1 Technical Details for ANCHOR

Let $\bar{Q}_{i,j}$ be a row-normalized matrix representing conditional probability of observing the word $j$ given that we see the word $i$ in a document. The novel result of ANCHOR states that the cooccurrence of any non-anchor word can be explained by a linear combination of those from anchor words,

$$\bar{Q}_{i,:} = \sum_{g_k \in G} C_{i,k} \bar{Q}_{g_k,:},$$

(1)

where $g_k$ is an anchor word for topic $k$ and $C_{i,k}$ is the probability of seeing topic $k$ given that we observe the word $i$ in the document. After learning coefficient matrix $C$, ANCHOR recovers topic word distribution $\bar{Q} = p(w = i | z = k)$ from $C$ using Bayes’ rule (Nguyen et al., 2014).

The supervised anchor algorithm (SANCHOR) improves on ANCHOR by adding additional dimensions as columns to $\bar{Q}$. Each column encodes the regression value associated with a word—for example, in a sentiment analysis setting, the conditional probability of observing a sentiment value given an observation of word. Therefore, SANCHOR indirectly incorporates document-level supervision information by producing a new set of anchor words, hence a new set of topics. Nguyen et al. (2015) provide details on how to compute word-level regression values for sentiment analysis. As they discuss, the anchor algorithms do not produce per-document assignment of topics. Therefore variational inference is used to learn documents’ topic distributions, which we use as features in our regression experiments.1

2 Technical Details for Supervised Nested Latent Dirichlet Allocation

Supervised Nested Latent Dirichlet Allocation (SNLDA) extends SLDA to capture the topics in a tree-structured hierarchy. The input of SNLDA is similar to that of SLDA, which consists of a collection of $D$ documents, each is associated with a response variable. In our setting, the response variable is binary. Here is the generative process of SNLDA:

1. For each node $k$ in the tree $T$, draw a topic $\phi_k$ and a regression parameter $\eta_k$.
   - If $k$ is the root, $\phi_k \sim \text{Dir}(\beta_0 \mathbf{1})$ and $\eta_k = 0$.
   - If $k$ is first-level, $\phi_k \sim \text{Dir}(\beta_1 \phi_k^\star)$ and $\eta_k \sim \mathcal{N}(0, \sigma_1)$, where $\phi_k^\star$ specifies an informed prior.
   - Otherwise, $\phi_k \sim \text{Dir}(\beta_k \phi_{p_k})$ and $\eta_k \sim \mathcal{N}(0, \sigma_k)$, where $l_k$ and $p_k$ are the level and the parent of node $k$ respectively.

2. For each document $d \in [1, D]$
   - For each non-terminal node $k$ in the tree
     (a) Draw a distribution over $k$’s children $\theta_{d,k} \sim \text{Dir}(\alpha_{l_k})$.
     (b) Draw a stochastic switching variable $\omega_{d,k} \sim \text{Beta}(\pi \gamma_{l_k})$.
   - For each token $n \in [1, N_d]$
     (a) Draw a node $z_{d,k} \sim \mathcal{B}(\theta_{d,k}, \omega_d)$ (c.f. Section 2.2).
     (b) Draw $w_{d,k} \sim \text{Mult}(\phi_{d,k})$.
   - The probability of the binary response being 1 is
     $$p(y_d = 1) = \Phi \left( \sum_{k \in T} \frac{N_{d,k} \eta_k}{N_d} \right),$$

where $\Phi(x) = \exp(x)/(1 + \exp(x))$ is the logistic function, $N_{d,k}$ is the number of tokens in document $d$ assigned to node $k$, and marginal count is denoted by ‘$\cdot$’.

2.1 Generating Polarized Topic Hierarchy

In SNLDA, we assume that the structure of the topic tree $T$ is given with a fixed height $L$ and a predefined branching factor $K_l$ at each level $l$. For each node $k$ in the tree, we draw (1) a topic $\phi_k$ specifying what this node $k$ is about and (2) a regression parameter $\eta_k$ specifying

1http://www.cs.princeton.edu/~blei/lda-c/
the weight of \( k \) in capturing the binary response variable. More specifically, the root node’s topic is drawn from a symmetric Dirichlet prior \( \text{Dir}(\beta_p \mu) \). Topics at first-level nodes are drawn from either an uninform or informed Dirichlet prior \( \text{Dir}(\beta_i \phi_k^* \mu) \). Topic \( \phi_k \) at a lower-level node \( k \) is drawn from a Dirichlet prior \( \text{Dir}(\beta_k \phi_p \mu) \) whose mean vector \( \phi_p \) is the topic of the parent node \( p_k \).

2.2 Generating Words

Given the topic hierarchy \( T \), we use the tree-structured Dirichlet, truncated stick breaking process prior, denoted by \( B \) (Nguyen et al., 2013a), to stochastically assign each token in every document to a node in the tree. Given a document \( d \), we associate each node \( k \) in the tree with (1) a stochastic switching variable \( \omega_{d,k} \sim \text{Beta}(\pi, \gamma_k) \) and (2) a multinomial distribution over \( k \)’s children \( \theta_{d,k} \sim \text{Dir}(\alpha_k) \). A token \( n \) in document \( d \) is generated by traversing the tree from the root downward. Suppose that the token reaches a node \( k \), it will stop at this node with probability \( 1 - \omega_{d,k} \) or move to one of \( k \)’s child nodes with probability \( \omega_{d,k} \). If moving on, a child node \( j \) of \( k \) is chosen with probability \( \theta_{d,k,j} \). Given the chosen node \( z_{d,n} \), we draw the word token from the corresponding topic \( w_{d,k} \sim \text{Mult}(\phi_{z_{d,k}}) \).

2.3 Generating Responses

Similar to previous supervised topic models (Blei and McAuliffe, 2007; Wang et al., 2009; Nguyen et al., 2013b), we model the response variable \( y_d \) of each document \( d \) based on the empirical distribution of the document over all topics. In particular, the probability of \( y_d \) being 1 is:

\[
p(y_d = 1 \mid z_{d}, \eta) = \Phi \left( \sum_{k \in T} \frac{N_{d,k}}{N_{d,c}} \eta_k \right) \tag{2}
\]

3 Posterior Inference

Given observed data of \( D \) documents \( \{w_d\} \), each is associated with a binary response \( y_d \), we estimate the posterior distribution over latent variables of SNLDA using stochastic EM. We alternate between (1) sampling the node assignment for each document token, (2) sampling node topics, and (3) optimizing the regression parameters.

3.1 Sampling Node Assignments for Tokens

To assign a token \( w_{d,n} \) to a node in the tree, we first propose a node by sampling from an approximate conditional probability distribution over the tree nodes and then use the Metropolis-Hastings algorithm to either accept or reject the proposed node.

To propose a node, we start at the root node and recursively sample a node level-by-level. At each node \( k \), we choose between either staying at \( k \) or moving to a child node \( j \) of \( k \) with the following probabilities:

\[
\begin{align*}
\lambda_{d,n}^{k \rightarrow k} \cdot \phi_{k,w_{d,n}}, & \quad \text{stay on;} \\
\lambda_{d,n}^{k \rightarrow j} \cdot \phi_{j,w_{d,n}}, & \quad \text{move to a child } j \text{ of } k;
\end{align*}
\]

where \( \lambda_{d,n}^{k \rightarrow k} \) and \( \lambda_{d,n}^{k \rightarrow j} \) are probabilities of a token staying at node \( k \) and moving to a child node \( j \) of \( k \) respectively, and are computed as

\[
\begin{align*}
\lambda_{d,n}^{k \rightarrow k} &= \frac{N_{d,k}^* - N_j + \gamma_k \pi}{N_{d,k} \gamma_k + \gamma_k} \\
\lambda_{d,n}^{k \rightarrow j} &= (1 - \lambda_{d,n}^{k \rightarrow k}) \frac{N_j^* + \alpha_k}{N_{d,k} \gamma_k + \alpha_k} \tag{4}
\end{align*}
\]

In Equations 4 and 5, \( N_{d,k} \) is the number of tokens in document \( d \) assigned to node \( k \), \( N_{d,j} \) is the number of document \( d \)’s tokens assigned to any nodes in the subtree rooted at \( k \), and \( N_{d,\geq k} = N_{d,k} \) denotes the number of tokens in document \( d \) assigned to any descendant nodes of \( k \). We use the superscript \( -d,n \) to denote the exclusion of the assignment of token \( n \) in document \( d \) from the corresponding counts. \( K_k \) is the number of child nodes of \( k \).

In Equation 3, if the token decides to stay at the current node \( k \), node \( k \) will be the proposed node. The proposal probability for a node \( k \) is \( q(k) = \)

\[
\frac{\sum_{j \in C_k} \lambda_{d,n}^{k \rightarrow j} \cdot \phi_{j,w_{d,n}} \prod_{(i=j) \in P(k)} \lambda_{d,n}^{i \rightarrow j} \cdot \phi_{j,w_{d,n}}}{\prod_{j \in C_k} \sum_{j' \in C_k} \lambda_{d,n}^{k \rightarrow j'} \cdot \phi_{j',w_{d,n}}} \tag{6}
\]

where \( C_i \) denotes the set of nodes which consists of \( i \) and all its child nodes and \( P(k) \) denotes the path from the root to node \( k \).

The true probability of assigning \( w_{d,n} \) to \( k \) is

\[
p(k) = \lambda_{d,n}^{k \rightarrow k} \cdot \phi_{k,w_{d,n}} \prod_{(i=j) \in P(k)} \lambda_{d,n}^{i \rightarrow j} \tag{7}
\]

Thus, assume the current node is \( k_{\text{CUR}} \) and the proposed node is \( k_{\text{PROP}} \), according to the Metropolis-Hastings algorithm, we accept the proposed node with the probability of

\[
\min \left( 1, \frac{q(k_{\text{PROP}}) p(k_{\text{CUR}})}{q(k_{\text{CUR}}) p(k_{\text{PROP}})} \right) \tag{8}
\]

3.2 Sampling Node Topics

The topics on each path start at a first-level node of the tree form a cascaded Dirichlet-multinomial chain where the topic \( \phi_k \) at a node \( k \) is drawn from a Dirichlet distribution whose mean is the topic \( \phi_p \) at the parent node \( p_k \). Following the approach described in (Ahmed et al.,
we explicitly sample the topic $\phi_k$ at each non-rooted node $k$ as follow

$$\phi_k \sim \begin{cases} 
\text{Dir}(m_k + \tilde{n}_k + \beta_1 \phi_k^*), & \text{if } l_k = 1; \\
\text{Dir}(m_k + \tilde{n}_k + \beta_l \phi_{p_k}), & \text{if } l_k > 1.
\end{cases}$$ (9)

where $m_k$ is the count vector of tokens assigned to node $k$ and $\tilde{n}_k$ denotes the count vector propagated from $k$'s children. The sampling process starts from the tree leaves to compute the smoothed count vector $\tilde{n}_k$ for all non-rooted nodes in the tree. After reaching first-level nodes, we perform the actual sampling in a top-down manner.

3.3 Optimizing Regression Parameters

We update the regression parameters $\eta$ by using L-BFGS (Liu and Nocedal, 1989) to optimize $\hat{L}(\eta)$

$$= \sum_{d=1}^{D} y_d \log p(y_d = 1) + (1 - y_d) \log p(y_d = 0)$$

$$- \sum_{k \in T \setminus \{\text{root}\}} \frac{\eta_k^2}{2\sigma_{l_k}}$$ (10)

References


