Social Prediction: Can we predict users’ action and emotions?

Jie Tang

Knowledge Engineering Group
Department of Computer Science and Technology
Tsinghua University
Email: jietang@tsinghua.edu.cn
Motivation

Social behavior

VS.

Emotion change
Motivation: A Happy System

Can we predict users’ activities and emotion?
Social Action Prediction

SIGKDD’10
What can we do in SNS?

- Facebook
  
  ![Facebook Interface](facebook_image)

- Flickr
  
  ![Flickr Interface](flickr_image)

- Twitter
  
  ![Twitter Interface](twitter_image)
Social Action

Twitter

Add favorites

Flickr

KDD

Tweet on “Haiti Earthquake”

Publish in KDD Conference
User Action in Social Networks

Questions:
- What factors influence you to add a photo into your favorite list?
- If you post a tweet on “Haiti Earthquake”, will your friends retweet it or reply?

Challenge:
- How to track and model users’ actions?
- How to predict users’ actions over time?
Social Action Modeling and Prediction

Action Prediction:
Will John post a tweet on “Haiti Earthquake”?

Personal attributes:
1. Always watch news
2. Enjoy sports
3. ....
Statistical Study: Influence

Y-axis: the likelihood that the user also performs the action at $t$

X-axis: the percentage of one’s friends who perform an action at $t - 1$
Statistical Study: Dependence

Y-axis: the likelihood that a user performs an action

X-axis: different time windows
Statistical Study: Correlation

Y-axis: the likelihood that two friends (random) perform an action together

X-axis: different time windows
Problem formulation

Input:
\[ G^t = (V^t, E^t, X^t, Y^t) \]
\[ t = 1, 2, \ldots, T \]

Output:
\[ F: f(G^t) \rightarrow Y^t \]
NTT-FGM Model

Influence

time 1
g(z₁, z₂)

time 2
h₁₂(z₁, z₂)

Dependence

g(z₁, z₅)

Action

Continuous latent action state

Correlation

time 3
h₁₃

Personal attributes

Dependence

y₁

Personal attributes

Action

x₁
How to estimate the parameters?

\[
p(Y|G) = \frac{1}{Z} \exp \left\{ \sum_{t=1}^{T} \sum_{i=1}^{N} \frac{(y_{ti} - z_{ti})^2}{2\sigma^2} + \sum_{t=1}^{T} \sum_{i=1}^{N} \sum_{j=1}^{N} \lambda_{ij} m_{ji}^{t-1} g(z_i^t, z_j^{t-1}) + \sum_{t=1}^{T} \sum_{i=1}^{N} \sum_{j=1}^{N} \sum_{k=1}^{d} \beta_{ij} m_{ij}^t h_{ij}(z_i^t, z_j^t) \right\}
\]

\[
g_{ji}(z_i^t, z_j^{t-1}) = -(z_i^t - z_j^{t-1})^2
\]

\[
h_{ij}(z_i^t, z_j^t) = -(z_i^t - z_j^t)^2
\]

\[
h_{k}(z_i^t, x_{ik}^t) = -(z_i^t - x_{ik}^t)^2
\]
Model Learning—Two-step learning

Input: number of iterations $I$ and learning rate $\eta$;

Output:

Initial
Initial
repeat

for $t = 1$ to $I$ do

Compute gradient $\nabla_{\log \alpha_k}$, $\nabla_{\log \beta_{ij}}$, $\nabla_{\log \lambda_{ij}}$;

Update $\log \alpha_k = \log \alpha_k + \eta \times \nabla_{\log \alpha_k}$;

Update $\log \beta_{ij} = \log \beta_{ij} + \eta \times \nabla_{\log \beta_{ij}}$;

Update $\log \lambda_{ij} = \log \lambda_{ij} + \eta \times \nabla_{\log \lambda_{ij}}$;

end

M Step: % fix $\alpha$, $\beta$, $\lambda$ learn $z$;

Solve the following linear equation:

$$(A + I)z = y + X\alpha$$

until convergence;

Extremely time costing!!

Our solution: distributed learning (MPI)
Experiment

- **Data Set** ([http://arnetminer.org/stnt](http://arnetminer.org/stnt))

<table>
<thead>
<tr>
<th>Action</th>
<th>Nodes</th>
<th>#Edges</th>
<th>Action Stats</th>
</tr>
</thead>
<tbody>
<tr>
<td>Twitter</td>
<td>7,521</td>
<td>304,275</td>
<td>730,568</td>
</tr>
<tr>
<td>Post tweets on “Haiti Earthquake”</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Flickr</td>
<td>8,721</td>
<td>485,253</td>
<td>485,253</td>
</tr>
<tr>
<td>Add photos into favorite list</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Arnetminer</td>
<td>2,062</td>
<td>34,986</td>
<td>2,960</td>
</tr>
<tr>
<td>Issue publications on KDD</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

- **Baseline**
  - SVM
  - wvRN (Macskassy, 2003)

- **Evaluation Measure:**
  Precision, Recall, F1-Measure
## Performance Analysis

<table>
<thead>
<tr>
<th>Data set</th>
<th>Method</th>
<th>Recall</th>
<th>Precision</th>
<th>F1-Measure</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Twitter</strong></td>
<td>SVM</td>
<td>10.41</td>
<td>16.71</td>
<td>13.85</td>
</tr>
<tr>
<td></td>
<td>wvRN</td>
<td>0.45</td>
<td>7.89</td>
<td>0.86</td>
</tr>
<tr>
<td></td>
<td>NTT-FGM</td>
<td>26.40</td>
<td>21.14</td>
<td><strong>23.47</strong></td>
</tr>
<tr>
<td><strong>Flickr</strong></td>
<td>SVM</td>
<td>34.48</td>
<td>45.05</td>
<td>39.06</td>
</tr>
<tr>
<td></td>
<td>wvRN</td>
<td>60.02</td>
<td>48.81</td>
<td><strong>53.84</strong></td>
</tr>
<tr>
<td></td>
<td>NTT-FGM</td>
<td>56.18</td>
<td>45.80</td>
<td>50.47</td>
</tr>
<tr>
<td><strong>ArnetMiner</strong></td>
<td>SVM</td>
<td>10.19</td>
<td>21.62</td>
<td>13.85</td>
</tr>
<tr>
<td></td>
<td>wvRN</td>
<td>14.83</td>
<td>16.39</td>
<td>15.57</td>
</tr>
<tr>
<td></td>
<td>NTT-FGM</td>
<td>31.14</td>
<td>44.28</td>
<td><strong>36.57</strong></td>
</tr>
</tbody>
</table>
Factor Contribution Analysis

- **NTT-FGM**: Our model
- **NTT-FGM-I**: Our model ignoring influence
- **NTT-FGM-CI**: Our model ignoring influence and correlation
### Efficiency Performance

<table>
<thead>
<tr>
<th>Data Set</th>
<th>Basic NTT-FGM</th>
<th>Distributed NTT-FGM</th>
</tr>
</thead>
<tbody>
<tr>
<td>Twitter</td>
<td>77.7hr</td>
<td>7.0hr</td>
</tr>
<tr>
<td>Flickr</td>
<td>9.14hr</td>
<td>0.68hr</td>
</tr>
<tr>
<td>Arnetminer</td>
<td>100min</td>
<td>6.2min</td>
</tr>
</tbody>
</table>

![Graphs](a) Speedup vs. #cores (b) CPU time vs. Network Density (c) Speedup vs. Network Density
MoodCast: Emotion Prediction via Dynamic Continuous Factor Graph Model

ICDM’10, IEEE Transactions on Affective Computing’11
Happy System

Can we predict users’ emotion?
MoodCast: Dynamic Continuous Factor Graph Model

MoodCast

Social correlation \( g(.) \)

Happy
Neutral
Allen
Mike

Temporal correlation \( h(.) \)

Jennifer today
Jennifer yesterday
Neutral

Attributes \( f(.) \)

Call
sms
location

Predict

Jennifer tomorrow

(a) Temporal correlation
(b) Social correlation
(c) Attribute correlation
Dynamic Continuous Attributes

We cannot use such a NTT-FGM model to do this...

Our solution

1. We directly define continuous feature function;
2. Use Metropolis-Hasting algorithm to learn the factor graph model.
Problem Formulation

\[ G^t = (V, E^t, X^t, Y^t) \]

Attributes:
- Location: Lab
- Activity: Working

Emotion: Sad

Learning Task:
\[ f(V, E^{(t+1)}, X^{(t+1)} \mid G^t) \rightarrow Y^{(t+1)} \]
Dynamic Continuous Factor Graph Model

\[ f_k(x_{ik}^t, y_i^t) \] : Binary function

\[ g(y_i^t, y_j^{t'}) = \exp\left\{ -\beta_{ji} (t - t')(y_i^t - y_j^{t'})^2 \right\} \]

\[ h(y_i^{t'}, y_i^t) = \exp\left\{ -\lambda_i (t - t')(y_i^t - y_i^{t'})^2 \right\} \]
Model Learning

\[
p(Y | G^t) = \frac{1}{Z} \exp \left\{ \sum_{v_i \in V} \sum_{x_{ik}^t \in X} \alpha_k f_k(x_{ik}^t, y_i^t) \right\}
\]

\[
+ \sum_{v_j \in NB(v_i)} \sum_{(y_i^t, y_j^t') \in Y^t} -\beta_{ji}(t - t') (y_i^t - y_j^t')^2
\]

\[
+ \sum_{v_i \in V} \sum_{(y_i^t, y_i^t') \in Y^t} -\lambda_i (t - t') (y_i^t - y_i^t')^2 \}
\]

\[
\theta^* = \arg \max_\theta \log p(Y = y | x, \theta)
\]
Experiment

• Data Set: [http://arnetminer.org/stnt](http://arnetminer.org/stnt)

<table>
<thead>
<tr>
<th></th>
<th>#Users</th>
<th>Avg. Links</th>
<th>#Labels</th>
<th>Other</th>
</tr>
</thead>
<tbody>
<tr>
<td>MSN</td>
<td>30</td>
<td>3.2</td>
<td>9,869</td>
<td>&gt;36,000hr</td>
</tr>
<tr>
<td>LiveJournal</td>
<td>469,707</td>
<td>49.6</td>
<td>2,665,166</td>
<td></td>
</tr>
</tbody>
</table>

• Baseline
  – SVM
  – SVM with network features
  – Naïve Bayes
  – Naïve Bayes with network features

• Evaluation Measure:
  Precision, Recall, F1-Measure
## Performance Result

<table>
<thead>
<tr>
<th>Classifier</th>
<th>Method</th>
<th><strong>MSN Dataset</strong></th>
<th></th>
<th><strong>LiveJournal Dataset</strong></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td>Precision</td>
<td>Recall</td>
<td>F1-score</td>
<td>Precision</td>
</tr>
<tr>
<td>Positive</td>
<td>MoodCast</td>
<td>68.42</td>
<td>69.23</td>
<td><strong>68.82</strong></td>
<td>52.50</td>
</tr>
<tr>
<td></td>
<td>SVM-Simple</td>
<td>60.88</td>
<td>71.08</td>
<td>65.58</td>
<td>49.56</td>
</tr>
<tr>
<td></td>
<td>SVM-Net</td>
<td>59.12</td>
<td>72.70</td>
<td>65.21</td>
<td>50.72</td>
</tr>
<tr>
<td></td>
<td>NB-Simple</td>
<td>67.30</td>
<td>56.21</td>
<td>61.25</td>
<td>57.08</td>
</tr>
<tr>
<td></td>
<td>NB-Net</td>
<td>71.89</td>
<td>56.59</td>
<td>63.33</td>
<td>59.1</td>
</tr>
<tr>
<td>Neutral</td>
<td>MoodCast</td>
<td>67.78</td>
<td>76.57</td>
<td><strong>71.90</strong></td>
<td>59.61</td>
</tr>
<tr>
<td></td>
<td>SVM-Simple</td>
<td>67.39</td>
<td>59.73</td>
<td>63.33</td>
<td>67.58</td>
</tr>
<tr>
<td></td>
<td>SVM-Net</td>
<td>68.42</td>
<td>55.11</td>
<td>61.05</td>
<td>71.21</td>
</tr>
<tr>
<td></td>
<td>NB-Simple</td>
<td>54.14</td>
<td>68.04</td>
<td>60.30</td>
<td>65.95</td>
</tr>
<tr>
<td></td>
<td>NB-Net</td>
<td>51.06</td>
<td>71.62</td>
<td>59.62</td>
<td>61.70</td>
</tr>
<tr>
<td>Negative</td>
<td>MoodCast</td>
<td>30.77</td>
<td>13.95</td>
<td>19.20</td>
<td>45.45</td>
</tr>
<tr>
<td></td>
<td>SVM-Simple</td>
<td>5.63</td>
<td>4.54</td>
<td>5.03</td>
<td>71.67</td>
</tr>
<tr>
<td></td>
<td>SVM-Net</td>
<td>8.18</td>
<td>16.90</td>
<td>11.02</td>
<td>68.78</td>
</tr>
<tr>
<td></td>
<td>NB</td>
<td>14.70</td>
<td>28.16</td>
<td>19.32</td>
<td>54.77</td>
</tr>
<tr>
<td></td>
<td>NB-Net</td>
<td>17.88</td>
<td>32.08</td>
<td><strong>22.96</strong></td>
<td>51.70</td>
</tr>
<tr>
<td>Average</td>
<td>MoodCast</td>
<td>55.66</td>
<td>53.25</td>
<td><strong>53.31</strong></td>
<td>52.52</td>
</tr>
<tr>
<td></td>
<td>SVM-Simple</td>
<td>44.63</td>
<td>45.12</td>
<td>44.65</td>
<td>62.94</td>
</tr>
<tr>
<td></td>
<td>SVM-Net</td>
<td>45.24</td>
<td>48.23</td>
<td>45.76</td>
<td>63.57</td>
</tr>
<tr>
<td></td>
<td>NB-Simple</td>
<td>45.38</td>
<td>50.80</td>
<td>46.95</td>
<td>59.26</td>
</tr>
<tr>
<td></td>
<td>NB-Net</td>
<td>46.94</td>
<td>53.43</td>
<td>48.63</td>
<td>57.5</td>
</tr>
</tbody>
</table>
Factor Contributions

- All factors are important for predicting user emotions.
Conclusions

• We propose Noise Tolerant Time-varying Factor Graphs for modeling and predicting user actions

• We propose MoodCast, a dynamic continuous factor graph model for modeling and predicting user emotions

• Next step: a joint model?
Future Work: Social Prediction…

• To predict more…
Retweet Predicting (CIKM’10)

When you post a tweet...

Who will retweet it?
Following Prediction (submitted)

1. When you follow a friend, how likely he will follow back?

2. Can we predict more such as the following scale and following speed?
A challenging question…

Can social prediction really help human beings?
Representative Publications

• Chenhao Tan, Jie Tang, Jimeng Sun, Quan Lin, and Fengjiao Wang. Social Action Tracking via Noise Tolerant Time-varying Factor Graphs. **KDD’10**.


• Yuan Zhang, Jie Tang, Jimeng Sun, Yiran Chen, and Jinghai Rao. MoodCast: Emotion Prediction via Dynamic Continuous Factor Graph Model. **ICDM'10**.
Thanks!

System: http://arnetminer.org
HP: http://keg.cs.tsinghua.edu.cn/persons/tj/