Learning Concept Graphs from Text with Stick-Breaking Priors

America L. Chambers, Padhraic Smyth, Mark Steyvers

Department of Computer Science, University of California, Irvine

Presented by Lingbo Li

NIPS, 2010
Concept Graph

- A rooted, directed graph where the nodes represent thematic units (called concepts) and the edges represent relationships between concepts.
Introduction II

Review

- LDA: no relationships between topics
- hLDA: tree structures; not well suited for large documents like Wikipedia
- hPAM(hierarchial Pachinko allocation model): learn a set of topics arranged in a fixed-sized graph
- Other methods for creating concept graphs: hierarchical clustering, pattern mining, formal concept analysis to construct ontologies, etc

The proposed approach

- Introduce a flexible probabilistic framework for learning general graph structures from text
- Utilize both unlabeled and labeled documents and prior knowledge in the form of existing graph structures
Stick-Breaking Distributions

- Stick-breaking distributions $\mathcal{P}(.)$ are discrete probability distributions of the form $\mathcal{P}(.) = \sum_{j=1}^{\infty} \pi_j \delta_{x_j}(.)$ where $\sum_{j=1}^{\infty} \pi_j = 1$, $0 \leq \pi_j \leq 1$
- $\delta_{x_j}(.)$ is the delta function centered at the atom $x_j$
- $x_j$ are sampled independently from a base distribution $H$
- The stick-breaking weights $\pi_j$ have the form

\[
\pi_1 = \nu_1, \quad \pi_j = \nu_j \prod_{k=1}^{j-1} (1 - \nu_k), \quad \text{for } j = 2, 3, \ldots, \infty
\]

$\nu_j \sim \text{Beta}(\alpha_j, \beta_j)$
- Stick-breaking process is non-exchangeable
For each node, there is a distribution (denoted as $\mathcal{P}_t$) that governs the probability of transitioning from node $t$ to another node in the graph.

Constraints on choosing $\mathcal{P}_t$

- Making a new transition must have non-zero probability
- Making a transition to a new node must have non-zero probability
Stick-breaking Distributions III

Defining $\mathcal{P}_t$

- Initial graph structure $G_0$ with $t = 1, ..., T$ nodes
- For each note $t$, define a feasible $\mathcal{F}_t$ as the collection of nodes to which $t$ can transition
- $\mathcal{F}_t$ is some subset of the nodes in $G_0$
- Add a special node called the “exit node” to $\mathcal{F}_t$
- The exact form of $\mathcal{P}_t$ is

$$
\mathcal{P}_t(.) = \sum_{j=1}^{\left|\mathcal{F}_t\right|} \pi_{tj} \delta_{f_{ij}}(.) + \sum_{j=\left|\mathcal{F}_t\right|+1}^{\infty} \pi_{tj} \delta_{x_{ij}}(.)
$$

- $\mathcal{F}_t$ could only include any node at a lower depth
Generative Process

GraphLDA

1. For node $t \in \{1, \ldots, \infty\}$
   i. Sample stick-break weights $\{v_{tj}\}|\alpha, \beta \sim \text{Beta}(\alpha, \beta)$
   ii. Sample word distribution $\phi_t|\eta \sim \text{Dirichlet}(\eta)$

2. For document $d \in \{1, 2, \ldots, D\}$
   i. Sample a distribution over levels $\tau_d|a, b \sim \text{Beta}(a,b)$
   ii. Sample path $p_d \sim \{P_t\}_{t=1}^{\infty}$
   iii. For word $i \in \{1, 2, \ldots, N_d\}$
       Sample level $l_{d,i} \sim \text{TruncatedDiscrete}(\tau_d)$
       Generate word $x_{d,i}|\{p_d, l_{d,i}, \Phi\} \sim \text{Multinomial}(\phi_{p_d[l_{d,i}]}$)

- In the case where we observe labeled documents and an initial graph structure, the paths for document $d$ is restricted
- Use Geometric distribution over levels in all experiments
Inference

- Marginalize over the topic distribution $\phi_t$ and the stick-breaking weights $\{\nu_{tj}\}$
- Use a collapsed Gibbs sampler to infer the path assignment $p_d$ for each document, the level distribution parameter $\tau_d$ for each document, and the level assignment $l_{di}$ for each word
- Place an Exponential prior on both $\eta$ and $\beta$
- Use a Metropolis-Hastings sampler to mix over stick-breaking permutations
Inference

Sampling Paths

\[ p(p_d|x, \lambda, p_{-d}, \tau) \propto p(x_d|x_{-d}, \lambda, p) \cdot p(p_d|p_{-d}) \]  \hspace{1cm} (1)

The first term in Equation 1 is the probability of all words in the document given the path \( p_d \). We compute this probability by marginalizing over the topic distributions \( \phi_t \):

\[ p(x_d|x_{-d}, \lambda, p) = \prod_{i=1}^{\lambda_d} \left( \prod_{v=1}^{V} \frac{\Gamma(\eta + N_{p_d[i],v})}{\Gamma(\eta + N_{p_d[i],v} - d)} \right) \right) * \frac{\Gamma(V \eta + \sum_v N_{p_d[i],v})}{\Gamma(V \eta + \sum_v N_{p_d[i],v})} \]

We use \( \lambda_d \) to denote the length of path \( p_d \). The notation \( N_{p_d[i],v} \) stands for the number of times word type \( v \) has been assigned to node \( p_d[i] \). The superscript \(-d\) means we first decrement the count \( N_{p_d[i],v} \) for every word in document \( d \).

The second term is the conditional probability of the path \( p_d \) given all other paths \( p_{-d} \). We present the sampling equation under the assumption that there is a maximum number of nodes \( M \) allowed at each level. We first consider the probability of sampling a single edge in the path from a node \( x \) to one of its feasible nodes \( \{y_1, y_2, \ldots, y_M\} \) where the node \( y_1 \) has the first position in the stick-breaking permutation, \( y_2 \) has the second position, \( y_3 \) the third and so on.

We denote the number of paths that have gone from \( x \) to \( y_1 \) as \( N(x,y_1) \). We denote the number of paths that have gone from \( x \) to a node with a strictly higher position in the stick-breaking distribution than \( y_i \) as \( N(x,y_{i+1}) \). That is, \( N(x,y_1) = \sum_{k=1}^{M} N(x,y_k) \). Extending this notation we denote the sum \( N(x,y_i) + N(x,y_{i+1}) \) as \( N(x,y_i) \). The probability of selecting node \( y_i \) is given by:

\[ p(x \rightarrow y_i | p_{-d}) = \frac{\alpha + N(x,y_i)}{\alpha + \beta + N(x,y_1)} \prod_{r=1}^{i-1} \frac{\beta + N(x,y_r)}{\alpha + \beta + N(x,y_r)} \]  \hspace{1cm} for \( i = 1 \ldots M \)

If \( y_m \) is the last node with a nonzero count \( N(x,y_m) \) and \( m \ll M \) it is convenient to compute the probability of transitioning to \( y_i \), for \( i \leq m \), and the probability of transitioning to any node higher than \( y_m \). The probability of transitioning to a node higher than \( y_m \) is given by:

\[ \sum_{k=m+1}^{M} p(x \rightarrow y_k | p_{-d}) = \Delta \left[ 1 - \frac{\beta}{\alpha + \beta} \right] \]

where \( \Delta = \prod_{r=1}^{m} \frac{\beta + N(x,y_r)}{\alpha + \beta + N(x,y_r)} \). A similar derivation can be used to compute the probability of sampling a node higher than \( y_m \) when \( M \) is equal to infinity. Now that we have computed the probability of a single edge, we can compute the probability of an entire path \( p_d \):

\[ p(p_d|p_{-d}) = \prod_{j=1}^{\lambda_d} p(p_{d_j} \rightarrow p_{d,j+1} | p_{-d}) \]
Inference

Sampling Levels

For the \( i \)th word in the \( d \)th document we must sample a level \( l_{di} \) conditioned on all other levels \( l_{-di} \), the document paths, the level parameters \( \tau \), and the word tokens.

\[
p(l_{di} | \mathbf{x}, l_{-di}, \mathbf{p}, \tau) = \left( \frac{\eta + N_{p_d[l_{di}], x_{di}}}{W \eta + N_{p_d[l_{di}]}} \right) \cdot \frac{(1 - \tau_d)^l_{di}}{(1 - (1 - \tau_d)\lambda_d + 1) \tau_d}
\]

The first term is the probability of word type \( x_{di} \) given the topic at node \( p_d[l_{di}] \). The second term is the probability of the level \( l_{di} \) given the level parameter \( \tau_d \).

Sampling \( \tau \) Variables

Finally, we must sample the level distribution \( \tau_d \) conditioned on the rest of the level parameters \( \tau_{-d} \), the level variables, and the word tokens.

\[
p(\tau_d | \mathbf{x}, l, \mathbf{p}, \tau_{-d}) = \left( \prod_{i=1}^{N_d} \frac{(1 - \tau_d)^l_{di} \tau_d}{(1 - (1 - \tau_d)\lambda_d + 1)} \right) \ast \left( \frac{\tau_d^{a-1} (1 - \tau_d)^{b-1}}{B(a, b)} \right)
\]
Inference

Metropolis Hastings for Stick-Breaking Permutations

In addition to the Gibbs sampling, we employ a Metropolis Hastings sampler presented in [10] to mix over stick-breaking permutations. Consider a node $x$ with feasible nodes $\{y_1, y_2, \ldots, y_M\}$. We sample two feasible nodes $y_i$ and $y_j$ from a uniform distribution. Assume $y_i$ comes before $y_j$ in the stick-breaking distribution. Then the probability of swapping the position of nodes $y_i$ and $y_j$ is given by

$$\min\left\{1, \prod_{k=0}^{N(x,y_i)-1} \frac{\alpha + \beta + N(x,y_i)^* + k}{\alpha + \beta + N(x,y_j) + k}, \prod_{k=0}^{N(x,y_j)-1} \frac{\alpha + \beta + N(x,y_j) + k}{\alpha + \beta + N(x,y_i)^* + k}\right\}$$

where $N(x,y_i)^* = N(x,y_i) - N(x,y_j)$. See [10] for a full derivation. After every new path assignment, we propose one swap for each node in the graph.
Simulated Text Data

- Simulate a concept graph with 10 nodes drawn according to stick-breaking process with parameter values $\eta = 0.025, \alpha = 10, \beta = 10, a = 2$ and $b = 5$
- Generate 4000 documents with 250 words each
- The vocabulary size is 1000 words
- Label each edge in the graph with the number of paths
- Label each node based upon the similarity of the learned topic at the node to the topics of the original graph structure
- With no labeled data, the Gibbs sampler is initialized to a root node
- With labeled data, the Gibbs sampler is initialized with the correct placement of nodes to levels
Simulated Text Data

(a) Simulated Graph

(b) Learned Graph (0 labeled documents)

(c) Learned Graph (250 labeled documents)

(d) Learned Graph (4000 labeled documents)
Compare the performance of GraphLDA to hPAM and hLDA on a set of 518 machine-learning articles from Wikipedia.

Restrict to learning a three-level hierarchical structure.

For both GraphLDA and hPAM, the number of nodes at each level was set to 25.

$\beta$ and $\eta$ were initialized to 1 and 0.001, and optimized using a Metropolis Hastings sampler.

All models were run for 9000 iterations for burn-in and samples were collected every 100 iterations thereafter, for a total of 10000 iterations.

The performance was evaluated on a hold-out set consisting of 20% of the articles using both empirical likelihood and the left-to-right evaluation algorithm.
GraphLDA is competitive when computing the empirical log likelihood.

GraphLDA’s lower performance in terms of left-to-right log-likelihood is due to the geometric distribution over levels.
Wikipedia Articles with a Graph Structure

How GraphLDA updates an existing category

- Learn topic distributions for each node using the aforementioned 518 machine-learning Wikipedia articles and their category labels
- Fix the path assignments for the Wikipedia articles after 2000 iterations and introduce a new set of documents (a collection of 400 machine learning abstracts from the ICML)
- Sample paths for the new collection of documents keeping the paths from the Wikipedia articles fixed
Wikipedia Articles with a Graph Structure
Discussion and Conclusion

Conclusion

- Present a flexible non-parametric Bayesian framework for learning concept graphs from text
- Combine unlabeled data with prior knowledge in the form of labeled documents and existing graph structures

Discussion

- May extend to learn multiple paths per document
- Could not scale to large graphs due to the computations
- Approximation inference methods may address this issue