In: 10th ACM SigKDD conference knowledge discovery and data mining (Seattle, 2004)

Probabilistic Author-Topic Models for Information Discovery

Mark Steyvers
Department of Cognitive Sciences
University of California, Irvine
Irvine, CA 92697-5100, USA
msteyver@uci.edu

Padhraic Smyth, Michal Rosen-Zvi
Department of Computer Science
University of California, Irvine
Irvine, CA 92697-3425, USA
{smyth,michal}@ics.uci.edu

Thomas Griffiths
Department of Psychology
Stanford University
Stanford, CA 94305
gruffydd@psych.stanford.edu

ABSTRACT

We propose a new unsupervised learning technique for extracting information from large text collections. We model documents as if they were generated by a two-stage stochastic process. Each author is represented by a probability distribution over topics, and each topic is represented as a probability distribution over words for that topic. The words in a multi-author paper are assumed to be the result of a mixture of each authors' topic mixture. 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 a large corpus of 160,000 abstracts and 85,000 authors from the well-known CiteSeer digital library, and learn a model with 300 topics. 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, significant trends in the computer science literature between 1990 and 2002, parsing of abstracts by topics and authors and detection of unusual papers by specific authors. An online query interface to the model is also discussed that allows interactive exploration of author-topic models for corpora such as CiteSeer.

Categories and Subject Descriptors

I.2.6 [**Artificial Intelligence**]: Learning

Keywords

Unsupervised learning, Gibbs sampling, text modeling

1. INTRODUCTION

With the advent of the Web and various specialized digital libraries, the automatic extraction of useful information from text has become an increasingly important research area in data mining. In this paper we discuss a new algorithm that extracts both the topics expressed in large text document collections and models how the authors of documents use those topics. The methodology is illustrated using a sample of 160,000 abstracts and 80,000 authors from the well-known CiteSeer digital library of computer science research papers (Lawrence, Giles, and Bollacker, 1999). The algorithm uses a probabilistic model that represents topics as probability distributions over words and documents as being composed of multiple topics. A novel feature of our model is the inclusion of author models, in which authors are modeled as probability distributions over topics. The author-topic models can be used to support a variety of interactive and exploratory queries on the set of documents and authors, including analysis of topic trends over time, finding the authors who are most likely to write on a given topic, and finding the most unusual paper written by a given author. Bayesian unsupervised learning is used to fit the model to a document collection.

Supervised learning techniques for automated categorization of documents into known classes or topics has received considerable attention in recent years (e.g., Yang, 1998). For many document collections, however, 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 accurately reflect rapidly evolving content.

Automatic extraction of topics from text, via unsupervised learning, has been addressed in prior work using a number of different approaches. One general approach is to represent the 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 nonlinear 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 (Berry, Dumais, and O' Brien, 1995). For example, Deerwester et al. (1990), while not using the term "topics" per se, state:

> Roughly speaking, these factors may be thought of as artificial concepts; they represent extracted common meaning components of many different

words and documents.

A somewhat different approach is to cluster the documents into groups containing similar semantic content, using any of a variety of well-known document clustering techniques (e.g., Cutting et al., 1992; McCallum, Nigam, and Ungar, 2000; Popescul et al., 2000). Each cluster of documents can then be associated with a latent topic (e.g., as represented 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. In particular, combinations of diverse topics within a single document are difficult to represent. For example, this present paper contains at least two significantly different topics: document modeling and Bayesian estimation. For this reason, other representations (such as those discussed below) 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, Ng, and Jordan (2003)).

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 new documents not seen in the training data. Blei, Ng, and Jordan (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) showed how Gibbs sampling, a Markov chain Monte Carlo technique, could be applied in this model, and illustrated this approach using 11 years of abstract data from the *Proceedings of the National Academy of Sciences*.

Our focus 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 problem of identifying what author wrote a given piece of text. 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, Sallis, and MacDonell, 1997), and the use of techniques such as 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, Newman (2001), Mutschke (2003) and Erten et al. (2003) use methods from bibliometrics, social networks, and graph theory to analyze and visualize co-author and citation relations in the scientific literature. Kautz, Selman, and Shah (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 text 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, and the application of this model to a large well-known document corpus in computer science. 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: in Section 2 we describe the author-topic model and outline 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 3 illustrates the application of the model to a large collection of abstracts from the CiteSeer system, with examples of specific topics and specific author models that are learned by the algorithm. In Section 4 we illustrate a number of applications of the model, including the characterization of topic trends over time (which provides some interesting insights on the direction of research in computer science), and the characterization of which papers are most typical and least typical for a given author. An online query interface to the system is described in Section 5, allowing users to query the model over the Web—an interesting feature of the model is the coupling of Bayesian sampling and relational database technology to answer queries in real-time. Section 6 contains a brief discussion of future directions and concluding comments.

2. AN OVERVIEW OF THE AUTHOR-TOPIC MODEL

2.1 The Probabilistic Generative Model

The author-topic model reduces the process of writing a scientific document to a simple series of probabilistic steps. The model not only discovers what topics are expressed in a document, but also which authors are associated with each topic. To simplify the representation of documents, we use

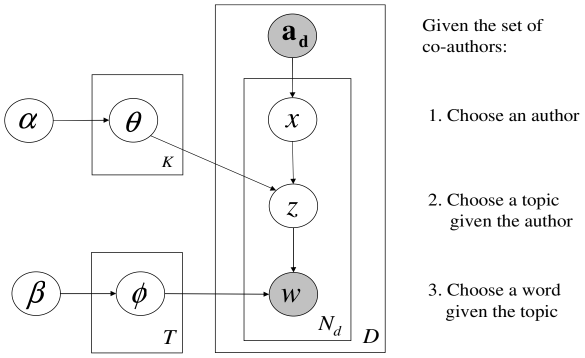

Figure 1: The graphical model for the author-topic model using plate notation.

a bag of words assumption that reduces each document to a vector of counts, where each vector element corresponds to the number of times a term appears in the document.

Each author is associated with a multinomial distribution over topics. A document with multiple authors has a distribution over topics that is a mixture of the distributions associated with the authors. When generating a document, an author is chosen at random for each individual word in the document. This author picks a topic from his or her multinomial distribution over topics, and then samples a word from the multinomial distribution over words associated with that topic. This process is repeated for all words in the document.

In the model, the authors produce words from a set of T topics. When T is kept relatively small relative to the number of authors and vocabulary size, the author-topic model applies a form of dimensionality reduction to documents; topics are learned which capture the variability in word choice across a large set of documents and authors. In our simulations, we use 300 topics (see Rosen-Zvi et al. (2004) for an exploration of different numbers of topics).

Figure 1 illustrates the generative process with a graphical model using plate notation. For readers not familiar with plate notation, shaded and unshaded variables indicate observed and latent variables respectively. An arrow indicates a conditional dependency between variables and plates (the boxes in the figure) indicate repeated sampling with the number of repetitions given by the variable in the bottom (see Buntine (1994) for an introduction). In the author-topic model, observed variables not only include the words w in a document but also the set of coauthors A_d on each document d. Currently, the model does not specify the generative process of how authors choose to collaborate. Instead, we assume the model is provided with the authorship information on every document in the collection.

Each author (from a set of K authors) is associated with a multinomial distribution over topics, represented by θ. Each topic is associated with a multinomial distribution over words, represented by ϕ. The multinomial distributions θ and ϕ have a symmetric Dirichlet prior with hyperparameters α and β (see Rosen-Zvi et al. (2004) for details). For each word in the document, we sample an author x uniformly from A_d, then sample a topic z from the multinomial distribution θ associated with author x and sample a word w from a multinomial topic distribution ϕ associated with topic z. This sampling process is repeated N times to form document d.

2.2 Bayesian Estimation of the Model Parameters

The author-topic model includes two sets of unknown parameters—the K author-topic distributions θ, and the T topic distributions ϕ—as well as the latent variables corresponding to the assignments of individual words to topics z and authors x. The Expectation-Maximization (EM) algorithm is a standard technique for estimating parameters in models with latent variables, finding a mode of the posterior distribution over parameters. However, when applied to probabilistic topic models (Hofmann, 1999), this approach is susceptible to local maxima and computationally inefficient (see Blei, Ng, and Jordan, 2003). We pursue an alternative parameter estimation strategy, outlined by Griffiths and Steyvers (2004), using Gibbs sampling, a Markov chain Monte Carlo algorithm to sample from the posterior distribution over parameters. Instead of estimating the model parameters directly, we evaluate the posterior distribution on just x and z and then use the results to infer θ and ϕ.

For each word, the topic and author assignment are sampled from:

$$P\left(z_i = j, x_i = k | w_i = m, \mathbf{z}_{-i}, \mathbf{x}_{-i}\right) \propto \frac{C^{WT}_{mj} + \beta}{\sum_{m'} C^{WT}_{m'j} + V\beta} \frac{C^{AT}_{kj} + \alpha}{\sum_{j'} C^{AT}_{kj'} + T\alpha} \quad (1)$$

where $z_i = j$ and $x_i = k$ represent the assignments of the ith word in a document to topic j and author k respectively, $w_i = m$ represents the observation that the ith word is the mth word in the lexicon, and $\mathbf{z}_{-i}, \mathbf{x}_{-i}$ represent all topic and author assignments not including the ith word. Furthermore, C^{WT}_{mj} is the number of times word m is assigned to topic j, not including the current instance, and C^{AT}_{kj} is the number of times author k is assigned to topic j, not including the current instance, and V is the size of the lexicon.

During parameter estimation, the algorithm only needs to keep track of a $V \times T$ (word by topic) count matrix, and a $K \times T$ (author by topic) count matrix, both of which can be represented efficiently in sparse format. From these count matrices, we can easily estimate the topic-word distributions ϕ and author-topic distributions θ by:

$$\phi_{mj} = \frac{C^{WT}_{mj} + \beta}{\sum_{m'} C^{WT}_{m'j} + V\beta} \quad (2)$$

$$\theta_{kj} = \frac{C^{AT}_{kj} + \alpha}{\sum_{j'} C^{AT}_{kj'} + T\alpha} \quad (3)$$

where ϕ_{mj} is the probability of using word m in topic j, and θ_{kj} is the probability of using topic j by author k. These values correspond to the predictive distributions over new words w and new topics z conditioned on w and z.

We start the algorithm by assigning words to random topics and authors (from the set of authors on the document). Each Gibbs sample then constitutes applying Equation (1) to every word token in the document collection. This sampling process is repeated for I iterations. In this paper we primarily focus on results based on a single sample so that specific topics can be identified and interpreted—in tasks involving prediction of words and authors one can average over topics and use multiple samples when doing so (Rosen-Zvi

TOPIC 95	
WORD	**PROB.**
PATTERNS	0.1965
PATTERN	0.1821
MATCHING	0.1375
MATCH	0.0337
TEXT	0.0242
PRESENT	0.0207
MATCHES	0.0167
PAPER	0.0126
SHOW	0.0124
APPROACH	0.0099
AUTHOR	**PROB.**
Navarro_G	0.0133
Amir_A	0.0099
Gasieniec_L	0.0062
Baeza-Yates_R	0.0048
Baker_B	0.0042
Arikawa_S	0.0041
Crochemore_M	0.0037
Rytter_W	0.0034
Raffinot_M	0.0032
Ukkonen_E	0.0032

TOPIC 293	
WORD	**PROB.**
USER	0.3290
INTERFACE	0.1378
USERS	0.1060
INTERFACES	0.0498
SYSTEM	0.0434
INTERACTION	0.0296
INTERACTIVE	0.0214
USABILITY	0.0132
GRAPHICAL	0.0092
PROTOTYPE	0.0086
AUTHOR	**PROB.**
Shneiderman_B	0.0051
Rauterberg_M	0.0046
Harrison_M	0.0025
Winiwarter_W	0.0024
Ardissono_L	0.0021
Billsus_D	0.0019
Catarci_T	0.0017
St_R	0.0017
Picard_R	0.0016
Zukerman_I	0.0016

TOPIC 29	
WORD	**PROB.**
MAGNETIC	0.0155
STARS	0.0145
SOLAR	0.0135
EMISSION	0.0127
MASS	0.0125
OBSERVATIONS	0.0120
STAR	0.0118
RAY	0.0112
GALAXIES	0.0105
OBSERVED	0.0098
AUTHOR	**PROB.**
Falcke_H	0.0140
Linsky_J	0.0082
Butler_R	0.0077
Knapp_G	0.0067
Bjorkman_K	0.0065
Kundu_M	0.0060
Christensen-D_J	0.0057
Mursula_K	0.0054
Cranmer_S	0.0051
Nagar_N	0.0050

TOPIC 58	
WORD	**PROB.**
METHODS	0.5319
METHOD	0.1403
TECHNIQUES	0.0442
DEVELOPED	0.0216
APPLIED	0.0162
BASED	0.0153
APPROACHES	0.0133
COMPARE	0.0113
PRACTICAL	0.0112
STANDARD	0.0102
AUTHOR	**PROB.**
Srinivasan_A	0.0018
Mooney_R	0.0018
Owren_B	0.0018
Warnow_T	0.0016
Fensel_D	0.0016
Godsill_S	0.0014
Saad_Y	0.0014
Hansen_J	0.0013
Zhang_Y	0.0013
Dietterich_T	0.0013

TOPIC 52	
WORD	**PROB.**
DATA	0.1622
MINING	0.0657
DISCOVERY	0.0408
ATTRIBUTES	0.0343
ASSOCIATION	0.0328
LARGE	0.0279
DATABASES	0.0257
KNOWLEDGE	0.0175
PATTERNS	0.0174
ITEMS	0.0173
AUTHOR	**PROB.**
Han_J	0.0164
Zaki_M	0.0089
Liu_B	0.0071
Cheung_D	0.0066
Shim_K	0.0051
Mannila_H	0.0049
Rastogi_R	0.0049
Ganti_V	0.0048
Toivonen_H	0.0043
Liu_H	0.0043

TOPIC 68	
WORD	**PROB.**
PROBABILISTIC	0.0869
BAYESIAN	0.0791
PROBABILITY	0.0740
MODEL	0.0533
MODELS	0.0466
PROBABILITIES	0.0308
INFERENCE	0.0306
CONDITIONAL	0.0274
PRIOR	0.0273
POSTERIOR	0.0228
AUTHOR	**PROB.**
Koller_D	0.0104
Heckerman_D	0.0079
Ghahramani_Z	0.0060
Friedman_N	0.0060
Myllymaki_P	0.0057
Lukasiewicz_T	0.0054
Geiger_D	0.0045
Muller_P	0.0044
Berger_J	0.0044
Xiang_Y	0.0042

TOPIC 298	
WORD	**PROB.**
RETRIEVAL	0.1208
INFORMATION	0.0613
TEXT	0.0461
DOCUMENTS	0.0385
INDEXING	0.0369
DOCUMENT	0.0316
QUERY	0.0261
CONTENT	0.0256
SEARCH	0.0174
RELEVANCE	0.0171
AUTHOR	**PROB.**
Oard_D	0.0097
Hawking_D	0.0065
Croft_W	0.0057
Jones_K	0.0053
Schauble_P	0.0052
Voorhees_E	0.0050
Callan_J	0.0046
Fuhr_N	0.0042
Smeaton_A	0.0042
Sanderson_M	0.0041

TOPIC 139	
WORD	**PROB.**
QUERY	0.1406
QUERIES	0.0947
DATABASE	0.0932
DATABASES	0.0468
DATA	0.0426
RELATIONAL	0.0384
JOIN	0.0188
PROCESSING	0.0165
SOURCES	0.0114
OPTIMIZATION	0.0110
AUTHOR	**PROB.**
Levy_A	0.0092
Naughton_J	0.0078
Suciu_D	0.0075
Raschid_L	0.0075
DeWitt_D	0.0062
Widom_J	0.0058
Abiteboul_S	0.0057
Chu_W	0.0055
Libkin_L	0.0054
Kriegel_H	0.0054

Figure 2: Eight example topics extracted from the CiteSeer database. Each is illustrated with the 10 most likely words and authors with corresponding probabilities.

et al., 2004).

3. AUTHOR-TOPICS FOR CITESEER

3.1 Learning the Model

Our collection of CiteSeer abstracts contains $D = 162,489$ abstracts with $K = 85,465$ authors. We preprocessed the text by removing all punctuation and common stop words. This led to a vocabulary size of $V = 30,799$, and a total of $11,685,514$ word tokens.

There is inevitably some noise in data of this form given that many of the fields (paper title, author names, year, abstract) were extracted automatically by CiteSeer from PDF or postscript or other document formats. We chose the simple convention of identifying authors by their first initial and second name, e.g., A_Einstein, given that multiple first initials or fully spelled first names were only available for a relatively small fraction of papers. 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 is a known limitation of working with this type of data (e.g., see Newman (2001) for further discussion). There are algorithmic techniques that could be used to automatically resolve these identity problems—however, in this paper, we don't pursue these options and instead for simplicity work with the first-initial/last-name representation of individual authors.

TOPIC 276	
WORD	**PROB.**
DATA	0.1468
MINING	0.0631
DISCOVERY	0.0396
ATTRIBUTES	0.0392
ASSOCIATION	0.0316
RULES	0.0252
PATTERNS	0.0210
LARGE	0.0207
ATTRIBUTE	0.0183
DATABASES	0.0179
AUTHOR	**PROB.**
Han_J	0.0157
Zaki_M	0.0104
Liu_B	0.0080
Cheung_D	0.0075
Hamilton_H	0.0058
Mannila_H	0.0056
Brin_S	0.0055
Ganti_V	0.0050
Liu_H	0.0050
Toivonen_H	0.0049

TOPIC 158	
WORD	**PROB.**
PROBABILISTIC	0.0826
BAYESIAN	0.0751
PROBABILITY	0.0628
MODEL	0.0364
PROBABILITIES	0.0313
INFERENCE	0.0294
MODELS	0.0273
CONDITIONAL	0.0262
DISTRIBUTION	0.0261
PRIOR	0.0259
AUTHOR	**PROB.**
Koller_D	0.0109
Heckerman_D	0.0079
Friedman_N	0.0076
Ghahramani_Z	0.0060
Lukasiewicz_T	0.0053
Myllymaki_P	0.0053
Poole_D	0.0050
Xiang_Y	0.0048
vanderGaag_L	0.0047
Berger_J	0.0040

TOPIC 213	
WORD	**PROB.**
RETRIEVAL	0.1381
INFORMATION	0.0600
INDEX	0.0529
INDEXING	0.0469
QUERY	0.0319
CONTENT	0.0299
BASED	0.0224
SEARCH	0.0219
RELEVANCE	0.0212
SIMILARITY	0.0178
AUTHOR	**PROB.**
Oard_D	0.0080
Voorhees_E	0.0053
Hawking_D	0.0053
Schauble_P	0.0051
Croft_W	0.0051
Jones_K	0.0041
Bruza_P	0.0041
Lee_D	0.0040
Smeaton_A	0.0040
Callan_J	0.0039

TOPIC 15	
WORD	**PROB.**
QUERY	0.1699
QUERIES	0.1209
JOIN	0.0258
DATA	0.0212
OPTIMIZATION	0.0171
PROCESSING	0.0162
RELATIONAL	0.0131
DATABASE	0.0128
AGGREGATION	0.0117
RESULT	0.0106
AUTHOR	**PROB.**
Naughton_J	0.0103
Suciu_D	0.0091
Levy_A	0.0080
DeWitt_D	0.0077
Wong_L	0.0071
Ross_K	0.0067
Kriegel_H	0.0055
Mumick_I	0.0054
Raschid_L	0.0053
Kossmann_D	0.0053

Figure 3: The four most similar topics to the topics in the bottom row of Figure 2, obtained from a different Markov chain run.

In our simulations, the number of topics T was fixed at 300 and the smoothing parameters α and β (Figure 1) were set at 0.16 and 0.01 respectively. We ran 5 independent Gibbs sampling chains for 2000 iterations each. On a 2GHz PC workstation, each iteration took 400 seconds, leading to a total run time on the order of several days per chain.

3.2 Author-Topic and Topic-Word Models for the CiteSeer Database

We now discuss the author-topic and topic-word distributions learned from the CiteSeer data. Figure 2 illustrates eight different topics (out of 300), obtained at the 2000th iteration of a particular Gibbs sampler run.

Each table in Figure 2 shows the 10 words that are most likely to be produced if that topic is activated, and the 10 authors who are most likely to have produced a word if it is known to have come from that topic. The words associated with each topic are quite intuitive and, indeed, quite precise in the sense of conveying a semantic summary of a particular field of research. The authors associated with each topic are also quite representative—note that the top 10 authors associated with a topic by the model are not necessarily the most well-known authors in that area, but rather are the authors who tend to produce the most words for that topic (in the CiteSeer abstracts).

The first 3 topics at the top of Figure 2, topics #163, #87 and #20 show examples of 3 quite specific and precise topics on string matching, human-computer interaction, and astronomy respectively. The bottom four topics (#205, #209, #289, and #10) are examples of topics with direct relevance to data mining—namely data mining itself, probabilistic learning, information retrieval, and database querying and indexing. The model includes several other topics related to data mining, such as predictive modeling and neural networks, as well as topics that span the full range of research areas encompassed by documents in CiteSeer. The full list is available at `http://www.datalab.uci.edu/author-topic`.

Topic #273 (top right Figure 2) provides an example of a topic that is not directly related to a specific research area.

A fraction of topics, perhaps 10 to 20%, are devoted to "non-research-specific" topics, the "glue" that makes up our research papers, including general terminology for describing methods and experiments, funding acknowledgments and parts of addresses(which inadvertently crept in to the abstracts), and so forth.

We found that the topics obtained from different Gibbs sampling runs were quite stable. For example, Figure 3 shows the 4 most similar topics to the topics in the bottom row of Figure 2, but from a different run. There is some variability in terms of ranking of specific words and authors for each topic, and in the exact values of the associated probabilities, but overall the topics match very closely.

4. APPLICATIONS OF THE AUTHOR-TOPIC MODEL TO CITESEER

4.1 Topic Trends over Time

Of the original 162,489 abstracts in our data set, estimated years of publication were provided by CiteSeer for 130, 545 of these abstracts. There is a steady (and well-known) increase year by year in the number of online documents through the 1990's. From 1999 through 2002, however, the number of documents for which the year is known drops off sharply—the years 2001 and 2002 in particular are under-represented in this set. This is due to fact that it is easier for CiteSeer to determine the date of publication of older documents, e.g., by using citations to these documents.

We used the yearly data to analyze trends in topics over time. Using the same 300 topic model described earlier, the documents were partitioned by year, and for each year all of the words were assigned to their most likely topic using the model. The fraction of words assigned to each topic for a given year was then calculated for each of the 300 topics and for each year from 1990 to 2002.

These fractions provide interesting and useful indicators of relative topic popularity in the research literature in recent years. Figure 4 shows the results of plotting several different topics. Each topic is indicated in the legend by the five most probable words in the topic. The top left plot shows a steady increase (roughly three-fold) in machine learning and data mining topics. The top right plot shows a "tale of two topics": an increase in information-retrieval coupled to an apparent decrease in natural language processing.

On the second row, on the left we see a steady decrease in two "classical" computer science topics, operating systems and programming languages. On the right, however, we see the reverse behavior, namely a corresponding substantial growth in Web-related topics.

In the third row, the left plot illustrates trends within database research: a decrease in the transaction and concurrency-related topic, query-related research holding steady over time, and a slow but steady increase in integration-related database research. The plot on the right in the third row illustrates the changing fortunes of security-related research—a decline in the early 90's but then a seemingly dramatic upward trend starting around 1995.

The lower left plot on the bottom row illustrates the somewhat noisy trends of three topics that were "hot" in the 1990's: neural networks exhibits a steady decline since the early 1990's (as machine learning has moved on to areas such as support vector machines), genetic algorithms appears to be relatively stable, and wavelets may have peaked in the 1994–98 time period.

Finally, as with any large data set there are always some surprises in store. The final figure on the bottom right shows two somewhat unexpected "topics". The first topic consists entirely of French words (in fact the model discovered 3 such French language topics). The apparent peaking of French words in the mid-1990s is likely to be an artifact of how CiteSeer preprocesses data rather than any indication of French research productivity. The lower curve corresponds to a topic consisting of largely Greek letters, presumably from more theoretically oriented papers—fans of theory may be somewhat dismayed to see that there is an apparent steady decline in the relative frequency of Greek letters in abstracts since the mid-1990s!

The time-trend results above should be interpreted with some caution. As mentioned earlier, the data for 2001 and 2002 are relatively sparse compared to earlier years. In addition, the numbers are based on a rather skewed sample (online documents obtained by the CiteSeer system for which years are known). Furthermore, the fractions per year only indicate the relative number of words assigned to a topic by the model and make no direct assessment of the quality or importance of a particular sub-area of computer science. Nonetheless, despite these caveats, the results are quite informative and indicate substantial shifts in research topics within the field of computer science.

In terms of related work, Popescul et al. (2000) investigated time trends in CiteSeer documents using a document clustering approach. 31K documents were clustered into 15 clusters based on co-citation information while the text information in the documents was not used. Our author-topic model uses the opposite approach. In effect we use the text information directly to discover topics and do not explicitly model the "author network" (although implicitly the co-author connections are used by the model). A direct quantitative comparison is difficult, but we can say that our model with 300 topics appears to produce much more noticeable and precise time-trends than the 15-cluster model.

4.2 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 our subset of the CiteSeer collection. Because our Monte Carlo algorithm requires significant processing time for 160K documents, it would be computationally inefficient to rerun the algorithm for every new document added to the collection (even though from a Bayesian inference viewpoint this is the optimal approach). Our strategy instead is to apply an efficient Monte Carlo algorithm that runs only on the word tokens 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 Equation 1 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 5 shows an example of this type of inference. Abstracts from two authors, B_Scholkopf and A_Darwiche were combined together into 1 "pseudo-abstract" and the document treated as if they had both written it. These two au-

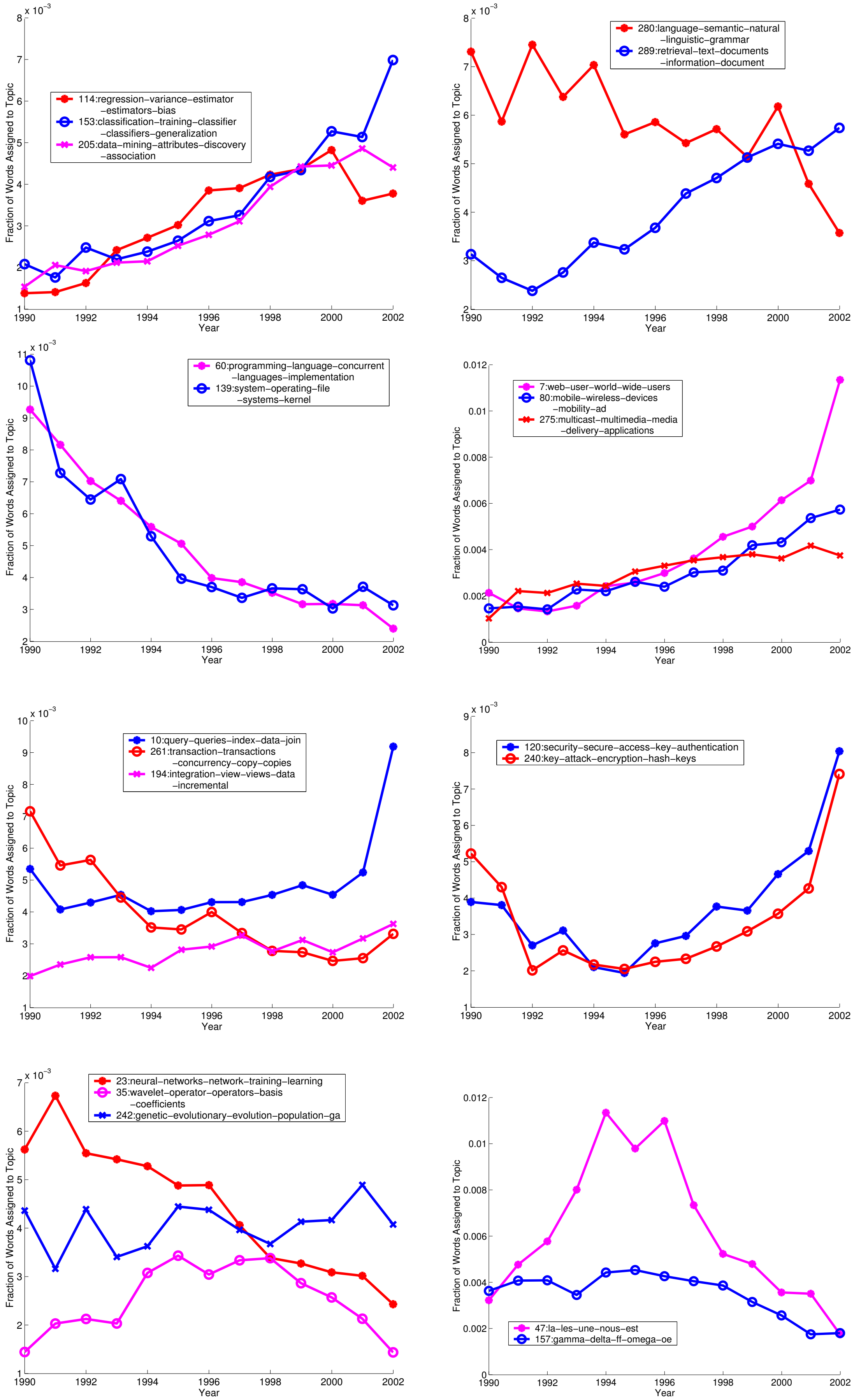

Figure 4: Topic trends for research topics in computer science.

[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 5: Automated labeling of a pseudo-abstract from two authors by the model.

thors 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.

Figure 5 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 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.

4.3 Detecting the Most Surprising and Least Surprising Papers for an Author

In Tables 1 through 3 we used the model to score papers attributed to three well-known researchers in computer science (Christos Faloutsos, Michael Jordan, and Tom Mitchell). For each document for each of these authors we calculate a perplexity score. Perplexity is widely used in language modeling to assess the predictive power of a model. It is a measure of how surprising the words are from the model's perspective, loosely equivalent to the effective branching factor. Formally, the perplexity score of a new unobserved document d that contains a set of words $\mathcal{W}_d$ and conditioned on a topic model for a specific author a is:

$$\text{Perplexity}(\mathcal{W}_d|a) = \exp\left(-\frac{\log p(\mathcal{W}_d|a)}{|\mathcal{W}_d|}\right)$$

where $p(\mathcal{W}_d|a)$ is the probability assigned by the author topic model to the words $\mathcal{W}_d$ conditioned on the single author a, and $|\mathcal{W}_d|$ is the number of words in the document. Even if the document was written by multiple authors we evaluate the perplexity score relative to a single author in order to judge perplexity relative to that individual.

Our goal here is not to evaluate the out-of-sample predictive power of the model, but to explore the range of perplexity scores that the model assigns to papers from specific authors. Lower scores imply that the words w are less surprising to the model (lower bounded by zero).In particular we are interested in the abstracts that the model considers most surprising (highest perplexity) and least surprising (lowest perplexity)—in each table we list the 2 abstracts with the highest perplexity scores, the median perplexity, and the 2 abstracts with the lowest perplexity scores.

Table 1 for Christos Faloutsos shows that the two papers with the highest perplexities have significantly higher perplexity scores than the median and the two lowest perplexity papers. The high perplexity papers are related to "query by example" and the QBIC image database system, while the low perplexity papers are on high-dimensional indexing. As far as the topic model for Faloutsos is concerned, the indexing papers are much more typical of his work than the query by example papers.

Tables 2 and 3 provide interesting examples in that the most perplexing papers (from the model's viewpoint) for each author are papers that the author did not write at all. As mentioned earlier, by combining all T_Mitchell's and M_Jordan's together, the data set may contain authors who are different from Tom Mitchell at CMU and Michael Jordan at Berkeley. Thus, the highest perplexity paper for T_Mitchell is in fact authored by a Toby Mitchell and is on the topic of estimating radiation doses (quite different from the machine learning work of Tom Mitchell). Similarly, for Michael Jordan, the most perplexing paper is on software

Table 1: Papers ranked by perplexity for C. Faloutsos, from 31 documents.

Paper Title	Perplexity Score
MindReader: Querying databases through multiple examples	1503.7
Efficient and effective querying by image content	1498.2
MEDIAN SCORE	603.5
Beyond uniformity and independence: analysis of R-trees using the concept of fractal dimension	288.9
The TV-tree: an index structure for high-dimensional data	217.2

Table 2: Papers ranked by perplexity for M. Jordan, from 33 documents.

Paper Title	Perplexity Score
Software configuration management in an object oriented database	1386.0
Are arm trajectories planned in kinematic or dynamic coordinates? An adaptation study	1319.2
MEDIAN SCORE	372.4
On convergence properties of the EM algorithm for Gaussian mixtures	180.0
Supervised learning from incomplete data via an EM approach	179.0

Table 3: Papers ranked by perplexity for T. Mitchell from 15 documents.

Paper Title	Perplexity Score
A method for estimating occupational radiation dose to individuals, using weekly dosimetry data	2002.9
Text classification from labeled and unlabeled documents using EM	845.4
MEDIAN SCORE	411.5
Learning one more thing	266.5
Explanation based learning for mobile robot perception	264.2

configuration management and was written by Mick Jordan of Sun Microsystems. In fact, of the 7 most perplexing papers for M_Jordan, 6 are on software management and the JAVA programming language, all written by Mick Jordan. However, the 2nd most perplexing paper was in fact co-authored by Michael Jordan, but in the area of modeling of motor planning, which is a far less common topic compared to the machine learning papers that Jordan typically writes.

5. AN AUTHOR-TOPIC BROWSER

We have built a JAVA-based query interface tool that supports interactive querying of the model[1]. The tool allows a user to query about authors, topics, documents, or words. For example, given a query on a particular author the tool retrieves and displays the most likely topics and their probabilities for that author, the 5 most probable words for each topic, and the document titles in the database for that author. Figure 6(a) (top panel) shows the result of querying on Pazzani_M and the resulting topic distribution (highly-ranked topics include machine learning, classification, rule-based systems, data mining, and information retrieval).

Mouse-clicking on one of the topics (e.g., the data mining topic as shown in the figure) produces the screen display to the left (Figure 6(b)). The most likely words for this topic and the most likely authors given a word from this topic are then displayed. We have found this to be a useful technique for interactively exploring topics and authors, e.g., which authors are active in a particular research area.

Similarly, one can click on a particular paper (e.g., the paper *A Learning Agent for Wireless News Access* as shown in the lower screenshot (Figure 6(c)) and the display in the panel to the right is then produced. This display shows the words in the documents and their counts, the probability distribution over topics for the paper given the word counts (ranked by highest probability first), and a probability distribution over authors, based on the proportion of words assigned by the model to each topic and author respectively.

The system is implemented using a combination of a relational database and real-time Bayesian estimation (a relatively rare combination of these technologies for a real-time query-answering system as far as we are aware). We use a database to store and index both (a) the sparse author-topic and topic-word count matrices that are learned by our algorithm from the training data, and (b) various tables describing the data such as document-word, document-author, and document-title tables. For a large document set such as CiteSeer (and with 300 topics) these tables can run into the hundred's of megabytes of memory—thus, we do not load them into main memory automatically but instead issue SQL commands to retrieve the relevant records in real-time.

For most of the queries we have implemented to date the queries can be answered by simple table lookup followed by appropriate normalization (if needed) of the stored counts to generate conditional probabilities. For example, displaying the topic distribution for a specific author is simply a matter of retrieving the appropriate record. However, when a document is the basis of a query (e.g., as in the lower screenshot, Figure 6(c)) we must compute in real-time the conditional distribution of the fraction of words assigned to each topic and author, a calculation that cannot be computed in closed form. This requires retrieving all the relevant word-topic counts for the words in the document via SQL, then executing the estimation algorithm outlined in Section 4.2 in real-time using Gibbs sampling, and displaying the results to the user. The user can change adjust the burn-in time, the number of samples and the lag time in the sampling algorithm—typically we have found that as few as 10 Gibbs samples gives quite reasonable results (and takes on the order of 1 or 2 seconds depending on the machine being used other factors).

[1]A prototype online version of the tool can be accessed at `http://www.datalab.uci.edu/author-topic`.

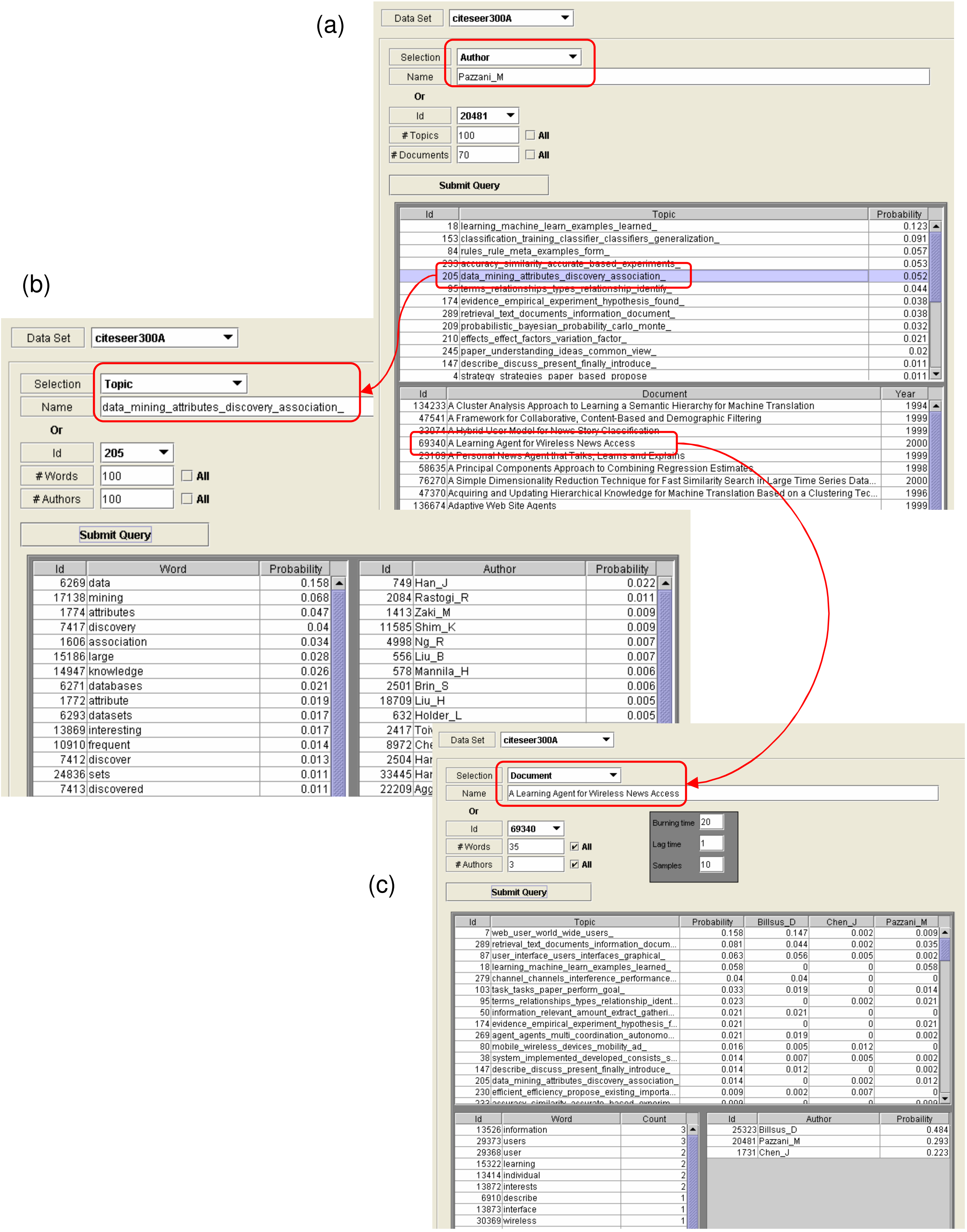

Figure 6: Examples of screenshots from the interactive query browser for the author-topic model with (a) querying on author Pazzani_M, (b) querying on a topic (data mining) relevant to that author, and (c) querying on a particular document written by the author.

6. CONCLUSIONS

We have introduced a probabilistic algorithm that can that can automatically extract information about authors, topics, and documents from large text corpora. The method uses a generative probabilistic model that links authors to observed words in documents via latent topics. We demonstrated that Bayesian estimation can be used to learn such author-topic models from very large text corpora, using CiteSeer abstracts as a working example. The resulting CiteSeer author-topic model was shown to extract substantial novel "hidden" information from the set of abstracts, including topic time-trends, author-topic relations, unusual papers for specific authors and so forth. Other potential applications not discussed here include recommending potential reviewers for a paper based on both the words in the paper and the names of the authors. Even though the underlying probabilistic model is quite simple, and ignores several aspects of real-world document generation (such as topic correlation, author interaction, and so forth), it nonetheless provides a useful first step in understanding author-topic structure in large text corpora.

Acknowledgements

We would like to thank Steve Lawrence, C. Lee Giles, and Isaac Council for providing the CiteSeer data used in this paper. We also thank Momo Alhazzazi, Amnon Meyers, and Joshua O'Madadhain for assistance in software development and data preprocessing. The research in this paper was supported in part by the National Science Foundation under Grant IRI-9703120 via the Knowledge Discovery and Dissemination (KD-D) program.

References

Blei, D. M., Ng, A. Y., and Jordan, M. I., (2003) Latent Dirichlet allocation, *Journal of Machine Learning Research* **3**, pp. 993–1022.

Buntine, W.L. (1994) Operations for learning with graphical models, *Journal of Artificial Intelligence Research* **2**, pp. 159-225.

Cutting, D., Karger, D. R., Pederson, J., and Tukey, J. W. (1992) Scatter/Gather: a cluster-based approach to browsing large document collections, in *Proceedings of the 15th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*, pp. 318–329.

Deerwester, S. C., Dumais, S. T., Landauer, T. K., Furnas, G. W., and Harshman, R. A. (1990) Indexing by latent semantic analysis, *Journal of the American Society of Information Science*, 41(6), pp. 391–407.

Diederich, J., Kindermann, J., Leopold, E., and Paass, G. (2003) Authorship attribution with support vector machines, *Applied Intelligence* **19** (1).

Erten, C., Harding, P. J., Kobourov, S. G., Wampler, K., and Yee, G. (2003) Exploring the computing literature using temporal graph visualization, Technical Report, Department of Computer Science, University of Arizona.

Gray, A., Sallis, P., MacDonell, S. (1997) Software forensics: Extending authorship analysis techniques to computer programs, *Proceedings of the 3rd Biannual Conference of the International Association of Forensic Linguists (IAFL)*, Durham NC.

Griffiths, T. L., and Steyvers , M. (2004) Finding scientific topics, *Proceedings of the National Academy of Sciences*, 101 (suppl. 1), 5228–5235.

Hofmann, T. (1999) Probabilistic latent semantic indexing, in *Proceedings of the 22nd International Conference on Research and Development in Information Retrieval* (SIGIR'99).

Kautz, H., Selman, B., and Shah, M. (1997) Referral Web: Combining social networks and collaborative filtering, *Communications of the ACM*, **3**, pp. 63–65.

Lagus, K, Honkela, T., Kaski, S., and Kohonen, T. (1999) WEBSOM for textual data mining, *Artificial Intelligence Review*, **13** (5–6), pp. 345–364.

Lawrence, S., Giles, C. L., and Bollacker, K. (1999) Digital libraries and autonomous citation indexing, *IEEE Computer*, 32(6), pp. 67–71.

McCallum, A., Nigam, K., and Ungar, L. (2000) Efficient clustering of high-dimensional data sets with application to reference matching, in *Proceedings of the Sixth ACM SIGKDD Conference on Knowledge Discovery and Data Mining*, pp. 169–178.

Mosteller, F., and Wallace, D. (1964) *Applied Bayesian and Classical Inference: The Case of the Federalist Papers*, Springer-Verlag.

Mutschke, P. (2003) 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, Springer Verlag, pp. 155–166

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

Popescul, A., Flake, G. W., Lawrence, S., Ungar, L. H., and Giles, C. L. (2000) Clustering and identifying temporal trends in document databases, *IEEE Advances in Digital Libraries*, ADL 2000, pp. 173–182.

Rosen-Zvi, M., Griffiths, T., Steyvers, M., Smyth, P. (2004) The author-topic model for authors and documents, *Proceedings of the 20th UAI Conference*, July 2004.

Thisted, B., and Efron, R. (1987) Did Shakespeare write a newly discovered poem?, *Biometrika*, pp. 445–455.

White, S. and Smyth, P. (2003) Algorithms for estimating relative importance in networks, in *Proceedings of the Ninth ACM SIGKDD Conference on Knowledge Discovery and Data Mining*, pp. 266–275.

Yang, Y. (1999) An evaluation of statistical approaches to text categorization, *Information Retrieval*, 1, pp. 69–90.