Mach Learn (2012) 88:157–208
DOI 10.1007/s10994-011-5272-5

Statistical topic models for multi-label document classification

Timothy N. Rubin · America Chambers · Padhraic Smyth · Mark Steyvers

Received: 1 October 2010 / Accepted: 9 November 2011 / Published online: 29 December 2011

Abstract Machine learning approaches to multi-label document classification have to date largely relied on discriminative modeling techniques such as support vector machines. A drawback of these approaches is that performance rapidly drops off as the total number of labels and the number of labels per document increase. This problem is amplified when the label frequencies exhibit the type of highly skewed distributions that are often observed in real-world datasets. In this paper we investigate a class of generative statistical topic models for multi-label documents that associate individual word tokens with different labels. We investigate the advantages of this approach relative to discriminative models, particularly with respect to classification problems involving large numbers of relatively rare labels. We compare the performance of generative and discriminative approaches on document labeling tasks ranging from datasets with several thousand labels to datasets with tens of labels. The experimental results indicate that probabilistic generative models can achieve competitive multi-label classification performance compared to discriminative methods, and have advantages for datasets with many labels and skewed label frequencies.

Keywords Topic models · LDA · Multi-label classification · Document modeling · Text classification · Graphical models · Probabilistic generative models · Dependency-LDA

Editors: Grigorios Tsoumakas, Min-Ling Zhang, and Zhi-Hua Zhou.

T.N. Rubin (✉) · M. Steyvers
Department of Cognitive Sciences, University of California, Irvine, Irvine, CA 92697, USA
e-mail: trubin@uci.edu

M. Steyvers
e-mail: mark.steyvers@uci.edu

A. Chambers · P. Smyth
Department of Computer Science, University of California, Irvine, Irvine, CA 92697, USA

A. Chambers
e-mail: ahollowa@uci.edu

P. Smyth
e-mail: smyth@ics.uci.edu

1 Introduction

The past decade has seen a wide variety of papers published on multi-label document classification, in which each document can be assigned to one or more classes. In this introductory section we begin by discussing the limitations of existing multi-label document classification methods when applied to datasets with statistical properties common to real-world datasets, such as the presence of large numbers of labels with power-law-like frequency statistics. We then motivate the use of generative probabilistic models in this context. In particular, we illustrate how these models can be advantageous in the context of large-scale multi-label corpora, through (1) explicitly assigning individual words to specific labels within each document—rather than assuming that all of the words within a document are relevant to each of its labels, and (2) jointly modeling all labels within a corpus simultaneously, which lends itself well to the task of accounting for the dependencies between these labels.

1.1 Background and motivation

Much of the prior work on multi-label document classification uses data sets in which there are relatively few labels, and many training instances for each label. In many cases, the datasets are constructed such that they contain few, if any, infrequent labels. For example, in the commonly used RCV1-v2 corpus (Lewis et al. 2004), the dataset was carefully constructed to have approximately 100 labels, with most labels occurring in a relatively large number of documents.

In other cases researchers have typically restricted the problem by only considering a subset of the full dataset. As an example, a popular source of experimental data has been the Yahoo! directory structure, which utilizes a multi-labeling classification system. The true Yahoo! directory structure contains thousands of labels and is a very difficult classification problem that traditional classification methods fail to adequately handle (Liu et al. 2005). However, the majority of multi-label research conducted using the Yahoo! directory data has been performed on the set of 11 sub-directory datasets constructed by Ueda and Saito (2002). Each of these datasets consists of only the second-level categories from a single top-level Yahoo! directory, leaving only about 20–30 labels in each of the classification tasks. Furthermore, many of the publications (e.g., Ueda and Saito 2002; Ji et al. 2008) that use the Yahoo! subdirectory datasets have removed the infrequent labels from the evaluation data, leaving between 14 and 23 unique labels per dataset. Similarly, experiments with the OHSUMED MeSH terms (Hersh et al. 1994) are typically performed on a small subdirectory that contains only 119 out of over 22,000 possible labels (for a discussion, see Rak et al. 2005).

In contrast to the datasets typically utilized in research, multilabel corpora in the real world can contain thousands or tens of thousands of labels, and the label frequencies in these datasets tend to have highly skewed frequency-distributions with power-law statistics (Yang et al. 2003; Liu et al. 2005; Dekel and Shamir 2010). Figure 1 illustrates this point for three large real-world corpora—each containing thousands of unique labels—by plotting the number of labels within each corpus as a function of label-frequency. For each corpus, the total number of labels is plotted as a function of label-frequency on a log-log scale (i.e., more precisely, number of unique labels [y-axis] that have been assigned to k documents in the corpus is plotted as a function of k [x-axis]). Of note is the power-law like distribution of label frequencies for each corpus, in which the vast majority of labels are associated with very few documents, and there are relatively few labels that are assigned to a large number of documents. For example, roughly one thousand labels are only assigned to a

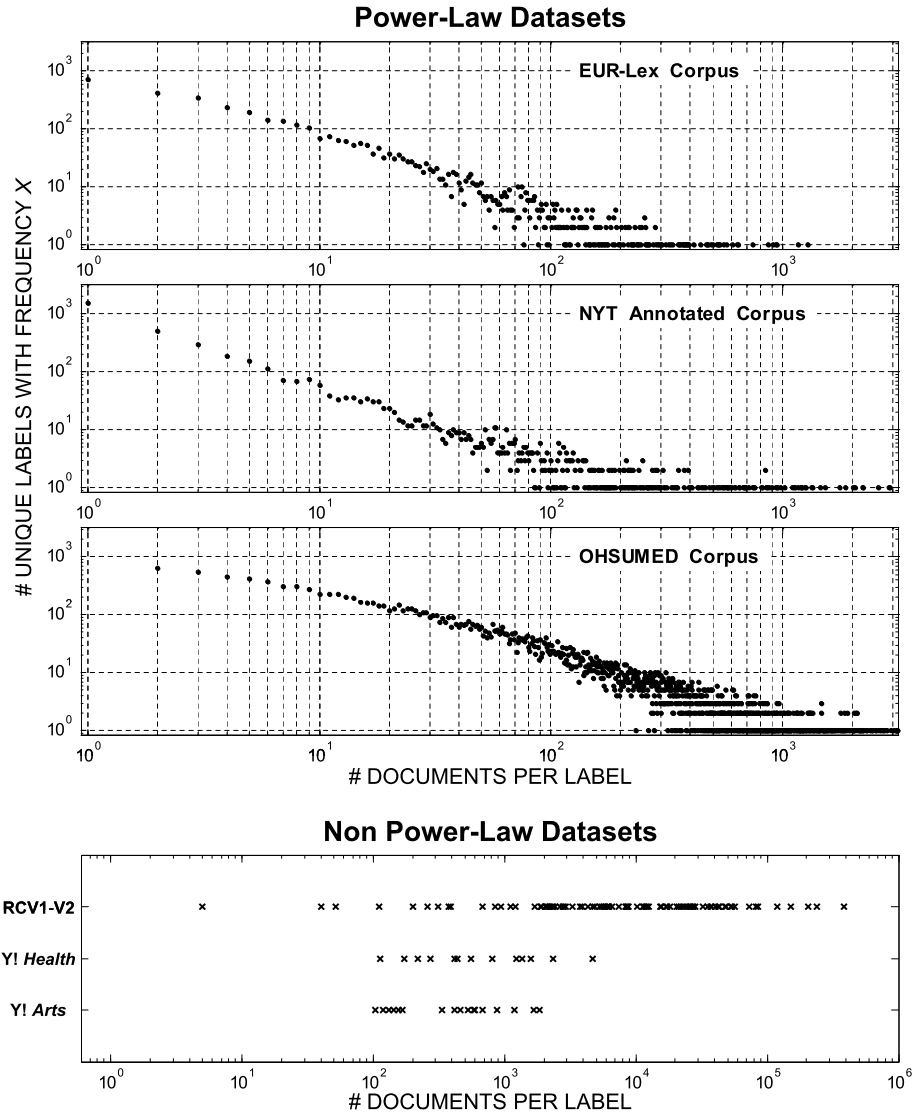


Fig. 1 *Top*: The number of unique labels (y-axis) that have K training documents (x-axis) for three large-scale multi-label datasets. Both axes are shown on a log-scale. The power-law-like relationship is evident from the near linear trend (in log-space) of this relationship. *Bottom*: The number of training documents(x-axis) for each unique label in three common (non-power-law) benchmark datasets. Since there are no label-frequencies at which there are more than one unique label in any of the datasets, if these plots were shown using the log-log scale used in the plots above, all points would fall along the y value corresponding to 10^0. Note that the scaling of the x-axis is not equivalent for the power-law and non power-law plots (this is necessary due to the high upper-bound of label-frequencies on the RCV1-V2 dataset)

single document in each corpus, and the median label-frequencies are 3, 6, and 12 for the NYT, EUR-Lex, and OHSUMED datasets, respectively. This stands in stark contrast to the widely-used Yahoo! *Arts*, Yahoo! *Health* and RCV1-v2 datasets (for example), which are shown at the bottom of Fig. 1. In these corpora, there are hardly any labels that occur in fewer

than 100 documents, and the median label-frequencies are 530, 500, and 7,410 respectively (see Sect. 4 for further details and discussion). To summarize, these popular benchmark datasets are drastically different from large-scale real-world corpora not only in terms of the number of unique labels they contain, but also with respect to the distribution of label-frequencies, and in particular the number of rare labels.

The mismatch between real-world and experimental datasets has been discussed previously in the literature, notably by Liu et al. (2005) who observed that although popular multi-label techniques—such as "one-vs-all" binary classification (e.g. Allwein et al. 2001; Rifkin and Klautau 2004)—can perform well on datasets with relatively few labels, performance drops off dramatically on real world datasets that contain many labels and skewed label frequency distributions. In addition, Yang (2001) illustrated that discriminative methods which achieve good performance on standard datasets do relatively poorly on larger datasets such as the full OHSUMED dataset. The obvious reason for this is that discriminative binary classifiers have difficulty learning models for labels with very few positively labeled documents. As stated by Liu et al. (2005), in the context of support vector machine (SVM) classifiers:

> In terms of effectiveness, neither flat nor hierarchical SVMs can fulfill the needs of classification of very large-scale taxonomies. The skewed distribution of the Yahoo! Directory and other large taxonomies with many extremely rare categories makes the classification performance of SVMs unacceptable. More substantial investigation is thus needed to improve SVMs and other statistical methods for very large-scale applications.

A second critical difference between large scale multi-label corpora and traditional benchmark datasets relates to the number of labels that are assigned to each document. Figure 2 compares the distributions of the number of labels per document for the same corpora shown in Fig. 1. The median number of labels per document for the real world, power-law style datasets are 6, 5, and 12 for EUR-Lex, NYT and OHSUMED, respectively. These numbers are significantly larger than those in the typical datasets used in multi-label classification experiments. For example, among the three benchmark datasets shown, the RCV1-v2 dataset has a median of 3 labels per document, and the Yahoo! *Arts* and *Health* datasets each have a median of only 1 label per document. These differences can significantly impact the performance of a classifier.

As the number of labels per document increases, it becomes more difficult for a discriminative algorithm to distinguish which words are discriminative for a particular label. This problem is further compounded when there is little training data per label. For the purposes of illustration, consider the following extreme case: suppose that we are training a binary classifier for a label, c_1, that has only been assigned to one document, d. Furthermore, assume that two additional labels, c_2 and c_3, have been assigned to document d, and that these labels occur in a relatively large number of documents. Since document d is the only positive training example for label c_1, an independent binary classifier trained on c_1 will learn a discriminant function that emphasizes not only words from document d that are relevant to label c_1, but also words that are relevant to labels c_2 and c_3, since the classifier has no way of "knowing" which words are relevant to these other labels. In other words, when training an independent binary classifier for label c_1, each additional label that co-occurs with c_1 will introduce additional confounding features for the classifier, thereby reducing the quality of the classifier.

Note however that in the above example it should be relatively easy to learn which features are relevant to the labels c_2 and c_3, since these labels occur in a large number of

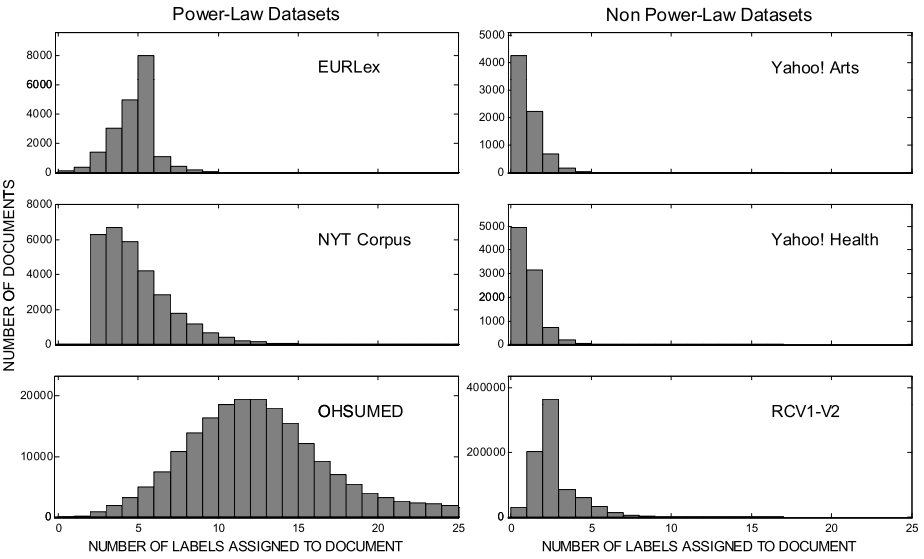


Fig. 2 Number of documents (y-axis) that have L labels (x-axis). The version of the NYT Annotated Corpus used in our experiments contains documents with 3 or more labels, hence the cutoff at 3

documents. Thus, we *should* be able to leverage this information to improve our classifier for c_1 by removing the features in d which we know to be relevant to these confounding labels. One possible approach to address this problem is to learn which individual word tokens within a document are likely to be associated with each label. If we could then use this information to identify which words within d are likely to be related to c_2 and c_3, we could "explain away" these words, and then use the remaining words for the purposes of learning a model for c_1. Note that for this purpose it is useful to (1) remove the assumption of label-wise independence, and (2) learn the models for all of the labels simultaneously, since learning which words within a document are irrelevant to a particular label is a key part of learning which words are relevant to the label.

1.2 A generative modeling approach

In a generative approach to document classification, one learns a model for the distribution of the words given each label, i.e., a model for $P(\underline{w}|c), 1 \leq c \leq C$, where $\underline{w}$ is the set of words in a document, and constructs a discriminant function for the label via Bayes rule. In standard supervised learning, with one label per document, these C distributions are typically learned independently. With multi-label data, the distributions should instead be learned simultaneously since we cannot separate the training data into C groups by label.

A useful approach in this context is a model known as latent Dirichlet allocation (LDA) (Blei et al. 2003), which we will also refer to as topic modeling, which models the words in a document as being generated by a mixture of topics, i.e., $P(w|d) = \sum_c P(w|c)P(c|d)$, where $P(w|d)$ is the marginal probability of word w in document d, $P(w|c)$ is the probability of word w being generated given label c, and $P(c|d)$ is the relative probability of each of the c labels associated with document d. LDA has primarily been viewed as an unsupervised learning algorithm, but can also be used in a supervised context (e.g., Blei and McAuliffe 2008; Mimno and McCallum 2008; Ramage et al. 2009). Using a supervised version of

LDA it is possible to learn both the word-label distributions $P(w|c)$ and the document-label weights $P(c|d)$ given a training corpus with multi-label data.

What is particularly relevant is that this approach (1) models the assignment of labels at the word-level, rather than at the document level as in discriminative models, and (2) learns a model for all labels at the same time, rather than treating each label independently. In particular, for the document d in our earlier example that was assigned the set of labels $\{c_1, c_2, c_3\}$, the model can explain away words that belong to labels c_2 and c_3—i.e., words that have high probability $P(w|c)$ under these labels. Since c_2 and c_3 are frequent labels, it will be relatively easy to learn which features are relevant to these labels, since the confounding features introduced by co-occurring labels in a multi-label scheme will tend to cancel out over many documents. The remaining words that cannot be explained well by c_2 or c_3 will be assigned to label c_1, and the model will learn to associate such words with this label and not associate with c_1 the words that are more likely to belong to labels c_2 and c_3. This general intuition is the basis for our approach in this paper. Specifically, we investigate supervised versions of topic models (LDA) as a general framework for multi-label document classification. In particular, the topic modeling approach allows for the type of "explaining away" effect at the word level that we hypothesize should be particularly helpful for the types of rare labels that pose challenges to purely discriminative methods.

Figure 3 illustrates the advantages an LDA-based approach has in terms of learning rare labels. On the left is the partial text of a news article, taken from the New York Times, along with three human-assigned labels: ANTITRUST ACTIONS AND LAWS and SUITS AND LITIGATION (which both occur in multiple other documents) and VIDEO GAMES (for which this document is the only positive example in the training data). On the right are the words with the highest weights from a binary SVM classifier trained on the label VIDEO GAMES. Beside this column are the highest probability words learned by an LDA-based model (described in more detail later in the paper). The words learned by the SVM classifier are quite noisy, containing a mixture of words relevant to the other two labels (e.g., *suing*, *infringement*, etc.), as well as rare words that are peculiar to the specific document rather than being relevant features for any of the labels (e.g., *futuristic*, *illusion*, etc.). These words do not match our intuition of words that would be discriminative for the concept VIDEO GAMES. Furthermore, as we will see later in the experimental results section, SVM classifiers trained on rare labels in this type of multi-label problem do not predict well on new test documents. While the set of words learned by LDA model is still somewhat noisy, it is nonetheless clear the model has done a better job in determining which words are relevant to the label VIDEO GAMES, and which of the words should be associated with the other two labels (e.g., there are no words with high probability that directly relate to lawsuits). The model benefits from not assuming independence between the labels, as with binary SVMs, as well as from the "explaining away" effect.

Thus far we have focused our discussion on the issue of learning appropriate models for labels during training. An additional issue that arises as the number of total labels (as well as the number of labels per document) increases, is the importance of accounting for higher-order dependencies between labels at prediction time (i.e., when classifying a new document). For example, suppose that we are predicting which labels should be assigned to a test-document that contains the word *steroids*. In a large-scale dataset like the NYT corpus, this word is a high-probability feature among many different labels, such as MEDICINE AND HEALTH, BASEBALL, and BLACK MARKETS. The ambiguity in the assignment of this word to a specific label can often be resolved if we account for the other labels within the document; e.g., the word *steroids* is likely to be related to the label BASEBALL given that the label SUSPENSIONS, DISMISSALS AND RESIGNATIONS is also assigned to the document,

NY Times Article		Models for VIDEO GAMES	
Document Labels	**Label Freq.**	**SVM (weight)**	**LDA (prob.)**
ANTITRUST ACTIONS AND LAWS	19	nintendo	nintendo
SUITS AND LITIGATION	67	mcgowan	games
VIDEO GAMES	1	futuristic	software
		compatible	video
		illusion	system
Document Excerpt		shrewd	game
		inception	chip
A flurry of lawsuits, started by a small American software developer, now surrounds the Nintendo Entertainment System, the best-selling toy in the United States last year...Atari Games argues that Nintendo's high degree of control is tantamount to monopoly, and is suing Nintendo for antitrust violations...		truthful	control
		profiles	market
		billionayear	home
		suing	computer
		infringement	shortage
		architecture	say
		handheld	buy
		tantamount	demand
		payoff	developer

Fig. 3 High-weight and high probability words for the label VIDEO GAMES learned by an SVM classifier and an LDA model (respectively) from the a set of New York Times articles, in which the label VIDEO GAMES only appeared once (text from the article is shown on the *left*)

whereas it is more likely to be related to MEDICINE AND HEALTH given the presence of the label CANCER.

Given this motivation, an additional beneficial feature of the topic model—and probabilistic methods in general— is that it is relatively straightforward to model the label dependencies that are present in the training data (a feature that we will elaborate on later in the paper). Modeling label dependencies is widely acknowledged to be important for accurate classification in multi-label problems, yet has been problematic in the past for datasets with large numbers of labels, as summarized in Read et al. (2009):

> The consensus view in the literature is that it is crucial to take into account label correlations during the classification process .... However as the size of the multi-label datasets grows, most methods struggle with the exponential growth in the number of possible correlations. Consequently these methods are able to be more accurate on small datasets, but are not as applicable to larger datasets.

Thus, the ability of probabilistic models to account for label dependencies is a strong motivation for considering these types of approaches in large-scale multi-label classification settings.

1.3 Contributions and outline

In the context of the discussion above, this paper investigates the application of statistical topic modeling to the task of multi-label document classification, with an emphasis on corpora with large numbers of labels. We consider a set of three models based on the LDA framework. The first model, *Flat-LDA*, has been employed previously in various forms. Additionally, we present two new models: *Prior-LDA*, which introduces a novel approach to account for variations in label frequencies, and *Dependency-LDA*, which extends this approach to account for the dependencies between the labels. We compare these three topic models to two variants of a popular discriminative approach (one-vs-all binary SVMs) on five datasets with widely contrasting statistics.

We evaluate the performance of these models on a variety of predictions tasks. Specifically, we consider (1) *document-based* rankings (rank all labels according to their relevance to a test document) and binary predictions (make a strict yes/no classification about each label for a given document), and (2) *label-based* rankings (rank all documents according to their relevance to a label) and binary predictions (make a strict yes/no classification about each document for a given label).

The specific contributions of this paper are as follows:

- We describe two novel generative models for multi-label document classification, including one (Dependency-LDA) which significantly improves performance over simpler models by accounting for label dependencies, and is highly competitive with popular discriminative approaches on large-scale datasets.
- We report extensive experimental results on two multi-label corpora with large numbers of labels as well as three smaller benchmark datasets, comparing the proposed generative models with discriminative SVMs. To our knowledge this is the first empirical study comparing generative and discriminative models on large-scale multi-label problems.
- We demonstrate that LDA-based models—in particular the Dependency-LDA model—can be highly competitive with, or better than, SVMs on large-scale datasets with power-law like statistics.
- For document-based predictions, we show that Dependency-LDA has a clear advantage over SVMs on large-scale datasets, and is competitive with SVMs on the smaller, benchmark datasets.
- For label-based predictions, we demonstrate that Dependency-LDA generally outperforms SVMs on large-scale datasets. We furthermore show that there is a clear performance advantage for the LDA-based methods on rare labels (e.g., labels with fewer than 10 training documents).

The remainder of the paper is organized as follows. We begin by describing how standard unsupervised LDA can be adapted to handle multi-labeled text documents, and describe our extensions that incorporate label frequencies and label dependencies. We then describe how inference is performed with these models, both for learning the model from training data and for making predictions on new test documents. An extensive set of experimental results are then presented on a wide range of prediction tasks on five multi-label corpora. We conclude the paper with a discussion of the relative merits of the LDA-based approaches vs. SVM-based approaches, particularly in the context of both the dataset statistics and prediction tasks being considered.

2 Related work

A number of approaches have been proposed for adapting the unsupervised LDA model to the case of supervised learning—such as the Supervised Topic Model (Blei and McAuliffe 2008), Semi-LDA (Wang et al. 2007), DiscLDA (Lacoste-Julien et al. 2008), and MedLDA (Zhu et al. 2009)—however, these adaptations are designed for single label classification or regression, and are not directly applicable to multilabel classification.

A more recent approach proposed by Ramage et al. (2009)—Labeled-LDA (L-LDA)—was designed specifically for multi-label settings. In L-LDA, the training of the LDA model is adapted to account for multi-labeled corpora by putting "topics" in 1-1 correspondence with labels and then restricting the sampling of topics for each document to the set of labels that were assigned to the document, in a manner similar to the Author-Model described by

Rosen-Zvi et al. (2004) (where the set of authors for each document in the Author Model is now replaced by the set of labels in L-LDA). The primary focus of Ramage et al. (2009) was to illustrate that L-LDA has certain qualitative advantages over discriminative methods (e.g., the ability to label individual words, as well as providing interpretable snippets for document summarization). Their classification results indicate that under certain conditions LDA-based models may be able to achieve competitive performance with discriminative approaches such as SVMs.

Our work differs from that of Ramage et al. (2009) in two significant aspects. Firstly, we propose a more flexible set of LDA models for multi-label classification—including one model that takes into account prior label frequencies, and one that can additionally account for label dependencies—which lead to significant improvements in classification performance. The L-LDA model can be viewed as a special case of these models. Secondly, we conduct a much larger range and more systematic set of experiments, including in particular datasets with large numbers of labels with skewed frequency-distributions, and show that generative models do particularly well in this regime compared to discriminative methods. In contrast, Ramage et al. (2009) compared their L-LDA approach with discriminative models only on relatively small datasets (primarily on the Yahoo! sub-directory datasets discussed in the introduction).

Our work (as well as the Author Model and L-LDA model) can be seen as building on earlier ideas from the literature in probabilistic modeling for multilabel classification. McCallum (1999) and Ueda and Saito (2002) investigated mixture models similar to L-LDA, where each document is composed of a number of word distributions associated with document labels. These papers can be viewed as early forerunners of the more general LDA frameworks we propose in this paper.

More recently Ghamrawi and McCallum (2005) demonstrated that the probabilistic framework of conditional random fields showed promise for multilabel classification, compared to discriminative classifiers, as the number of labels within test documents increased. In follow-up work on these models, Druck et al. (2007) illustrated that this approach has the further benefit of being able to naturally incorporate unlabeled data for semi-supervised learning. A drawback of the CRF approach is scalability, particularly when accounting for label dependencies. Exact inference "is tractable only for about 3-12 [labels]" (Ghamrawi and McCallum 2005). Alternatives to exact inference considered in Ghamrawi and McCallum (2005) include a "supported inference" method which learns only to classify the label combinations that occur in the training set, and a binary-pruning method that employs an intelligent pruning method which ignores dependencies between all but the most commonly observed pairs of labels. Although this method may improve upon approaches that ignore dependencies when restricted to datasets with few labels and many examples (such as traditional benchmark datasets), it seems unlikely that any such methods will be able to properly account for dependencies in datasets with power-law frequency statistics (since nearly all dependencies in these datasets are between labels which have very sparse training data).

Zhang and Zhang (2010) present a hybrid generative-discriminative approach to multi-label classification. They first learn a Bayesian network structure that represents the independencies between labels. They then learn a discriminative classifier for each label in the order specified by the Bayesian network where the classifier for label c takes as features not only the words in the document but also the output of the classifiers for each of the labels in the parent set of c (i.e. the parent set specified by the Bayesian network). However, they apply their model to only small-scale datasets (the largest having 158 labels).

In terms of discriminative approaches to multi-label classification, there is a large body of prior work, which has been well-summarized elsewhere in the literature (e.g.,

see Tsoumakas and Katakis 2007; Tsoumakas et al. 2009). Most discriminative approaches to multi-label classification have employed some variant of the "binary problem-transformation" technique, in which the multi-label classification problem is transformed into a *set* of binary-classification problems, each of which can then be solved using a suitable binary classifier (Rifkin and Klautau 2004; Tsoumakas and Katakis 2007; Tsoumakas et al. 2009; Read et al. 2009). The most commonly employed method in the literature is the "one-vs-all" transformation, in which C independent binary classifiers are trained—one classifier for each label. These binary classification tasks are then handled using discriminative classifiers, most notably SVMs, but also via other methods such as perceptrons, naive Bayes, and kNN classifiers. As our baseline discriminative method in this paper, we use the "one-vs-all" approach with SVMs as the binary classifier, since this is the most commonly used discriminative approach in the current multi-label classification literature, and has been defended in the literature in the face of an increasing number of proposed alternative methods (e.g., see Rifkin and Klautau 2004). We note also that there is a prior thread of work on discriminative approaches that can handle label-dependencies. For example, another problem-transformation technique known as the "Label Powerset" method (Tsoumakas et al. 2009; Read et al. 2009) builds a binary classifier for *each* distinct subset of label-combinations that exist in the training data—however, these approaches tend not to scale well with large label sets due to combinatorial effects (Read et al. 2009).

3 Topic models for multilabel documents

In this section, we describe three models (depicted in Fig. 4 using graphical model notation) that extend the techniques of topic modeling to multi-label document classification. Before providing the details for each model, we first briefly introduce the notation that will be used to describe these topic models within the multi-label inference setting, as well as provide a high-level description of the relationships between the three models.

The general setup of the inference task for the multi-label topic models we describe is as follows: the observed data for each document $d \in \{1, \ldots, D\}$ are a set of words $\mathbf{w}^{(\mathbf{d})}$ and labels $\mathbf{c}^{(\mathbf{d})}$. For all models, each label-type $c \in \{1, \ldots, C\}$ is modeled as a multinomial distribution ϕ_c over words. Each document d is modeled as a multinomial distribution θ_d over the document's observed label-types. Words for document d are generated by first sampling a label-type z from θ_d, and then sampling a word-token w from ϕ_z. The three models that we present differ with respect to how they model the generative process for labels.

The first model we describe is a straightforward extension of LDA to labeled documents, which we will refer to as Flat-LDA, where the labels are treated as given; this model makes no generative assumptions regarding how labels $\mathbf{c}^{(\mathbf{d})}$ are generated for a document. We then describe an extension to the Flat-LDA model—Prior-LDA—that incorporates a generative process for the labels themselves via a single corpus-wide multinomial distribution over all the label-types in the corpus. This assumption of Prior-LDA is very useful for making predictions when the label-frequencies are highly non-uniform. Lastly, we describe Dependency-LDA, which is a hierarchical extension to the previous two models that captures the dependencies between the labels by modeling the generative process for labels via a topic model; in Dependency-LDA, label-tokens for each document d are sampled from a set of T corpus-wide topics, according to a document-specific distribution θ'_d over the topics. We note that the Flat-LDA and Prior-LDA models can be viewed as special cases of the Dependency-LDA model. In particular, the Prior-LDA model is equivalent Dependency-LDA if we set the number of topics $T = 1$.

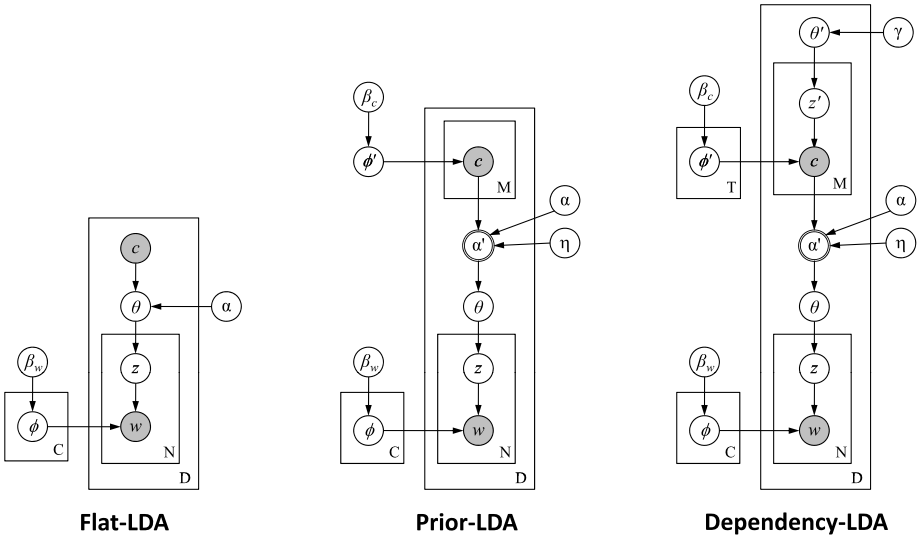


Fig. 4 Graphical models for multi-label documents. The observed data for each document d are a set of words $\mathbf{w}^{(\mathbf{d})}$ and labels $\mathbf{c}^{(\mathbf{d})}$. *Left*: In Flat-LDA, no generative assumptions are made regarding how labels are generated; labels for each document are assumed to be given. *Center*: The Prior-LDA model assumes that the label-tokens $c^{(d)}$ for each document are generated by sampling from a corpus-wide multinomial distribution over label-types ϕ', which captures the relative frequencies of different label-types across the corpus. *Right*: The Dependency-LDA model assumes that the label-tokens for each document are sampled from a set of T corpus-wide *topics*—where each "topic" t corresponds to a multinomial distribution over label-types ϕ'_t—according to a document-specific distribution θ'_d over these topics

3.1 Flat-LDA

The latent Dirichlet allocation (LDA) model, also referred to as the topic model, is an unsupervised learning technique for extracting thematic information, called topics, from a corpus. LDA represents topics as multinomial distributions over the W unique word-types in the corpus and represents documents as a mixture of topics. Flat-LDA is a straightforward extension of the LDA model to labeled documents. The set of LDA *topics* is substituted with the set of unique *labels* observed in the corpus. Additionally, each document's distribution over topics is restricted to the set of observed labels for that document.

More formally, let C be the number of unique labels in the corpus. Each label c is represented by a W-dimensional multinomial distribution ϕ_c over the vocabulary. For document d, we observe both the words in the document $\mathbf{w}^{(\mathbf{d})}$ as well as the document labels $\mathbf{c}^{(\mathbf{d})}$. The generative process for Flat-LDA is shown below. Each document is associated with a multinomial distribution θ_d over its set of labels. The random vector θ_d is sampled from a symmetric Dirichlet distribution with hyper-parameter α and dimension equal to the number of labels $|\mathbf{c}^{(\mathbf{d})}|$. Given the distribution over topics θ_d, generating the words in the document follows the same process as LDA:

1. For each label $c \in \{1, \ldots, C\}$, sample a distribution over word-types $\phi_c \sim \text{Dirichlet}(\cdot|\beta)$
2. For each document $d \in \{1, \ldots, D\}$

   (a) Sample a distribution over its observed labels $\theta_d \sim \text{Dirichlet}(\cdot|\alpha)$

(b) For each word $i \in \{1, \ldots, N_d^W\}$
  i. Sample a label $z_i^{(d)} \sim \text{Multinomial}(\theta_d)$
  ii. Sample a word $w_i^{(d)} \sim \text{Multinomial}(\phi_c)$ from the label $c = z_i^{(d)}$

Note that this model assigns each word token within a document to just a single label—specifically to one of the labels that was assigned to the document. The model is depicted using graphical model notation in the left panel of Fig. 4.

Due to the similarity between the Flat-LDA model presented here, and both the Author-Model from Rosen-Zvi et al. (2004) and the L-LDA model from Ramage et al. (2009), it is important to note precisely the relationships between these models. The Author-Model is conditioned on the set of *authors* in a document (and a "topic" is learned for each author in the corpus), whereas L-LDA and Flat-LDA are conditioned on the set of *labels* assigned to a document (and a "topic" is learned for each label in the corpus). L-LDA and Flat-LDA are *in practice* equivalent models, but employ different generative descriptions. Specifically, L-LDA models the generative process for each label in a document as a Bernoulli variable (where the parameter of the Bernoulli distribution is label-dependent). However, during training, estimating the Bernoulli parameters is independent from learning the assignment of words to labels (i.e. the z variables). Thus, during training both L-LDA and Flat-LDA reduce to standard LDA with an additional restriction that words can only be assigned to the observed labels in the document. Similarly, when performing inference for unlabeled documents (i.e. at test time), Ramage et al. (2009) assume that L-LDA reduces to standard LDA. In this way, both Flat-LDA and L-LDA are *in practice* equivalent despite L-LDA including a generative process for labels.[1] Due to the mismatch between the generative description of L-LDA and how it is employed in practice, we find it pedagogically useful to distinguish between the models presented here and L-LDA.

3.2 Prior-LDA

An obvious issue with Flat-LDA is that it does not account for differences in the relative frequencies of the labels within a corpus. This is not a problem during training, because all labels are observed for training documents. However, for the purpose of prediction (labeling new documents at test-time), accounting for the prior probabilities of each label becomes important, particularly when there are dramatic differences in the frequencies of labels in a corpus (as is the case with power-law datasets, as well as with many traditional datasets, such as RCV1-V2). In this section we present Prior-LDA, which extends Flat-LDA by incorporating a generative process for labels that accounts for differences in the observed frequencies of different label types. This is achieved using a two-stage generative process for each document, in which we first sample a set of observed labels from a corpus-wide multinomial distribution, and then given these labels, generate the words in the document.

Let ϕ' be a corpus-wide multinomial distribution over labels (reflecting, for example, a power-law distribution of label frequencies). For document d, we draw M_d samples from ϕ'. Each sample can be thought of as a single vote for a particular label. We replace $\alpha^{(d)}$, the symmetric Dirichlet prior with hyperparameter α, with a C-dimensional vector $\alpha'^{(d)}$ where the ith component is proportional to the total number of times label i was sampled from ϕ'. Formally, the vector $\alpha'^{(d)}$ is defined to be:

$$\alpha'^{(d)} = \left[\eta * \frac{N_{d,1}}{M_d} + \alpha, \eta * \frac{N_{d,2}}{M_d} + \alpha, \ldots, \eta * \frac{N_{d,C}}{M_d} + \alpha\right] \tag{1}$$

[1] Due to equivalence of Flat-LDA and L-LDA in practice, the experimental results we present for Flat-LDA are equivalent to what would be expected for L-LDA.

where $N_{d,i}$ is the number of times label i was sampled from ϕ'. In other words, $\alpha'^{(d)}$ is a scaled, smoothed, normalized vector of label counts.[2] The hyper-parameter η specifies the total weight contributed by the observed labels $\mathbf{c^{(d)}}$ and the hyper-parameter α is an additional smoothing parameter that contributes a flat pseudocount to each label. We define the document's label set $\mathbf{c^{(d)}}$ to be the set of labels with a non-zero component in $\alpha'^{(d)}$. To make this model fully generative, we place a symmetric Dirichlet prior on ϕ'.

Consider, for example, three labels $\{c_1, c_2, c_3\}$ with frequencies $\phi' = \{0.5, 0.3, 0.2\}$ in the corpus. For document d, we draw M_d samples from ϕ'. Assume $M_d = 5$ and the set $\{c_1, c_2, c_1, c_1, c_1\}$ was sampled. Then the hyper-parameter $\alpha'^{(d)}$ would be:

$$\alpha'^{(d)} = \left[\eta * \frac{4}{5} + \alpha \ , \ \eta * \frac{1}{5} + \alpha \ , \ \eta * \frac{0}{5} + \alpha \right]$$

If hyperparameter $\alpha = 0$, then $\alpha'^{(d)}$ has only two non-zero components (because the last component equals zero) and $\mathbf{c^{(d)}} = \{\mathbf{c_1}, \mathbf{c_2}\}$. In this case, the multinomial vector θ_d drawn from Dirichlet $(\alpha'^{(d)})$ will always have zero count for the third label (i.e. label c_3 will have probability zero in the document). If $\alpha > 0$, then $\mathbf{c^{(d)}} = \{\mathbf{c_1}, \mathbf{c_2}, \mathbf{c_3}\}$ and label c_3 will have non-zero probability in the document. As M_d goes to infinity, $\alpha'^{(d)}$ approaches the vector $\eta\, \phi' + \alpha$.

The multinomial distribution may seem like an unnatural choice for a label-generating distribution since the observed labels in a document are most naturally represented using binary variables rather than counts. We experimented with alternative parameterizations such as a multivariate Bernoulli distribution. However, this introduced problems during both training and testing. As noted by Schneider (2004) in relation to modeling document *words* (rather than labels), the multivariate Bernoulli distribution tends to overweight negative evidence (i.e. the absence of a word in a document) during training, due to the sparsity of the word-document matrix. This problem is compounded when modeling document *labels* because there are considerably fewer labels in a document than words. Furthermore, at test time when the document labels are unobserved, a Bernoulli model will converge more slowly since the probability of turning on a label in a document is higher than the probability of turning off a label in a document (this is due to the fact that a label can only be turned off after all words assigned to that label have been assigned elsewhere).[3]

The generative process for the Prior-LDA model is:

1. Sample a multinomial distribution over labels $\phi' \sim \text{Dirichlet}(\cdot|\beta_C)$
2. For each label $c \in \{1, \ldots, C\}$, sample a distribution over word-types $\phi_c \sim \text{Dirichlet}(\cdot|\beta_W)$
3. For each document $d \in \{1, \ldots, D\}$:
   (a) Sample M_d label tokens $c_j^{(d)} \sim \text{Multinomial}(\phi'),\ 1 \le j \le M_d$
   (b) Compute the Dirichlet prior $\alpha'^{(d)}$ for document d according to (1)
   (c) Sample a distribution over labels $\theta_d \sim \text{Dirichlet}(\cdot|\alpha'^{(d)})$
   (d) For each word $i \in \{1, \ldots, N_d^W\}$
      i. Sample a label $z_i^{(d)} \sim \text{Multinomial}(\theta_d)$
      ii. Sample a word $w_i^{(d)} \sim \text{Multinomial}(\phi_c)$ from the label $c = z_i^{(d)}$

This model is depicted using graphical model notation in the center panel of Fig. 4.

[2] In the training data, we set M_d equal to the number of observed labels in document d and $N_{d,i}$ equal to 0 or 1 depending upon whether the label is present in the document.

[3] A related issue was the reason given by Ramage et al. (2009) for resorting in practice to a Flat-LDA scheme during inference.

3.3 Dependency-LDA

Prior-LDA accounts for the prior label frequencies observed in the training set, but it does not account for the dependencies between the labels, which is crucial when making predictions for new documents. In this section, we present Dependency-LDA, which extends Prior-LDA by incorporating another topic model to capture the dependencies between labels. The labels are generated via a topic model where each "topic" is a distribution over labels. Dependency-LDA is an extension of Prior-LDA in which there are T corpus-wide probability distributions over labels, which capture the dependencies between the labels, rather than a single corpus-wide distribution that merely reflects relative label frequencies. We note that several models that represent or induce topic dependencies have been investigated in the past for unsupervised topic modeling (e.g., Blei and Lafferty 2005; Teh et al. 2004; Mimno et al. 2007; Blei et al. 2010). Although these models are related to varying degrees to the Dependency-LDA model, as unsupervised models they are not directly applicable to document classification.

Formally, let T be the total number of topics where each topic t is a multinomial distribution over labels denoted ϕ'_t. Generating a set of labels for a document is analogous to generating a set of words in LDA. We first sample a distribution over topics θ'_d. To generate a single label we sample a topic z' from θ'_d and then sample a label from the topic $\phi'_{z'}$. We repeat this process M_d times. As in Prior-LDA, we compute the hyper-parameter vector $\alpha'^{(d)}$ according to (1) and define the label set $\mathbf{c^{(d)}}$ as the set of labels with a non-zero component. Given the set of labels $\mathbf{c^{(d)}}$, generating the words in the document follows the same process as Prior-LDA.

1. For each topic $t \in \{1, \ldots, T\}$, sample a distribution over labels, ${\phi'}_t \sim \text{Dirichlet}(\beta_{\mathcal{C}})$
2. For each label $c \in \{1, \ldots, C\}$, sample a distribution over words, $\phi_c \sim \text{Dirichlet}(\beta_{\mathcal{W}})$
3. For each document $d \in \{1, \ldots, D\}$:
   (a) Sample a distribution over topics ${\theta'}_d \sim \text{Dirichlet}(\gamma)$
   (b) For each label $j \in \{1, \ldots, M_d\}$
      i. Sample a topic $z'^{(d)}_j \sim \text{Multinomial}({\theta'}_d)$
      ii. Sample a label $c^{(d)}_j \sim \text{Multinomial}(\phi'_t)$ from the topic $t = z'^{(d)}_j$
   (c) Compute the Dirichlet prior $\alpha'^{(d)}$ for document d according to (1)
   (d) Sample a distribution over labels $\theta_d \sim \text{Dirichlet}(\cdot|\alpha'^{(d)})$
   (e) For each word $i \in \{1, \ldots, N^W_d\}$
      i. Sample a label $z^{(d)}_i \sim \text{Multinomial}(\theta_d)$
      ii. Sample a word $w^{(d)}_i \sim \text{Multinomial}(\phi_c)$ from the label $c = z^{(d)}_i$

The Dependency-LDA model is depicted using graphical model notation in the right panel of Fig. 4.

3.4 Topic model inference methods—model training

This section gives an overview of the inference methods used with the three LDA-based models (Flat-LDA, Prior-LDA, and Dependency-LDA). We first describe how to perform inference and estimate the model parameters during training (i.e., when document labels are observed). We then describe how to perform inference for test documents (i.e., when labels are unobserved).

Training all three LDA-based models requires estimating the C multinomial distributions ϕ_c of labels over word-types. Additionally, Prior-LDA and Dependency-LDA require

estimation of the T multinomial distributions ϕ'_t of topics over label types, where $T = 1$ for Prior-LDA and $T > 1$ for Dependency-LDA. Additionally, training (and testing) for all models requires setting several hyperparameter values.

Note that we set the hyperparameter $\alpha = 0$ in Prior-LDA and Dependency-LDA during training—but not during testing/prediction—which restricts the assignments of words to the set of observed labels for each document (see (1)). This is consistent with the assumptions of these models, because in the training corpus all labels are observed, and the models assume that words are generated by one of the true labels. This also greatly simplifies training, because it serves to decouple the upper and lower parts of the models (namely, with $\alpha = 0$, the topic-label distributions ϕ'_t and the label-word distributions ϕ_c are conditionally independent from each other, given that we have observed all labels).

Furthermore, estimation of the ϕ_c distributions is in fact *equivalent* for all three models when $\alpha = 0$ for Prior-LDA and Dependency-LDA (and, for consistency, we used the same set of parameter estimates for ϕ_c when evaluating all models). A benefit—in terms of model evaluation—of using the same estimates for ϕ_c across all models is that it controls for one possible source of performance variability; i.e., it ensures that observed performance differences are due to factors other than estimation of ϕ_c. Specifically, differences in model performance can be directly attributed to qualitative differences between the models in terms of how they parameterize the Dirichlet prior $\alpha'^{(d)}$ for each test document.

In addition to the smoothing parameter α, there are several other hyperparameters in the models that must be chosen by the experimenter. For all experiments, hyperparameters were chosen heuristically, and were not optimized with respect to any of our evaluation metrics. Thus, we would expect that at least a modest improvement in performance over the results presented in this paper could be obtained via hyperparameter optimization. For details regarding the hyperparameter values we used for all experiments in this paper, and a discussion regarding our choices for these values, see Appendix B.

3.4.1 Learning the label-word distributions: Φ

To learn the C multinomial distributions ϕ_c over words, we use a modified form of the collapsed Gibbs sampler described by Griffiths and Steyvers (2004) for unsupervised LDA. In collapsed Gibbs sampling, we learn the distributions ϕ_c over words, and the D distributions θ_d over labels, by sequentially updating the latent indicator $z_i^{(d)}$ variables for all word tokens in the training corpus (where the ϕ_c and θ_d multinomial distributions are integrated—i.e., "collapsed"—out of the update equations).

For Flat-LDA, the assignment of words in document d is restricted to the set of observed labels $\mathbf{c}^{(\mathbf{d})}$. For Prior-LDA and Dependency-LDA a word can be assigned to any label as long as the smoothing parameter α is non-zero. The Gibbs sampling equation used to update the assignment of each word token $z_i^{(d)}$ to a label c is:

$$P(z_i^{(d)} = c \mid w_i^{(d)} = w, \mathbf{w}_{-i}, \mathbf{c}^{(d)}, \boldsymbol{\alpha}'^{(d)}, \mathbf{z}_{-i}, \beta_{\mathcal{W}}) \propto \frac{N^{WC}_{wc,-i} + \beta_{\mathcal{W}}}{\sum_{w'=1}^{W}(N^{WC}_{w'c,-i} + \beta_{\mathcal{W}})} * (N^{CD}_{cd,-i} + \alpha'^{(d)}_c) \tag{2}$$

where N^{WC}_{wc} is the number of times the word w has been assigned to the label c (across the entire training set), and N^{CD}_{cd} is the number of times the label c has been assigned to a word in document d. We use a subscript $-i$ to denote that the current token, z_i, has been removed from these counts. The first term in (2) is the probability of word w in label c computed by integrating over the ϕ_c distribution. The second term is proportional to the probability of label c in document d, computed by integrating over the θ_d distribution.

Table 1 The eight most likely words for five labels in the NYT Dataset, along with the word probabilities. The number to the right of the labels indicates the number of training documents assigned the label

POLITICS AND GOVERNMENT	285	ARMS SALES ABROAD	176	ABORTION	24	ACID RAIN	11	AGNI MISSILE	1
Party	.014	Iran	.021	Abortion	.098	Acid	.070	Missile	.032
Government	.014	Arms	.019	Court	.033	Rain	.067	India	.031
Political	.011	Reagan	.014	Abortions	.028	Lakes	.028	Technology	.016
Leader	.006	House	.014	Women	.017	Environmental	.026	Missiles	.016
President	.005	President	.014	Decision	.016	Sulfur	.024	Western	.015
Officials	.005	North	.012	Supreme	.016	Study	.023	Miles	.014
Power	.005	Report	.011	Rights	.015	Emissions	.021	Nuclear	.013
Leaders	.005	White	.011	Judge	.015	Plants	.021	Indian	.013

For all results presented in this paper, during training we set $\alpha = 0$ and η equal to 50. Early experimentation indicated that the exact value of η was generally unimportant as long as $\eta \gg 1$. We ran multiple independent MCMC chains, and took a single sample at the end of each chain, where each sample consists of the current vector of $\mathbf{z}$ assignments (see Appendix B for additional details). We use the $\mathbf{z}$ assignments to compute a point estimate of the distributions over words:

$$\hat{\phi}_{w,c} = \frac{N_{wc}^{WC} + \beta_{\mathcal{W}}}{\sum_{w'=1}^{W}(N_{w'c}^{WC} + \beta_{\mathcal{W}})} \tag{3}$$

where $\hat{\phi}_{w,c}$ is the estimated probability of word w given label c. The parameter estimates $\hat{\phi}_{w,c}$ were then averaged over the samples from all chains. Several examples of label-word distributions, learned from a corpus of NYT documents, are presented in Table 1.

Similarly, a point estimate of the posterior distribution over labels θ_d for each document is computed by:

$$\hat{\theta}_{c,d} = \frac{N_{cd}^{CD} + \alpha'^{(d)}_{c}}{\sum_{c'=1}^{C}(N_{c'd}^{CD} + \alpha'^{(d)}_{c'})} \tag{4}$$

where $\hat{\theta}_{c,d}$ is the estimated probability of label c given document d.

3.4.2 Learning the topic-label distributions: Φ'

Note that this section only applies to the Prior-LDA and Dependency-LDA models since the Flat-LDA model does not employ a generative process for labels.[4] Learning the T multinomial distributions ϕ'_t over labels is equivalent to applying a standard LDA model to the label tokens. In our experiments, we employed a collapsed Gibbs sampler (Griffiths and Steyvers 2004) for unsupervised LDA, where the update equation for the latent topic indicators $z_i^{\prime(d)}$

[4] Additionally, since there is only one "topic" to learn for the Prior-LDA model, the estimation problem for this model simplifies to computing a single maximum-a-posteriori estimate of the Dirichlet-multinomial distribution ϕ'.

is given by:

$$P(z'^{(d)}_i = t \mid c^{(d)}_i = c,\ \mathbf{c}_{-i}, \mathbf{z'}_{-i},\ \gamma,\ \beta_{\mathcal{C}}) \propto \frac{N^{CT}_{ct,-i} + \beta_{\mathcal{C}}}{\sum_{c'=1}^{C}(N^{CT}_{c't,-i} + \beta_{\mathcal{C}})} * \left(N^{DT}_{dt,-i} + \gamma\right) \quad (5)$$

where N^{CT}_{ct} is the number of times label c has been assigned to topic t (across the entire training set), and N^{DT}_{dt} is the number of times topic t has been assigned to a label in document d. The subscript $-i$ denotes that the current label-token z'_i has been removed from these counts. The first term in (5) is the probability of label c in topic t computed by integrating over the ϕ'_t distribution. The second term is proportional to the probability of topic t in document d, computed by integrating over the θ'_d distribution.

For training, we experimented with different values of $T \leq C$ (for Dependency-LDA). We set $\gamma \ll 1$, and adjusted $\beta_{\mathcal{C}}$ in proportion to the ratio of the number of topics T to the total number of observed labels in each training corpus (see Appendix B for additional details).

For each MCMC chain, we ran the Gibbs sampler for a burn-in of 500 iterations, and then took a single sample of the vector of $\mathbf{z}'$ assignments. Given this vector, we compute a posterior estimate for the ϕ'_t distributions:

$$\hat{\phi'}_{c,t} = \frac{N^{CT}_{ct} + \beta_{\mathcal{C}}}{\sum_{c'=1}^{C}(N^{CT}_{c't} + \beta_{\mathcal{C}})} \quad (6)$$

where $\hat{\phi'}_{c,t}$ is the estimated probability of label c given topic t. For each training corpus, we ran ten MCMC chains (giving us ten distinct sets of topics).[5] Several examples of topics, learned from a corpus of NYT documents, are presented in Table 2.

Similarly, a point estimate of the posterior distribution over topics θ'_d for each document is computed by:

$$\hat{\theta'}_{d,t} = \frac{N^{DT}_{dt} + \gamma}{\sum_{t'=1}^{T}(N^{DT}_{dt'} + \gamma)} \quad (7)$$

where $\hat{\theta'}_{d,t}$ is the estimated probability of topic t given document d.

3.5 Topic model inference methods—test documents

In this section, we first describe a proper inference method for sampling the three LDA-based models during test time, when the document labels are unobserved. In the following section, we describe an approximation to the proper inference method which is computationally much faster, and achieved performance that was as accurate as the true sampling methods. We note again that the hyperparameter settings used for all experiments are provided in Appendix B.

At test time, we fix the label-word distributions $\hat{\phi}_c$, and topic-label distributions $\hat{\phi'}_t$, that were estimated during training. Inference for a test document d involves estimating its distribution over label types θ_d and a set of label-tokens $\mathbf{c}^{(\mathbf{d})}$, given the observed word

[5] We can not average our estimates of ϕ'_t over multiple chains as we did when estimating ϕ_c. This because the topics are being learned in an unsupervised manner, and do not have a fixed meaning between chains. Thus, each chain provides a distinct estimate of the set of T ϕ'_t distributions. For test documents, we average our predictions over the set of 10 chains. See Appendix B for additional details.

Table 2 The ten most likely labels within three of the topics learned by the Dependency LDA model on the NYT dataset. Topic labels (in quotes) are subjective interpretations provided by the authors

"Consumer Safety"	.017	"Warfare And Disputes"	.024	"Cheating and Athletics"	.016
CANCER	.078	ARMAMENT, DEFENSE AND MILITARY FORCES	.162	OLYMPIC GAMES (1988)	.052
HAZARDOUS AND TOXIC SUBSTANCES	.039	INTERNATIONAL RELATIONS	.133	SUSPENSIONS, DISMISSALS AND RESIGNATIONS	.038
PESTICIDES AND PESTS	.021	UNITED STATES INTERNATIONAL RELATIONS	.132	BASEBALL	.033
RESEARCH	.021	CIVIL WAR AND GUERRILLA WARFARE	.098	SUMMER GAMES (OLYMPICS)	.031
SURGERY AND SURGEONS	.021	MILITARY ACTION	.053	FOOTBALL	.029
TESTS AND TESTING	.021	CHEMICAL WARFARE	.029	ATHLETICS AND SPORTS	.026
FOOD	.018	REFUGEES AND EXPATRIATES	.019	COLLEGE ATHLETICS	.019
RECALLS AND BANS OF PRODUCTS	.018	INDEPENDENCE MOVEMENTS	.013	STEROIDS	.019
CONSUMER PROTECTION	.016	BOUNDARIES AND TERRITORIAL ISSUES	.011	GAMBLING	.017
HEALTH, PERSONAL	.016	KURDS	.010	WINTER GAMES (OLYMPICS)	.017

tokens $\mathbf{w}^{(\mathbf{d})}$. Additionally, inference for Dependency-LDA involves estimating a document's distribution over topics, θ'_d. We first describe inference at the word-label level (which is equivalent for all three LDA models given the Dirichlet prior $\alpha'^{(d)}$), and then describe the additional inference steps involved in Dependency-LDA. Note that for all models, inference for each test document is independent.

The θ_d parameter is estimated by sequentially updating the $z_i^{(d)}$ assignments of word tokens to label types. The Gibbs update equation is modified from (2) to account for the fact that we are now using fixed values for the ϕ_c distributions, which were learned during training, rather than an estimate computed from the current values of z assignments via N_{wc}^{WC}:

$$P\left(z_i^{(d)} = c \mid w_i^{(d)} = w,\ \mathbf{w}_{-i}^{(d)},\ \alpha'^{(d)},\ z_{-i}^{(d)},\ \hat{\phi}_{w,c}\right) \propto \hat{\phi}_{w,c} * \left(N_{cd,-i}^{CD} + \alpha_c'^{(d)}\right) \tag{8}$$

where $\hat{\phi}_{w,c}$ was estimated during training using (3), N_{cd}^{CD} is the number of times the label c has been assigned to a word in document d, and where $\alpha_c'^{(d)}$ is the value of the document-specific Dirichlet prior on label-type c for document d, as defined in (1).

The only difference that arises between the three LDA models when sampling the $\mathbf{z}$ variables is in the document-specific prior $\alpha'^{(d)}$. To simplify the following discussion, we describe inference in terms of Dependency-LDA. We note again that Prior-LDA is a special case of Dependency-LDA in which $T = 1$, and therefore the descriptions of inference for Dependency-LDA are fully applicable to Prior-LDA.[6]

Since the label tokens are unobserved for test documents, exact inference requires that we sample the label tokens $\mathbf{c}^{(\mathbf{d})}$ for the document. The label tokens $\mathbf{c}^{(\mathbf{d})}$ are dependent on the assignment $\mathbf{z}'$ of label-tokens to topics in addition to the vector of word-assignments $\mathbf{z}$. We therefore must also sample the variables $z'^{(d)}$. The Gibbs sampling equation for $c_i^{(d)}$, given the trained model, and a document's vector of z and z' assignments, is:

$$p\left(c_i^{(d)} = c \mid z_i'^{(d)} = t,\ z_{-i}'^{(d)},\ c_{-i}^{(d)},\ z^{(d)},\ \hat{\phi}'_{t,c}\right) \propto \frac{\prod_{c'=1}^{C} \Gamma(\alpha_{c'}'^{(d)} + N_{c',d}^{CD})}{\prod_{c=1}^{C} \Gamma(\alpha_{c'}'^{(d)})} \cdot \hat{\phi}'_{t,c} \tag{9}$$

[6] In Flat-LDA, there is no document-specific Dirichlet prior. Instead, the prior for each document is simply a symmetric Dirichlet with hyperparameter α, i.e. $\alpha_c'^{(d)} = \alpha, c \in 1, \ldots, C$. Since this does not depend on any additional parameters, the remaining steps provided in this section are irrelevant to Flat-LDA.

where the first term on the right-hand side of the equation is the likelihood of the current vector of word assignments to labels $\mathbf{z^{(d)}}$ given the proposed set of label-tokens $\mathbf{c^{(d)}}$ (i.e., updated with value $c_i^{(d)} = c$), and N_{cd}^{CD} is the total number of words in document d that have been assigned to label c. The second term $\hat{\phi}'_{c,t}$ was estimated during training using (6). Since the update equation for $c_i^{(d)}$ is not transparent from the model itself, and has not been presented elsewhere in the literature, we provide a derivation of (9) in Appendix C.

Given the current values of the label tokens $\mathbf{c^{(d)}}$, the topic assignment variables $z'^{(d)}$ are conditionally independent of the label assignment variables $z^{(d)}$. The update equations for the $z'^{(d)}$ variables are therefore equivalent to (8), except that we are now updating the assignment of labels to topics rather than words to labels:

$$P\left(z'^{(d)}_i = t \mid c_i^{(d)} = c,\ \gamma,\ \mathbf{z}'^{(d)}_{-i},\ \hat{\phi}'_{t,c}\right) \propto \hat{\phi}'_{c,t} * \left(N_{dt,-i}^{DT} + \gamma\right) \tag{10}$$

where $N_{dt,-i}^{DT}$ is the number of times topic t has been assigned to a label in document d, and the document-specific distribution over topics θ'_d has been integrated out.

For each test document d, we sequentially update each of the values in the vectors $\mathbf{z^{(d)}}$, $\mathbf{c^{(d)}}$, and $\mathbf{z'^{(d)}}$. Since the $\mathbf{z^{(d)}}$ variables are conditionally independent of the $\mathbf{z'^{(d)}}$ variables given the $\mathbf{c^{(d)}}$ variables, the $\mathbf{c^{(d)}}$ variables are the means by which the word-level information contained in $\mathbf{z^{(d)}}$ and the topic-level information contained in $\mathbf{z'^{(d)}}$ can propagate back and forth. Thus, a reasonable update order is as follows:

1. Update the assignment of the *observed* word tokens $w^{(d)}$ to the labels: $z^{(d)}$ (8)
2. Sample a new set of label-tokens: $c^{(d)}$ (9)
3. Update the assignment of the *sampled* label-tokens to one of T topics: $z'^{(d)}$ (10)
4. Sample a new set of label-tokens: $c^{(d)}$ (9)

Each full cycle of these updates provides a single 'pass' of information from the words up to the topics and back down again. Once the sampler has been sufficiently burned in, we can then use the vectors $\mathbf{z^{(d)}}$, $\mathbf{c^{(d)}}$ and $\mathbf{z'^{(d)}}$ to compute a point estimate of a test document's distribution $\hat{\theta}_d$ over the label types using (4) (and the prior as defined in (1)).

Unfortunately, the proper Gibbs sampler runs into problems with computational efficiency. Intuitively, the source of these problems is that the c variables act as a bottleneck during inference since they are the only means by which information is propagated between the z and z' variables. To limit the extent of this bottleneck, we can increase the number of label tokens M_d that we sample. However, this is computationally expensive because sampling each c value requires substantially more computation than sampling the z and z' assignments, since computing each proposal value requires taking a product of C gamma values.[7]

3.5.1 Fast inference for dependency-LDA

We now describe an efficient alternative to the sampling method described above. Experimentation with this alternative inference method suggests that, in addition to requiring substantially less time, it in fact achieves similar or better prediction performance compared to proper inference.

[7]There are methods to optimize the sampler for $c^{(d)}$, which reduces the amount of computation required by several orders of magnitude (using simplification of the expression in (9) and careful storage and updating of the vector of gamma values). However, this method was still slower by an order of magnitude per iteration than the 'fast inference' method presented in the following section, and required a much longer burn-in (while giving similar, or worse, prediction performance).

The idea behind the fast-inference method is that, rather than explicitly sampling the values of c, we directly pass information between the label-level and topic-level parameters (thus avoiding the information bottleneck created by the c tokens, and also avoiding this costly inference step). This can be achieved by directly passing the z values up to the topic-level, and treating each z value as if it was an observed label token c. In other words, we substitute the vector of sampled label tokens $c^{(d)}$ with the vector of label assignments $z^{(d)}$ for each document; since both $z_i^{(d)}$ and $c_i^{(d)}$ can take on the same set of values (between 1 and C), these vectors can be treated equivalently when sampling the topic-assignments $z'^{(d)}_i$ for them. Then, after updating the z' values, we can directly compute the posterior predicted distribution over label types, $p(c|d)$, by conditioning on the current z' assignments, and use this to compute $\alpha'^{(d)}$.

To motivate this approach, let Φ' be the T-by-C matrix where row t contains ϕ'_t. Let θ'_d be the T-dimensional multinomial distribution over topics. We can directly compute the posterior predictive distribution over labels given Φ' and θ'_d, as follows:

$$p(c_i^{(d)} = c \mid \theta'_d, \Phi') \propto \sum_{t=1}^{T} p(c_i^{(d)} = c \mid z'^{(d)}_i = t) \cdot p(z'^{(d)}_i = t \mid d)$$

$$= \sum_{t=1}^{T} \Phi'_{t,c} \cdot \theta'_{d,t} \tag{11}$$

Thus, given the matrix Φ' (learned during training) and an estimate of the T-dimensional vector θ'_d, which we can compute using (7), the hyper-parameter vector $\alpha'^{(d)}$ can be directly computed using:

$$\alpha'^{(d)} = \eta\ (\hat{\theta'}_d \cdot \Phi') + \alpha \tag{12}$$

Once we have updated the z' variables, (12) allows us to compute $\alpha'^{(d)}$ directly without explicitly sampling the c variables.[8] An alternative defense of this approach is that as M_d goes to infinity in the generative model for Dependency-LDA, the vector $\alpha'^{(d)}$ approaches the expression given in (12).

The sequence of update steps we use for this approximate inference method is:

1. Update the assignment of the *observed* word tokens $w^{(d)}$ to one of the C label types: $z^{(d)}$ (8)
2. Set the label-tokens ($c^{(d)}$) equal to the label assignments: $c_i^{(d)} = z_i^{(d)}$
3. Update the assignment of the label tokens to one of T topics: $z'^{(d)}$ (10)
4. Compute the hyperparameter vector: $\alpha'^{(d)}$ (12)

As before, each full cycle of these updates provides a single 'pass' of information from the words up to the topics and back down again. But rather than sampling the $c^{(d)}$ label-tokens, we directly pass the $z^{(d)}$ variables up to the topic-level sampler, and use these as an approximation of the vector $c^{(d)}$. Then, given the current estimate of $\theta'^{(d)}$ (shown in (7)), we compute the $\alpha'^{(d)}$ prior directly using (12).[9]

[8] This is in fact the correct posterior-predicted value of $\alpha'^{(d)}$ in the generative model, given the variables Φ' and θ'_d. However, technically this is not correct during inference, because it ignores the values of the $\mathbf{z^{(d)}}$ variables, which are accounted for in the first term in (9).

[9] Note that the computational steps involved in this method are in fact very close to the proper inference methods. The first and third steps (updating z and z') are equivalent to the true sampling updates. The second

Table 3 Computational Complexity (per iteration) for the three LDA-based methods. N_W: Number of word-tokens in the dataset; N_C: Number of observed label tokens in the (training) set; D: Number of documents in the training set; C: Number of unique label-types; T: Number of topics

Training		Testing	
Training Φ	$O(N_W(N_C/D))$	Flat-LDA	$O(N_W C)$
Training Φ'	$O(N_C T)$	Prior-LDA	$O(N_W C)$
		Dep-LDA	$O(N_W(C+T))$

Once the sampler has been sufficiently burned in, we can then use the assignments $\mathbf{z}^{(\mathbf{d})}$, and $\mathbf{z}'^{(\mathbf{d})}$ to compute a point estimate of a test document's distribution $\hat{\theta}_d$ over the label types using (4) (and the prior as defined in (12)).

We compared performance between this method and the proper inference method (with $M_d = 1000$) on a single split of the EURLex corpus. In addition to providing significantly better predictions on the test dataset, the fast inference method was more efficient. Even after optimizing the $c_i^{(d)}$ sampling, the fast inference method was well over an order of magnitude faster (per iteration) than proper inference, and also converged in fewer iterations. Due to its computational benefits, we employed the fast inference method for all experimental results presented in this paper.

The computational complexity for training and testing the three LDA-based algorithms is presented in Table 3.[10] Note that the complexity of Dependency-LDA does not involve a term corresponding to the square of the number of unique labels (C), which is often the case for algorithms that incorporate label dependencies (a discussion of this issue can be found in, e.g., Read et al. 2009).

3.6 Illustrative comparison of predictions across different models

To illustrate the differences between the three models, consider a word w that has equal probability under two labels c_1 and c_2 (i.e., $\phi_{1,w} = \phi_{2,w}$). In Flat-LDA, the Dirichlet prior on θ_d is uninformative, so the only difference between the probabilities that z will take on value c_1 versus c_2 are due to the differences in the number of current assignments (N^{CD} for c_1 and c_2) of word tokens in document d. In Prior-LDA, the Dirichlet prior reflects the relative *a-priori* label-probabilities (from the single corpus-wide topic), and therefore the z assignment probabilities will reflect the baseline frequencies of the two labels in addition to the current z counts for this document. In Dependency-LDA, the Dirichlet prior reflects a prior distribution over labels given an (inferred) document-specific mixture of the T topics, and therefore the assignment probabilities reflect the relationships between the (inferred) document's labels and all other labels, in addition to the current counts of z.

Figure 5 shows an illustrative example of the predictions different models made for a single document in the NYT collection. An excerpt from this document is shown alongside the four true labels that were manually assigned by the NYT editors. The top ten label predictions (with the true labels in bold) illustrate how Dependency-LDA leverages both

step actually closely replicates what we would expect if we set $M_d = N_d^W$ and then sampled each $c_i^{(d)}$ explicitly, except that we are now ignoring the topic-level information when we actually construct the vector $c^{(d)}$ (although this information has a strong influence on the z assignments, so it is not unaccounted for in the $c^{(d)}$ vector).

[10] Complexity for Dependency-LDA during testing is given for the fast-inference method.

baseline frequencies and correlations to improve predictions over the simpler Prior-LDA and Flat-LDA models. Additionally, this illustration indicates how Dependency-LDA can achieve better performance than SVMs by improving performance on rare labels.

Given the set of label-word distributions learned during training, Flat-LDA predicts the labels which most directly correspond to the words in the document (i.e., the labels that are assigned the most words when we do not account for any information beyond the label-word distributions, due to the words having high probabilities $\phi_{c,w}$ under the models for these labels). As shown in Fig. 5, this Flat-LDA approach ranks two out of four of the true labels among its top ten predictions, including the rare label IMMUNITY FROM PROSECUTION. Prior-LDA improves performance over Flat-LDA by excluding infrequent labels, except when the evidence for them overwhelms the small prior. For example, the rare label MIDGETMAN (MISSILE) which is ranked sixth for Flat-LDA—but has a relatively small probability under the model—is not ranked in the top ten for Prior-LDA, whereas IMMUNITY FROM PROSECUTION, which is also a rare label but has a much higher probability under the model, stays in the same ranking position under Prior-LDA. Also, the label UNITED STATES INTERNATIONAL RELATIONS, which isn't ranked in the top ten under Flat-LDA, is ranked sixth under Prior-LDA due in part to its high prior probability (i.e. its high baseline frequency in the training set).

The Dependency-LDA model improves upon Prior-LDA by additionally including ARMAMENT, DEFENSE AND MILITARY FORCES high in its rankings. This improvement is attributed to the semantic relationship between this label and the labels ARMS SALES ABROAD and UNITED STATES INTERNATIONAL RELATIONS (e.g., note that the labels ARMAMENT, DEFENSE AND MILITARY FORCES and UNITED STATES INTERNATIONAL RELATIONS are, respectively, the first and third most likely labels under the middle topic shown in Table 2). Lastly, note that binary SVMs[11] performed well on the three frequent labels, but missed the rare label IMMUNITY FROM PROSECUTION. This is because the binary SVMs learned a poor model for the label due to the infrequency of training examples, which—as discussed in the introduction—is one of the key problems with the binary SVM methods.

4 Experimental datasets

The emphasis of the experimental work in this paper is on two multi-label datasets each containing many labels and skewed label-frequency distributions: the NYT annotated corpus (Sandhaus 2008) and the EUR-Lex text dataset (Loza Mencía and Fürnkranz 2008b). We use a subset of 30,658 articles from the full NYT annotated corpus of 1.5 million documents, with over 4000 unique labels that were assigned manually by The New York Times Indexing Service. The EUR-Lex dataset contains 19,800 legal documents with 3,993 unique labels. In addition, for comparison, we present results from three more commonly used benchmark multi-label datasets: the RCV1-v2 dataset of Lewis et al. (2004) and the *Arts* and *Health* subdirectories from the Yahoo! dataset (Ueda and Saito 2002; Ji et al. 2008), all of which have significantly fewer labels, and more examples per label, than the NYT and EUR-Lex datasets. Complete details on all of the datasets are provided in Appendix A.

Aspects of document classification relating to feature-selection and document-representation are active areas of research (e.g., see Forman 2003; Zhang et al. 2009). In order to

[11] These predictions were generated by the "Tuned SVM" implementation, the details of which are provided in Sect. 5.1.

New York Times Article

LEAD: The special Senate and House committees investigating the Iran-contra affair decided today to hold joint hearings, and set a timetable for granting limited immunity from prosecution to the two central witnesses.The extraordinary agreement, which also calls for merging the committee staffs and for sharing evidence, is expected to speed the inquiry...

True Document Labels	Label Freq.
IMMUNITY FROM PROSECUTION	4
ARMS SALES ABROAD	176
ARMAMENT, DEFENSE AND MILITARY FORCES	409
UNITED STATES INTERNATIONAL RELATIONS	630

	Flat-LDA	p	Prior-LDA	p	Dependency-LDA	p	Binary SVMs
1	**ARMS SALES ABROAD**	.204	**ARMS SALES ABROAD**	.261	**ARMS SALES ABROAD**	.291	CONGRESSIONAL INVESTIGATIONS
2	CONGRESSIONAL INVESTIGATIONS	.182	CONGRESSIONAL INVESTIGATIONS	.237	CONGRESSIONAL INVESTIGATIONS	.234	**ARMS SALES ABROAD**
3	LAW AND LEGISLATION	.059	LAW AND LEGISLATION	.102	**UNITED STATES INTERNATIONAL RELATIONS**	.110	**ARMAMENT, DEFENSE AND MILITARY FORCES**
4	**IMMUNITY FROM PROSECUTION**	.042	**IMMUNITY FROM PROSECUTION**	.062	LAW AND LEGISLATION	.100	**UNITED STATES INTERNATIONAL RELATIONS**
5	ETHICS	.004	ETHICS	.045	**IMMUNITY FROM PROSECUTION**	.063	CIVIL WAR AND GUERRILLA WARFARE
6	MIDGETMAN (MISSILE)	.003	**UNITED STATES INTERNATIONAL RELATIONS**	.024	**ARMAMENT, DEFENSE AND MILITARY FORCE**	.049	ETHICS
7	VETOES (US)	.003	TRIALS	.018	CIVIL WAR AND GUERRILLA WARFARE	.014	DISCLOSURE OF INFORMATION
8	UNITED STATES ARMAMENT AND DEFENSE	.003	UNITED STATES ARMAMENT AND DEFENSE	.010	DECISIONS AND VERDICTS	.007	FOREIGN AID
9	CONGRESSIONAL COMMITTEES	.003	INTERNATIONAL RELATIONS	.008	FOREIGN AID	.007	UNITED STATES ARMAMENT AND DEFENSE
10	B-2 AIRPLANE	.003	FINANCES	.007	TRIALS	.005	LAW AND LEGISLATION

Fig. 5 Illustrative comparison of a set of prediction results for a single NYT test document

Table 4 Statistics of the experimental datasets. Traditional benchmark datasets are presented in the first three rows, and datasets with power-law-like statistics are presented in the last two rows

Dataset	Labels (C)	Documents (D)	Cardinality	Density	Mean Label Freq.	Median Label Freq.	Mode Label Freq.	Distinct Labelsets	Labelset Freq.	Unique Labelset Prop.
Y! Arts	19	7,441	1.6	.0855	636	530	–	527	14.1	.0406
Y! Health	14	9,109	1.6	.1149	1,047	500	–	241	37.8	.0113
RCV1-V2	103	804,414	3.2	.0315	25,310	7,410	–	13,922	57.8	.0093
NY Times	4,185	30,658	5.4	.0013	40	3	1	27,207	1.13	.8371
EUR-Lex	3,993	19,800	5.3	.0013	26	6	1	16,871	1.17	.7548

avoid confounding the influence of feature selection and document representation methods with performance differences between the models, we employed straightforward methods for both. Feature selection for all datasets was carried out by (1) removing stop words and (2) removing highly-infrequent words. For LDA-based models, each document was represented using a *bag-of-words* representation (i.e, a vector of word counts). For the binary SVM classifiers, we normalized the word counts for each document such that each document feature-vector summed to one (i.e., a vector of reals).

Table 4 presents the statistics for the datasets considered in this paper. In addition to several statistics that have been previously presented in the multi-label literature, we present additional statistics which we believe help illustrate some of difficulties with classification for large scale power-law datasets. All statistics are explained in detail below:

- CARDINALITY: The average number of labels per document
- DENSITY: The average number of labels per document divided by the number of unique labels (i.e., the cardinality divided by C), or equivalently, the average number of documents per label divided by the number of documents (i.e., Mean Label-Frequency divided by d)
- LABEL FREQUENCY (MEAN, MEDIAN, AND MODE): The mean, median, and mode of the distribution of the number of documents assigned to each label.
- DISTINCT LABEL SETS: The number of distinct combinations of labels that occur in documents.
- LABEL-SET FREQUENCY (MEAN): The average number of documents per distinct combination of labels (i.e., D divided by Distinct Label-sets).
- UNIQUE LABEL-SET PROPORTION: The proportion of documents containing a unique combination of labels.

The cardinality of a dataset reflects the degree to which a dataset is truly multi-label (a single-label classification corpus will have a cardinality $=1$). The density of a dataset is a measure of how frequently a label occurs on average. The mean, median, and mode for label frequency reflects how many training examples exist for each label (see also Fig. 1). All of these statistics reflect the sparsity of labels, and are clearly quite different among the two groups of datasets.

The last three measures in the table relate to the notion of label combinations. For example, the label-set proportion tells us the average number of documents that have a unique combination of labels, and the label-set frequency tells us on average how many examples we have for each of these unique combinations. These types of measures are particularly

relevant to the issue of dealing with label dependencies. For example, one approach to handling label-dependencies is to build a binary classifier for each unique *set* of labels (e.g., this approach is described as the "Label Powerset" method in Tsoumakas et al. 2009). For the three smaller datasets, there is a relatively low proportion of documents with unique combinations of labels, and in general numerous examples of each unique combination. Thus, building a binary classifier for each combination labels of could be a reasonable approach for these datasets. On the other hand, for the NYT and EUR-Lex datasets these values are both close to 1, meaning that nearly all documents have a unique set of labels, and thus there would not be nearly enough examples to build effective classifiers for label-combinations on these datasets.

5 Experiments

In this section we introduce the prediction tasks and evaluation metrics used to evaluate model performance for the three LDA-based models and two SVM methods. The results of all evaluations described in this section—which are performed on the five datasets shown in Table 4—will be presented in the following section. The objectives of these experiments were (1) to compare the Dependency-LDA model to the simpler LDA-based models (Prior-LDA and Flat-LDA), (2) to compare the performance of the LDA-based models with SVM-based models, and (3) to explore the conditions under which LDA-based models may have advantages over more traditional discriminative methods, with respect to both prediction tasks and to the dataset statistics.

Before delving into the details of our experiments, we first describe the binary SVM classifiers we implemented for comparisons with our LDA-based models.

5.1 Implementation of binary SVM classifiers

In both of our SVM approaches we used a "one-vs-all" (sometimes referred to as "one-vs-rest") scheme, in which a binary Support Vector Machine (SVM) classifier was independently trained for each of the C labels. Documents were represented as a normalized vector of word counts, and SVM training was implemented using the LibLinear version 1.33 software package (Fan et al. 2008).

For "Tuned-SVMs", we followed the approach of Lewis et al. (2004) for training C binary support vector machines (SVMs). All parameters except the weight parameter for positive instances were left at the default value. In particular, we used an L2-loss SVM with a regularization parameter of 1. The weight parameter for negative instances was kept at the default value of 1. The weight parameter for positive instances (w1) was determined using a hold-out set. The weight parameters alter the penalty of a misclassification for a certain class. This is especially useful for labels with small support where it is often desirable to penalize misclassifying a positive instance more heavily than misclassifying a negative instance (Japkowicz and Shaju 2002). The parameter w1 was selected from the following values:

$$\{1, 2, 5, 10, 25, 50, 100, 250, 500, 1000, w_c\}$$

The last value, w_c, is a ratio of the number of negative instances to the number of positive instances in the training set for label c. If there are an equal number of negative and positive instances then $w_c = 1$.

The hold-out set consisted of 10% of the positive instances and 10% of the negative instances from the training set. If a label had only one positive instance it was included in both the training set and the hold-out set. The weight value that had the highest accuracy on the hold-out set was selected. If there was a tie, the weight value closest to 1 was chosen. Once the best value of w1 was determined, the final SVM was re-trained on the entire training set.

We additionally provide results for "Vanilla SVMs", which were generated using LibLinear with default parameter settings (the default parameter value for w1 was 1) for all labels.

5.2 Multi-label prediction tasks

Numerous prediction tasks and evaluation metrics have been adopted in the multi-label literature (Sebastiani 2002; Tsoumakas et al. 2009; de Carvalho and Freitas 2009). There are two broad perspectives on how to approach multi-label datasets: (1) *document-pivoted* (also known as *instance-based* or *example-based*), in which the focus is on generating predictions for each test-document, and (2) *label-pivoted* (also known as *label-based*), in which the focus is on generating predictions for each label. Within each of these classes, there are two types of predictions that we can consider: (1) *binary* predictions, where the goal is to make a strict yes/no classification about each test item, and (2) *ranking* predictions, in which the goal is to rank relevant cases above irrelevant cases. Taken together, these choices comprise four different prediction tasks that can be used to evaluate a model, providing an extensive basis for comparing LDA and SVM-based models.

Figure 5 illustrates the relationship between both the *label-pivoted* vs. *document-pivoted* and the *binary* vs. *ranking* tasks. In order to produce as informative and fair a comparison of the LDA-based and SVM-based models as possible, we considered both ranking-predictions and binary-predictions for both the document-pivoted and label-pivoted prediction tasks.

Traditionally, multi-label classification has emphasized the label-pivoted binary classification task, but increasingly there has been growing interest in performance on document-pivoted ranking (e.g., see Har-Peled et al. 2002; Crammer and Singer 2003; Loza Mencía and Fürnkranz 2008a, 2008b) and binary predictions (e.g., see Fürnkranz et al. 2008). To calibrate our results with respect to this literature, we adopt many of the ranking-based evaluation metrics used in this literature in addition to the more traditional metrics based on ROC-analysis. We also provide results which can be compared with values that have been published in the literature (although this is often difficult, due to the dearth of published results for large multi-label datasets and the variability of different versions of benchmark datasets, as well as the lack of consensus over evaluation metrics and prediction tasks). Appendix D contains a detailed discussion of how our results compare to earlier results reported in the literature.

5.3 Rank-based evaluation metrics

On the label-ranking task, for each test document we predict a ranking of all C possible labels, where the broad goal is to rank the relevant labels (i.e., the labels that were assigned to the document) higher than the irrelevant labels (the labels that were not assigned to the document).[12] We consider several evaluation metrics that are rooted in ROC-analysis, as well

[12]For simplicity, we describe the rank-based evaluation metrics in terms of the document-pivoted rankings. However, we also use these metrics for evaluating label-pivoted rankings (where the goal is to predict a ranking of all D documents, for each label).

as measures that have been used more recently in the label-ranking literature. We provide a general description of these measures below (more formal definitions of these measures can be found in, e.g., Crammer and Singer 2003).[13] For each measure, the range of possible values is given in brackets, and the best possible score is in bold:

- AUC_{ROC} $[0 - \mathbf{1}]$: The area under the ROC-curve. The ROC-curve plots the false-alarm rate versus the true-positive rate for each document as the number of positive predictions changes from $0 - C$. To combine scores across documents we compute a macro-average (i.e. the AUC_{ROC} is first computed for each document and is then averaged across documents).
- AUC_{PR} $[0 - \mathbf{1}]$: The area under the precision-recall curve.[14] This is computed for each document using the method described in Davis and Goadrich (2006), and scores are combined using a macro-average.
- AVERAGE PRECISION $[0 - \mathbf{1}]$: For each relevant label x, the fraction of all labels ranked higher than x which are correct. This is first averaged over all relevant labels within a document and then averaged across documents.
- ONE-ERROR $[\mathbf{0} - 100]$: The percentage of all documents for which the highest-ranked label is incorrect.
- IS-ERROR $[\mathbf{0} - 100]$: The percentage of documents without a *perfect* ranking (i.e., the percentage of all documents for which all relevant labels are not ranked above all irrelevant labels.
- MARGIN $[\mathbf{1} - C]$: The difference in ranking between the highest-ranked irrelevant label and the lowest ranked relevant label, averaged across documents.
- RANKING LOSS $[\mathbf{0} - 100]$: Of all possible comparisons between the rankings of a single relevant label and single irrelevant label, the percentage of these that are incorrect. First averaged across all comparisons within a document, then across all documents.[15]

5.4 Binary prediction measures

The basis of all binary prediction measures that we consider are macro-averaged and micro-averaged F1 scores (*Macro-F1* and *Micro-F1*) (Yang 1999; Tsoumakas et al. 2009). Traditionally, the literature has emphasized the label-pivoted perspective, in which F1 scores are first computed for each label and then averaged across labels. However, recently there has been an increased interest in binary predictions on a per-document basis (e.g., see Fürnkranz et al. 2008, who refer to this task as *calibrated label-ranking*). We consider both the document-pivoted and label-pivoted approaches to the evaluation of binary predictions.

The F1 score for a document d_i, or a label c_i, is the harmonic mean of precision and recall of the set of binary predictions for that item. Given the set of C binary predictions for a document, or the set of D binary predictions for a label, the F1-score is defined as:

$$F1(i) = \frac{2 \times Recall(i) \times Precision(i)}{Recall(i) + Precision(i)} \tag{13}$$

[13] In order to provide results consistent with published scores on the EURLex dataset we use the same [0, 100] scaling used by Loza Mencía and Fürnkranz (2008a) of the last four measures.

[14] Although the area under the ROC curve is more traditionally used in ROC-analysis, Davis and Goadrich (2006) demonstrated that the area under the Precision-Recall curve is actually a more informative measure for imbalanced datasets.

[15] We note that the RANKING LOSS statistic corresponds to the complement of the area under the ROC curve (scaled): $\text{RANKLOSS} = 100 \times (1 - \text{AUC}_{\text{ROC}})$, which, furthermore is equivalent to the Mann-Whitney U statistic. To simplify comparisons with published results, we present the results in terms of both the Ranking Loss and the AUC_{ROC}.

Table 5 Illustration of the relationship between the two prediction tasks (binary predictions vs. rankings), for both the label-pivoted and document-pivoted perspectives on multi-label datasets. The table on the *left* shows the ground-truth for a toy dataset with three documents and five labels. For binary predictions, the goal is to reproduce this table by making hard classifications for each label or each document (for example, a perfect document-pivoted binary prediction for document $d1$ assigns a positive prediction '+' to labels c_1, c_2 and c_3, and a negative prediction '−' to labels c_4 and c_5). For ranking-based predictions, one ranks all items for each test-instance and the goal is to rank relevant items above irrelevant items (for example, a perfect document-pivoted ranking for document $d1$ is any predicted ordering in which labels c_1, c_2 and c_3 are all ranked above c_4 and c_5). In the notation used for this illustration, the vertical bar '|' indicates the ranking which partitions positive and negative items; thus, any permutation on the order of the items between a vertical-bar '|' and a bracket is equivalent from an accuracy viewpoint (since there is no ground truth about the relative values within the set of true labels or within the set of false labels)

	Binary					Ranking-based			
	Label-pivoted					Document-pivoted		Label-pivoted	
Document-pivoted	c1	c2	c3	c4	c5				
d1	+	+	+	−	−	d1:	$\{c_1, c_2, c_3 \mid c_4, c_5\}$	c1:	$\{d_1, d_2 \mid d_3\}$
d2	+	−	+	+	−	d2:	$\{c_1, c_3, c_4 \mid c_2, c_5\}$	c2:	$\{d_1, d_3 \mid d_2\}$
d3	−	+	−	−	+	d3:	$\{c_2, c_5 \mid c_1, c_3, c_4\}$	c3:	$\{d_1, d_2 \mid d_3\}$
								c4:	$\{d_2 \mid d_1, d_3\}$
								c5:	$\{d_3 \mid d_1, d_2\}$

After computing the F1 scores for all items, the performance can be summarized using either *micro-averaging* or *macro-averaging*. In macro-averaging, one first computes an F1-score for each of the individual test items using its own confusion matrix, and then takes the average of the F1-scores. In micro-averaging, a single confusion matrix is computed for all items (by summing across the individual confusion matrices), and then the F1-score is computed for this single confusion matrix. Thus, the micro-average gives more weight to the items that have more positive test-instances (e.g., the more frequent labels), whereas the macro-average gives equal weight to each item, independent of its frequency.

We note that one must be careful when interpreting F1-scores, since these measures are very sensitive to differences in dataset statistics as well as to differences in model performance. As the label frequencies become increasingly skewed (as in the power-law datasets like NY Times and EUR-Lex), the potential disparity between the Macro-F1 and Micro-F1 becomes increasingly large; a model that performs well on frequent labels but very poorly on infrequent labels (which are in the vast majority for a power-law dataset) will have a poor Macro-F1 score but can still have a reasonably good Micro-F1 score.

5.5 Binary predictions and thresholding

As illustrated in Table 5, a binary-prediction task can be seen as a direct extension of a ranking task. If we have a classifier that outputs a set of real-valued predictions for each of the test instances, then a predicted ranking can be produced by sorting on the prediction values. We can transform this ranking into a set of binary predictions by either (1) learning a threshold on the prediction values, above which all instances are assigned a positive prediction (e.g. the 'SCut' method (Yang 2001) is one example of this approach), or (2) making a positive prediction for the top N ranked instances for some chosen N.

The issue of choosing a threshold-selection method is non-trivial (particularly for large-scale datasets) and threshold selection comprises a significant research problem in and of itself (e.g., see Yang 2001; Fan and Lin 2007; Ioannou et al. 2010). Since threshold-selection

is not the emphasis of our own work, and we do not wish to confound differences in the models with the effects of thresholding, we followed a similar approach to that of Fürnkranz et al. (2008) and considered several rank-based cutoff approaches.[16] The three rank-cutoff values which we consider are:

1. PROPORTIONAL: Set $\hat{N}_i$ equal the expected number of positive instances for item i, based on training-data frequencies:
   - For label c_i (i.e., label-pivoted predictions): $\hat{N}_i = \text{ceil}(\frac{D^{TEST}}{D^{TRAIN}} * N_i^{TRAIN})$, where N_i^{TRAIN} is the number of training documents assigned label c_i, and D^{TRAIN} and D^{TEST} are the total number of documents in the training and test sets, respectively.[17]
   - For test document d_i (i.e., document-pivoted predictions): $\hat{N}_i = \text{median}(N_d^{TRAIN})$ where N_d^{TRAIN} is the number of labels for training document d.
2. CALIBRATED: Set $\hat{N}_i$ equal to the true number of positive instances for item i.
3. BREAK-EVEN-POINT (BEP): Set $\hat{N}_i$ such that it optimizes the F1-score for that item, given the predicted order. This method is commonly referred to as the *Break-Even Point* (BEP) because it selects the location on the Precision-Recall curve at which *Precision* = *Recall*.

Note that the latter two methods both use information from the test set, and thus do not provide an accurate representation of performance we would expect for the models in a real-world application. However, in addition to the practical value of these methods for model comparison, they each provide measures of model performance at points of theoretical interest: The CALIBRATED method gives us a measure of model performance if we assume that there is some external method (or model) which tells us the *number* of positive instances, but not *which* of these instances are positive. The BEP method (which has been commonly employed in multi-label classification literature) tells us the highest attainable F1-score for each item given the predicted ordering. Thresholding methods which attempt to maximize the macro-averaged F1 score are in fact searching for a threshold as close to the BEP as possible. Note that although the BEP provides the highest possible macro-F1 score on a dataset, this does not mean that it will optimize the Micro-F1 score; in fact, since the method optimizes the F1-score for each label independently, it will generate a large number of false-positives when the predicted ordering has assigned the actual positive instances a low rank, which can have large negative impact on Micro-F1 scores.

We additionally point out that whereas the BEP method will vary the number of positive predictions to account for a model's specific ranking, the PROPORTIONAL and CALIBRATED methods will produce the same number of positive predictions for all models. Thus, scores on these predictions reflect model performance at a fixed cutoff point which is independent of the model's ranking.

[16] Note that the cutoff-points we use are slightly different from those presented in Fürnkranz et al. (2008). In particular, since our models are not learning a calibrated cutoff during inference, we substituted their PREDICTED method with the more traditional BREAK-EVEN-POINT (BEP) method. Additionally, our PROPORTIONAL cutoff has been modified from the MEDIAN approach that they use in order to extend it to the label-pivoted case, since the median value is generally not applicable for label-pivoted predictions.

[17] For label-pivoted predictions, SVMs do in fact learn a threshold which partitions the data during training, unlike the LDA models. However, we found that in most cases the performance at these thresholds is much worse than performance using the PROPORTIONAL method (this is particularly true on the power-law datasets, due to the difficulties with learning a proper SVM model on rare labels). This is consistent with results that have been noted previously in the literature—e.g., see Yang (2001).

6 Experimental results

Results below are organized as follows: (1) document-pivoted results on all datasets for (a) ranking-predictions and (b) binary-predictions, and then (2) label-pivoted results on all datasets for (a) ranking-predictions and (b) binary-predictions. For completeness, we provide a table for each of the four tasks using all evaluation metrics and datasets.

6.1 Document-pivoted results

The document-pivoted predictions provide a ranking of all labels in terms of their relevance to each test-document d. The seven *ranking*-based metrics directly evaluate aspects of each of these rankings. The six *binary* metrics evaluate the binary predictions after these rankings have been partitioned into positive and negative labels for each document, using the three aforementioned cutoff-points. Results for the rank-based evaluations are shown in Fig. 6, and results for the binary predictions are shown in Fig. 7.

6.1.1 Comparison within LDA-based and SVM-based models (Doc-pivoted)

Among the LDA-based models, Dependency-LDA performs significantly better than both Prior-LDA and Flat-LDA across all datasets on all 13 evaluation metrics across Figs. 6 and 7. For the simpler LDA models, Prior-LDA outperformed Flat-LDA on the EUR-Lex (12/13), Yahoo! *Arts* (13/13) and RCV1-v2 (12/13) datasets whereas performance on NYT and Yahoo! *Health* was more evenly split. In almost all cases, the absolute differences between the Prior-LDA and Flat-LDA scores is much smaller than the differences between either of them and Dependency-LDA. The scale of the differences between the three LDA-based models demonstrates that Dependency-LDA is achieving its improved performance by successfully incorporating information beyond simple baseline label-frequencies.

Among the SVM models, the Tuned-SVMs convincingly outperform Vanilla-SVMs on the three non power-law datasets. On NYT, the Tuned-SVMs generally outperformed Vanilla-SVMs (9/13), whereas they performed worse on the EUR-Lex dataset (3/12). Generally, in the cases in which there were significant differences between the two SVM approaches on the power-law datasets, the Vanilla-SVMs outperformed Tuned-SVMs on measures that emphasize the full range of ratings (such as the MARGIN and the Areas Under Curves), whereas Tuned-SVMs outperformed Vanilla-SVMs on metrics emphasizing the top-ranked predictions (such as the ONE-ERROR and ISERROR metrics). This overall pattern indicates that Tuned-SVMs may generally make better predictions among the top-ranked labels but have difficulty calibrating predictions for the lower-ranked labels (which will in general be largely comprised of infrequent labels). Thus, the observed contrast between overall SVM performance on the EUR-Lex and NYT datasets may reflect the fact that predictions for the NYT dataset were evaluated on only labels that showed up in test documents (thereby excluding many of the infrequent labels from these rankings), whereas predictions for EUR-Lex were evaluated across *all* labels. This observation is supported by performance of the SVMs on the benchmark datasets, on which Tuned-SVMs clearly outperform Vanilla-SVMs; on these datasets, there are many fewer total labels to rank, and a much higher percentage of these labels is present in the test-documents, so therefore the scores on these datasets are much less influenced by low-ranked labels.

Document-Pivoted Ranking Predictions

Dataset	Model	ROC Analyses ⇑			MultiLabel Metrics ⇓			
POWER-LAW DATASETS								
		AUC_{PR}	AUC_{ROC}	Avg-Prec	Rnk-Loss	One-Err	Is-Err	Margin
NYT	$SVM_{Vanilla}$	.449	.984	.468	1.61	30.5	98.1	148
	SVM_{Tuned}	.477	.965	.492	3.51	21.2	97.0	282
	$LDA_{Dependency}$	**.612**	**.991**	**.631**	**.93**	**16.6**	**94.3**	**99**
	LDA_{Prior}	.518	.977	.537	2.25	21.3	97.6	233
	LDA_{Flat}	.514	.981	.533	1.95	20.2	97.5	198
EURLex	$SVM_{Vanilla}$	.435	.975	.454	2.51	37.5	98.1	387
	SVM_{Tuned}	.416	.967	.430	3.28	**31.6**	98.2	436
	$LDA_{Dependency}$	**.492**	**.982**	**.511**	**1.77**	32.0	**97.2**	**269**
	LDA_{Prior}	.387	.949	.402	5.15	34.7	98.6	708
	LDA_{Flat}	.380	.942	.396	5.78	35.6	98.8	841
NON POWER-LAW DATASETS								
Dataset	Model	ROC Analyses ⇑			MultiLabel Metrics ⇓			
		AUC_{PR}	AUC_{ROC}	Avg-Prec	Rnk-Loss	One-Err	Is-Err	Margin
Y! *Arts*	$SVM_{Vanilla}$	.553	.828	.565	17.15	55.5	68.3	4.28
	SVM_{Tuned}	.615	.833	.625	16.71	**44.2**	**60.9**	4.28
	$LDA_{Dependency}$	**.619**	**.855**	**.630**	**14.51**	45.4	62.4	**3.76**
	LDA_{Prior}	.607	.853	.619	14.67	46.8	64.6	3.87
	LDA_{Flat}	.579	.810	.589	18.99	47.1	66.7	5.01
Y! *Health*	$SVM_{Vanilla}$	.682	.887	.694	11.30	44.1	58.0	2.21
	SVM_{Tuned}	.779	.898	.788	10.17	**24.3**	**43.0**	2.01
	$LDA_{Dependency}$	**.795**	**.926**	**.805**	**7.45**	24.7	44.1	**1.52**
	LDA_{Prior}	.738	.909	.750	9.06	34.3	53.9	1.89
	LDA_{Flat}	.744	.893	.757	10.66	27.0	53.1	2.20
RCV1	$SVM_{Vanilla}$	.865	.987	.876	1.32	5.85	44.3	3.33
	SVM_{Tuned}	**.888**	**.988**	**.896**	**1.19**	**5.82**	**37.5**	**2.87**
	$LDA_{Dependency}$	.863	.987	.873	1.32	7.14	42.9	3.13
	LDA_{Prior}	.686	.967	.711	3.32	14.78	88.1	9.49
	LDA_{Flat}	.587	.939	.608	6.08	22.08	87.6	15.15

Fig. 6 Document-Pivoted-Ranking-Predictions. For each dataset and model, we present scores on all rank-based evaluation metrics. These have been grouped in accordance with how they are used in the literature (where the first three evaluation metrics are used in ROC-analysis literature, and the remaining four metrics are used in used in the label-ranking literature). We note again that $\text{RANKLOSS} = 100 \times (1 - AUC_{ROC})$; we provide results for both metrics for ease of comparison with published results

6.1.2 Comparison between LDA-based and SVM-based models (Doc-pivoted)

Looking across all document-pivoted model results, one can see a clear distinction between the relative performance of LDA and SVMs on the power law datasets vs. the non power-law datasets. The Dependency-LDA model clearly outperforms SVMs on the power-law datasets (on 13/13 measures for NYT, and on 12/13 measures on EUR-Lex). Note that on the NYT dataset, which has the most skewed label-frequency distribution and the largest cardinality, both the Prior-LDA and the Flat-LDA methods outperform the Tuned-SVMs as well.

On the non power-law datasets, results are more mixed. For rank-based metrics on both of the Yahoo! datasets, Dependency-LDA outperforms SVMs on the five measures which emphasize the full range of rankings, but are outperformed by Tuned-SVM's on the measures emphasizing the very top-ranked labels (namely, the One-Error and Is-Error measures). For binary evaluations, Dependency-LDA generally outperforms Tuned-SVMs on the *Health* dataset (5/6) but performs worse on the *Arts* dataset. On the RCV1 dataset, Tuned-SVMs

Document-Pivoted Binary Predictions

Dataset	Model	N - PROPORTIONAL		N - CALIBRATED		N - BEP	
		$F1_{MACRO}\Uparrow$	$F1_{MICRO}\Uparrow$	$F1_{MACRO}\Uparrow$	$F1_{MICRO}\Uparrow$	$F1_{MACRO}\Uparrow$	$F1_{MICRO}\Uparrow$
POWER-LAW DATASETS							
NYT	$SVM_{Vanilla}$	.402	.404	.415	.424	.540	.483
	SVM_{Tuned}	.453	.453	.470	.469	.580	.481
	$LDA_{Dependency}$	**.542**	**.539**	**.566**	**.564**	**.676**	**.652**
	LDA_{Prior}	.477	.473	.494	.489	.608	.575
	LDA_{Flat}	.473	.469	.490	.483	.603	.565
EURLex	$SVM_{Vanilla}$	.406	.409	.417	.420	.537	.417
	SVM_{Tuned}	.402	.405	.420	.421	.526	.324
	$LDA_{Dependency}$	**.458**	**.461**	**.468**	**.471**	**.586**	**.508**
	LDA_{Prior}	.387	.389	.403	.402	.512	.379
	LDA_{Flat}	.381	.383	.396	.396	.506	.383
NON POWER-LAW DATASETS							
Dataset	Model	N - PROPORTIONAL		N - CALIBRATED		N - BEP	
		$F1_{MACRO}\Uparrow$	$F1_{MICRO}\Uparrow$	$F1_{MACRO}\Uparrow$	$F1_{MICRO}\Uparrow$	$F1_{MACRO}\Uparrow$	$F1_{MICRO}\Uparrow$
Y! *Arts*	$SVM_{Vanilla}$	.376	.339	.420	.397	.648	.502
	SVM_{Tuned}	**.461**	**.425**	**.508**	**.482**	.689	.519
	$LDA_{Dependency}$	.454	.416	.494	.464	**.698**	**.548**
	LDA_{Prior}	.448	.406	.479	.439	.690	.545
	LDA_{Flat}	.438	.403	.464	.431	.660	.496
Y! *Health*	$SVM_{Vanilla}$	.463	.428	.574	.573	.763	.687
	SVM_{Tuned}	.617	**.580**	.693	.670	.824	.724
	$LDA_{Dependency}$	**.619**	.577	**.700**	**.675**	**.841**	**.766**
	LDA_{Prior}	.543	.503	.629	.613	.803	.736
	LDA_{Flat}	.594	.559	.633	.605	.796	.710
RCV1	$SVM_{Vanilla}$	.745	.736	.809	.797	.883	.863
	SVM_{Tuned}	**.767**	**.757**	**.840**	**.828**	**.903**	**.880**
	$LDA_{Dependency}$	.743	.733	.810	.793	.881	.852
	LDA_{Prior}	.572	.562	.582	.572	.731	.703
	LDA_{Flat}	.485	.479	.515	.503	.658	.603

Fig. 7 Document-pivoted binary predictions. For each dataset and model, we present the Micro- and Macro-F1 scores achieved using the three different cutoff-point methods (from left to right: PROPORTIONAL, CALIBRATED, and BEP). Note that the absolute difference between the Micro and Macro scores for a model are generally smaller for the document-pivoted results than for the label-pivoted evaluations; this is due to the relatively low variability in the number of labels per document (as opposed to the generally large variability in the number of documents per label)

have a clear advantage over all of the LDA models (outperforming them across all 13 measures).

6.1.3 Relationship between model performance and dataset statistics (Doc-pivoted)

The overall pattern of results indicates that there is a strong interaction between the statistics of the datasets and the performance of LDA-based models relative to SVM-based models. These effects are illustrated in Figs. 8 and 9. To help illustrate the *relative* performance differences between models, the results within each dataset have been centered around zero in these figures (without the centering, it is more difficult to see the interaction between the datasets and models, since most of the variance in model performance is accounted for by the main effect of the datasets).

In Fig. 8, performance on each of the five datasets has been plotted in order of the dataset's median label-frequency (i.e., the median number of documents per label). One

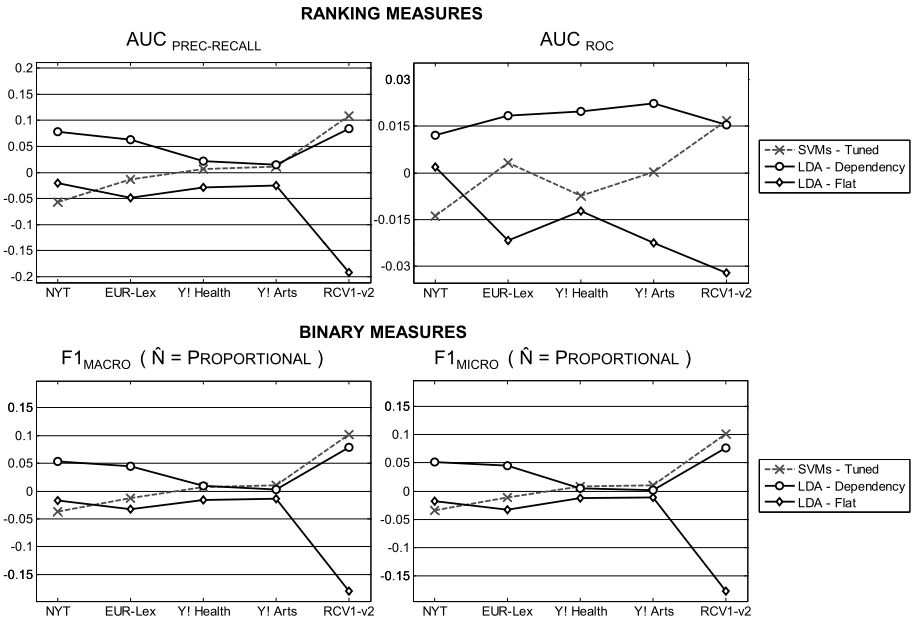


Fig. 8 *Dataset Label-Frequency and Model Performance*: Relative performance of Tuned-SVMs, Dependency-LDA, and Flat-LDA on several of the evaluation metrics for document-pivoted predictions. Datasets are ordered in terms of their median label-frequencies (the median number of documents-per-label increases from *left* to *right*). Scores have been centered around zero in order for each dataset to emphasize *relative* performance of the models. As the amount of training data per label decreases, performance for LDA-based models tends to improve relative to SVM performance

can see that as the amount of training data increases, the performance of Dependency-LDA *relative* to Tuned-SVMs drops off and eventually becomes worse. A similar pattern exists for Flat-LDA. Note that although both LDA-based models are worse than Tuned-SVMs on the RCV1-v2 dataset (which has the most training data), Dependency-LDA performance is in the range of Tuned-SVMs, whereas Flat-LDA performs drastically worse.

Figure 9 plots the same results as a function of dataset Cardinality (i.e., the average number of labels per document). Here, one can see that the relative performance improvement for Dependency-LDA over Flat-LDA increases as the number of labels per document increases. Since both Flat-LDA and Dependency-LDA use the same set of label-word distributions learned during training, this performance boost can only be attributed to inference for Dependency-LDA at test time (where unlike Flat-LDA, Dependency-LDA accounts for the dependencies between labels). These results are consistent with the intuition that it is increasingly important to account for label-dependencies as the number of labels per document increases.

6.2 Label-pivoted results

The label-pivoted predictions provide a ranking of all documents in terms of their relevance to each label c. The seven *ranking*-based metrics directly evaluate aspects of each of these rankings. The six *binary* metrics evaluate the binary predictions after the rankings have been partitioned into positive and negative documents for each label, using the three aforemen-

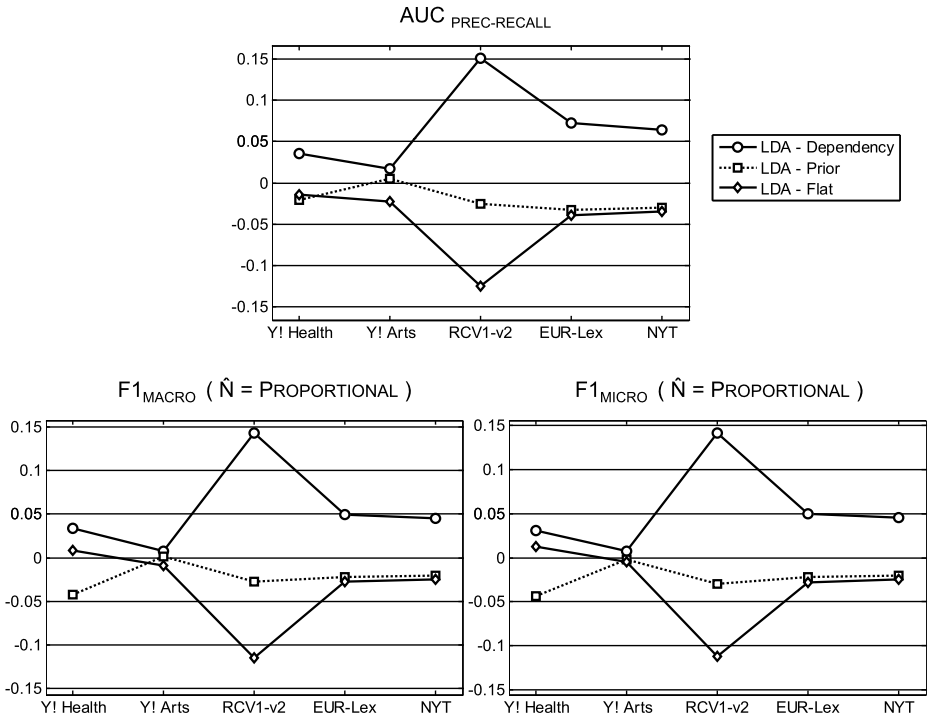


Fig. 9 *Dataset Cardinality and Model Performance*: Relative performance of the three LDA-based models on several of the evaluation metrics for document-pivoted predictions (evaluation metrics for the binary predictions are shown in the *top row*, and a rank-based evaluation metric is shown *below*). Datasets are ordered in terms of their cardinality (the average number of labels-per-document increases from *left* to *right*). Scores have been centered around zero in order for each dataset to emphasize *relative* performance of the models. As the average number of labels-per document increases, the relative improvement of Dependency-LDA over the simpler LDA-based models increases

tioned cutoff-points. Results for the rank-based evaluations are shown in Figure 10, and results for the binary predictions are shown in Fig. 11.

6.2.1 Comparison within LDA-based and SVM-based models (Label-pivoted)

The relative performance among the LDA-based models follows a similar pattern to what was observed for the document-pivoted predictions. Dependency-LDA consistently outperforms both Prior-LDA and Flat-LDA, beating them on nearly every measure on all five datasets.

The improvement achieved by Dependency-LDA seems generally to be related to the number of labels per document; there is a very large performance gap in the power-law datasets (which have about 5.5 labels per document each on average), whereas this gap is relatively smaller on the Yahoo! datasets (which have on average 1.6 labels per document). The improvement observed for RCV1 is nearly as large or even larger than for the power-law datasets, which may be in part due to the automated, rule-based assignment of labels in the dataset's construction (which introduces very strict dependencies in the true label-assignments).

Label-Pivoted Ranking Predictions

Dataset	Model	ROC Analyses ⇑			MultiLabel Metrics ⇓			
		AUC_{PR}	AUC_{ROC}	Avg-Prec	Rnk-Loss	One-Err	Is-Err	Margin
POWER-LAW DATASETS								
NYT	$SVM_{Vanilla}$	.302	**.960**	.309	**4.05**	59.4	94.1	2746
	SVM_{Tuned}	.302	.959	.309	4.05	59.3	94.2	2750
	$LDA_{Dependency}$	**.376**	.958	**.383**	4.20	**49.5**	**92.2**	**2634**
	LDA_{Prior}	.350	.913	.356	8.66	50.3	92.6	4089
	LDA_{Flat}	.347	.918	.353	8.18	50.3	92.7	4067
EURLex	$SVM_{Vanilla}$	.450	.959	.459	4.14	51.4	84.3	193
	SVM_{Tuned}	.456	**.960**	.466	**4.03**	51.2	83.9	**184**
	$LDA_{Dependency}$	**.463**	.958	**.472**	4.18	**49.5**	**81.9**	193
	LDA_{Prior}	.398	.880	.404	12.00	53.4	83.7	480
	LDA_{Flat}	.395	.881	.402	11.91	53.6	84.0	482
NON POWER-LAW DATASETS								
Dataset	Model	ROC Analyses ⇑			MultiLabel Metrics ⇓			
		AUC_{PR}	AUC_{ROC}	Avg-Prec	Rnk-Loss	One-Err	Is-Err	Margin
Y! *Arts*	$SVM_{Vanilla}$	.297	.751	.298	24.89	28.4	100	6370
	SVM_{Tuned}	.329	**.757**	.330	**24.25**	**27.4**	100	6367
	$LDA_{Dependency}$	**.339**	.755	**.341**	24.49	44.2	100	**6355**
	LDA_{Prior}	.332	.748	.333	25.24	46.3	100	6378
	LDA_{Flat}	.328	.749	.329	25.09	46.3	100	6377
Y! *Health*	$SVM_{Vanilla}$	.541	.846	.542	15.38	20.0	100	**7965**
	SVM_{Tuned}	**.569**	.849	**.570**	15.08	**14.3**	100	7968
	$LDA_{Dependency}$	.568	**.850**	.569	**15.03**	17.1	100	7988
	LDA_{Prior}	.526	.820	.526	18.04	17.1	100	8016
	LDA_{Flat}	.513	.813	.514	18.69	15.7	100	8013
RCV1	$SVM_{Vanilla}$	.598	.979	.599	2.08	16.8	100	47953
	SVM_{Tuned}	**.607**	**.981**	**.608**	**1.90**	**13.9**	100	**46233**
	$LDA_{Dependency}$	.558	.971	.559	2.91	16.8	100	49130
	LDA_{Prior}	.491	.940	.492	6.04	17.8	100	56497
	LDA_{Flat}	.488	.938	.489	6.17	**13.9**	100	56156

Fig. 10 Label-Pivoted-Ranking-Predictions. For all rank-based evaluation metrics, we present results for the label-pivoted model predictions. Note that the IS-ERROR measure is not well-suited for the label-pivoted results on the non power-law datasets. Specifically, since all labels have numerous test-instances, and the number of documents is very large, it is extremely difficult to predict a perfect ordering of all documents for *any* labels. In fact, none of the models assigned a perfect ordering for a single label, which is why all scores are 100. We have nonetheless included these results for completeness

Tuned-SVMs consistently outperformed Vanilla-SVMs on all datasets except for NYT, where the two methods show nearly equivalent performance overall. This is notable in that it indicates that the NYT dataset poses a problem for binary SVMs which parameter tuning cannot fix; in other words, it suggests that there is some feature of this dataset which binary SVMs have an intrinsic difficulty dealing with. Since the straightforward answer—given what we have seen, as well as our motivations presented in the introduction—is that this difficulty relates to the power-law statistics of the NYT dataset, it is somewhat surprising that there is not a similar effect for the EUR-Lex dataset (on which the Tuned-SVMs outperform Vanilla-SVMs on all measures). Why should these two datasets, both of which have fairly similar statistics, show different improvement due to parameter tuning?

We conjecture that the differences in the effect of parameter tuning between the NYT and EUR-Lex datasets are misleading. First, although Tuned-SVMs achieve better scores on all measures for EUR-Lex, the *scale* of these differences is actually quite small. Secondly, and perhaps more importantly, some of these differences are likely to be due to the relative proportion of training vs. testing data between the two datasets. For EUR-Lex, only

Label-Pivoted Binary Predictions

Dataset	Model	N - PROPORTIONAL		N - CALIBRATED		N - BEP	
		$F1_{MACRO}\Uparrow$	$F1_{MICRO}\Uparrow$	$F1_{MACRO}\Uparrow$	$F1_{MICRO}\Uparrow$	$F1_{MACRO}\Uparrow$	$F1_{MICRO}\Uparrow$
POWER-LAW DATASETS							
NYT	$SVM_{Vanilla}$	.270	.481	.288	.492	.380	.115
	SVM_{Tuned}	.270	.487	.288	.497	.380	**.116**
	$LDA_{Dependency}$	**.325**	**.541**	**.350**	**.552**	**.444**	.112
	LDA_{Prior}	.308	.501	.335	.512	.412	.047
	LDA_{Flat}	.304	.499	.333	.509	.410	.051
EURLex	$SVM_{Vanilla}$	.368	.465	.389	.489	.528	.125
	SVM_{Tuned}	.373	**.471**	.395	**.495**	.534	**.128**
	$LDA_{Dependency}$	**.382**	.467	**.409**	.492	**.537**	.124
	LDA_{Prior}	.337	.405	.360	.427	.466	.043
	LDA_{Flat}	.334	.402	.357	.424	.464	.044
NON POWER-LAW DATASETS							
Y! *Arts*	$SVM_{Vanilla}$	.325	.428	.324	.429	.350	.420
	SVM_{Tuned}	.355	**.454**	.357	**.457**	.376	**.448**
	$LDA_{Dependency}$	**.367**	.451	**.368**	.452	**.385**	.439
	LDA_{Prior}	.358	.440	.359	.442	.378	.419
	LDA_{Flat}	.355	.435	.354	.437	.373	.417
Y! *Health*	$SVM_{Vanilla}$	.548	.638	.553	.641	.571	.650
	SVM_{Tuned}	**.571**	**.656**	**.575**	**.658**	**.593**	**.669**
	$LDA_{Dependency}$	.562	.646	.567	.649	.589	.659
	LDA_{Prior}	.521	.610	.526	.611	.544	.617
	LDA_{Flat}	.512	.599	.517	.601	.532	.610
RCV1	$SVM_{Vanilla}$	.571	.780	.585	.784	.600	.782
	SVM_{Tuned}	**.579**	**.787**	**.594**	**.790**	**.609**	**.788**
	$LDA_{Dependency}$	.539	.762	.553	.764	.568	.750
	LDA_{Prior}	.484	.629	.496	.632	.510	.595
	LDA_{Flat}	.482	.617	.495	.619	.508	.602

Fig. 11 Label-Pivoted-Binary-Predictions. For each set of results, we present the Micro-F1 and Macro-F1 scores achieved using the three different cutoff-point methods (from *left* to *right*: PROPORTIONAL, CALIBRATED, and BEP). Note that the only results representing *true* performance are the PROPORTIONAL results, and thus these are the values which should be used for comparison with benchmarks presented in the literature (although for model comparison, all values are useful, since they easily calculated from model output)

one-tenth of the documents are in each test-set, whereas NYT has a roughly 50-50 train-test split. As a result, far fewer rare-labels are tested in any given split of EUR-Lex (since a label is only included in the label-wise evaluations if it appears in both the train and test-data). Thus, the EUR-Lex splits somewhat de-emphasize performance on rare labels. This assertion is strongly supported by the Document-pivoted results for EUR-Lex (in which *all* labels that appeared in the training set must be ranked, and thus, influence the performance scores); Tuned-SVMs perform worse than Vanilla SVMs on 10/13 of the Document-Pivoted evaluation metrics for EUR-Lex. Overall, it appears that the intrinsic difficulties that SVMs have on rare labels is a problem with both NYT and EUR-Lex, and that the observed differences between Tuned and Vanilla-SVMs on these two datasets is likely due in part to the differences in the construction of the datasets.

6.2.2 Comparison between LDA-based and SVM-based models (Label-pivoted)

The performance of Dependency-LDA relative to SVMs was highly dependent on the dataset. On the power-law datasets, Dependency-LDA generally outperformed SVMs;

Dependency-LDA outperformed Tuned-SVMs on 10/13 measures for the NYT dataset and on 7/13 measures for the EUR-Lex dataset.

Of special interest is the Macro-F1 measures since Macro-averaging gives equal weight to all labels (regardless of their frequency in the test set). Since power-law datasets are dominated by rare labels, the Macro-F1 measures reflect performance on rare labels. On EUR-Lex, Dependency-LDA outperforms the SVMs for all Macro-F1 measures. On NYT, all three LDA models outperform the SVMs for all Macro-F1 measures. This supports the hypothesis—motivated in our introduction—that LDA is able to handle rare labels better than binary SVMs.

On the non power-law datasets, results were much more mixed, with SVMs generally outperforming Dependency-LDA. Dependency-LDA was competitive with Tuned-SVMs for the *Arts* subset, but generally inferior in performance on the *Health* subset. Performance was even worse on the RCV1-v2 dataset where both SVM methods clearly outperformed all LDA-based methods. Some of the variability in performance on the three datasets may be due to the amount of training data per label. RCV1-v2 has the most training data per label (despite containing more labels) and on this dataset the SVM methods dominate the LDA methods. The *Arts* subset has the least amount of training data per label and on this dataset the LDA methods fair better.

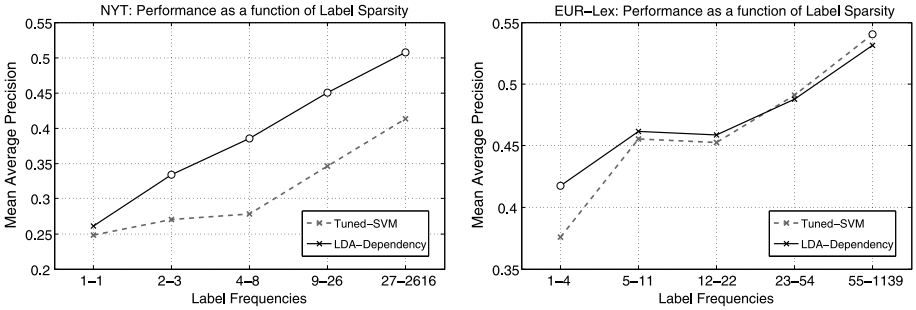
Again, it is of interest that on the *Arts* subset Dependency-LDA dominates the SVM methods on the Macro-F1 measures. In fact, the PROPORTIONAL Macro-F1 scores for this dataset seem to be higher than any of the Macro-F1 scores previously reported in the literature (including the large set of results for discriminative methods published by Ji et al. (2008), which includes a method that accounts for label-dependencies); see Appendix D for additional comparisons.

6.3 Comparing algorithm performance across label frequencies

As discussed in the introduction, there are reasons to believe that LDA-based models should have an advantage over one-vs-all binary SVMs on labels with sparse training data. To address this question, we can look at the relative performance of the models as a function of the amount of training data. Figures 12(a) and 12(b) compare the average precision scores for Dependency-LDA and Tuned-SVMs across labels with different training frequencies in the NYT and EUR-Lex datasets, respectively. To compute these scores, labels were first binned according to their quintile in terms of training frequency, and the Macro-average of the average precision scores was computed for each label within each bin. For each bin, significance was computed via a paired t-test.[18]

On both datasets, it is clear that Dependency-LDA has a significant advantage over Tuned-SVMs on the rarest labels. On the EUR-Lex dataset, Dependency-LDA significantly outperforms SVMs on labels with training frequencies of less than five, and performs better than SVMs (though not significantly at the $\alpha = .05$ level) on the three lower quintiles of label frequencies. SVM performance catches up to Dependency-LDA on labels somewhere in the upper-middle range of label-frequencies, and surpasses Dependency-LDA (significantly) for the labels in the most frequent quintile. On the NYT dataset, Dependency-LDA outperforms SVMs across all label frequencies (this difference is significant on all quintiles except the one containing labels with a frequency of one).

[18]To be precise: The performance score for SVMs and Dependency LDA on each label with a training frequency in the appropriate range, for each split of the dataset, was treated as single a pair of values for the t-test.



(a) NYT (b) EUR-Lex

Fig. 12 Mean Average Precision scores for the NYT and EUR-Lex datasets as a function of the number of training documents per label. For each dataset, labels have been binned into quintiles by training frequency. Performance scores are macro-averaged across all labels within each of the bins. Circle ('○') markers indicate where the differences were statistically significant at the $\alpha = .05$ level as determined by pairwise t-tests within each bin. (In all cases in which the difference was significant: $p < .001$)

6.4 Summary: dependency-LDA vs. tuned SVMs

There are several key points which are evident from the experimental results presented above. First, the Dependency-LDA model significantly outperforms the simpler Prior-LDA and Flat-LDA models, and that the scale of this improvement depends on the statistics of the datasets. Secondly, under certain conditions, the LDA-based models (and most notably, Dependency-LDA) have a significant advantage over the binary SVM methods, but under other conditions the SVMs have a significant advantage. We have already discussed some of the specific factors that play a role in these differences. However, it is useful to take a step back, and consider the key model comparisons across all four of the prediction tasks. Namely, we wish to more generally explore the conditions in which probabilistic generative models such as topic models may have benefits compared to discriminative approaches such as SVMs, and vice-versa. To this purpose, we now focus on the *overall* performance of our best LDA-based approach (Dependency-LDA) and our best discriminative approach (Tuned-SVMs), rather than focusing on performance with respect to specific evaluation metrics.

In Fig. 13, we present a summary of the performance for Dependency-LDA and Tuned-SVMs across all four prediction tasks and all five datasets. For each dataset and prediction task, we present the total number of evaluation metrics for which each model achieved the *best* score out of all five of our models (in the case of ties, both models are awarded credit).[19] The results have been ordered from top-to-bottom by the relative amount of training data there is in each dataset. Note that these datasets fall into three qualitative categories: (1) the power-law datasets (NYT and EUR-Lex), (2) the Yahoo! datasets (which are not highly multi-label, and do not have large amounts of training data per label), and (3) the RCV1-V2 dataset, which has a large amount of training data for each label, and is more highly multi-label than the Yahoo! datasets but less than the power-law datasets (and, additionally, unlike the other datasets, had many algorithmically-assigned labels).

[19] Note that although we presented seven rank-based evaluation metrics in the previous tables, the maximum score for each element of the table is *six*, because we collapse the performance for the AUC_{ROC} and RANKLOSS metrics, due to their equivalence.

Dataset	Median Label Freq.	Mean Label Freq.	Model	DOCUMENT-PIVOTED Rankings (6)	DOCUMENT-PIVOTED Binary (6)	LABEL-PIVOTED Rankings (6)	LABEL-PIVOTED Binary (6)	TOTALS Doc-Pivot (12)	TOTALS Label-Pivot (12)	TOTALS Total (24)
NYT	3	40	$LDA_{Dependency}$	**6**	**6**	**5**	**5**	**12**	**10**	**22**
			SVM_{Tuned}	0	0	0	1	0	1	1
EURLex	6	26	$LDA_{Dependency}$	**5**	**6**	**4**	**3**	**11**	**7**	**18**
			SVM_{Tuned}	1	0	2	**3**	1	5	6
Y! Arts	530	636	$LDA_{Dependency}$	**4**	2	**4**	**3**	**6**	**7**	**13**
			SVM_{Tuned}	2	**4**	3	**3**	**6**	6	12
Y! Health	500	1,047	$LDA_{Dependency}$	**4**	**5**	2	0	**9**	2	11
			SVM_{Tuned}	2	1	**4**	**6**	3	**10**	**13**
RCV1	7,410	25,310	$LDA_{Dependency}$	0	0	1	0	0	1	1
			SVM_{Tuned}	**6**	**6**	**6**	**6**	**12**	**12**	**24**

Fig. 13 Summary comparison of the performance of Dependency-LDA vs. Tuned SVMs across the five datasets. For each type of prediction (Document/Label Pivoted), we show the number of evaluation metrics on which each model achieved the best overall score. Performance is first broken down by the type of evaluation metric used (Rank-Based vs. Binary). Totals are shown in the three right columns. Note that *six* is the maximum achievable value here for both binary and rank-based predictions; although seven rank-based scores were presented in previous tables, the AUC_{ROC} and RANK-LOSS metrics have been combined here

Looking at the full totals in the rightmost column of Fig. 13, one can see that for the power-law datasets, Dependency-LDA has a significant overall advantage over SVMs. For the two Yahoo! datasets, the overall performance of the two models is quite comparable. Finally, for the RCV1-V2 dataset, Tuned SVMs clearly outperform Dependency-LDA. This general interaction between the amount of training data and the relative performance of these two models has been discussed earlier in the paper, but is perhaps most clearly illustrated in this simple figure.

A second feature that is evident in Fig. 13 is that, all else being equal, the Dependency-LDA model seems better suited for Document-Pivoted predictions and SVMs seem better suited for Label-Based predictions. For example, although Dependency-LDA greatly outperforms SVMs overall on EUR-Lex, the performance for Label-Pivoted predictions on this dataset are in fact quite close. And although overall performance is quite similar for the Yahoo! *Health* dataset, Dependency-LDA dominates SVMs for Document-pivoted predictions, and the reverse is true for Label-pivoted predictions. A likely explanation for this difference is the fundamentally different way that each model handles multi-label data. In Dependency-LDA (and all of the LDA-based models), although we learn a model for each label during training, at test time it is the *documents* that are being modeled. Thus the "natural direction" for LDA-based models to make predictions is *within* each document, and *across* the labels. The SVM approach, in contrast, builds a binary classifier for each label, and thus the "natural direction" for Binary SVMs to make predictions is *within* each label, and *across* documents. Thus, if one is to consider which type of classifier would be preferable for a given application, it seems important to consider whether label-pivoted or document-pivoted predictions are more suited to the task, in addition to what the statistics of the corpus look like.

7 Conclusions

In conclusion, in terms of the three LDA-based models considered in this paper, our experiments indicate that (1) Prior-LDA improves performance over the Flat-LDA model by accounting for baseline label-frequencies, (2) Dependency-LDA significantly improves performance relative to both Flat-LDA and Prior-LDA by accounting for label dependencies,

and (3) The relative performance improvement that is gained by accounting for label dependencies is much larger in datasets with large numbers of labels per document.

In addition, the results of comparing LDA-based models with SVM models indicate that on large-scale datasets with power-law like statistics, the Dependency-LDA model generally outperforms binary SVMs. This effect is more pronounced for document-pivoted predictions, but is also generally the case for label-pivoted predictions. The results of label-pivoted predictions across different label-frequencies indicate that the performance benefit observed for Dependency-LDA is in part due to improved performance on rare labels.

Our results with SVMs are consistent with those obtained elsewhere in the literature; namely, binary SVM performance degrades rapidly as the amount of training data decreases, resulting in relatively poor performance on large scale datasets with many labels. Our results for the LDA-based methods, most notably for the Dependency-LDA model, indicate that probabilistic models are generally more robust under these conditions. In particular, the comparison of Dependency-LDA and SVMs on labels at different training frequencies demonstrates that Dependency-LDA clearly outperformed SVMs on the rare labels on our large scale datasets. Additionally, Dependency-LDA was competitive with, or better than, SVMs on labels across all training frequencies on these datasets (except on the most frequent quintile of labels in the EUR-Lex dataset). Furthermore, Dependency-LDA clearly outperformed SVMs on the document-pivoted predictions on both large scale datasets.

Robustness in the face of large numbers of labels and small numbers of training documents has not been extensively commented on in the literature on multi-label text classification, since the majority of studies have focused on corpora with relatively few labels, and many examples of each label. Given that human labeling is an expensive activity, and that many annotation applications consist of a large number of labels with a long tail of relatively rare labels, prediction with large numbers of labels is likely to be an increasingly important problem in multi-label text classification and one that deserves further attention.

A potentially useful direction for future research is to combine discriminative learning with the types of generative models proposed here, possibly using extensions of existing discriminative adaptations of the LDA model (e.g., Blei and McAuliffe 2008; Lacoste-Julien et al. 2008; Mimno and McCallum 2008). A hybrid approach could combine the benefits of generative LDA models—such as explaining away, natural calibration for sparse data, semi-supervised learning (e.g., Druck et al. 2007), and interpretability (e.g., Ramage et al. 2009)—with the advantages of discriminative models such as task-specific optimization and good performance under conditions with many training examples. The approach we propose can also be applied to domains outside of text classification; for example, it can be applied to multi-label images in computer vision (Cao and Fei-fei 2007).

Acknowledgements The authors would like to thank the anonymous reviewers of this paper, as well as the guest editors of this issue of the Machine Learning Journal, for their helpful comments and suggestions for improving this paper. This material is based in part upon work supported by the National Science Foundation under grant number IIS-0083489, by the Intelligence Advanced Research Projects Activity (IARPA) via Air Force Research Laboratory contract number FA8650-10-C-7060, by the Office of Naval Research under MURI grant N00014-08-1-1015, by a Microsoft Scholarship (AC), by a Google Faculty Research award (PS). The US Government is authorized to reproduce and distribute reprints for Government purposes notwithstanding any copyright annotation thereon. Disclaimer: the views and conclusions contained herein are those of the authors and should not be interpreted as necessarily representing the official policies or endorsements, either expressed or implied, of NSF, IARPA, AFRL, ONR, or the US Government.

Appendix A: Details of experimental datasets

A.1 The New York Times (NYT) Annotated Corpus

The New York Times Annotated Corpus (available from the Linguistic Data Consortium) contains nearly every article published by The New York Times over the span of 10 years from January 1st, 1987 to June 19th, 2007 (Sandhaus 2008). Over 1.5 million of these articles have "descriptor" tags that were manually assigned by human labelers via the New York Times Indexing Service, and correspond to the subjects mentioned within each article[20] (see Tables 1 and 2 for numerous examples of these descriptors).

To construct an experimental corpus of NYT documents, we selected all documents that had both text in their body and at least three descriptor labels from the "News\U.S" taxonomic directory. After removing common stopwords, we randomly selected 40% of the articles for training and reserved the remaining articles for testing. Any test article containing label(s) that did not occur in the training set was then re-assigned to the training set so that all labels had at least one positive training instance. This procedure resulted in a training corpus containing 14,669 documents and 4,185 unique labels, and a test corpus with 15,989 documents and 2,400 unique labels.

For feature selection in all models, we removed words that appeared fewer than 20 times within the training data, which left us with a vocabulary of 24,670 unique words. For this dataset, evaluation on test documents was restricted to the subset of 2,400 labels that occurred at least once in both the training and test sets (this approach for handling missing labels is consistent with common practice in the literature).

A.2 The EUR-Lex text dataset

The EUR-Lex text collection (Loza Mencía and Fürnkranz 2008b) contains documents related to European Union law (e.g. treaties, legislation), downloaded from the publicly available EUR-Lex online repository (EUR 2010). The dataset we downloaded contained 19,940 documents and 3,993 EUROVOC descriptors. Note that there are additional types of meta-data available in the dataset, but we restricted our analysis to the EUROVOC descriptors, which we will refer to as labels.

The dataset provides 10 cross-validation splits of the documents into training and testing sets equivalent to those used in Loza Mencía and Fürnkranz (2008b). We downloaded the stemmed and tokenized forms of the documents and performed our own additional preprocessing of the data splits. For each split, we first removed all empty documents (documents with no words),[21] and then removed all words appearing fewer than 20 times within the training set. This was done independently for each split so that no information about test documents was used during training.

[20] Note that additional types of meta-data are available for many of these documents. This includes additional labeling schemes, such as the "general online descriptors" —which are algorithmically assigned—and that for the purposes of this paper we specifically used the hand-assigned "descriptor" tags. We refer to these "descriptor" tags as "labels" for consistency throughout the paper.

[21] We note that after removing empty documents, we were left with 19,348 documents. The dataset statistics in terms of the EUROVOC descriptors (shown in Table 4) however are based on the 19,800 documents for which there was at least one descriptor assigned to the document.

A.3 The RCV1-v2 dataset

The RCV1-v2 dataset (Lewis et al. 2004)—an updated version of the Reuters RCV1 corpus—is one of the more commonly used benchmark datasets used in multi-label document classification research (e.g., see Fan and Lin 2007; Fürnkranz et al. 2008; Tsoumakas et al. 2009). The dataset consists of over 800,000 newswire stories that have been assigned one or more of the 103 available labels (categories). We used the original training set from the *LYRL2004* split given by Lewis et al. (2004). Only 101 of the 103 labels are present in the 23,149 document training-set, and we employ the commonly-used convention of restricting our evaluation to these 101 labels. We randomly selected 75,000 of the documents from the *LYRL2004* test split for our test set.[22]

One problematic feature of the RCV1-v2 dataset is that many of the labels were not manually assigned by editors but were instead automatically assigned via automated expansion of the topic hierarchy (Lewis et al. 2004). Although it is possible to avoid evaluating predictions on these automatically-assigned labels—by only considering the subset of labels which are leaves within the topic hierarchy (Lewis et al. 2004)—these automatically-assigned labels still play a major role during training. Furthermore, this type of automated hierarchy expansion (although sensible) leads to some unnatural and perhaps misleading statistical features in the dataset. For example, although the average number of labels per document in the dataset is relatively large (which indicates that this is a highly multi-label dataset), the number of unique sets of labels is actually quite small relative to the number of documents, likely due to the fact that many of the documents were originally single-label and then automatically expanded such that they seemed multi-label. Although there is nothing inherently wrong with this approach, it (1) may lead to misleadingly positive results for models that are able to pick up on the automatically assigned labels, rather than the human-assigned labels, (2) leads to statistics which significantly deviate from the types of power-law distributions observed in many real-world situations, and (3) can lead one to assume that the dataset contains a much more complex space of label-combinations than is actually contained in the dataset. Note that, as illustrated in Table 4, the RCV1-V2 dataset is in most respects much more similar to the small Yahoo! subdirectory datasets than to the real-world power-law datasets.

A.4 The Yahoo! Subdirectory datasets

The Yahoo! datasets that we use consist of the *Arts* and the *Health* subdirectories from the collection used by Ueda and Saito (2002). We use the same training and test splits as presented in recent work by Ji et al. (2008) (where each training split consists of 1000 documents, and all remaining documents are used for testing). These datasets contain 19 and 14 unique labels respectively. The number of labels per document in each dataset is quite small; about 55–60% of training documents are assigned a single label, and about 85–90% are assigned either one or two labels. This was in large part due to the methods used to collect and pre-process the data, wherein only the second-level categories of the Yahoo! directory structure which had at least 100 examples were kept (Ji et al. 2008). We evaluated models across all of five of the available train/test splits for both the *Arts* and the *Health* sub-directories.

[22]Early experiments that we performed found that results on this subset were nearly identical to those for the full *LYRL2004* test set. The only score that is significantly different is the MARGIN for the label-pivoted results (because this metric is closely tied to the total number of documents in the test set).

Appendix B: Hyperparameter and sampling parameter settings for topic model inference

In this section, we present the complete set of parameter settings used for training and testing all LDA-based models, and motivate our particular choices for these settings. Note that all parameter settings were chosen heuristically were not optimized with respect to any of the evaluation metrics. It would be reasonable to expect some improvement in performance over the results presented in this paper by optimizing the hyperparameters via cross-validation on the training sets (as we did with Binary SVMs).

B.1 Hyperparameter settings

Table 6 shows the hyperparameter values that were used for training and testing the three LDA-based models on the five experimental datasets. Note that not all parameters are applicable for all models; for example, since Flat-LDA does not incorporate any ϕ' distributions of topics over labels, parameters such as β_C and γ do not exist in this model.

For all models, we used the same set of parameters to train the ϕ distributions of labels over words; $\eta = 50$, and $\beta_W = .01$. Early experimentation indicated that the exact values of η and β_W were generally unimportant as long as $\eta \gg 1$ and $\beta_W \ll 1$. The total strength of the Dirichlet prior on θ, which is dictated by η, is significantly larger than what is typically used in topic modeling. This makes sense in terms of the model; unlike in unsupervised LDA, we know a-priori which labels are present in the training documents, and setting a large value for η reflects this knowledge.

Parameters used to train the ϕ'_t distributions of topics over labels were chosen heuristically as follows. In Dependency-LDA, we first chose the number of topics (T). For the smaller datasets, we set the number of topics (T) approximately equal to the number of unique labels (C). For the two datasets with power-law like statistics, NYT and EUR-Lex, we set $T = 200$, which is significantly smaller than the number of unique labels. In addition to controlling for model complexity, some early experimentation indicated that setting $T \ll C$ improved the interpretability of the learned topic-label distributions in these datasets.[23]

Given the value of T for each dataset, we set β_C such that the *total* number of pseudocounts that were added to all topics was approximately equal to one-tenth of the total number of counts contributed by the observed labels. For example, each split of EUR-Lex contains approximately 90,000 label tokens in total. Given our choice of $T = 200$ topics, and the approximately 4,000 unique label types, by setting $\beta_C = .01$, the total number of pseudocounts that are added to all topics is $200 \times 4000 \times .01 = 8000$ (which is approximately one-tenth the total number of observed labels). For Prior-LDA, since there is only one topic ($T = 1$), we increased the value of β_C in order to be consistent with this general principle.

For setting the parameters for test documents, we kept the total number of pseudocounts that were added to the test documents consistent across all models. To help illustrate this, the

[23] Specifically, early in experimentation for NYT and EUR-Lex we trained a set of topics with $T = 50, 100, 200, 400$, and 1000. Visual inspection of the resulting topic-label distributions indicated that setting T to be too small (e.g., $T \leq 100$) over-penalized infrequent labels; labels that had fewer than approximately 25 training documents rarely had high probabilities in the model, even when the labels were clearly relevant to a topic. Setting T to be too large (e.g., $T = 1000$) led to both redundancy among the topics and to topics which appeared to be over-specialized (i.e., some of the topics had only a few documents with labels assigned to them).

Table 6 Hyperparameter values used for training and testing Dependency-LDA, Prior-LDA, and Flat-LDA, on all datasets. Note that the test-document parameters values for γ and α are given in terms of their *sums*; the actual pseudocount added to each element of θ'_d is γ/T and the flat pseudocount added to each element of θ_d is α/C

Dataset	Model	Training parameters						Testing parameters		
		Parameters for training Φ			Parameters for training Φ'			Parameters for test docs		
		η	α	β_W	T	β_C	γ	γ (*sum*)	η	α (*sum*)
NYT	Flat-LDA	50	0	.01	200	0.01	0.01	10	150	30
	Prior-LDA	50	0	.01	1	1	–	–	150	30
	Dependency-LDA	50	0	.01	–	–	—	–	–	180
EUR-Lex	Flat-LDA	50	0	.01	200	0.01	0.01	10	150	30
	Prior-LDA	50	0	.01	1	1	–	–	150	30
	Dependency-LDA	50	0	.01	–	–	–	–	–	180
Y! *Arts*	Flat-LDA	50	0	.01	20	1	0.01	1	100	1
	Prior-LDA	50	0	.01	1	10	–	–	70	30
	Dependency-LDA	50	0	.01	–	–	–	–	–	100
Y! *Health*	Flat-LDA	50	0	.01	20	1	0.01	1	100	1
	Prior-LDA	50	0	.01	1	10	–	–	70	30
	Dependency-LDA	50	0	.01	–	–	–	–	–	100
RCV1-V2	Flat-LDA	50	0	.01	100	1	0.01	10	100	1
	Prior-LDA	50	0	.01	1	100	–	–	100	1
	Dependency-LDA	50	0	.01	–	–	–	–	–	101

hyperparameter settings for test document parameters α and γ are shown in terms of their *sums* in Table 6, rather than in terms of their element-wise values. For the two power-law datasets, the total weight of the prior on θ was equal to 180, and for the three benchmark datasets the total weight of the prior on θ was equal to 100. We used smaller priors for the benchmark datasets because these documents were shorter on average, and we wished to keep the pseudocount totals roughly proportional to document lengths.

B.2 Details about sampling and predictions

Here we provide details regarding the number of chains and samples taken at each stage of inference (e.g., the total number of samples that were taken for each test document). These settings were equivalent for all three of the LDA-based models and for all datasets.

To train the C label-word distributions ϕ_c, we ran 48 independent MCMC chains (each initialized using a different random seed).[24] After a burn-in of 100 iterations we took a single sample at the end of each chain, where a sample consists of all z_i assignments for the training documents. These samples were then averaged to compute a single estimate for all ϕ_c distributions (as mentioned elsewhere in the paper, the same estimates of ϕ_c were used across all three LDA-based models).

[24]The exact number of chains is unimportant. However, it is well known that averaging multiple samples from an MCMC chain systematically improves parameter estimates. The particular number of chains that we ran (48) is circumstantial; we had 8 processors available and ran 6 chains on each.

To train the T topic-label distributions ϕ'_t for Dependency-LDA, we ran 10 MCMC chains, taking a single sample from each after a burn-in of 500 iterations. One can not average the estimates of ϕ'_t over multiple chains as we did when estimating ϕ, because the topics are being learned in an unsupervised manner and do not have a fixed interpretation between chains. Thus, each chain provides a unique set of T topic distributions. These 10 estimates are then stored for test time (at which point we can eventually average over them).

At test time, we took 900 total samples of the estimated parameters for each test document (θ_d for all models, plus θ'_d for Dependency-LDA).[25] For each model, we ran 60 independent MCMC chains, and took 15 samples from each chain using an initial burn-in of 50 iterations and a 5 iteration lag between samples (to reduce autocorrelation). For Dependency-LDA, in order to incorporate the ten 10 separate estimates of ϕ'_t, we distributed the 60 MCMC chains across the different sets of topics; specifically, 6 chains were run using each of the 10 sets of topics (giving us 60 in total).

In order to average estimates across the chains, we used our 900 samples to compute the posterior estimates of θ_d and $\alpha'^{(d)}$ (where $\alpha'^{(d)}$ only changes across samples for Dependency-LDA; for Prior-LDA, this estimate is fixed, and it is not applicable to Flat-LDA). The final (averaged) estimate of the prior $\alpha'^{(d)}$ is added to the final estimate of θ_d to generate a single posterior predictive distribution for θ_d (due to the conjugacy of the Dirichlet and multinomial distributions). We note that at this step we used one last heuristic; when combining the estimates of the $\alpha^{(d)}$ and θ_d for each document, we set the total weight of the Dirichlet prior $\alpha'^{(d)}$ equal to the total number of words in the document (i.e., we set $\sum_c \alpha'^{(d)} = \sum_c \theta_d$). We chose to do this because, whereas the total weight of $\alpha'^{(d)}$ used during sampling was fixed across all documents, the documents themselves had different numbers of words. Therefore, for very long documents, the final predictions would otherwise be mostly influenced by the word-assignments, and for very short documents the prior would overwhelm word-assignments.[26] The final posterior estimate of θ_d computed from the 900 samples was used to generate all predictions.

Appendix C: Derivation of sampling equation for label-token variables (C)

In this appendix, we provide a derivation of (9), for sampling a document's label-tokens $\mathbf{c^{(d)}}$.

The variable $c_i^{(d)}$ can take on values $\{1, 2, \ldots, C\}$. We need to compute the probability of $c_i^{(d)} = c$ (for $c \leq C$) conditioned on the label assignments $z^{(d)}$, the topic assignments $z'^{(d)}$, and the remaining variables $c_{-i}^{(d)}$.

$$\begin{aligned} p(c_i^{(d)} = c \mid z^{(d)}, z'^{(d)}, c_{-i}^{(d)}) &= \frac{p(z^{(d)}, c^{(d)} \mid z'^{(d)})}{p(z^{(d)} \mid z'^{(d)})} \\ &\propto p(z^{(d)}, c^{(d)} \mid z'^{(d)}) \\ &= p(z^{(d)} | c^{(d)}) \cdot p(c^{(d)} | z'^{(d)}_i) \\ &\propto p(z^{(d)} | c^{(d)}) \cdot p(c_i^{(d)} | z'^{(d)}_i, c_{-i}^{(d)}) \end{aligned} \tag{14}$$

[25] As noted previously, we used the "fast inference" method, in which we do not actually sample the c parameters.

[26] Early experimentation with a smaller version of the NYT dataset indicated that this method leads to modest improvements in performance.

Thus, the conditional probability of $c_i^{(d)} = c$ is a product of two factors. The first factor in (14) is the likelihood of the label assignments $z^{(d)}$ given the labels $c^{(d)}$. It can be computed by marginalizing over the document's distribution over labels $\theta^{(d)}$:

$$\begin{aligned}
p(z^{(d)}|c^{(d)}) &= \int_{\theta^{(d)}} p(z^{(d)}|\theta^{(d)}) \cdot p(\theta^{(d)}|c^{(d)})\, d\theta^{(d)} \\
&= \int_{\theta^{(d)}} \left(\prod_{i=1}^{N} \theta_{z_i^{(d)}}^{(d)} \right) \left(\frac{1}{\mathcal{B}(\alpha'^{(d)})} \prod_{j=1}^{C} \left(\theta_j^{(d)}\right)^{\alpha'^{(d)}_j - 1} \right) d\theta^{(d)} \\
&= \frac{1}{\mathcal{B}(\alpha'^{(d)})} \int_{\theta^{(d)}} \left(\theta_\cdot^{(d)}\right)^{N_{\cdot,d}^{CD}} \prod_{j=1}^{C} \left(\theta_j^{(d)}\right)^{\alpha'^{(d)}_j - 1} d\theta^{(d)} \\
&= \frac{1}{\mathcal{B}(\alpha'^{(d)})} \int_{\theta^{(d)}} \prod_{j=1}^{C} \left(\theta_j^{(d)}\right)^{\alpha'^{(d)}_j + N_{j,d}^{CD} - 1} d\theta^{(d)} \\
&= \frac{\mathcal{B}(\alpha'^{(d)}_j + N_{\cdot,d}^{CD})}{\mathcal{B}(\alpha'^{(d)})}
\end{aligned} \tag{15}$$

Here $N_{j,d}^{CD}$ represents the number of words in document d assigned the label $j \in \{1, 2, \ldots, C\}$ and $\mathcal{B}(\alpha)$ represents the multinomial Beta function whose argument is a real vector α. The numerator on the last line is an abuse of notation that denotes the Beta function whose argument is the vector sum $([\alpha'^{(d)}_1 \ldots \alpha'^{(d)}_C] + [N_{1,d}^{CD} \ldots N_{C,d}^{CD}])$. The Beta function can be expressed in terms of the Gamma function:

$$\begin{aligned}
p(z^{(d)}|c^{(d)}) &= \frac{\mathcal{B}(\alpha'^{(d)} + N_{\cdot,d}^{CD})}{\mathcal{B}(\alpha'^{(d)})} \\
&= \frac{\prod_{j=1}^{C} \Gamma(\alpha'^{(d)}_j + N_{j,d}^{CD})}{\prod_{j=1}^{C} \Gamma(\alpha'^{(d)}_j)} * \frac{\Gamma(\sum_{j=1}^{C} \alpha'^{(d)}_j)}{\Gamma(\sum_{j=1}^{C} \alpha'^{(d)}_j + N_{j,d}^{CD})} \\
&\propto \frac{\prod_{j=1}^{C} \Gamma(\alpha'^{(d)}_j + N_{j,d}^{CD})}{\prod_{j=1}^{C} \Gamma(\alpha'^{(d)}_j)}
\end{aligned} \tag{16}$$

Here the Gamma function takes as argument a real-valued number. As the value of $c_i^{(d)}$ iterates over the range $\{1, 2, \ldots, C\}$, the prior vector $\alpha'^{(d)}$ changes but the summation of its entries $\sum_{j=1}^{C} \alpha'^{(d)}_j$ and the data counts $N_{j,d}^{CD}$ do not change.

The second term in (14), $p(c_i^{(d)}|z'^{(d)}_i, c_{-i}^{(d)})$, is the probability of the label $c_i^{(d)}$ given its topic assignment $z'^{(d)}_i$ and the remaining labels $c_{-i}^{(d)}$. This is analogous to the probability of a word given a topic in standard unsupervised LDA (where the $c_i^{(d)}$ variable is analogous to a "word", and the $z'^{(d)}_i$ variable is analogous to the "topic-assignment" for the word). This probability—denoted as $\phi'^{(t)}_c$—is estimated during training time. Thus, the final form of (14) is given by:

$$p(c_i^{(d)} = c \mid z^{(d)}, z'^{(d)}, c_{-i}^{(d)}) \propto \frac{\prod_{j=1}^{C} \Gamma(\alpha'^{(d)}_j + N_{j,d}^{CD})}{\prod_{j=1}^{C} \Gamma(\alpha'^{(d)}_j)} \cdot \phi'^{(t)}_c \tag{17}$$

Appendix D: Comparisons with published results

The one-vs-all SVM approach we employed for comparison with our LDA-based methods is a highly popular benchmark in the multi-label classification literature. However, there are a large number of alternative methods (both probabilistic and discriminative) which have been proposed, and this is an active area of research. In order to put our results in the larger context of the current state of multi-label classification, we compare below our results with published results for alternative classification methods. Because of the variability of published results—due to the lack of consensus in the literature in terms of the prediction-tasks, evaluation metrics, and versions of datasets that have been used for model evaluation—there are relatively few results that we can compare to. Nonetheless, for all but one of our datasets (the NYT dataset, which we constructed ourselves), we were able to find published values for at least some of the evaluation metrics we utilized in this paper.

In this Appendix we present a comparison of our own scores (for the two SVM and three LDA-based approaches) with published scores on equivalent training-test splits of equivalent datasets. The goals of this Appendix are (1) to put our own results in the context of the larger state of the area of multi-label classification, (2) to demonstrate that our Tuned-SVM approach is competitive with other similar Tuned-SVM benchmarks that have been used elsewhere, and (3) to demonstrate that on power-law datasets, our Dependency-LDA model achieves scores that are competitive with state-of-the art discriminative approaches.

Comparison with published scores on the EUR-Lex dataset

To the best of our knowledge, only one research group has published results using the EUR-Lex dataset Loza Mencía and Fürnkranz (2008a, 2008b). Figure 14 compares our results with all results presented in Loza Mencía and Fürnkranz (2008a)[27] for the EUR-Lex Eurovoc descriptors. The best two algorithms from Loza Mencía and Fürnkranz (2008a)—MMP (*Multilabel Multiclass Perceptron*) and DMLPP (*Dual Multilabel Pairwise Perceptrons*)—are discriminative, perceptron-based algorithms. Both algorithms account for label-dependencies, and are designed specifically for the task of document-pivoted label-ranking (thus, no results are presented for label-pivoted predictions). Training of these algorithms was performed to optimize rankings with respect to the Is-Error loss function.

Dependency-LDA outperforms all algorithms (on all five measures) from Loza Mencía and Fürnkranz (2008a) except MMP (at 5 epochs) and DMLPP (at 2 epochs).[28] Dependency-LDA outperforms MMP(5) on all metrics but Is-Error (which was the metric the algorithm was tuned to optimize). Dependency-LDA beats DMLPP at 1 epoch on all metrics, but at 2 epochs (which gave their best overall set of results) performance between the two algorithms is quite close overall; Dependency-LDA outperforms DMLPP(2) on 2/5 measures, and performs worse on 3/5 measures (although, the relative improvement of DMLPP over Dependency-LDA on Average-Precision is fairly small relative to differences on other scores). In terms of overall performance, it is not clear that either Dependency-LDA or

[27] Note that we did not use an equivalent feature selection method as in their paper; due to memory constraints of their algorithms, Loza Mencía and Fürnkranz (2008a) reduced the number of features to 5,000 for each split of the dataset, where as our feature selection method (where we removed words occurring fewer than 20 times in the training set) left us with approximately 20,000 features for each split.

[28] In the perceptron-based algorithms from Loza Mencía and Fürnkranz (2008a), the number of Epochs corresponds to the number passes over the training corpus during which the model weights are tuned. See reference for further details.

Comparisons With Published Results (EUR-Lex): Document-Pivoted Ranking Predictions

Publication	Model		ROC Analyses ⇑	MultiLabel Metrics ⇓			
	Model	**Epoch**	**Avg-Prec**	**Rnk-Loss**	**One-Err**	**Is-Err**	**Margin**
Current Paper	$SVM_{Vanilla}$	--	45.4	2.51	37.5	98.1	387
	SVM_{Tuned}	--	43.0	3.28	* 31.6	98.2	436
	$LDA_{Dependency}$	--	* 51.1	* **1.77**	32.0	* 97.2	* **269**
	LDA_{Prior}	--	40.2	5.15	34.7	98.6	708
	LDA_{Flat}	--	39.6	5.78	35.6	98.8	841
	Model	**Epoch**	**Avg-Prec**	**Rnk-Loss**	**One-Err**	**Is-Err**	**Margin**
Mencia & Furnkranz, (2008)	MLNB	--	1.1	22.9	100.0	99.6	1,644
	BR	1	26.9	40.4	48.7	98.6	3,231
	BR	2	31.6	35.5	41.5	98.2	3,050
	BR	5	35.9	31.0	37.3	97.2	2,843
	MMP	1	29.3	3.91	75.9	98.8	598
	MMP	2	39.5	4.35	54.4	97.5	694
	MMP	5	47.3	4.70	40.2	* **96.0**	761
	DMLPP	1	46.7	2.78	35.5	97.9	434
	DMLPP	2	* **52.3**	* 2.50	* **29.5**	96.6	* 397

Fig. 14 Comparison of results from the current paper with result from Loza Mencía and Fürnkranz (2008a), on document-pivoted ranking evaluations

DMLPP(2) is a clear winner. However, it seems fairly clear that the Dependency-LDA outperforms MMP overall, and at the very least is reasonably competitive with DMLPP(2). This is particularly surprising given that both the MMP and DMLPP algorithms are designed specifically for the task of label-ranking, and were optimized specifically for one of the measures considered (whereas Dependency-LDA was not optimized with respect to any specific measure, or even with the specific task of label-ranking in mind).

Comparison with published scores on Yahoo! datasets

To the best of our knowledge, the only paper which has been published using an equivalent version of the Yahoo! *Arts* and *Health* datasets is Ji et al. (2008). Note that numerous additional papers have been published using this dataset, but most of these have used different sets of train-test splits, or used a different number of labels (e.g., Ueda and Saito 2002; Fan and Lin 2007).[29] In Fig. 15 we compare our results on the Yahoo! subdirectory datasets with the numerous discriminative methods presented in Ji et al. (2008). For complete details on all the algorithms from Ji et al. (2008), we refer the reader to their paper. However, we note that our $SVM_{VANILLA}$ and SVM_{TUNED} methods are essentially equivalent to their SVM_C and SVM methods, respectively. Additionally, the Multi-Label Least Squares (ML_{LS}) method introduced in their paper, uses a discriminative approach for accounting for label-dependencies.

First, we note that the results from our own SVM scores are quite similar to the SVM scores from Ji et al. (2008), which serves to demonstrate that the discriminative classification method we have used throughout the paper for comparison with LDA methods is competitive

[29]The version we used had some of the infrequent labels removed from the dataset, and had exactly 1,000 training documents in each of the five train-test splits.

Comparisons With Published Results (Yahoo!) : Label-Pivoted Binary Predictions

Publication	Model	Yahoo! *Arts* F1 $_{MACRO}$ ⇑	Yahoo! *Arts* F1 $_{MICRO}$ ⇑	Yahoo! *Health* F1 $_{MACRO}$ ⇑	Yahoo! *Health* F1 $_{MICRO}$ ⇑
Current Paper (N-PROPORTIONAL)	$SVM_{Vanilla}$	.325	.428	.548	.638
	SVM_{Tuned}	.355	* .454	* .571	* .656
	$LDA_{Dependency}$	* **.367**	.451	.562	.646
	LDA_{Prior}	.358	.440	.521	.610
	LDA_{Flat}	.355	.435	.512	.599
Ji et al., (2008)	ML_{LS}	* .358	* **.472**	* **.597**	* **.681**
	CCA + Ridge	.319	.444	.543	.677
	CCA + SVM	.316	.452	.534	.680
	ASO_{SVM}	.357	.445	.581	.675
	SVM_{C}	.322	.445	.563	.671
	SVM	.338	.457	.571	.677

Fig. 15 Comparison of Macro-F1 and Micro-F1 scores for the models utilized in the current paper with previously published results from Ji et al. (2008)

with similar methods that have been presented in the literature. The ML_{LS} method that they introduced in the paper outperforms all SVMs, as well as the additional methods that they considered, on all scores.

Performance of the LDA-based methods was generally worse than the best discriminative method (ML_{LS}) presented in Ji et al. (2008). However, on the Yahoo! *Arts* dataset, Dependency-LDA outperformed all methods on the Macro-F1 scores (which, as a reminder, emphasizes the performance on the less frequent labels), and Prior-LDA performed as good as the best discriminative method. On the Micro-F1 scores for Yahoo! *Arts*, Dependency-LDA performance was slightly worse than the CCA + SVM and tuned SVM methods, and was clearly worse than the ML_{LS} method, but did outperform the other three discriminative methods. On the Yahoo! *Health* dataset—which has fewer labels, and more training data per label than the *Arts* dataset—Dependency-LDA fared worse relative to the discriminative methods. Dependency-LDA scored better than or similarly to just three of the six methods for Macro-F1 scores, and was beaten by all methods for the Micro-F1 scores.

We note that, although overall performance the LDA-based methods is generally worse than it is for the best discriminative methods on the two Yahoo! datasets, this provides additional evidence that even on non power-law datasets, the LDA-based approaches show a particular strength in terms of performance on infrequent labels (as evidenced by the relatively good Macro-F1 scores for Dependency-LDA). Furthermore, on these types of datasets, depending on the evaluation metrics being considered, and the exact statistics of the dataset, the Dependency-LDA method is in some cases competitive with or even better than SVMs and more advanced discriminative methods.

Comparison with published scores on RCV1-v2 datasets

The RCV1-v2 dataset is a common multi-label benchmark, and numerous results on this dataset can be found in the literature. We chose to compare with results from both Lewis et al. (2004) and Eyheramendy et al. (2003) since this provides us with a very wide range of algorithms for comparison (where the former paper considers several of the most popular discriminative classification methods, and the latter paper considers numerous Bayesian

Comparisons With Published Results (RCV1-v2): Label-Pivoted Binary Predictions

Publication	Model	F1 $_{MACRO}$ ⇑	F1 $_{MICRO}$ ⇑
Current Paper (N-PROPORTIONAL)	SVM $_{Vanilla}$	.571	.780
	SVM $_{Tuned}$	* **.579**	* .787
	LDA $_{Dependency}$	.539	.762
	LDA $_{Prior}$	.484	.629
	LDA $_{Flat}$	.482	.617
Eyheramendy et al. (2003)	Probit $_{Jeffreys\ (300)}$	.394	.725
	Probit $_{Laplace\ (300)}$	.477	.744
	Probit $_{Gaussian\ (300)}$	.453	.749
	Logistic $_{Laplace\ (300)}$	.480	.755
	Logistic $_{Laplace\ (3,000)}$	* .530	.789
	Logistic $_{Gaussian\ (3,000)}$	.518	* .797
Lewis et al. (2004)	SVM.1*	* **.579**	* **.816**
	SVM.2	.577	.810
	k-NN	.499	.767
	Rocchio	.509	.695

Fig. 16 Comparison of Macro-F1 and Micro-F1 scores for the models utilized in the current paper with previously published results

style regression methods). Note that the Macro-F1 and Micro-F1 scores for the *SVM-1* algorithm presented in Lewis et al. (2004) were the result of two distinct sets of predictions (where one set of SVM predictions was thresholded to optimize Micro-F1, and a separate set of predictions were optimized for Macro-F1). Since all other methods presented in Fig. 16 (as well as throughout our paper) used a single set of predictions to compute all scores, we re-computed the Macro-F1 scores using the predictions optimized for Micro-F1,[30] in order to be consistent across all results. The *SVM-1* algorithm nonetheless is tied for the best Macro-F1 score (with our own SVM results) and achieves the best Micro-F1 score overall.

In terms of the LDA-based methods, the Dependency LDA model clearly performs worse than SVMs on RCV1-v2. However, it outperforms all non-SVM methods on Macro-F1 (including methods from both Lewis et al. 2004 and Eyheramendy et al. 2003). It additionally achieves a Micro-F1 score that is competitive with most of the non-SVM methods (although it is significantly worse than most logistic-regression methods, in addition to SVMs).

References

The EUR-Lex repository, June 2010. URL http://www.ke.tu-darmstadt.de/resources/eurlex/eurlex.html.

Allwein, E. L., Schapire, R. E., & Singer, Y. (2001). Reducing multiclass to binary: a unifying approach for margin classifiers. *Journal of Machine Learning Research*, *1*, 113–141.

[30] These were re-computed from the confusion matrices made available in the online appendix to their paper.

Blei, D., & McAuliffe, J. (2008). Supervised topic models. In J. C. Platt, D. Koller, Y. Singer, & S. Roweis (Eds.), *Advances in neural information processing systems 20* (pp. 121–128). Cambridge: MIT Press.

Blei, D. M., & Lafferty, J. D. (2005). Correlated topic models. In *Advances in neural information processing systems*.

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

Blei, D. M., Griffiths, T. L., & Jordan, M. I. (2010). The nested Chinese restaurant process and Bayesian nonparametric inference of topic hierarchies. *Journal of the ACM*, *57*, 7:1–7:30.

Cao, L., & Fei-fei, L. (2007). Spatially coherent latent topic model for concurrent object segmentation and classification. In *Proceedings of IEEE International Conference in Computer Vision (ICCV)*.

Crammer, K., & Singer, Y. (2003). A family of additive online algorithms for category ranking. *Journal of Machine Learning Research*, *3*, 1025–1058.

Davis, J., & Goadrich, M. (2006). The relationship between precision-recall and roc curves. In *ICML'06: proceedings of the 23rd international conference on machine learning* (pp. 233–240). New York: ACM.

de Carvalho, A. C. P. L. F., & Freitas, A. A. (2009). A tutorial on multi-label classification techniques. In *foundations of computational intelligence*: *Vol. 5. Studies in computational intelligence 205* (pp. 177–195). Berlin: Springer.

Dekel, O., & Shamir, O. (2010). Multiclass-multilabel classification with more classes than examples. *Journal of Machine Learning Research—Proceedings Track*, *9*, 137–144.

Druck, G., Pal, C., McCallum, A., & Zhu, X. (2007). Semi-supervised classification with hybrid generative/discriminative methods. In *KDD'07: proceedings of the 13th ACM SIGKDD international conference on knowledge discovery and data mining* (pp. 280–289). New York: ACM.

Eyheramendy, S., Genkin, A., Ju, W.-H., Lewis, D. D., & Madigan, D. (2003). *Sparse Bayesian classifiers for text categorization* (Technical report). Journal of Intelligence Community Research and Development.

Fan, R.-E., & Lin, C.-J. (2007). *A study on threshold selection for multi-label classification* (Technical report). National Taiwan University.

Fan, R.-E., Chang, K.-W., Hsieh, C.-J., Wang, X.-R., & Lin, C.-J. (2008). LIBLINEAR: A library for large linear classification. *Journal of Machine Learning Research*, *9*, 1871–1874.

Forman, G. (2003). An extensive empirical study of feature selection metrics for text classification. *Journal of Machine Learning Research*, *3*, 1289–1305.

Fürnkranz, J., Hüllermeier, E., Mencía, E. L., & Brinker, K. (2008). Multilabel classification via calibrated label ranking. *Machine Learning*, *73*(2), 133–153.

Ghamrawi, N., & McCallum, A. (2005). Collective multi-label classification. In *CIKM'05: proceedings of the 14th ACM international conference on information and knowledge management* (pp. 195–200). New York: ACM.

Griffiths, T. L., & Steyvers, M. (2004). Finding scientific topics. *Proceedings of the National Academy of Sciences of the United States of America*, *101*(Suppl 1), 5228–5235.

Har-Peled, S., Roth, D., & Zimak, D. (2002). *Constraint classification: A new approach to multiclass classification and ranking* (Technical report). Champaign, IL, USA.

Hersh, W., Buckley, C., Leone, T. J., & Hickam, D. (1994). OHSUMED: an interactive retrieval evaluation and new large test collection for research. In *SIGIR'94: proceedings of the 17th annual international ACM SIGIR conference on research and development in information retrieval* (pp. 192–201). New York: Springer.

Ioannou, M., Sakkas, G., Tsoumakas, G., & Vlahavas, I. (2010). Obtaining bipartitions from score vectors for multi-label classification. In *Proceedings of the 2010 22nd IEEE international conference on tools with artificial intelligence—Volume 01, ICTAI'10* (pp. 409–416). Washington: IEEE Comput. Soc. ISBN 978-0-7695-4263-8. doi:http://dx.doi.org/10.1109/ICTAI.2010.65. URL http://dx.doi.org/10.1109/ICTAI.2010.65.

Japkowicz, N., & Stephen, S. (2002). The class imbalance problem: A systematic study. *Intelligent Data Analysis*, *6*(5), 429–449.

Ji, S., Tang, L., Yu, S., & Ye, J. (2008). Extracting shared subspace for multi-label classification. In *KDD'08: proceedings of the 14th ACM SIGKDD international conference on knowledge discovery and data mining* (pp. 381–389). New York: ACM.

Lacoste-Julien, S., Sha, F., & Jordan, M. I. (2008). DiscLDA: discriminative learning for dimensionality reduction and classification. In *NIPS* (pp. 897–904).

Lewis, D. D., Yang, Y., Rose, T. G., & Li, F. (2004). RCV1: A new benchmark collection for text categorization research. *Journal of Machine Learning Research*, *5*, 361–397.

Liu, T.-Y., Yang, Y., Wan, H., Zeng, H.-J., Chen, Z., & Ma, W.-Y. (2005). Support vector machines classification with a very large-scale taxonomy. *SIGKDD Explorations Newsletter*, *7*(1), 36–43.

Loza Mencía, E., & Fürnkranz, J. (2008a). Efficient pairwise multilabel classification for large-scale problems in the legal domain. In *ECML PKDD'08: proceedings of the European conference on machine learning and knowledge discovery in databases—Part II* (pp. 50–65). Berlin: Springer.

Loza Mencía, E., & Fürnkranz, J. (2008b). Efficient multilabel classification algorithms for large-scale problems in the legal domain. In *Proceedings of the LREC 2008 workshop on semantic processing of legal texts*.

McCallum, A. K. (1999). Multi-label text classification with a mixture model trained by EM. In *AAAI 99 workshop on text learning*.

Mimno, D., & McCallum, A. (2008). Topic models conditioned on arbitrary features with Dirichlet-multinomial regression. In *Proceedings of the 24th conference on uncertainty in artificial intelligence (UAI'08)*.

Mimno, D., Li, W., & McCallum, A. (2007). Mixtures of hierarchical topics with pachinko allocation. In *ICML'07: proceedings of the 24th international conference on machine learning* (pp. 633–640). New York: ACM.

Rak, R., Kurgan, L., & Reformat, M. (2005). Multi-label associative classification of medical documents from medline. In *ICMLA'05: proceedings of the fourth international conference on machine learning and applications*, Washington, DC, USA (pp. 177–186).

Ramage, D., Hall, D., Nallapati, R., & Manning, C. D. (2009). Labeled LDA: A supervised topic model for credit attribution in multi-labeled corpora. In *Proceedings of the 2009 conference on empirical methods in natural language processing*, Singapore, August 2009 (pp. 248–256). Association for Computational Linguistics.

Read, J., Pfahringer, B., Holmes, G., & Frank, E. (2009). Classifier chains for multi-label classification. In *ECML/PKDD (2)* (pp. 254–269).

Rifkin, R. & Klautau, A. (2004). In defense of one-vs-all classification. *Journal of Machine Learning Research*, *5*, 1532–4435.

Rosen-Zvi, M., Griffiths, T., Steyvers, M., & Smyth, P. (2004). The author-topic model for authors and documents. In *AUAI'04: proceedings of the 20th conference on uncertainty in artificial intelligence* (pp. 487–494). Arlington: AUAI Press.

Sandhaus, E. (2008). *The New York Times Annotated Corpus*. Linguistic Data Consortium. Philadelphia.

Schneider, K.-M. (2004). On word frequency information and negative evidence in naive Bayes text classification. In *España for natural language processing, EsTAL*.

Sebastiani, F. (2002). Machine learning in automated text categorization. *ACM Computing Surveys*, *34*(1), 1–47.

Teh, Y. W., Jordan, M. I., Beal, M. J., & Blei, D. M. (2004). Hierarchical Dirichlet processes. *Journal of the American Statistical Association*, *101*, 1566–1581.

Tsoumakas, G., & Katakis, I. (2007). Multi label classification: An overview. *International Journal of Data Warehouse and Mining*, *3*(3), 1–13.

Tsoumakas, G., Katakis, I., & Vlahavas, I. (2009). *Data mining and knowledge discovery handbook*. *Mining multi-label data*. Berlin: Springer.

Ueda, N., & Saito, K. (2002). Parametric mixture models for multi-labeled text. In *NIPS* (pp. 721–728).

Wang, Y., Sabzmeydani, P., & Mori, G. (2007). Semi-latent Dirichlet allocation: a hierarchical model for human action recognition. In *Proceedings of the 2nd conference on human motion: understanding, modeling, capture and animation* (pp. 240–254). Berlin: Springer.

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

Yang, Y. (2001). A study of thresholding strategies for text categorization. In *SIGIR'01: proceedings of the 24th annual international ACM SIGIR conference on research and development in information retrieval* (pp. 137–145). New York: ACM.

Yang, Y., Zhang, J., & Kisiel, B. (2003). A scalability analysis of classifiers in text categorization. In *SIGIR'03: proceedings of the 26th annual international ACM SIGIR conference on research and development in information retrieval* (pp. 96–103). New York: ACM.

Zhang, M.-L., & Zhang, K. (2010). Multi-label learning by exploiting label dependency. In *KDD'10: proceedings of the 16th ACM SIGKDD international conference on knowledge discovery and data mining* (pp. 999–1008). New York: ACM.

Zhang, M.-L., Peña, J. M., & Robles, V. (2009). Feature selection for multi-label naive Bayes classification. *Information Science*, *179*(19), 3218–3229.

Zhu, J., Ahmed, A., & Xing, E. P. (2009). MedLDA: maximum margin supervised topic models for regression and classification. In *Proceedings of the 26th annual international conference on machine learning, ICML'09* (pp. 1257–1264). New York: ACM.