Tree-based Label Dependency Topic Models

Viet-An Nguyen[1], Jordan Boyd-Graber[1,2,4], Jonathan Chang[5] and Philip Resnik[1,3,4]

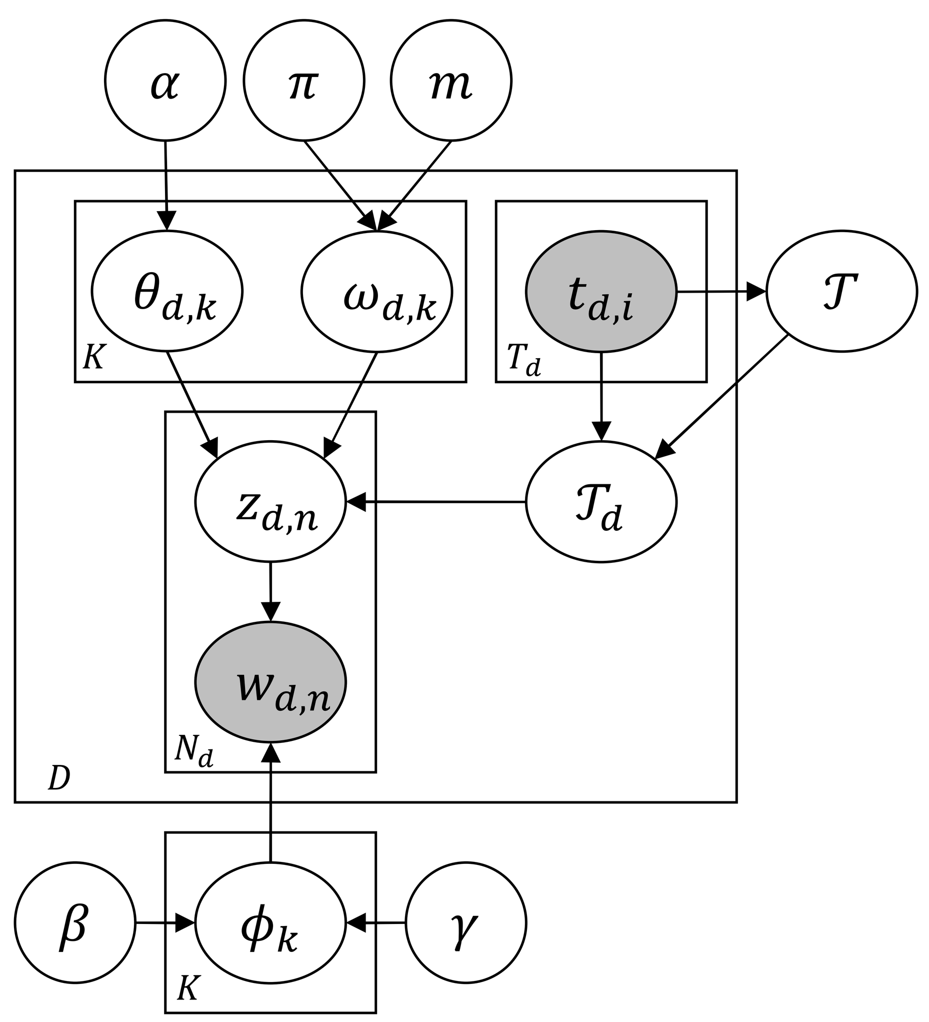
[1]Computer Science, [2]iSchool, [3]Linguistics, [4]UMIACS
University of Maryland, College Park, MD

[5]Facebook
Menlo Park, CA

Motivations

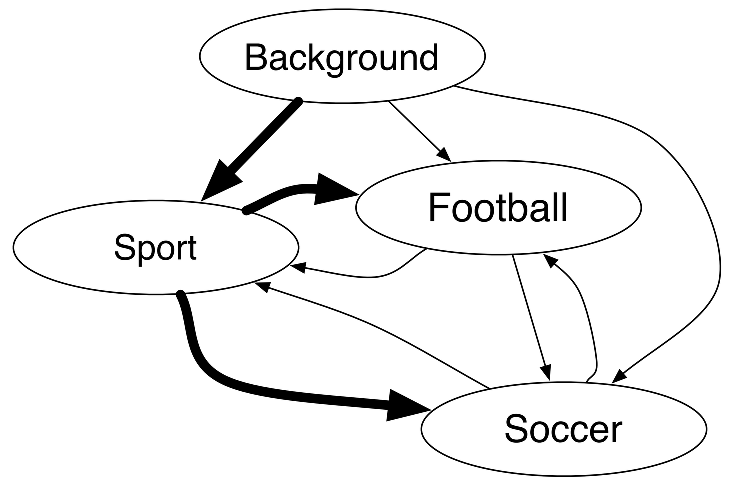
- **Multi-labeled data**, in which each document is tagged with a set of labels, are ubiquitous.
- Previous topic models for multi-labeled data often
  - assume labels are independent
  - or capture the dependencies among labels by projecting them onto some latent space
- In this work, we propose a tree-based label dependency topic model, TREELAD, which **captures the label dependencies using a tree-structured hierarchy**.

Tree-based label dependency topic model



1. Create the label graph $\mathcal{G}$ and generate a tree $\mathcal{T}$ from $\mathcal{G}$ (See I)
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 III)
   (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 stochastic 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 II)
      ii. Draw $w_{d,n} \sim \text{Mult}(\phi_{z_{d,n}})$

(I) Generating tree

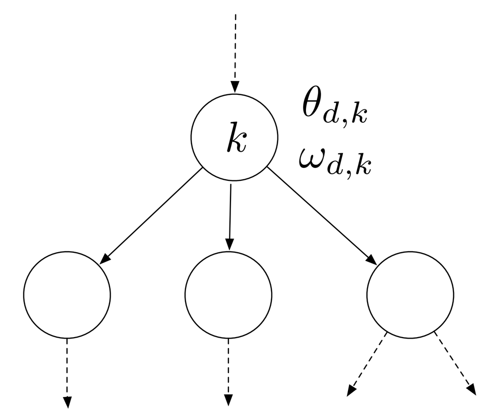


- Construct a complete directed graph where each label is a node and the edge from i to j has weight $w_{i,j} = P(i \mid j) = C_{i,j}/C_j$.
- Add a "background" node to the graph and add edges from the background node to all nodes, with the weight being the marginal probability.
- Run Chu-Liu/Edmonds' algorithm to find the maximum spanning tree starting at the background node.

(II) Assigning tokens

For each document d, we associate each node k with:

- a stochastic switching variable $\omega_{d,k} \sim \text{Beta}(m, \pi)$
- a multinomial distribution over k's children $\theta_{d,k} \sim \text{Dirichlet}(\alpha)$

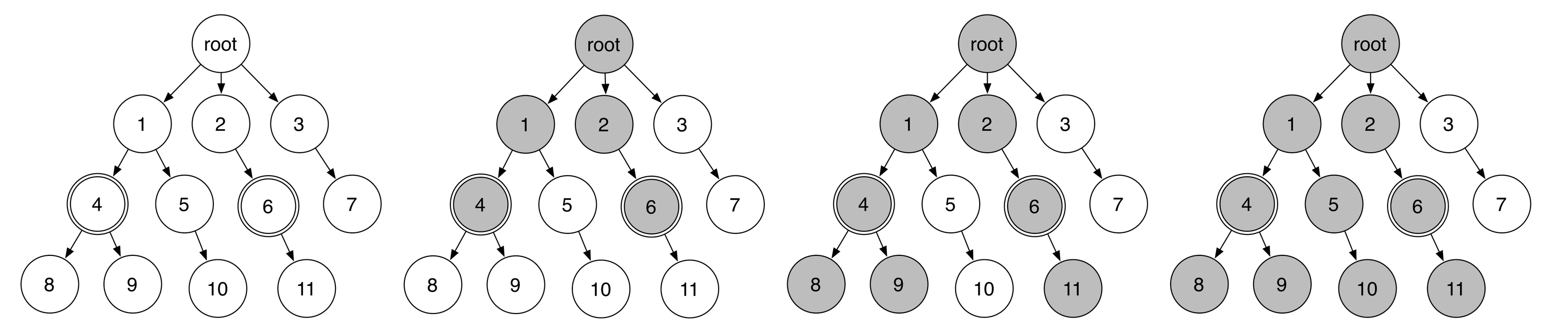


We stochastically assign each token to a node in the tree as follows:

- The token starts traversing the tree from the root.
- Suppose 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 $\theta_{d,k,k'}$.

(III) Restricting subtrees

To avoid considering all labels and to leverage the information from the labels, for each document d, only a subset of nodes, called *restricted subtree* $\mathcal{T}_d$, can generate tokens of d.



(a) $\mathcal{T}$ (b) $\mathcal{T}_d^1$ (c) $\mathcal{T}_d^2$ (d) $\mathcal{T}_d^3$

Different restricted subtrees for a document labeled with nodes 4 and 6 (double-circled in (a)), (b) $\mathcal{T}_d^1$ contains nodes from the root to nodes 4 and 6, (c) $\mathcal{T}_d^2$ contains $\mathcal{T}_d^1$ and nodes in the subtrees rooted at nodes 4 and 6, and (d) $\mathcal{T}_d^3$ contains $\mathcal{T}_d^2$ and other nodes in the subtrees rooted at nodes 1 and 2 (first-level nodes on paths from the root to nodes 4 and 6 respectively).

Inference

After running Chu-Liu/Edmonds' algorithm, we fix the tree structure and alternate between the following two steps:

1. Sample node assignment $z_{d,n}$ for each token:
   $P(z_{d,n} = k \mid \text{rest}) \propto$

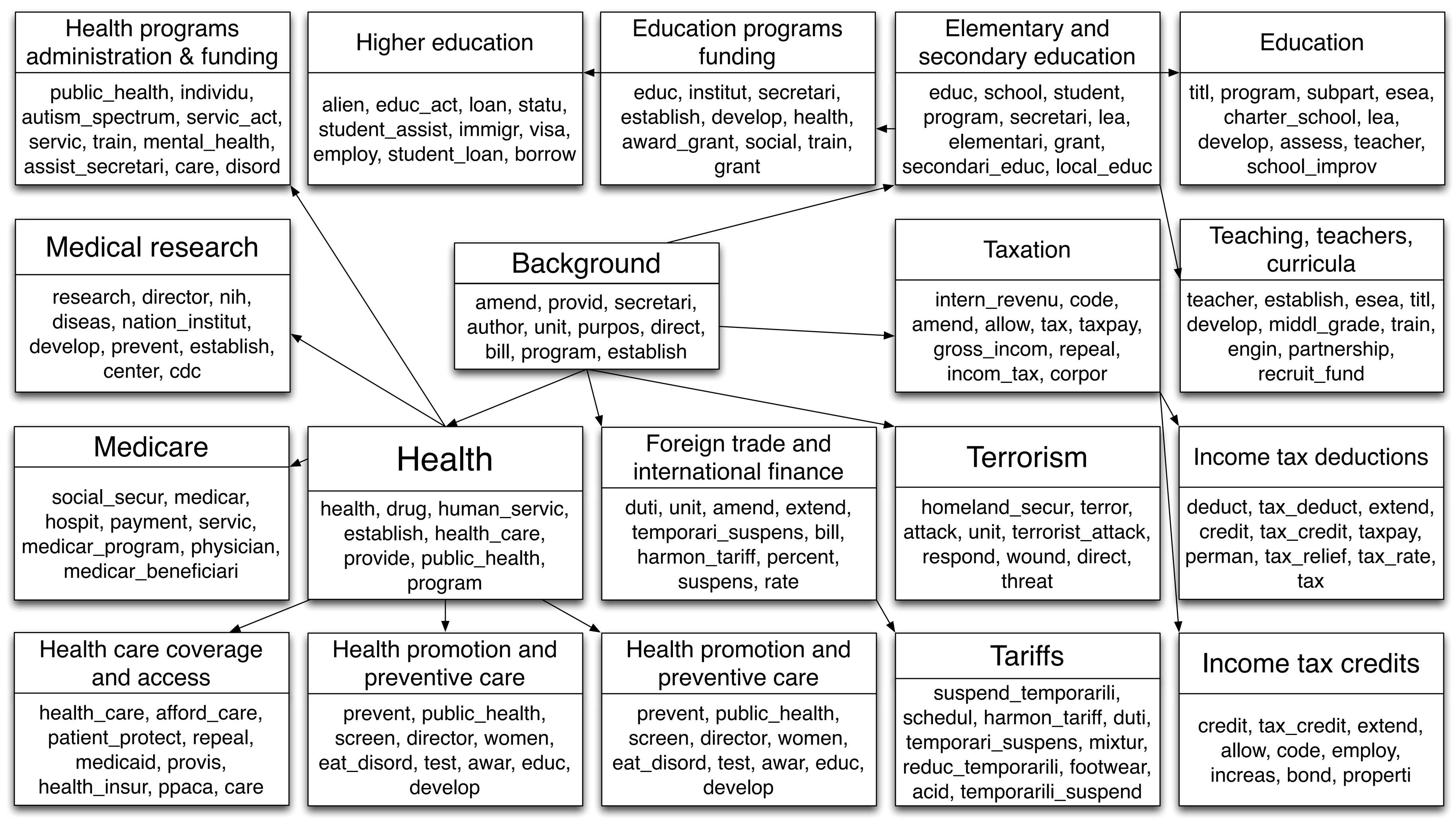
$$\frac{N_{d,k}^{-d,n} + m\pi}{N_{d,\geq k}^{-d,n} + \pi} \prod_{i \in \mathcal{P} \setminus \{k\}} \frac{N_{d,>i}^{-d,n} + (1-m)\pi}{N_{d,\geq i}^{-d,n} + \pi} \cdot \prod_{j \in \mathcal{P} \setminus \{\text{root}\}} \frac{N_{d,\geq j}^{-d,n} + \alpha}{\sum_{j' \in \mathcal{C}_{d,\sigma(j)}} (N_{d,\geq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

   - $N_{d,k}$ 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. $N_{d,\geq k} \equiv N_{d,>k} + N_{d,k}$.
2. Sample topic ϕ_k at each node in the tree:

$$\phi_k \sim \text{Dirichlet}(m_k + \tilde{m}_k + \gamma \cdot \phi_{\sigma(k)})$$

   - $m_{k,v}$ is the number of times that word type v is assigned to node k
   - $\tilde{m}_k$ 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. $\tilde{m}_k$ is estimated using either minimal or maximal path assumption.
   - $\sigma(k)$ is the parent node of k.

Part of the label hierarchy learned from Congressional bills



Future directions

- Update the tree structure during inference to capture the word usages
- Evaluate more formally the proposed model on downstream applications such as multi-label document classification

Part of this work was done while the first author interned at Facebook.

vietan@cs.umd.edu, jbg@umiacs.umd.edu, jonchang@fb.com, resnik@umd.edu