Magnet Community Identification on Social Networks

Guan Wang [†] Yuchen Zhao[†] Xiaoxiao Shi [†] Philip S. Yu[†*]
[†]University of Illinois at Chicago, USA [*]King Abdulaziz University, Saudi Arabia
{gwang26, yzhao24, xshi9, psyu}@uic.edu

ABSTRACT

Social communities connect people of similar interests together and play essential roles in social network applications. Examples of such communities include people who like the same objects on Facebook, follow common subjects on Twitter, or join similar groups on LinkedIn. Among communities, we notice that some of them are *magnetic* to people. A *magnet community* is such a community that attracts significantly more people's interests and attentions than other communities of similar topics. With the explosive number of self-formed communities in social networks, one important demand is to identify magnet communities for users. This can not only track attractive communities, but also help improve user experiences and increase their engagements, e.g., the login frequencies and user-generated-content qualities. In this paper, we initiate the study of magnet community identification problem. First we observe several properties of magnet communities, such as attention flow, attention qualify, and attention persistence. Second, we formalize these properties with the combination of community feature extraction into a graph ranking formulation based on constraint quadratic programming. In details, we treat communities of a network as super nodes, and their interactions as links among those super nodes. Therefore, a network of communities is defined. We extract community's magnet features from heterogeneous sources, i.e., a community's standalone features and its dependency features with other communities. A graph ranking model is formulated given these features. Furthermore, we define constraints reflecting communities' magnet properties to regularize the model. We demonstrate the effectiveness of our framework on real world social network data.

Categories and Subject Descriptors

H.2.8 [**Database Management**]: Database Applications–Data Mining

Keywords

Magnet community, social networks

1. INTRODUCTION

Community is an important building-block of social networks. It attracts people with common backgrounds, goals or interests, and it is the key element of social media's dominating success. For instance, college-level social community leads to the success of social-media giant Facebook; company-level social community initiates the popularity of professional network LinkedIn; business-level social community boosts the development of peer-to-peer business such as eBay. Moreover, in academic field, countless communities are formed to improve the academic society, such as the special interest group (SIG) for computer scientists, the Society for Industrial and Applied Mathematics (SIAM), etc.

Among all social communities, we notice that some of them attract people's interests more than the others, and we call them *magnet communities*. More specifically, *magnet communities* are such communities that draw significantly more attention than others even if they are all about the same topic. Examples of magnet communities include the magnet colleges with over 20,000 talented applicants each year (e.g., Stanford, MIT, etc), magnet research communities with over 1000 high quality paper submissions (e.g., KDD), and magnet companies professionals would love to work for. Communities draw people's attention from each other and form an interesting network. For example, Figure 1 depicts such a network where the employees migrate among some prestige IT companies/communities [1]. According to this statistics, every time there is one person joining Google from Facebook, there are 15.5 people joining Facebook from Google. Though we are not discussing which company is "*better*" from the figure, we notice that Facebook is one of the *magnet* communities that people prefer to join.

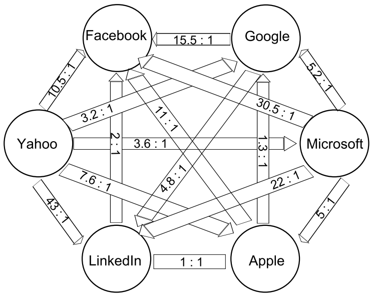


Figure 1: Community Interactions

Identifying magnet communities in a specific domain is an important task. First, the magnet communities help people understand the trends of their domains, e.g., which topics are hot, what

KDD'12, August 12–16, 2012, Beijing, China.
Copyright 2012 ACM 978-1-4503-1462-6 /12/08 ...$15.00.

[1] The information is from [4]

kinds of new interests people are developing, and which types of new communities are forming. Second, they help people make decisions when joining communities. For example, irrelevant to communities' sizes or histories, people may want to join magnet ones that represent energetic present and promising future. These two functions of magnet community identification can further serve as building blocks of many other applications, e.g., community recommendation, etc. Despite its importance, magnet community identification problem has not been studied before, to the best of our knowledge. Researchers have focused on community detection in various scenarios. Given a network, their goal is to detect subnetworks of similar nodes as communities. Our goal is one step further to identify magnet communities in a given domain through quantitative analysis. More specifically, given communities in a domain, we want to rank them based on their attractiveness to people among the communities of that domain. In the end, the top ranked communities are the ones people tend to adhere to.

To solve this problem, a quick path to go down is the classic random walk scheme. PageRank or its variations seems to be the simple and right choice. However, we demonstrate their incompetence on this problem using Table 1, where we list 3 ranking results based on different mechanisms. The first column lists the companies with highest PageRank scores among 6853 IT companies in the employee migration network that we have. Interestingly enough, they are also the 5 largest companies in terms of employee numbers in our dataset. Therefore, standard PageRank skews heavily towards the size of a community. What about we simply normalize the migration flow by community size? The second column of Table 1 lists the results after normalization. The top ranked companies become tiny start-ups with about 100 employees. The third column are the companies recognized as "ideal employers" according to a survey result from Universumglobal [2]. It aligns better with people's common sense about IT industry. As we have seen, the classic random walk schemes cannot accurately measure the attractiveness of communities. Although the survey result can create better performance, it takes a lot of efforts and manual works, which becomes infeasible for large-scale identification tasks.

Rank	PageRank	Normalized PR	Survey Result
1	Hewlett Packard	Zuora	Google
2	IBM	Silver Peak Systems	Microsoft
3	Oracle	Kony Solutions	Apple
4	Microsoft	Palo Alto Networks	Facebook
5	Cisco Systems	Quickoffice	IBM

Table 1: Top Ranked IT Companies

Therefore, the magnet community identification task is more challenging than it appears to be. First, there is no single criterion that we could rely on to determine the attractiveness of a community. For example, we cannot blindly say that a larger community is magnetic while a smaller one is not. Therefore, the first challenge is how to extract features from these heterogeneous sources of impacting factors of a community's attractiveness. The second challenge is how to naturally combine all heterogeneous information into a unified ranking model that also depicts attractive properties. For example, some features are only related to each individual community, e.g., size or specific topic, which have little to do with other communities; while others depend on other communities, e.g., people's preferences. The third challenge is the noise handling. People may have different tastes about communities. It is possible that a few people may prefer the less attractive communities. However, we are focusing on the major trend.

We attack the problem by first identify some common properties of magnet communities and design appropriate solution accordingly. First, we notice that a community's attractiveness associates with its *attention flow*. A community is attractive when it draws more people's attentions than losing attentions to other communities. In other words, the in-flow of people's attentions could be larger than out-flow for magnet communities. Second, a community's attractiveness associates with *attention quality*. It goes higher if it could draw people's attention from other magnet communities. Therefore, magnet communities could not only attract people, but also attract them from other attractive communities. Third, a community's attractiveness also associates with the *persistence* of people's attention. Being able to have the first two properties for only a short period does not make a real magnet community. When a community can persistently sustain the two properties, it becomes a magnet one. With these observations, we formulate the magnet community identification problem as a constrained graph ranking problem with heterogeneous information from nodes and edges, where constraints represent the relations between these features and magnet properties we just discussed. The whole model becomes an optimization framework of a synthesis measure of attractiveness of communities, which we will go into details later.

We summarize our contributions in three folds:

- We propose a new direction on social network analysis, namely magnet community identification. We believe that the work towards this direction is of huge demands of social network applications, since it can improve user experiences, increase their engagements to social network platforms, and enrich application features on these platforms.
- We propose one definition of magnet communities by identifying their properties. We develop a framework for large community graphs, which models both community features and the attractiveness properties together to identify magnet communities that reflect people's preferences among them.
- We demonstrate the effectiveness of our framework on a particular domain of magnet community identification, namely company's employee magnet community identification. As a tip of the iceberg of general magnet community identification, this domain specific example could serve as an interesting topic by itself, which further proves the wide range of applications magnet community identification may have.

This paper is organized as follows. In the next section, we formally define the magnet community identification problem and propose our framework in details. In section 3, we show our experiment results on real world social network data. In section 4, we explain the relationship between our work and previous related ones. Section 5 is the conclusion.

2. MAGNET COMMUNITY IDENTIFICATION FRAMEWORK

Starting from a bird's-eye view of the magnet community identification framework, we will go deeply through the formulation of our model in this section.

We first represent a social network as a graph $G(V, E, f_V, f_E)$, where V and E are vectors of nodes and edges, and f_V and f_E are features of nodes and edges respectively. To deal with the large scale nature of social networks, we define a summarization function $\mathscr{L} : G \rightarrow \mathcal{G}_C$ that maps an original graph to its condensed graph $\mathcal{G}_C(\mathcal{V}, \mathcal{E}, \mathcal{F}_\mathcal{V}, \mathcal{F}_\mathcal{E})$, where $\mathcal{V} = \{\mathcal{V}_1, \mathcal{V}_2, ..., \mathcal{V}_k\}$ ($\forall i, \mathcal{V}_i \subset V$) and $\mathcal{E} = \{\mathcal{E}_1, \mathcal{E}_2, ..., \mathcal{E}_n\}$ ($\forall i, \mathcal{E}_i \subset E$) are subsets of nodes and edges

[2] www.universumglobal.com/Top50

of original graph. Moreover, $\mathcal{F}_{\mathcal{V}}$ and $\mathcal{F}_{\mathcal{E}}$ are aggregated nodes' and edges' features in the condensed graph. For the sake of modeling simplicity, we further require $\forall i, j, \mathcal{V}_i \cap \mathcal{V}_j = \emptyset$. Therefore, every element in $\mathcal{V}$ is a community, and every element in $\mathcal{E}$ is a hyper-connection between two communities (a set of edges across the two communities in original graph). Such connections could be defined in various ways, e.g., their domain relation, or people's preference on each other. Figure 2 illustrates such a compression process. Community detection algorithms could be applied in this process. However, it is not the emphasis of this paper. We directly focus on magnet community identification on $\mathcal{G}_{\mathcal{C}}$.

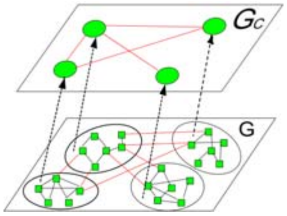


Figure 2: Graph Compression

2.1 Basic definitions

Generally speaking, magnet communities are the ones that people prefer to pay attention to among all social communities. In order to quantitatively identify magnet communities, we first give some more precise definitions related to them, from which we formally depict magnet communities later on in our model.

DEFINITION 2.1. *(Attention to a community) People's attention to a community is a generic representation of their selectively concentration on the community while ignoring others.*

In other words, we assume one person only pays attention to one community at a given time. The person's attention may change from one community to another from time to time, but one at a time. Paying attention has generic meaning, e.g., working for a particular company, or submitting paper to a special interest group. When we mention that a community is observed to attract people's attention, we assume that the people being attracted only pay attention to that community but not others at the time of observation.

DEFINITION 2.2. *(Attention migration flow among communities) Attention migration flow is the overall changes of people's attentions among communities, at the time of observation.*

When a person loses her interest in one community and starts to pay attention to another, there is an attention migration. For one community, its *in-flow* represents the total attentions drawn from other communities, and *out-flow* represents the total attentions lost to other communities.

DEFINITION 2.3. *(Magnet community) A community is a magnet community, i.e., with high attractiveness level, if it has the following properties.*

- *Attention flow: Its in-flow is larger than the out-flow.*
- *Attention quality: The in-flow comes from other communities with high attractiveness levels.*
- *Attention persistency: Its first two properties should be persistent.*

The first property ensures that the size of a magnet community should be growing, because it could attract more and more people. The second property tells the fact that a community's attractiveness should depend on other communities as well. It is more attractive if it could draw people's attention over other attractive ones. The third property, persistence, guarantees that we identify true magnet communities rather than some communities only with a momentarily attention burst. We also notice that the community size does not directly contribute to the attractiveness value. However, magnet communities usually have a certain size that is neither too small nor too large. On the one hand, the persistence property will ensure an accumulative effect of community growth to make itself become larger. On the other hand, when the community has grown into a large one, it is hard to keep in-flow larger than out-flow. That is the reason that very large communities tend to become less attractive due to the loss of people's attention. For example, one particular friendship community of a person may be more attractive when it contains certain number of friends sharing information, but when it grows too large and interesting information get diluted, it becomes less attractive.

2.2 Attractiveness computation framework

We use a vector $\mathcal{M} = (m_1, m_2, ..., m_k)$ to denote the set of estimated attractiveness levels for all communities. As we mentioned, attractiveness level of a community m_i is determined not only by heterogeneous features $\mathcal{F}_{\mathcal{V}}$ and $\mathcal{F}_{\mathcal{E}}$ from nodes and edges in the condensed community graph, but also by the attractiveness values of other communities in the same domain. Therefore, we have an abstract representation of $\mathcal{M}$ as

$$\mathcal{M} = f(\mathcal{F}_{\mathcal{V}}, \mathcal{F}_{\mathcal{E}}, \mathcal{M}) \tag{1}$$

Let $\mathcal{M}^*$ be the real attractiveness values. Our goal is to make the gap between the real values and the estimated ones as small as possible. Therefore, our objective function at high level is to minimize the Frobenius norm of the two, with the constraint φ to ensure that the final attractiveness levels for communities are consistent with their magnet properties.

$$min||\mathcal{M}^* - \mathcal{M}||_F^2 \tag{2}$$

$$s.t., \varphi(\mathcal{M}, \mathcal{F}_{\mathcal{V}}, \mathcal{F}_{\mathcal{E}}) \geq 0 \tag{3}$$

In the following subsections, we will concentrate on more details about feature extractions and concrete formula of our objective function and its constraint.

2.3 Attractiveness features

2.3.1 Standalone features

As the name indicates, a community's standalone features are those not depending on other communities. For example, the topic of a community is a standalone feature. For example, if a group on LinkedIn is about text mining, it is very unlikely that it changes that topic due to other groups' activities.

Suppose we have h standalone features $\Gamma = \{\gamma_1, ..., \gamma_h\}$, the score of standalone features of community i is a function of Γ_i. $f_i = g(\Gamma_i)$, where Γ_i is the values of the standalone features of community i. Therefore, $\mathcal{F}_{\mathcal{V}} = (f_1, f_2, ..., f_k)$ becomes the standalone feature vector for all the nodes in $\mathcal{G}_{\mathcal{C}}$.

2.3.2 Attention migrating matrix as dependency features

We call a community magnet because it can draw people's attention from other communities. Thus, a straightforward way to derive dependency features is to start from people's attention migration flow. People in one community may stay active in that community, or become inactive because some other communities draw their attention away, so that they become active in somewhere else. Therefore, communities draw people's attention among each other. This unique relation is modeled as an attention migrating matrix $D = (d_{ij})_{k*k}$, where d_{ij} is the actual number of people who *depart* from community i and *join* j. Departing from i means the person becomes inactive in community i. Respectively, joining j means the person becomes active in j. Note that what we assume here is that every person could only be active in one community at one time. Let vector $A = (a_i)_{k*1} = D \cdot e$ be the attention vector, where e is a k-*by*-1 unit vector. Thus, a_i is the total number of people who depart from community i. Let $\mathcal{A}$ be the element-wise inverted vector of A, where $\mathcal{A} = (a_i^{-1})_{k*1}$. We have dependency features of communities as $\mathcal{F}_{\mathcal{E}} = \mathcal{A} \circ D^T$, which is the Hadamard product of $\mathcal{A}$ and D.

The dependency matrix, or edge feature, $\mathcal{F}_{\mathcal{E}}$ is a probabilistic transitional matrix [3]. Each column of $\mathcal{F}_{\mathcal{E}}$ is the distribution of people whose attentions are migrating to other communities.

2.4 Concrete formula of magnet community ranking framework

One way to model the attractiveness of a community, i.e., a node in $\mathcal{G}_{\mathcal{C}}$ is to use random walk with restart. A node's attractiveness value depends on the probability of it being visited from other nodes. In other words, it is the probability that people's attention would arrive from other communities. Upon combining heterogeneous features about node and edge on $\mathcal{G}_{\mathcal{C}}$, we have

$$\mathcal{M} = \alpha \mathcal{F}_{\mathcal{E}} \mathcal{M} + (1-\alpha)\mathcal{F}_{\mathcal{V}}, 0 \leq \alpha \leq 1 \tag{4}$$

where α is a weighting parameter. With that formula, we can rewrite the objective function as

$$\begin{aligned} & min\ ||\mathcal{M}^* - \mathcal{M}||_F^2 && (5)\\ = & \ min\ ||\alpha \mathcal{F}_{\mathcal{E}} \mathcal{M} + (1-\alpha)\mathcal{F}_{\mathcal{V}} - \mathcal{M}||_F^2 && (6)\\ = & \ min\ ||(\alpha \mathcal{F}_{\mathcal{E}} - I)\mathcal{M} + (1-\alpha)\mathcal{F}_{\mathcal{V}}||_F^2 && (7)\end{aligned}$$

Now let us focus on the constraint for the above objective function. When we say one community is more magnetic than the other, at least one of the following two conditions are very likely to happen. First, this community has better standalone features. Second, it draws people's attention out of other similar communities. On the contrast, it is unlikely for a community to be more magnetic than others if it is inferior on both conditions. Formally, when i is more magnetic than j, i.e., $m_i - m_j > 0$, we want at least one of the following conditions hold.

- $f_i > f_j$
- $\frac{d_{ji}}{S_i} > \frac{d_{ij}}{S_j}$

The first condition is straightforward. It simply states that i's standalone feature should be higher than j's, if i has high attractiveness level than j. The second one needs more explanation. When people depart from community i and join j, we say that i has contribution to j. If community i is more attractive than j, j's contribution to i should be larger than i's contribution to j. The significance of this contribution depends on two factors: the number of migrated people from j and the size of i. Larger communities are harder to be contributed significantly than smaller ones with the same size of in-flow (see Figure 3). Therefore, we call the second condition *contribution imbalance*.

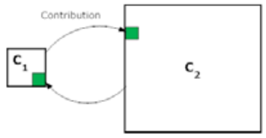


Figure 3: Contribution imbalance

Although it is possible for people to move from i to j if only one of the above conditions is true, it is very unlikely for them to prefer i over j if none of the two condition is true. Thus, we make our constraint as follows, where μ is a weighting parameter and ζ is a lower bound.

$$\sum_{(i,j)} (m_i - m_j) * (\mu(\frac{d_{ji}}{S_i} - \frac{d_{ij}}{S_j}) + (1-\mu)(f_i - f_j)) \geq \zeta \tag{8}$$

Therefore, the three properties of magnet communities are captured into Eq. 7 and Eq. 8 in a subtle way. Eq. 7 states that a community would have better chance to be a magnet one if it attracts attentions from other high magnet communities, which implies the second property. Eq. 8 constraints the magnet computation results must consistent with the first and third properties in Definition 2.3, which are reflected by the term $\frac{d_{ji}}{S_i} - \frac{d_{ij}}{S_j}$.

We rewrite the constraint after combining like terms as

$$\sum_{i=1}^{n} \left(\sum_{u \in n_{in}^i} \varphi_{iu} - \sum_{v \in n_{out}^i} \varphi_{vi} \right) m_i \geq \zeta \tag{9}$$

where n_{in}^i is node i's in-degree neighbors in $\mathcal{G}_{\mathcal{C}}$ and

$$\varphi_{iu} = \mu(\frac{d_{ui}}{S_i} - \frac{d_{iu}}{S_u}) + (1-\mu)(f_i - f_u)$$

n_{out}^i is node i's out-degree neighbors in $\mathcal{G}_{\mathcal{C}}$, and φ_{vi} has similar meaning with φ_{ui}.

We organize Eq. 9 in a more abstract form as

$$\Phi \mathcal{M} \geq \zeta \tag{10}$$

Here, $\mathcal{M}$ is the vector of $\{m_i\}_{1*n}$ and Φ is its coefficient vector.

Now we discuss how to solve the optimization framework.

THEOREM 1. *Our optimization framework is equivalent to the following canonical quadratic programming form:*

$$min \left(\mathcal{M}^T Q \mathcal{M} - 2u^T \mathcal{M} \right) \tag{11}$$

$$s.t., H\mathcal{M} \leq \xi \tag{12}$$

PROOF. The objective function of Eq. 7 can be rewritten as

$$||(\alpha \mathcal{F}_{\mathcal{E}} - I)\mathcal{M} + (1-\alpha)\mathcal{F}_{\mathcal{V}}||_F^2$$

$$= tr(\mathcal{M}^T(\alpha {\mathcal{F}_{\mathcal{E}}}^T - I)(\alpha \mathcal{F}_{\mathcal{E}} - I)\mathcal{M} + (1-\alpha)\mathcal{M}^T(\alpha {\mathcal{F}_{\mathcal{E}}}^T - I)\mathcal{F}_{\mathcal{V}}$$

$$+ (1-\alpha){\mathcal{F}_{\mathcal{V}}}^T(\alpha \mathcal{F}_{\mathcal{E}} - I)\mathcal{M} + (1-\alpha)^2 {\mathcal{F}_{\mathcal{V}}}^T \mathcal{F}_{\mathcal{V}})$$

Notice that the first three terms here are single number variables, and the fourth term is a constant. Therefore, above objective equation is equivalent to the form below:

$$(\mathcal{M}^T(\alpha {\mathcal{F}_{\mathcal{E}}}^T - I)(\alpha \mathcal{F}_{\mathcal{E}} - I)\mathcal{M} + 2(1-\alpha){\mathcal{F}_{\mathcal{V}}}^T(\alpha \mathcal{F}_{\mathcal{E}} - I)\mathcal{M}$$

[3] If two nodes are unreachable, we add a small probability for them to directly reach each other to make $\mathcal{F}_{\mathcal{E}}$ irreducible and aperiodic. (PageRank computation has a similar procedure [18])

Let $Q = (\alpha {\mathcal{F}_{\mathcal{E}}}^T - I)(\alpha \mathcal{F}_{\mathcal{E}} - I)$, $u = (1-\alpha)(\alpha {\mathcal{F}_{\mathcal{E}}}^T - I)\mathcal{F}_{\mathcal{V}}$, $H = -\Psi$, and $\xi = -\zeta$. We have our optimization framework reduced to the target form. □

Next, we explore the property of the solution of our optimization, i.e., whether it has feasible solution, or whether the global minimal exist, and the complexity of solving the optimization.

LEMMA 1. *Q in the optimization framework is positive definite, if $\alpha \neq 0$ and $\alpha \neq \frac{1}{\lambda}$, where $\lambda \neq 0$ is the eigenvalue of $\mathcal{F}_{\mathcal{E}}$.*

PROOF. Since $\mathcal{F}_{\mathcal{E}}$ is a stochastic matrix, $|\mathcal{F}_{\mathcal{E}}| \neq 0$. We have $\mathcal{F}_{\mathcal{E}} X = \lambda X$. Therefore, $(\alpha \mathcal{F}_{\mathcal{E}} - I)X = \alpha \mathcal{F}_{\mathcal{E}} X - X = (\alpha\lambda - 1)X$. We have the eigenvalue of $(\alpha \mathcal{F}_{\mathcal{E}} - I)$ is $\alpha\lambda - 1$. Thus, $|\alpha \mathcal{F}_{\mathcal{E}} - I| = \alpha\lambda - 1 \neq 0$. Since $Q = (\alpha \mathcal{F}_{\mathcal{E}} - I)^T(\alpha \mathcal{F}_{\mathcal{E}} - I)$, Q is positive definite. □

THEOREM 2. *Our objective function is strictly convex and it has a unique local minimum which is also a global minimum. Ellipsoid method can find global minimum of the objective function in polynomial time.*

PROOF. The convexity of objective function and the existence of global minimum are guaranteed by the positive definite Q, which also ensures there are polynomial time algorithms can find the minimum solution [7]. □

With the solution techniques, attractiveness computation framework for communities can be formally illustrated below.

Algorithm 1 Attractiveness Computation Framework

```
**Input:** The social network with node and edge features $G(V, E, f_V, f_E)$
**Output:** The magnet vector $\mathcal{M}$
Define the graph transformation function $\mathcal{L}$, generate condensed graph $\mathcal{G}_{\mathcal{C}} = \mathcal{L}(G)$
$\mathcal{F}_{\mathcal{V}} \leftarrow$ Standalone features extraction from $\mathcal{G}_{\mathcal{C}}$
$\mathcal{F}_{\mathcal{E}} \leftarrow$ Dependency features extraction from $\mathcal{G}_{\mathcal{C}}$
Constraint generation from graph $\mathcal{G}(\mathcal{V}, \mathcal{E}, \mathcal{F}_{\mathcal{V}}, \mathcal{F}_{\mathcal{E}})$
Solve the optimization problem using Eq. 7 and Eq. 10, based on $\mathcal{F}_{\mathcal{V}}$ and $\mathcal{F}_{\mathcal{E}}$
```

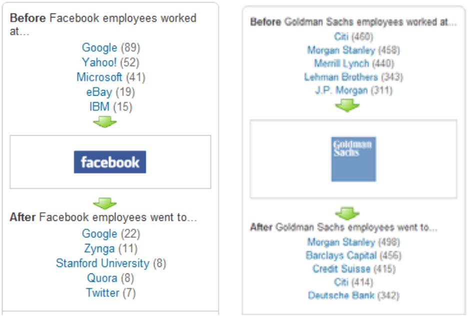
3. EVALUATION

In this section, we apply the magnet community identification framework to one real world application, namely magnet company identification, on a social network of professional relations in two industries. Magnet company identification is an interesting research topic by itself. As it serves as a good instance for our framework as well, we select this unique angle to show the effectiveness of our model. All experiments are conducted on machines with Intel Xeon^{TM} Quad-Core CPUs of 2.27 GHz and 24 GB RAM. We use the Matlab optimization toolbox as the solver for our framework.

3.1 Data collection and features extraction

3.1.1 Data collection

We crawl the company community data from the world's largest online professional network platform www.linkedin.com. It provides information about many aspects of companies, one of which is the employee flow (Figure 4). For most companies with certain size or reputation, LinkedIn lists up to top 5 companies of their employee sources and destinations. With this publicly available information, we can construct the raw departure matrix. LinkedIn also displays an estimated size of every company. Therefore, we can piece the information together to have our contribution matrix, which serves as the dependency features. Although LinkedIn only contains partial information about these companies (not all the employees have LinkedIn accounts), it is a close estimation of real world situation and a good sample of the ground truth data.



Before Facebook employees worked at...
Google (89)
Yahoo! (52)
Microsoft (41)
eBay (19)
IBM (15)

After Facebook employees went to...
Google (22)
Zynga (11)
Stanford University (8)
Quora (8)
Twitter (7)

(a) Facebook employee flow

Before Goldman Sachs employees worked at...
Citi (460)
Morgan Stanley (458)
Merrill Lynch (440)
Lehman Brothers (343)
J.P. Morgan (311)

After Goldman Sachs employees went to...
Morgan Stanley (498)
Barclays Capital (456)
Credit Suisse (415)
Citi (414)
Deutsche Bank (342)

(b) Goldman Sachs employee flow

Figure 4: Employee Migration Flows on Linkedin

For standalone features, we select a company's revenue per employee [4] (a commonly used factor to describe company efficiency [6]), industry, location, and age as its raw features. We believe the four factors contribute much to one's decision about joining a company.

Notice that we can directly consider the information we crawled the features on condensed graph $\mathcal{G}_{\mathcal{C}}$. In fact, in this application domain, the original graph G should be the whole professional network, including users features and their connections. Since that information is not publicly available, we could omit the graph compression process. However, we argue that the performance of our model may not be impacted significantly, because the publicly available information is already an aggregated representation of the hidden condensed graph we need. As a result, the different aspects of information about company communities we crawled can be plugged right in to the optimization framework. In total, we have 39527 companies' information in 142 industries.

3.1.2 Feature extraction

Our raw features for community standalone features in this case are a company's revenue-per-employee, industry, location, and founded year. They are categorical valued except for the first feature. Now we discuss how we translate them into numerical forms.

For each industry, we count how many people flow into it and out of it, using company level departure and arrival data. With that data normalized by industry size, we have the percentage growth of each industry. This percentage indicates the attractiveness of the industries. Table 2 is such an example. It lists growth of all industries related to IT and Finance. We use this growth rate as the industrial feature value.

For locations, we measure their popularities. If a location has more companies nearby, it is more popular. For example, Silicon Valley and New York City are two popular places for IT and finance industries, respectively. A company should be more attractive if it is at a more popular location. We use the number of companies of a location as its feature. We further normalize it to the range (0,1).

Founded year feature is treated similar to location features. For example, people in high-tech industry would like to work for companies that are not only mature enough to be stable, but also not founded too long ago to be active. Thus, we count the number of

[4] Since LinkedIn does not have this information, we crawl that from www.reuters.com/finance.

IT Industry	Growth	Financial Industry	Growth
hardware	3%	banking	1%
software	3%	insurance	0%
network	6%	finance	2%
internet	11%	real estate	0%
semiconductor	2%	investment banking	3%
telecomm.	-2%	investment mngt.	3%
electronics	0%	accounting	1%
information service	1%	venture capital	5%
online game	6%	commercial real estate	6%
information security	4%	capital markets	-1%

Table 2: Industry Growth

companies founded for each year. We use normalized value to represent year feature.

After the categorical-numerical translation, we use a linear combination of the values from the four features as the standalone feature. The dependency features are from the transition data we mentioned in data collection section.

3.2 Ranking performance

3.2.1 Baseline Description

We choose PageRank with one simple but important variation as our baseline. We replace its originally binary adjacent matrix with $\mathcal{F}_{\mathcal{E}}$, and its uniform distributed restart vector with $\mathcal{F}_{\mathcal{V}}$. We still call it PageRank for the ease of presentation. However, this variation captures more information than original PageRank and slightly improves its results, so it serves better as a baseline.

It makes more sense to compare companies in the same industries. Therefore, we pick two representative industries, IT and financial, to conduct our experiments. Although there are divisions within each meta industry (e.g., IT has segments of software, hardware, etc), they are comparable because they share many common properties and attract similar employees. Since our objective is to identify *magnet* companies, the desired result should be companies that have the most potential at current time, which will accelerate the new technology development, produce new ideas, and significantly change people's lives. They are not necessarily giant or well-established ones, but rather innovative and attractive to talents. We searched through media for such rankings as the ground truth. Unfortunately, despite many kinds of company rankings, there is no one directly based on the criteria we mentioned above. Take the arguably most recognized company ranking system *Fortune Magazine* as an example. The famous *Fortune 500* is for largest corporations, because it is based on size and revenue. Other simple rankings proposed by Fortune are based on straightforward criteria. For example, its "best companies to work for" is a series of rankings each of which is based on one of the conditions like *job growth, no layoffs, best for women, big pay*, etc. Out of our best effort, we found no single ranking that takes as many aspects of *attractiveness* level as we did. Nevertheless, we found two rankings that are related to our criteria. One such ranking is the 2011 *ideal employer ranking* proposed by *Universumglobal*while the other is the 2011 most admired company ranking by Fortune [5]. Both of them are based on survey results from professionals and university students. We will use them as our reference sets to compare the effectiveness of our results with PageRankresults. Before that, we would like to first demonstrate some interesting case studies. This provides intuitions and insights of the performance of our magnet community identification model, and why it identifies real magnet companies that attract talents.

3.2.2 Case studies

First we want to be clear that, in this study, we are not judging which companies are better. We only neutrally comment on the trend of the companies attractiveness on talents.

IT magnet companies general trend. Table 3 lists the top-10 ranked companies by PageRank and magnet identification model (MIM) results, along with the survey results from Universumglobal and Fortune about ideal employer and most admired companies, respectively. The magnet company list we identified contains both well-established companies, such as Google, Apple, and Microsoft, and energetic start-ups such as Facebook, Salesforce, and LinkedIn. As we know, the technology trend of IT industry is migrating from software to internet in the early 2000s, and from internet to mobile applications and social networks in recent years. The top magnet companies that we identified correctly reflect this trend. For example, software company Microsoft has become less magnetic than internet company Google; Apple is attractive because of the popularity of mobile applications; and social network company Facebook is catching up with other prestigious companies on attractiveness level even if it was just founded a few years ago. In contrast, PageRank prefers big companies because they have far more employee flows than small companies. The PageRank's score depends heavily on a company's size, which is why IBM and Hewlett Packard outrank Microsoft, and Microsoft outranks Google. Compact yet competitive companies like Apple do not earn the chance to get into top 10. These clearly counter people's general understanding of these IT companies and the industry trend. Moreover, in its top 10 list, PageRank does not identify small but promising companies.

We also notice that our result has some important differences with the two survey results, even if our results are already closer to them than with PageRank's result. In the survey results, well-established companies trend to have higher ranks, maybe because people are more familiar with them. For example, companies such as Cisco, Intel, and Texas Instruments will be more recognized due to their legacies. However, well-established companies do not necessarily correspond to magnet companies. Sometimes small or medium start-ups represent the direction of brand new trends which may not even be realized by majority of the population taking the survey. Therefore, these kinds of surveys could not capture this phenomenon. That is why MIM works better than them. For example, we are able to identify promising companies such as Facebook, Salesforce.com and LinkedIn in top 10, and Twitter, Zynga in top 25, where they are hardly recognized by the survey results.

Micro-level case study It is interesting to note that our model, in its top 10 list, is able to identify all the companies in Figure 1 that is generated from a separate source [4]. From that figure, we can see Yahoo! is contributing its talents to all others, and Facebook and LinkedIn are drawing talents from the rest of the companies. That phenomenon has been captured in our list. First, according to our result, Yahoo! is less magnetic than all the companies in the figure (except LinkedIn). Second, Facebook and LinkedIn already make themselves in the top 10 list, despite the fact that they are much younger and smaller than others. They have not outranked others due to other factors, since we are not only using talent traffic as it does in Figure 1.

Financial industry general trend. Table 4 lists the top 10 finance companies identified by PageRank and MIM results. J.P. Morgan

[5] money.cnn.com/magazines/fortune/mostadmired/2011/index.html

Rank	PageRank	**MIM**	Ideal Employer	Admired Company
1	IBM	**Google**	Google	Apple
2	Hewlett Packard	**Amazon.com**	Microsoft	Google
3	Oracle	**Apple**	Apple	Amazon.com
4	Microsoft	**Microsoft**	Facebook	IBM
5	Cisco Systems	**Facebook**	IBM	Qualcomm
6	Google	**Salesforce.com**	Electronics Arts	Intel
7	Tata Consult. Services	**Cisco Systems**	Amazon	Texas Instruments
8	Cognizant Tech. Solu.	**Juniper Networks**	Cisco Systems	Cisco Systems
9	Dell	**Yahoo!**	Intel	Adobe Systems
10	EMC	**Linkedin**	Sony	Oracle

Table 3: Top 10 IT Companies

Rank	PageRank	**MIM**	Ideal Employer	Admired Company
1	J.P. Morgan Chase	**J.P. Morgan Chase**	Goldman Sachs	US Bank
2	Citigroup	**Goldman Sachs**	J.P. Morgan Chase	Goldman Sachs
3	HSBC	**Morgan Stanley**	Boston Consult. Grp.	J.P. Morgan Chase
4	PWC	**Citigroup**	Deloitte	Merrill Lynch
5	Merrill Lynch	**Merrill Lynch**	Merrill Lynch	Northern Trust Corp.
6	Ernst & Young	**CB Richard Ellis**	Ernst & Young	Credit Suisse
7	Deutsche Bank	**Wells Fargo**	Morgan Stanley	CB Richard Eills
8	Credit Suisse	**PWC**	PWC	HSBC
9	Barclays Capital	**Jones Lang LaSalle**	American Express	Barclays
10	Goldman Sachs	**Blackrock**	Bain & Company	Jones Lang LaSalle

Table 4: Top 10 Finance Companies

Chase and Goldman Sachs are relatively unscathed by the recent financial crisis. That means the two companies stay attractive to talents because of their stabilities and capital power. Our model correctly identifies them as the top two. Goldman Sachs is widely recognized as one of the best financial companies, but PageRank fails to give it a proper rank. That's a clear advantage of our method in terms of magnet community identification.

We also have good diversity in the list in terms of segments and sizes. For example, we have leading companies of banks, real estates, financial services, and accounting. CB Richard Ellis and Jones Lang LaSalle are the well-known top 2 leaders in real estate segment. Blackrock is the leader in assess management segment. PWC (PriceWaterhouseCooper) is known as one of the "big four" accounting firms. Unlike IT industry where small start-ups may attract talents by their innovations, financial companies may be attractive due to their capitalization and customer relations, which makes smaller firms harder to stand out as magnet ones. Despite that fact, we also discover some smaller boutique ones in addition to big firms, such as Blackrock. Although it is a much younger and smaller company than others in the list, it has grown to be the world's largest asset manager and one of the leading investment management companies. PageRank's performance is similar to that of IT industry. It heavily prefers big cooperations, e.g., large banks, while ignoring the top companies in segments other than banking. Since big firms tend to be magnet ones in financial industry, PageRank's performance becomes better in financial industry compared to IT industry. We observe that in financial industry, the most admired companies and ideal employers survey results are not as consistent as that of IT industry. The reason for this phenomenon may be that, despite they have some basic functionalities in common, financial companies are more specialized than IT companies, which makes the attraction criteria vary more than that of IT.

3.2.3 Overall Correctness measures

We use discounted cumulative gain (DCG [11]) to evaluate the ranking quality. It emphasizes the correctness of high ranked entities, which is the major goal of many information retrieval tasks. Figure 5(a) and (b) compare the marginal gain on DCG values at different cut-off positions on both ideal employer and admired company data in *IT industry*. As it shows, MIM outperforms PageRank significantly, because it identifies much more top-ranked companies correctly in the beginning. In the rank list generated by PageRank, the top-ranked companies are not the top in both datasets.

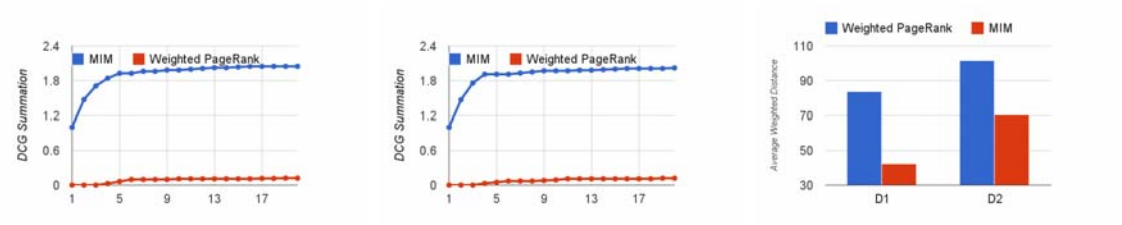
On aggregated level, we define a *weighted pairwise distance* measure of retrieved list L_1 and relevant list L_2, which emphasizes more on ranking order and correctness. $wDist(L_1, L_2) = \sum_{a_i \in L_2} (|L_2| - i + 1)(|i - \Lambda(a_i)|)$, where $\Lambda(a_i)$ is the rank of a_i in L_1. Based on this definition, if an algorithm misplaces top-ranked entities, it will receive more penalties than another algorithm which misplaces low-ranked entities. Hence the smaller the value, the better the performance, Figure 5(c) shows MIM result has smaller weighted distance than PageRank result.

Similar to IT industry, we also examine the performance of MIM on financial industry. Figure 6(a) and (b) show its marginal DCG gain results comparing with PageRank. The performance is similar to IT industry, where MIM dominates PageRank performance in high-ranked company set.

Figure 6(c) shows average weighted distance results. We notice that MIM does not outperforms as much as it does in IT industry data. We suspect two reasons for this phenomenon. First, the two benchmark data, i.e., survey results, are based on ad hoc criteria and not as diverse as MIM result. They always biased towards large corporations in banking and accounting. Moreover, some of their results are not consistent with people's usual understanding of the industry, e.g., US Bank ranks higher than any other prestigious financial firms. If the benchmark needs calibration, the comparison results may be affected. Second, in financial industry, well-established ones tend to be magnetic automatically. In that sense, PageRank also does well because it identifies large companies. Even so, MIM still outperforms PageRank by a wide margin in terms of average weighted distance. This comparison between IT industry and financial industry clarifies that, when large communities are not necessarily the magnet targets, MIM is the ideal tool to discover magnet communities.

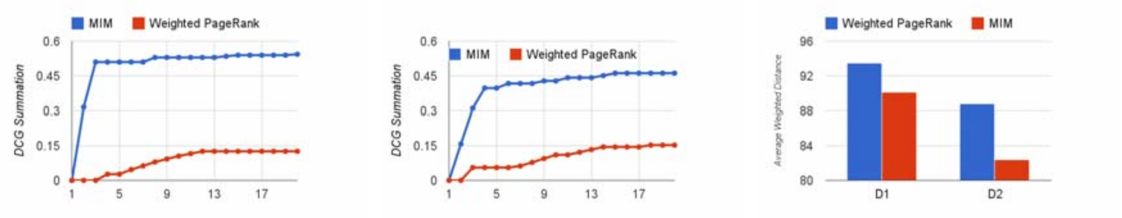
3.2.4 Parameter sensitivity

According to Eq. 7 and Eq. 8, the MIM algorithm has two parameters α and μ. Now we study how sensitive our model is to



(a) DCG on IT Ideal Employers (b) DCG on IT Admired Companies (c) Average Weighted Distance on IT

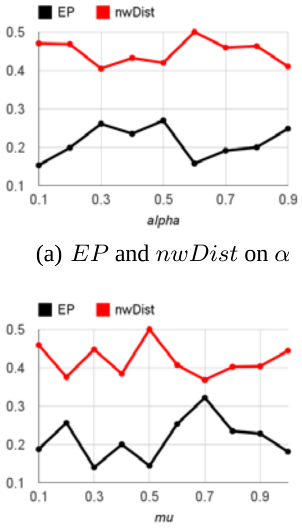
Figure 5: Performance on IT Industry



(a) DCG on Finance Ideal Employers (b) DCG on Finance Admired Corp. (c) Avg Weighted Dist. on Finance

Figure 6: Performance on Finance Industry

the two parameters. Due to space limitation, we only show the results on IT industry data and using admired company list as comparison. (Financial industry data give similar results.) Since we care more on high ranked entities, in addition to weighted distance, we also measure the model's performance on average precision $EP = \sum_{k=1}^{n} P(k)\Delta R(k)$, where $P(k)$ is the precision at cut-off k and $\Delta R(k)$ is the change of recall from position $k-1$ to k. We further normalize $wDist$ using $nwDist = \frac{1}{Z}wDist$, where Z is a normalization factor to make it align in the same scale as EP. In Figure 7 we can see that our model performs consistently on different α values. The fluctuations are in a small range. We also observe that the best performances are achieved at $\alpha = 0.6$.



(a) EP and $nwDist$ on α

(b) EP and $nwDist$ on μ

Figure 7: Sensitivity Evaluation

Figure 7 also shows that μ has similar effects as α. Though the performance varies on different μ values, they are bounded by a small range. The highest average precision and smallest distance are achieved simultaneously at $\mu = 0.5$. Therefore, we assign 0.6 and 0.5 to α and μ to generate results of IT industry, and 0.2 and 0.7 to them for the same reason for financial industry in case studies.

4. RELATED WORK

Network community analysis has been an active research field for a long time. However, previous works mainly focus on static or dynamic community detection, or most recently community evolution. To our best knowledge, there is no previous work directly related to *magnet community* identification.

Initially, people paid great attention on community detection via the structural properties of communities within a network, e.g., connection densities, etc [15]. While majority of community detection has been done on static networks, new topics and methods addressing dynamic networks are proposed. Rosvall, et.al [19] used random walks to identify the structure of communities. Their work could also rank communities by their structural relations. However, their method also prefers large communities. Aggarwal, et.al. [3] captured the changing phenomena of groups of nodes in a network. They intent to track the gradual changes of one given community. Sun et.al. [20] identified communities that are locally static. The work in [21] identified communities with social interaction and their dynamics based an optimization model that minimizes social cost within communities. The work in [16] also tried to discover dynamic communities, such as growing or birthing communities from hidden member relations. Other studies focused on the discovery of inter-community structure using learning algorithms based on Markov process, which maximize the likelihood of interaction data [17, 23].

The goals of static and dynamic community detections are similar, which is to identify communities. They do not go beyond that to further analyze properties of identified communities, e.g., how magnet community is forming. Therefore, we are addressing a totally different problem where identifying community is only the starting point of our framework.

Another somewhat related direction is to discover the trend or properties related to community members. Falkowski et. al, tried to answer the question of how inactively participating a community would represent a user's interests [9]. It focused on the fluctuation of members on a single community and the insights of the member-community interplay. Although our work also reflects members' interests about communities, it is about their selective attention over multiple communities. Furthermore, we are not emphasizing on the growth or demise of one single community, but the magnet interactions among multiple communities. Another work stood on a single node's (or a small subset of nodes') point of view to analyze how critical events affect *one* dynamic network [5]. In our model, we take account of majority members' attentions rather than a small set of them to help determine the attractiveness levels of *many* communities. The authors in [13] studied the bursts of a short lived "community", or transient crowd, when a hot news emerges on a social network. Their work is mainly about detecting a special type of community of highly dynamic nature. In our case, the communities are not transformed that rapidly. Also, they are not considering anything related to *magnet communities*.

Regarding to our solution technique, constraint optimization formulation in link based graph data ranking has been proposed before with different intentions and design emphasis. The work in [8] constrains HITS algorithm [14] on user opinions about authorities. The work in [22] constrains PageRank algorithm on teleportation parameters to reflect personalization. The work in [1] learns parameters on transition matrix among nodes. The work in [2] tries to rank the whole set of nodes based on the known rankings of a small subset of them. The work in [10] proposes a more complete framework on utilizing meta data such as node and edge features and user preferences, which has never been completely considered in one work before. Our solution differs from the above on several points. First, our work is unsupervised. We don't have labels that indicate which communities are *magnetic*, or which one should absolutely has more attractiveness value than the other. Second, our constraint is derived on meta data, in addition to pairwise preferences. On the comparison with unsupervised approaches, e.g., [12, 18], our method considers more meta data such as node and edge features and magnet properties.

5. CONCLUSION

We introduced the magnet community identification problem. It is not only an important problem for social network research, but also a potential building-block for other analysis. We formulate the problem by graph ranking with heterogeneous information and constraints. Our model takes dependency and standalone features, as well as the magnet properties of communities to generate reasonable communities' attractiveness values. Tailored for this type of ranking, our model outperforms classical link based ranking algorithm, i.e., PageRank and its variation. This paper is only an initial step towards the new direction of research on social network communities. Possible future expansions include applying our framework to other heterogeneous network or analyzing business problems with company attractiveness as a starting point.

6. ACKNOWLEDGE

This work is supported in part by NSF through grants IIS-0905215, IIS-0914934, CNS-1115234, DBI-0960443 and OISE-1129076, US Department of Army through grant W911NF-12-1-0066, and Google Mobile 2014 Program. Thank Xiangnan Kong for providing computation resources.

7. REFERENCES

[1] A. Agarwal, S. Chakrabarti, and S. Agarwal. Learning to rank networked entites. In *KDD'06*.

[2] S. Agarwal. Ranking on graph data. In *ICML'06*.

[3] C. C. Aggarwal and P. S. Yu. Online analysis of community evolution in data streams. In *The 5th SIAM International Conference on Data Mining*, 2005.

[4] Alex, Andrew, and Courtney. http://blog.topprospect.com/2011/06/the-biggest-talent-losers-and-winners.

[5] S. Asur and S. Parthasarathy. A viewpoint-based approach for interaction graph analysis. In *KDD'09*.

[6] A. Barua, P. Konana, A. B. Whinston, and F. Yin. Driving e-business excellence. *MIT Sloan Management Review*, 2001.

[7] S. P. Boyd and L. Vandenberghe. Convex optimization. *Cambridge University Press*, 2004.

[8] H. Chang, D. Cohn, and A. K. McCallum. Learning to create customized authority lists. In *ICML'00*.

[9] T. Falkowski and Myra Spiliopoulou. Users in volatile communities: Studying active participation and community evolution. *Lecture Notes in Computer Science*, 4511.

[10] B. Gao, T. Liu, W. Wei, T. Wang, and H. Li. Semi-supervised ranking on very large graph with rich metadata. In *KDD'11*.

[11] K. Jarvelin and J. Kekalainen. Cumulated gain-based evaluation of ir techniques. 2002.

[12] T. Joachims. Optimizing search engines using clickthrough data. In *KDD'02*.

[13] K. Y. Kamath and J. Caverlee. Transient crowd discovery on the real-time social web. In *WSDM'11*.

[14] JM Kleinberg. Authoritative sources in a hyperlinked environment. In *Journal of the ACM (JACM)*, 1999.

[15] A. Lancichinetti and S. Fortunato. Community detection algorithms: A comparative analysis. 2009.

[16] J. Li, W.K. Cheung, J. Liu, and C.H. Li. On discovering community trends in social networks. In *WI-IAT'09*.

[17] Y. Lin, Y. Chi, S. Zhu, H. Sundaram, and B. L. Tseng. Facetnet: a framework for analyzing communities and their evolution in dynamic networks. In *WWW'08*.

[18] L. Page, S. Brin, R. Motwani, and T. Winograd. The pagerank citation ranking: Bringing order to the web. In *Technical Report, Stanford Digital Library Technologies Project*, 1998.

[19] Martin Rosvall and Carl T. Bergstrom. Maps of random walks on complex networks reveal community structure. *PNAS*, 2008.

[20] J. Sun, S. Papadimitriou, P.S. Yu, and C. Faloutsos. Graphscope: Parameter-free mining of large time-evolving graphs. In *KDD'07*, 2007.

[21] C. Tantipathananandh, T. B. Wolf, and D. Kempe. A framework for community identification in dynamic social networks. In *KDD'07*.

[22] A. C. Tsoi, G. Morini, F. Scarselli, M. Hagenbuchner, and M. Maggini. Adaptive ranking of web pages. In *WWW'03*.

[23] T. Yang, Y. Chi, S. Zhu, Y. Gong, and R. Jin. Detecting communities and their evolutions in dynamic social networks–a bayesian approach. *Mach Learn*.