Information Sciences 309 (2015) 102–118

Contents lists available at ScienceDirect

Information Sciences

journal homepage: www.elsevier.com/locate/ins

Friend recommendation with content spread enhancement in social networks

Zhi Yu, Can Wang *, Jiajun Bu, Xin Wang, Yue Wu, Chun Chen

Zhejiang Provincial Key Laboratory of Service Robot, College of Computer Science, Zhejiang University, Hangzhou 310027, China

ARTICLE INFO

Article history:
Received 12 October 2013
Received in revised form 20 January 2015
Accepted 9 March 2015
Available online 18 March 2015

Keywords:
Friend recommendation
Social network
Algebraic connectivity
Content spread
Information dissemination

ABSTRACT

Social network is becoming an increasingly popular media for information sharing. More and more people are interacting with others via major social network sites such as Twitter and Flickr. An important aspect of a social network is its capability in efficiently spreading content, not only within a small circle but also in the whole network. However, most existing methods for recommending friends in social networks only aim at achieving high recommendation success rate. The network grown from such recommendations is not optimized for content spread. In this paper, we propose a novel friend recommendation method ACR-FoF (algebraic connectivity regularized friends-of-friends) that considers both success rate and content spread in the network. Using the *algebraic connectivity* of a connected network to estimate its capability for spreading contents, our recommendation method naturally extends existing friend recommendation algorithms such as FoF to achieve both recommendation relevance and content spread in a social network. Experimental results on simulated and real social network data sets show that our method can significantly improve content spread in a social network with only a very tiny compromise on friend recommendation success rate.

1. Introduction

The past several years witnessed an explosive growth of social network sites and applications. People are increasingly relying on various social networks for sharing contents and interacting with each other. Popular sites such as Twitter and Facebook are seeing a great amount of tweets or posts generated by active users all around the world every day. Social media nowadays has become one of the most important information sources [33] and we intuitively expect information spread in a viral fashion in a social network.

However, recent works indicate information or content in social networks may not spread as efficiently as people believe. Cha et al. explain in [3] the formation of *content locality* in Flickr as they observed from a Flickr dataset ranging over 104 consecutive days, which shows that even popular photos many only circulate within a small clique and result in a quick burnout in content spread. Bakshy et al. echo in [2] with similar discoveries by noting increasing *homophily* in social networks. They point out that individuals with similar characteristics tend to associate with each other, leading to greater

* Corresponding author.
E-mail addresses: yuzhirenzhe@zju.edu.cn (Z. Yu), wcan@zju.edu.cn (C. Wang), bjj@zju.edu.cn (J. Bu), xinwang@zju.edu.cn (X. Wang), wy1988@zju.edu.cn (Y. Wu), chenc@zju.edu.cn (C. Chen).

http://dx.doi.org/10.1016/j.ins.2015.03.012
0020-0255/

opportunities of aligned information source and less possibilities for accessing novel and diverse information. Homophily not only impairs people's motivation for sharing contents, but also may slow down audience growth in social networks.

On the other hand, it is generally believed that the value and revenue of a social network are closely related with its capability in efficiently spreading contents. Content spreading plays an important role in motivating users to express their opinions through the social network because the goal of sharing information is for others to receive [35,21]. Quicker information dissemination can drive up user engagement and in turn improve user retention. Wide outreach of contents in a social network also improves the access to novel information for users. Most social network sites are now offering connectivity boosting features such as "*friend recommendation*" to users to improve content spread. However, traditional friend recommendation methods such as friends-of-friends (FoF) mainly consider number of common neighbors or similarity of user profiles in recommending new friends. One drawback of these methods is that they usually increase connectivity only within a small range of neighborhood and may not necessarily lead to improved content spread capability of the whole network.

Although connecting unknown people seems to be an effective method in discovering novel contents and reducing homophily in social networks, most users lack enough motivation to contact unknown people and thus making the recommendations fail. Considering the limitations of these traditional friend recommendation methods, Chaoji et al. recently proposed in [4] a connection recommendation algorithm that can also improve content spread in a social network. A two-step process is used in their recommendation by first selecting a candidate set of similar users based on the number of common neighbors, similarity of user profiles, etc. and then recommending a certain number of friends in the social network (no more than k friends for each user) that maximize the content spread. The limitation of their method is that the user is not likely to accept all the k recommended friends, which makes the added connections less optimal for content spread. Also the relatively high computational cost of their algorithm makes it less scalable in practical applications. But as the first attempt to incorporate content spread capability into friend recommendation, their work inspires us to seek solutions that better balance the two objectives.

The content spread in a network is usually modeled as a stochastic process, e.g. independent cascade (IC) model [17]. An edge $e(u, v)$ in this model is associated with a probability $p(u, v)$, which is the possibility in step $t + 1$ the node u independently propagate to node v the content it receives in step t. Metrics based on IC model, such as the expected amount of content received by nodes [4], are used to evaluate the content spread. But directly optimizing such metric is NP-hard [4]. Even approximation algorithms will incur expensive computational cost. Some existing approaches use the *largest eigenvalue* of the **Adjacency Matrix** [31,26,29] to analyze or improve content spread in a network. In this paper, we use the *algebraic connectivity* of a connected network's **Laplacian Matrix** to improve its capability in spreading content. We think the algebraic connectivity of a network is more closely related to its content spread capability than the largest eigenvalue of the adjacency matrix, which is related more closely to the threshold of an epidemic contagion. Using the algebraic connectivity as a regularizer, we develop a new friend recommendation algorithms called *algebraic connectivity regularized friends-of-friends* (ACR-FoF), which effectively take into account both relevance and content spread in friend recommendation. We summarize the major contributions as follows:

- To the best of our knowledge, this is the first time the algebraic connectivity of a network is used to optimize its content spread capability.
- We propose a new evaluation metric EnPair (enhanced pairs) for evaluating improvement in content spread in social networks and the reduction in homophily.
- We develop a new friend recommendation algorithm ACR-FoF that considers both relevance and content spread in a social network.

The rest of the paper is organized as follows: Section 2 introduces to the related works. Our new friend recommendation algorithm is explained in Section 3. In Section 4, we present and discuss experiments and results. Finally, we conclude the whole paper and present some directions for future work in Section 5.

2. Related works

Our work is related to information disseminations and recommendation in social networks. Here we briefly review these related works.

2.1. Information dissemination

Study of information dissemination or content spread in social networks can be traced back to [27], in which Rogers explains how new ideas spread via communication channels over time. He also analyzes how information spread transforms the way people communicate and adopt new ideas. Some more recent works start exploiting combinatorial optimization in minimizing the network diameter and the average shortest path distances [8,23]. However, since not all recommendations will be accepted by users in practice, these algorithms will lead to a suboptimal solution in recommendation. We will provide more analysis into this issue in Section 3.

A hot topic in information dissemination is influence maximization in social contagion. Many new models and algorithms have been proposed to find a small subset of seed nodes in a social network that maximize the spread of influence. Domingos and Richardson model this diffusion process as a Markov random field and provide solutions based on greedy search and hill-climbing search [9]. Kempe et al. study this diffusion process using both *independent cascade* (IC) model and *linear threshold* model and propose greedy strategies for both models, with provable approximation guarantees showing the solutions are within 63% of the optimal [17]. They further model the diffusion process with a *decreasing cascade model* in [18] and use a greedy algorithm to find the most influential nodes. Chen et al. improve the greedy strategy over IC with *degree discount* heuristic that drastically reduces the algorithm's running time while maintains almost the same influence spread capability in networks with a small propagation probability [7]. Although influence maximization problem aims to identifying a set of nodes instead of recommending edges as we do in this paper, the diffusion models and algorithms for influence maximization problem are of great value to our work.

Particularly relevant to our work are algorithms for information dissemination in a network using eigenvalue optimization. Tong et al. propose in [29] novel methods to most effectively speedup or contain a contagion in a network by adding or deleting k edges. They boil down the problem to the optimization of the largest eigenvalue of the adjacency matrix. This relation between the dissemination in a network and the largest eigenvalue of the adjacency matrix is explored in [31,26]. Both works confirm that the largest eigenvalue of the adjacency matrix will determine whether a dissemination in a network will become an epidemic.

2.2. Recommendation in social networks

With the growing popularity of social network applications, various recommendation systems emerge like mushrooms after rain. These include recommending item ratings [34], tags [12], documents [14], friends [5,15], experts [21,13] and many others in social networks. Various recommendation algorithms are exploited in these systems, ranging from the canonical collaborative filtering [16], graph-based propagation [12,14], to the new list-wise probabilistic matrix factorization [22], etc. The enthusiasm in this area is expected to last for the years to come as many new business models in social networks are highly dependent on recommendation accuracy.

Among these recommendation systems, friend recommendation lies in the very core of a social network, as it essentially determines how a social network might grow. Most of the existing friend recommendation approaches are based on the similarity of user profiles, or the geographical vicinity or the number of common friends [5,15]. Some other works cast friend recommendation as a link prediction problem by finding the most probable links among existing nodes [20,28]. More recently, Dong et al. [10] and Yang et al. [32] take into consideration heterogeneous structures in social networks and achieve better accuracy. Oyama et al. [25] find more links among nodes in different time frames and make recommendations by combining this information in dynamic environments.

Most similar to our work is the friend recommendation method proposed by Chaoji et al. [4]. Their algorithm aims to achieve both relevance and efficient content spread in a social network using a two-step process. Different from their method, we develop a new one-step algorithm using algebraic connectivity regularization in friend recommendation. Our algorithm alleviates the content locality problem by providing more access to novel information, as well as enhances content spread by optimizing network connectivity.

3. Recommendation mechanism

The objective of friend recommendation in a social network is to attach new edges between unconnected nodes in the network. Most previous studies focus on improving the success rate in friend recommendation while ignoring the impact on the network structure. Our work in this paper aims to improve both the content spread capability of a social network and recommendation success rate.

3.1. Definition

In this paper, a social network is modeled as an undirected graph $G = (V, E)$ where V stands for the set of user nodes and E connections between them. G can be represented as an *adjacency matrix* W in which:

$$W_{i,j} = \begin{cases} 1: & V_i \text{ and } V_j \text{ are friends} \\ 0: & \text{otherwise} \end{cases} \tag{1}$$

W obviously is a symmetric matrix, i.e. $W_{i,j} = W_{j,i}$. A *path* $Path_{i,j}$ exists if there are a sequence of edges connecting V_i and V_j and we denote

$$Path_{i,j} = \{V_i, V_{l_1}, \ldots, V_{l_m}, V_j\} \tag{2}$$

The *length* of a path $Path_{i,j}$ is the number of edges that the path uses: $|Path_{i,j}| = m + 1$. The *distance* between V_i and V_j is the length of the minimum path between them, i.e.: $Dist_{i,j} = \min(|Path_{i,j}|)$. We define the distance from a node to itself to be 0. The *diameter* of a graph G is then defined as follows:

$$D(G) = \max(Dist_{i,j}) \tag{3}$$

Note that there may exist multiple node pairs in a graph with distance equaling to its diameter, which are termed as *external pairs* in [30].

The mean distance $\bar{\rho}(G)$ of a graph G is the average distance between distinct vertices of G. That is:

$$\bar{\rho}(G) = \frac{1}{n(n-1)} \sum_{i=1}^{n} \sum_{j=1}^{n} Dist_{i,j} \tag{4}$$

We also define a candidate edge set $E2$ with $E2 \cap E = \emptyset$. The objective of edge recommendation is to choose a set of edges $E' \subset E2$. We summarize the symbols used in Table 1.

3.2. Content spread metrics in a social network

Content spread in social networks has been a hot research issue. But few works exist on how we can measure the capability of a social network in spreading contents. In this paper, we will focus on the topological aspect of a social network in discussing its content spread capability. To comprehensively evaluate the capability of a social network in content spread, we will use three metrics in this paper: (1) expected content spreading (ECS), as used in [4]; (2) network diameter; and (3) enhanced pairs (EnPair), the number of node pairs whose content sharing probability is improved by adding new edges. EnPair is a new evaluation metric we propose in this paper. We will show in the following discussions how these three metrics are interrelated. We start with the *maximum probability path* (MPP) proposed by Chen et al. [6]. MPP models the content sharing probability $P_{i,j}$ between two nodes i and j as a function of the distance between them:

$$P_{i,j} = p^{Dist_{i,j}} \tag{5}$$

where p is the probability of a message propagating from one node to its neighbors. Note that this uniform assignment of probability is a simplified model for representing propagation among nodes. A more sophisticated model will use weighted assignment of probability considering factors such as the node's degree. We will leave weighted assignment to the future work and focus on models based on uniform assignment in this paper. Based on MPP, we define the minimal propagation probability $LowP(G)$ in a social network G as follows:

$$LowP(G) = \min(P_{i,j}) \tag{6}$$

Since $P_{i,j} = p^{Dist_{i,j}}$ and $0 < p < 1$, Eq. (6) can be rewritten as:

$$LowP(G) = p^{D(G)} \tag{7}$$

where $D(G)$ is the diameter of the social network. Although $LowP(G)$ can be directly used as an important indicator for content spread, its value might be too small to be well represented in a computer. Thus we will use the diameter of the social network $D(G)$ as the metric for the minimal propagation probability in a social network.

Eq. (7) indicates that maximizing $LowP(G)$ is equivalent to minimizing the diameter $D(G)$. However, minimizing the diameter of a network is a challenging issue and the diameter $D(G)$ is sometimes insensitive to network changes. Toueg and Steiglize suggest an improvement by optimizing the number of *external pairs* in a network instead [30]. We present an example in Fig. 1 and Table 2 to illustrate this method. Fig. 1 depicts a network with 21 nodes and 20 edges, and 12 external pairs. Adding an edge a or b to a node pair i and j with $Dist_{i,j} = 2$ (or a *2-hop edge* for short), the diameter remains unchanged. But as shown in Table 2, when adding the edge a, the number of external pairs is reduced to 7, while adding the edge b the number of external pairs is reduced to 3, meaning that adding edge b is a better choice. The practical problem with external pair optimization is it incurs a time complexity of $O(n^5)$, making an algorithm difficult to scale.

We define the second metric for content spread in a social network G, *expected content spread* ($ECS(G)$) as follows:

$$ECS(G) = \frac{1}{n} \sum_{i=1}^{n} \sum_{j=1}^{n} P_{i,j} \tag{8}$$

$ECS(G)$ represents average number of nodes who can receive the message sent by a node in the social network. We have the following approximation for $ECS(G)$:

$$ECS(G) \approx \frac{1}{n} \sum_{i=1}^{n} \sum_{j=1}^{n} p^{\bar{\rho}(G)} = np^{\bar{\rho}(G)} \tag{9}$$

where $\bar{\rho}(G)$ is the mean distance of G. Eq. (9) shows that $ECS(G)$ is closely related with the mean distance of G.

Although $ECS(G)$ is generally a good indicator for content spread capability of a social network G, it may fail to distinguish the *homophily* in a social network. We thus introduce the third metric, *enhanced pairs* (EnPair), to show the number of node pairs whose content sharing probability is improved by adding new edges. The enhanced pairs for adding new edges X is defined as follows:

Table 1
Symbols.

Symbol	Definition and description
A	Matrices
$A_{i,j}$	The element at the ith row and the jth column of A
$\mathbf{1}$	Vector with all ones
$\mathbf{0}$	Vector with all zeros
$G = (V, E)$	The original graph
n	The number of the nodes in the graph
$Dist_{i,j}$	The shortest path length from node i to j
$D(G)$	The diameter of G
$\bar{\rho}(G)$	The mean distance of G
p	The probability of one node sharing a message
$P_{i,j}$	The content sharing probability between i and j
$LowP(G)$	The minimal propagation probability of G
$ECS(G)$	The expected content spread of G
$EnPair(G)$	The number of enhanced node pairs
W	The adjacent matrix
$W2$	The adjacent matrix for all 2-hop edges
L	The Laplacian matrix
$L2$	The Laplacian matrix for all 2-hop edges
$\lambda_i(A)$	The ith smallest eigenvalue of the matrix A
$v_i(A)$	The ith smallest eigenvector of the matrix A

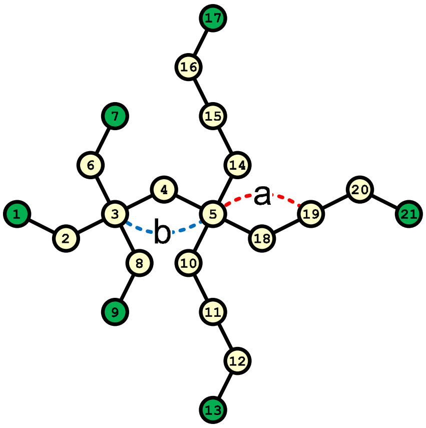


Fig. 1. A connected network with 21 nodes and 20 edges. a and b are 2 candidate edges to be added.

Table 2
External pairs changed for Fig.1 added with one edge: √ means this pair is still a diameter after add the given edge, while × means not.

No.	External pair	Diameter path	Add a	Add b
1	(1,13)	1-2-3-4-5-10-11-12-13	√	×
2	(1,17)	1-2-3-4-5-14-15-16-17	√	×
3	(1,21)	1-2-3-4-5-18-19-20-21	×	×
4	(7,13)	7-6-3-4-5-10-11-12-13	√	×
5	(7,17)	7-6-3-4-5-14-15-16-17	√	×
6	(7,21)	7-6-3-4-5-18-19-20-21	×	×
7	(9,13)	9-8-3-4-5-10-11-12-13	√	×
8	(9,17)	9-8-3-4-5-14-15-16-17	√	×
9	(9,21)	9-8-3-4-5-18-19-20-21	×	×
10	(13,17)	13-12-11-10-5-14-15-16-17	√	√
11	(13,21)	13-12-11-10-5-18-19-20-21	×	√
12	(17,21)	17-16-15-14-5-18-19-20-21	×	√

$$EnPair(G_{new}) = |\{(i,j)|P_{i,j}^{G_{new}} > P_{i,j}^{G}\}| \quad (10)$$

where $G_{new} = (V, E \cup X)$.

It can be easily proven that if we add a new edge between 2-hop nodes i and j and there are x node pairs whose shortest distance are reduced by 1, the mean distance will then be reduced by $x/(n(n-1))$. This indicates the edge that reduces the mean distance most is the edge that will produce the largest number of enhanced pairs.

3.3. Algebraic connectivity

The three content spread metrics are difficult to be optimized directly. Instead, we consider optimizing the *algebraic connectivity* of a connected network to improve its content spread capability. Let $G = (V, E)$ be an undirected graph with n nodes and m edges. The algebraic connectivity of G is the second-smallest eigenvalue of the Laplacian matrix of G. Given a graph with n nodes and m undirected edges, for an edge l between node i and j, we define the edge vector $a_l \in \mathbb{R}^n$ as $a_{l_i} = 1, a_{l_j} = -1$, and other entries 0. Then we get the incidence matrix $A \in \mathbb{R}^{n\times m}$ with a_l being the l-th column. The Laplacian matrix L is the $n \times n$ matrix:

$$L = AA^T = \sum_{l=1}^{m} a_l a_l^T \quad (11)$$

Obviously, L is positive semidefinite and $L\mathbf{1} = \mathbf{0}$, with $\mathbf{1}$ being the vector of all ones. Let $\lambda_1 \leqslant \lambda_2 \leqslant \cdots \leqslant \lambda_n$ be the eigenvalues of L. The second smallest eigenvalue λ_2 is called the *algebraic connectivity* of graph G. Previous studies have shown that the algebraic connectivity can be used as an upper bound on the diameter and the mean distance of graph G as follows [24]:

$$D(G) \leqslant 2\left\lceil \frac{\Delta + \lambda_2}{4\lambda_2} \ln(n-1) \right\rceil \quad (12)$$

$$\bar{\rho}(G) \leqslant \frac{n}{n-1}\left(\left\lceil \frac{\Delta + \lambda_2}{4\lambda_2} \ln(n-1) \right\rceil + \frac{1}{2}\right) \quad (13)$$

where Δ is the maximal vertex degree of graph G. When adding edges, maximizing λ_2 is equivalent to minimizing the upper bound of $D(G)$ and $\bar{\rho}(G)$. This is why optimizing *algebraic connectivity* will improve all the three content spread metric we defined above.

To better illustrate the connection between algebraic connectivity and the three content spread metrics proposed in this paper, we present an example in Figs. 2, 3 and Table 3. Fig. 2 shows a simple connected network of 20 nodes and 19 edges. Table 3 displays the contrast in the change of algebraic connectivity ($\lambda_2(L)$), diameter ($D(G)$), expected content spread ($ECS(G)$) and enhanced pairs ($EnPair(G)$) after adding two different edges. Fig. 3(a)–(d) shows algebraic connectivity and the other three metrics as a function of adding an edge between two different endpoints in the network, where no constraint is imposed on the edge to be added. Fig. 3(e)–(f) shows correlations between algebraic connectivity and $ECS(G), EnPair(G)$ respectively, while restricting the added edge to be a 2-hop edge. We omit the correlation between algebraic connectivity and diameter $D(G)$ because in this case, $D(G) = 18$ for any one 2-hop edge added.

3.4. Perturbations on algebraic connectivity

Calculating λ_2 of a matrix will incur a heavy computational cost of $O(n^3)$. Performing an exhaustive search in the edge space and then calculating λ_2 for each outcome will be computationally prohibitive. Drawing on recent works on algebraic connectivity and largest eigenvalue [11,29], we present in this part an approximate algorithm for efficiently calculating the perturbation on algebraic connectivity by adding an arbitrary edge in a network.

Given a network represented as a graph $G1 = (V, E)$, we define a candidate edge graph $G2 = (V, E2)$, where $E2$ is the candidate edge set as defined previously. We denote the graph $G' = (V, E')$ where $E' \subset E2$, $v_i(L)$ and $v_i(L')$ are the ith smallest eigenvectors of L and L' respectively. We have:

$$v_1(L) = \mathbf{1} \quad (14)$$

$$\forall i, \quad v_i(L)^T v_i(L) = 1 \quad (15)$$

$$\forall i \neq j, \quad v_i(L)^T v_j(L) = 0 \quad (16)$$

The eigenvectors $v_2(L), v_3(L), \ldots, v_n(L)$ will span the subspace $\mathbf{1}^{\perp}$, as pointed out by Kim and Mesbahi in [19]. Where

$$\mathbf{1}^{\perp} := x \in \mathbb{R}^n | \mathbf{1}^T x = 0 \quad (17)$$

and we have:

$$Lv_i(L) = \lambda_i(L) v_i(L), \quad 1 \leqslant i \leqslant n \quad (18)$$

Using Eqs. (14) and (16), we get:

$$v_i(L)^T L v_i(L) = \lambda_i(L) v_i(L)^T v_i(L) = \lambda_i(L) \quad (19)$$

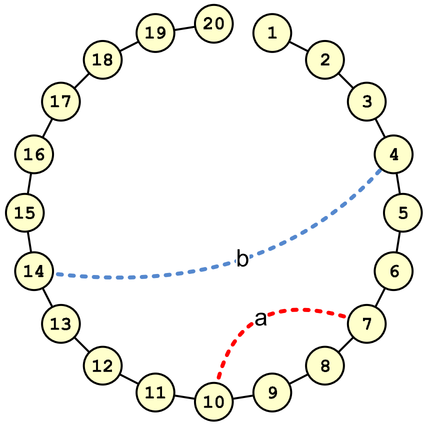


Fig. 2. A sample connected network of 20 nodes and 19 edges. *a* and *b* are 2 candidate edges waiting to be added.

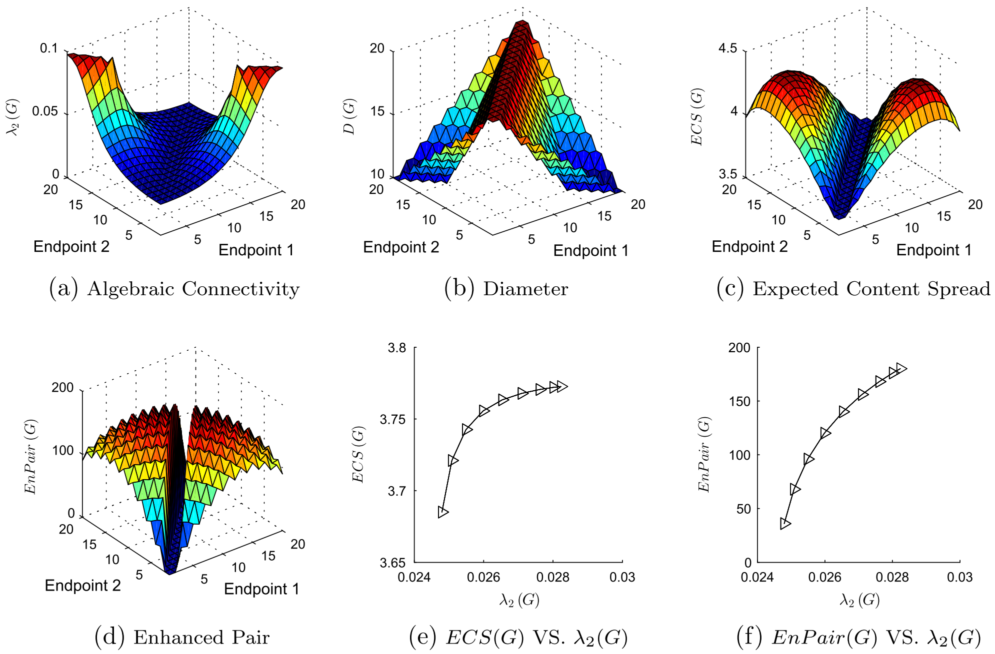


Fig. 3. Network changes resulted by adding 1 edge to the network in Fig. 2.

and

$$v_i(L)^T L v_j(L) = \lambda_i(L) v_i(L)^T v_j(L) = 0 \tag{20}$$

Since $v(L')^T v(L') = 1$ and $v(L')^T \mathbf{1} = 0$, we have:

$$v_i(L') = \sum_{i=2}^{n} \gamma_i * v_i(L) \tag{21}$$

and

$$v_i(L')^T * v_i(L') = \sum_{i=2}^{n}\sum_{j=2}^{n} \gamma_i * v_i(L)^T * v_j(L) * \gamma_j = \sum_{i=2}^{n} \gamma_i^2 = 1 \tag{22}$$

Finally we have:

$$v_i(L')^T L v_i(L') = \sum_{i=2}^{n}\sum_{j=2}^{n} \gamma_i * v_i(L)^T * L * v_j(L) * \gamma_j = \sum_{i=2}^{n} \gamma_i^2 * v_i(L)^T * L * v_j(L) = \sum_{i=2}^{n} \gamma_i^2 * \lambda_i(L) \geqslant \lambda_2(L) * \sum_{i=2}^{n} \gamma_i^2 = \lambda_2(L) \tag{23}$$

Table 3
Network changes resulted by adding 1 edge to the network in Fig. 2.

Add edge	$\lambda_2(G)$	$D(G)$	$ECS(G)$	$EnPair(G)$
a(7-10)	0.0303	17	3.8574	154
b(4-14)	0.0621	11	4.3027	156

Since $L_{new} = L + L'$, we can easily obtain

$$\lambda_2(L_{new}) = v_2(L_{new})^T L v_2(L_{new}) + v_2(L_{new})^T L' v_2(L_{new}) \geqslant \lambda_2(L) + \lambda_2(L') \tag{24}$$

So maximizing $\lambda_2(L_{new})$ can be relaxed to maximizing $\lambda_2(L')$. Note that $E' \subset E2$, maximizing $\lambda_2(G')$ is equivalent to selecting edges from $E2$ that have the greatest impact on $\lambda_2(L2)$. When removing an edge $l = (a, b)$ from $G2$, the Laplacian matrix for the newly obtained graph is:

$$L2_{new} = L2 - A_l \tag{25}$$

where A_l is the incidence matrix of edge $l = (a, b)$ with $A_{a,a} = A_{b,b} = 1, A_{a,b} = A_{b,a} = -1$ and other entries being 0. So the impact of removing edge l on algebraic connectivity is $D_l = (v_2(L2)_a - v_2(L2)_b)^2$, which happens to be the gradient of $\lambda_2(L2)$ to the edge $l = (a, b)$ [11]. This result can also be derived in a more intuitive manner: if $G2$ is a large graph with a great number of edges, removing one edge has only insignificant impact on its second eigenvector. Therefore we have:

$$\begin{aligned}\lambda_2(L2_{new}) &= v_2(L2_{new})^T L2_{new} v_2(L2_{new}) \approx v_2(L2)^T L2_{new} v_2(L2) = v_2(L2)^T L2\, v_2(L2) - v_2(L2)^T A_l v_2(L2) \\ &= \lambda_2(G2) - (v_{2a} - v_{2b})^2\end{aligned} \tag{26}$$

where v_{2a} and v_{2b} are the ath and bth entries of vector $v_2(L)$ in short.

By computing all the D_l and sorting them in a decreasing order, we obtain a ranking of the perturbation on algebraic connectivity by adding an arbitrary edge.

3.5. Algebraic Connectivity Regularized Friend Recommendation (ACR-FoF)

Here we propose a novel friend recommendation algorithm ACR-FoF that balances recommendation relevance and content spread. For adding a new edge l connecting the node a and node b, we use an edge scoring function to rank the edges as follows:

$$Score(l) = Score_{FoF}(l) + \alpha f(Score_{ACR}(l)) \tag{27}$$

The recommendation score for an edge l consists of two parts: (1) $Score_{FoF}(l)$ obtained from the FoF algorithm. The FoF algorithm is based on the intuition that "a friend of my friend can probably be my friend and if a person shares many common friends with me, then there is a great chance that he or she may also be my friend." So here: $Score_{FoF}(l)$ = *the number of common friends between node a and node b*. (2) $Score_{ACR}(l)$ shows the impact on algebraic connectivity λ_2 by adding an edge l. From Eq. (26) we have: $Score_{ACR} = (v_{2a} - v_{2b})^2$. The regularization parameter α controls the contribution from the two parts. To identify good edges which can improve content spread capability efficiently, we define the $f(Score_{ACR}(l))$ as follows:

$$f(Score_{ACR}(l)) = \log(Score_{ACR}(l)) \tag{28}$$

Using the logarithm function will effectively lower the scores of the edges that only make little change on algebraic connectivity. As a result, the ranking scores from ACR-FoF can efficiency discriminate between good edges and bad edges. This characteristic is demonstrated in our experiments. We summarize the above ranking algorithm in Algorithm 1.

Algorithm 1. Algebraic Connectivity Regularized Friend Recommendation

Input: The original social network graph, G; The candidate edges graph, $G2$; Parameter α
Output: Recommend order list of candidate edges *List*;
1: Obtain the Laplacian matrix $L2$ for $G2$;
2: Calculate the second smallest eigenvector v_2 of $L2$;
3: Initialize all entries in rating matrix S to 0;
4: **for** each $l = (a, b) \in G2$ **do**
5: $Score_{FoF}$ = the number of common friends between node a and b;
6: $Score_{ACR} = (v_{2a} - v_{2b})^2$;
7: $S(l) = Score_{FoF} + \alpha \log(Score_{ACR})$
8: **end for**;
9: Sort all candidate edges by $S(l)$ in decreasing order, and save the sequence in *List*;
10: **return** *List*;

The ranking scores from ACR-FoF will lead to effective content spread by taking the algebraic connectivity into consideration. Meanwhile, it ensures relevance in recommendation by both incorporating the FoF part into the ranking score and constraining the candidate edge set in 2-hop edges (i.e. only recommending friends of friends). This way, it balances the two objectives by setting the appropriate regularization parameter α.

To recommend multiple edges, we use a greedy scheme instead of combinatorial optimization by selecting the top-k edges ranked by $Score(l)$. The rationale behind this is that users are unlikely to accept all the recommendations as a whole, i.e. they tend to accept some of the recommendations while rejecting others. This will make the combinatorial optimization result a suboptimal solution. To better illustrate this situation, we present an example of a graph with 7 nodes on a line in Fig. 4(a), and we will add three 2-hop edges to reduce the mean distance. The three blue lines in Fig. 4(a) are the result of combinatorial optimization, while the red lines are selected by the greedy scheme. If all the three recommendations are accepted, the three blue line will achieve the best performance. If recommendation success rate for each edge $p < 1$, the situation can be tricky. We calculate the expected mean distance (EMS) for both solution, and show the Greedy to Combinatorial EMS ratio under different p value in Fig. 4(b). The region where the EMS ratio is less than 1 is the region the greedy scheme will achieve better performance. As we can see from Fig. 4(b), the greedy scheme will excel when $p < 0.6$. In practice, when the number of the recommended edges are big, the recommendation success rate tends to be quite low. This means the greedy scheme will produce better result with a much lower computational cost.

4. Experiments and result

In this section, we will evaluate the performance of our algorithm in: (1) improving content spread in a network and (2) friend recommendation success rate. We perform our experiments in two different types of data sets: first in synthetic data sets to test the algorithms' capability in content spread and then in real social network data sets to test how different algorithms perform in both content spread and friend recommendation. We will begin with the content spread capability experiments in synthetic data sets first.

4.1. Content spread experiments

In this part, we compare our methods with existing algorithms to examine their effectiveness in improving content spread capability of a network. First of all, we shall notice that ACR-FoF is a friend recommendation algorithm. So the purpose of ACR-FoF is not to solely optimize content spread capability, but rather to achieve a good balance in boosting content spread and recommending relevant friends. Thus in the experiments, our main purpose is to examine the content spread capability of both the ACR algorithm and ACR-FoF, i.e. how much our friend recommendation algorithm will improve the content spread. We will begin our description of the experiments with the data sets.

4.1.1. Data sets

As the algorithms are supposed to work on a connected network, we generate two types of connected networks: (1) RTG Network and (2) 2-Core RTG Network.

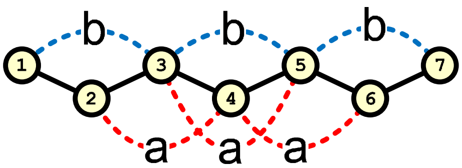
RTG Network: Graphs generated by a Random Typing Generator bear much resemblance to real graphs such as social networks [1]. Experiments on RTG networks can show how different algorithms will perform in real social networks. We generate 20 RTG networks as described in [1],[1] with the following parameter settings: $W = 1000$, $k = 5$, $\beta = 0.5$, $q = 0.2$, $isBipartite = True$, $isSelfLoop = True$ and $numTimeTicks = 100$. The statistics of the generated RTG networks are summarized in Table 4.

2-Core RTG Network: The RTG networks used in previous experiments are single-core networks, while real social networks are multi-core, i.e. consisting of multiple densely connected subgraphs. Thus the experiments in RTG networks cannot differentiate whether the added edges will improve the connection within one specific core (increasing homophily), or improve the connection within multiple cores (increasing homophily) or connect different cores (tending to reduce homophily). So we add a random edge between 2 RTG networks to make a *2-Core RTG Network*. Thus we obtain 10 2-Core RTG networks from these 20 RTG networks. An example of the 2-Core RTG Network is shown in Fig. 5(c). The statistics for the ten 2-Core RTG networks used in our experiments are summarized in Table 5.

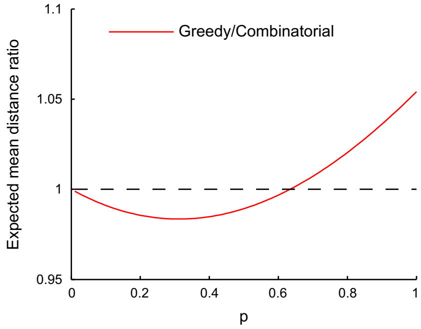
4.1.2. Evaluation metrics

We will use the three metrics discussed previously to evaluate the algorithms' impact on the content spread capability of a social network. To highlight the improvement brought by different algorithms, we will use the improvement ratio on the three metrics in the experiments. Thus the evaluation metrics finally used in the experiments are: (1) Improved Network Diameter Ratio (INDR); (2) Improved Expected Content Spread Ratio (IECSR); and (3) Enhanced Pairs Ratio (EPR). The three metrics are defined as follows respectively:

[1] Codes download from www.cs.stonybrook.edu/~leman/pubs.html.



(a) Adding three edges to minimize the *mean distance* of an example network, red edges are resulted from the greedy solution and the blue edges from combinatorial optimization

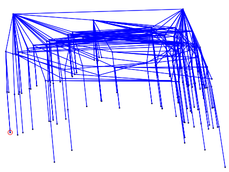


(b) Expected mean distance ratio VS. recommendation success rate p

Fig. 4. Greedy solution vs. combinatorial optimization.

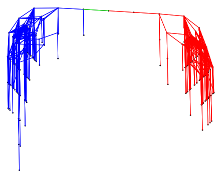
Table 4
Statistics for RTG Networks.

	Minimum	Maximum	Average
Points	311	412	369.2
Edges	436	514	479.25
Diameter	9	14	10.45
Mean distance	3.8114	4.3343	4.0277
Average degree	2.3669	3.1254	2.6110
Maximum degree	29.2269	35.8531	32.4995
Candidate edges	4373	7084	5639.3



(a) RTG Network1

(b) RTG Network2



(c) 2-Core RTG Network

Fig. 5. RTG Network and 2-Core RTG Network examples.

Table 5
Statistics for 2-Core RTG Networks.

	Minimum	Maximum	Average
Points	689	785	738.4
Edges	926	1006	959.5
Diameter	13	16	14.6
Mean distance	5.0386	6.1876	5.6433
Average degree	2.4517	2.8338	2.6024
Maximum degree	54	79	66
Candidate edges	10,129	12,581	11,325

$$INDR = \frac{D(G) - D(G_{new})}{D(G)} \quad (29)$$

$$IECSR = \frac{ECS(G_{new}) - ECS(G)}{ECS(G)} \quad (30)$$

$$EPR = \frac{EnPair(G_{new})}{n(n-1)} \quad (31)$$

where G stands for the original graph and G_{new} the new graph after adding the recommended edges. $D(\cdot)$, $ECS(\cdot)$ and $EnPair(\cdot)$ are defined in Eqs. (3), (8) and (10) respectively. For each metric, higher values mean better performance.

4.1.3. Compared algorithms

To demonstrate the extent of improvement in content spread of our ACR algorithm, we compared with some of the state-of-the-art algorithms. The algorithms tested in our experiments include:

ACR: Although we term our algorithm as algebraic connectivity regularized friends-of-friends, the regularization term ACR (algebraic connectivity regularization) can be used separately to optimize the algebraic connectivity of a network as follows:

$$Score_{ACR}(l) = (v_{2a} - v_{2b})^2 \tag{32}$$

where v_2 is the second smallest eigenvector of $L2$.

LE: Eigenvalues of the adjacency matrix or the Laplacian matrix are actively explored in graph optimization problems. Some recent works use the largest eigenvalue of the adjacency matrix to tune the network structure. Here we compare the method in [31,26,29], which maximizes the Largest Eigenvalue (LE) of the adjacency matrix to improve the network content spread.

$$Score_{LE}(l) = (v_{ma} - v_{mb})^2 \tag{33}$$

where v_m is the largest eigenvector of G.

RMPP: The Restricted Maximum Probability Path Model (RMPP) was proposed in [4] to improve content spread in a network. Notice that by directly optimizing for $ECS(G)$ of a network, RMPP can find a near-optimal solution by iteratively calculating $ECS(G)$ after adding an edge. However, this will incur expensive time cost, making it impractical in real applications.

$$Score_{RMPP}(l) = ECS(G_{new}) - ECS(G) \tag{34}$$

where G_{new} is graph after adding edge l to G.

FoF: Recommend the edge with the largest number of common friends.

ACR-FoF: Combine the weight of ACR and FoF using Eq. (27), where α is set to 0.5, and choose the edge with maximum weight.

RAND: Randomly add edges.

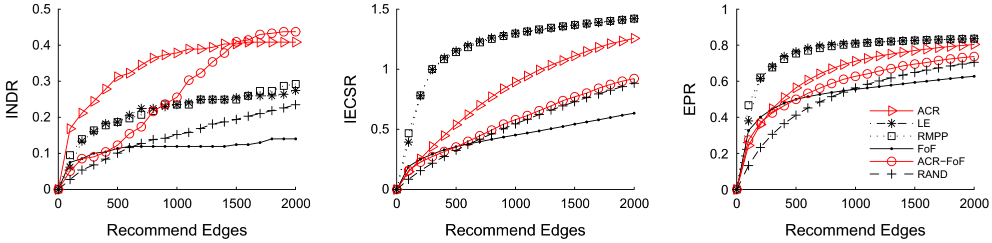
We compare the algorithms' performance in recommending top-k edges and show the change of the content spread capability after adding all the k recommendations. The result of RAND is averaged over 10 runs.

4.1.4. Results and discussion

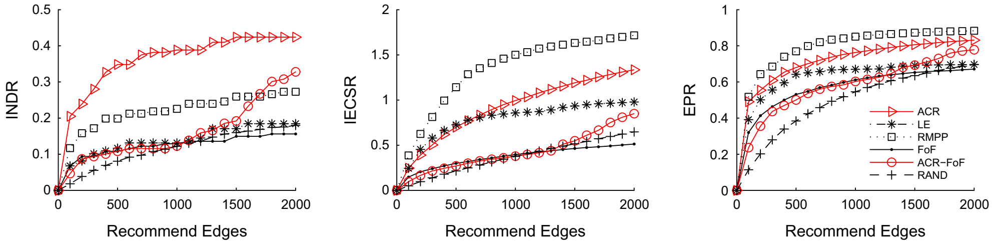
The experimental results for content spread are shown in Figs. 6–9. We also show three network examples after adding 2000 edges by three different algorithms (ACR, LE, RMPP), with newly added edges displayed in red color in Figs. 7 and 8. The following observations and conclusions can be drawn from the experimental results:

(1) It can be seen from the experiments that ACR outperforms LE and RMPP in reducing the network Diameter. This can be partly explained by Eq. (12), which indicates that increasing algebraic connectivity tends to reduce network diameter $D(G)$. This is further verified by the examples in Figs. 7 and 8, where we can see ACR is more likely to connect *external pairs*.
(2) The examples in Figs. 7 and 8 reveal that LE tends to improve the connection within a single core, while ACR and RMPP will improve the connections in different cores. However, a closer observation into Fig. 8 reveals that ACR also tries to connect different cores, while this is not observed in LE and RMPP. We can thus conclude that ACR will more likely reduce homophily in a network comparing with LE and RMPP. The experimental results in Fig. 6 also confirms that LE will have a poor performance in multi-core network since it tends to increase connection within a single core. In contrast, both ACR and RMPP achieve better performance in 2-Core RTG Networks.
(3) The most important observation from the experiments is that ACR-FoF outperforms RAND in almost all the evaluation metrics. Although the candidate edges are restricted to be 2-hop edges for friend recommendation, ACR-FoF's superiority over RAND demonstrates that ACR-FoF can effectively improve content spread in a social network.

The computer we use to run the experiment has a 4-core 3.20 GHz CPU and 4 GB memory in total. Fig. 9 shows the time needed to calculate scores for all the candidate edges in a network using RMPP, ACR and LE. On average, RMPP algorithm needs 22.3027 s for one of the RTG Network test cases and 210.4734 s for 2-Core RTG networks, while ACR algorithm only needs 0.0452 s and 0.2122 s, which is a significant improvement in terms of time complexity. The reason is that in ACR algorithm, we only need to calculate the second smallest eigenvector v_2 for once, then we can calculate the approximation term $(v_{2a} - v_{2b})^2$ in $O(1)$ time after adding a new edge to the original network, while in RMPP algorithm, we need to calculate the approximate *expected content spread (ECS)* after adding a new edge, which needs $O(n^2)$ time.

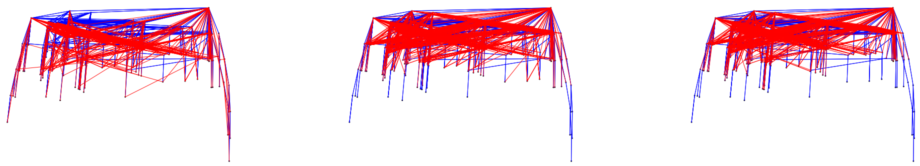


(a) RTG Network (b) RTG Network (c) RTG Network



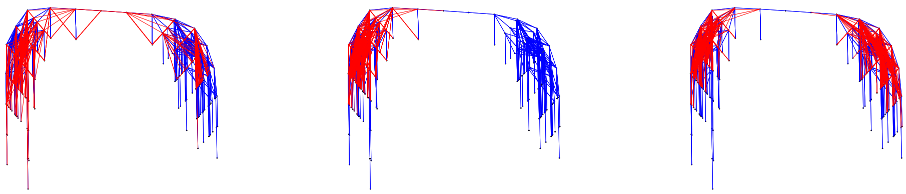
(d) 2-Core RTG Network (e) 2-Core RTG Network (f) 2-Core RTG Network

Fig. 6. Content spread capability experiment on simulated datasets.



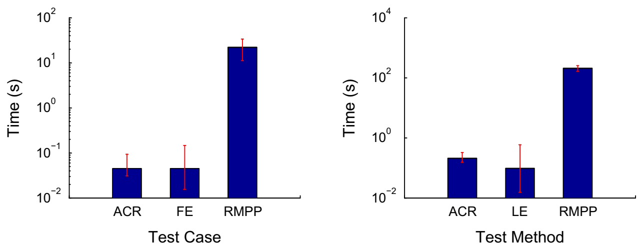
(a) ACR (b) LE (c) RMPP

Fig. 7. RTG Network after adding 2000 edges, blue edges are original connections and red edges are new ones. (For interpretation of the references to color in this figure legend, the reader is referred to the web version of this article.)



(a) ACR (b) LE (c) RMPP

Fig. 8. 2-Core RTG Network after adding 2000 edges, blue edges are original connections and red edges are new ones. (For interpretation of the references to color in this figure legend, the reader is referred to the web version of this article.)



(a) RTG Network (b) 2-Core RTG Network

Fig. 9. Time needed for calculate scores.

4.2. Real social networks experiments

In this part, we experiment on real-world social network data sets to test the effectiveness of our algorithm in content spread and recommending friends in social networks. We start with a description of the data sets.

4.2.1. Data sets

The data sets used in the following experiments are from Flickr and Twitter. As the networks used in our experiments are supposed to be connected, we select 10 connected subgraphs from Flickr and Twitter data sets using breadth-first-search (BFS) as follows: (1) starting with an empty node set; (2) randomly selecting a node and inserting it in into the empty node set; (3) using breadth-first-search to reach a neighbor of the inserted node and with probability $p_{select} = 0.5$ inserting this neighbor node into the node set; and (4) continuing BFS search and inserting new nodes into the node set until it contains more than 1000 nodes. We obtain 20 connected networks from the Flickr and Twitter data sets using these steps (10 from each).

For each of the 20 connected network we obtained, we delete a subset of its edges and then run friend recommendation algorithms to see whether these deleted edges are selected by the recommendation algorithms. The edges in a connected network are deleted as follows: (1) randomly selecting 3 nodes that are mutually connected; (2) randomly deleting an edge from the 3 edges connecting the 3 nodes; and (3) repeating the previous two steps until 500 edges are removed from a connected network. For each of the 20 connected networks we obtained, we repeat the edge-deleting process 10 times to create 10 new connected networks. Thus finally we obtain 200 connected networks in our experiments. In total, we use 9805 distinct nodes in Flickr, and 10,175 distinct nodes in Twitter. The statistics are summarized in Tables 6 and 7.

4.2.2. Evaluation metrics

To evaluate the friend recommendation success rate, we use the deleted edges for each of the 200 connected networks to calculate the *Precision*, *Recall* and *F*1 score as follows:

$$Precision = \frac{|E_r \cap E_d|}{|E_r|} \tag{35}$$

$$Recall = \frac{|E_r \cap E_d|}{|E_d|} \tag{36}$$

$$F_1 = \frac{2 \times Precision \times Recall}{Precision + Recall} \tag{37}$$

where E_d is set of formerly deleted edges, E_r is the set of recommended edges, and function $|A|$ is the number of elements in set A. Thus we can use these metrics to evaluate the accuracy of the recommendations. Also Improved Network Diameter Ratio (INDR), Improved Expected Content Spread Ratio (IECSR) and Enhanced Pairs Ratio (EPR) need to be tested for the content spread capability.

In the real social networks experiments, we need to consider both improvement in content spread capability and friend recommendation success rate. So we only add a new edge to the network if it is one of formerly deleted edges, which means a successful recommendation. The content spread capability is only calculated on the success edges.

4.2.3. Compared algorithms

In this section, we will make comparisons among the following recommendations algorithms, which are the combinations between content spread algorithms and the traditional friend recommendation method. The algorithms tested in our experiments include:

FoF: We test the traditional friend recommendation method *friends-of-friends*, which recommend new friends based on the number of mutual friends. For each candidate edge l, we use the number of mutual friends between the two end nodes i, j of the edge l to calculate its FoF score as follows:

$$Score_{FoF}(l) = |\{x|(x,i) \in E \text{ and } (x,j) \in E\}| \tag{38}$$

The following are regularized friend recommendation algorithms, where the three content spread algorithms are used as the regularizers.

ACR-FoF: FoF is regularized by **ACR** as follows:

$$Score_{ACR-FoF}(l) = Score_{FoF}(l) + \alpha \log(Score_{ACR}(l)) \tag{39}$$

LE-FoF: **FoF** is regularized by **LE** as follows:

$$Score_{LE-FoF}(l) = Score_{FoF}(l) + \alpha \log(Score_{LE}(l)) \tag{40}$$

RMPP-FoF: **FoF** is regularized by **RMPP** as follows:

$$Score_{RMPP-FoF}(l) = Score_{FoF}(l) + \alpha \log(Score_{RMPP}(l)) \tag{41}$$

Table 6
Statistics for Flickr.

	Minimum	Maximum	Average
Nodes	1009	1272	1085.9
Edges	2442	5970	3764.3
Diameter	5	8	6.1300
Mean distance	2.7552	3.5812	3.0095
Average degree	4.7510	10.5664	6.8980
Maximum degree	170	465	389.66

Table 7
Statistics for Twitter.

	Minimum	Maximum	Average
Nodes	1007	1769	1192.2
Edges	5205	27,133	13,804
Diameter	3	9	5.6700
Mean distance	2.1368	3.1311	2.6169
Average degree	9.8580	39.3842	22.5183
Maximum degree	335	1265	654.28

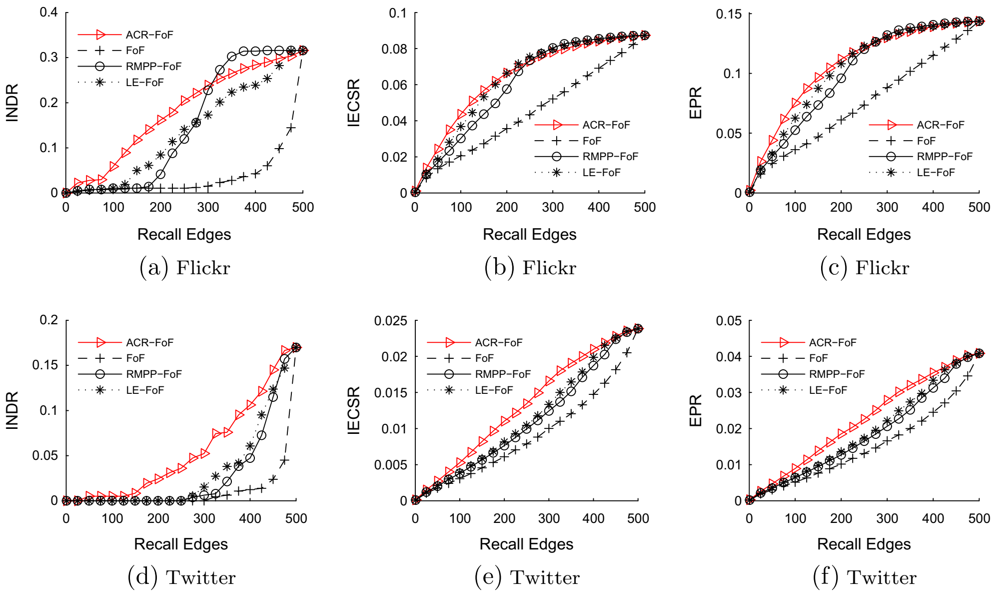


Fig. 10. Content spread capability experiment on real-world datasets.

where the regularization parameter α is set to 0.5 for all algorithms and we use the same method in *Content Spread Experiments* to calculate $Score_{ACR}(l), Score_{LE}(l)$ and $Score_{RMPP}(l)$, as Eqs. (32)–(34).

4.2.4. Results and discussion

The results are shown in Figs. 10 and 11. First thing we notice in the experimental results is that ACR-FoF excels in all the three content spread metrics, with obviously leads over other algorithms observed. This confirms our assumption that the real social network is multi-core and ACR-FoF can achieve best performance in multi-core network.

We can also see that, traditional friend recommendation algorithms such as *FoF* do not take content spread into consideration and thus will result in poor performance on the 3 evaluation metrics for content spread capability, as shown in Fig. 10. From the experimental results, we can also see that the three regularized algorithms gain more improvements over FoF in Flickr data set than in Twitter data set. The reason for that can be found from the statistics in Tables 6 and 7, which shows that Twitter data set is more densely connected and thus adding new edges will have less effect. Another important result from the experiments is that ACR-FoF gains a significant improvement in content spread capability than FoF with only a very tiny loss in recommendation success rate.

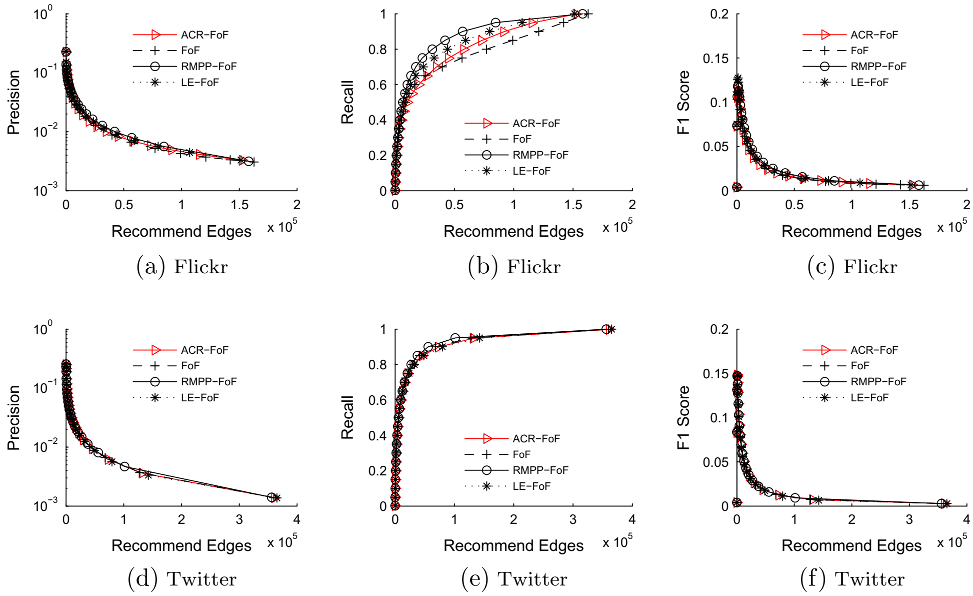


Fig. 11. Recommendation success rate.

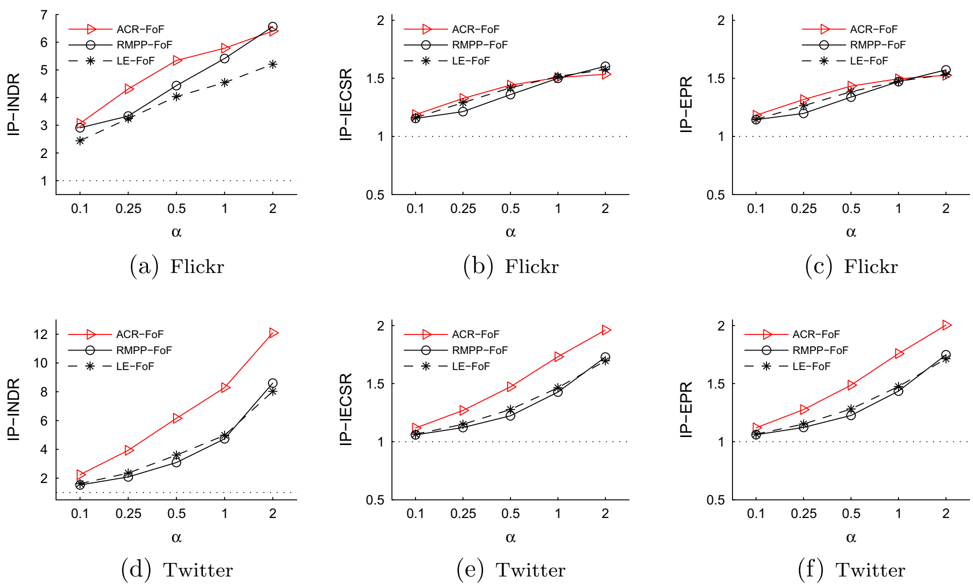


Fig. 12. Content spread capability vs. regularization parameter α.

After taking logarithms on the three regularizers, FoF and the three regularized recommendation algorithms show relatively similar performance in friend recommendation success rate, in terms of Precision, Recall and F_1 Score. The reason we take logarithms on the regularizer is that we consider success rate as the most important objective in recommending friends in social network. The experimental results also verify that we have achieved a good balance between friend recommendation success rate and content spread capability in our algorithm.

It will be interesting to examine how the performance of the above algorithms change with varying regularization parameter values, so as to check the algorithms' sensitivity to α. We performed experiments with α varying from 0.1 to 2. Figs. 12 and 13 shows how the performance of three regularized FoF methods vary with different α. Note that here the vertical axis is *Improvement Percentage*, the ratio between the size of the area under each curve and the size of the area under FoF curve, which we use here to characterize the general performance of a specific algorithm under a certain α value. The value more than 1 means that this algorithm is better than FoF on testing metric, and higher values mean better performance.

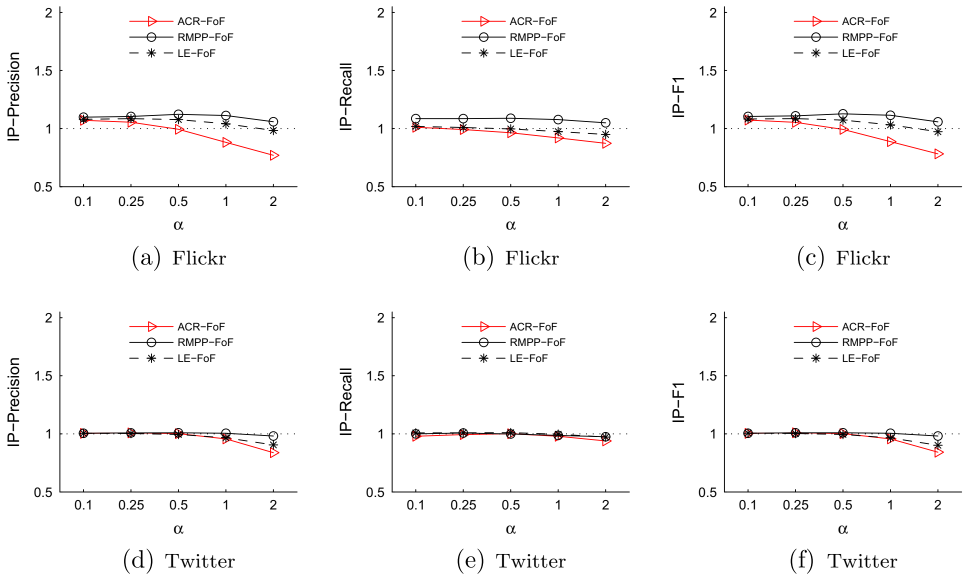


Fig. 13. Recommendation success rate vs. regularization parameter α.

From Fig. 12 we can see that as α increases, all the three algorithms show steady increase in content spread capabilities as indicated by *IP-INDR*, *IP-IECSR* and *IP-EPR*. Among them, ACR-FoF achieves the largest improvement. We can also obviously see from the experimental results that ACR-FoF can achieve more significant improvement than the other two algorithms in content spread in more densely connected networks such as Twitter.

From Fig. 13 we can see that as α increases, all the three algorithms experience some drop in recommendation success rate, as indicated by *IP-Precision*, *IP-Recall* and *IP-F1*. The difference between Flickr and Twitter data set can once again be attributed to varying densities of connections in two social network data sets. For a more densely connected social network such as Twitter, the friends-of-friends part of the score tends to be high and thus is less affected by the regularization term.

It can be seen when $\alpha = 0.5$, the loss in recommendation success rate is tiny but the improvement in content spread is big. Thus in the content spread experiments and recommendation experiments, we set $\alpha = 0.5$.

5. Conclusions and future works

In this paper,we present a novel Algebraic Connectivity Regularized Friend Recommendation algorithm (ACR-FoF) for recommending friends in social networks. Different from traditional recommendation methods that mainly consider success rate in recommendation, ACR-FoF takes into account both recommendation success rate and content spread in a social network. Experimental results on synthetic data sets and real social network data sets show that ACR-FoF achieves significant improvement in content spread at a very tiny loss on recommendation accuracy. We believe ACR-FoF can greatly enhance the value of a social network by boosting its power in spreading contents.

There are several interesting problems to be investigated in our future work: (1) The expected content spread (ECS(G)) proposed in this paper is based on the information cascade (IC) model. But optimization based on algebraic connectivity is not limited to IC model. It will be interesting to generalize our work to other information dissemination models by considering weighted assignment of propagation probability among nodes, or defining new content spread metrics and deriving their relations to the algebraic connectivity, etc. (2) It will be meaningful to explore more on how local and global network structures are related to the algebraic connectivity of the network. We can further refine our algorithm to improve content spread by more explicitly considering the underlying network structure.

Acknowledgements

This work is supported by National Science Foundation of China (Grant Nos. 61173185, 61173186) and Zhejiang Provincial Natural Science Foundation of China (Grant No: LZ13F020001).

References

[1] L. Akoglu, C. Faloutsos, Rtg: a recursive realistic graph generator using random typing, Data Min. Knowl. Discov. 19 (2009) 194–209.
[2] E. Bakshy, I. Rosenn, C. Marlow, L. Adamic, The role of social networks in information diffusion, in: Proceedings of the 21st International Conference on World Wide Web, ACM, 2012, pp. 519–528.

[3] M. Cha, A. Mislove, K. Gummadi, A measurement-driven analysis of information propagation in the Flickr social network, in: Proceedings of the 18th International Conference on World Wide Web, ACM, 2009, pp. 721–730.
[4] V. Chaoji, S. Ranu, R. Rastogi, R. Bhatt, Recommendations to boost content spread in social networks, in: Proceedings of the 21st International Conference on World Wide Web, ACM, 2012, pp. 529–538.
[5] J. Chen, W. Geyer, C. Dugan, M. Muller, I. Guy, Make new friends, but keep the old: recommending people on social networking sites, in: Proceedings of the 27th International Conference on Human Factors in Computing Systems, ACM, 2009, pp. 201–210.
[6] W. Chen, C. Wang, Y. Wang, Scalable influence maximization for prevalent viral marketing in large-scale social networks, in: Proceedings of the 16th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, ACM, 2010, pp. 1029–1038.
[7] W. Chen, Y. Wang, S. Yang, Efficient influence maximization in social networks, in: Proceedings of the 15th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, ACM, 2009, pp. 199–208.
[8] E. Demaine, M. Zadimoghaddam, Minimizing the diameter of a network using shortcut edges, in: Algorithm Theory-SWAT 2010, 2010, pp. 420–431.
[9] P. Domingos, M. Richardson, Mining the network value of customers, in: Proceedings of the Seventh ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, ACM, 2001, pp. 57–66.
[10] Y. Dong, J. Tang, S. Wu, J. Tian, N.V. Chawla, J. Rao, H. Cao, Link prediction and recommendation across heterogeneous social networks, in: Data Mining (ICDM), 2012 IEEE 12th International Conference on, IEEE, 2012, pp. 181–190.
[11] A. Ghosh, S. Boyd, Growing well-connected graphs, in: Decision and Control, 2006 45th IEEE Conference on, IEEE, 2006, pp. 6605–6611.
[12] Z. Guan, J. Bu, Q. Mei, C. Chen, C. Wang, Personalized tag recommendation using graph-based ranking on multi-type interrelated objects, in: Proceedings of the 32Nd International ACM SIGIR Conference on Research and Development in Information Retrieval, ACM, 2009, pp. 540–547.
[13] Z. Guan, G. Miao, R. McLoughlin, X. Yan, D. Cai, Co-occurrence-based diffusion for expert search on the web, IEEE Trans. Knowl. Data Eng. 25 (2013) 1001–1014.
[14] Z. Guan, C. Wang, J. Bu, C. Chen, K. Yang, D. Cai, X. He, Document recommendation in social tagging services, in: Proceedings of the 19th International Conference on World Wide Web, ACM, 2010, pp. 391–400.
[15] I. Guy, I. Ronen, E. Wilcox, Do you know? Recommending people to invite into your social network, in: Proceedings of the 14th International Conference on Intelligent User Interfaces, ACM, 2009, pp. 77–86.
[16] J. Herlocker, J. Konstan, J. Riedl, Explaining collaborative filtering recommendations, in: Proceedings of the 2000 ACM Conference on Computer Supported Cooperative Work, ACM, 2000, pp. 241–250.
[17] D. Kempe, J. Kleinberg, Tardos, Maximizing the spread of influence through a social network, in: Proceedings of the Ninth ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, ACM, 2003, pp. 137–146.
[18] D. Kempe, J. Kleinberg, Tardos, Influential nodes in a diffusion model for social networks, Automata, Lang. Program. (2005) 99.
[19] Y. Kim, M. Mesbahi, On maximizing the second smallest eigenvalue of a state-dependent graph Laplacian, IEEE Trans. Autom. Control 51 (2006) 116–120.
[20] D. Liben-Nowell, J. Kleinberg, The link-prediction problem for social networks, J. Am. Soc. Inform. Sci. Technol. 58 (2007) 1019–1031.
[21] C. Lin, R. Xie, X. Guan, L. Li, T. Li, Personalized news recommendation via implicit social experts, Inform. Sci. 254 (2014) 1–18.
[22] J. Liu, C. Wu, Y. Xiong, W. Liu, List-wise probabilistic matrix factorization for recommendation, Inform. Sci. 278 (2014) 434–447.
[23] A. Meyerson, B. Tagiku, Minimizing average shortest path distances via shortcut edge addition, in: Approximation, Randomization, and Combinatorial Optimization. Algorithms and Techniques, Springer, 2009, pp. 272–285.
[24] B. Mohar, Eigenvalues, diameter, and mean distance in graphs, Graphs Combin. 7 (1991) 53–64.
[25] S. Oyama, K. Hayashi, H. Kashima, Cross-temporal link prediction, in: Data Mining (ICDM), 2011 IEEE 11th International Conference on, IEEE, 2011, pp. 1188–1193.
[26] B.A. Prakash, D. Chakrabarti, M. Faloutsos, N. Valler, C. Faloutsos, Threshold conditions for arbitrary cascade models on arbitrary networks, in: Data Mining (ICDM), 2011 IEEE 11th International Conference on, IEEE, 2011, pp. 537–546.
[27] E. Rogers, Diffusion of innovations, Simon and Schuster, 1995.
[28] R. Schifanella, A. Barrat, C. Cattuto, B. Markines, F. Menczer, Folks in folksonomies: social link prediction from shared metadata, in: Proceedings of the Third ACM International Conference on Web Search and Data Mining, ACM, 2010, pp. 271–280.
[29] H. Tong, B.A. Prakash, T. Eliassi-Rad, M. Faloutsos, C. Faloutsos, Gelling, and melting, large graphs by edge manipulation, in: Proceedings of the 21st ACM International Conference on Information and Knowledge Management, ACM, 2012, pp. 245–254.
[30] S. Toueg, K. Steiglitz, The design of small-diameter networks by local search, IEEE Trans. Comput. 100 (1979) 537–542.
[31] Y. Wang, D. Chakrabarti, C. Wang, C. Faloutsos, Epidemic spreading in real networks: an eigenvalue viewpoint, in: Reliable Distributed Systems, 2003. Proceedings. 22nd International Symposium on, IEEE, 2003, pp. 25–34.
[32] Y. Yang, N. Chawla, Y. Sun, J. Hani, Predicting links in multi-relational and heterogeneous networks, in: Data Mining (ICDM), 2012 IEEE 12th International Conference on, IEEE, 2012, pp. 755–764.
[33] E. Yom-Tov, F. Diaz, Out of sight, not out of mind: on the effect of social and physical detachment on information need, in: Proceedings of the 34th International ACM SIGIR Conference on Research and Development in Information Retrieval, Dover, 2011, pp. 385–394.
[34] S.J. Yu, The dynamic competitive recommendation algorithm in social network services, Inform. Sci. 187 (2012) 1–14.
[35] B. Zhang, X. Guan, M.J. Khan, Y. Zhou, A time-varying propagation model of hot topic on BBS sites and Blog networks, Inform. Sci. 187 (2012) 15–32.