Modeling Large-Scale Networks Based on Mobility Data

Tony Jebara

COLUMBIA UNIVERSITY

What can we learn about people from their social-mobile data exhaust?
Mobile data

Sense Networks, 2006-

Mobile data from apps & call detail records

100m users, each with 20 events/day:

- calls & sms
- location
- app data
A network model of places

Look at flow A to B

Markov transition

Apply Minimum Volume Embedding* on graph

Color code clusters in graph

A network model of places

Look at flow A to B

Markov transition

Apply Minimum Volume Embedding*
on graph

Color code clusters in graph

Location patterns of users

<table>
<thead>
<tr>
<th>Week</th>
<th>FLO 1</th>
<th>FLO 2</th>
<th>...</th>
<th>FLO 20</th>
<th>COM 1</th>
<th>COM 2</th>
<th>...</th>
<th>COM 97</th>
<th>DEM 1</th>
<th>DEM 2</th>
<th>...</th>
<th>DEM 78</th>
<th>SMS</th>
<th>CAL</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>.03</td>
<td>.31</td>
<td>...</td>
<td>.14</td>
<td>.03</td>
<td>.05</td>
<td>...</td>
<td>.41</td>
<td>.11</td>
<td>.04</td>
<td>...</td>
<td>.01</td>
<td>.04</td>
<td>.11</td>
</tr>
<tr>
<td>2</td>
<td>.14</td>
<td>.34</td>
<td>...</td>
<td>.02</td>
<td>.04</td>
<td>.05</td>
<td>...</td>
<td>.52</td>
<td>.01</td>
<td>.01</td>
<td>...</td>
<td>.00</td>
<td>.05</td>
<td>.12</td>
</tr>
<tr>
<td>...</td>
<td>...</td>
<td>...</td>
<td>...</td>
<td>...</td>
<td>...</td>
<td>...</td>
<td>...</td>
<td>...</td>
<td>...</td>
<td>...</td>
<td>...</td>
<td>...</td>
<td>...</td>
<td>...</td>
</tr>
<tr>
<td>168</td>
<td>.07</td>
<td>.34</td>
<td>.51</td>
<td>.02</td>
<td>.06</td>
<td>.48</td>
<td>.02</td>
<td>.01</td>
<td>.00</td>
<td>.08</td>
<td>.08</td>
<td>.01</td>
<td>.08</td>
<td>.08</td>
</tr>
</tbody>
</table>
Location patterns of users

<table>
<thead>
<tr>
<th>Week Hour</th>
<th>FLO 1</th>
<th>FLO 2</th>
<th>...</th>
<th>FLO 20</th>
<th>COM 1</th>
<th>COM 2</th>
<th>...</th>
<th>COM 97</th>
<th>DEM 1</th>
<th>DEM 2</th>
<th>...</th>
<th>DEM 78</th>
<th>SMS</th>
<th>CAL</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>.03</td>
<td>.31</td>
<td>.14</td>
<td>.03</td>
<td>.05</td>
<td>.41</td>
<td>.11</td>
<td>.04</td>
<td>.01</td>
<td>.04</td>
<td>.11</td>
<td>.04</td>
<td>.07</td>
<td>.34</td>
</tr>
<tr>
<td>2</td>
<td>.14</td>
<td>.34</td>
<td>.02</td>
<td>.04</td>
<td>.05</td>
<td>.52</td>
<td>.01</td>
<td>.01</td>
<td>.00</td>
<td>.05</td>
<td>.12</td>
<td>.05</td>
<td>.08</td>
<td>.08</td>
</tr>
<tr>
<td>...</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>168</td>
<td>.07</td>
<td>.34</td>
<td>.51</td>
<td>.02</td>
<td>.06</td>
<td>.48</td>
<td>.02</td>
<td>.01</td>
<td>.00</td>
<td>.08</td>
<td>.08</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
Location patterns of users

- Location patterns of 9 users
- X-axis is frequency at different places
- Y-axis is hour of week

- Each location pattern is vectorized into an x vector
- Get Euclidean distances between users

\[
D_{ij} = \| \vec{x}_i - \vec{x}_j \|^2 \\
\text{where } D \in \mathbb{R}^{n \times n}
\]
Linking a network of users

Use location patterns or *co-location* to link people
Find sparse network of users that minimizes distances*

*B. Huang and T. Jebara. Loopy Belief Propagation for Bipartite Maximum Weight b-Matching. Artificial Intelligence and Statistics (AISTATS), March 2007.*
Labeling the network of users

Use contagion on network to propagate behavior labels*

Labeling the network of users

- From sparse (1k-5k) training data of surveyed labels on users
- Machine learned label prediction model
- Evaluated on held out ground truth dataset (25% holdout)
- Obtain 4x to 11x improvement over random baseline
- Label parent, single, gender, student, age, income, travels, zip...
Labeling the network of users

- Lift Curves for Parent, Traveler and Single Predictions

User is a Parent
Up to 9x lift over random

User is a Business Traveler
Up to 8x lift over random

User is Single
Up to 5x lift over random
Better linking & link prediction

- Did we build a good network? How to improve it?
- Is there something better than Euclidean distance?

- Maybe some co-locations matter more than others...
- If you co-locate with someone at home, you are more likely to know them, call them, and be like them... than if you co-locate at the subway station
Better linking & link prediction

- Many link prediction heuristics exist
  (Adamic Adar 03) (Nowell Kleinberg 03)

1) link nodes with many common neighbors
2) weight common neighbors by their degrees
3) weight short paths exponentially more than long paths
Better linking & link prediction

• But, how to add edges to a fully disconnected network??

“Link prediction is a problem of estimating distances between pairs of nodes. Nodes lie at unknown positions in some latent space and the observed presence or absence of links between nodes provides clues about their distances” (Sarkar Chakrabati Moore 11)

• So... do distance-metric learning on node attributes
• We call this the “Freshmen problem”
The freshmen problem

• New students show up to school, nobody knows anyone
• Have a matrix of their profile vectors: \( X = [\vec{x}_1, \ldots, \vec{x}_n] \in \mathbb{R}^{d \times n} \)
The freshmen problem

- New students show up to school, nobody knows anyone
- Have a matrix of their profile vectors: $X = \left[\vec{x}_1, \ldots, \vec{x}_n\right] \in \mathbb{R}^{d \times n}$
- At graduation, observe formed network

\[
X = \begin{bmatrix}
\vec{x}_1, & \ldots, & \vec{x}_n
\end{bmatrix} \in \mathbb{R}^{d \times n}
\]
The freshmen problem

- New students show up to school, nobody knows anyone
- Have a matrix of their profile vectors: \( X = \begin{bmatrix} \vec{x}_1, \ldots, \vec{x}_n \end{bmatrix} \in \mathbb{R}^{d \times n} \)
- At graduation, observe formed network
- Predict the network for next year’s freshmen?
The freshmen problem

- Example real data sources with both X and A information

- Facebook: X features are user profiles
  A edges from friendships

- Wikipedia: X features are page word-counts
  A edges from hyperlinks

- Foursquare: X features are places user visits
  A edges from friendships

- Telco: X features are places user visits
  A edges from calling network
The freshmen problem

1) Learn by observing an \( X \) and observing its \( A \)

\[
X = \begin{bmatrix}
4'8'' & 108lbs \\
5'8'' & 119lbs \\
6'1'' & 184lbs \\
6'2'' & 178lbs \\
5'9'' & 209lbs \\
\end{bmatrix}
\]

\[
A = \begin{bmatrix}
0 & 1 & 1 & 0 & 0 \\
1 & 0 & 0 & 1 & 0 \\
1 & 0 & 0 & 1 & 1 \\
0 & 1 & 1 & 0 & 1 \\
0 & 0 & 1 & 1 & 0 \\
\end{bmatrix}
\]

2) Then, given a new \( X' \) we must predict its \( A' \)

\[
X' = \begin{bmatrix}
4'9'' & 178lbs \\
5'4'' & 159lbs \\
6'4'' & 224lbs \\
6'3'' & 198lbs \\
5'4'' & 149lbs \\
\end{bmatrix}
\]

\[
A' = \begin{bmatrix}
? \\
\end{bmatrix}
\]
The freshmen problem

- We will learn a Mahalanobis distance via the $M$ matrix

\[ D_{ij} = \left\| \vec{x}_i - \vec{x}_j \right\|_M^2 = \left( \vec{x}_i - \vec{x}_j \right)^T M \left( \vec{x}_i - \vec{x}_j \right) \]

- Examples of what we could learn:
  - Not all attributes are created equal
  - Height difference matters more than weight
  - Age matters less than weight + height
  - Height matters but only when people are 6 feet or taller
  - Co-locating at a church matters more than the airport
  - Etc...
Structure-Preserving Metric Learning

- Structure-Preserving Metric Learning (SPML) (Shaw Huang Jebara 11)
- Finds optimal Mahalanobis distance $M$ from $X$ & $A$
- The connectivity algorithm then reconstructs $A$ perfectly
Structure-Preserving Metric Learning

• Given A and X, SPML finds Mahalanobis distance metric M

• The optimal M minimizes the following cost

\[
\min_M \lambda \|M\|^2 - \frac{1}{|T|} \sum_{(i,j,k) \in T} \max \left( \|\bar{x}_i - \bar{x}_j\|_M^2 - \|\bar{x}_i - \bar{x}_k\|_M^2 + 1, 0 \right)
\]

where \( T = \left\{ (i, j, k) \middle| A_{ij} = 1, A_{ik} = 0 \right\} \)

• Considers triplets of nodes and repels non-neighbors while attracting neighbors

• We find the optimum using stochastic sub-gradient descent
Structure-Preserving Metric Learning

- Convergence of stochastic descent on FaceBook dataset
Structure-Preserving Metric Learning

- ROC curve for edge predictions A' on Wikipedia dataset
Structure-Preserving Metric Learning

• SPML predicts network $A'$ with better AUC

<table>
<thead>
<tr>
<th></th>
<th>$n$</th>
<th>$m$</th>
<th>$d$</th>
<th>Euclidean</th>
<th>RTM</th>
<th>SVM</th>
<th>SPML</th>
</tr>
</thead>
<tbody>
<tr>
<td>Graph Theory</td>
<td>223</td>
<td>917</td>
<td>6695</td>
<td>0.624</td>
<td>0.591</td>
<td>0.610</td>
<td>0.722</td>
</tr>
<tr>
<td>Philosophy Concepts</td>
<td>303</td>
<td>921</td>
<td>6695</td>
<td>0.705</td>
<td>0.571</td>
<td>0.708</td>
<td>0.707</td>
</tr>
<tr>
<td>Search Engines</td>
<td>269</td>
<td>332</td>
<td>6695</td>
<td>0.662</td>
<td>0.487</td>
<td>0.611</td>
<td>0.742</td>
</tr>
<tr>
<td>Philosophy Crawl</td>
<td>100,000</td>
<td>4,489,166</td>
<td>7702</td>
<td>0.547</td>
<td>–</td>
<td>–</td>
<td>0.601</td>
</tr>
<tr>
<td>Harvard</td>
<td>1937</td>
<td>48,980</td>
<td>193</td>
<td>0.764</td>
<td>0.562</td>
<td>0.839</td>
<td>0.854</td>
</tr>
<tr>
<td>MIT</td>
<td>2128</td>
<td>95,322</td>
<td>173</td>
<td>0.702</td>
<td>0.494</td>
<td>0.784</td>
<td>0.801</td>
</tr>
<tr>
<td>Stanford</td>
<td>3014</td>
<td>147,516</td>
<td>270</td>
<td>0.718</td>
<td>0.532</td>
<td>0.784</td>
<td>0.808</td>
</tr>
<tr>
<td>Columbia</td>
<td>3050</td>
<td>118,838</td>
<td>251</td>
<td>0.717</td>
<td>0.519</td>
<td>0.796</td>
<td>0.818</td>
</tr>
<tr>
<td>Foursquare</td>
<td>83</td>
<td>4322</td>
<td>24082</td>
<td>0.760</td>
<td>0.501</td>
<td>0.710</td>
<td>0.829</td>
</tr>
</tbody>
</table>
Caveat!!!

• SPML needs to know # of neighbors per node a priori
• To build connectivity $A'$, need to know degree of each node

• But we don’t have an oracle to give us the degrees of $A'$ 😞
• How can we predict degrees of each node from $X'$ data?

• SPML just learned $M$ the Mahalanobis metric
• Let’s also learn a mapping $S$ to predict degrees 😊
Degree-Distributional Metric Learning

• Degree-Distributional Metric Learning (DDML) (Huang Shaw Jebara 11)
  1) Given X and A
  2) Learn metric M and degree-predictor mapping S
  3) Given X’ fully predict A’

• Instead of predicting node degrees deterministically (degree=2 with 100% probability)
• Use generalized matching to predict degree probabilistically (degree=0,1,2,... with probabilities summing to 1)
Degree-Distributional Metric Learning

- Minimize distances with degree distribution for each node
- Also known as a generalized matching problem
- Solvable via reduction to $b$-matching (Huang Jebara 09)
Degree-Distributional Metric Learning

- Receiver-Operator-Characteristic for Wikipedia dataset
Degree-Distributinal Metric Learning

• DDML predicts network A' with better AUC

• Beats SPML while also fully predicting node degrees!

|                          | n  | |E|  | d  | Euclid. | RTM  | SVM  | SPML | DDML  |
|--------------------------|----|----|----|-----|--------|------|------|------|-------|
| Graph Theory             | 223| 917| 6695|     | 0.624  | 0.591| 0.610| 0.722| 0.691*|
| Philosophy Concepts      | 303| 921| 6695|     | 0.705  | 0.571| 0.708| 0.707| 0.746*|
| Search Engines           | 269| 332| 6695|     | 0.662  | 0.487| 0.611| 0.742| 0.725*|
| Philosophy Crawl         | 100k| 4m | 7702|     | 0.547  | –    | –    | 0.601| 0.562 |
| Harvard                  | 1937| 48k| 193 |     | 0.764  | 0.562| 0.839| 0.854| 0.848 |
| MIT                      | 2128| 95k| 173 |     | 0.702  | 0.494| 0.784| 0.801| 0.797 |
| Stanford                 | 3014| 147k| 270 |     | 0.718  | 0.532| 0.784| 0.808| 0.810 |
| Columbia                 | 3050| 118k| 251 |     | 0.717  | 0.519| 0.796| 0.818| 0.821 |
Results

- Metrics recovered from FaceBook dataset

- FourSquare dataset: too small to interpret

- Telco dataset: results to appear soon, finds metric on places that lead to communication...
Mobile/Social

Mobile & Social data

What behaviors, attributes & colocations predict social link formation

What social links predict attributes, behavior and behavior contagion
References


