Title
Mining User Groups in the Social News Website: Community Detection in Bipartite Networks.

Permalink
https://escholarship.org/uc/item/3zw7332w

Author
YANG, HO-SHUN

Publication Date
2012

Peer reviewed|Thesis/dissertation
Mining User Groups in the Social News Website: Community Detection in Bipartite Networks

A thesis submitted in partial satisfaction of the requirements for the degree Master of Science in Statistics

by

Ho-shun Yang

2012
Abstract of the Thesis

Mining User Groups in the Social News Website: Community Detection in Bipartite Networks

by

Ho-shun Yang

Master of Science in Statistics

University of California, Los Angeles, 2012

Professor Mark S. Handcock, Chair

Community clustering is well-studied in the context of social network analysis. However, in web services such as online retailing, music sharing library and social news website, the user data are more naturally modeled by the bipartite networks, where users are one class of the nodes and the product, song or story are the other. In this work, we propose a logistic weight model to transform the bipartite network to a weighted uni-partite network for clustering purpose. We experiment the model on Balatarin.com, a social news website and show the model leads to remarkably better clustering results by identifying and exploiting more informative users from the data.
The thesis of Ho-shun Yang is approved.

Vwani Roychowdhury

Qing Zhou

Mark S. Handcock, Committee Chair

University of California, Los Angeles

2012
To my mother . . .

For her love and devotion to the family
# Table of Contents

1 Introduction ................................................. 1
   1.1 Balatarin.com ........................................... 2

2 Approach ..................................................... 3
   2.1 Network construction process ......................... 4
      2.1.1 The logistic weight model ......................... 4
   2.2 Community clustering algorithms ....................... 5

3 Experiment .................................................. 7
   3.1 Data summary ............................................ 7
   3.2 Network construction ................................... 8
      3.2.1 The Jaccard similarity measure ................... 8
      3.2.2 The logistic weight model ......................... 9

4 Analysis and Results ....................................... 11
   4.1 Evaluation metrics .................................... 11
   4.2 Results ................................................ 12
   4.3 The logistic weight model ............................. 13

5 Discussion .................................................. 18
   5.1 Network representation ................................. 18
   5.2 Mining user behaviors with indirect information .... 19

References .................................................... 20
List of Figures

3.1 Log-log plots of distribution of votes over users and links in Politics category. ...................................................... 8

3.2 Clustering result on the network constructed with the Jaccard similarity measure. There are only two very large communities, which in combination, account for 84.05% of the total number of nodes, leading to an undesirable low community resolution. Nodes in these two communities are respectively colored black and red. ............. 10

4.1 The sum of edge weights for an user with certain number of votes. The logistic weight model specifically gives users with intermediate number of votes higher weights, suggesting users with these number of votes are more informative for the mining purpose. On the other hand, the Jaccard similarity measure treats users with different number of votes roughly the same. ......................... 14

4.2 Clustering result on the network constructed with logistic weighted model with \((\beta_{SUV}, \beta_{LV}) = (2.0, 0.1)\). There are 6 non-trivial communities with sizes of 873, 841, 711, 381, 336 and 286. .............. 17
LIST OF TABLES

3.1 Brief summary for the Balatarin.com data .......................... 7

4.1 The metrics of the constructed networks with $\beta_{SUV} = 0.1$ and different $\beta_{LV}$. NSD is the normalized standard deviation. ED is the effective diameter. MOD is the modularity measure. The definitions of the above metrics are defined in Section 4.1. comp. size is the largest component size of the network. $\omega_{th}$ is the lower bound for the edge weights. n-th is the n-th largest community size output by the clustering algorithm. .......................... 15

4.2 The metrics of the constructed networks with $\beta_{LV} = 0.1$ and different $\beta_{SUV}$. NSD is the normalized standard deviation. ED is the effective diameter. MOD is the modularity measure. The definitions of the above metrics are defined in Section 4.1. comp. size is the largest component size of the network. $\omega_{th}$ is the lower bound for the edge weights. n-th is the n-th largest community size output by the clustering algorithm. .......................... 16
ACKNOWLEDGMENTS

I am grateful to Professor Mark S. Handcock for his supervision for the thesis. Without his encouragements and high standard, I would have made too many compromises. Thanks also go to Professor Vwani Roychowdhury for supporting me with the laboratory resources; and Professor Qing Zhou for guiding me into the field of Statistics. I also like to thank PhD candidate Roja Bandari for introducing me to Balatarin.com, which provides the main data of the project. I am appreciate to Dr. Hai Qian for his patient helps in computation problems; PhD candidate Lichao Chen for the wise advices; and special thanks to PhD candidate Benyue Chang for all the helpful discussions.
CHAPTER 1

Introduction

In many web services, users do not directly interact with each other, instead, they interact with the service they use. For example, users shop with the online retailer, listening songs from the music sharing library and reading stories from social news website. Those relations between the users and specific objects are naturally modeled by the bipartite networks. Mining user behavior from those bipartite data is essential for developing better recommendation systems [9, 16], advertisement techniques [12] and even understanding to provided contents [8].

However, graph mining methods for bipartite networks are less well-developed compared to those for uni-partite networks. Therefore, it is common practice to transform the bipartite network to the uni-partite network by direct projection\(^1\) [2, 15] or computing the edge weight between nodes in the same class [16]. In this work, we propose a logistic weight model to compute the edge weights for the community detection purpose. The model is incorporated into a user group mining process as described in Chapter 2. In Chapter 3, We experiment the process with Balatarin.com, a Persian social news website. The results and the special property of logistic weight model are discussed in Chapter 4.

\(^1\)Assume that a bipartite network has two node classes \(A\) and \(B\). The direct projection of the bipartite network on to class \(A\) is to connect nodes in \(A\) if they have at least one common neighbor in \(B\).
1.1 Balatarin.com

Balatarin.com was one of the most popular Persian website and played a crucial rule in sharing news during the 2009 political turmoil in Iran [1]. In the website, registered users can post links (mostly an URL) of their interests in one of the seven categories\(^2\). After a link is posted, other users can vote up or down on the link, showing their attitude to the associated contents. If a link receives enough votes, it would be promoted to the front page and subsequently become more popular. In this work, we focus on mining user groups in terms of the political views or interests, and therefore utilize link information only from the Politics category.

\(^2\)The seven categories are Politics, Art and Literature, Science and Technology, Economy, Community, Exercise and Entertainment
CHAPTER 2

Approach

Superficially, we can view the data from social news website as a bipartite network, where users are one class of node and the posted links are the other. The user and link are connected if the user posts, comments or up/down votes the links\(^1\). For clarity, we will use user and link to denote the nodes from the two classes of the bipartite network. However, our approach is general and can be applied to other data sets with the bipartite structure.

To take advantage of the most graph algorithms and metrics, we first construct an user network from the bipartite network by computing the edge weights between users. In the other words, we transform the indirect relationship between users to the direct relationship according to their features defined in the link space. We call this process the network construction process. Secondly, we apply the graph clustering algorithm to obtain a set of communities and select those communities with non-trivial sizes as our user groups. Finally, for each user who is not in the existing groups, we classify the user to a group with the largest sum of edge weights between the user and the group members. We call this the classification process.

\(^1\)It would be more clear to call the link as "story", but we stay with the notation used by the Balatarin.com.
2.1 Network construction process

The network construction process is the essence of our approach because the representation of the network strongly effects the clustering result. Noticeably, we do not need to use all of the nodes in the bipartite network to construct the new network because not all the nodes are informative. For example, an user votes for a few stories and an users votes for almost every stories both provide little or no information for the clustering purpose, even more, the user who votes a lot may lure the clustering algorithm to merge irrelevant communities. Therefore, for noise reduction reason, those users should be excluded from the network. Then after the communities are detected, they can be classify into one of the communities through the classification process.

2.1.1 The logistic weight model

Here, we propose a logistic weight model to compute the edge weights for the uni-partite network. Let \( G(U, L, E) \) be the bipartite graph where \( U, L \) are the user set and link set and \( E \) is the edge set. Given a pair of users, \( u_s \) and \( u_t \), assume that they both vote for links \( \ell_1, \ell_2, \ldots, \ell_m \). The edge weight between them is given by Equation (2.1).

\[
\omega(u_s, u_t) = \frac{\sum_{i=1}^{m} \exp \left( -\bar{\beta} \cdot \tilde{f}(u_s, u_t, \ell_i) \right)}{1 + \exp \left( -\bar{\beta} \cdot \tilde{f}(u_s, u_t, \ell_i) \right)} \tag{2.1}
\]

where \( \tilde{f}(u_s, u_t, \ell_i) \) describes the strength of the relation between \( u_s \) and \( u_t \) through \( \ell_i \) and \( \bar{\beta} \) is the model parameter. In particular, we let \( \tilde{f}(u_s, u_t, \ell_i) \) to be

\[
\tilde{f}(u_s, u_t, \ell_i) = \left( \frac{\text{Deg}(u_s) + \text{Deg}(u_t)}{2 \times \text{Avg}(u)}, \frac{\text{Deg}(\ell_i)}{\text{Avg}(\ell)} \right) \tag{2.2}
\]

and

\[
\bar{\beta} \cdot \tilde{f}(u_s, u_t, \ell_i) = \beta_{\text{UV}} \left( \frac{\text{Deg}(u_s) + \text{Deg}(u_t)}{2 \times \text{Avg}(u)} \right) + \beta_{\text{LV}} \frac{\text{Deg}(\ell_i)}{\text{Avg}(\ell)} \tag{2.3}
\]
Deg(u) is the degree, or number of votes, of u and Deg(l) is the degree of l. Avg(u) = ∥E∥/∥U∥ and Avg(l) = ∥E∥/∥L∥ are the average degree of the vertices. The intuition is that if a link is voted by only a few users, then these users are likely to be similar to each other. On the contrary, two users may not be similar even if they both vote for a very popular links, which is voted by almost every users. Finally, edge weights less than a specific threshold ω_{th} are discarded for the efficiency purpose.

The \vec{f}(u_s, u_t, ℓ_i) we use here involve only graph structure information from the bipartite network. However, other information can be easily incorporated into the model. For example, we can add the gender, age, or other quantifiable features of users to Equation 2.2. There is no clear objective function for fitting the model parameter. In this work, we choose the parameters that leads to best clustering results as discussed in Section 4.2.

### 2.2 Community clustering algorithms

The input of the graph clustering algorithm is an uni-partite graph, which could be undirected, directed or weighted. The output of the algorithm is a set of subgraphs (communities), which may or may not contain all the vertices from the graph. Intuitively, the quality of the cluster is good if the vertices in the same cluster tend to connect to each others, while seldom connect to vertices outside the cluster.

A thorough overview of the large body of works on clustering is beyond the scope and we refer the interested readers to the review papers of Fortunato [4] and Schaeffer [14]. For our clustering purpose, the algorithm should meet two requirements. First, it should partition the whole graph, that is, every vertex would be assigned to a community. Second, the algorithm should try to produce many communities with comparable sizes, instead of a few very large communities.
We have tested the simulated annealing algorithm [13], fast greedy algorithm [3] and walk trap algorithm [11] on our data. The performances of algorithms are measured by the Newman modularity measure [10] and the community size distribution. In general, the simulated annealing algorithm produces the highest modularity values, but the difference is very limited, usually within 0.01 compared to the other two algorithms. However, the cost of this slight gain is very high, the algorithm easily takes more than 10 times longer to complete the clustering because of its slow annealing process. The remaining two algorithms have comparable performances in terms of modularity, but the fast greedy algorithm seems to have little advantage on producing communities with comparable sizes. Therefore, in this work, we will perform the clustering with the fast greedy algorithm. However, in practice, it is recommended to apply a set of clustering algorithms to see which one gives the better results on a specific graph.
CHAPTER 3

Experiment

We will experiment the groups mining approach on the Balatarin.com data. In particular, we will utilize two methods to construct the uni-partite network. The first one is the well-known Jaccard similarity measure, and the other is the proposed logistic weight model.

3.1 Data summary

We are granted to use the user data of Balatarin.com from August 16, 2008 to November 25, 2010. The brief summary of the whole data set is shown in Table 3. In this work, we will focus on the Politics category. There are 351,801 links in the Politics category and 22,943 users have voted for at least one of those links. The total number of votes received by those links is 9,899,035. The links, users and votes respectively account for 28.30%, 62.77% and 31.91% of the total number of links, users and votes. It is apparent that the political category is one of the most active category in the website.

The cumulative distributions of votes per users and votes per link are shown in Figure 3.1. The distributions roughly follow the power law distribution, meaning

<table>
<thead>
<tr>
<th>users</th>
<th>links</th>
<th>votes</th>
<th>category</th>
<th>Time span</th>
</tr>
</thead>
<tbody>
<tr>
<td>36,549</td>
<td>1,243,289</td>
<td>31,020,346</td>
<td>7</td>
<td>51 months</td>
</tr>
</tbody>
</table>

Table 3.1: Brief summary for the Balatarin.com data
many users do not vote much and some of the users vote extremely frequently. As discussed in Section , those users with extreme number of votes provide less information and may bring in additional noises. Thus, in the first phase of network construction, we keep users with more than 100 votes and less than 3000 votes, which leave us with 6985 users, or 30.44% of the total political users.

3.2 Network construction

3.2.1 The Jaccard similarity measure

For the comparison purpose, we construct a network with edge weights computed by the Jaccard similarity measure using Equation (3.1).

$$\omega(u_s, u_t) = \frac{\text{total number of links voted by } u_s \text{ and } u_t}{\text{total number of links voted by } u_s \text{ or } u_t}$$

(3.1)

The clustering result of the constructed network is shown in Figure 3.2.1. The three largest communities sizes are 3195, 2676 and 365 and all of the rest
community sizes are less than 2. It shows that we can only classify users into two or three groups, which is not that informative. We would discuss the implication of this result later along side with the logistic weight model.

3.2.2 The logistic weight model

The logistic weight model have two parameters $\beta_{SUV}$ and $\beta_{LV}$ as shown in Equation 2.3, where $\beta_{SUV}$ puts penalty on the number of votes of the two users and $\beta_{LV}$ puts penalty on the number of votes of the link.

As discussed before, the objective function for the parameters are hard to define. Here, we first perform a grid search on different combination of the parameters and filter out parameters that leads to network with low modularity measures. Then we closely examine the remaining parameters with the metrics described in Section 4.1.
Figure 3.2: Clustering result on the network constructed with the Jaccard similarity measure. There are only two very large communities, which in combination, account for 84.05% of the total number of nodes, leading to an undesirable low community resolution. Nodes in these two communities are respectively colored black and red.
CHAPTER 4

Analysis and Results

4.1 Evaluation metrics

The communities or the user groups we extracted may reveal interesting patterns or turn out to be artifacts. The final judgments should be left to the domain experts. However, it is our responsibility to provide quantitative evidences to support our results. The following are the proposed metrics:

1. Normalized edge weight variance (NSD): It is defined by dividing the standard deviation of the edge weights by the mean of the edge weights. We prefer this value to be small. Because if the variance is large, the clustering algorithm would tend to focus on those edges with disproportionately high weights and ignoring information provided by the other edges.

2. Effective diameter (ED): The effective diameter is a more flexible version of the graph diameter. Unlike graph diameter, which defined on all the vertices pairs, effective diameter is the path length within which 90% of vertices pairs can reach each others. We prefer this value to be larger. Because the small value implies a graph that is well-connected and that would make clustering more difficult.

3. Newman Modularity measure (MOD): Modularity is the measure for the quality of the communities. It measures the strength of connectivity within communities compared to the "baseline strength" given by randomly rewiring
the same degree sequence [5, 10]. The formal definition is

\[
MOD = \sum_{i=1}^{k} \left[ e_i - \left( \frac{d_i}{2\|E\|} \right)^2 \right]
\]  

(4.1)

where \(e_i\) is the number of edges in community \(i\), \(d_i\) is the total degrees of a community and \(\|E\|\) is the number of edges in the network. The value range of MOD is \([-1/2, 1]\). Although theoretically the value larger than 0 outperforms the null model. It is not uncommon to see modularity values around 0.3 to 0.5 in random graphs with specific degree sequences [6].

4. Community sizes distribution: Community sizes distribution is the most relevant measure for our purpose. We strongly prefer communities with large and comparable sizes because that would give us more information about the underlying patterns.

4.2 Results

With different model parameters, we construct the networks and evaluate them by the metrics described above. Table 4.1 and Table 4.2 demonstrate some explanatory results, where in Table 4.1, we fix \(\beta_{SUV}\) and change \(\beta_{LV}\) and do the contrary in Table 4.2.

First, we observe that the constructed network consists of a large connected component and many peripheral tiny components. With a fixed \(\omega_{th}\), the large penalty terms lead to smaller number of edges, hence decreasing the size of the largest connected component. We see the size of the largest connected component is very sensitive to \(\beta_{LV}\) that from \(\beta_{LV} = 0.1\) to \(\beta_{LV} = 0.4\), the largest component size drops by half. In the mean while, we find the community resolution is still low and the other metrics remain stable. It hints us to focus more on \(\beta_{SUV}\) to choose the right parameters. The metrics and the parameter selection would be discussed in details in the next chapter.
Surprisingly, in Table 4.2, we see a dramatic change from $\beta_{SUV} = 0.5$ to $\beta_{SUV} \geq 1.0$. Firstly, the modularity value jumps from 0.57% to 0.71, 0.79 and 0.80, which are very significant for the measure. Secondly, the number of large communities also increases from 2 to more than 5, which improves the community resolution and potentially reveals much more information. Moreover, unlike other parameter sets, the community distribution of $(\beta_{SUV}, \beta_{LV}) = (2.0, 0.1)$ and $(2.5, 0.1)$ show good agreements, which is a strong sign that the community structure is not an artifact. As a result, we will choose the communities extracted from the network with $(\beta_{SUV}, \beta_{LV}) = (2.0, 0.1)$ as the basis of our user groups. The clustering result is shown in Figure 4.3. But why the logistic weight model performs so well? We show in next section that the model identifies informative users and exploits them to achieve better results.

4.3 The logistic weight model

When constructing the network, we find that not every user has the same influence on the community structure. In particular, the user with higher edge weights would have a stronger influences because he/she can act as a core to bind other users together. Therefore, how to choose the user to receive higher weights becomes the key problem.

From Figure 4.3, we see how the logistic weight model and the Jaccard similarity measure deal with this problem. The Jaccard similarity measure is based on the idea of eliminating bias and it does this well on the data set. We see from the figure that users with more than 700 votes all have fair chances of receiving large weights.

On the contrary, the logistic weight model specifically focus on users with the number of votes lies in small range, in this case, from 250 to 1,250. The intuition is that the users with intermediate number of votes should receive higher weights
Figure 4.1: The sum of edge weights for an user with certain number of votes. The logistic weight model specifically gives users with intermediate number of votes higher weights, suggesting users with these number of votes are more informative for the mining purpose. On the other hand, the Jaccard similarity measure treats users with different number of votes roughly the same.

because they are both active and selective, which gives us the best information to noise ratio. This interesting fact suggests that sometimes, in addition to balancing out the extreme values, we have to intentionally suppress them. We thought the observation is valuable because extreme values are common in the real world networks, where the power-law distribution is a hallmark feature.
<table>
<thead>
<tr>
<th>$(\beta_{SUV}, \beta_{LV})$</th>
<th>(0.1, 0.1)</th>
<th>(0.1, 0.2)</th>
<th>(0.1, 0.3)</th>
<th>(0.1, 0.4)</th>
</tr>
</thead>
<tbody>
<tr>
<td>NSD</td>
<td>0.76</td>
<td>0.58</td>
<td>0.67</td>
<td>0.76</td>
</tr>
<tr>
<td>ED</td>
<td>2.82</td>
<td>3.81</td>
<td>4.28</td>
<td>4.83</td>
</tr>
<tr>
<td>MOD</td>
<td>0.41</td>
<td>0.56</td>
<td>0.57</td>
<td>0.55</td>
</tr>
<tr>
<td>$\omega_{th}$</td>
<td>5</td>
<td>5</td>
<td>5</td>
<td>5</td>
</tr>
<tr>
<td>comp. size</td>
<td>5,897</td>
<td>4,193</td>
<td>3,377</td>
<td>2,802</td>
</tr>
<tr>
<td>1st size</td>
<td>2,621</td>
<td>1,518</td>
<td>1117</td>
<td>841</td>
</tr>
<tr>
<td>2nd size</td>
<td>2,607</td>
<td>1,207</td>
<td>880</td>
<td>703</td>
</tr>
<tr>
<td>3rd size</td>
<td>305</td>
<td>1,026</td>
<td>819</td>
<td>671</td>
</tr>
<tr>
<td>4th size</td>
<td>92</td>
<td>78</td>
<td>89</td>
<td>118</td>
</tr>
<tr>
<td>5th size</td>
<td>48</td>
<td>46</td>
<td>47</td>
<td>47</td>
</tr>
<tr>
<td>6th size</td>
<td>42</td>
<td>32</td>
<td>39</td>
<td>39</td>
</tr>
<tr>
<td>7th size</td>
<td>32</td>
<td>18</td>
<td>37</td>
<td>37</td>
</tr>
</tbody>
</table>

Table 4.1: The metrics of the constructed networks with $\beta_{SUV} = 0.1$ and different $\beta_{LV}$. NSD is the normalized standard deviation. ED is the effective diameter. MOD is the modularity measure. The definitions of the above metrics are defined in Section 4.1. comp. size is the largest component size of the network. $\omega_{th}$ is the lower bound for the edge weights. $n$-th is the n-th largest community size output by the clustering algorithm.
<table>
<thead>
<tr>
<th>$(\beta_{SUV}, \beta_{LV})$</th>
<th>(0.5, 0.1)</th>
<th>(1.0, 0.1)</th>
<th>(1.5, 0.1)</th>
<th>(2.0, 0.1)</th>
<th>(2.5, 0.1)</th>
<th>(3.0, 0.1)</th>
</tr>
</thead>
<tbody>
<tr>
<td>NSD</td>
<td>0.64</td>
<td>0.88</td>
<td>1.00</td>
<td>1.03</td>
<td>1.03</td>
<td>1.04</td>
</tr>
<tr>
<td>ED</td>
<td>3.58</td>
<td>4.84</td>
<td>5.73</td>
<td>5.87</td>
<td>5.88</td>
<td>5.88</td>
</tr>
<tr>
<td>MOD</td>
<td>0.57</td>
<td>0.71</td>
<td>0.77</td>
<td>0.79</td>
<td>0.80</td>
<td>0.80</td>
</tr>
<tr>
<td>$\omega_{th}$</td>
<td>3</td>
<td>3</td>
<td>3</td>
<td>3</td>
<td>3</td>
<td>3</td>
</tr>
<tr>
<td>comp. size</td>
<td>5,479</td>
<td>5,005</td>
<td>4,490</td>
<td>4,193</td>
<td>4,032</td>
<td>3,953</td>
</tr>
<tr>
<td>1st size</td>
<td>1,993</td>
<td>1,317</td>
<td>1,021</td>
<td>873</td>
<td>850</td>
<td>802</td>
</tr>
<tr>
<td>2nd size</td>
<td>1,955</td>
<td>1,049</td>
<td>871</td>
<td>841</td>
<td>804</td>
<td>653</td>
</tr>
<tr>
<td>3rd size</td>
<td>1,185</td>
<td>994</td>
<td>748</td>
<td>711</td>
<td>657</td>
<td>630</td>
</tr>
<tr>
<td>4th size</td>
<td>50</td>
<td>845</td>
<td>507</td>
<td>381</td>
<td>377</td>
<td>369</td>
</tr>
<tr>
<td>5th size</td>
<td>33</td>
<td>212</td>
<td>434</td>
<td>336</td>
<td>313</td>
<td>307</td>
</tr>
<tr>
<td>6th size</td>
<td>32</td>
<td>133</td>
<td>289</td>
<td>286</td>
<td>270</td>
<td>293</td>
</tr>
<tr>
<td>7th size</td>
<td>17</td>
<td>62</td>
<td>64</td>
<td>105</td>
<td>112</td>
<td>261</td>
</tr>
</tbody>
</table>

Table 4.2: The metrics of the constructed networks with $\beta_{LV} = 0.1$ and different $\beta_{SUV}$. NSD is the normalized standard deviation. ED is the effective diameter. MOD is the modularity measure. The definitions of the above metrics are defined in Section 4.1. comp. size is the largest component size of the network. $\omega_{th}$ is the lower bound for the edge weights. $n$-th is the n-th largest community size output by the clustering algorithm.
Figure 4.2: Clustering result on the network constructed with logistic weighted model with \((\beta_{SUV}, \beta_{LV}) = (2.0, 0.1)\). There are 6 non-trivial communities with sizes of 873, 841, 711, 381, 336 and 286.
CHAPTER 5

Discussion

5.1 Network representation

Network structure is a great way of representing relational data. However, defining network structure from the data deserves more attentions. Our work shows that the graph mining results are dramatically different with different network representations of the same data and some of the representations are clearly better.

We can understand the "right" network representation in two ways. The first one is to construct the network that better models the data. For example, in the World-Wide Web, inter-domain hyper-links are more important than the intra-domain links for the crawling purpose. Therefore, for such a purpose, an unweighted Web graph is not a good representation. The other way of thinking the right network representation is from the graph mining perspective. Graph mining methods, such as graph clustering, have their practical limitations. For example, empirically, the graph clustering algorithm works rather poorly on the dense graphs and can produce artifacts for certain degree sequences [5, 6]. However, because theoretical analysis is hard to perform for the graph related algorithms, finding the right network representation for a certain purpose seems to rely more on the experience. We thought the empirical studies in this regard would be very helpful.
5.2 Mining user behaviors with indirect information

We consider our approach of transforming the bipartite network representation to uni-partite network representation an useful technique for mining user behaviors. In most of the business, users do not directly interact with each others, rather, they interact with the provided services and products. This kind of relation can naturally be modeled by the bipartite network. Our approach bridges the bipartite network to uni-partite network for the clustering purpose, which is also a common preprocessing step for many mining techniques. In addition, the model also identifies the kind of user behavior that is more informative for the analysis purpose. This information can be adopted to identify credible users or, on the contrary, detect the spammers. However, the model has a main drawback of the lack of an efficient optimization method. The grid search method works for our simple model, but does not scale with more parameters with non-trivial interactions. Therefore the future work would be devoted to combine the proposed metrics to find an effective objective function for the model.
REFERENCES


