Latent factor models for social network data

Peter Hoff and Mike Ward

Statistics, Political Science and the CSSS
University of Washington
Outline

Latent factor models

Dataset 1: Static network

Dataset 2: Dynamic multi-way network

Dataset 3: Moderate size network

Dataset 4: Very large network

Discussion

Take Away
Relational data

**Relational data:** consist of

- a set of units or nodes $A$, and
- a set of measurements $Y \equiv \{y_{i,j}\}$ specific to pairs of nodes $(i,j) \in A \times A$.

**Examples:**

**International relations**
- $A =$countries,
- $y_{i,j} =$ indicator of a dispute initiated by $i$ with target $j$.

**Needle-sharing network**
- $A =$IV drug users,
- $y_{i,j} =$ needle-sharing activity between $i$ and $j$.

**Protein-protein interactions**
- $A =$proteins,
- $y_{i,j} =$ the interaction between $i$ and $j$.

**Business locations**
- $A_1 =$banks, $A_2 =$cities,
- $y_{i,j} =$ presence of an office of bank $i$ in city $j$. 
Inferential goals

\( y_{i,j} \) measures \( i \rightarrow j \), \( x_{i,j} \) is a vector of explanatory variables.

\[
Y = \begin{pmatrix}
y_{1,1} & y_{1,2} & y_{1,3} & \text{NA} & y_{1,5} & \cdots \\
y_{2,1} & y_{2,2} & y_{2,3} & y_{2,4} & y_{2,5} & \cdots \\
y_{3,1} & \text{NA} & y_{3,3} & y_{3,4} & \text{NA} & \cdots \\
y_{4,1} & y_{4,2} & y_{4,3} & y_{4,4} & y_{4,5} & \cdots \\
: & : & : & : & : & \cdots \\
\end{pmatrix}
\]

\[
X = \begin{pmatrix}
x_{1,1} & x_{1,2} & x_{1,3} & x_{1,4} & x_{1,5} & \cdots \\
x_{2,1} & x_{2,2} & x_{2,3} & x_{2,4} & x_{2,5} & \cdots \\
x_{3,1} & x_{3,2} & x_{3,3} & x_{3,4} & x_{3,5} & \cdots \\
x_{4,1} & x_{4,2} & x_{4,3} & x_{4,4} & x_{4,5} & \cdots \\
: & : & : & : & : & \cdots \\
\end{pmatrix}
\]

Consider a basic (generalized) linear model

\[ y_{i,j} \sim \beta' x_{i,j} + \epsilon_{i,j} \]

A model can provide

- a measure of the association between \( X \) and \( Y \): \( \hat{\beta}, \text{se}(\hat{\beta}) \)
- predictions of missing or future observations: \( p(y_{1,4}|Y, X) \)
Node heterogeneity
Latent variable models

Deviations from ordinary regression models can be represented as

\[ y_{i,j} \sim \beta' x_{i,j} + z_{i,j} \]

A simple “latent variable” model might include row and column effects:

\[ z_{i,j} = u_i + v_j + \epsilon_{i,j} \quad \Rightarrow \quad y_{i,j} \sim \beta' x_{i,j} + u_i + v_j + \epsilon_{i,j} \]

\( u_i \) and \( v_j \) induce across-node heterogeneity that is additive on the scale of the regressors. Inclusion of these effects in the model can dramatically improve

▶ within-sample model fit (measured by \( R^2 \), likelihood ratio, BIC, etc.);
▶ out-of-sample predictive performance (measured by cross-validation).

But this model only captures heterogeneity of outdegree/indegree, and can’t represent more complicated structure, such as clustering, transitivity, etc.
Multiplicative effects

\[ y_{i,j} \sim \beta' x_{i,j} + u_i' Dv_j + \epsilon_{i,j} \]

**Interpretation:**
Think of \( \{u_1, \ldots, u_m\}, \{v_1, \ldots, v_n\} \) as vectors of **latent nodal attributes**:

\[
u_i' Dv_j = \sum_{k=1}^{K} d_k u_{i,k} v_{j,k}
\]

In general, a latent variable model relating \( X \) to \( Y \) is

\[
\begin{align*}
g(E[y_{i,j}]) &= \beta' x_{i,j} + u_i' Dv_j + \epsilon_{i,j} \\
\end{align*}
\]

For example, some potential models are

- If \( y_{i,j} \) is binary, \[
\log \text{odds } (y_{i,j} = 1) = \beta' x_{i,j} + u_i' Dv_j + \epsilon_{i,j}
\]
- If \( y_{i,j} \) is count data, \[
\log E[y_{i,j}] = \beta' x_{i,j} + u_i' Dv_j + \epsilon_{i,j}
\]
- If \( y_{i,j} \) is continuous, \[
E[y_{i,j}] = \beta' x_{i,j} + u_i' Dv_j + \epsilon_{i,j}
\]

**Estimation:** Given \( D, V \), the predictor is linear in \( U \). This bilinear structure can be exploited (EM, Gibbs sampling, variational methods).
Analyzing undirected data

In many applications $y_{i,j} = y_{j,i}$ by design, and so

$$(y_{i,j} = y_{j,i}) \sim \beta'x_{i,j} + z_{i,j},$$

and $Z = \{z_{i,j}\}$ is a symmetric array. How should $Z$ be modeled?

**Modeling via matrix decomposition:** Write $Z = M + E$, with all matrices symmetric. All such $M$ have an eigenvalue decomposition

$$M = U\Lambda U'$$

$$m_{i,j} = u'_i\Lambda u_j$$

This suggests a model of the form

$$(y_{i,j} = y_{j,i}) \sim \beta'x_{i,j} + u'_i\Lambda u_j + \epsilon_{i,j}$$
Dataset 1
Dataset 1
Model-based link prediction

1. Estimate the model parameters \( \{\mu, \mathbf{u}_1, \ldots, \mathbf{u}_n, \mathbf{v}_1, \ldots, \mathbf{v}_n\} \) with the available data \( \{y_{i,j} : y_{i,j} \neq \text{NA}\} \);

2. Using the parameter estimates, obtain a predictive probability of a link for each missing data point: \( \tilde{p}_{i,j} = \Phi(\mu + \mathbf{u}_i^T \mathbf{v}_j) \) for each \((i, j)\) such that \( y_{i,j} = \text{NA} \);

3. Check the pairs with missing data for links, starting with the ones with the highest predictive probabilities.
1. Randomly divide the set of \( n \times (n - 1) \) ordered pairs of indices \((i, j)\) into four subsets of indices, say \( S_1, S_2, S_3, S_4 \).

2. For each \( S_k, \; k = 1 \ldots, 4; \)
   2.1 Estimate the model parameters using the data \( Y_{-k} = \{ y_{i,j} : (i,j) \notin S_k \} \) and treating the data with indices in \( S_k \) as missing.
   2.2 Based on these parameter estimates, obtain the predictive probability of a link \( \tilde{p}_{i,j} \) for each \((i,j) \in S_k \).

3. Compare \( \tilde{p}_{i,j} \) to \( y_{i,j} \) for each pair \((i,j)\).
Dataset 1

- Number of pairs checked
- Number of links found
- Perfect selection
- 2-d factor model
- Random selection

Graph showing the number of links found against the number of pairs checked for different selection methods.
Dataset 2: Affiliation network

**Affiliation network:** interactions occur between two different types of nodes:

\[ y_{i,j} = \begin{cases} 
1 & \text{if } i \text{ affiliated with } j \\
0 & \text{if } i \text{ not affiliated with } j 
\end{cases} \]

- \( i \) indexes actors
- \( j \) indexes documents

**Model:** as before,

\[ y_{i,j} \sim \mu + u_i'v_j \]

- \( u_i \) are actor-specific latent factors
- \( v_j \) are document-specific latent factors
Dataset 2
## Dataset 2

<table>
<thead>
<tr>
<th>$t$</th>
<th>links at time $t$</th>
<th>persistent links</th>
<th>random pred</th>
<th>empirical pred</th>
<th>model pred</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>8419</td>
<td>108</td>
<td>12.8</td>
<td>55</td>
<td>75</td>
</tr>
<tr>
<td>2</td>
<td>7338</td>
<td>161</td>
<td>21.9</td>
<td>61</td>
<td>74</td>
</tr>
<tr>
<td>3</td>
<td>13029</td>
<td>199</td>
<td>15.3</td>
<td>78</td>
<td>91</td>
</tr>
<tr>
<td>4</td>
<td>9324</td>
<td>293</td>
<td>31.4</td>
<td>133</td>
<td>132</td>
</tr>
<tr>
<td>5</td>
<td>24444</td>
<td>399</td>
<td>16.3</td>
<td>136</td>
<td>152</td>
</tr>
<tr>
<td>6</td>
<td>16942</td>
<td>438</td>
<td>25.9</td>
<td>129</td>
<td>188</td>
</tr>
<tr>
<td>7</td>
<td>13265</td>
<td>39</td>
<td>2.9</td>
<td>23</td>
<td>26</td>
</tr>
</tbody>
</table>
Multi-way data

This dataset is more properly called *multi-way data*:

\[ y_{i,j,k} \sim \text{relationship between actors } i \text{ and } j \text{ in document } k \]

The two-way latent factor model extends naturally to multi-way data:

\[
y_{i,j} \sim \mu + u_i^\prime v_j \quad \text{(two-way data)}
\]

\[
= \mu + \sum_{r=1}^{R} u_{i,r} v_{i,r}
\]

\[
y_{i,j,k} \sim \mu + \sum_{r=1}^{R} u_{i,r} v_{j,r} w_{k,r} \quad \text{(three-way data)}
\]

Including time, this data in fact could be considered four-way data:

\[ y_{i,j,k,t} = \text{relationship between } i \text{ and } j \text{ in document } k \text{ at time } t \]

(although there are few documents in more than one time point)
Multiple Views of Network of 428 Actors, with covariates
Plain, Sex, Age, Country
Multiple Views of Network of 428 Actors, with covariates
Plain, Sex, Age, Country
Multiple Views of Network of 428 Actors, with covariates
Plain, Sex, Age, Country
Multiple Views of Network of 428 Actors, with covariates
Plain, Sex, Age, Country
Cross-Validation on 428 node network

3 Dimensions >> 2 Dimensions

Data Set 3:

<table>
<thead>
<tr>
<th>N Checked</th>
<th>N Correct</th>
</tr>
</thead>
<tbody>
<tr>
<td>k=2</td>
<td>133</td>
</tr>
<tr>
<td>k=3</td>
<td></td>
</tr>
</tbody>
</table>

Graph showing N Correct vs N Checked for k=2 and k=3.
Multiple Views of Network of 428 Actors

More informative than standard

goal is dynamic visualization of latent positions

Power Nodes: 217 53 21 181 132 110 15 17 18 119 219 222 274
Multiple Views of Network of 428 Actors

goal is dynamic visualization of latent positions

Dynamic, 3-d with Color Draping
Multiple Views of Network of 428 Actors

Ring of Fire, especially for asymmetric networks

goal is dynamic visualization of latent positions
Multiple Views of Network of 428 Actors

goal is dynamic visualization of latent positions

Ben Fry’s Animations at http://acg.media.mit.edu/people/fry/valence/movie.html seems like the goal.
Very Large $\implies$ Viewable

Dataset 4 is huge and curious and almost symmetric i.e. NOT
Likelihood-based statistical inference for a set of network parameters $\theta$ from the observed data can be obtained by

$$\Pr(\{y_{i,j} : o_{i,j} = 1\}, O|\theta) = \int_{\{y_{i,j} : o_{i,j} = 0\}} \Pr(Y|\theta)Pr(O|Y, \theta) \prod_{(i,j): o_{i,j} = 0} dy_{i,j}. \Pr(O|Y, \theta)$$

represents the sampling scheme of the data, indicating the probability that each link was measured.
This means that Missing Data is not Equivalent to the absence of a link, which seems obvious, but many analyses confuse this point. Quantifying the data collection scheme provides an estimate of the probability that each potential link was actually measured for the presence of a link is important. It can enable likelihood inference.

- Sampling and Data Collection
- Missing Data is not equivalent to a Missing Link
- Many Analyses Confuse This Point
- Quantifying the Data Collection Scheme
Main Points

Hard Problem

▶ General Bilinear Mixed Effects Model addresses dependencies
▶ Estimates latent positions of nodes and probabilities of links between them
▶ Is a considerable improvement in out of sample predictions over most other approaches
▶ Reduces dimensions and has links to visualizations
▶ Plausibly scalable and real time, but huge matrices—whether sparse or not—are huge challenges