Modeling Tag Dependencies in Tagged Documents

Timothy N. Rubin
Cognitive Sciences
U.C. Irvine
trubin@uci.edu

America Holloway
Computer Science
U.C. Irvine
ahollowa@uci.edu

Padhraic Smyth
Computer Science
U.C. Irvine
smyth@ics.uci.edu

Mark Steyvers
Cognitive Sciences
U.C. Irvine
mark.steyvers@uci.edu

Abstract

We present a general approach for modeling tagged documents with topic models. This approach extends related topic models by exploiting the dependencies between tags. We show how this model improves performance in a prediction task where the goal is to predict missing tags for new documents. Predictions also compare favorably with SVMs.

1 Introduction

There now exist many document collections where each document has been manually assigned one or more tags. For example, Wikipedia articles are typically assigned several category tags by its users. The recently released New York Times (NYT) Annotated corpus [1] contains over one million articles where tags have been manually assigned by a group of editors. These descriptor tags often provide a useful summary for the content of a document. The broad goal of this research is to develop models for these datasets in order to automatically tag new documents with sets of these human-provided tags.

The traditional approach to prediction with tagged documents has been to use multi-class discriminative models such as binary SVMs [2]. This approach often performs well on prediction tasks, but accuracy tends to dramatically suffer as the number of available tags increases. This discriminative approach also has limitations in terms of interpretability because it is often not clear what parts of the document relate to each of the predicted tags. More importantly, because the decision to assign a document each tag is performed independently across tags, it can be difficult for these multi-class discriminative models to exploit the dependencies between tags [3]. These limitations motivate the investigation of probabilistic generative models of document tags.

A number of probabilistic models for tagged documents have been proposed. McCallum [4] as well as Ueda and Saito [5] presented mixture models in which each document is composed of a number of word distributions that correspond to the tags assigned to the document. Recently, Ramage et al. [6] proposed a similar model based on Latent Dirichlet Allocation (LDA) [7], where each of the words in a document is assigned to a "topic" corresponding to one of the tags associated with a document. One issue that arises within this framework is that when tags are unobserved it can be difficult to find the appropriate assignment for words, particularly as the number of available tags increases. For example, consider assigning the word *steroids* to one of the several thousands of tags available within the NYT Annotated Corpus. This word has a high probability under many of the tags, such as MEDICINE AND HEALTH, BLACK MARKETS, and BASEBALL. This ambiguity can often be resolved if we account for the other tags present within the document; e.g., the word *steroids* is likely to be related to the tag BASEBALL given that the tag SUSPENSIONS, DISMISSALS AND RESIGNATIONS is also assigned to the document, whereas it may be more likely to be related to MEDICINE AND HEALTH given the presence of the tag CANCER. In other words, the dependencies between tags are helpful in resolving ambiguity in word assignments and are potentially useful when inferring tags for new documents.

We present a topic modeling approach for tagged documents which extends current models by explicitly modeling the dependencies between tags. We will first show how topic models can be used

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

Table 1: The eight most likely words for five tags, along with the word probabilities. The number to the right of the tags indicates the number of occurrences of the tag in training documents.

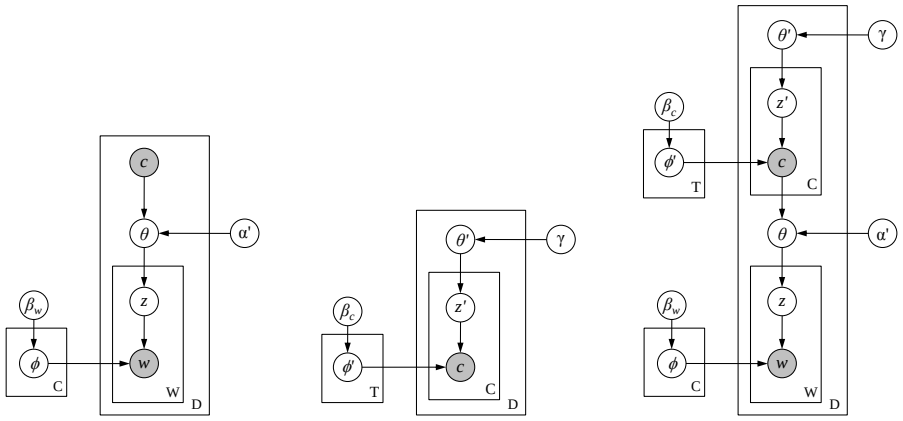

Figure 1: Graphical Models for three topic modeling approaches: Topic model using words only (left), A topic model using tags only (middle), and a model using both tags and words (right).

to learn appropriate word distributions for each tag within a corpus as in [6]. We then extend this model to account for the dependencies between these tags, and show that this improves tag predictions for new documents. We evaluate all models using a subset of 4,000 tagged documents taken from the the New York Times (NYT) Annotated Corpus [1]. This subset included 1585 unique tags. All documents had between two and 16 tags assigned to them, with an average of 4.5 tags per document.

2 Topic models for tagged documents

2.1 Learning word distributions for tags

In this section, we will first show how topic models can be used to learn a distribution of words for each tag within a corpus as shown in [6]. In the following section, we extend this model to account for the dependencies between these tags. We assume that each document $d \in \{1, \ldots, D\}$ is represented by a multinomial distribution $\theta^{(d)}$ over the set of observed tags, and that each tag $j \in \{1, \ldots, C\}$ is associated with a multinomial distribution ϕ_j over word types. In the standard finite-mixture version of LDA, the number of topics is set by the user. In contrast, here the total number of mixture components C equals the number of unique tags that appear in the document collection. The generative process for each document is:

1. For each tag c, sample a multinomial distribution over words ϕ_c with a Dirichlet(β_w) prior
2. For each document d, sample a multinomial distribution over tags $\theta^{(d)}$ from a Dirichlet($\boldsymbol{\alpha}^{(d)}$) prior where the hyperparameters $\boldsymbol{\alpha}^{(d)}$ are non-zero only for the set of tags $c^{(d)}$ assigned to the document.
3. To generate each word token for document d
   (a) Sample a tag z from $\theta^{(d)}$
   (b) Sample a word w from ϕ_z

This model is depicted using graphical model notation in Figure 1(a). We applied this model to the NYT dataset. We used collapsed Gibbs sampling [8] to infer the assignments of word-tokens

"Consumer Safety"	**.017**	**"Warfare And Disputes"**	**.024**	**"Cheating and Athletics"**	**.016**
CANCER	.078	ARMAMENT, DEFENSE AND MILITARY...	.162	OLYMPIC GAMES (1988)	.052
HAZARDOUS AND TOXIC SUBSTANCES	.039	INTERNATIONAL RELATIONS	.133	SUSPENSIONS, DISMISSALS AND RESIG...	.038
PESTICIDES AND PESTS	.021	UNITED STATES INTERNATIONAL RELA...	.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

Table 2: Three example topics learned by the LDA model applied to tags. Topic labels (in quotes) are subjective interpretations provided by the authors.

to tags. From these assignments, we inferred the tag-document distributions $\theta^{(d)}$ and the word-tag distributions ϕ_c. Table 1 presents the ten most likely words for several of the NYT tags.

2.2 Incorporating dependencies between tags

In this section, we develop a topic modeling approach that learns the statistical relationships between words, tags, and a latent set of topics that is used to capture the dependencies between tags. We first illustrate that applying LDA directly to the document tags themselves captures meaningful relationships between the tags. Each document is assumed to be composed of a set of topics θ'_d , where each topic is a distribution ϕ' over tags[1]. The generative model for document tags is shown in Figure 1(b). Table 2 shows three example topics that were learned from a total of 50 topics, and their corresponding distributions over tags.

To generate both the set of tags for a document and the document text itself, we propose a two-stage generative process. The first process describes how the set of tags within the document are selected from a latent set of topics. The second process describes how the document text is generated, conditional on these tags. The full generative process for documents is as follows:

1. For each topic j, sample a multinomial distribution over tags ϕ'_j from a Dirichlet(β_C) prior
2. For each tag c, sample a multinomial distribution over words ϕ_c from a Dirichlet(β_W) prior
3. To generate tags for the dth document:
   (a) Sample a multinomial mixture of topics $\theta'^{(d)}$ from a Dirichlet(γ) prior
   (b) To generate each tag:
      i. Sample a topic z' from $\theta'^{(d)}$
      ii. Sample a tag c from $\phi'_{z'}$
4. Generate words for the dth document using the generative process described in section 2.1

The graphical model for this generative process is shown in Figure 1(c). This model is easily trained using collapsed Gibbs sampling. Because the tags are observed during training, this decouples the tag-document distributions θ^d and word-tag distributions ϕ_c from the topic-document distributions θ'^d and tag-topic distributions ϕ'_j. Therefore, inference for the full model during training is equivalent to training the two models previously described (Figure 1(a) and Figure 1(b)). To be precise, using the standard collapsed Gibbs sampling equations we make assignments of words to tags, and *independently* make assignments of tags to topics.

Inference becomes more complex when making predictions for new documents. We do not know at this time how to perform correct inference on the full model at test time, and instead approximate it using the following approach: each word-tag assignment z within a document d is treated as a fractional observation of that tag for the document, and is given a corresponding tag-topic assignment z'. Intuitively, we can think of each of the the z assignments as providing partial evidence for a particular tag, and the z' assignments as a way to use this evidence to learn from the dependencies of the tag). The z' assignments are weighted such that the total count for a document is approximately the number of expected tags for the document. To condition the z assignments on the z' assignments, parameterize the Dirichlet prior $\alpha^{(d)}$, using $\alpha^{(d)} = \theta'^{(d)} \cdot \phi'_j \cdot \eta$, where η is a constant which determines how strongly the Dirichlet prior is weighted. We set $\eta = 50$, although initial

[1]Technically, this is not a proper generative model for tags, because LDA assumes that words are sampled with replacement, whereas tags can only be applied once to a document. However, this model provides a good approximation to a true generative model

		LDA Models			Non LDA	
		Words Only	Words + Tag Baselines	Words + Tag Dependencies	SVM	Tag Baselines
Rankings	Mean	3	3	2	3	54
	Median	69.89	49.45	27.28	37.58	158.03
Binary Predictions	Macro-F1	.428	.411	.411	.263	0.0022
	Micro-F1	.536	.547	.570	.451	0.099
Probabilities	Perplexity	53.46	27.11	18.88	–	338.35
	Mean	0.03	0.08	0.10	–	0.00
	Median	0.09	0.13	0.15	–	0.01

Table 3: Comparison of prediction measured for different modeling approaches. The *Words only* and *Words + Tag Dependencies* LDA Models were described in sections 2.2 and 2.1. *Words + Tag Baselines* uses baseline tag frequencies to determine a prior on θ.

experimentation suggested that the exact value we chose did not significantly impact performance. Although this method is an approximation, it allows information to be passed between the word-tag and tag-document levels of the model, and experimental results indicate that it works reasonably well.

2.3 Experimental Results

To evaluate model performance, we removed 424 test documents from the set of 4,000 NYT documents. The task was to predict which tags should be assigned to the test documents. We considered several measures of the accuracy of predictions. Measures of predicted tag rankings were computed after removing all other true tags, leaving behind only one test item and all remaining non true tags. Probability measures were computed after normalizing the posterior predicted probabilities of tags to sum to one. Binary predictions were made by thresholding the posterior predicted probabilities. SVM predictions were made using the LIBLINEAR package [9]. SVM tuning methods were calibrated on two categories from the Yahoo! dataset and achieved performance that is competitive with published results.

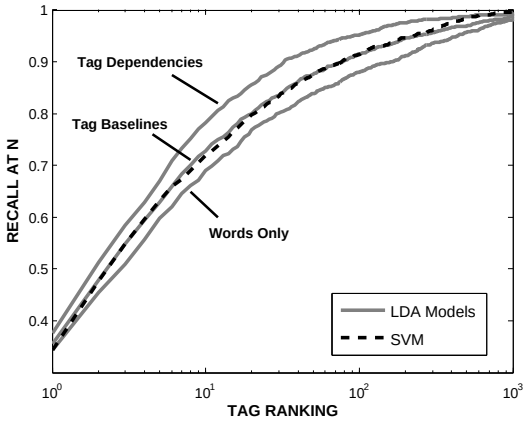

Figure 2: Comparison of recall values for the ranked tag predictions for the three LDA models and the binary SVMs (ranks shown in log scale)

Results for all measures and models are are presented in Table 3. On the binary prediction measures, the LDA models all perform significantly better than the baseline and SVM predictions. On the ranking measures, the SVM model outperforms all models except the model which includes tag dependencies, although from Figure 2 you can see that the SVM rankings perform nearly equivalent to the LDA model which accounts for tag baseline frequencies but not dependencies. A comparison of the LDA model for words only and the model with tag dependencies indicates that explicitly accounting for tag dependencies can improve the performance topic-modeling approaches to tagging new documents. Furthermore, the results demonstrate that this performance improvement is not due merely to capturing the baseline frequencies of the tags.

2.4 Related Work on Topic Dependencies

Ghamrawi and McCallum [10] described a conditional random field (CRF) model which captured pairwise dependencies between tags. However, it is unknown whether this model could be extended to effectively capture higher order dependencies, nor whether it would possess some of the advantages inherent to topic modeling approaches. Several extensions to LDA have been proposed to capture dependencies between topics learned with unlabeled data, including the Correlated Topic Model [11], Pachinko Allocation Model (PAM) [12], and hierarchical LDA (hLDA) [13]. It likely that all of these models could be extended to modeling tagged documents. Our model more closely resembles PAM than CTM in that it uses a fully-connected set of super-topics to model the tag dependencies. There are several fundamental differences however between the model presented here and PAM, most notably that the sampling of sub-topics (i.e. tags) from the super-topics is a separate process from the sampling of words from tags. On a more practical level, this confers a significant

computational advantage over PAM, because we only need consider $C + T$ total paths for each word in each iteration of the Gibbs sampler, whereas in PAM one needs to consider $C * T$ total paths.

References

[1] Evan Sandhaus. *The New York Times Annotated Corpus*. Linguistic Data Consortium, Philadelphia, 2008.

[2] Ryan Rifkin and Aldebaro Klautau. In defense of one-vs-all classification. *Journal of Machine Learning Research*, 5:1532–4435, 2004.

[3] Tao Li, Chengliang Zhang, and Shenghuo Zhu. Empirical studies on multi-label classification. In *ICTAI '06: Proceedings of the 18th IEEE International Conference on Tools with Artificial Intelligence*, pages 86–92, Washington, DC, USA, 2006. IEEE Computer Society.

[4] Andrew Kachites Mccallum. Multi-label text classification with a mixture model trained by em. In *AAAI 99 Workshop on Text Learning*, 1999.

[5] Naonori Ueda and Kazumi Saito. Parametric mixture models for multi-labeled text. In *NIPS*, pages 721–728, 2002.

[6] Daniel Ramage, David Hall, Ramesh Nallapati, and Christopher D. Manning. 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*, pages 248–256, Singapore, August 2009. Association for Computational Linguistics.

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

[8] T. L. Griffiths and M. Steyvers. Finding scientific topics. *Proc Natl Acad Sci U S A*, 101 Suppl 1:5228–5235, April 2004.

[9] Rong-En Fan, Kai-Wei Chang, Cho-Jui Hsieh, Xiang-Rui Wang, and Chih-Jen Lin. Liblinear: A library for large linear classification. *Journal of Machine Learning Research*, 9:1871–1874, August 2008.

[10] Nadia Ghamrawi and Andrew McCallum. Collective multi-label classification. In *CIKM '05: Proceedings of the 14th ACM international conference on Information and knowledge management*, pages 195–200, New York, NY, USA, 2005. ACM.

[11] David M. Blei and John D. Lafferty. Correlated topic models. In *NIPS*, 2005.

[12] Wei Li and Andrew McCallum. Pachinko allocation: Dag-structured mixture models of topic correlations. In *ICML '06: Proceedings of the 23rd international conference on Machine learning*, pages 577–584, New York, NY, USA, 2006. ACM.

[13] David M. Blei, Thomas L. Griffiths, Michael I. Jordan, and Joshua B. Tenenbaum. Hierarchical topic models and the nested chinese restaurant process. In *Advances in Neural Information Processing Systems*, page 2003. MIT Press, 2004.