Partially Labeled Topic Models for Interpretable Text Mining

Daniel Ramage
Computer Science Dept.
Stanford University
Stanford, California
dramage@cs.stanford.edu

Christopher D. Manning
Computer Science Dept.
Stanford University
Stanford, California
manning@cs.stanford.edu

Susan Dumais
Microsoft Research
Redmond, Washington
sdumais@microsoft.com

ABSTRACT

Much of the world's electronic text is annotated with human-interpretable labels, such as tags on web pages and subject codes on academic publications. Effective text mining in this setting requires models that can flexibly account for the textual patterns that underlie the observed labels while still discovering unlabeled topics. Neither supervised classification, with its focus on label prediction, nor purely unsupervised learning, which does not model the labels explicitly, is appropriate. In this paper, we present two new *partially* supervised generative models of labeled text, Partially Labeled Dirichlet Allocation (PLDA) and the Partially Labeled Dirichlet Process (PLDP). These models make use of the unsupervised learning machinery of topic models to discover the hidden topics within each label, as well as unlabeled, corpus-wide latent topics. We explore applications with qualitative case studies of tagged web pages from del.icio.us and PhD dissertation abstracts, demonstrating improved model interpretability over traditional topic models. We use the many tags present in our del.icio.us dataset to quantitatively demonstrate the new models' higher correlation with human relatedness scores over several strong baselines.

Categories and Subject Descriptors

H.2.8 [**Database Applications**]: Data mining; I.2.7 [**Natural Language Processing**]: Text analysis

General Terms

Algorithms, Experimentation

1. INTRODUCTION

The world's text is increasingly created and consumed electronically, providing opportunities for tools to help make sense of that text. As web technologies have evolved, the amount of human-provided annotations on that text has grown, too, becoming a source of information our text mining tools should not ignore. In this paper, we address a set of related challenges presented by such datasets. How can we understand and interpret the meaning of labels and the different ways they are used? Can we model the labels while simultaneously uncovering topics in the dataset that are not labeled? Can we discover which words in documents should be attributed to which of the documents' labels? Or to none at all? We take these challenges as examples of partially supervised text mining, in which models discover textual topics in a corpus that contains some labels, although those labels may not be the primary object of study.

This form of partially supervised learning straddles the boundary between unsupervised learning, in which models discover unmarked statistical relationships in the data, and supervised learning, which emphasizes the relationship between word features and a given space of labels for the purpose of classifying new documents. Popular unsupervised approaches like Latent Dirichlet Allocation [5], Latent Semantic Indexing [11] and related methods [20, 3] are well suited for exploratory text analysis—e.g. [13]—but most of these models do not account for the label space. When they do, it is usually to improve the quality of a shared set of latent topics—such as in [19, 4, 16, 17, 34, 23]—rather than to directly model the contents of the provided labels. As a result, practitioners face challenges in interpreting what these topics really mean, how they should be named, and to what extent trends based on them can be trusted in qualitative applications. While unsupervised topics are successful in capturing broad patterns in a document collection, the learned topics do not, in general, align with human provided labels.

In contrast, supervised learning and (multi-) label prediction explicitly model the label space for the purpose of prediction (such as in [12, 18]), but by design do not discover latent sub-structure or other latent patterns. Other learning formulations exist in the space between supervised learning and unsupervised learning, most notably semi-supervised learning [10], in which the goal is to improve label classification performance by making use of unsupervised data [35]. Another, similar learning paradigm is semi-supervised clustering—such as [2, 33]—in which some supervised information is used to improve an unsupervised task. Usually this information comes in the form of human-provided pair-wise similarity/dissimilarity scores or constraints. As a result, these approaches can be used effectively for label prediction or document clustering, but do not lend themselves to more fine-grained questions about how the terms and label space interact. By contrast, the *partially supervised* approach pursued here is explicitly designed to im-

KDD'11, August 21–24, 2011, San Diego, California, USA.
Copyright 2011 ACM 978-1-4503-0813-7/11/08 ...$10.00.

prove upon the exploratory and descriptive analyses that draw practitioners to unsupervised topic models to begin with—i.e. to discover and characterize the relationships between patterns, but with the added ability to constrain those patterns to align with label classes that are meaningful to people.

Our approach to the challenges of partially supervised text mining is through methods that make use of the unsupervised learning machinery of topic modeling, but with constraints that align some learned topics with a human-provided label. In this paper, we introduce two models, unifying and generalizing the popular unsupervised topic model, Latent Dirichlet Allocation (LDA) [5] as well as the multinomial naive Bayes supervised text classifier's event model [22] and the more recent multi-label generative model Labeled LDA [27]. As in LDA, Partially Labeled Dirichlet Allocation (PLDA), assumes that each document's words are drawn from a document-specific mixture of latent topics, where each topic is represented as a distribution over words. Unlike LDA, PLDA assumes that each document can use only those topics that are in a topic class associated with one or more of the document's labels. In particular, we introduce one class (consisting of multiple topics) for each label in the label set, as well as one latent class that applies to all documents. This construction allows PLDA to discover large-scale patterns in language usage associated with each individual label, variations of linguistic usage within a label, and background topics not associated with any label. A parallel learning and inference algorithm for PLDA allows it to scale to large document collections. Our second model, the Partially Labeled Dirichlet Process (PLDP), extends PLDA by incorporating a non-parametric Dirichlet process prior over each class's topic set, allowing the model to adaptively discover how many topics belong to each label, at the expense of parallelizability. We evaluate our approach qualitatively on PhD dissertation abstracts and quantitatively on a document-similarity task derived from tagged web pages on del.icio.us.

Related work

Recently, researchers in the topic modeling community have begun to explore new ways of incorporating meta-data and hierarchy into their models, which is the approach to partially supervised text mining that we take here. For instance, Markov Random Topic Fields [16] and Markov Topic Models [32] both allow information about document groups to influence the learned topics. There has also been a great amount of work on simultaneously modeling relationships among several variables, such as authors and topics in the Author-Topic model [28], tags and words in [29], and topics sentiment in [21]. All of these models assume a latent topic space that is influenced by external label information of some form. By contrast, we use topics to model the substructure of labels and unlabeled structure around them. Other ways to constrain and exploit topic models for text mining tasks include recent work in mining product reviews such as, Titov and McDonald [31] and later Branavan, et al. [6] who extract ratable aspects of product reviews. And recently, the Nubbi model of topics and social networks [8] introduced by Chang, et al., constrains an LDA-like topic model to learn topics that correspond to individual entities (such as heads of state in Wikipedia) and the relationships between them. Topic models that account for an extra level of topic correlation have been studied as well, with notable papers such as Blei et al.'s hierarchical topic models [3] and Li and McCallum's Pachinko Allocation [20]. These types of models assume an extra hidden layer of abstraction that models topic-topic correlation. The label classes in this work can be seen as an analogous layer, but here they are supervised, hard assignments constraining only some topics to be active depending on a document's observed labels.

The work builds upon prior work the multi-label generative model, Labeled LDA, introduced by Ramage, et al. in 2009 [27], and similar models such as the extension of Rubin et al. in [29]. Like PLDA and PLDP, Labeled LDA assumes that each document is annotated with a set of observed labels, and that these labels play a direct role in generating the document's words from per-label distributions over terms. However, Labeled LDA does not assume the existence of any latent topics (neither global nor within a label)—only the documents' distributions over their observed labels, as well as those labels' distributions over words, are inferred. Labeled LDA borrows the machinery of LDA primarily for the purpose of *credit attribution*—associating which words in each document are best associated with each of the document's labels. As a result, Labeled LDA does not support latent sub-topics within a given label nor any global latent topics. In this sense, "Labeled Latent Dirichlet Allocation" is not so latent: every output dimension is in one-to-one correspondence with the input label space. In this work, we introduce two new models, PLDA and PLDP, that by incorporating classes of latent topics extend, generalizes, and unify LDA with Labeled LDA. This simple change opens new opportunities in interpretable text mining and results in a large and surprising boost in the models' ability to correlate with human similarity judgments, as we demonstrate in Section 3.3.

2. PARTIALLY SUPERVISED MODELS

In our formalization of partially supervised text mining, we are given a collection of documents $\mathbb{D}$, each containing a multi-set of words $\vec{w}_d$ from a vocabulary $\mathbb{V}$ of size V and a set of labels Λ_d from a space of labels $\mathbb{L}$. We would like to recover a set of topics Φ that fit the observed distribution of words in the multi-labeled documents, where each topic is a multinomial distribution over words $\mathbb{V}$ that tend to co-occur with each other and some label $l \in \mathbb{L}$. Latent topics that have no associated label are optionally modeled by assuming the existence of a background *latent* label $\mathbb{L}$ that is applied to all documents in the collection. In the sections below, we define PLDA and PLDP, both of which assume that the word w at position i in each document d is generated by first picking a label l from Λ_d and then a topic z from the set of topics associated with that label. Then word w is picked from the topic indexed $\Phi_{l,z}$. In this way, both PLDA and PLDP can be used for *credit attribution* of words to labels by examining the posterior probability over labels for a particular word instance. Both PLDA and PLDP are generative probabilistic graphical models, and so for each we will use an approximate inference algorithm to re-construct the per-document mixtures over labels and topics, as well as the set of words associated with each label. By incorporating the latent class of topics in addition to the label classes, the model effectively forces each word to decide if it is better modeled by a broad, latent topic, or a topic that applies specifically to one of its document's labels.

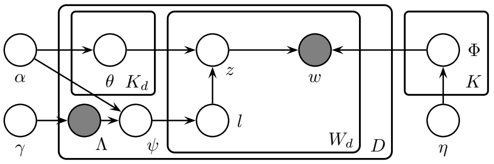


Figure 1: Graphical model for PLDA. Each document's words w and labels Λ are observed, with the per-doc label distribution ψ, per-doc-label topic distributions θ, and per-topic word distributions Φ hidden variables. Because each document's label-set Λ_d is observed, its sparse vector prior γ is unused; included for completeness.

2.1 Partially Labeled Dirichlet Allocation

Partially Labeled Dirichlet Allocation (PLDA) is a generative model for a collection of labeled documents, extending the generative story of LDA [5] to incorporate labels, and of Labeled LDA [27] to incorporate per-label latent topics. Formally, PLDA assumes the existence of a set of $\mathbb{L}$ labels (indexed by $1..L$), each of which has been assigned some number of topics $\mathbb{K}_l$ (indexed by $1..K_L$) and where each topic $\phi_{l,k}$ is represented as a multinomial distribution over all terms in the vocabulary $\mathbb{V}$ drawn from a symmetric Dirichlet prior η. One of these labels may optionally denote the shared global *latent* topic class, which can be interpreted as a label *"latent"* present on every document d. PLDA assumes that each topic takes part in exactly one label.

Figure 1 shows the Bayesian graphical model for PLDA. Each document d is generated by first drawing a document-specific subset of available label classes, represented as a sparse binary vector Λ_d from a sparse binary vector prior. A document-specific mix $\theta_{d,j}$ over topics $1..K_j$ is drawn from a symmetric Dirichlet prior α for each label $j \in \Lambda_d$ present in the document. Then, a document-specific mix of observed labels ψ_d is drawn as a multinomial of size $|\Lambda_d|$ from a Dirichlet prior $\vec{\alpha_L}$, with each element $\psi_{d,j}$ corresponding to the document's probability of using label $j \in \Lambda_d$ when selecting a latent topic for each word. For derivational simplicity, we define the element at position j of $\vec{\alpha_L}$ to be αK_j, so $\vec{\alpha_L}$ is not a free parameter. Each word w in document d is drawn from some label's topic's word distribution, i.e. it is drawn by first picking a label j from ψ_d, a topic z from $\theta_{d,l}$, and then a word w from $\phi_{l,k}$. Ultimately, this word will be picked in proportion to how much the enclosing document prefers the label l, how much that label prefers the topic z, and how much that topic prefers the word w.

We are interested in finding an efficient way to compute the joint likelihood of the observed words $\vec{w}$ with the unobserved label and topic assignments, $\vec{l}$ and $\vec{z}$, respectively: $P(\vec{w}, \vec{z}, \vec{l}|\vec{\Lambda}, \alpha, \eta, \gamma) = P(\vec{w}|\vec{z}, \eta)P(\vec{z}, \vec{l}|\vec{\Lambda}, \alpha, \gamma)$. Later, we will use this joint likelihood to derive efficient updates for the parameters Θ, Ψ, and Φ (where the capital Greek letters represent the full set of $\vec{\theta}$, $\vec{\psi}$, and $\vec{\phi}$, respectively). First, we note that the left term $P(\vec{w}|\vec{z}, \eta) = \int_\Phi P(\vec{w}|\vec{z}, \Phi)P(\Phi|\eta)d\Phi$ is the same as for standard latent Dirichlet allocation and ultimately contributes the same terms to the full conditional as well as to the sampling formula for updating individual topic assignments $z_{d,i}$, so we use the same derivation as in e.g. [13]. Using the model's independence assumptions, we consider the joint probability of the topics and labels, $P(\vec{z}, \vec{l}|\vec{\Lambda}, \alpha, \gamma) = P(\vec{z}|\vec{l}, \alpha)P(\vec{l}|\vec{\Lambda}, \gamma, \alpha)$. We will examine each half of this expression in turn. First, observe that:

$$P(\vec{z}|\vec{l}, \alpha) \quad = \quad \int_\Theta P(\vec{z}|\vec{l}, \Theta)P(\Theta|\alpha)d\Theta \tag{1}$$

$$P(\vec{z}|\vec{l}, \Theta) \quad = \quad \prod_{d=1}^{D}\prod_{i=1}^{W_d} P(z_{d,i}|l_{d,i}, \theta_{d,l_{d,i}}) \tag{2}$$

$$= \quad \prod_{d=1}^{D}\prod_{i=1}^{W_d} \theta_{d,l_{d,i},z_{d,i}} = \prod_{d=1}^{D}\prod_{j\in\Lambda_d}\prod_{k=1}^{K_j}(\theta_{d,j,k})^{n_{d,j,k,\cdot}}$$

Here we introduce $n_{d,j,k,t}$ as the number of occurrences of label $j \in \Lambda_d$ topic $k \in \mathbb{K}_j$ within document d as applied to term $t \in \mathbb{V}$. In this notation, we sum counts using "·" and select a vector of counts using ":", so for example $n_{d,j,k,\cdot}$ refers to $\sum_{t=1}^{V} n_{d,j,k,t}$ or the number of occurrences of label j and topic k in document d. Similarly, $n_{d,j,:,\cdot}$selects the vector of size K_j with the term at position k equal to $n_{d,j,k,\cdot}$. After multiplying by θ's Dirichlet prior and applying the standard Dirichlet-multinomial integral, we see that $P(\vec{z}|\vec{l}, \alpha) = \prod_{d=1}^{D}\prod_{j\in\Lambda_d}\frac{\Delta(n_{d,j,:,\cdot}+\vec{\alpha})}{\Delta(\vec{\alpha})}$ making use of the notation in [14] where $\Delta(\vec{x}) = \frac{\prod_{k=1}^{\dim \vec{x}}\Gamma(x_k)}{\Gamma(\sum_{k=1}^{\dim \vec{x}} x_k)}$ and we treat $\vec{\alpha}$ as a vector of size K_j with each value equal to α. Note that because each label has its own distinct subset of topics, the topic assignment alone is sufficient to determine which label was assigned, so there is no need to represent $\vec{l}$ explicitly in order to compute $n_{d,j,:,\cdot}$.

Now let's return to the computation of $P(\vec{l}|\vec{\Lambda}, \gamma, \alpha)$, which, because $\vec{\Lambda}$ is considered observed, can be factorized into: $P(\vec{l}|\vec{\Lambda}, \Psi)P(\Psi|\alpha, \vec{\Lambda}) =$
$\int_\Phi \prod_{d=1}^{D} P(\psi_d|\alpha, \Lambda_d)\prod_{i=1}^{W_d} P(l_{d,i}|\Lambda_d, \psi_d)d\Phi$
By re-indexing over label types, and applying the standard Dirichlet prior and Dirichlet-multinomial integral to get our final probability $P(\vec{l}|\vec{\Lambda}, \vec{\alpha_L}) = \prod_{d=1}^{D}\prod_{j\in\Lambda_d}\frac{\Delta(n_{d,:,\cdot,\cdot}+\vec{\alpha_L})}{\Delta(\vec{\alpha_L})}$.

Observing that the actual values of $\vec{l}$ are never used explicitly, and because every topic takes part in only a single label, we can represent the model using a Gibbs sampler tracking only the topic assignments $\vec{z}$. We do not need to allocate memory to represent which label $\vec{l}$ is assigned to each token. After combining terms, applying Bayes rule, and folding terms into the proportionality constant, the sampling update formula for assigning a new label and topic to a word token is defined as follows:
$P(l_{d,i} = j, z_{d,i} = k|l_{\neg d,i}, z_{\neg d,i}, w_{d,i} = t; \alpha, \eta)$

$$\propto \quad I[j \in \Lambda_d \wedge k \in 1..K_j]\left(\frac{n_{\cdot,j,k,t}^{(\neg d,i)} + \eta}{n_{\cdot,j,k,\cdot}^{(\neg d,i)} + V\eta}\right) \cdot \tag{3}$$

$$\left(\frac{n_{d,j,\cdot,\cdot}^{(\neg d,i)} + (\vec{\alpha_L})_j}{n_{d,\cdot,\cdot,\cdot}^{(\neg d,i)} + \sum_{j'\in\Lambda_d}(\vec{\alpha_L})_j}\right) \cdot \left(\frac{n_{d,j,k,\cdot}^{(\neg d,i)} + \alpha}{n_{d,j,\cdot,\cdot}^{(\neg d,i)} + K_j\alpha}\right)$$

$$\propto \quad I[j \in \Lambda_d \wedge k \in 1..K_j]\left(\frac{n_{\cdot,j,k,t}^{(\neg d,i)} + \eta}{n_{\cdot,j,k,\cdot}^{(\neg d,i)} + V\eta}\right)\left(n_{d,j,k,\cdot}^{(\neg d,i)} + \alpha\right)$$

The notation $n^{(\neg d,i)}$ refers to the corresponding count excluding the current assignment of topic z and label l in document d position i. Here we have used the definition of $\vec{\alpha_L}$ at position j is αK_j, which allows the numerator in the second fraction to cancel the denominator in the last term. Because the denominator in the second fraction is independent of the topic and label assignment, it is folded into the proportionality constant. Interestingly, this sampler's update rule is like that of Latent Dirichlet Allocation [13] with the intuitive restriction that only those topics corresponding to the document's labels may be sampled.

The similarity of the model and the resulting sampling equations suggests some interesting contrasts to existing models. In particular, if we use PLDA in a purely unsupervised setting with no labels beyond the *latent* label class of k topics, the model reduces exactly to traditional LDA. At the other extreme, if every document has only a single label, if we have no *latent* topic class, and if we give each label's class a single topic, our model's per-class learning function becomes the same count and divide of terms within a class as used in the multinomial naive Bayes model [22]. Similarly, if we have no *latent* topic class, and if we give each label access to only a single topic by setting $K_l = 1$ for all labels l, then the model reduces to Labeled LDA [27]. Interestingly, Labeled LDA can be used to approximate PLDA by the construction of a synthetic label space where, for any given label l, we construct a class of labels of size K_l as labels "l-1 l-2 l-3 ... l-K_l" with all those labels applied to every document with label l. In this case, Labeled LDA will output multiple versions of the same label which, if symmetry is broken during initialization, may result in topics that look like our latent sub-labels in PLDA but has no theoretical guarantees as such. This construction was applied to microblogging data from Twitter with in [26] to good effect, seeding the development of the models in this paper.

Learning and Inference. An efficient Gibbs sampling algorithm can be developed for estimating the hidden parameters in PLDA based on the collapsed sampling formula in Equation 3. Efficient computation of the counts n can be done by keeping histograms over the number of times each term has been associated with each topic within each document and how often each topic has been associated with each term. The sampler loops over the corpus, re-assigning topic assignment variables z and updating the corresponding histograms. However, Gibbs sampling is inherently sequential and we would like this model scale to the size of modern web collections, so we developed a parallelizable learning and inference algorithm for PLDA based on the CVB0 variational approximation to the LDA objective as described in [1]. For each word at position i in each post d, we store a distribution $\gamma_{d,i}$ over the likelihood that each label and topic generated that word in that document using the normalized probabilities from the Gibbs sampling update formula in Equation 3. These distributions are summed into fractional counts of how often each word is paired with each topic and label globally, denoted $\#_{j,k,w}$, and how often each label appears in an each document, denoted $\#_{d,j,k}$. The algorithm alternates between assigning values to $\gamma_{d,i,j,k}$ and then summing assignments in a counts phase. The update equations are listed below. Initially, we use small random values to initialize $\#_{j,k,w}$ and $\#_{d,j,k}$.

Assign:

$$\gamma_{d,i,j,k} \propto I[j \in \Lambda_d, k \in 1..K_j] \cdot \tfrac{\#_{j,k,w} - \gamma_{d,i,j,k} + \eta}{\#_{j,k} - \gamma_{d,i,j,k} + W\eta} \cdot (\#_{d,j,k} - \gamma_{d,i,j,k} + \alpha)$$

Count:

$$\begin{aligned} \#_{d,j,k} &= \textstyle\sum_i \gamma_{d,i,j,k} \\ \#_{j,k,w} &= \textstyle\sum_{d,i} \gamma_{d,i,j,k} \cdot I[w_{d,i} = w] \\ \#_{j,k} &= \textstyle\sum_w \#_{j,k,w} \end{aligned}$$

The references to $\gamma_{d,i,j,k}$ on the right side of the proportionality in the assignment phase refer to the value at the previous iteration. This formulation allows for a data-parallel implementation, by distributing documents across a cluster of compute nodes. Assignments are done in parallel on all nodes based on the previous counts $\#_{d,j,k}$, $\#_{j,k,w}$ and $\#_{j,k}$ (initially small random values). The resulting assignments $\gamma_{d,i,j,k}$ are then summed in parallel across all compute nodes in a tree sum, before being distributed to all compute nodes for a new assignments phase. The process repeats until convergence. Like in [1], we find that the CVB0 learning and inference algorithm converges more quickly than the Gibbs sampler to a solution of comparable quality. In practice, we find that this algorithm scales to very large datasets—experiments on a corpus of one million PhD dissertation abstracts resulted in models that trained in less than a day on a cluster of twelve 4-core machines.

2.2 Partially Labeled Dirichlet Process

PLDA provides a great deal of flexibility in defining the space of latent topics to effectively learn latent topics both within labels and in a common background space. Unfortunately, PLDA introduces an important new parameter for each label, K_l, representing the number of topics available within each label's topic class. Fortunately, non-parametric statistical techniques can help estimate an appropriate size for each per-label topic set automatically. Concretely, we replace PLDA's per-label topic mixture θ_l with a Dirichlet process mixture model [24], which can be seen as the infinite limit of the finite mixture of topics per label used in PLDA. Formally, PLDP assumes a generative process similar to PLDA, with a multi-set of words $\vec{w}_d$ for each document and an observed set of labels Λ_d. Like in PLDA, each word $w_{d,i}$ has an associated label variable $l_{d,i}$ and topic variable $z_{d,i}$. Here, the label $l_{d,i}$ is drawn from a document-specific multinomial over labels, which for efficiency we assume is drawn from a symmetric Dirichlet prior with parameter α. To generate a topic assignment $z_{d,i}$, PLDP picks an existing topic within label $l_{d,i}$ for word $w_{d,i}$ in proportion to how often it is used, or generates a new topic with held-out mass parameter α (the same as the Dirichlet prior for the document-specific multinomial over labels). The word $w_{d,i}$, is then generated according to the topic distribution $\phi_{l_{d,i} z_{d,i}}$ as in PLDA. The Gibbs sampling formula for updating the joint label and topic assignment $l_{d,i}$ and $z_{d,i}$ in PLDP is:

$$\begin{aligned} &P(l_{d,i} = j, z_{d,i} = k | l_{\neg d,i}, z_{\neg d,i}, w_{d,i} = t; \alpha, \eta) \\ &\propto \quad I[j \in \Lambda_d] \cdot \left(\frac{n^{(\neg d,i)}_{\cdot,j,k,t} + \eta}{n^{(\neg d,i)}_{\cdot,j,k,\cdot} + V\eta} \right) \left(\frac{n^{(\neg d,i)}_{d,j,\cdot,\cdot} + \alpha}{n^{(\neg d,i)}_{d,\cdot,\cdot,\cdot} + \alpha|\Lambda_d|} \right) \qquad (4) \\ &\qquad \cdot \begin{cases} \frac{n^{(\neg d,i)}_{d,j,k,\cdot}}{n^{(\neg d,i)}_{d,j,\cdot,\cdot} + \alpha} & \text{for } k \text{ existing} \\ \frac{\alpha}{n^{(\neg d,i)}_{d,j,\cdot,\cdot} + \alpha} & \text{for } k \text{ new} \end{cases} \\ &\propto \quad I[j \in \Lambda_d] \cdot \left(\frac{n^{(\neg d,i)}_{\cdot,j,k,t} + \eta}{n^{(\neg d,i)}_{\cdot,j,k,\cdot} + V\eta} \right) \cdot \begin{cases} n^{(\neg d,i)}_{d,j,k,\cdot} & \text{for } k \text{ existing} \\ \alpha & \text{for } k \text{ new} \end{cases} \end{aligned}$$

As in the Gibbs expression for PLDA in Equation 3, we cancel the numerator in the second fraction with the denominator in both versions of the final term. Again, the denominator in the second fraction is independent of label and topic assignments, so it is folded into the proportionality constant. The Gibbs re-assignment parameters in Equation 4, paired with data structures updated to reflect the appropriate counts of interest at reassignment, can be used to create an efficient Gibbs sampling algorithm for the Partially Labeled Dirichlet Process. Unfortunately, the embedded Dirichlet process mixture model complicates the parallelizability of learning and inference in this model.

It is worth noting that PLDP's embedding of the Dirichlet Process is, in some ways, an even more natural fit than in standard topic modeling applications such as the Hierarchical Dirichlet Process [30]. HDPs and related models discover a global set of latent topics within a corpus as a function of both the concentration parameter α and the corpus being analyzed. So for a known corpus of interest, text mining practitioners still have a single parameter to choose—instead of picking the number of topics, they pick a concentration parameter. In practice, this is often no easier than picking the number of topics directly. In contrast, for PLDP, a single DP concentration parameter α selects the number of topics for each label in $\mathbb{L}$, effectively reducing the number of model parameters related to topic cardinality from $|\mathbb{L}|$ to one, α.

3. CASE STUDIES

We illustrate applications of PLDA and PLDP to partially supervised text mining tasks on two kinds of labeled corpora with very different distributional properties: PhD dissertation abstracts annotated with subject code designations and tagged web pages from del.icio.us. Our PhD dissertation dataset contains over 1 million United States PhD dissertation abstracts from the ProQuest UMI database[1], averaging about just over 2 subject codes from a controlled vocabulary of roughly 260 codes curated by ProQuest staff. These subject codes correspond to high-level field designations such as biochemistry, public administration, cultural anthropology, etc. Each document contains 179 non-stop words on average, corresponding to about two paragraphs of text from each abstract. Our del.icio.us dataset is a subset of 3,200 popular, heavily tagged documents from the Stanford Tag Crawl Dataset [15] collected in the Summer of 2007, with an average length of 1263 words from a word vocabulary of 321,062 terms, and an average of 122 distinct tags out of a vocabulary of 344,540 tags.

These datasets have very different distributional statistics, both in terms of the underlying texts and the label spaces. The del.icio.us documents are longer and have high overlap in common tags, whereas the dissertations tend to be shorter and carefully filed in a small number of subjects. In the following subsections, we examine these datasets from the partially supervised text mining perspective, finding that, despite their differences, both datasets can be effectively modeled. Because of the size of the dissertation dataset in the case study below, we focus on qualitative results that can be achieved through our parallelized PLDA model. Because of the smaller size of the del.icio.us data, we use the del.icio.us case study to quantify our intuitions about the model's ability to approach text mining challenges and compare PLDA with PLDP. Where not otherwise specified, we used fixed hyperparameters of 0.1 for α and η.

3.1 PhD Dissertation Abstracts

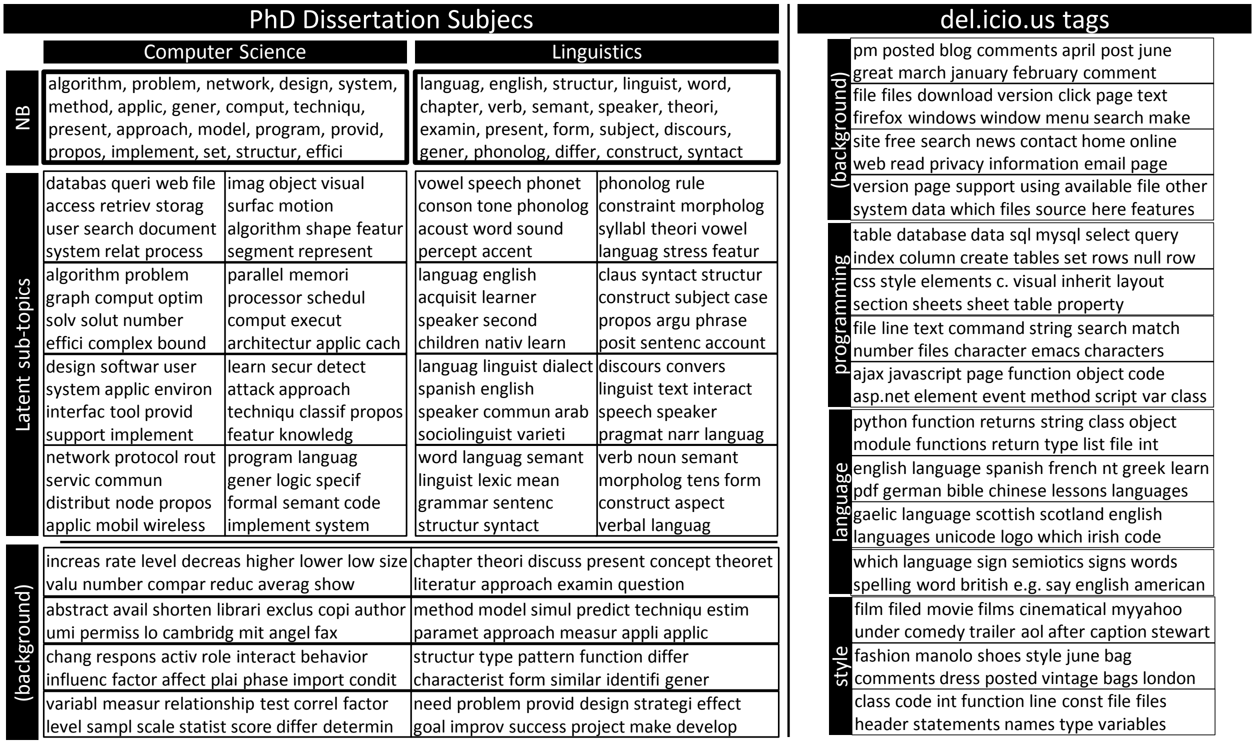
Traditional digital libraries often annotate documents with a controlled vocabulary maintained by domain experts to ease indexing, searching, and browsing. While these collections represent a shrinking fraction of all the world's electronic text, they do contain some of the most focused and important content within a limited domain. One such collection is the UMI database of PhD dissertation abstracts maintained by ProQuest, the official archival agency for dissertations written in the United States as designated by the US Library of Congress. In collaboration with social scientists in Stanford's School of Education, we collected 1,023,084 PhD dissertation abstracts from the Proquest UMI database filed by US students since 1980 from any of 151 schools classified as research-intensive by the Carnegie Foundation since 1994.[2] These dissertations are an excellent basis for the study of the history academia because they reflect the entire academic output of universities, as seen through their graduating students, and do not reflect the coverage biases toward scientific or engineering publications found in most databases of academic publications, such as ISI (with a biomedicine) and CiteSeer (with computer science). A more complete study of this dataset can be found in [25] and warrants more space than is available here, so we use the PhD dissertation database as a concrete illustration of the advantages of partially supervised topic models in an active computational social science collaboration.

While the subject codes in our data cover the full range of academic fields, they are not evenly distributed in usage, reflecting real differences in field sizes. Indeed, the most common subject code in our dataset (electrical engineering) has 44,551 instances, whereas the least common (African literature) has only 1,041. Models like PLDA are a natural fit for analyzing these controlled-vocabulary document collections due to their ability to model both the text content in terms of latent usages of the known indexing vocabulary. By contrast, latent topics on this dataset collapse distinctions between small fields (folding them into a single topic) and overly emphasize the importance of larger ones, just based on the amount of support in the data. For example, one run of LDA on this dataset—using 100 latent topics—associated topics to fields in proportion to their prevalence in the data: electrical engineering was assigned three topics, whereas African literature was split between one topic related to all forms of race culture in America ("american, black, white, ethnic, african") and another on all forms of literature ("literari, novel, narr, text, writer"). By seeing which subject codes appeared in each topic, we can see that these two topics are themselves dominated by larger subjects: anthropology and political science for the former and modern and classical literature for the latter. This result is reasonable from the perspective of how much support there is for topics in the dataset. But by losing smaller subject codes in the tails of larger topics, we lose the ability to describe topic dimensions in terms of the known, human interpretable objects of study (fields) while simultaneously losing all latent sub-structure within each field.

As a modeling alternative, we could train an independent topic model on all dissertations in each subject code. However, almost all dissertations have more than one subject code. As a result, many words in the corpus will be double-counted whereas PLDA will attribute each word in each dissertation to the appropriate subject code's latent topics. As a modeling framework, PLDA further allows for the automatic construction of shared latent background topics that

[1] http://www.proquest.com/en-US/products/dissertations/

[2] http://classifications.carnegiefoundation.org/resources/



PhD Dissertation Subjecs

	Computer Science		Linguistics	
NB	algorithm, problem, network, design, system, method, applic, gener, comput, techniqu, present, approach, model, program, provid, propos, implement, set, structur, effici		languag, english, structur, linguist, word, chapter, verb, semant, speaker, theori, examin, present, form, subject, discours, gener, phonolog, differ, construct, syntact	
Latent sub-topics	databas queri web file access retriev storag user search document system relat process	imag object visual surfac motion algorithm shape featur segment represent	vowel speech phonet conson tone phonolog acoust word sound percept accent	phonolog rule constraint morpholog syllabl theori vowel languag stress featur
	algorithm problem graph comput optim solv solut number effici complex bound	parallel memori processor schedul comput execut architectur applic cach	languag english acquisit learner speaker second children nativ learn	claus syntact structur construct subject case propos argu phrase posit sentenc account
	design softwar user system applic environ interfac tool provid support implement	learn secur detect attack approach techniqu classif propos featur knowledg	languag linguist dialect spanish english speaker commun arab sociolinguist varieti	discours convers linguist text interact speech speaker pragmat narr languag
	network protocol rout servic commun distribut node propos applic mobil wireless	program languag gener logic specif formal semant code implement system	word languag semant linguist lexic mean grammar sentenc structur syntact	verb noun semant morpholog tens form construct aspect verbal languag
(background)	increas rate level decreas higher lower low size valu number compar reduc averag show		chapter theori discuss present concept theoret literatur approach examin question	
	abstract avail shorten librari exclus copi author umi permiss lo cambridg mit angel fax		method model simul predict techniqu estim paramet approach measur appli applic	
	chang respons activ role interact behavior influenc factor affect plai phase import condit		structur type pattern function differ characterist form similar identifi gener	
	variabl measur relationship test correl factor level sampl scale statist score differ determin		need problem provid design strategi effect goal improv success project make develop	

del.icio.us tags

(background)	pm posted blog comments april post june great march january february comment
	file files download version click page text firefox windows window menu search make
	site free search news contact home online web read privacy information email page
	version page support using available file other system data which files source here features
programming	table database data sql mysql select query index column create tables set rows null row
	css style elements c. visual inherit layout section sheets sheet table property
	file line text command string search match number files character emacs characters
	ajax javascript page function object code asp.net element event method script var class
language	python function returns string class object module functions return type list file int
	english language spanish french nt greek learn pdf german bible chinese lessons languages
	gaelic language scottish scotland english languages unicode logo which irish code
	which language sign semiotics signs words spelling word british e.g. say english american
style	film filed movie films cinematical myyahoo under comedy trailer aol after caption stewart
	fashion manolo shoes style june bag comments dress posted vintage bags london
	class code int function line const file files header statements names type variables

Figure 2: PLDA output on dissertation abstracts (left) and del.icio.us tags (right). Computer Science and Linguistics are two subject codes. "NB" (upper, left) refers to the naive Bayes term estimates associated with each respective code, contrasted with the latent topics learned within each. The "(background)" class (for del.icio.us in upper right, dissertations in lower-left) is the latent topic class shared by all documents in the respective collection.

extract common words found in most abstracts across all per-field topics. The background topics in PLDA are explicitly labeled as such by the model and so do not need to manually identified as they would if each subject code had an independently trained model.

Examples of the topics learned by PLDA are in Figure 2. At the top, we see the most common words associated with each subject code by the simple count-and-divide multinomial naive Bayes estimate, as well as latent sub-topics discovered for each subject code. We used a distributed implementation of PLDA to learn a model with eight global latent background topics and eight latent topics per subject area, resulting in a total of 2,080 latent topics. The results shown are representative of the quality of discovered topics across all academic disciplines. Note that the major distinctions within each subject code roughly correspond to the broad areas of study within computer science and linguistics. The latent topics capture shared common structure in PhD dissertations,[3] including basic things such as variables that increase or decrease, rates of change, and structural starting points about needs, problems, and goals.

A high quality topic space with labeled groups of latent dimensions, as output by PLDA, can be used to ask and answer questions about the nature of academia. For example, we can re-run inference on any given dissertation (while allowing it to use all subject labels) in order to compute a per-dissertation distribution over fields of study—which words might the dissertation have borrowed from which other fields. This can tell us which dissertations within an area of interest are more or less interdisciplinary. Did computational biology get an earlier head start at public or private institutions? Or, at a larger-scale, we can ask which schools tend to have the most inter-disciplinary research in general, or even whether interdisciplinary dissertations are more likely to result in productive future research careers. As always, external, dataset-specific validation metrics need to be in place in order to trust the output of such analyses.

Although this section is merely descriptive, we hope it serves to illustrate the practical impact that human-interpretable topic dimensions can bring to text mining practitioners and computational social scientists. In the next section, we examine content from the social bookmarking website del.icio.us, and use that dataset's abundance of tags as the basis for extrinsic comparison between models.

3.2 Tagged web pages

Users of social bookmarking websites like del.icio.us bookmark the pages they encounter with single word tags [7]. In contrast to more traditional supervised learning problems, user-generated tags are not predetermined nor applied uniformly to all items. For example, the tag *language* on the social bookmarking site http://del.icio.us/ might be applied to web pages about human languages or programming languages. We call these variations in usage of the same tag *sub-tags*. The right half of Figure 2 summarizes some of the types of trends discovered within each tag on del.icio.us. The model was run on a randomly selected 3,200 tagged

[3] Note that common stopwords and very rare words were removed before training. Terms were stemmed using a Porter stemmer to further reduce the vocabulary size for memory efficiency.

Table 1: HTJS within a tag (left) and within sub-tags (right). % change is relative to the .0183 score for randomly selected documents.

	Docs by tag		Docs by sub-tag	
Tag	**HTJS**	**Change**	**HTJS**	**Change**
books	.0254	39%	.1292	605%
computer	.0362	97%	.1609	777%
culture	.0259	41%	.0780	326%
design	.0269	47%	.0510	178%
education	.0206	12%	.1784	873%
english	.0263	44%	.0531	189%
language	.0314	71%	.1996	989%
style	.0290	58%	.2244	1124%
Overall	**.0273**	**49%**	**.1191**	**550%**

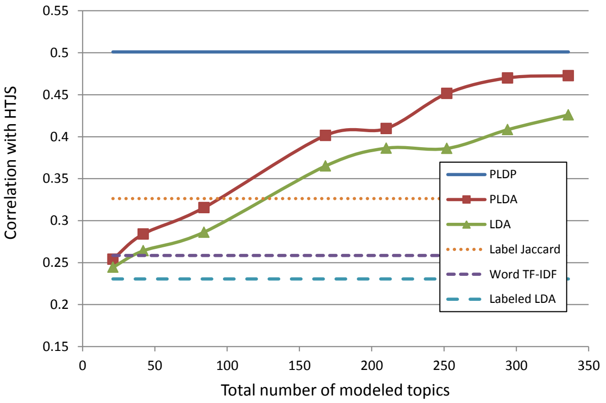


Figure 3: Correlation with HTJS for varying numbers of topic dimensions (PLDA, LDA) or as decided by model form (PLDP, Labeled LDA). Higher is better.

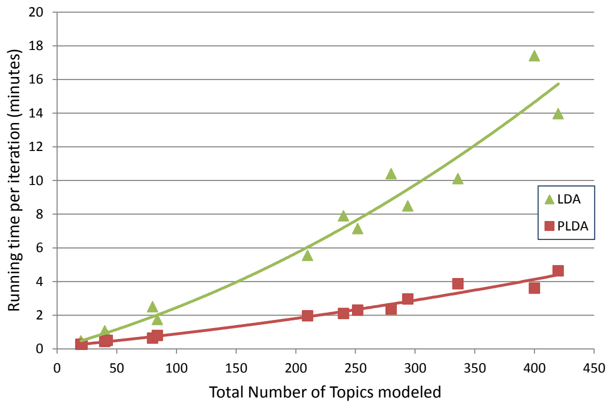


Figure 4: Average training time per iteration (in minutes) while varying the total number of latent topics for LDA (top) and PLDA (bottom) on tagged web pages. PLDA is substantially faster than LDA at a comparable number of topics because of the sparsity inherent to PLDA's sampling distribution.

web pages from [15], using 20 tags hand-selected to be relatively common but also broad in scope: reference, design, programming, internet, computer, web, java, writing, english, grammar, style, language, books, education, philosophy, politics, religion, science, history and culture. We used five latent topics and five topics for each tag. Qualitatively, the figure illustrates the model's ability to discover meaningful sub-tags, even some with a common meaning.

Because the model was trained on only a subset of all tags, we can use the remaining tags as a form of extrinsic model evaluation for computing the correlation of our model's output with a surrogate human relatedness judgment. Such an evaluation is preferable to the standard perplexity-based evaluations common in topic modeling, which have been shown to disagree with human judgments of topic quality, such as in [9]. Here, we refer to the tags not explicitly modeled as held-out tags. In our experiments, most tags are held-out (128 / 132 per document, on average). Because two related documents are more likely to be tagged the same way, overlap between their held-out tags is a natural surrogate gold-standard metric for those pages' relatedness. Formally, we measure the relatedness of a pair of documents d_1 and d_2 as their *held-out tag Jaccard score* (HTJS), defined to be the Jaccard coefficient of overlap in their held out tag sets, $G(d_1)$ and $G(d_2)$, respectively: $HTJS(d_1, d_2) = \frac{|G(d_1) \cap G(d_2)|}{|G(d_1) \cup G(d_2)|}$. To measure the average relatedness within a group of documents, we randomly select k pairs of distinct documents from within the group, with replacement. Here we set $k = 500$, finding little deviation in a set's scores across different random initializations and finding no significant impact from increasing k.

HTJS is a sensible basis for evaluating the effectiveness of our model at capturing latent sub-structure in the data. We computed HTJS on a random subset of documents in our dataset, finding the average score to be 0.0183, showing little overlap in tags of randomly chosen pages, as expected. We expect that pairs of documents that are both tagged with t will have higher held-out tag similarity than the baseline. Columns 2 and 3 in Table 1 show the improvement in HTJS scores from some of the 20 modeled tags in the dataset. Indeed, we find that documents tagged with *computer* (which is not a held-out tag) have an average HTJS score of .0362, a 97% increase over the set of all documents (.0183). On average, grouping by tag increases HTJS scores by 49%, in line with our expectation that knowing the document's tag tells us something about its other tags.

We can further utilize HTJS to quantify our model's ability to isolate coherent sub-tags within a tag. The HTJS for sub-tag s of tag t is computed on all documents labeled with tag t that use sub-tag s with at least as much probability as the sum of the other sub-tags of t. For example, the HTJS of the documents using *computer*'s first sub-tag ("security news may version update network mac") scores as high as 0.312, improving the HTJS of just knowing computer by another 31%. The right-most columns in Table 1 report the HTJS averaged across all sub-tags of the tag named in the left-most column. Not all documents tagged with t will necessarily participate in one of these subsets, as not all documents will be guaranteed to be strongly biased toward one sub-tag. The large improvements shown in Table 1 (550% relative to the baseline and 336% relative to the single tag) demonstrate PLDA's ability to model coherent sub-usages of tags.

3.3 Model comparison by HTJS Correlation

In this section we use HTJS to compare PLDA and PLDP to several strong baselines. Better performing models should have better agreement with HTJS similarity scores across a wide range of document pairs. We quantify this intuition with Pearson's correlation coefficient: for any given model, we compute the correlation of similarity scores generated by the model with HTJS scores over 5,000 randomly selected document pairs. Higher correlations mean that the similarity score implied by the model better aligns with our surrogate human judgments.

Figure 3 shows the correlation of PLDP, PLDA, LDA, Labeled LDA, and tf-idf cosine similarity with HTJS scores as the total number of latent topics changes. The way we compute similarity scores depends on the model form: the partially supervised models introduced here, like other topic models, project documents into a lower dimensional topic space through their per-document topic loadings. In the case of standard topic models such as LDA, this loading is just the per-document topic distribution θ. For PLDA and PLDP, we take a document's "θ" to be the concatenation of the documents' topic loading on all labels (even those not present in the document), resulting in a vector that is dense for topics corresponding to the document's labels and zero elsewhere. Values of these signature vectors are compared using cosine similarity, which we have found to be a stable and high performing metric in this context.

We also included two baselines: tf-idf cosine similarity (in word space) and the Jaccard score of the modeled (i.e. not held-out) tags. For all models, we used fixed hyperparameters of $\alpha = .01$ and $\eta = .01$. Along the x-axis is the total number of latent topics used by PLDA (varying the number of topics allocated per class from 1 to 16) and of LDA. Labeled LDA has a horizontal line corresponding to using 20 topics, one per class (and no latent class) and performs substantially worse than the other models because of its inability to model the sub-structure of each tag. PLDP demonstrates a higher correlation with the HTJS scores across the whole dataset by adapting to the label and word distributions in the data. PLDP's embedded Dirichlet process allows it to allocate different numbers of topics to each tag as a function of its concentration parameter α. Here, our PLDP model allocated 293 topics with substantial probability mass (and several hundred more occurring with very low frequency). These topics were allocated differentially according to the frequency of each tag and the variety of ways in which it is used—most were given to the latent class and common tags such as design, politics, and internet. Only four topics were allocated to the least common tag in the dataset (*grammar*). We experimented with several values of α for PLDP, resulting in more or fewer topics, but with similar ratios of topics allocated to each tag and similar (but not always superior) overall performance results.

As the number of topics grows, the performance of PLDA approaches that of PLDP, but with substantial computational advantages. In particular, our PLDA implementation can be parallelized in a straightforward manner and PLDA does not have the additional overhead of constructing (and possibly pruning) new topics. On average, our PLDA implementation is between 5 and 20 times faster per iteration than our (much less optimized) PLDP implementation, and takes fewer iterations to converge.

4. SCALABILITY

The expense of adding more label classes is directly proportional to how many documents each label participates in, and is always faster than modeling more global latent topics. Indeed, the impact of a label l's topics on running time appears only in computing the sampling proportions in documents with $l \in \Lambda_d$. This allows PLDA models such as those trained on the PhD dissertation dataset to scale to very large topic spaces and in an appreciably shorter period of time—indeed, training our 8 topics-per-subject PLDA model on one million abstracts ran in under a day on a small cluster of multi-core computers. Training a comparable number of latent topics (2,080) on this dataset took, on average 82 times longer per iteration and more iterations to converge. Like LDA, the running time of PLDA (for a fixed number of iterations) is linear in the size of the input data.

On collections with more common labels that have a higher degree of overlap, such as del.icio.us, incorporating more label classes or topics per class increases the computational load, but at a rate much slower than the cost of adding more global shared latent topics, as most tags are not applied to most documents. Figure 4 shows the running time per iteration (in minutes) for the collapsed variational Bayes learning algorithm on roughly twelve thousand documents from del.icio.us as the effective number of topics increases (using the same schedule of topics as in Figure 3). PLDA is substantially faster to train, and also results in a better correspondence with human similarity judgments. We note, however, that practitioners should use models like PLDA with care, choosing the set of labels modeled and topics per label depending on the statistics of the dataset. PLDP can help by automatically determining an appropriate number of topics per label class, but its flexibility comes at the expense of speed, as the model takes substantially longer to train than the Gibbs sampler for PLDA, and does not yet have a data parallel implementation.

5. CONCLUSION

This work proposes two topic models that incorporate label supervision in novel ways: PLDA and PLDP, which learn latent topic structure within the scope of observed, human-interpretable labels. The models introduce high-level constraints on latent topics that cause them to align with human provided labels, essentially "filling in the details" with the use of unsupervised machine learning. The addition of these constraints improves interpretability of the resulting topics, shortens running time, and improves correlation with similarity judgments. And because these models fit into the Bayesian framework, they can be extended to incorporate other features, such as time or sequence information.

PLDA and PLDP provide a direct solution to the problem of label ambiguity: as in linguistic word usage, labels on tagging sites like del.icio.us and social media sites like Twitter are used with different meanings in different contexts. PLDA and PLDA can tease these meanings apart into separate latent topics within each label. However, the models do not directly address the inverse problem of synonymy, where several labels may refer to the same thing. Future work could look at recognizing (partially-) synonymous labels via a post-processing step or with explicit topic sharing. However, the independence of topics across labels is central to the favorable scalability of the models, as discussed in Section 4, so care must be taken when relaxing this constraint. We believe that PLDA, PLDP, and similar future models hold promise for addressing the challenges of partially supervised learning for more interpretable text mining, where human provided labels are present but do not always align with the needs of text mining practitioners.

Acknowledgments

This work was supported by the National Science Foundation under Grant No. 0835614.

6. REFERENCES

[1] A. Asuncion, M. Welling, P. Smyth, and Y.W. Teh. On smoothing and inference for topic models. In *UAI*, 2009.

[2] S. Basu, M. Bilenko, A. Banerjee, and R. J. Mooney. *In Semi-Supervised Learning*, chapter Probabilistic Semi-Supervised Clustering with Constraints, pages 73–102. MIT Press, 2006.

[3] D. Blei, T.L. Griffiths, M.I. Jordan, and J.B. Tenenbaum. Hierarchical topic models and the nested Chinese restaurant process. In *NIPS*, 2003.

[4] D. Blei and J. McAuliffe. Supervised Topic Models. In *NIPS*, volume 21, 2007.

[5] D. M. Blei, A.Y. Ng, and M.I. Jordan. Latent Dirichlet allocation. *JMLR*, 2003.

[6] SRK Branavan, H. Chen, J. Eisenstein, and R. Barzilay. Learning document-level semantic properties from free-text annotations. *Journal of Artificial Intelligence Research*, 34(1):569–603, 2009.

[7] C. Cattuto, A. Barrat, A. Baldassarri, Schehr G., and V. Loreto. Collective dynamics of social annotation. In *PNAS*, pages 10511–10515, 2009.

[8] J. Chang, J. Boyd-Graber, and D.M. Blei. Connections between the lines: augmenting social networks with text. In *KDD*, pages 169–178. ACM, 2009.

[9] J. Chang, J. Boyd-Graber, S. Gerrish, C. Wang, and D. Blei. Reading tea leaves: How humans interpret topic models. In *Neural Information Processing Systems*. Citeseer, 2009.

[10] O. Chapelle, B. Schölkopf, and A. Zien. *Semi-supervised learning*. MIT Press, 2006.

[11] S. Deerwester, S.T. Dumais, G.W. Furnas, T.K. Landauer, and R. Harshman. Indexing by latent semantic analysis. volume 41, pages 391–407. John Wiley & Sons, 1990.

[12] O. Dekel, P.M. Long, and Y. Singer. Online learning of multiple tasks with a shared loss. *JMLR*, 8:2233–2264, 2007.

[13] T. L. Griffiths and M. Steyvers. Finding scientific topics. *PNAS*, 1:5228–35, 2004.

[14] G. Heinrich. Parameter estimation for text analysis. Technical report, 2004. *http://www.arbylon.net/publications/text-est.pdf.*

[15] P. Heymann, G. Koutrika, and H. Garcia-Molina. Can social bookmarking improve web search. In *WSDM*, 2008.

[16] H Daumé III. Markov Random Topic Fields. In *ACL*, 2009.

[17] T. Iwata, T. Yamada, and N. Ueda. Modeling Social Annotation Data with Content Relevance using a Topic Model. In *NIPS*, 2009.

[18] S. Ji, L. Tang, S. Yu, and J. Ye. Extracting shared subspace for multi-label classification. In *KDD*, pages 381–389, New York, NY, USA, 2008. ACM.

[19] S. Lacoste-Julien, F. Sha, and M. I. Jordan. DiscLDA: Discriminative learning for dimensionality reduction and classification. In *NIPS*, volume 22, 2008.

[20] Wei Li and Andrew McCallum. Pachinko allocation: Dag-structured mixture models of topic correlations. In *International conference on Machine learning*, pages 577–584, 2006.

[21] C. Lin and Y. He. Joint sentiment/topic model for sentiment analysis. In *CIKM*, pages 375–384. ACM, 2009.

[22] A. McCallum and K. Nigam. A comparison of event models for naive bayes text classification. In *AAAI-98 workshop on learning for text categorization*, volume 7, 1998.

[23] D. Mimno and A. McCallum. Expertise modeling for matching papers with reviewers. In *KDD*, pages 500–509. ACM, 2007.

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

[25] D. Ramage. *Studying People, Institutions, and the Web with Statistical Text Models*. PhD thesis, Stanford University, 2011.

[26] D. Ramage, S. Dumais, and D. Liebling. Characterizing microblogs with topic models. In *ICWSM*, 2010.

[27] D. Ramage, D. Hall, R. Nallapati, and C. D. Manning. Labeled LDA: A supervised topic model for credit attribution in multi-label corpora. In *EMNLP*, pages 248–256, 2009.

[28] M. Rosen-Zvi, T. Griffiths, M. Steyvers, and P. Smyth. The author-topic model for authors and documents. In *UAI*. AUAI Press, 2004.

[29] T.N. Rubin, UC Irvine, A. Holloway, P. Smyth, and M. Steyvers. Modeling Tag Dependencies in Tagged Documents. In *NIPS 2009 Applications for Topic Models Workshop*, 2009.

[30] Y.W. Teh, M.I. Jordan, M.J. Beal, and D.M. Blei. Hierarchical dirichlet processes. *Journal of the American Statistical Association*, 101(476):1566–1581, 2006.

[31] I. Titov and R. McDonald. Modeling online reviews with multi-grain topic models. In *WWW*, pages 111–120. ACM, 2008.

[32] C. Wang, B. Thiesson, C. Meek, and D. Blei. Markov topic models. In *AISTATS*, pages 583–590, 2009.

[33] E.P. Xing, A.Y. Ng, M.I. Jordan, and S. Russell. Distance metric learning with application to clustering with side-information. In *NIPS*, pages 521–528, 2003.

[34] J. Zhu, A. Ahmed, and E.P. Xing. MedLDA: Maximum margin supervised topic models for regression and classification. In *ICML*, pages 1257–1264. ACM, 2009.

[35] X. Zhu. Semi-Supervised Learning Literature Survey. Technical Report 1530, Computer Sciences, University of Wisconsin-Madison, 2005.