Scalable Text and Link Analysis with Mixed-Topic Link Models

Yaojia Zhu ¹  Xiaoran Yan ¹  Lise Getoor ²  Cristopher Moore ³

¹University of New Mexico  ²The University of Maryland College Park  ³Santa Fe Institute

Aug 13, 2013
## General form of the problem

### Observed data
1. rich information (attributes) about the nodes
2. pairwise relationship (links) between the nodes

### Learning tasks
1. detecting hidden communities in the network
2. predicting node attributes and links
Our work

1. node → document
   attributes → text
   pairwise relationship → citation relationship/hyperlinks

2. learning tasks: document topic detection, link prediction

3. the goal: develop models and algorithms which use both text- and link-information to classify documents by topic and predict missing links with higher accuracy and better scalability.
the original patent for CT (computed tomography)

patent related to CT
patent related to CT
patent related to CT
patent related to CT

Why we are interested in this problem:

1. detecting communities in complex networks beyond topology

2. nodes’ rich information is extremely helpful when the link information is insufficient to the learning task
It’s hard to achieve both high **accuracy** and high **scalability**.
Motivation

It’s hard to achieve both high accuracy and high scalability.

1. The PHITS-PLSA [Cohn, Hofmann (2001)] model is efficient but prone to overfitting.
2. The Relational Topic Model (RTM) [Chang, Blei (2010)] is not quite scalable.
It’s hard to achieve both high accuracy and high scalability.

1. The PHITS-PLSA [Cohn, Hofmann (2001)] model is efficient but prone to overfitting.
2. The Relational Topic Model (RTM) [Chang, Blei (2010)] is not quite scalable.

Why we think we can achieve both:

1. we use a network model, called Ball-Karrer-Newman (BKN) model, recently introduced in the physics and network community, which has an especially simple and efficient inference algorithm
2. we combine the classic text-based PLSA model with the link-based BKN model, and get a fast and simple EM algorithm to analyse both text and links
Probabilistic Latent Semantic Analysis (PLSA) [?]

Graphical model

Generative process for the text

For each document $1 \leq d \leq N$ and each $1 \leq n \leq L_d$, we independently
1. choose a topic $z = z_{dn} \sim \text{Multi}(\theta_d)$, and
2. choose the word $w_{dn} \sim \text{Multi}(\beta_z)$.

Thus the total probability that $w_{dn}$ is a given word $w$ is

$$
\Pr[w_{dn} = w | \theta, \beta] = \sum_{z=1}^{K} \theta_{dz} \beta_{zw}.
$$
Generative process for the links between $d$ and $d'$

1. Choose a topic $z_d \sim \text{Multi}(\theta_d)$.
2. Choose a topic $z_{d'} \sim \text{Multi}(\theta_{d'})$.
3. If $z = z_d = z_{d'}$, choose $A_{dd'} \sim \text{Poi}(\eta_z)$; otherwise, $A_{dd'} = 0$.

$\eta_z$ is the link density for topic $z$. The number of links $A_{dd'}$ between $d$ and $d'$ is drawn from a Poisson distribution

$$A_{dd'} \sim \text{Poi} \left( \sum_{z=1}^{K} \theta_{dz} \theta_{d'z} \eta_z \right).$$
### Generative process for the links between $d$ and $d'$

1. Choose a topic $z_d \sim \text{Multi}(\theta_d)$.
2. Choose a topic $z_{d'} \sim \text{Multi}(\theta_{d'})$.
3. If $z = z_d = z_{d'}$, choose $A_{dd'} \sim \text{Poi}(S_d S_{d'} \eta_z)$; otherwise, $A_{dd'} = 0$.

$\eta_z$ is the link density for topic $z$. The number of links $A_{dd'}$ between $d$ and $d'$ is drawn from a Poisson distribution

$$A_{dd'} \sim \text{Poi} \left( S_d S_{d'} \sum_{z=1}^{K} \theta_{dz} \theta_{d'z} \eta_z \right).$$
### Generative process for the links between $d$ and $d'$

1. Choose a topic $z_d \sim \text{Multi} (\theta_d)$.
2. Choose a topic $z_{d'} \sim \text{Multi} (\theta_{d'})$.
3. If $z = z_d = z_{d'}$, choose $A_{dd'} \sim \text{Poi} (S_d S_{d'} \eta_z)$; otherwise, $A_{dd'} = 0$.

$\eta_z$ is the link density for topic $z$. The number of links $A_{dd'}$ between $d$ and $d'$ is drawn from a Poisson distribution

$$A_{dd'} \sim \text{Poi} \left( S_d S_{d'} \sum_{z=1}^{K} \theta_{dz} \theta_{d'z} \eta_z \right).$$

### A similar model

*Mixed-membership stochastic block model (MMSB)* [Airoldi, Fienberg, Xing (2008)] uses the Bernoulli distribution. *MMSB* doesn’t scale well.
Graphical model

Poisson Mixed-Topic Link Model (\textbf{PMTLM})

\[
L_d \theta_d \quad Z_{dd'} \quad Z_{d'd} \quad \theta_{d'}
\]

\[
W_{dn} \quad Z_{dn} \quad A_{dd'} \quad W_{d'n}
\]

\[
\beta \quad \eta
\]
Graphical model

Poisson Mixed-Topic Link Model with Degree-Correction (PMTLM-DC)

\[ \begin{align*}
\theta_d & \rightarrow Z_{dd'} \\
\eta & \rightarrow Z_{dd'} \\
\eta & \rightarrow Z_{d'd} \\
\theta_{d'} & \rightarrow Z_{d'n} \\
S_d & \rightarrow S_{d'} \\
A_{dd'} & \rightarrow W_{dn} \\
\beta & \rightarrow W_{d'n} \\
L_d & \rightarrow L_{d'} 
\end{align*} \]
The log-likelihood of document $d$’s content is

$$
\mathcal{L}_d^{\text{content}} = \log P(w_{d1}, \ldots, w_{dL_d} | \theta_d, \beta)
= \sum_{w=1}^{W} C_{dw} \log \left( \sum_{z=1}^{K} \theta_{dz} \beta_{zw} \right).
$$

Here $C_{dw}$ is the number of times a word $w$ appears in document $d$. 
Likelihood functions

1. The log-likelihood of document $d$’s content is

$$
L^\text{content}_d = \log P(w_{d1}, \ldots, w_{dL_d} | \theta_d, \beta) \\
= \sum_{w=1}^{W} C_{dw} \log \left( \sum_{z=1}^{K} \theta_{dz} \beta_{zw} \right).
$$

Here $C_{dw}$ is the number of times a word $w$ appears in document $d$.

2. The log-likelihood for the links $A_{dd'}$ between $d$ and $d'$ is

$$
L^\text{links}_{dd'} = \log P(A_{dd'} | \theta_d, \theta_{d'}, \eta) \\
= A_{dd'} \log \left( \sum_{z} \theta_{dz} \theta_{d'z} \eta_z \right) - \sum_{z} \theta_{dz} \theta_{d'z} \eta_z.
$$
Likelihood functions

1. The log-likelihood of document $d'$'s content is

$$L_d^{\text{content}} = \log P(w_{d1}, \ldots, w_{dL_d} | \theta_d, \beta) = \sum_{w=1}^{W} C_{dw} \log \left( \sum_{z=1}^{K} \theta_{dz} \beta_{zw} \right).$$

Here $C_{dw}$ is the number of times a word $w$ appears in document $d$.

2. The log-likelihood for the links $A_{dd'}$ between $d$ and $d'$ is

$$L_{dd'}^{\text{links}} = \log P(A_{dd'} | \theta_d, \theta_{d'}, \eta) = A_{dd'} \log \left( \sum_z \theta_{dz} \theta_{d'z} \eta_z \right) - \sum_z \theta_{dz} \theta_{d'z} \eta_z.$$

3. The total likelihood for text and links is

$$\mathcal{L} = \mathcal{L}^{\text{content}} + \mathcal{L}^{\text{links}} = \sum_d L_d^{\text{content}} + \frac{1}{2} \sum_{dd'} L_{dd'}^{\text{links}}.$$
The algorithm is simple.

E-Step:

\[ h_{dw}(z) = \frac{\theta_{dz} \beta_{zw}}{\sum_{z'} \theta_{dz'} \beta_{z'w}} , \quad q_{dd'}(z) = \frac{\theta_{dz} \theta_{d'z} \eta_z}{\sum_{z'} \theta_{dz'} \theta_{d'z'} \eta_{z'}} . \]

M-Step:

\[ \beta_{zw} = \frac{\sum_d C_{dw} h_{dw}(z)}{\sum_d \sum_{w'} C_{dw'} h_{dw'}(z)} , \quad \eta_z = \frac{\sum_{dd'} A_{dd'} q_{dd'}(z)}{(\sum_d \theta_{dz})^2} , \]

\[ \theta_{dz} = \frac{\sum_w C_{dw} h_{dw}(z) + \sum_{d'} A_{dd'} q_{dd'}(z)}{L_d + \kappa_d} . \]

Here \( \kappa_d \) is the degree of document \( d \).
The algorithm is efficient, linear in the size of the data set when $K$ is a constant.

Size statistics:
1. $N$ is the number of documents in the data set.
2. $M$ is the number of links in the network.
3. $R = \sum_d R_d$, here $R_d$ is the number of distinct words in document $d$.

The time complexity for each EM iteration is: $O(K(N + M + R))$. 
The algorithm is efficient, linear in the size of the data set when $K$ is a constant.

Size statistics:
1. $N$ is the number of documents in the data set.
2. $M$ is the number of links in the network.
3. $R = \sum_d R_d$, here $R_d$ is the number of distinct words in document $d$.

The time complexity for each EM iteration is: $O(K(N + M + R))$.

How many EM iterations?
Real-world data sets

Real-world document citation networks\(^1\):

1. **Cora** and **Citeseer** contain papers in machine learning, with 7 topics for Cora and 6 topics for Citeseer. Doesn’t count the number of occurrences of the words in each document: \(C_{dw} = 0 \) or \(1\).

2. **PubMed** consists of medical research papers on 3 topics, namely three types of diabetes. Counts the number of occurrences of each word. \(C_{dw}\) is a non-negative integer.

<table>
<thead>
<tr>
<th></th>
<th>Cora</th>
<th>Citeseer</th>
<th>PubMed</th>
</tr>
</thead>
<tbody>
<tr>
<td>(K) – number of topics</td>
<td>7</td>
<td>6</td>
<td>3</td>
</tr>
<tr>
<td>(N) – number of documents</td>
<td>2,708</td>
<td>3,312</td>
<td>19,717</td>
</tr>
<tr>
<td>(M) – number of links</td>
<td>5,429</td>
<td>4,608</td>
<td>44,335</td>
</tr>
<tr>
<td>(R) – corpus size</td>
<td>49,216</td>
<td>105,165</td>
<td>1,333,397</td>
</tr>
</tbody>
</table>

\(^1\)These data sets are available for download at [http://www.cs.umd.edu/projects/linqs/projects/lbc/](http://www.cs.umd.edu/projects/linqs/projects/lbc/)
Convergence curves

1. Averaged log-likelihood over 100 runs, each run executes 5000 iterations.
2. Scaled curves start at 0 and achieve 1 at the end of 5000 iterations.
Running time on real-world data sets

1. Running times for our algorithms, PLSA, and PHITS-PLSA are given for one run of 5000 EM iterations.

2. Running times for the Relational Topic Model (RTM) consist of up to 500 EM iterations.

<table>
<thead>
<tr>
<th></th>
<th>Cora (sec)</th>
<th>Citeseer</th>
<th>PubMed</th>
</tr>
</thead>
<tbody>
<tr>
<td>EM (PLSA)</td>
<td>28</td>
<td>61</td>
<td>362</td>
</tr>
<tr>
<td>EM (PHITS-PLSA)</td>
<td>40</td>
<td>67</td>
<td>445</td>
</tr>
<tr>
<td>EM (RTM)</td>
<td>992</td>
<td>597</td>
<td>2,194</td>
</tr>
<tr>
<td>EM (PMTLM)</td>
<td>33</td>
<td>64</td>
<td>419</td>
</tr>
<tr>
<td>EM (PMTLM-DC)</td>
<td>36</td>
<td>64</td>
<td>402</td>
</tr>
</tbody>
</table>
Discrete labels and local search

- Mixtures of topics are more informative.
- Assign the most-likely topic $1 \leq z_d \leq K$ to each document $d$.
  $$z_d = \arg\max_z \theta_{dz}.$$

- Local search optimization algorithms:
  - Kernighan-Lin heuristic [?]
  - Markov chain Monte Carlo (MCMC)
Performance on document classification

Given two clusterings/labelings $C_1$ and $C_2$:

1. **Normalized Mutual Information (NMI)**
   
   $$\text{NMI}(C_1, C_2) = \frac{\text{MI}(C_1, C_2)}{\max(H(C_1), H(C_2))}.$$  

2. **Variation of Information (VI)**
   
   $$\text{VI}(C_1, C_2) = H(C_1) + H(C_2) - 2\text{MI}(C_1, C_2).$$

3. **Pairwise F-score (PWF)**
   
   $$\text{Precision} = \frac{\text{Number of pairs correctly predicted in same cluster}}{\text{Total number of pairs predicted in same cluster}},$$
   
   $$\text{Recall} = \frac{\text{Number of pairs correctly predicted in same cluster}}{\text{Total number of pairs actually in same cluster}},$$
   
   $$\text{F-score} = \frac{2 \times \text{Precision} \times \text{Recall}}{\text{Precision} + \text{Recall}}.$$
Performance on document classification

<table>
<thead>
<tr>
<th>Algorithm</th>
<th>Cora NMI</th>
<th>Cora VI</th>
<th>Cora PWF</th>
<th>Citeseer NMI</th>
<th>Citeseer VI</th>
<th>Citeseer PWF</th>
</tr>
</thead>
<tbody>
<tr>
<td>PHITS-PLSA</td>
<td>0.382</td>
<td>2.285</td>
<td>0.447</td>
<td>0.366</td>
<td>2.226</td>
<td>0.480</td>
</tr>
<tr>
<td>LINK-LDA</td>
<td>0.359†</td>
<td>—</td>
<td>0.397†</td>
<td>0.192†</td>
<td>—</td>
<td>0.305†</td>
</tr>
<tr>
<td>C-PLDC</td>
<td>0.489†</td>
<td>—</td>
<td>0.464†</td>
<td>0.276†</td>
<td>—</td>
<td>0.361†</td>
</tr>
<tr>
<td>RTM</td>
<td>0.349</td>
<td>2.306</td>
<td>0.422</td>
<td>0.369</td>
<td>2.209</td>
<td>0.480</td>
</tr>
<tr>
<td>PMTLM</td>
<td>0.417</td>
<td>2.152</td>
<td>0.463</td>
<td>0.393</td>
<td>2.129</td>
<td>0.507</td>
</tr>
<tr>
<td>PMTLM-DC</td>
<td>0.430</td>
<td>2.100</td>
<td>0.475</td>
<td>0.387</td>
<td>2.152</td>
<td>0.503</td>
</tr>
</tbody>
</table>
### Performance on document classification

#### Pubmed

<table>
<thead>
<tr>
<th>Algorithm</th>
<th>NMI</th>
<th>VI</th>
<th>PWF</th>
</tr>
</thead>
<tbody>
<tr>
<td>PHITS-PLSA</td>
<td>0.233</td>
<td>1.633</td>
<td>0.486</td>
</tr>
<tr>
<td>RTM</td>
<td>0.228</td>
<td>1.646</td>
<td>0.482</td>
</tr>
<tr>
<td>PMTLM</td>
<td>0.229</td>
<td>1.644</td>
<td>0.483</td>
</tr>
<tr>
<td>PMTLM-DC</td>
<td>0.233</td>
<td>1.635</td>
<td>0.485</td>
</tr>
</tbody>
</table>

The likelihood for the content is much lower than the likelihood for the links. All the algorithms focus only on the content.

---

YZ h u , XY a n , LG e t o o r , CM o o r e

Scalable Text and Link Analysis with Mixed-Topic Link Models
Performance on document classification

<table>
<thead>
<tr>
<th>Algorithm</th>
<th>NMI</th>
<th>VI</th>
<th>PWF</th>
</tr>
</thead>
<tbody>
<tr>
<td>PHITS-PLSA</td>
<td>0.233</td>
<td>1.633</td>
<td>0.486</td>
</tr>
<tr>
<td>RTM</td>
<td>0.228</td>
<td>1.646</td>
<td>0.482</td>
</tr>
<tr>
<td>PMTLM</td>
<td>0.229</td>
<td>1.644</td>
<td>0.483</td>
</tr>
<tr>
<td>PMTLM-DC</td>
<td>0.233</td>
<td>1.635</td>
<td>0.485</td>
</tr>
</tbody>
</table>

- The likelihood for the content is much lower than the likelihood for the links.
- All the algorithms focus only on the content.
Balancing content and links

- Content normalization

\[ \mathcal{L}_{\text{content}} = \sum_d \left( \frac{1}{L_d} \right) \mathcal{L}_d^{\text{content}}. \]

- Linear interpolation

\[ \mathcal{L} = \alpha \mathcal{L}_{\text{content}} + (1 - \alpha) \mathcal{L}^{\text{links}}. \]

**Content normalization**
- documents of different sizes are equally important
- increases the likelihood for the content

**Linear interpolation**
- lets us go from only caring about links to only caring about content
- the optimal value of the relative weight \( \alpha \) for text vs. links might depend on the data set and the task
Performance on document classification

Cora

\[
\begin{array}{c}
\text{NMI} \\
\alpha
\end{array}
\]

PubMed

\[
\begin{array}{c}
\text{NMI} \\
\alpha
\end{array}
\]

\*

Citeseeer

\[
\begin{array}{c}
\text{NMI} \\
\alpha
\end{array}
\]
Performance on link prediction

Cora

AUC

0.7 0.8 0.9 1

0.8 0.9 1

RTM

0 0.2 0.4 0.6 0.8 1

α

PMTLM−DC

PMTLM

Citeseer

AUC

0.5 0.6 0.7 0.8 0.9 1

0.85 0.9 0.95 1

RTM

0 0.2 0.4 0.6 0.8 1

α

PMTLM−DC

PMTLM

PubMed

AUC

0.5 0.6 0.7 0.8 0.9 1

0.8 0.9 1

RTM

0 0.2 0.4 0.6 0.8 1

α

PMTLM−DC

PMTLM

YZ h u , X Yan, L Getoor, C Moore

Scalable Text and Link Analysis with Mixed-Topic Link Models
Conclusions

1. We combine the PLSA model for text, and BKN model for links
   1. naturally compatible with each other
   2. simple and efficient inference algorithm with high accuracy

2. We balance information from the text and the links
   1. the optimal value of the relative weight $\alpha$ for text vs. links depends on both data set and learning task
   2. balancing the text and link is really important