aim is to estimate the posterior distribution, $p(\Theta,\Phi|\mathcal{D}^{\text{train}},\alpha,\beta)$. Samples from this distribution can be useful in many applications, as illustrated in Section 4.3. This is also the distribution used for evaluating the predictive power of the model, (e.g., see Section 6.4) and for deriving other quantities, such as the most surprising paper for an author (Section 5).

Our inference scheme is based upon the observation that

$$p(\Theta,\Phi|\mathcal{D}^{\text{train}},\alpha,\beta) = \sum_{\mathbf{z},\mathbf{x}} p(\Theta,\Phi|\mathbf{z},\mathbf{x},\mathcal{D}^{\text{train}},\alpha,\beta)P(\mathbf{z},\mathbf{x}|\mathcal{D}^{\text{train}},\alpha,\beta).$$

We obtain an approximate posterior on Θ and Φ by using a Gibbs sampler to compute the sum over $\mathbf{z}$ and $\mathbf{x}$. This process involves two steps. First, we obtain an empirical sample-based estimate of $P(\mathbf{z},\mathbf{x}|\mathcal{D}^{\text{train}},\alpha,\beta)$ using Gibbs sampling. Second, for any specific sample corresponding to a particular $\mathbf{x}$ and $\mathbf{z}$, $p(\Theta,\Phi|\mathbf{z},\mathbf{x},\mathcal{D}^{\text{train}},\alpha,\beta)$ can be computed directly by exploiting the fact that the Dirichlet distribution is conjugate to the multinomial. In the next two sections we will explain each of these two steps in turn.

3.1 Gibbs Sampling

Gibbs sampling is a form of Markov chain Monte Carlo, in which a Markov chain is constructed to have a particular stationary distribution [e.g., Gilks et al., 1996]. In our case, we wish to construct a Markov chain which converges to the posterior distribution over $\mathbf{x}$ and $\mathbf{z}$ conditioned on $\mathcal{D}^{\text{train}}$, α, and β. Using Gibbs sampling we can generate a sample from the joint distribution $P(\mathbf{z},\mathbf{x}|\mathcal{D}^{\text{train}},\alpha,\beta)$ by (a) sampling an author assignment x_i and a topic assignment z_i for an individual word w_i, conditioned on fixed assignments of authors and topics for all other words in the corpus, and (b) repeating this process for each word. A single Gibbs sampling iteration consists

of sequentially performing this sampling of author and topic assignments for each individual word in the corpus.

In Appendix A we show how to derive the following basic equation needed for the Gibbs sampler:

$$P\left(x_i = a, z_i = t | w_i = w, \mathbf{z}_{-i}, \mathbf{x}_{-i}, \mathbf{w}_{-i}, \mathcal{A}, \alpha, \beta\right) \propto \frac{C^{WT}_{wt} + \beta}{\sum_{w'} C^{WT}_{w't} + W\beta} \frac{C^{TA}_{ta} + \alpha}{\sum_{t'} C^{TA}_{t'a} + T\alpha} \tag{4}$$

for $a \in \mathbf{a}_d$. C^{TA} represents the topic-author count matrix, where C^{TA}_{ta} is the number of words assigned to topic t for author a. Similarly C^{WT} is the word-topic count matrix, where C^{WT}_{wt} is the number of words from the wth entry in the vocabulary assigned to topic t. The other symbols are summarized in Table 1. This equation can be manipulated further to obtain the conditional probability of the topic of the ith word given the rest, $P(z_i = t|\mathbf{z}_{-i}, \mathbf{x}, \mathcal{D}^{\text{train}}\alpha, \beta)$, and for the conditional probability of the author of the ith word given the rest, $P(x_i = a|\mathbf{z}, \mathbf{x}_{-i}, \mathcal{D}^{\text{train}}\alpha, \beta)$. In the results in this paper, however, we use a blocked sampler where we sample x_i and z_i jointly, as this improves convergence of the Gibbs sampler when the variables are highly dependent.

The algorithms for Gibbs sampling works as follows. We initialize the author and topic assignments, $\mathbf{x}$ and $\mathbf{z}$, randomly. In each Gibbs sampling iteration we sequentially draw the topic and author assignment of the ith word from the joint conditional distribution in Equation 4 above. After a predefined number of iterations (the so-called burn-in time of the Gibbs sampler) we begin recording samples $\mathbf{x}^s$, $\mathbf{z}^s$. The burn-in is intended to allow the sampler to approach its stationary distribution—the posterior distribution $P(\mathbf{z}, \mathbf{x}|\mathcal{D}^{\text{train}}, \alpha, \beta)$. For these samples to be equivalent to independent samples from the posterior we either have to use one chain and to have the number of iterations between two different samples be on the order of the mixing time of the chain (a quantity that is hard to evaluate), or accumulate samples from multiple chains, each starting with different initial conditions. While the samples are generally not independent, expectations of functions across these samples will converge to the same value as the expectation of those functions across the true posterior. In Section 4.1 we discuss convergence issues in more detail.

3.2 The posterior on Θ and Φ

Given $\mathbf{z}$, $\mathbf{x}$, $\mathcal{D}^{\text{train}}$, α, and β, computing posterior distributions on Θ and Φ is straightforward. Using the fact that the Dirichlet is conjugate to the multinomial, we have

$$\phi_t|\mathbf{z}, \mathcal{D}^{\text{train}}, \beta \sim \text{Dirichlet}(C^{WT}_{\cdot t} + \beta) \tag{5}$$

$$\theta_a|\mathbf{x}, \mathbf{z}, \mathcal{D}^{\text{train}}, \alpha \sim \text{Dirichlet}(C^{TA}_{\cdot a} + \alpha) \tag{6}$$

where $C^{WT}_{\cdot t}$ is the vector of counts of the number of times each word has been assigned to topic t. Evaluating the posterior mean of Φ and Θ given $\mathbf{x}$, $\mathbf{z}$, $\mathcal{D}^{\text{train}}$, α, and β is straightforward. From Equations 5 and 6, it follows that

$$E[\phi_{wt}|\mathbf{z}^s, \mathcal{D}^{\text{train}}, \beta] = \frac{(C^{WT}_{wt})^s + \beta}{\sum_{w'}(C^{WT}_{w't})^s + W\beta} \tag{7}$$

$$E[\theta_{ta}|\mathbf{x}^s, \mathbf{z}^s, \mathcal{D}^{\text{train}}, \alpha] = \frac{(C^{TA}_{ta})^s + \alpha}{\sum_{t'}(C^{TA}_{t'a})^s + T\alpha}. \tag{8}$$

where $(C^{WT})^s$ is the matrix of topic-word counts exhibited in $\mathbf{z}^s$. These posterior means also provide point estimates for Φ and Θ, and correspond to the posterior predictive distribution for the next word from a topic and the next topic in a document respectively.

In many applications, we wish to evaluate the expectation of some function of Φ and Θ, such as the posterior probability of a document, $P(\mathbf{w}_d|\Theta, \Phi, \mathbf{a}_d)$, given $\mathcal{D}^{\text{train}}$, α, and β. Denoting such a function $f(\Phi, \Theta)$, we can use the results above to define a general strategy for evaluating such expectations. We wish to compute

$$E[f(\Phi,\Theta)|\mathcal{D}^{\text{train}},\alpha,\beta] = E_{\mathbf{x},\mathbf{z}}\left[E[f(\Phi,\Theta)|\mathbf{x},\mathbf{z},\mathcal{D}^{\text{train}},\alpha,\beta]\right] \quad (9)$$

$$\approx \frac{1}{S}\sum_{s=1}^{S} E[f(\Phi,\Theta)|\mathbf{x}^s,\mathbf{z}^s,\mathcal{D}^{\text{train}},\alpha,\beta] \quad (10)$$

where S is the number of samples obtained from the Gibbs sampler. In practice, computing $E[f(\Phi,\Theta)|\mathbf{x}^s,\mathbf{z}^s,\mathcal{D}^{\text{train}},\alpha,\beta]$ may be difficult, as it requires integrating the function over the posterior Dirichlet distributions. When this is the case, we use the approximation $E[f(\Phi,\Theta)] \approx f(E[\Phi],E[\Theta])$, where $E[\Phi]$ and $E[\Theta]$ refer to the posterior means given in Equations 7 and 8. This is exact when f is linear, and provides a lower bound when f is convex.

Finally, we note that this strategy will only be effective if $f(\Phi, \Theta)$ is invariant under permutations of the columns of Φ and Θ. Like any mixture model, the author topic model suffers from a lack of identifiability: the posterior probability of Φ and Θ is unaffected by permuting their columns. Consequently, there need be no correspondence between the values in a particular column across multiple samples produced by the Gibbs sampler.

4 Experimental Results

We trained the author topic model on three large document data sets. The first is a set of papers from 13 years (1987 to 1999) of the Neural Information Processing (NIPS) Conference[1]. This data set contains $D = 1,740$ papers, $A = 2,037$ different authors, a total of $N = 2,301,375$ word tokens, and a vocabulary size of $W = 13,649$ unique words. The second corpus consists of a large collection of extracted abstracts from the CiteSeer digital library Lawrence et al. [1999], with $D = 150,045$ abstracts with $A = 85,465$ authors and $N = 10,810,003$ word tokens and a vocabulary of $W = 30,799$ unique words. The third corpus is the recently released Enron email data set[2], where we used a set of $D = 121,298$ emails, with $A = 11,195$ unique authors, and $N = 4,699,573$ word tokens. We preprocessed each set of documents by removing stop words from a standard list.

For each data set we ran 10 different Markov chains, where each was started from a different set of random assignments of authors and topics. Each of the 10 Markov chains was run for a fixed number of 2000 iterations. For the NIPS data set and a 100-topic solution, 2000 iterations of the Gibbs sampler took 12 hours of wall-clock time on a standard 2.5 Ghz PC workstation (22 seconds per iteration). For a 300-topic solution, CiteSeer took on the order of 200 hours for 2000 iterations (6 minutes per iteration), and for a 200-topic solution Enron took 23 hours for 2000 iterations (42 secs per iteration).

As mentioned earlier, in the experiments described in this paper we do not estimate the hyperparameters α and β—instead they are fixed at $50/T$ and 0.01 respectively in each of the experiments described below.

[1] Available on-line at http://www.cs.toronto.edu/~roweis/data.html

[2] Available on-line at http://www-2.cs.cmu.edu/~enron/

4.1 Analyzing the Gibbs Sampler using Perplexity

Assessing the convergence of the Markov chain used to sample a set of variables is a common issue that arises in applying MCMC techniques. This issue can be divided into two questions: the practical question of when the performance of a model trained by sampling begins to level out, and the theoretical question of when the Markov chain actually reaches the posterior distribution. In general, for real data sets, there is no foolproof method for answering the latter question. In this paper we will focus on the former, using the perplexity of the model on test documents to evaluate when the performance of the model begins to stabilize.

The perplexity score of a new unobserved document d that contains words $\mathbf{w}_d$, and is conditioned on the known authors of the document $\mathbf{a}_d$, is defined as

$$\text{Perplexity}(\mathbf{w}_d|\mathbf{a}_d, \mathcal{D}^{\text{train}}) = \exp\left(-\frac{\log p(\mathbf{w}_d|\mathbf{a}_d, \mathcal{D}^{\text{train}})}{N_d}\right) \tag{11}$$

where $p(\mathbf{w}_d|\mathbf{a}_d, \mathcal{D}^{\text{train}})$ is the probability assigned by the author topic model (trained on $\mathcal{D}^{\text{train}}$) to the words $\mathbf{w}_d$ in the test document, conditioned on the known authors $\mathbf{a}_d$ of the test document, and where N_d is the number of words in the test document. For multiple test documents, we report the average perplexity over documents, i.e., $\langle\text{Perplexity}\rangle = \sum_{d=1}^{D^{\text{test}}} \text{Perplexity}(\mathbf{w}_d|\mathbf{a}_d, \mathcal{D}^{\text{train}})/D^{\text{test}}$. The lower the perplexity the better the performance of the model.

We can obtain an approximate estimate of perplexity by averaging over multiple samples, as in Equation 10:

$$p(\mathbf{w}_d|\mathbf{a}_d, \mathcal{D}^{\text{train}}) \approx \frac{1}{S}\sum_{s=1}^{S}\prod_{i=1}^{N_d}\left[\frac{1}{A_d}\sum_{a\in A_d,t} E[\theta_{at}\phi_{tw_i}|\mathbf{x}^s, \mathbf{z}^s, \mathcal{D}^{\text{train}}, \alpha, \beta]\right].$$

In order to ensure that the sampler output covers the entire space we run multiple replications of the MCMC, i.e., the samples are generated from multiple chains, each starting at a different state (e.g., [Brooks, 1998]). Empirical results with both the CiteSeer and NIPS data sets, using different values for S, indicated that $S = 10$ samples is a reasonable choice to get a good approximation of the perplexity.

Figure 4 shows perplexity as a function of the number of iterations of the Gibbs sampler, for a model with 300 topics fit to the CiteSeer data. Samples $\mathbf{x}^s, \mathbf{z}^s$ obtained from the Gibbs sampler after s iterations (where s is the x-axis in the graph) are used to produce a perplexity score on test documents. Each point represents the averaged perplexity over $D^{\text{test}} = 7502$ CiteSeer test documents. The inset in Figure 4 shows the perplexity for two different cases. The upper curves show the perplexity derived from a single sample $S = 1$ (upper curves), for 10 different such samples (10 different Gibbs sampler runs). The lower curve in the inset shows the perplexity obtained from averaging over $S = 10$ samples. It is clear from the figure that averaging helps, i.e., significantly better predictions (lower perplexity) are obtained when using multiple samples from the Gibbs sampler than just a single sample.

It also appears from Figure 4 that performance of models trained using the Gibbs sampler appears to stabilize rather quickly (after about 100 iterations), at least in terms of perplexity on test documents. While this is far from a formal diagnostic test of convergence, it is nonetheless reassuring, and when combined with the results on topic stability and topic interpretation in the next sections, lends some confidence that the model finds a relatively stable topic-based representation of the corpus. Qualitatively similar results were obtained for the NIPS corpus, i.e., averaging provides a significant reduction in perplexity and the perplexity values "flatten out" after a 100 or so iterations of the Gibbs sampler.

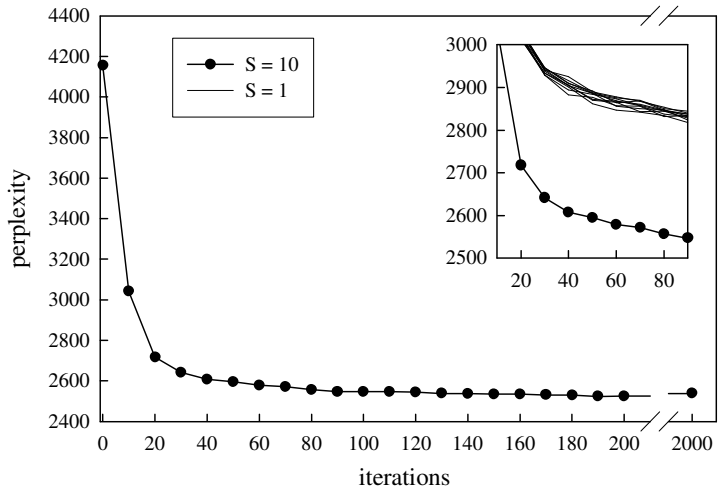

Figure 4: Perplexity as a function of iterations of the Gibbs sampler for a $T = 300$ model fit to the CiteSeer dataset. The inset shows the perplexity values (upper curves) from 10 individual chains during early iterations of the sampler, while the lower curve shows the perplexity obtained by averaging these 10 chains. The full graph shows the perplexity from averaging again, but now over a larger range of sampling iterations.

4.2 Topic Stability

While perplexity computations can and should be averaged over different Gibbs sampler runs, other applications of the model rely on the interpretations of individual topics and are based on the analysis of individual samples. Because of exchangeability of the topics, it is possible that quite different topic solutions are found across samples. In practice, however, we have found that the topic solutions are relatively stable across samples, with only a small subset of unique topics appearing in any sample. We assessed topic stability by a greedy alignment algorithm that tries to find the best one-to-one topic correspondences across samples. The algorithm calculates all pairwise symmetrized KL distances between the T topic distributions over words from two different samples (in this analysis, we ignored the accompanying distributions over authors). It starts by finding the topic pair with lowest (symmetrized) KL distance and places those in correspondence, followed in greedy fashion with the next best topic pair.

Figure 5 illustrates the alignment results for two 100 topic samples for the NIPS data set taken at 2000 iterations from different Gibbs sampler runs. The bottom panel shows the rearranged distance matrix that shows a strong diagonal structure. Darker colors indicate lower KL distances. The top panel shows the best and worst aligned pair of topics across two samples (corresponding to the top-left and bottom-right pair of topics on the diagonal of the distance matrix). The best aligned topic pair has an almost identical probability distribution over words whereas the worst aligned topic pair shows no correspondence at all. Roughly 80 of 100 topics have a reasonable degree of correspondence that would be associated with the same subjective interpretation. We obtained similar results for the CiteSeer data set.

4.3 Interpreting Author Topic Model Results

We can use point estimates of the author topic parameters to look at specific author-topic and topic-word distributions and related quantities that can be derived from these parameters (such as the probability of an author given a randomly selected word from a topic). In the results described

BEST KL = 1.03

sample 1		sample 2	
topic 81		topic 41	
WORD	PROB.	WORD	PROB.
MOTOR	.0415	MOTOR	.0405
TRAJECTORY	.0311	ARM	.0297
ARM	.0267	TRAJECTORY	.0296
HAND	.0224	HAND	.0244
MOVEMENT	.0217	MOVEMENT	.0227
INVERSE	.0190	INVERSE	.0209
DYNAMICS	.0188	JOINT	.0208
CONTROL	.0181	DYNAMICS	.0179
JOINT	.0176	CONTROL	.0152
POSITION	.0166	POSITION	.0152

WORST KL = 9.49

sample 1		sample 2	
topic 64		topic 22	
WORD	PROB.	WORD	PROB.
ORDER	.1748	FUNCTION	.0913
SCALE	.0527	ORDER	.0637
HIGHER	.0353	EQUATION	.0482
MULTI	.0281	TERMS	.0273
NOTE	.0276	TERM	.0269
VOLUME	.0188	THEORY	.0138
TERMS	.0185	APPROXIMATION	.0137
STRUCTURE	.0170	FUNCTIONS	.0137
SCALES	.0169	FORM	.0136
INVARIANT	.0117	OBTAINED	.0126

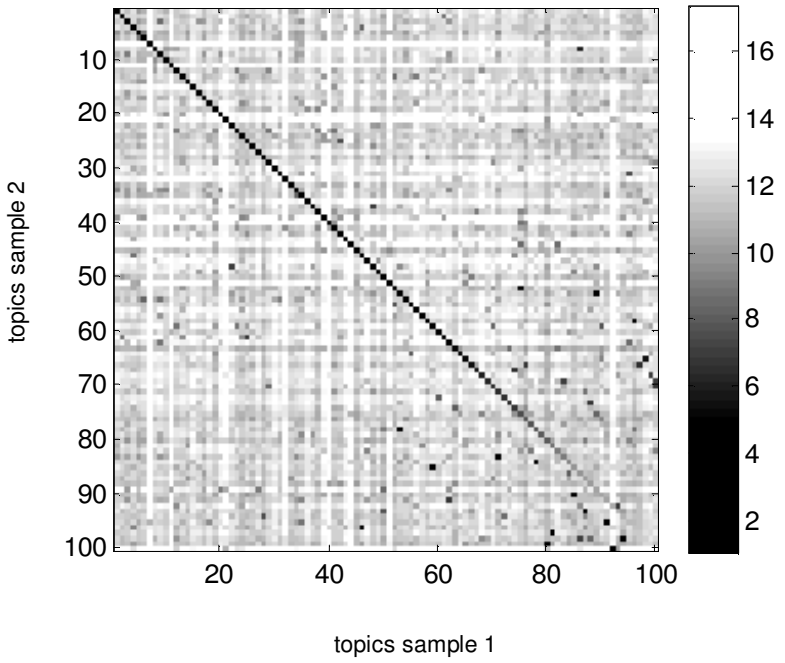

Figure 5: Topic stability across two different runs on the NIPS corpus: best and worst aligned topics (top), and KL distance matrix between topics (bottom).

below we take a specific sample $\mathbf{x}^s, \mathbf{z}^s$ after 2000 iterations from a single (arbitrarily selected) Gibbs run, and then generate point estimates of Φ and Θ using Equation 8. Equations for computing the conditional probabilities in the different tables are provided in Appendix B.

Complete lists of tables for the 100-topic NIPS model, the 300-topic CiteSeer model, and the 200-topic Enron email model are available at `http://www.datalab.uci.edu/author-topic`. In addition there is an online JAVA browser for interactively exploring authors, topics, and documents.

4.3.1 Examples from a NIPS Author Topic Model

The NIPS conference is characterized by contributions from a number of different research communities within both machine learning and neuroscience. Figure 1 illustrates examples of 8 topics (out of 100) as learned by the model for the NIPS corpus. Each topic is illustrated with (a) the top 10 words most likely to be generated conditioned on the topic, and (b) the top 10 most likely authors to have generated a word conditioned on the topic. The first 6 topics we selected for display (left to right across the top and the first two on the left on the bottom) are quite specific representations of different topics that have been popular at the NIPS conference over the time-period 1987–99: visual modeling, handwritten character recognition, SVMs and kernel methods, source separation methods, Bayesian estimation, and reinforcement learning. For each topic, the top 10 most likely authors are well-known authors in terms of NIPS papers written on these topics (e.g., Singh, Barto, and Sutton in reinforcement learning). While most (order of 80 to 90%) of the 100 topics in the model are similarly specific in terms of semantic content, the remaining 2 topics we display illustrate some of the other types of "topics" discovered by the model. Topic 62 is somewhat generic, covering a broad set of terms typical to NIPS papers, with a somewhat flatter distribution over authors compared to other topics. These types of topics tend to be broadly spread over many documents in the corpus, and can be viewed as syntactic in the context of NIPS papers. In contrast, the "semantic content topics" (such as the first 6 topics in Figure 1) are more narrowly concentrated within a smaller set of documents. Topic 16 is somewhat oriented towards Geoff Hinton's group at the University of Toronto, containing the words that commonly appeared in NIPS papers authored by members of that research group, with an author list consisting largely of Hinton and his students and collaborators.

4.3.2 Examples from a CiteSeer Author Topic Model

Results from a 300 topic model for a set of 150,000 CiteSeer abstracts are shown in Figure 6, again in terms of top 10 most likely words and top 10 most likely authors per topic. The first four topics describe specific areas within computer science, covering Bayesian learning, data mining, information retrieval, and database querying. The authors associated with each topic are quite specific to the words in that topic. For example, the most likely authors for the Bayesian learning topic are well-known authors who frequently write on this topic at conferences such as UAI and NIPS. Similarly, for the data mining topic, all of the 10 most likely authors are frequent contributors of papers at the annual ACM SIGKDD conference on data mining. The full set of 300 topics discovered by the model for CiteSeer provide a broad coverage of modern computer science and can be explored online using the aforementioned browser tool.

Not all documents in CiteSeer relate to computer science. Topic 82, on the right side of Figure 6, is associated with astronomy. This is due to the fact that CiteSeer does not crawl the Web looking for computer science papers per se, but instead searches for documents that are similar in some sense to a general template format for research papers.

TOPIC 54	
WORD	**PROB.**
BAYESIAN	.0743
MODEL	.0505
MODELS	.0401
PRIOR	.0277
DATA	.0271
MIXTURE	.0254
INFERENCE	.0222
EM	.0211
POSTERIOR	.0200
STATISTICAL	.0197
AUTHOR	**PROB.**
Ghahramani_Z	.0098
Koller_D	.0083
Friedman_N	.0078
Heckerman_D	.0075
Jordan_M	.0066
Williams_C	.0058
Jaakkola_T	.0053
Hinton_G	.0052
Raftery_A	.0050
Tresp_V	.0049

TOPIC 136	
WORD	**PROB.**
DATA	.1577
MINING	.0671
DISCOVERY	.0425
ASSOCIATION	.0326
ATTRIBUTES	.0325
LARGE	.0288
DATABASES	.0234
PATTERNS	.0212
KNOWLEDGE	.0172
ITEMS	.0171
AUTHOR	**PROB.**
Han_J	.0165
Zaki_M	.0088
Cheung_D	.0076
Liu_B	.0067
Mannila_H	.0053
Rastogi_R	.0050
Hamilton_H	.0050
Shim_K	.0047
Toivonen_H	.0047
Ng_R	.0047

TOPIC 23	
WORD	**PROB.**
RETRIEVAL	.1209
INFORMATION	.0623
TEXT	.0539
DOCUMENTS	.0422
DOCUMENT	.0329
QUERY	.0243
CONTENT	.0241
INDEXING	.0238
BASED	.0195
USER	.0175
AUTHOR	**PROB.**
Oard_D	.0094
Jones_K	.0064
Croft_W	.0060
Hawking_D	.0058
Callan_J	.0052
Smeaton_A	.0052
Voorhees_E	.0052
Schauble_P	.0047
Singhal_A	.0042
Fuhr_N	.0042

TOPIC 49	
WORD	**PROB.**
QUERY	.1798
QUERIES	.1262
DATABASE	.0432
RELATIONAL	.0396
DATABASES	.0298
DATA	.0159
OPTIMIZATION	.0147
RELATIONS	.0127
ANSWER	.0118
RESULT	.0115
AUTHOR	**PROB.**
Suciu_D	.0120
Libkin_L	.0098
Wong_L	.0093
Naughton_J	.0076
Levy_A	.0066
Abiteboul_S	.0065
Lenzerini_M	.0058
Raschid_L	.0055
DeWitt_D	.0055
Ross_K	.0051

TOPIC 82	
WORD	**PROB.**
STARS	.0165
OBSERVATIONS	.0160
SOLAR	.0153
RAY	.0134
MAGNETIC	.0130
GALAXIES	.0129
MASS	.0126
EMISSION	.0115
SUBJECT	.0112
DENSITY	.0111
AUTHOR	**PROB.**
Falcke_H	.0167
Linsky_J	.0152
Butler_R	.0090
Bjorkman_K	.0068
Christen.-D_J	.0067
Mursula_K	.0067
Knapp_G	.0065
Nagar_N	.0059
Cranmer_S	.0055
Gregg_M	.0055

Figure 6: Examples of topics and authors learned from the CiteSeer corpus.

topic 182	
WORD	**PROB.**
TEXANS	.0145
WIN	.0143
FOOTBALL	.0137
FANTASY	.0129
SPORTSLINE	.0129
PLAY	.0123
TEAM	.0114
GAME	.0112
SPORTS	.0110
GAMES	.0109

topic 113	
WORD	**PROB.**
GOD	.0357
LIFE	.0272
MAN	.0116
PEOPLE	.0103
CHRIST	.0092
FAITH	.0083
LORD	.0079
JESUS	.0075
SPIRITUAL	.0066
VISIT	.0065

topic 23	
WORD	**PROB.**
ENVIRONMENTAL	.0291
AIR	.0232
MTBE	.0190
EMISSIONS	.0170
CLEAN	.0143
EPA	.0133
PENDING	.0129
SAFETY	.0104
WATER	.0092
GASOLINE	.0086

topic 54	
WORD	**PROB.**
FERC	.0554
MARKET	.0328
ISO	.0226
COMMISSION	.0215
ORDER	.0212
FILING	.0149
COMMENTS	.0116
PRICE	.0116
CALIFORNIA	.0110
FILED	.0110

topic 18	
WORD	**PROB.**
POWER	.0915
CALIFORNIA	.0756
ELECTRICITY	.0331
UTILITIES	.0253
PRICES	.0249
MARKET	.0244
PRICE	.0207
UTILITY	.0140
CUSTOMERS	.0134
ELECTRIC	.0120

Figure 7: Examples of topics learned from the Enron email corpus.

4.4 Examples from an Enron Author Topic Model

Figure 7 shows a set of topics from a model trained on a set of 120,000 publicly available Enron emails. We automatically removed all text from the emails that was not necessarily written by the sender, such as attachments or text that could be clearly identified as being from an earlier email (e.g., in reply quotes). The topics learned by the model span both topics that one might expect to see discussed in emails within a company that deals with energy (topics 23 and 54) as well as "topical" topics such as topic 18 that directly relate to the California energy crisis in 2001–2002. Two of the topics are not directly related to official Enron business, but instead describe employees' personal interests such as Texas sports (182) and Christianity (113).

Figure 8 shows a table with the most likely topics for 6 of the 11,195 possible authors (email accounts). The first three are institutional accounts: Enron General Announcements, Outlook Migration Team (presumably an internal email account at Enron for announcements related to the Outlook email program), and The Motley Fool (a company that provides financial education and advice). The topics associated with these authors are quite intuitive. The most likely topic ($p = 0.942$) for Enron General Announcements is a topic with words that might typically be associated with general corporate information for Enron employees. The topic distribution for the Outlook Migration Team is skewed towards a single topic ($p = 0.991$) containing words that are quite specific to the Outlook email program. Likely topics for the Motley Fool include both finance and investing topics, as well as topics with HTML-related words and a topic for dates. The other 3 authors shown in Figure 8 correspond to email accounts for specific individuals in Enron—although the original data identifies individual names for these accounts we do not show them here to respect the privacy of these individuals. Author A's topics are typical of what we might expect of a senior employee in Enron, with topics related to rates and customers, to the FERC (Federal Energy Regulatory Commission), and to the California energy crisis (including mention of the California governor at the time, Gray Davis). Author B's topics are focused more on day-to-day Enron operations (pipelines, contracts, and facilities) with an additional topic for more personal matters ("good, time", etc). Finally, Author C appears to be involved in legal aspects of Enron's international activities, particularly in Central and South America. The diversity of the topic distributions for different authors in this example demonstrates clearly how the author topic model can learn about the roles and interests of different individuals from text that they have written.

5 Illustrative Applications of the Author Topic Model

In this section we provide some illustrative examples of how the author topic model can be used to answer different types of questions and prediction problems about authors and documents.

5.1 Automated Detection of Unusual Papers by Authors

Perplexity can be used to estimate the likelihood of a particular document conditioned on a particular author. We first train the model on $\mathcal{D}^{\text{train}}$. For a specific author name $\hat{a}$ of interest, we then score each document by that author as follows. We calculate a perplexity score for each document in $\mathcal{D}^{\text{train}}$ as if $\hat{a}$ was the only author, i.e., even for a document with other authors, we condition on only $\hat{a}$.

We use the same equation for perplexity as defined in Section 4.2 except that now $\mathbf{w}_d$ is a document that is in the training data $\mathcal{D}^{\text{train}}$. Thus, the words in a document are *not* conditionally independent, given the distribution over the model parameters Θ and Φ, as inferred from the training documents. We use as a tractable approximation $P(\mathbf{w}_d|\hat{a}, \mathcal{D}^{\text{train}}) \approx \frac{1}{S}\sum_s \prod_i \sum_t E[\theta_{\hat{a}t}\phi_{tw_i}|\mathbf{x}^s, \mathbf{z}^s, \mathcal{D}^{\text{train}}, \alpha, \beta]$.

AUTHOR = Enron General Announcements (509 emails)		
PROB.	TOPIC	WORDS
.9420	39	ENRON, EMPLOYEES, DAY, CARD, BUILDING, CALL, PLANTS, MEMBERSHIP, TRANSFER, CENTER
.0314	200	DECEMBER, JANUARY, MARCH, NOVEMBER, FEBRUARY, WEEK, FRIDAY, SEPTEMBER, WEDNESDAY, TUESDAY
.0028	147	MAIL, CUSTOMER, SERVICE, LIST, SEND, ADDRESS, CONTACT, RECEIVE, BUSINESS, REPLY
.0026	125	MEETING, CALL, MONDAY, CONFERENCE, FRIDAY, TIME, THURSDAY, OFFICE, MORNING, TUESDAY

AUTHOR = Outlook Migration Team (132 emails)		
PROB.	TOPIC	WORDS
.9910	82	OUTLOOK, MIGRATION, NOTES, OWA, INFORMATION, EMAIL, BUTTON, SEND, MAILBOX, ACCESS
.0016	91	ENRON, CORP, SERVICES, BROADBAND, EBS, ADDITION, BUILDING, INCLUDES, ATTACHMENT, COMPETITION
.0005	77	EMAIL, ADDRESS, INTERNET, SEND, ECT, MESSAGING, BUSINESS, ADMINISTRATION, QUESTIONS, SUPPORT
.0004	83	ISSUE, GENERAL, ISSUES, CASE, DUE, INVOLVED, DISCUSSION, MENTIONED, PLACE, POINT

AUTHOR = The Motley Fool (145 emails)		
PROB.	TOPIC	WORDS
.3593	17	ANALYST, SERVICES, INDUSTRY, TELECOM, ENERGY, MARKETS, FOOL, BANDWIDTH, ESOURCE, TRAINING
.0773	177	ACCOUNT, ONLINE, OFFER, TRADE, TIME, INVESTMENT, ACCOUNTS, FREE, INFORMATION, ACCESS
.0713	169	HTTP, WWW, GIF, IMAGES, ASP, SPACER, EMAIL, CGI, HTML, CLICK
.0660	200	DECEMBER, JANUARY, MARCH, NOVEMBER, FEBRUARY, WEEK, FRIDAY, SEPTEMBER, WEDNESDAY, TUESDAY

AUTHOR = Individual A (411 emails)		
PROB.	TOPIC	WORDS
.1855	105	CUSTOMERS, RATE, PG, CPUC, SCE, UTILITY, ACCESS, CUSTOMER, DECISION, DIRECT
.1289	54	FERC, MARKET, ISO, COMMISSION, ORDER, FILING, COMMENTS, PRICE, CALIFORNIA, FILED
.0920	44	MILLION, BILLION, YEAR, NEWS, CORP, CONTRACTS, GAS, COMPANY, COMPANIES, WATER
.0719	124	STATE, PUBLIC, DAVIS, SAN, GOVERNOR, COMMISSION, GOV, SUMMER, COSTS, HOUR

AUTHOR = Individual B (193 emails)		
PROB.	TOPIC	WORDS
.2590	178	CAPACITY, GAS, EL, PASO, PIPELINE, MMBTU, CALIFORNIA, SHIPPERS, MMCF, RATE
.0902	74	GAS, CONTRACT, DAY, VOLUMES, CHANGE, DAILY, DAN, MONTH, KIM, CONTRACTS
.0645	70	GOOD, TIME, WORK, TALK, DON, BACK, WEEK, DIDN, THOUGHT, SEND
.0599	116	SYSTEM, FACILITIES, TIME, EXISTING, SERVICES, BASED, ADDITIONAL, CURRENT, END, AREA

AUTHOR = Individual C (159 emails)		
PROB.	TOPIC	WORDS
.1268	42	MEXICO, ARGENTINA, ANDREA, BRAZIL, TAX, OFFICE, LOCAL, RICHARD, COPY, STAFF
.1045	189	AGREEMENT, ENA, LANGUAGE, CONTRACT, TRANSACTION, DEAL, FORWARD, REVIEW, TERMS, QUESTIONS
.0815	176	MARK, TRADING, LEGAL, LONDON, DERIVATIVES, ENRONONLINE, TRADE, ENTITY, COUNTERPARTY, HOUSTON
.0784	135	SUBJECT, REQUIRED, INCLUDING, BASIS, POLICY, BASED, APPROVAL, APPROVED, RIGHTS, DAYS

Figure 8: Selected "authors" from the Enron data set, and the four highest probability topics for each author from the author topic model.

For our CiteSeer corpus, author names are provided with a first initial and second name, e.g., A_Einstein. This means of course that for some very common names (e.g., J_Wang or J_Smith) there will be multiple actual individuals represented by a single name in the model. This "noise" in the data provides an opportunity to investigate whether perplexity scores are able to help in separating documents from different authors who have the same first initial and last name.

We focused on names of four well-known researchers in machine learning, Michael Jordan (M_Jordan), Daphne Koller (D_Koller), Tom Mitchell (T_Mitchell) and Stuart Russell (S_Russell), and derived perplexity scores in the manner described above using $S = 10$ samples. In Table 2, for each author, we list the two CiteSeer abstracts with the highest perplexity scores (most surprising relative to this author's model), the median perplexity, and the two abstracts with the lowest perplexity scores (least surprising). (Perplexity scores for all papers with these author names are provided online at `http://www.datalab.uci.edu/author-topic`).

In these examples, the most perplexing papers (from the model's viewpoint) for each author are papers that were written by a different person than the person we are primarily interested in. In each case (for example for M_Jordan) most of the papers in the data set for this author were written by the machine learning researcher of interest (in this case, Michael Jordan of UC Berkeley). Thus, the model is primarily "tuned" to the interests of that author and assigns relatively high perplexity scores to the small number of papers in the set that were written by a different author with the same name. For M_Jordan, the most perplexing paper is on programming languages and was in fact written by Mick Jordan of Sun Microsystems. In fact, of the 6 most perplexing papers for M_Jordan, 4 are on software management and the JAVA programming language, all written by Mick Jordan. The other two papers were in fact co-authored by Michael Jordan of UC Berkeley, but in the area of link analysis, which is an unusual topic relative to the many of machine learning-oriented topics that he has typically written about in the past. The highest perplexity paper for T_Mitchell is in fact authored by Toby Mitchell and is on the topic of estimating radiation doses (quite different from the machine learning work of Tom Mitchell). The two most perplexing papers for D_Koller are also not authored by Daphne Koller of Stanford, but by two different researchers, Daniel Koller and David Koller. Moreover, the two most typical (lowest perplexity) papers of D_Koller are prototypical representatives of the research of Daphne Koller, with words such as *learning*, *Bayesian* and *probabilistic network* appearing in the titles of these two papers. For S_Russell the two most unlikely papers are about the Mungi operating system and have Stephen Russell as an author. These papers are relative outliers in terms of their perplexity scores since most of the papers for S_Russell are about reasoning and learning and were written by Stuart Russell from UC Berkeley.

5.2 Topics and Authors for New Documents

In many applications, we would like to quickly assess the topic and author assignments for new documents not contained in a text collection. Figure 9 shows an example of this type of inference. CiteSeer abstracts from two authors, B_Scholkopf and A_Darwiche were combined together into a single "pseudo-abstract" and the document was treated as if they had both written it. These two authors work in relatively different but not entirely unrelated sub-areas of computer science: Scholkopf in machine learning and Darwiche in probabilistic reasoning. The document is then parsed by the model. i.e., words are assigned to these authors. We would hope that the author topic model, conditioned now on these two authors, can separate the combined abstract into its component parts.

Instead of rerunning the algorithm for every new document added to a text collection, our strategy instead is to apply an efficient Monte Carlo algorithm that runs only on the word tokens

[AUTH1=Scholkopf_B (69%, 31%)]
[AUTH2=Darwiche_A (72%, 28%)]

A method[1] is described which like the kernel[1] trick[1] in support[1] vector[1] machines[1] SVMs[1] lets us generalize distance[1] based[2] algorithms to operate in feature[1] spaces usually nonlinearly related to the input[1] space This is done by identifying a class of kernels[1] which can be represented as norm[1] based[2] distances[1] in Hilbert spaces It turns[1] out that common kernel[1] algorithms such as SVMs[1] and kernel[1] PCA[1] are actually really distance[1] based[2] algorithms and can be run[2] with that class of kernels[1] too As well as providing[1] a useful new insight[1] into how these algorithms work the present[2] work can form the basis[1] for conceiving new algorithms

This paper presents[2] a comprehensive approach for model[2] based[2] diagnosis[2] which includes proposals for characterizing and computing[2] preferred[2] diagnoses[2] assuming that the system[2] description[2] is augmented with a system[2] structure[2] a directed[2] graph[2] explicating the interconnections between system[2] components[2] Specifically we first introduce the notion of a consequence[2] which is a syntactically[2] unconstrained propositional[2] sentence[2] that characterizes all consistency[2] based[2] diagnoses[2] and show[2] that standard[2] characterizations of diagnoses[2] such as minimal conflicts[1] correspond to syntactic[2] variations[1] on a consequence[2] Second we propose a new syntactic[2] variation on the consequence[2] known as negation[2] normal form NNF and discuss its merits compared to standard variations Third we introduce a basic algorithm[2] for computing consequences in NNF given a structured system[2] description We show that if the system[2] structure[2] does not contain cycles[2] then there is always a linear size[2] consequence[2] in NNF which can be computed in linear time[2] For arbitrary[1] system[2] structures[2] we show a precise connection between the complexity[2] of computing[2] consequences and the topology of the underlying system[2] structure[2] Finally we present[2] an algorithm[2] that enumerates[2] the preferred[2] diagnoses[2] characterized by a consequence[2] The algorithm[2] is shown[1] to take linear time[2] in the size[2] of the consequence[2] if the preference criterion[1] satisfies some general conditions

Figure 9: Automated labeling of a pseudo-abstract from two authors by the model.

in the new document, leading quickly to likely assignments of words to authors and topics. We start by assigning words randomly to co-authors and topics. We then sample new assignments of words to topics and authors by applying the Gibbs sampler only to the word tokens in the new document each time temporarily updating the count matrices C^{WT} and C^{AT}. The resulting assignments of words to authors and topics can be saved after a few iterations (10 iterations in our simulations).

Figure 9 shows the results after the model has classified each word according to the most likely author. Note that the model only sees a bag of words and is not aware of the word order that we see in the figure. For readers viewing this in color, the more red a word is then the more likely it is to have been generated (according to the model) by Scholkopf (and blue for Darwiche). For readers viewing the figure in black and white, the superscript 1 indicates words classified by the model for Scholkopf, and superscript 2 for Darwiche. The results show that all of the significant content words (such as kernel, support, vector, diagnoses, directed, graph) are classified correctly. As we might expect most of the "errors" are words (such as "based" or "criterion") that are not specific to either authors' area of research. Were we to use word order in the classification, and classify (for example) whole sentences, the accuracy would increase further. As it is, the model correctly classifies 69% of Scholkopf's words and 72% of Darwiche's.

6 Comparing Different Generative Models

In this section we describe several alternative generative models that model authors and words and discuss similarities and differences between these models with our proposed author topic model. Many of these models are special cases of the author topic model. Appendix C presents a characterization of several of these models in terms of methods of matrix factorization, which reveals some of these relationships. In this section, we also compare the predictive power of the author topic model (in terms of perplexity on out-of-sample documents) with a number of these alternative models.

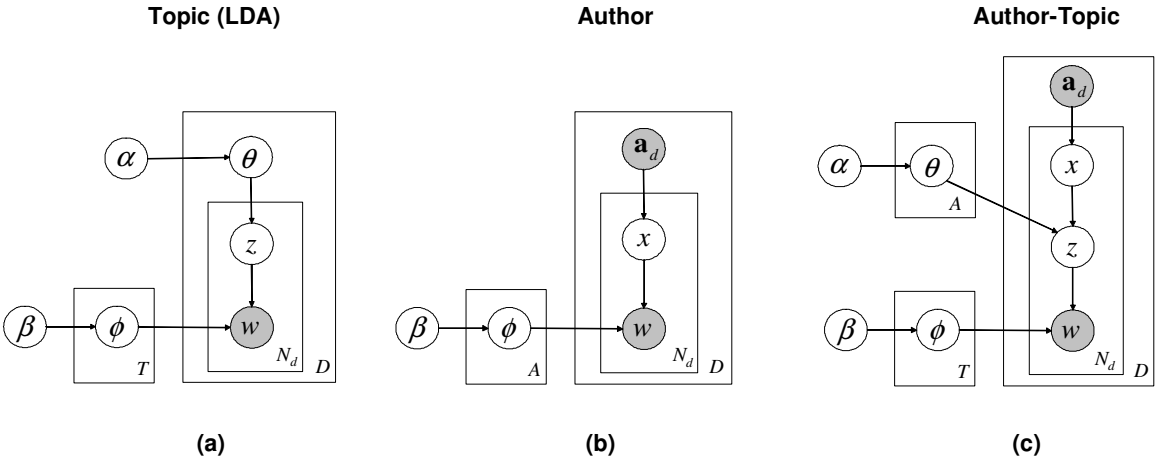

Figure 10: Different generative models for documents.

6.1 A Simple Topic (LDA) Model

As mentioned earlier in the paper, there have been a number of other earlier approaches to modeling document content are based on the idea that the probability distribution over words in a document can be expressed as a mixture of topics, where each topic is a probability distribution over words [Blei et al., 2003, Hofmann, 1999, Ueda and Saito, 2003, Iyer and Ostendorf, 1999]. Here we will focus on one such model—Latent Dirichlet Allocation [LDA; Blei et al., 2003].[3] In LDA, the generation of a corpus is a three step process. First, for each document, a distribution over topics is sampled from a Dirichlet distribution. Second, for each word in the document, a single topic is chosen according to this distribution. Finally, each word is sampled from a multinomial distribution over words specific to the sampled topic.

The parameters of this model are similar to those of the author topic model: Φ represents a distribution over words for each topic, and Θ represents a distribution over topics for each document. Using this notation, the generative process can be written as:

1. For each document $d = 1, ..., D$ choose $\theta_d \sim \text{Dirichlet}(\alpha)$
   For each topic $t = 1, ..., T$ choose $\phi_t \sim \text{Dirichlet}(\beta)$
2. For each document $d = 1, ..., D$
   For each word w_i, indexed by $i = 1, ..N_d$
   Conditioned on d choose a topic $z_i \sim \text{Discrete}(\theta_d)$
   Conditioned on z_i choose a word $w_i \sim \text{Discrete}(\phi_{z_i})$

A graphical model corresponding to this process is shown in Figure 10(a).

Latent Dirichlet Allocation is a special case of the author topic model, corresponding to the situation in which each document has a unique author. Estimating Φ and Θ provides information about the topics that participate in a corpus and the weights of those topics in each document respectively. However, this topic model provides no explicit information about the interests of

[3]The model we describe is actually the *smoothed* LDA model with symmetric Dirichlet priors [Blei et al., 2003] as this is closest to the author topic model.

authors: while it is informative about the content of documents, authors may produce several documents—often with co-authors—and it is consequently unclear how the topics used in these documents might be used to describe the interests of the authors.

6.2 A Simple Author Model

Topic models illustrate how documents can be modeled as mixtures of probability distributions. This suggests a simple method for modeling the interests of authors, namely where words in documents are modeled directly by author-word distributions without any hidden latent topic variable, as originally proposed by McCallum [1999]. Assume that a group of authors, $\mathbf{a}_d$, decide to write the document d. For each word in the document an author is chosen uniformly at random, and a word is chosen from a probability distribution over words that is specific to that author.

In this model, Φ denotes the probability distribution over words associated with each author. The generative process is as follows:

1. For each author $a = 1, ..., A$ choose $\theta_a \sim \text{Dirichlet}(\alpha)$
2. For each document $d = 1, ..., D$
   Given the set of authors $\mathbf{a}_d$
   For each word w_i, indexed by $i = 1, ..N_d$
   Conditioned on $\mathbf{a}_d$ choose an author $x_i \sim \text{Uniform}(\mathbf{a}_d)$
   Conditioned on x_i choose a word $w_i \sim \text{Discrete}(\theta_{x_i})$

A graphical model corresponding to this generative process is shown in Figure 10(b).

This model is also a special case of the author topic model, corresponding to a situation in which there is a unique topic for each author. When there is a single author per document, it is equivalent to a naive Bayes model. Estimating Φ provides information about the interests of authors, and can be used to answer queries about author similarity and authors who write on subjects similar to an observed document. However, this author model does not provide any information about document content that goes beyond the words that appear in the document and the identities of authors of the document.

6.3 An Author Topic Model with Fictitious Authors

A potential weakness of the author topic model is that it does not allow for any idiosyncratic aspects of a document. The document is assumed to be generated by a mixture of the authors' topic distributions and nothing else. The LDA model is in a sense at the other end of this spectrum—it allows each document to have its own document-specific topic mixture. In this context it is natural to explore models that lie between these two extremes. One such model can be obtained by adding an additional unique "fictitious" author to each document. This fictitious author can account for topics and words that appear to be document-specific and not accounted for by the authors. The fictitious author mechanism in effect provides the advantage of an LDA element to the author topic model. In terms of the algorithm, the only difference between the standard author topic algorithm and the one that contains fictitious authors is that the number of authors is increased from A to $A + D$, and the number of authors per document A_d is increased by 1—the time complexity of the algorithm increases accordingly. One also has the option of putting a uniform distribution over authors (including the fictitious author) or allowing a non-uniform distribution over both true authors and the fictitious author.

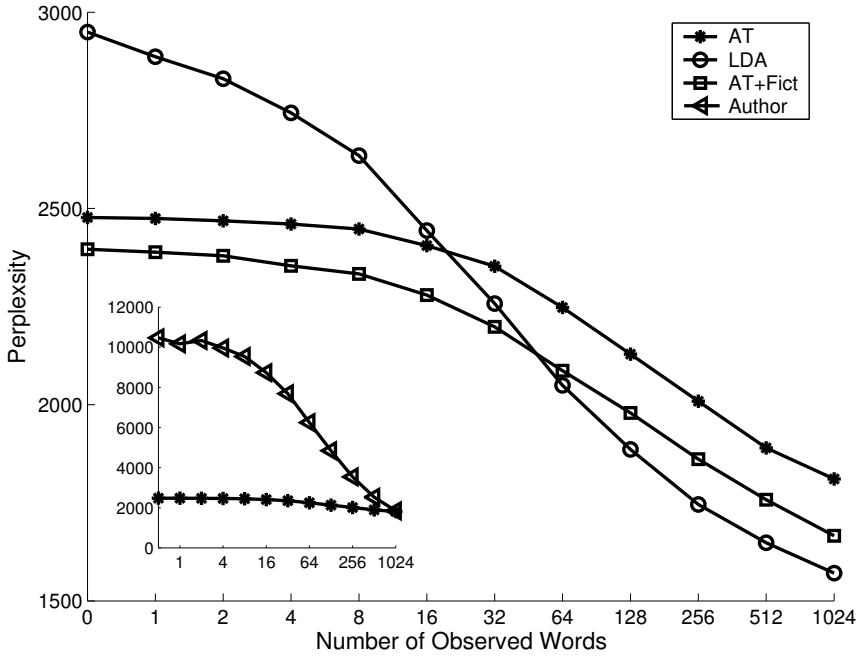

Figure 11: Averaged perplexity as a function of observed words in the test documents. The main plot shows results for the topic (LDA) model, the author topic (AT) model, and the author topic model with fictitious authors. The insert shows results for the author model and author topic model.

6.4 Comparing Perplexity for Different Models

We compare the predictive power (using perplexity) of the models discussed in this section on the NIPS document set. We divided the $D = 1,740$ NIPS papers into a training set of 1,557 papers and a test set of 183 papers of which 102 are single-authored papers. We chose the test data documents such that each author of a test set document also appears in the training set as an author. For each model, we generated $S = 10$ chains from the Gibbs sampler, each starting from a different initial conditions. We kept the 2000th sample from each chain and estimated the average perplexity using Equation 11. For all models we fixed the number of topics at $T = 100$ and used the same training set D^{train} and test set. The hyperparameter values were set in the same manner as in earlier experiments, i.e, in the LDA model and the author topic model $\alpha = 50/T = 0.5$ and $\beta = 0.01$. The single hyperparameter of the author model was set to 0.01.

In Figure 11 we present the average perplexity as a function of the number of observed words from the test documents. All models were trained on the training data and then a number of randomly selected words from the test documents (indicated on the x-axis) were used for further training. In order to reduce the time complexity of the algorithm we approximated the posterior distributions by making use of the same Monte-Carlo chains for all derivations where for each point in the graph the chain is trained further only on the observed test words. In the graph we present results for the author topic model, the topic (LDA) model, and the author topic (AT) model with fictitious authors. The insert shows the author model and the author topic model.

The author model (insert) has by far the worst performance—including latent topics significantly improves the predictive log-likelihood of such a model (lower curve in the insert). In the main plot, for relatively small numbers of observed words (up to 16), the author topic models (with and

without fictitious authors) have lower perplexity than the LDA model. The LDA model learns a topic mixture for each document in the training data. Thus, on a new document with zero or even just a few observed words, it is difficult for the LDA model to provide predictions that are tuned to that document. In contrast, the author topic model performs better than LDA with few (or even zero) words observed from a document, by making use of available the side-information about the authors of the document.

Once enough words from a specific document have been observed the predictive performance of the LDA model improves since it can learn a more accurate predictive model for that specific document. On average, after about 16 words, the LDA predictions have lower perplexity than the author-topic predictions. Above 16 observed words, on average, the author topic model is not as accurate as the LDA model since it does not have a document-specific topic mixture that can be tuned to the specific word distribution of the test document. Adding one (unique) fictitious author per document results in a curve that is systematically better than the author topic model (without a fictitious author). The fictitious author model is not quite as accurate as the LDA (topic) model after 64 words or so (on average). This is intuitively to be expected: the presence of a fictitious author gives this model more modeling flexibility compared to the author topic model, but it is still more constrained than the LDA model for a specific document.

7 Conclusions

The author topic model proposed in this paper provides a relatively simple probabilistic model for exploring the relationships between authors, documents, topics, and words. This model provides significantly improved predictive power in terms of perplexity compared to a more impoverished author model, where the interests of authors are directly modeled with probability distributions over words. When compared to the LDA topic model, the author topic model was shown to have more focused priors when relatively little is known about a new document, but the LDA model can better adapt its distribution over topics to the content of individual documents as more words are observed. The primary benefit of the author topic model is that it allows us to explicitly include authors in document models, providing a general framework for answering queries and making predictions at the level of authors as well as the level of documents.

We presented results of applying the author topic model to large text corpora, including NIPS proceedings papers, CiteSeer abstracts, and Enron emails. Potential applications include automatic reviewer recommender systems where potential reviewers or reviewer panels are matched to papers based on the words expressed in a paper as well the names of the authors. The author topic model could be incorporated in author identification systems to infer the identity of an author of a document not only on the basis of stylistic features, but also using the topics expressed in a document.

The underlying probabilistic model of the author topic model is quite simple and ignores several aspects of real-world document generation that could be explored with more advanced generative models. For example, as with many statistical models of language, the generative process does not make any assumptions about the order of words as they appear in documents. Griffiths et al. [2005] present an extension of the LDA model in which words are factorized into function words, handled by a hidden Markov model (HMM) and content words handled by a topic model. Because these models automatically parse documents into content and non-content words, there is no the need for a preprocessing stage where non-content related words are filtered out based on a predefined stop-word list. These HMM extensions could also be incorporated into the author topic model, to highlight parts of documents where content is expressed by particular authors.

Beyond the authors of a document, there are several other sources of information that can provide opportunities to learn about the set of topics expressed in a document. For example, for email documents McCallum et al. [2004] propose an extension of the author topic model where topics are conditioned on both the sender as well as the receiver. For scientific documents we have explored simple extensions within the author topic modeling framework to generalize the notion of an author to include any information source that might constrain the set of topics. For example, one can redefine $\mathbf{a}_d$ to include not only the set of authors for a document but also the set of citations. In this manner, words and topics expressed in a document can be associated with either an author or a citation. These extensions are attractive because they do not require changes to the generative model. The set of topics could also be conditioned on other information about the documents (beyond authors and citations), such as the journal source, the publication year, and the institutional affiliation of the authors. An interesting direction for future work is to develop efficient generative models where the distribution over topics is conditioned jointly on all such sources of information.

Appendix A: Deriving the Sampling Equations in the Author Topic Model

In this appendix, we set out the details of the derivation of the sampling equation, Equation 4, used to generate the samples for the author topic model. Our starting point is Equation 2; It defines the probability for a set of words. It contains probabilities of author and topic assignments and the sums over all possible assignments. As usually happens with discrete random variables and multinomial distribution, the probability distribution for the set of words can be manipulated to include sums over all possible combinations of *vector* assignments, **x,z**,

$$P\left(\mathbf{w}|\alpha,\beta,\mathcal{A},T\right)=\int\int p(\Theta,\Phi|\alpha,\beta)\prod_{d=1}^{D}\left[\frac{1}{A_d}\right]^{N_d}\sum_{\mathbf{x},\mathbf{z}}\prod_{a\in\mathbf{a}_d}\prod_{t=1}^{T}\prod_{w=1}^{W}\phi_{wt}^{C_{wt}^{WT}}\theta_{ta}^{C_{ta}^{TA}}d\Theta d\Phi \tag{12}$$

The summation here goes through all possible combinations (sometimes called the trace over all possible configurations), it contains $\prod_{d=1}^{D}\left(A_d^{N_d}\times T^{N_d}\right)$ different elements, different possible assignments. The assignments are summarized into two variables, C_{ta}^{TA}, the number of words assigned to topic t for author a, and C_{wt}^{WT} the number of words from the w entry in the vocabulary that are assigned to topic t.

One should bear in mind that in the training phase, the word vector, $\mathbf{w}$, is observed and the aim is to estimate the posterior distributions of the latent variables. After these distributions are estimated, as often happens in Bayesian models they become priors for estimates of word distributions in new, test, documents. As a first step we estimate the posterior distribution of $\mathbf{x}$ and $\mathbf{z}$, the author and topic assignments to words. They are inferred by a standard sampling technique, Gibbs sampling. Gibbs sampling requires knowing the full conditional probability distribution, the probability of assigning topic t to the ith word in the dth document, conditioned on all observed words and current assignments of authors and topics. This conditional distribution can be derived from Equation 12.

By writing the Dirichlet distributions over Θ and Φ explicitly in Equation 12 one gets

$$P\left(\mathbf{w}|\alpha,\beta,\mathcal{A},T\right)=\sum_{\mathbf{x},\mathbf{z}}\int\int P\left(\mathbf{z},\mathbf{x},\mathbf{w},\Theta,\Phi|\mathcal{A},\alpha,\beta\right)d\Theta d\Phi \tag{13}$$

where

$$P\left(\mathbf{z},\mathbf{x},\mathbf{w},\Theta,\Phi|\mathcal{A},\alpha,\beta\right)=\text{Const}\prod_{a=1}^{A}\prod_{t=1}^{T}\prod_{w=1}^{W}\theta_{ta}^{\alpha-1}\phi_{wt}^{\beta-1}\theta_{ta}^{C_{ta}^{TA}}\phi_{wt}^{C_{wt}^{WT}} \tag{14}$$

with

$$\text{Const}=\left[\frac{\Gamma(T\alpha)}{(\Gamma(\alpha))^{T}}\right]^{A}\left[\frac{\Gamma(W\beta)}{(\Gamma(\beta))^{W}}\right]^{T}\prod_{d=1}^{D}\frac{1}{A_{d}^{N_{d}}} \tag{15}$$

provided $\mathbf{x}$ assigns only authors $a\in\mathbf{a}_d$ for each document d, and 0 otherwise. The integration over both random variables, Θ and Φ, in Equation 13, is over the simplex. These Dirichlet integrals are well-known (see, e.g, [Box and Tiao, 1973]), they are of the type $\int\prod_{m=1}^{M}[x_m^{k_m}\ dx_m]=\frac{\prod_{m=1}^{M}\Gamma(k_m)}{\sum_{m=1}^{M}k_m}$, with the integral over the simplex. Making use of this identity one obtains

$$P\left(\mathbf{z},\mathbf{x},\mathbf{w}|\mathcal{A},\alpha,\beta\right)=\text{Const}\prod_{a=1}^{A}\left[\frac{\prod_{t=1}^{T}\Gamma(C_{ta}^{TA}+\alpha)}{\Gamma(\sum_{t'}C_{t'a}^{TA}+T\alpha)}\right]\prod_{t=1}^{T}\left[\frac{\prod_{w=1}^{W}\Gamma(C_{wt}^{WT}+\beta)}{\Gamma(\sum_{w'}C_{w't}^{WT}+W\beta)}\right] \tag{16}$$

Note that so far no approximation is employed. We need to estimate $P(\mathbf{z},\mathbf{x}|\mathcal{D}^{\text{train}},\alpha,\beta)$—this estimation is carried by a Gibbs sampler. The Gibbs sampler utilizes the conditional distribution in Equation 17, found by employing Bayes rule,

$$P\left(z_i=t,x_i=a|w_i=w,\mathbf{z}_{-i},\mathbf{x}_{-i},\mathbf{w}_{-i},\mathcal{A},\alpha,\beta\right)=\frac{P\left(\mathbf{z},\mathbf{x},\mathbf{w}|\mathcal{A},\alpha,\beta\right)}{\sum_{z_i,x_i}P\left(\mathbf{z},\mathbf{x},\mathbf{w}|\mathcal{A},\alpha,\beta\right)} \tag{17}$$

Here $\mathbf{y}_{-i}$ stands for all components of the vector $\mathbf{y}$ except for the ith component. Note that the constant in Equation 15 cancels out, and from the Γ functions only terms that contain the value of the ith word, w, and the assignment of the ith topic to t and the ith author to a, remain.

Appendix B: Computing probabilities from a single sample

In Figures 1, 2, 6, 7 we presented examples of topics, with predictive distributions for words and authors given a particular topic assignment. In this appendix we provide the details on how to compute these predictive distributions for a particular sample.

The probability that a new word, in a particular sample s, would be $w_{N+1}=w$, given that it is generated from topic $z_{N+1}=t$, is given by

$$P\left(w_{N+1}=w|z_{N+1}=t\right)=\frac{(C_{wt}^{WT})^{s}+\beta}{\sum_{w'}(C_{w't}^{WT})^{s}+W\beta}. \tag{18}$$

Similarly, the probability that a novel word generated by the author $x_{N+1}=a$ would be assigned to topic $z_{N+1}=t$ is obtained by

$$P\left(z_{N+1}=t|x_{N+1}=a\right)=\frac{(C_{ta}^{TA})^{s}+\alpha}{\sum_{t'}(C_{t'a}^{TA})^{s}+T\alpha} \tag{19}$$

(Note that for the sake of clarity we omitted terms that we condition on from the probabilites in this section; Terms such as $\mathbf{x}^s$, $\mathbf{z}^s$, $\mathcal{D}^{\text{train}}$, α β and T). We can also compute the probability that a novel word is authored by author $x_{N+1}=a$ given that it is assigned to topic $z_{N+1}=t$ given a sample from the posterior distribution, $\mathbf{x}^s$,$\mathbf{z}^s$. The novel word is part of a new unobserved

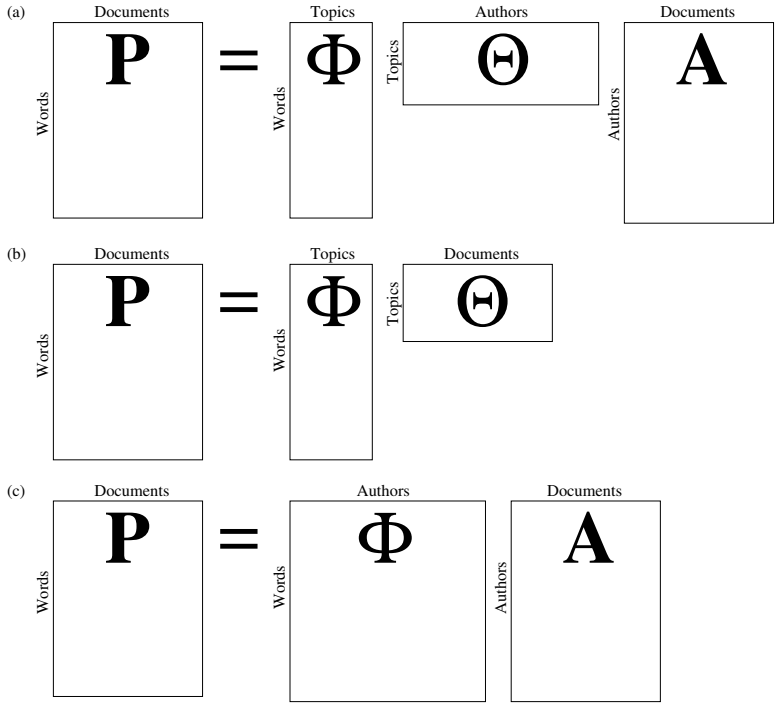

Figure 12: Matrix factorization interpretation of different models. (a) The author topic model (b) A simple topic model. (c) A simple author model.

document that contains a single word and is authored by all authors in the corpus. One can first calculate the joint probability of the author assignment as well as the topic assignment,

$$P\left(z_{N+1}=t, x_{N+1}=a\right)=P\left(z_{N+1}=t|x_{N+1}=a\right)P\left(x_{N+1}=a\right)=\frac{1}{A}\frac{(C_{ta}^{TA})^{s}+\alpha}{\sum_{t'}(C_{t'a}^{TA})^{s}+T\alpha}, \tag{20}$$

and using Bayes rule one obtains

$$P\left(x_{N+1}=a|z_{N+1}=t\right)=\frac{\frac{(C_{ta}^{TA})^{s}+\alpha}{\sum_{t'}(C_{t'a}^{TA})^{s}+T\alpha}}{\sum_{a'}\frac{(C_{ta'}^{TA})^{s}+\alpha}{\sum_{t'}(C_{t'a'}^{TA})^{s}+T\alpha}}. \tag{21}$$

Appendix C: Interpreting models as matrix factorization

The relationships among the models discussed in Section 6 can be illustrated by interpreting each model as a form of matrix factorization [c.f. Lee and Seung, 1999, Canny, 2004]. Each model specifies a different scheme for obtaining a probability distribution over words for each document in a corpus. These distributions can be assembled into a $W \times D$ matrix $\mathbf{P}$, where p_{wd} is the probability of word w in document d. In all three models, $\mathbf{P}$ is a product of matrices. As shown in Figure 12, the models differ only in which matrices are used.

In the author topic model, $\mathbf{P}$ is the product of three matrices: the $W \times T$ matrix of distributions over words Φ, the $T \times A$ matrix of distributions over topics Θ, and an $A \times D$ matrix $\mathbf{A}$, as shown in Figure 12 (a). The matrix $\mathbf{A}$ expresses the uniform distribution over authors for each document, with a_{ad} taking value $\frac{1}{A_d}$ if $a \in \mathbf{a}_d$ and zero otherwise. The other models each collapse together one

pair of matrices in this product. In the topic model, Θ and $\mathbf{A}$ are collapsed together into a single $T \times D$ matrix Θ, as shown in Figure 12 (b). In the author model, Φ and Θ are collapsed together into a single $W \times A$ matrix Φ, as shown in Figure 12 (c).

Under this view of the different models, parameter estimation can be construed as matrix factorization. As Hofmann [1999] pointed out for the topic model, finding the maximum-likelihood estimates for Θ and Φ is equivalent to minimizing the Kullback-Leibler divergence between $\mathbf{P}$ and the empirical distribution over words in each document. The three different models thus correspond to three different schemes for constructing an approximate factorization of the matrix of empirical probabilities, differing only in the elements into which that matrix is decomposed.

References

Michael W. Berry, Susan T. Dumais, and Gavin W. O'Brien. Using linear algebra for intelligent information retrieval. *SIAM Review*, pages 573–595, December 1994.

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

George E. P. Box and George C. Tiao. *Bayesian Inference in Statistical Analysis*. Addison-Wesley, Reading, Mass., 1973.

Stephen Brooks. Markov chain Monte Carlo method and its application. *The Statistician*, 47: 69–100, 1998.

Wray L. Buntine and Aleks Jakulin. Applying discrete PCA in data analysis. In Max Chickering and Joseph Halpern, editors, *Proceedings of the 20th Conference on Uncertainty in Artificial Intelligence*, pages 59–66, San Francisco, CA, 2004. Morgan Kaufmann Publishers.

John Canny. GaP: a factor model for discrete data. In *SIGIR '04: Proceedings of the 27th Annual International Conference on Research and Development in Information Retrieval*, pages 122–129, New York, NY, 2004. ACM Press.

David Cohn and Thomas Hofmann. The missing link—a probabilistic model of document content and hypertext connectivity. In Todd K. Leen, Thomas G. Dietterich, and Volker Tresp, editors, *Advances in Neural Information Processing Systems 13*, pages 430–436, Cambridge, MA, 2001. MIT Press.

Douglass R. Cutting, David Karger, Jan O. Pedersen, and John W. Tukey. Scatter/gather: A cluster-based approach to browsing large document collections. In *Proceedings of the Fifteenth Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*, pages 318–329, New York, NY, 1992. ACM Press.

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

Inderjit S. Dhillon and Dharmendra S. Modha. Concept decompositions for large sparse text data using clustering. *Machine Learning*, 42(1/2):143–175, 2001.

Joachim Diederich, Jorg Kindermann, Edda Leopold, and Gerhard Paass. Authorship attribution with support vector machines. *Applied Intelligence*, 19(1):109–123, 2003.

Elena Erosheva, Stephen Fienberg, and John Lafferty. Mixed-membership models of scientific publications. *Proceedings of the National Academy of Sciences*, 101:5220–5227, 2004.

Cesim Erten, Philip J. Harding, Stephen G. Kobourov, Kevin Wampler, and Gary Yee. Exploring the computing literature using temporal graph visualization. Technical report, Department of Computer Science, University of Arizona, 2003.

Wally Gilks, Sylvia Richardson, and David Spiegelhalter. *Markov Chain Monte Carlo in Practice*. Chapman & Hall, New York, NY, 1996.

Amir Globerson and Naftali Tishby. Sufficient dimensionality reduction. *Journal of Machine Learning Research*, 3:1307–1331, 2003.

Andrew Gray, Philip Sallis, and Stephen MacDonell. Software forensics: Extending authorship analysis techniques to computer programs. In *Proceedings of the 3rd Biannual Conference of the International Association of Forensic Linguists (IAFL)*, pages 1–8, Durham, NC., 1997.

Thomas L. Griffiths and Mark Steyvers. Finding scientific topics. *Proceedings of the National Academy of Sciences of the United States of America*, 101:5228–5235, 2004.

Thomas L. Griffiths, Mark Steyvers, David M. Blei, and Joshua B. Tenenbaum. Integrating topics and syntax. In Lawrence K. Saul, Yair Weiss, and Léon Bottou, editors, *Advances in Neural Information Processing Systems 17*. MIT Press, Cambridge, MA, 2005.

Thomas Hofmann. Probabilistic latent semantic indexing. In *Proceedings of the 22nd Annual ACM Conference on Research and Development in Information Retrieval*, pages 50–57, New York, NY, 1999. ACM Press.

David I. Holmes. The evolution of stylometry in humanities scholarship. *Literary and Linguistic Computing*, 13(3):111–117, 1998.

Rukmini Iyer and Mari Ostendorf. Modelling long distance dependence in language: Topic mixtures versus dynamic cache models. *IEEE Transactions on Speech and Audio Processing*, 7(1):30–39, 1999.

Henry Kautz, Bart Selman, and Mehul Shah. Referral Web: Combining social networks and collaborative filtering. *Communications of the ACM*, 40(3):63–65, 1997.

Bradley Kjell. Authorship determination using letter pair frequency features with neural network classifiers. *Literary and Linguistic Computing*, 9(2):119–124, 1994.

Krista Lagus, Timo Honkela, Samuel Kaski, and Teuvo Kohonen. WEBSOM for textual data mining. *Artificial Intelligence Review*, 13(5-6):345–364, 1999.

Steve Lawrence, C. Lee Giles, and Kurt Bollacker. Digital libraries and autonomous citation indexing. *IEEE Computer*, 32(6):67–71, 1999.

Daniel D. Lee and H. Sebastian Seung. Learning the parts of objects by non-negative matrix factorization. *Nature*, 401:788, 1999.

Katherine W. McCain. Mapping authors in intellectual space: a technical overview. *Journal of the American Society of Information Science*, 41(6):433–443, 1990.

Andrew McCallum, , Andres Corrada Emmanuel, and Xuerui Wang. The author-recipient-topic model for topic and role discovery in social networks: experiments with Enron and academic email. Technical Report UM-CS-2004-096, Department of Computer Science, University of Massachusetts, 2004.

Andrew McCallum. Multi-label text classification with a mixture model trained by EM. In *AAAI Workshop on Text Learning*. 1999.

Andrew McCallum, Kamal Nigam, and Lyle H. Ungar. Efficient clustering of high-dimensional data sets with application to reference matching. In *Proceedings of the Sixth ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, pages 169–178, New York, NY, 2000. ACM Press.

Thomas Minka and John Lafferty. Expectation-propagation for the generative aspect model. In *Proceedings of the 18th Conference on Uncertainty in Artificial Intelligence*, pages 352–359, San Francisco, CA, 2002. Morgan Kaufmann Publishers.

Fredrick Mosteller and David Wallace. *Inference and Disputed Authorship: The Federalist Papers*. Addison-Wesley, Reading, MA, 1964.

Peter Mutschke. Mining networks and central entities in digital libraries: a graph theoretic approach applied to co-author networks. Intelligent Data Analysis 2003, Lecture Notes in Computer Science 2810, pages 155–166. Springer Verlag, 2003.

Mark Newman. Scientific collaboration networks. I. Network construction and fundamental results. *Physical Review E*, 64(1):016131, 2001.

Alexandrin Popescul, Lyle H. Ungar, Gary William Flake, Steve Lawrence, and C. Lee Giles. Clustering and identifying temporal trends in document databases. In *Proceedings of the IEEE Advances in Digital Libraries 2000*, pages 173–182, Los Alamitos, CA, 2000. IEEE Computer Society.

Jonathan Pritchard, Matthew Stephens, and Peter Donnelly. Inference of population structure using multilocus genotype data. *Genetics*, 155:945–959, 2000.

Yee Whye Teh, Michael I. Jordan, Matthew J. Beal, and David M. Blei. Hierarchical Dirichlet processes. In Lawrence K. Saul, Yair Weiss, and Léon Bottou, editors, *Advances in Neural Information Processing Systems 17*, Cambridge, MA, 2005. MIT Press.

Ronald Thisted and Bradley Efron. Did Shakespeare write a newly-discovered poem? *Biometrika*, 74:445–455, 1987.

Naonori Ueda and Kazumi Saito. Parametric mixture models for multi-labeled text. In Suzanna Becker, Sebastian Thrun, and Klaus Obermayer, editors, *Advances in Neural Information Processing Systems 15*, pages 721–728, Cambridge, MA, 2003. MIT Press.

Max Welling, Michal Rosen-Zvi, and Geoffrey Hinton. Exponential family harmoniums with an application to information retrieval. In Lawrence K. Saul, Yair Weiss, and Léon Bottou, editors, *Advances in Neural Information Processing Systems 17*. MIT Press, Cambridge, MA, 2005.

Scott White and Padhraic Smyth. Algorithms for estimating relative importance in networks. In *Proceedings of the Ninth ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, pages 266–275, New York, NY, 2003. ACM Press.

Yiming Yang. An evaluation of statistical approaches to text categorization. *Information Retrieval*, 1(1-2):69–90, 1999.

Table 2: Papers ranked by perplexity for different authors

Paper Titles for M_Jordan, from 57 documents	**Perplexity Score**
An Orthogonally Persistent Java	16021
Defining and Handling Transient Fields in PJama	14555
MEDIAN SCORE	2567
Learning From Incomplete Data	702
Factorial Hidden Markov Models	687

Paper Titles for D_Koller, from 74 documents	**Perplexity Score**
A Group and Session Management System for Distributed Multimedia Applications	9057
An Integrated Global GIS and Visual Simulation System	7879
MEDIAN SCORE	1854
Active Learning for Parameter Estimation in Bayesian Networks	756
Adaptive Probabilistic Networks with Hidden Variables	755

Paper Titles for T_Mitchell, from 13 documents	**Perplexity Score**
A method for estimating occupational radiation dose to individuals, using weekly dosimetry data	8814
Text classification from labeled and unlabeled documents using EM	3802
MEDIAN SCORE	2837
Learning to Extract Symbolic Knowledge from the World Wide Web	1196
Explanation based learning for mobile robot perception	1093

Paper Titles for S_Russell, from 36 documents	**Perplexity Score**
Protection Domain Extensions in Mungi	10483
The Mungi Single-Address-Space Operating System	5203
MEDIAN SCORE	2837
Approximating Optimal Policies for Partially Observable Stochastic Domains	981
Adaptive Probabilistic Networks with Hidden Variables	799