# Tree-based Label Dependency Topic Models

**Anonymous Author(s)**

## Abstract

Multi-labeled corpora, where each document is tagged with a set of labels, are ubiquitous. When the number of unique labels in the dataset is large, there are naturally some dependencies among the labels. In this paper, we propose TREELAD—a hierarchical topic model to capture these label dependencies using a tree-structured topic hierarchy. We apply TREELAD on a real-world dataset and show some promising empirical results.

## 1 Introduction

The past decade has seen probabilistic topic models being used to study the thematic structure of documents in a wide range of forms including news, blogs, web pages etc [1]. Standard unsupervised topic models such as latent Dirichlet allocation (LDA) [2] aim to discover a set of topics from input data which only consists of a set of documents. In many settings, documents are associated with additional information, which motivates work to simultaneously model the documents' text with their continuous response variables [3, sLDA] or their categorical labels [4, DiscLDA].

In this work, we focus on modeling *multi-labeled data*, in which each document is tagged with a set of labels. These data are ubiquitous. Web pages are tagged with multiple directories[1], books are labeled with different categories or political speeches are annotated with multiple issues[2]. Previous topic models on multi-labeled data focus on cases where the number of labels is relatively small and labels are assumed independent [5, 6, 7]. Unfortunately, in many real-world examples, the number of labels range from hundreds to thousands, which often makes independence assumptions too strong. Recent work captures the dependency among labels by projecting them onto some latent space [8, 9].

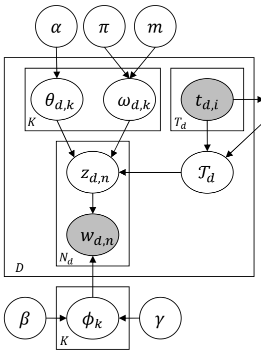
However, when the number of labels is very large, simply projecting the labels into a latent space may not be enough. Understanding an unwieldy label space requires discovering a structure within the labels. In this work, we attempt to capture the dependence between labels using a learned hierarchy. Our model, TREELAD, learns from label co-occurence and word usage to discover a hierarchy of topics associated with user-generated labels.

## 2 Model

We introduce TREELAD—a tree-based label dependency topic model to capture the hierarchy of topics from a set of multi-labeled documents. The inputs of TREELAD consist of a set of $D$ documents $\{\boldsymbol{w}_d\}$, each tagged with a set of labels $\boldsymbol{t}_d$. The number of unique labels is $K$ and the word vocabulary size is $V$. TREELAD associates each of the $K$ labels with a topic—a multinomial distribution over the vocabulary, and uses a tree-structure to capture the relationships among them. Each document is generated by repeatedly traversing the topic tree from the root downward to a node in the tree, whose associated topic is used to generated the corresponding word token. Figure 1 shows the plate diagram notations of TREELAD, together with its generative process.

[1] Open Directory Project (`http://www.dmoz.org/`)

[2] Policy Agenda Codebook (`http://policyagendas.org/`)



1. Create the label graph $\mathcal{G}$ and generate a tree $\mathcal{T}$ from $\mathcal{G}$ (See § 2.1)
2. For each node $k \in [1, K]$ in $\mathcal{T}$
   (a) If $k$ is the root, draw background topic $\phi_k \sim \text{Dir}(\beta)$
   (b) Otherwise, draw topic $\phi_k \sim \text{Dir}(\gamma \cdot \phi_{\sigma(k)})$
3. For each document $d \in [1, D]$ having labels $\boldsymbol{t}_d$
   (a) Define a subtree $\mathcal{T}_d \equiv \mathcal{R}(\mathcal{T}, \boldsymbol{t}_d)$ (See § 2.3)
   (b) For each node $k$ in $\mathcal{T}_d$
      i. Draw a multinomial over $k$'s children $\theta_{d,k} \sim \text{Dir}(\alpha)$
      ii. Draw a binary switching variable $\omega_{d,k} \sim \text{Beta}(m, \pi)$
   (c) For each word $n \in [1, N_d]$
      i. Draw $z_{d,n} \sim \mathcal{B}(\boldsymbol{\theta}_d, \boldsymbol{\omega}_d)$ (See § 2.2)
      ii. Draw $w_{d,n} \sim \text{Mult}(\phi_{z_{d,n}})$

Figure 1: The generative process with plate diagram notations of our model TREELAD.

### 2.1 Tree generation

We assume that we are given a directed complete graph between labels $i$ and $j$. The edge between these two labels is the conditional probability of the labels, $w_{i,j} = P(i \,|\, j) = C_{i,j}/C_j$. We also assume an additional "background" node. Edges to the background node have weight $0$ and edges from the background node to label $i$ are weighted with the marginal probability of node $i$. Because each non-root node in the tree corresponds to exactly one label, the tree has $K + 1$ nodes. We model the tree $\mathcal{T}$ generated from this weighted graph as proportional to the probability of the constituent edges, $p(\mathcal{T}) \propto \prod_{(i,j) \in \mathcal{T}} w_{i,j}$. Because the background node has no non-zero incomming edges, it must be the root node of the resulting tree. Given only this information (ignoring the content of the documents), we can construct the maximum probability tree by running Chu-Liu/Edmonds' algorithm to find the maximum spanning tree starting at the root node.

### 2.2 Tree-structured Dirichlet, truncated stick breaking process

Given $\mathcal{T}$, we will stochastically assign each token in every document to a node in the tree. This is done by a tree-structured Dirichlet, truncated stick breaking process, denoted by $\mathcal{B}$. For a document $d$, we associate each node $k$ in the tree with (1) a stochastic switching variable $\omega_{d,k} \sim \text{Beta}(m, \pi)$ and (2) a multinomial distribution over $k$'s children $\theta_{d,k} \sim \text{Dirichlet}(\alpha)$. The detailed process is as follows: a token in document $d$ traverses the tree from the root. Suppose that the token reaches a node $k$, it will stop at this node with probability $\omega_{d,k}$ or move to one of $k$'s child nodes with probability $1 - \omega_{d,k}$. If moving on, the token will choose a child node $k'$ of $k$ with probability $\theta_{d,k,k'}$.

This process provides a prior distribution over all nodes in a tree. The process is, in spirit, similar to the TSSB prior [10] in which a datum traverses downward from the root and can stop at any node in the tree. However, since the number of nodes in our tree is fixed, instead of using stick breaking processes for distributions over the depth and width of the tree, we use truncated stick breaking processes and Dirichlet distributions respectively.

### 2.3 Restricted subtrees

With the tree $\mathcal{T}$ and the prior distribution $\mathcal{B}$ over all its nodes, we can assign each token to a node by first computing the probability of this token being at every node in the tree, and then sampling from this distribution. However, doing so is inefficient given the large number of labels. To speed things up and to leverage the information from the labels, for each document we only consider assigning its tokens to a subset of nodes, called *restricted subtree* $\mathcal{T}_d$, based on the set of labels of the document. For a document $d$ having labels $\boldsymbol{t}_d$, we consider three ways of constructing $\mathcal{T}_d$:

1. Exact paths ($\mathcal{T}_d^1$): contains nodes on the paths from the root to document's label nodes $\boldsymbol{t}_d$.
2. Inside subtrees ($\mathcal{T}_d^2$): contains nodes in $\mathcal{T}_d^1$ and all nodes in the subtrees rooted at the document's label nodes $\boldsymbol{t}_d$.
3. Inside-outside subtrees ($\mathcal{T}_d^3$): contains nodes in $\mathcal{T}_d^2$ and all nodes in the subtrees rooted at the first-level nodes on paths from the root to document's label nodes $\boldsymbol{t}_d$.

# 3 Inference

Given a set of documents with observed words $\boldsymbol{w}$ and labels $\boldsymbol{t}$, the inference task is to find the posterior distribution over the latent variables. We approximate TreeLaD's posterior by collapsing out $\boldsymbol{\theta}$ and $\boldsymbol{\omega}$, alternating between (1) sampling node assignments for each token $z_{d,n}$ and (2) sampling topics $\phi_k$. Currently, we fix the tree structure by running Chu-Liu/Edmonds' algorithm to find the maximum spanning tree using only label co-occurrence information as described in § 2.1. In principle, in order to incorporate the word usage, the tree should be updated during inference to take into account the word distribution at each tree node. We leave this as our future work.

## 3.1 Sampling $z_{d,n}$

The probability of assigning a token $w_{d,n}$ to a node specified by a path $\mathcal{P} = (\text{root}, \cdots, k)$ from the root to a node $k$ is $P(z_{d,n} = k \,|\, \text{rest}) \propto$

$$\frac{N_{d,k}^{-d,n} + m\pi}{N_{d,\geq k}^{-d,n} + \pi} \prod_{i\in\mathcal{P}\setminus\{k\}} \frac{N_{d,>i}^{-d,n} + (1-m)\pi}{N_{d,\geq i}^{-d,n} + \pi} \cdot \prod_{j\in\mathcal{P}\setminus\{\text{root}\}} \frac{N_{d,\geq j}^{-d,n} + \alpha}{\sum_{j'\in\mathcal{C}_{d,\sigma(j)}} (N_{d,\geq j'}^{-d,n} + \alpha)} \tag{1}$$

where $N_{d,k}$ is the number of tokens in document $d$ assigned to node $k$, $N_{d,>k}$ is the number of tokens in document $k$ assigned to any nodes in the subtree rooted at $k$ excluding $k$. We define $N_{d,\geq k} \equiv N_{d,k} + N_{d,>k}$. Conventionally, superscript ${}^{-d,n}$ denotes counts excluding $w_{d,n}$. We also use $\sigma(k)$ to denote the parent node of $k$ and $\mathcal{C}_{d,k}$ to denote the set of children of $k$ in the document-specific tree $\mathcal{T}_d$.

## 3.2 Sampling topics $\phi_k$

The topics in our tree form a cascaded Dirichlet-multinomial chain where the topic $\phi_k$ at a node $k$ is draw from a Dirichlet distribution with the mean vector being the topic $\phi_{\sigma(k)}$ at the parent node $\sigma(k)$. In our inference process, we explicitly sample the topic at each node, following the approach described in [11]. More specifically, for a node $k$, we sample $\phi_k \sim \text{Dirichlet}(\boldsymbol{m}_k + \tilde{\boldsymbol{m}}_k + \gamma \cdot \phi_{\sigma(k)})$ where $\boldsymbol{m}_k$ is the word type count vector at node $k$. In other words, $m_{k,v}$ is the number of times that word type $v$ is assigned to node $k$. $\tilde{\boldsymbol{m}}_k$ is a smoothed count vector in which $\tilde{m}_{k,v}$ captures the number of times node $k$ is used when sampling $v$ at any of $k$'s children nodes. $\tilde{\boldsymbol{m}}_k$ can be approximated by sampling from an Antoniak distribution [12, 13] or summing over counts from all $k$'s children using minimal/maximal path assumptions [14, 15]. In this work, for efficiency we choose to approximate the counts using the two path assumptions. More specifically, the smoothed count vector $\tilde{\boldsymbol{m}}_k$ at node $k$ will be the sum of the propagated count vectors from all $k$'s children, which are defined as follow:

- Minimal path: each child node $i$ of $k$ will propagate a value of 1 to $k$ if $m_{i,v} > 0$.
- Maximal path: each child node $i$ of $k$ will propagate its full count vector $\boldsymbol{m}_i$ to $k$.

The sampling process starts from the bottom of the tree to compute the smoothed count vectors for all nodes in the tree. After reaching to root node, we perform the actual sampling in a top-down manner.

# 4 Empirical Evaluations

We run TreeLaD on a set of congressional bill descriptions that are discussed during the $112^{th}$ U.S. Congress, each of which is labeled with a set of subjects. The dataset is collected from `https://www.govtrack.us/`. After performing standard pre-processing steps such as stemming, removing stop words, removing short documents (having less than 5 tokens) etc, we have 12,299 documents with 5,000 words in the vocabulary and 281 unique labels. We run our Gibbs sampler for 1,000 iterations. Figure 2 shows a small portion of the label tree that is learned by our TreeLaD using minimal path assumption and inside-outside subtrees.

# 5 Conclusion

In this paper, we propose TreeLaD, a hierarchical topic model for multi-labeled documents. TreeLaD aims to capture the dependency between labels using a tree-structure hierarchy. We

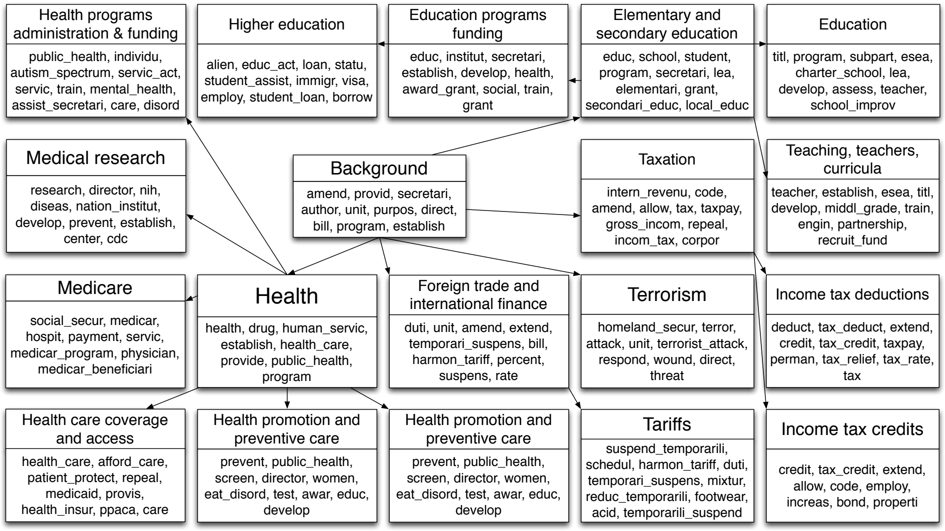

Health programs administration & funding
public_health, individu, autism_spectrum, servic_act, servic, train, mental_health, assist_secretari, care, disord
Higher education
alien, educ_act, loan, statu, student_assist, immigr, visa, employ, student_loan, borrow
Education programs funding
educ, institut, secretari, establish, develop, health, award_grant, social, train, grant
Elementary and secondary education
educ, school, student, program, secretari, lea, elementari, grant, secondari_educ, local_educ
Education
titl, program, subpart, esea, charter_school, lea, develop, assess, teacher, school_improv
Medical research
research, director, nih, diseas, nation_institut, develop, prevent, establish, center, cdc
Background
amend, provid, secretari, author, unit, purpos, direct, bill, program, establish
Taxation
intern_revenu, code, amend, allow, tax, taxpay, gross_incom, repeal, incom_tax, corpor
Teaching, teachers, curricula
teacher, establish, esea, titl, develop, middl_grade, train, engin, partnership, recruit_fund
Medicare
social_secur, medicar, hospit, payment, servic, medicar_program, physician, medicar_beneficiari
Health
health, drug, human_servic, establish, health_care, provide, public_health, program
Foreign trade and international finance
duti, unit, amend, extend, temporari_suspens, bill, harmon_tariff, percent, suspens, rate
Terrorism
homeland_secur, terror, attack, unit, terrorist_attack, respond, wound, direct, threat
Income tax deductions
deduct, tax_deduct, extend, credit, tax_credit, taxpay, perman, tax_relief, tax_rate, tax
Health care coverage and access
health_care, afford_care, patient_protect, repeal, medicaid, provis, health_insur, ppaca, care
Health promotion and preventive care
prevent, public_health, screen, director, women, eat_disord, test, awar, educ, develop
Health promotion and preventive care
prevent, public_health, screen, director, women, eat_disord, test, awar, educ, develop
Tariffs
suspend_temporarili, schedul, harmon_tariff, duti, temporari_suspens, mixtur, reduc_temporarili, footwear, acid, temporarili_suspend
Income tax credits
credit, tax_credit, extend, allow, code, employ, increas, bond, properti

Figure 2: A portion of the label tree learned by TREELAD from a collection of congressional bill descriptions during the $112^{th}$ U.S. Congress

present a Gibbs sampling inference algorithm to approximate the model's posterior distribution. Preliminary experiments on a set of multi-labeled congressional bill descriptions show promising results and the learned label hierarchy captures qualitative relationships between labels in the data. We are working toward evaluating TREELAD more thoroughly on the quality of the learned hierarchy as well as on other downstream applications such as multi-label document classification.

## References

[1] Blei, D. M. Probabilistic topic models. *Commun. ACM*, 55(4):77–84, 2012.

[2] Blei, D. M., A. Ng, M. Jordan. Latent Dirichlet allocation. *Journal of Machine Learning Research*, 2003.

[3] Blei, D. M., J. D. McAuliffe. Supervised topic models. In *NIPS*. 2007.

[4] Lacoste-Julien, S., F. Sha, M. I. Jordan. DiscLDA: Discriminative learning for dimensionality reduction and classification. In *NIPS*, pages 897–904. 2008.

[5] Mimno, D. M., A. McCallum. Topic models conditioned on arbitrary features with Dirichlet-multinomial regression. In *UAI*, pages 411–418. 2008.

[6] Ramage, D., D. Hall, R. Nallapati, et al. Labeled LDA: A supervised topic model for credit attribution in multi-labeled corpora. In *EMNLP*. 2009.

[7] Wang, C., D. Blei, L. Fei-Fei. Simultaneous image classification and annotation. In *CVPR*. 2009.

[8] Ramage, D., C. D. Manning, S. Dumais. Partially labeled topic models for interpretable text mining. In *SIGKDD*, pages 457–465. 2011.

[9] Rubin, T. N., A. Chambers, P. Smyth, et al. Statistical topic models for multi-label document classification. *Mach. Learn.*, 88(1-2):157–208, 2012.

[10] Adams, R., Z. Ghahramani, M. Jordan. Tree-structured stick breaking for hierarchical data. In *NIPS*. 2010.

[11] Ahmed, A., L. Hong, A. Smola. The nested Chinese restaurant franchise process: User tracking and document modeling. In *ICML*. 2013.

[12] Antoniak, C. E. Mixtures of Dirichlet processes with applications to Bayesian nonparametric problems. *The Annals of Statistics*, 2(6):1152–1174, 1974.

[13] Teh, Y. W., M. I. Jordan, M. J. Beal, et al. Hierarchical Dirichlet processes. *JASA*, 101(476), 2006.

[14] Cowans, P. J. *Probabilistic Document Modelling*. Ph.D. thesis, University of Cambridge, 2006.

[15] Wallach, H. M. *Structured Topic Models for Language*. Ph.D. thesis, University of Cambridge, 2008.