This paper was presented as part of the Mini-Conference at IEEE INFOCOM 2011

"Who Are Your Friends?" — A Simple Mechanism that Achieves Perfect Network Formation

Felix Ming Fai Wong
Department of Electrical Engineering
Princeton University
Email: mwthree@princeton.edu

Peter Marbach
Department of Computer Science
University of Toronto
Email: marbach@cs.toronto.edu

***Abstract*—A fundamental challenge in peer-to-peer and online social networks is the design of a simple, distributed algorithm that allows users to discover, and connect to, peers who closely match their interests or preferences. In this paper, we consider an algorithm that is based on simple, local comparisons, and analyze it to provide insights into why similar peer discovery algorithms work well in practice. To do so, we use a mathematical framework to characterize the closeness of individual interests, and formally introduce the notion of a "perfect network formation" under the framework. Our analysis shows that the proposed algorithm indeeds achieves perfect network formation. Our analysis uses bounding techniques based on Chernoff bounds.**

I. Introduction

In this paper we consider the network formation problem in peer-to-peer and online social networks. In particular, we consider the situation where users would like to discover and connect to peers with similar interests. In peer-to-peer networks, users would like to connect to peers who are interested in (and are therefore more likely to have) similar content, as this will reduce query overhead and query delay [1]. In online social networks, users would like to become friends with those who have similar interests. From a practical perspective, we are interested in network algorithms that are simple and can easily be implemented in a distributed fashion.

The most simple distributed algorithm is to let users interact directly with each other to find peers with similar interests. We refer to this process as *neighborhood discovery*. In the context of a peer-to-peer content sharing application, this could be done by peers polling each other and comparing the content they have stored locally. In the neighborhood discovery process, if a user finds another user who has similar interests (e.g., another peer who stores content in which the user is interested), then they conclude that they are similar.

However, neighborhood discovery is generally *noisy* as it is based on sample ovservations that are available (see for example [2]), and two users who have similar interests may falsely conclude that they are dissimilar, or two users may conclude that they have similar interests even though they are dissimilar. Therefore, a fundamental question that arises in the context of network formation is whether it is possible to devise a mechanism to overcome the noise problem, leading to a "perfect network formation process" in the sense that: (a) it allows an individual to discover everyone who has similar interests; and (b) it makes no mistake of connecting two dissimilar users.

One simple and intuitive approach, often seen in the literature, is to introduce a "neighborhood refinement" step where users compare their *initial* neighborhood sets obtained from the neighborhood discovery step ("Who are your Friends?"). Then two users connect only if their initial sets have a sufficiently large overlap. Network formation algorithms that use a neighborhood refinement step have been applied to both peer-to-peer networks [2] and online social networks [3], and they have been shown to perform well.

The goal of this paper is to provide insights into why such simple mechanisms work well from a theoretical perspective. The main contributions of the paper are as follows: (a) we introduce the notion of "perfect network formation" as a formal performance measure to study network formation algorithms (this criteria is generic and is not limited to the network formation mechanisms that we study in this paper), and (b) we rigorously show that a network formation algorithm based on neighborhood refinement can achieve perfect network formation. Our analysis also provides new insights into how the threshold function in the neighborhood refinement step should be chosen. We illustrate how this insight can be applied to practical problems with a numerical case study based on a real dataset.

II. Related Work

The idea of using a "neighborhood discovery" process to identify other peers that have similar interests has been applied to peer-to-peer networks, where the underlying observation is that users with similar interests are likely to have content of common interest (also called *Interest-based locality*) [1]. In [1], the authors propose a mechanism such that a user in a peer-to-peer file-sharing system (e.g., Gnutella, eDonkey) queries a cached list of peers with similar interests before querying the whole network. They show empirically that this mechanism improves the performance of Gnutella both in terms of query traffic and query delay.

The mechanism in [1] only uses a neighborhood discovery step to find other peers with similar interests. In [2], the authors extend the mechanism of [1] by using a neighborhood refinement step. The authors show empirically that the resulting

978-1-4244-9921-2/11/$26.00 ©2011 Crown

mechanism improves performance in terms of query success rate.

Mechanisms that use a neighbourhood refinement step have also been used for the *link prediction problem* in social networks [4]: given a past snapshot of a social network, what links will be added in the next time step? In [4], it is shown that in scientific collaboration networks the probability of two scientists collaborating increases with the number of collaborators they have in common. This motivates neighborhood refinement for the use of link prediction. Surprisingly, simple proximity measures such as the number of common neighbors, and Adamic/Adar [6] (giving higher weighting to rarer common neighbors) outperform more sophisticated measures based on random walks and shortest paths [5]. While the studies performed in [4], [6], [5] are experimental, [7] presents a formal analysis on neighborhood refinement applied to the link prediction problem. The perfect network formation problem (that we consider in this paper) is different from the link prediction problem in the sense that we try to identify all users who have similar interests (without making any mistakes), whereas the link prediction problem focuses on identifying, for a set of given users, the N links that are most likely to be created at the next time step. This difference in the objective function makes the perfect network formation problem a considerably harder problem compared with the link prediction problem.

The model that we use to characterize similarities between users falls within the class of *latent space models* [8] which assume that each node (user) lies in a latent space with an unobserved position, and edge probabilities depend on node positions (more specifically, edges are conditionally independent given nodes' positions). To simplify the analysis, we consider a one-dimensional lattice as the latent space model. The extension to more general settings is future work.

Our work is also related to the study of planted partition model [9], where the set of nodes can be partitioned into a finite number of clusters such that nodes within the same clusters have exactly the same interests [10], [11]. Two nodes in the same cluster (or different clusters) are connected by an edge with probability p (or q) with $p > q$. Note that this is a different model from the latent space model that we consider in this paper. The analysis in this paper can be interpreted as extending the work in [10] to the latent space model. This extension requires a different set of mathematical tools: whereas the analysis in [10] relies on techniques from spectral analysis, here we use bounding techniques based on Chernoff bounds.

III. Model and Problem Statement

In this section we introduce the mathematical model uses for our analysis, and provide a formal description of the proposed algorithm. We use a latent space model [8] to characterize and measure the *closeness of the interests* of two users, as well as to characterize noise in the neighborhood discovery process.

User Distance Function: We begin with a model to characterize users' interests and the closeness of the interests between two users. To do this, we assume that the interests of a given user u can be characterized by a *feature vector* f_u that is an element of a metric space. The distance of the interests $d(u, v)$ between two users u and v is then given by $d(u, v) = d(f_u, f_v)$, where $d(f_u, f_v)$ is the distance between the corresponding feature vectors f_u and f_v in the metric space.

We consider a sequence of systems (networks) indexed by n, where $n = 1, 2, ...$, and the nth system has $2n+1$ users. In the following, we denote $\mathcal{N}(n)$ as the set of users in system n. To simplify the analysis, we assume that in each system the feature vectors of users are arranged on a ring as illustrated in Fig. 1, and that the distance $d(f_u, f_v)$ between the feature vectors f_u and f_v is given by the ring distance between f_u and f_v, which is the minimum number of hops between them as illustrated by Fig. 1 (left).

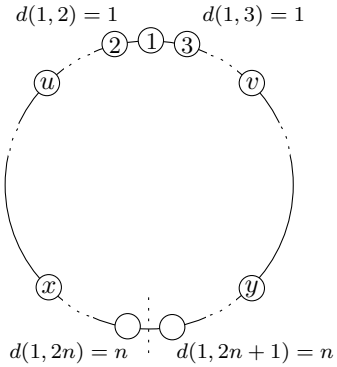


Fig. 1. Set $\mathcal{N}(n)$ and ring distance that is used as a metric.

Neighborhood Discovery We assume users cannot directly observe the distance $d(u, v)$, i.e., users have no direct knowledge of which others are close to them. Instead, users have to interact with others and use the information obtained to identify users with similar interests, i.e., those who are within a short ring distance. We refer to this process as *neighborhood discovery*.

As mentioned, the neighborhood discovery process is noisy and users who have similar interests might conclude during the neighborhood discovery process that they are dissimilar, and vice versa. We model this noise for system n with $2n+1$ users as follows. Consider two users u and v who are at a ring distance $d(u, v)$ from each other. User u concludes in the neighborhood discovery process that v has similar interests as itself (u connects to v) with probability $\Pr(u \to v)$, given by

$$\Pr(u \to v) = C_u \alpha(n)^{d(u,v)}, \tag{1}$$

where $\alpha(n)$ is such that $0 < \alpha(n) < 1$, and C_u is a constant depending on u such that $0 < C_u \leq 1$. This model captures the intuition that user u is more likely to connect to a user v with a smaller distance $d(u, v)$.

Furthermore, we assume that $\alpha(n)$ is of the form

$$\alpha(n) = e^{-2/\lambda(n)} \tag{2}$$

where $\lambda(n)$ is an increasing function with $\lim_{n\to\infty} \lambda(n) = \infty$.

Let S_u be the set of nodes to which user u connects during the neighborhood discovery process. In the following, we refer to S_u as the *initial neighborhood set*.

Under the above model, it can be shown that the expected size of the initial neighborhood set S_u satisfies $\mathbb{E}[\#S_u] = C_u\lambda(n)(1+o(1))$, i.e., $\lambda(n)$ captures how the size of the initial neighborhood set scales with n, and C_u captures the "relative socialness" of a user, i.e., how large the initial neighborhood set of user u is, relative to other users.

Neighborhood Refinement: As the neighborhood discovery process is probabilistic (see Eq. (1)), it is noisy in the sense that the initial neighborhood set S_u of user u may include a user v that is in fact at a large distance $d(u, v)$ from u. Similarly, the initial neighborhood set S_u of user u may not include some users that are very close to u. To overcome this problem, we consider the following *neighborhood refinement* mechanism.

Given the initial neighborhood sets obtained during the neighborhood discovery process, user u forms a final neighborhood set N_u as follows. Using a threshold parameter $k_{uv} > 0$, user u includes user v in its final neighborhood set N_u if and only if the overlap between the two sets S_u and S_v is no less than the threshold k_{uv}. More precisely, let $S_{u,v} = S_u \cap S_v$ be the intersection of the initial neighborhood sets S_u and S_v. User u then includes user v in its final neighborhood set N_u if the size $\#S_{u,v}$ is no less than the threshold k_{uv}, i.e., if we have $\#S_{u,v} \geq k_{uv}$. We also set $k_{uv}(n)$ to be a function of n.

Research Questions: There are two main research questions that arise for the above network formation algorithm consisting of the network discovery and refinement steps: (1) how should one choose the threshold parameters k_{uv}, and (b) is the network refinement step indeed able to overcome the noise of the neighborhood discovery process?

IV. Main Results

The goal of this paper is to study and characterize the effectiveness of the proposed algorithm. In particular, we would like to study whether the neighborhood refinement step can lead to perfect network formation despite the noise in the neighborhood discovery process. Ideally, perfect network formation should have the property that there exists a threshold value $\Delta(n)$ such that the final neighborhood N_u of user u obtained by the proposed algorithm is equal to the set of users whose interests lie within a distance $\Delta(n)$ of user u, i.e., we have $N_u = \{v \in \mathcal{N}(n) : d(f_u, f_v) \leq \Delta(n)\}$. However, this is too stringent a criterion (probably no algorithm can achieve it), and we define a slightly weaker notion instead.

We use the following notation to formally define the criteria for perfect network formation. Fix a user u. Let $\rho_u(n)$ be the accuracy ratio given by

$$\rho_u(n) = \frac{\#\{v \in N_u : d(f_u, f_v) \leq \Delta(n)\}}{\#\{v \in \mathcal{N}(n) : d(f_u, f_v) \leq \Delta(n)\}}, \tag{3}$$

i.e., $\rho_u(n)$ is the fraction of users within a distance of $\Delta(n)$ from u that are included in the final neighborhood set N_u, and let the false positive ratio $\tau_u(n)$ be given by

$$\tau_u(n) = \frac{\#\{v \in N_u : d(f_u, f_v) > \Delta(n)\}}{\#\{v \in \mathcal{N}(n) : d(f_u, f_v) \leq \Delta(n)\}}, \tag{4}$$

i.e., the relative size of the set of users in the final neighborhood set N_u that are further than $\Delta(n)$ away from u (and therefore should not be included in N_u).

Using the above notation, we then define perfect network formation as follows.

Definition 1. *We say the algorithm leads to* perfect network formation with resolution $\Delta(n)$ for user u *if there exist threshold functions $k_{uv}(n)$ such that $\rho_u(n)$ and $\tau_u(n)$ converges to* 1 *and* 0 *respectively in probability, i.e., for all $\xi > 0$,*

$$\lim_{n\to\infty} \Pr(|\rho_u(n) - 1| \geq \xi) = 0$$
$$\lim_{n\to\infty} \Pr(|\tau_u(n) - 0| \geq \xi) = 0$$

Note this definition captures the notion that perfect network formation should have the properties that it allows a user (a) to connect to all other users who have similar interests (i.e., at distance less than $\Delta(n)$) and we have

$$\lim_{n\to\infty} \Pr(|\rho_u(n) - 1| \geq \xi) = 0,$$

and (b) not to connect to any users who have dissimilar interests (i.e., at distance larger than $\Delta(n)$) and we have

$$\lim_{n\to\infty} \Pr(|\tau_u(n) - 0| \geq \xi) = 0.$$

The main result from our analysis is to derive sufficient conditions under which the network formation mechanism using a neighborhood refinement step leads to perfect network formation. We have the following result.

Theorem 1. *Fix any user u. If $\lambda(n)$ satisfies*

$$\lambda(n) = \omega(\log n) \text{ and } \lambda(n) = o(n/\log n),$$

then for all $\Delta(n)$ satisfying

$$\Delta(n) = \omega(\lambda^{\frac{3}{4}}(n)\log^{\frac{1}{4}} n) \text{ and } \Delta(n) = O(\lambda(n)),$$

choosing threshold functions $k_{uv}(n)$ as $k_{uv}(n) = C_uC_vk(n)$ with

$$k(n) = \left[\left(\frac{\lambda(n)}{2} + \Delta(n) - 2\right)e^{-\frac{2\Delta(n)}{\lambda(n)}} + O\left(\frac{1}{\lambda(n)}\right)\right], \tag{5}$$

leads to perfect network formation with resolution $\Delta(n)$.

To illustrate the conditions and the result of Theorem 1, suppose that $\lambda(n) = n^{1/3}$; note this function satisfies the conditions on $\lambda(n)$ in Theorem 1. Then Theorem 1 states that the network refinement mechanism allows user u to perfectly detect its neighborhood with resolution $\Delta(n)$, where $\Delta(n)$ can be, for example, any n^p with $1/4 < p \leq 1/3$.

Recall that $C_u\lambda(n)$ is roughly the expected size of user u's initial neighborhood set, and $\Delta(n)$ is the desired distance in the network formation process, i.e., a user would like to identify all other users whose interests are closer than $\Delta(n)$, and exclude all users that are further away than $\Delta(n)$.

Theorem 1 then gives (a) sufficient conditions on $\lambda(n)$ and $\Delta(n)$ for perfect network formation to be possible, and (b) an explicit expression for the threshold values $k_{uv}(n)$ that should be used in the neighborhood refinement step in order to achieve perfect network formation.

More precisely, Theorem 1 states that $\lambda(n)$ and $\Delta(n)$ cannot be too small or too large in order to be able to achieve perfect network formation. The intuition is as follows. On one hand, note that the smaller $\lambda(n)$ (i.e., the smaller the initial set obtained in the neighborhood discovery step) the less likely it is for the initial neighborhood sets of two users to overlap. Therefore, if $\lambda(n)$ is too small then the initial sets of two users may not overlap even if they have similar interests—and perfect network formation cannot be achieved. On the other hand, if $\lambda(n)$ is too large (i.e., close to n) then all users will have a large overlap, even when they have very dissimilar interests—again, in this case perfect network formation cannot be achieved. On the other hand, the fact that the resolution $\Delta(n)$ of network formation cannot be larger than the size of the initial set obtained in the neighborhood discovery step (which is of the order $\lambda(n)$) is intuitive. In addition, Theorem 1 also states that $\Delta(n)$ cannot be too small. Note that if $\Delta(n)$ is very small, the differences in the probabilities $\Pr(u \to v)$ between nodes that are just below/above the resolution $\Delta(n)$ is too small to achieve a perfect network formation.

Theorem 1 gives an explicit expression for the threshold values $k_{uv}(n)$ that should be used in the neighborhood refinement step to achieve perfect network formation. However, evaluating this expression requires the explicit knowledge of the the parameters C_u for all users u, as well as the explicit knowledge of the function $\lambda(n)$, which is unrealistic in practical situations. As a result, in order to apply the algorithm in a practical setting, the parameters C_u need to be estimated, and the threshold $k(n)$ needs to be determined through a training ("trial and error") phase. In Section V, we provide a numerical case study to show how this can be done.

A. Comparison with Network Discovery Process

One can show that under the latent space model of Section III, the network formation mechanism consisting only of a network discovery process is *unable* to achieve perfect network formation. This results shows that the mechanism using a network refinement step is indeed superior to one that only uses the network discovery process.

V. Numerical Case Study

We use a numerical case study to illustrate how the above network formation algorithm can be implemented in a practical setting. For the case study, we use a real-life dataset that was provided for the Netflix competition [12]; the dataset consists of actual ratings (from 1 to 5) that users gave to movies rented from the online video rental company Netflix. The dataset that we used for our case study consists of 1785 users and 5000 movies (not all users rated all movies).

We divide the dataset into two disjoint sets D_1 and D_2. We use the set D_1 for the above network formation mechanism, and use the set D_2 to evaluate the quality of the obtained network. More precisely, we use the network formation mechanism of Section III to identify for each user u a set of other users N_u who have similar preferences for movies, i.e., who tend to give similar movie ratings (we provide below a precise describtion of the algorithm that we use). We then use the obtained neighborhood sets N_u to try to predict the actual ratings that user u gave to a particular movie in the set D_2 by taking the average rating of users in the set N_u for this movie. To evaluate the quality of the network formation process, we determine how well the average ratings over a neighborhood set predicts the actual ratings by computing the Root-Mean-Square-Error (RMSE) for a total of $d = 602$ movies in the dataset D_2 by computing RMSE $= \sqrt{\frac{1}{d}\sum_{i=1}^{d}(\hat{T}_i - T_i)^2}$, where T_i and $\hat{T}_i$ are respectively the actual and predicted ratings of the i-th movie in the dataset D_2.

The algorithm that we use for network formation is given as follows.

Neighborhood Discovery: For user pair u and v we compute a similarity measure $s_{uv} \in [0, 1]$ given by

$$s_{uv} = 1 - \sum_{m \in \mathcal{M}_{uv}} |R_{um} - R_{vm}|/(4 \cdot \#\mathcal{M}_{uv}),$$

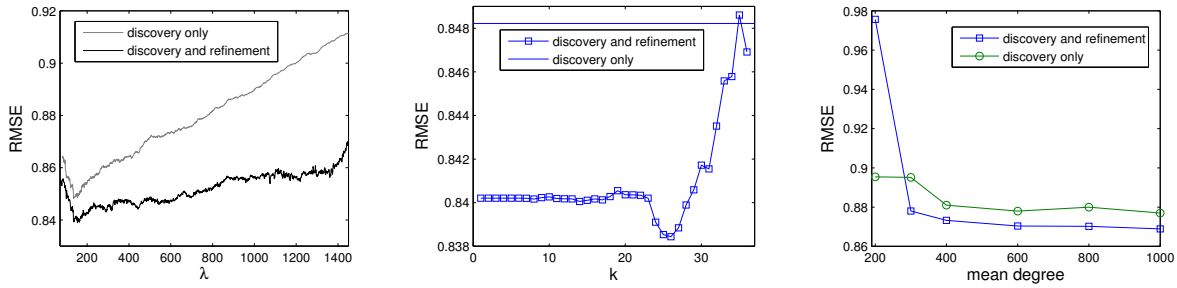
where $\mathcal{M}_{uv}$ is the set of movies in the set D_1 for which both users u and v provide a rating, and R_{um} and R_{vm} are the ratings that users u and v gave to movie m. The initial neighborhood set S_u of user u then consists of users v such that s_{uv} is within the top λ_u highest scores for user u. Here λ_u is a parameter that we can tune to test how the variance in the "relative socialness" of users influence the quality of the network formation process.

Neighborhood Refinement: Given a parameter k, the final neighborhood set N_u of user u consists of all users v in the initial neighborhood set S_u of user u such that the size of their initial neighborhood overlap is larger than the threshold $k_{uv} = (\lambda_u \lambda_v / \lambda^2_{average})k$.

Note that in the above algorithm, we use the size of the initial neighborhood set λ_u as an estimate for C_u in the algorithm given in Section III. In our simulations, we choose the parameter k in the neighborhood refinement step through "trial and error" as described below.

A. Full Information and Homogeneous Socialness

We first consider the case where each user computes the similarity measure for all other users and where all users are "equally social", i.e., we have $\lambda_u = \lambda$ for all users. Fig. 2(a) shows the performance of the network formation mechanism for different values of λ, where for each value of λ we choose through "trial and error" the threshold value k that minimizes the RMSE. Fig. 2(a) also shows the performance for the case where the network formation mechanism only uses the neighborhood discovery step, but not the neighborhood refinement step, i.e., uses the initial neighborhood set S_u to predict the ratings for user u. Note that the neighborhood refinement step improves the performance by at least 1%, which is a significant improvement in this context [13].



(a) Performance for different values for λ in homogeneous socialness case.

(b) Peformance for $\lambda = 136$ and different values of k, in homogeneous socialness case.

(c) Performance for values for the mean degree m and with $\lambda = 0.3$ and $k = 5$.

Fig. 2. Results for numerical case study.

Fig. 2(b) shows the performance of the network formation mechanism with the neighborhood refinement step for $\lambda = 136$ and different values of k. We note that the performance of the mechanism is surprisingly robust with respect to the threshold value k; this is a highly desirable property as it implies that picking a threshold value k in a practical setting is not too difficult.

B. Partial Information and Heterogeneous Socialness

Next we consider the more general (and realistic) setting where (a) users are able to compute the similarity measure only for a random subset of all users, and (b) users have different relative socialness resulting in different values of λ_u.

To do this, we first generate an undirected random graph $\mathcal{G}$ where the node degrees follow a Gaussian distribution [14] with mean m (a varying parameter) and standard deviation taken as one-fourth of the mean. The graph $\mathcal{G}$ determines "who meets who", i.e., for which other users v a given user u is able to compute the similarity measure s_{uv}.

The parameter λ_u used in the neighborhood discovery step to form the initial set S_u is then given by $\lambda_u = 0.3e_u$, where e_u is the node degree of user u in the random graph $\mathcal{G}$. The parameter k in the neighborhood refinement step is set to be 5. Purposely, we do in this case not carefully choose the threshold parameter k for the refinement step, to investigate how robust the mechanism is in more realistic settings.

Fig. 2(c) shows the performance of the two network formation mechanisms for different values of the mean node degree m in the random graph $\mathcal{G}$. Note that the network formation mechanism with the neighborhood refinement step again significantly outperforms the mechanism that only uses a neighborhood discovery step, until the mean degree reaches $m = 200$. The reason for the performance drop of the mechanism for $m = 200$ is that the initial neighborhood sets S_u become too small (the expected size of the initial neighborhood set in this case is equal to 60) and it becomes unlikely that the initial neighborhood sets of two users overlap (even if these users have very similar interests). As a result, the final neighborhood sets are too small to provide a good estimate of a users rating. Note this result is consistent with Theorem 1, which states the size of the initial neighborhood set, i.e., $\lambda(n)$, cannot be too small if one wants to achieve perfect network formation.

Fig. 2(c) suggests that choosing the size λ_u of the initial neighborhood set S_u is a good way to estimate the parameter C_u of the original algorithm, and that the network formation mechanism works well even without carefully choosing the threshold parameter k for the neighborhood refinement step. As it has been experimentally observed elsewhere [2], [4], [6], [5], these properties make the network formation mechanism with neighborhood refinement an algorithm that can be used well in practice.

REFERENCES

[1] K. Sripanidkulchai, B. M. Maggs and H. Zhang, "Efficient content location using interest-based locality in peer-to-peer systems," in *Proc. of INFOCOM*, 2003.

[2] Y. Zhang, G. Shen and Y. Yu, "LiPS: efficient p2p search scheme with novel link prediction techniques," in *Proc. of ICC*, 2007.

[3] H. H. Song, T. W. Cho, V. Dave, Y. Zhang, L. Qiu, "Scalable proximity estimation and link prediction in online social networks," in *Proc. of IMC*, 2009.

[4] M. E. J. Newman, "Clustering and preferential attachment in growing networks," *Physical Review E*, vol. 64, 2001.

[5] D. Liben-Nowell, J. Kleinberg, "The link-prediction problem for social networks," *Journal of the American Society for Information Science and Technology*, vol. 58, no. 7, pp. 1019-1031, 2007.

[6] L. A. Adamic, E. Adar, "Friends and neighbors on the web," *Social Networks*, vol. 25, no. 3, pp. 211-230, 2003.

[7] P. Sarkar, D. Chakrabarti and A. W. Moore, "Theoretical justification of popular link prediction heuristics," in *Proc. of COLT*, 2010.

[8] P. D. Hoff, A. E. Raftery and M. S. Handcock, "Latent space approaches to social network analysis," *Journal of the American Statistical Association*, vol. 97, no. 460, pp. 1090-1098, 2002.

[9] A. Condon, R. M. Karp, " Algorithms for graph partitioning on the planted partition model," *Random Structures and Algorithms*, vol. 19, no. 2 pp. 116-140, 2001.

[10] F. McSherry, "Spectral partitioning of random graphs," in *Proc. of FOCS*, 2001.

[11] D. Kempe and F. McSherry, "A decentralized algorithm for spectral analysis," in *Proc. of STOC*, 2004.

[12] J. Bennett and S. Lanning, "The Netflix Prize," in *Proc. of KDD Cup and Workshop*, 2007.

[13] Y. Koren, "Factorization meets the neighborhood: a multifaceted collaborative filtering model," in *Proc. of KDD*, 2008.

[14] M. E. J. Newman, S. H. Strogatz and D. J. Watts, "Random graphs with arbitrary degree distributions and their applications," *Physical Review E*, vol. 64, 2001.