Patent Partner Recommendation in Enterprise Social Networks

Sen Wu¹, Jimeng Sun², Jie Tang¹

¹Tsinghua University
²IBM TJ Watson Research Center
Patent in Enterprise

- Apple VS. Samsung
- Google VS. Microsoft
- Facebook VS. Microsoft VS. Google
Patenting Competition Network

How companies compete with each other on patents

- Patenting Competition Network
- Microsoft
- Nintendo
- Sony
- Oracle
- Kinsoft
- IBM
- Google
- Facebook
- Twitter
- LinkedIn
- Yandex
- Baidu
- Nokia
- Mozilla
- Apple
- Search Engine
- Web Browser
- Game Console
- Office Suite
- Mobile OS
- SNS
- Computer Hardware

How companies compete with each other on patents.
Patent Collaboration in Enterprise

- Top patent assignee in U.S. (2011)
  - IBM 6,180 (Tops Patent List for 19th Consecutive Year)
  - Samsung (4,894), Canon (2,821)

- Patent collaboration has become everywhere
  - Increasing trend of patent collaborations over the past 35 years
Example

Finding co-inventors in IBM (>300,000 employers)

Recommend Candidates

Interactive feedback

Refined Recommendations

Existing co-inventors

Recommendation

Luo Gang

Philip S. Yu

Kun-Lung Wu

Jimeng Sun

Ching-Yung Lin

Milind R Naphade

Find me a partner to collaborate on Healthcare…

Kun-Lung Wu is matching to me

Philip is not a healthcare people

Recommended collaborators by interactive learning
Challenges

• What are the fundamental factors that influence the formation of co-invention relationships?

• How to design an interactive mechanism so that the user can provide feedback to the system to refine the recommendations?

• How to learn the interactive recommendation framework in an online mode?
Approach Framework

- Candidate generation
- Ranking Factor Graph Model
- Interactive feedback
Stage 1

Candidate generation

Ranking Factor Graph Model

Interactive feedback
Candidate Generation

- Given an inventor network $G=(V, E, X)$, a particular user (inventor) $v_q$ and the topic $t$
- Step 1: use the language model to retrieve a list of “relevant” inventors;
- Step 2: use homophily, referral chaining and recency to select top $K$ candidates.
Candidate Generation

- Homophily
- Referral chaining
- Recency

(a) Interest similarity

\[ CI(v_i, v_j) = \frac{I_{v_i} \cap I_{v_j}}{I_{v_i} \cup I_{v_j}} \]

(b) Referral chaining length

\[ Re(v_i, v_j) = dist(v_i, v_j) \]

(c) Recency

\[ R(v_i, v_j) = \sum_{d_i \in S} e^{-\left(\frac{t_{\text{now}} - t_{d_i}}{\lambda}\right)} \]
Stage 2

Candidate generation

Ranking Factor Graph Model

Interactive feedback
RankFG Model

Map each inventor pair to a node in the graphical model.

Random variable

Social correlation factor function

Pairwise factor function

Recommended inventor
Ranking Factor Graphs

- Pairwise factor function:
  \[ f(v_q, v_i, y_i) = \frac{1}{Z_a} \exp\left\{ \sum_k \alpha_k \psi_k(x_q, x_i, y_i) \right\} \]

- Correlation factor function:
  \[ g(y_i, y_j) = \frac{1}{Z_b} \exp\left\{ \sum_l \beta_l \phi_l(y_i, y_j) \right\} \]

- Log-likelihood objective function:
  \[ \log P(Y|X, \theta) = \sum_{y_i \in Y} \sum_k \alpha_k \psi_k(x_q, x_i, y_i) \]
  \[ + \sum_{v_i \sim v_j} \sum_l \beta_l \phi_l(y_i, y_j) - \log Z \]

- Model learning
  \[ \theta^* = \arg \max_\theta \log P(Y|X, \theta) \]
Learning Algorithm

Input: Query inventors $Q = \{v_q\}$ with corresponding topics $\{q\}$, $G = (V, E, X)$, and the learning rate $\eta$;
Output: learned parameters $\theta$;

$\theta \leftarrow 0$;
repeat
  foreach $v_q \in Q$ and $q$ do
    // Initialization;
    $L \leftarrow$ initialization list;
    Factor graph $FG \leftarrow \text{BuildFactorGraph}(L)$;
    // Learn the parameter $\theta$ for factor graph model;
    repeat
      foreach $v_i \in \text{order}$ do
        Update the messages of $v_i$ by Eqs. 8 and 9;
      end
    until (all messages $\mu$ do not change);
    foreach $\theta_i \in \theta$ do
      Calculate gradient $\nabla_i$ according to Eq. 7;
      Update $\theta_{\text{new}} = \theta_{\text{old}} + \eta \cdot \nabla_i$;
    end
  end
until converge;

Algorithm 1: Learning algorithm for RankFG.
Stage 3

- Candidate generation
- Ranking Factor Graph Model
- Interactive feedback
Still Challenge

How to incrementally incorporate users’ feedback?
Learning Algorithm

Input: Query inventors $Q = \{v_q\}$ with corresponding topics $\{q\}$, $G = (V, E, X)$, and the learning rate $\eta$;

Output: learned parameters $\theta$;

$\theta \leftarrow 0$;

repeat

\begin{enumerate}
  \item \textbf{foreach} $v_q \in Q$ \textbf{and} $q$ \textbf{do}
    \begin{enumerate}
      \item Initialization;
      \item $L \leftarrow$ initialization list;
      \item Factor graph $FG \leftarrow \text{BuildFactorGraph}(L)$;
    \end{enumerate}

  \end{enumerate}

\begin{enumerate}
  \item \textbf{// Learn the parameter $\theta$ for factor graph model;}
    \begin{enumerate}
      \item \textbf{repeat}
        \begin{enumerate}
          \item \textbf{foreach} $v_i \in \text{order}$ \textbf{do}
            \begin{enumerate}
              \item Update the messages of $v_i$ by Eqs. 8 and 9;
            \end{enumerate}
          \end{enumerate}

        \end{enumerate}

    \end{enumerate}

\begin{enumerate}
  \item \textbf{until} (all messages $\mu$ do not change);
\end{enumerate}

\begin{enumerate}
  \item \textbf{foreach} $\theta_i \in \theta$ \textbf{do}
    \begin{enumerate}
      \item Calculate gradient $\nabla_i$ according to Eq. 7;
      \item Update $\theta_{new} = \theta_{old} + \eta \cdot \nabla_i$;
    \end{enumerate}
\end{enumerate}

\end{enumerate}

\begin{enumerate}
  \item \textbf{end}
\end{enumerate}

\begin{enumerate}
  \item \textbf{until} converge;
\end{enumerate}

Algorithm 1: Learning algorithm for RankFG.
Interactive Learning

1) add new factor nodes to the factor graph built in the model learning process.

2) $\ell$-step message passing:
   - Start from the new variable node $y_{N+1}$ (root node).
   - Send messages to all of its neighborhood factors.
   - Propagate the messages up to $\ell$-step
   - Perform a backward messages passing.

3) Calculate an approximate value of the marginal probabilities of the newly factors.

\[
E_{new}[.] = \frac{N}{N+1}E_{old}[.] + \frac{1}{N+1} \sum_k \theta_k \phi_k(x_{N+1}, y_{N+1})
\]
Experiments
Data Set

- PatentMiner (pminer.org)

<table>
<thead>
<tr>
<th>Data Set</th>
<th>Inventors</th>
<th>Patents</th>
<th>Average increase #patent</th>
<th>Average increase #co-invention</th>
</tr>
</thead>
<tbody>
<tr>
<td>IBM</td>
<td>55,967</td>
<td>46,782</td>
<td>8.26%</td>
<td>11.9%</td>
</tr>
<tr>
<td>Intel</td>
<td>18,264</td>
<td>54,095</td>
<td>18.8%</td>
<td>35.5%</td>
</tr>
<tr>
<td>Sony</td>
<td>8,505</td>
<td>31,569</td>
<td>11.7%</td>
<td>13/0%</td>
</tr>
<tr>
<td>Exxon</td>
<td>19,174</td>
<td>53,671</td>
<td>10.6%</td>
<td>14.7%</td>
</tr>
</tbody>
</table>

- Baselines:
  - Content Similarity (Content)
  - Collaborative Filtering (CF)
  - Hybrid
  - SVM-Rank
# Performance Analysis - IBM

**Training:** collaboration before 2000  
**Validation:** 2001-2010

<table>
<thead>
<tr>
<th>Data</th>
<th>ALG</th>
<th>P@5</th>
<th>P@10</th>
<th>P@15</th>
<th>P@20</th>
<th>MAP</th>
<th>R@100</th>
</tr>
</thead>
<tbody>
<tr>
<td>IBM</td>
<td>Content</td>
<td>23.0</td>
<td>23.3</td>
<td>18.8</td>
<td>15.6</td>
<td>24.0</td>
<td>33.7</td>
</tr>
<tr>
<td></td>
<td>CF</td>
<td>13.8</td>
<td>12.8</td>
<td>11.3</td>
<td>11.5</td>
<td>21.7</td>
<td>36.4</td>
</tr>
<tr>
<td></td>
<td>Hybrid</td>
<td>13.9</td>
<td>12.8</td>
<td>11.5</td>
<td>11.5</td>
<td>21.8</td>
<td>36.7</td>
</tr>
<tr>
<td></td>
<td>SVMRank</td>
<td>13.3</td>
<td>11.9</td>
<td>9.6</td>
<td>9.8</td>
<td>22.2</td>
<td>43.5</td>
</tr>
<tr>
<td></td>
<td>RankFG</td>
<td>31.1</td>
<td>27.5</td>
<td>25.6</td>
<td>22.4</td>
<td>40.5</td>
<td>46.8</td>
</tr>
<tr>
<td></td>
<td>RankFG+</td>
<td>31.2</td>
<td>27.5</td>
<td>26.6</td>
<td>22.9</td>
<td>42.1</td>
<td>51.0</td>
</tr>
</tbody>
</table>

RankFG+: it uses the proposed RankFG model with 1% interactive feedback.
Interactive Learning Analysis

Interactive learning achieves a close performance to the complete learning with only $1/100$ of the running time used for complete training.
Parameter Analysis

Factor contribution analysis

Convergence analysis

**RankFG-C**: stands for ignoring referral chaining factor functions.

**RankFG-CH**: stands for ignoring both referral chaining and homophily.

**RankFG-CHR**: stands for further ignoring recency.
Conclusion

• Formulate the problem of patent partner recommendation in enterprise social networks.

• Present a ranking factor graph (RankFG) model for suggesting co-invention relationships.

• Evaluate our proposed model on large patent data sets and illustrate the effectiveness and efficiency of the proposed approach.
Thanks!

Data&Code Download: http://arnetminer.org/patents/
Related Work - Collaboration recommendation

- Collaborative creation of communal hierarchical taxonomies in social tagging systems.
  - P. Heymann and H. Garcia-Molina. [2006]

- Referral web: Combining social networks and collaborative filtering.

- Suggesting friends using the implicit social graph.
  - M. Roth, A. Ben-David, D. Deutscher, G. Flysher, I. Horn, A. Leichtberg, N. Leiser, Y. Matias, and R. Merom. [2010]

- Factorization vs. regularization: fusing heterogeneous social relationships in top-n recommendation.
  - Q. Yuan, L. Chen, and S. Zhao. [2011]
Related Work - Patent search and analysis

- Patentminer: topic-driven patent analysis and mining.
  - J. Tang, B. Wang, Y. Yang, P. Hu, Y. Zhao, X. Yan, B. Gao, M. Huang, P. Xu, W. Li, and A. K. Usadi.[2012]

- Latent graphical models for quantifying and predicting patent quality.

- Text mining techniques for patent analysis.

- Patent maintenance recommendation with patent information network model.
  - X. Jin, S. Spangler, Y. Chen, K. Cai, R. Ma, L. Zhang, X. Wu, and J. Han.[2011]