Community Detection in Heterogeneous Social Network

Deng Cai, Zheng Shao, Xiaofei He, Xifeng Yan, and Jiawei Han. *Mining Hidden Community in Heterogeneous Social Networks*. Proc. of ACM SIGKDD on Link Analysis, 58-65. 2005.

Prof. Shou-de Lin
CSIE/GINM, NTU

2009/11/24
Two Types of Social Networks

- **Homogeneous** ⇒ Single-Relational Network
  - Single node types & Link types

- **Heterogeneous (HSN)** ⇒ Multi-Relational Network
  - Multiple link types and/or node types

- **Example**
  - Homogeneous
  - Heterogeneous
Why Heterogeneous Social Networks

• Drawbacks for **homogeneous** networks
  – Ignore the semantic information.
    • Types of nodes and labels of links.
  – Cannot represent complicated relationships among nodes (e.g. 亦師亦友).

• Real-world social networks are heterogeneous
  – Multiple relationships exist, and each relation can be treated as a **homogeneous network**
  – The homogeneous networks are then overlaid to form an HSN
An Example

• We are given people with different relationships
  – “Work at the same place”, “Share the same interests”, “Send mail frequently”, “Go to the same hospital”, etc.

• Suppose an infectious disease breaks out ...
  – The government needs to find those most likely to be infected
    • Relationships are not equivalent. “Work at the same place” and “Go to the same hospital” are more important.

• Some relations are more important for certain communities
  – Work at the same place → colleague
  – Share the same interests → friends
  – Go to the same hospital →
Examples for the Problem

• Given: A HSN of three different relational graphs, and **four colored nodes** known to be belonged to **the same community**.
  
  – **Relation importance**: (a) > (b) > (c)
  
  – (c) provides negative information for such community
Problem Statement

• Given: An HSN, some labeled examples (e.g., a subset of nodes that belong to the same community)
  – How to evaluate the importance of different relations?
  – How to get a combination of the existing relations which can best match the relation of labeled examples?

• Based on the obtained weighted combination of relations, apply existing methods to detect communities
Examples for the Problem  (cont.)

• Question: what if two objects with lighter color and the two with darker color belong to different communities?

  – Relation importance: (b) > (c) = (a)

[Observation] In HSN, there exists different communities, revealed by different relation types.
Overview

• **Relation Extraction**
  – Find the relations which are most relevant to user-specified label examples
  – Output a weighted relation-combined graph

• **Algorithms**
  1. *Regression-based Algorithm* for multiple communities in query examples
  2. *Minimum-Cut Method* for single community issue
Problem Definition

• Given a set of nodes and a set of relations represented by a set of graphs $G_i(V, E_i), i=1,...,n$ (**n is #relations, V is the set of objects, $E_i$ is the set of edges w.r.t. $i^{th}$ relation**)

• Given $M_i$ as the weight matrix associated with $G_i$
  – Weight represents the strength of the edge (relation)

• Suppose there exists a hidden relation represented by a graph $\hat{G}(V, \hat{E})$, and $\hat{M}$ denote the weight matrix (e.g. represents the group info) associated with $\hat{G}$

• Given a set of labeled object $X=[x_1, ..., x_m]$ and $y=[y_1, ..., y_m]$ where $y_i$ is the label of $x_j$

• Find a **linear combination of the weight matrices** which provides the **best estimation of** $\hat{M}$
Basic Idea

• Find an combined relation which make
  – The relationship between the intra-community nodes as tight as possible AND
  – The relationship between the inter-community nodes as loose as possible
Regression-based Algorithm

- Represent the user-given label examples
  - Construct the target relation between label objects

\[
\tilde{M}_{ij} = \begin{cases} 
1, & \text{example } i \text{ and example } j \text{ have the same label;} \\
0, & \text{otherwise.} 
\end{cases}
\]

- The probability version

\[
\tilde{M}_{ij} = \text{Prob}(x_i \text{ and } x_j \text{ belong to the same community})
\]
Regression-based Algorithm

• Find a **linear combination** of the existing relations to optimally approximate the target relation in $L_2$ norm

• Let $\mathbf{a} = [a_1, a_2, ..., a_n]^T \in \mathbb{R}^n$ be the **coefficients** for different relations

• This approximation problem can be solved as

$$
\mathbf{a}^{opt} = \arg \min_{\mathbf{a}} \| \hat{\mathbf{M}} - \sum_{i=1}^{n} a_i \mathbf{M}_i \|^2
$$

- **Vector** form (since $\mathbf{M}$ is symmetric), $m(m-1)/2$ dim

$$
\mathbf{a}^{opt} = \arg \min_{\mathbf{a}} \| \hat{\mathbf{v}} - \sum_{i=1}^{n} a_i \mathbf{v}_i \|^2 \rightarrow \text{Linear regression}
$$
Two Problems for Linear Regression

1. **Accuracy**
   
   - The least-squares estimates often have **low bias** but **large variance**
     
     - Low bias $\rightarrow$ small training error
     - Large variance $\rightarrow$ tend to overfit data
   
   - To reduce the variance of the predicted relation strength, we need to **sacrifice or set some coefficients to zero**

2. **Interpretation**
   
   - We often want to obtain a smaller subset that exhibit the strongest effect
   
   - Need to **sacrifice some of the small details**
An Illustration for Overfitting

• Suppose we have a user query \((o_1, ..., o_5)\), where \(o_1, o_2,\) and \(o_3\) belong to one community, but \(o_4\) and \(o_5\) belong to another

• The target relation network can be...

\[
\begin{array}{cccccc}
  & o_1 & o_2 & o_3 & o_4 & o_5 \\
 o_1 & * & 1 & 1 & 0 & 0 \\
 o_2 & 1 & * & 1 & 0 & 0 \\
 o_3 & 1 & 1 & * & 0 & 0 \\
 o_4 & 0 & 0 & 0 & * & 1 \\
 o_5 & 0 & 0 & 0 & 1 & * \\
\end{array}
\]

* means we do not consider the self-relation strength
An Illustration (cont.)

• The four basic relation matrices are

\[
\begin{array}{cccccccccccc}
  o_1 & o_2 & o_3 & o_4 & o_5 & o_1 & o_2 & o_3 & o_4 & o_5 & o_1 & o_2 & o_3 & o_4 & o_5 \\
  o_1 & * & 0.8 & 0.7 & 0 & 0 & o_1 & * & 0.1 & 0 & 0 & o_1 & * & 0.1 & 0 & 0 & 0 & 0 \\
  o_2 & 0.8 & * & 0.9 & 0 & 0 & o_2 & 0 & * & 0 & 0 & o_2 & 0.1 & * & 0 & 0 & 0 & 0 \\
  o_3 & 0.7 & 0.9 & * & 0 & 0 & o_3 & 0.1 & 0 & * & 0 & 0 & o_3 & 0 & 0 & * & 0 & 0 \\
  o_4 & 0 & 0 & 0 & * & 0.6 & o_4 & 0 & 0 & 0 & * & 0 & o_4 & 0 & 0 & 0 & * & 0 \\
  o_5 & 0 & 0 & 0 & 0.6 & * & o_5 & 0 & 0 & 0 & 0 & * & o_5 & 0 & 0 & 0 & 0 & * \\
\end{array}
\]

(a) Relation $M_1$  (b) Relation $M_2$  (c) Relation $M_3$  (d) Relation $M_4$

• $0M_1 + 10M_2 + 10M_3 + 10M_4$ can exactly match the example relation matrix
  – However, it is not a good approximation

• Using $M_1$ itself can approximate the original matrix accurately.
Apply **Coefficient Shrinkage**

- Normalize all the weights on the edges to $[0,1]$
- Add the constraint
  \[ \sum_{i=1}^{n} a_i^2 \leq 1 \]
  on the objective function
- Finally we solve this minimization problem
  \[
  a^{opt} = \arg \min_{a} \| \tilde{v} - \sum_{i=1}^{n} a_i e_i \|^2
  \]
  subject to \[ \sum_{i=1}^{n} a_i^2 \leq 1 \]
- When using such constraint, the above coefficients become 1, 0, 0, 0

Such a constraint regression is called **Ridge Regression**
Regression-based Algorithm May Fail

- When **the example provided** by the user **belong to only one community**
  - Referred as “**Single Community Issue**”

- Eventually, when the **user-provided examples** belong to
  - **Different** communities → **Regression**-based Algo.
  - **The same** community → **MinCut**-based Algo.
Single Community Issue

• Suppose we have a user query \((o_1, \ldots, o_5)\), which belong to the same community

• Given **two** relation matrices:

\[
\begin{array}{ccccc}
  o_1 & o_2 & o_3 & o_4 & o_5 \\
  o_1 & * & 0.9 & 0.8 & 1 & 0.1 \\
  o_2 & 0.9 & * & 1 & 0.9 & 0.1 \\
  o_3 & 0.8 & 1 & * & 1 & 0.1 \\
  o_4 & 1 & 0.9 & 1 & * & 0.1 \\
  o_5 & 0.1 & 0.1 & 0.1 & 0.1 & * \\
\end{array}
\quad \begin{array}{ccccc}
  o_1 & o_2 & o_3 & o_4 & o_5 \\
  o_1 & * & 0.4 & 0.4 & 0.5 & 0.5 \\
  o_2 & 0.4 & * & 0.3 & 0.2 & 0.4 \\
  o_3 & 0.4 & 0.3 & * & 0.3 & 0.3 \\
  o_4 & 0.5 & 0.2 & 0.3 & * & 0.4 \\
  o_5 & 0.5 & 0.4 & 0.3 & 0.4 & * \\
\end{array}
\]

(a) Relation \(M_1\)  
(b) Relation \(M_2\)

• The **Regression method would prefer** \(M_1\), since the higher connectivity between \(o_1, o_2, o_3, o_4\) achieves a lower square error to the target relation
Single Community Issue (cont.)

- However, in $M_1$, **the connectivity between $o_5$ and the other four nodes is very weak.**
- The connectivity in $M_2$ is much more uniform than that in $M_1$ with comparable strength.
  - Therefore, $M_2$ should be a better choice for the query.
- Unfortunately, **the square error of $M_2$ is larger** than that of $M_1$, Regression method may fail.

<table>
<thead>
<tr>
<th></th>
<th>$o_1$</th>
<th>$o_2$</th>
<th>$o_3$</th>
<th>$o_4$</th>
<th>$o_5$</th>
<th>$o_1$</th>
<th>$o_2$</th>
<th>$o_3$</th>
<th>$o_4$</th>
<th>$o_5$</th>
</tr>
</thead>
<tbody>
<tr>
<td>$o_1$</td>
<td>*</td>
<td>0.9</td>
<td>0.8</td>
<td>1</td>
<td>0.1</td>
<td>$o_1$</td>
<td>*</td>
<td>0.4</td>
<td>0.4</td>
<td>0.5</td>
</tr>
<tr>
<td>$o_2$</td>
<td>0.9</td>
<td>*</td>
<td>1</td>
<td>0.9</td>
<td>0.1</td>
<td>$o_2$</td>
<td>0.4</td>
<td>*</td>
<td>0.3</td>
<td>0.2</td>
</tr>
<tr>
<td>$o_3$</td>
<td>0.8</td>
<td>1</td>
<td>*</td>
<td>1</td>
<td>0.1</td>
<td>$o_3$</td>
<td>0.4</td>
<td>0.3</td>
<td>*</td>
<td>0.3</td>
</tr>
<tr>
<td>$o_4$</td>
<td>1</td>
<td>0.9</td>
<td>1</td>
<td>*</td>
<td>0.1</td>
<td>$o_4$</td>
<td>0.5</td>
<td>0.2</td>
<td>0.3</td>
<td>*</td>
</tr>
<tr>
<td>$o_5$</td>
<td>0.1</td>
<td>0.1</td>
<td>0.1</td>
<td>0.1</td>
<td>*</td>
<td>$o_5$</td>
<td>0.5</td>
<td>0.4</td>
<td>0.3</td>
<td>0.4</td>
</tr>
</tbody>
</table>

(a) Relation $M_1$  
(b) Relation $M_2$
Minimum-Cut Method (1/5)

• To deal with the single community issue, the minimum cut can be used to evaluate the tightness of the graph.
• Let G be a weighted graph with weight matrix M.
• A cut is a set of edges which separates the vertices into two disconnected groups A & B such that $A \cap B = \emptyset$ and $A \cup B = G$. The value of that cut is defined as

$$\text{cut}(G) = \sum_{i \in A} \sum_{j \in B} M(i, j)$$
Minimum-Cut Method (2/5)

• Totally $2^m - 2$ different cuts in $G$ (m: #vertices)
• Let $\text{cut}_k(G) = (A_k, B_k)$ denote the $k$-th cut
• The minimum cut of $G$ is defined as

$$\text{mincut}(G) = \min_k \{\text{cut}_k(G)\}$$

  – If $G$ is easily cut into two subgraphs $\Rightarrow$ small minCut
  – If $G$ is disconnected $\Rightarrow$ minCut = 0

• Thus, for single community issue, we try to extract the optimal relation graph by maximizing its minimum cut value
Minimum-Cut Method (3/5)

• Let $G_i$, $i=1,...,n$, be the existing relation graphs defined only on the user query example and $M_i$ be the corresponding weight matrices.

• Let $a = [a_1,...,a_n]^T \in \mathbb{R}^n$ be the combination coefficients for different relation graphs.
  
  – Thus, $M = \sum_{i=1}^{n} a_i M_i$ is the weight matrix of the combined relation graph $G$.

• Objective function

$$a^{opt} = \arg \max_a \{ \text{mincut}(\sum_{i=1}^{n} a_i G_i) \}$$
Minimum-Cut Method (4/5)

\[
\text{mincut}(G) = \min_{1 \leq k \leq 2^m} \{ \text{cut}_k(G) \}
\]

\[
= \min_{1 \leq k \leq 2^m-2} \left\{ \sum_{i \in A(k)} \sum_{j \in B(k)} M(i, j) \right\}
\]

\[
= \min_{1 \leq k \leq 2^m-2} \left\{ \sum_{i \in A(k)} \sum_{j \in B(k)} \left( \sum_{l=1}^{n} a_l M_l(i, j) \right) \right\}
\]

\[
= \min_{1 \leq k \leq 2^m-2} \left\{ \sum_{l=1}^{n} a_l \left( \sum_{i \in A(k)} \sum_{j \in B(k)} M_l(i, j) \right) \right\}
\]

\[
= \min_{1 \leq k \leq 2^m-2} \left\{ \sum_{l=1}^{n} a_l \cdot \text{cut}_k(G_l) \right\}
\]
Minimum-Cut Method (5/5)

• Let $v = \text{minCut}(G)$, it can be reduced to the following linear programming problem

$$
\begin{align*}
\text{max} & \quad v \\
\text{st.} & \quad \sum_{l=1}^{n} a_l \cdot \text{cut}_k(G_l) - v \geq 0, \quad (1 \leq k \leq 2^{m} - 2) \\
& \quad \sum_{l=1}^{n} a_l = 1 \\
& \quad a_l \geq 0, \quad (1 \leq l \leq n)
\end{align*}
$$
Community Detection in HSN

1. Learning the weight $a_i$ for each type of the relations.
2. Construct a new matrix using $\sum a_i M_i$
3. Apply Existing homogeneous community detection algorithm on this new matrix
Experiment-1: Synthetic HSN on Iris Data

• It contains 3 classes of 50 instances, each class refers to a type of Iris plant
  – Each instance has 4 features
• It is viewed as a HSN with 3 hidden communities
  – The 4 relation matrices $M_1, \ldots, M_4$ are constructed from the 4 features
Experiment-1 (cont.)

• Baseline
  – Take the HSN as a homogeneous one as
    \[ M' = \sum_{i=1}^{4} 0.25M_i \]
  – Apply existing Normalized-Cut algo. for community

• Extracted Relation
  – Use Regression-based algo. to get the combined-relation graph as
    \[ M' = \sum_{i=1}^{4} a_i M_i \]
  – Apply existing Normalized-Cut algo. for community
Experiment-1 (cont.)

![Graph showing community mining accuracy vs. examples ratio (%)](image1)

(a) Baseline Relation

(b) Extracted Relation
Experiment-2: HSN on DBLP Data

• By May 2004, 500000 papers, 1000 conferences
• Authors as nodes, conferences as relations
  – Thus, totally 1000 relations
• Qualitative experiment to show the effectiveness
• Given some examples (i.e., a group of authors), this is to study how to
  1. Extract a combined-relation using such examples
  2. And find all the other groups in such relation
  – The extracted relation can be interpreted as the groups of authors share similar interests
Experiment-2: Results

**Experiment 1.** In the first case, there are two queries provided by the user.


### Resulted coefficients of different conference graphs:

<table>
<thead>
<tr>
<th>Conference</th>
<th>Coefficient</th>
<th>Conference</th>
<th>Coefficient</th>
</tr>
</thead>
<tbody>
<tr>
<td>KDD</td>
<td>1</td>
<td>SIGMOD</td>
<td>0.528</td>
</tr>
<tr>
<td>ICDE</td>
<td>0.262</td>
<td>VLDB</td>
<td>0.210</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Researcher</th>
<th>KDD</th>
<th>ICDE</th>
<th>SIGMOD</th>
<th>VLDB</th>
</tr>
</thead>
<tbody>
<tr>
<td>Philip S. Yu</td>
<td>7</td>
<td>15</td>
<td>10</td>
<td>11</td>
</tr>
<tr>
<td>Rakesh Agrawal</td>
<td>6</td>
<td>10</td>
<td>13</td>
<td>15</td>
</tr>
<tr>
<td>Hans-Peter Kriegel</td>
<td>7</td>
<td>9</td>
<td>11</td>
<td>8</td>
</tr>
<tr>
<td>Padhraic Smyth</td>
<td>10</td>
<td>1</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>Bing Liu</td>
<td>8</td>
<td>1</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>Pedro Domingos</td>
<td>8</td>
<td>0</td>
<td>2</td>
<td>0</td>
</tr>
<tr>
<td>Hector Garcia-Molina</td>
<td>0</td>
<td>15</td>
<td>12</td>
<td>12</td>
</tr>
<tr>
<td>David J. DeWitt</td>
<td>1</td>
<td>4</td>
<td>20</td>
<td>16</td>
</tr>
<tr>
<td>Michael Stonebraker</td>
<td>0</td>
<td>12</td>
<td>19</td>
<td>15</td>
</tr>
</tbody>
</table>
Experiment-2: Results

• With the above extracted relation graph, perform community detection, example results:

1. Community for query 1: Alexander Tuzhilin, Bing Liu, Charu C. Aggarwal, Dennis Shasha, Eamonn J. Keogh, 
   ....

2. Community for query 2: Alfons Kemper, Amr El Abbadi, Beng Chin Ooi, Bernhard Seeger, Christos Faloutsos, 
   ....

• Take only the first three names for relation extraction

<table>
<thead>
<tr>
<th>Conference Name</th>
<th>Coefficient</th>
</tr>
</thead>
<tbody>
<tr>
<td>SIGMOD</td>
<td>0.302</td>
</tr>
<tr>
<td>KDD</td>
<td>0.279</td>
</tr>
<tr>
<td>ICDE</td>
<td>0.217</td>
</tr>
<tr>
<td>VLDB</td>
<td>0.202</td>
</tr>
</tbody>
</table>
Experiment 2: Results

Experiment 2. Let us try another example. The two queries are:

1. Pat Langley, Andrew W. Moore, Michael J. Pazzani, James P. Callan, Yiming Yang, Thomas G. Dietterich

2. Pat Langley, Andrew W. Moore, Michael J. Pazzani, Raymond T. Ng, Philip S. Yu

<table>
<thead>
<tr>
<th>Query 1</th>
<th>Query 2</th>
</tr>
</thead>
<tbody>
<tr>
<td>Conference</td>
<td>Coefficient</td>
</tr>
<tr>
<td>ICML</td>
<td>1</td>
</tr>
<tr>
<td>ICML</td>
<td>0.329</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Researcher</th>
<th>ICML</th>
<th>KDD</th>
<th>SIGIR</th>
<th>SIGMOD</th>
</tr>
</thead>
<tbody>
<tr>
<td>Pat Langley</td>
<td>11</td>
<td>4</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>Andrew W. Moore</td>
<td>10</td>
<td>3</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>Michael J. Pazzani</td>
<td>10</td>
<td>6</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>James P. Callan</td>
<td>3</td>
<td>0</td>
<td>10</td>
<td>1</td>
</tr>
<tr>
<td>Yiming Yang</td>
<td>5</td>
<td>1</td>
<td>8</td>
<td>0</td>
</tr>
<tr>
<td>Thomas G. Dietterich</td>
<td>12</td>
<td>1</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>Raymond T. Ng</td>
<td>0</td>
<td>8</td>
<td>0</td>
<td>7</td>
</tr>
<tr>
<td>Philip S. Yu</td>
<td>1</td>
<td>7</td>
<td>0</td>
<td>10</td>
</tr>
</tbody>
</table>

2009/11/24
SNA09f, Prof. Shou-de Lin
Problems for such Approach

1. Still, it treats different types of relations independently (never consider the joint-exist of different types of relations).

2. In the min-cut method, the surrounding environment is not considered.

3. More?
Review: Community in HSN

• Relation Extraction
  – Find the relations which are most relevant to user-specified label examples
  – Output a weighted relation-combined graph

• Algorithms
  1. Regression-based Algorithm for multiple communities in query examples
  2. Minimum-Cut Method for single community issue

• Community Detection in HSN
  – Then apply existing homogeneous network methods to discover the communities