doi:10.11959/j.issn.1000-436x.2015035

基于局部近邻传播及用户特征的社区识别算法

郭昆[1]，郭文忠[1]，邱启荣[2]，张岐山[2]
(1. 福州大学 数学与计算机科学学院, 福建 福州 350108; 2. 福州大学 管理学院, 福建 福州 350108)

摘 要：提出一种将局部近邻传播和考虑用户特征的相似性测度相结合实现社交网络中的社区识别的算法。一方面，通过放松代表点约束条件及限制消息传播范围为节点的局部近邻，算法在降低时间和空间复杂度的同时保持较小的识别精度损失，从而能够适应社交网络挖掘需要；另一方面，通过将节点的拓扑相似度和特征相似度相结合来描述节点的综合相似度，使算法能够适应社交网络采样数据中用户关联信息不完整的情况。通过在人工数据集和真实数据集上的对比实验表明，所提方法不仅具有近似线性的时间复杂度及线性的空间复杂度，而且在网络中的节点关联边信息不完整时仍保持较好的识别精度。
关键词：社交网络；近邻传播；社区识别；聚类
中图分类号：TP393 **文献标识码**：A

Community detection algorithm based on local affinity propagation and user profile

GUO Kun[1], GUO Wen-zhong[1], QIU Qi-rong[2], ZHANG Qi-shan[2]
(1. College of Mathematics and Computer Science, Fuzhou University, Fuzhou 350108, China;
2. Management School, Fuzhou University, Fuzhou 350108, China)

Abstract: An algorithm based on local affinity propagation and a new similarity measure concerning user profile is proposed. On one hand, by loosening the exemplar constraint and requiring the messages propagate around a node's neighbors, the algorithm achieves lower time and space complexity without too much lost in clustering accuracy, which makes it adaptable to the mining of large-scale social networks. On the other hand, by designing a hybrid similarity measure based on the topological similarity and the profile similarity of the nodes, the algorithm can effectively tackle the situation of the social networks data without complete user relation information. The experimental results on the artificial datasets and the real-world datasets demonstrate that the algorithm not only has near-linear time complexity and linear space complexity, but also retains high detecting accuracy when handling incomplete networks.
Key words: social network; affinity propagation; community detection; clustering

1 引言

近年来，以微博、Facebook、Youtube 和 Twitter 等为代表的社交网络服务（SNS, social network service）在世界范围内得到迅速发展，越来越多人开始通过社交网络进行在线聊天、购物、聚会等活动。在社交网络中识别具有相近的年龄、背景、兴趣等特征的用户组成的社区，不仅在理论上是对复杂网络聚类研究的进一步深化和发展，在实践上对基于社交网络的搜索、推荐等商业应用也具有重要

收稿日期：2013-10-16；**修回日期**：2014-01-18
基金项目：国家自然科学基金资助项目(61103175, 61300104)；教育部科学技术研究重点基金资助项目(212086)；福建省科技创新平台建设基金资助项目(2009J10007)；福建省自然科学基金资助项目(2013J01230)；福建省高校杰出青年科学基金资助项目(JA12016)；福建省高等学校新世纪优秀人才支持计划基金资助项目(JA13021)
Foundation Items: The National Natural Science Foundation of China (61103175, 61300104); The Key Project of Chinese Ministry of Education (212086); The Technology Innovation Platform Project of Fujian Province (2009J10007); The Natural Science Foundation of Fujian Province (2013J01230); The Fujian Province High Science Fund for Distinguished Young Scholars (JA12016); The Program for New Century Excellent Talents in Fujian Province University (JA13021)

意义。因此，基于社交网络的社区识别已经成为数据挖掘领域的一个研究热点。

广义上，社交网络上的社区识别可以看作是一种复杂网络上的聚类[1]或图上的社区识别[2]。但社交网络也有一些其自身独有的特点。首先，社交网络通常具有庞大的用户群，例如，Facebook 和腾讯微博的用户数量均达到亿级。这对社交网络上的社区识别算法的时间和空间复杂度提出了更加严格的要求，通常要求算法的复杂度降低至接近线性才能有效处理具有亿级节点的网络，而目前具有线性复杂度的算法还不多见[2]。其次，社交网络具有显著的动态性，网络内部节点及节点之间的联系经常随时间、地点而不断变化，这意味着任何一次数据采样只能得到某一个时间段某一部分用户和用户关联信息的样本数据。目前，多数复杂网络聚类算法均假设数据分析在一个完整的网络上进行，当部分节点和边的信息缺失时，数据分析的准确性将受到不同程度的影响。最后，社交网络的一个显著特点是每个用户节点都包含描述用户的年龄、性别、爱好等详细特征的信息。当采样得到的网络数据中的节点或边信息不完整时，充分利用节点自身包含的特征信息帮助判断节点相似性是一种简单有效的提高社区识别精度的方法。目前，在社交网络分析领域，多数方法基于网络拓扑计算节点相似度，同时考虑节点特征相似度的研究成果还不多见。

针对上述社交网络的特殊性以及现有复杂网络聚类方法在处理这些问题时存在的不足，本文提出以近邻传播聚类为主要手段，通过放松代表点约束以及引入局部近邻传播机制，并利用节点特征信息辅助社区识别，设计一种具有线性复杂度且能够有效处理边结构不完整的社交网络的高效社区识别算法。

2　相关工作

目前，复杂网络聚类或图上的社区识别方法根据采用的求解策略，主要可以分为基于优化的方法和启发式方法。

基于优化的方法通过设置目标函数并迭代逼近函数最优值实现社区识别。谱方法[3]将社区识别问题转换为放松的二次型优化问题，通过求 Laplacian 矩阵的特征向量得到网络的近似最优划分。KL（Kernighan-Lin）算法[4]早期用于图的划分，优化目标为极小化社区内连接数与社区间连接数之差。FN（Fast Newman）算法[5]、GA（Guimera-Amaral）算法[6]和 EO（extremal optimization）算法[7]均以 Newman 和 Girvan 提出的模块度（modularity，又称 Q 函数）为优化目标。模块度的定义为

$$Q=\sum_i\left(e_{ii}-a_i^2\right)=Trace(\boldsymbol{e})-\left\|\boldsymbol{e}^2\right\| \tag{1}$$

其中，$\boldsymbol{e}$ 是一个对称矩阵，其元素 e_{ij} 表示社区 i 和社区 j 之间的边占全部边的比例，$Trace(\boldsymbol{e})$表示矩阵的迹，$a_i=\sum_j e_{ij}$ 表示与社区 i 内部节点相连的边数。由于模块度能够同时描述社区内部高内聚、社区之间低耦合的特性，具有清晰的物理含义，除了作为算法优化目标之外，它也成为评价算法性能的重要指标。基于 Potts 模型的算法[8]借助物理学中转子自旋态的概念，通过最小化描述系统能量的 Hamilton 算子实现网络的最优划分。淦文燕[9]等提出基于拓扑势场的高低划分群体的方法。何东晓[10]等提出一种将聚类融合与遗传算法（GA, genetic algorithm）结合的社区挖掘算法。叶镇清[11]等提出一种新的模块度 Qd 并采用迭代聚类法实现其最优化。林旺群[12]等提出新的基于层次社区树的社区结构模型，并构建相应的并行化社区发现算法。杨博[13]等提出基于局部搜索和模拟退火（SA, simulated annealing）的能够同时处理同配及异配网络的社区挖掘算法。韩毅[14]等提出一种基于密度估计的特征簇发现算法，能够发现社区间的层次结构。在这一类方法中，精确求解算法通常具有超线性的时间复杂度，需要将复杂度降低为线性，而近似求解算法需要克服收敛速度慢及易陷入局部最优解的问题，以适应象社交网络这样的大规模网络的数据分析。

启发式方法通过设置启发规则来寻找最优社区划分。GN（Girvan-Newman）算法[15]以社区间连接的边介数（betweeness）应大于社区内连接的边介数为启发规则，通过不断删除具有最大边介数的边来发现社区。WH（Wu- Huberman）算法[16]将复杂网络建模为电路系统来设计启发规则：当在不同社区中分别选取 2 个节点作为正负极后，由于社区间电阻远大于簇内电阻，相同社区节点势能应相近，而相异社区节点势能应有显著差别。CPM（clique percolation method）算法[17]的启发规则为：网络社区由多个相邻的 k-团（k-clique）组成，每个 k-团唯一属于一个社区，但不同社区的 k-团可能共享相邻节点，通过建立团-团重叠矩阵可以计算出重叠网络社区的结构。基于标签传播（label propaga-

tion）的算法[18]采用标签描述节点的社区信息，其启发规则为：不断在节点及其近邻间传递标签信息，经过多次迭代后，属于同一个社区的节点的标签将趋于一致。基于随机游走的算法[19]将社区结构的识别过程建模为图上的随机游走，其启发规则为：当网络存在明显社区结构时，随机游走 agent 在社区内节点间游走的概率要大于在社区间节点间游走的概率。对于此类方法，如何设计能够准确描述复杂网络中的社区结构特征的启发规则、提高算法的普适性仍是其面临的主要挑战。

目前，将网络拓扑特征与节点自身特征相结合进行社区识别的研究主要见于社交媒体分析（social media analysis），其典型代表是主题建模（topic modeling）。主题建模的目标是通过分析在一个用户群体之间交流的文本、图片等多媒体数据之间存在的相关性找出从属于不同主题的用户子群[19~22]。虽然与社交网络中的社区识别存在相似性，但是，主题建模的目标是建立隐藏在用户交流数据背后的主题模型，其社区划分以主题为中心，而不以用户为中心，一个用户可以加入多个主题群，当主题变动时社区也随之变化，因此，不能直接将主题建模方法应用于社交网络中的社区识别。不过，受到主题建模思想的启发，已经有学者开始将网络拓扑特征与节点特征结合应用于描述节点的综合相似度，从而更好地在社交网络中识别社区。例如，Yoshida 设计了一种考虑节点特征相似度的复合相似度，在具有不同边缺失比的复杂网络上的实验取得良好效果[23]，但其采用的谱方法具有较高的时间复杂度。McAuley[24]等提出同时考虑节点的拓扑相似度和特征相似度的社交圈子模型，并应用于设计在个人社交网络中识别不同的社交圈子的算法。但由于采用了基于统计的方法，算法的时间开销较大。总体而言，这方面的研究还处于起步阶段。

3 近邻传播

近邻传播（AP, affinity propagation）算法是一种通过在近邻节点间传播消息实现聚类的方法[25]，是圈信任传播（loopy belief propagation）[26]在聚类方面的最新应用。AP 算法通过多次迭代使簇中心点（或代表点）逐渐显现，因此不需要预先输入簇数参数。此外，AP 算法不要求节点具有对称相似度，因此能够适应相似性测度不满足三角不等式的应用。

AP 算法需要输入相似度矩阵 $\boldsymbol{S}$。矩阵元素 $s(i,k)$ 表示点 x_k 与点 x_i 的相似度。文献[25]采用 2 个节点之间的欧式距离作为其相似度，即 $s(i,k)=-|x_i-x_k|^2$ 。相似度矩阵 $\boldsymbol{S}$ 的主对角线元素 $s(k,k)$具有特别含义：它表示节点 x_k 适合作为代表点的程度。$s(k,k)$的值越大，节点 k 被选为代表点的可能性就越大。AP 算法中将所有的 $s(k,k)$设为一个共同值 p。因此，p 是 AP 算法的一个非常重要参数，直接影响到最终生成的簇的数量。

AP 算法在节点之间传播的消息分为支持度消息(responsibility)和适选度消息(availability)。前者由矩阵 $\boldsymbol{R}=r(i,k)$描述，$r(i,k)$表示节点 i 向节点 k 发送的消息，反映节点 i 在考虑其他潜在代表点后对节点 k 作为其代表点的支持程度。后者由矩阵 $\boldsymbol{A}=a(i,k)$描述，$a(i,k)$表示节点 k 向节点 i 发送的消息，反映节点 k 综合了其他点对其支持度后向节点 i 表明自己作为节点 i 的代表点的适合程度。近邻传播过程即表现为 2 个消息矩阵的交替更新。每次更新后，通过计算决策矩阵 $\boldsymbol{E}=\boldsymbol{R}+\boldsymbol{A}=\boldsymbol{e}(i,k)$确定节点 i 的代表点。消息更新公式如下。

$$r(i,k)\leftarrow s(i,k)-\max_{k's.t.k'\neq k}\{a(i,k')+s(i,k'))\} \quad (2)$$

$$a(i,k)\leftarrow\begin{cases}\min\{0,r(k,k)+\sum_{i's.t.i'\neq\{i,k\}}\max\{0,r(i',k)\}\}, & i\neq k\\ \sum_{i's.t.i'\neq\{i,k\}}\max\{0,r(i',k)\}, & i=k\end{cases} \quad (3)$$

4 结合局部近邻传播及用户特征的社区识别

4.1 局部近邻传播与代表点约束放松

与互联网、生物网络等其他复杂网络不同，社交网络中的社区是通用户与其近邻（包括亲戚、朋友、同事等）间的不断交互逐渐产生并发展的。这与 AP 算法通过在近邻间传播消息，以使社区结构自然涌现的设计思想存在相似性。但是，直接将 AP 算法应用于社交网络中的社区识别存在一些困难。首先，AP 算法中的消息是在所有节点之间传播的，而社交网络中的用户一般仅与其相近用户传递信息。2 个距离较远的用户直接进行消息传递的概率很低。其次，AP 算法要求每个代表点必须选择自身作为其代表点（又称为代表点约束）。文献[27]发现这限制了其聚类精度的进一步提高，并提出放松这一约束以得到更低的聚类错误率。最后，AP 算法的时间复杂度为 $O(n^2)$，n 为网络节点数。当应

用于像社交网络这样规模较大的数据集时，其时间开销较大。

通过将消息传播范围局限于节点的直接近邻，一方面使算法的运行过程更符合社交网络中社区的形成与发展过程，从而有可能得到具有更高内聚的社区；另一方面，更重要的是，由于每个节点发送和接收的消息数由 $O(n)$下降为 $O(d_{\max})$，$d_{\max}$ 为节点最大度，算法的时间复杂度将下降为 $O(d_{\max}n)$，具体参见 4.5 节的分析。此时，消息更新公式应修改为

$$r(i,k) \leftarrow s(i,k) - \max_{k's.t.k' \neq k \cap k' \in NB(i)} \{a(i,k') + s(i,k')\} \tag{4}$$

$$a(i,k) \leftarrow \begin{cases} \min\{0, r(k,k) + \sum_{i's.t.i' \neq \{i,k\} \cap i \in NB(k)} \max\{0, r(i',k)\}\}, & i \neq k \\ \sum_{i's.t.i' \neq \{i,k\} \cap i \in NB(k)} \max\{0, r(i',k)\}, & i = k \end{cases} \tag{5}$$

其中，$NB(i)$和 $NB(k)$分别表示节点 i 和节点 k 的近邻节点集。

从另一方面来看，将消息传播范围限制于节点近邻也会导致近邻消息不能充分在全网节点之间传播。但是，正如前面指出的，针对社交网络这种特殊的复杂网络，远程节点之间的联系远少于近邻节点之间的联系。因此，可以预期远程节点之间交换的消息对社区生成过程的影响很小。这一假设也在实验中得到了验证，具体实验结果的分析参见第 5 节。

在生成代表点时，放松代表点约束，即允许代表点选择其自身以外的其他节点作为其代表点，能够进一步提高聚类的精确度。但是，带来的问题是可能导致代表点间出现循环代表，形成圈结构。例如，节点 1 的代表点为节点 2，节点 2 的代表点为节点 3，而节点 3 的代表点又是节点 1，如图 1 所示。

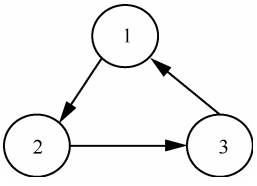


图 1　代表点的圈结构

代表点之间的这种圈结构将使近邻传播过程陷入无限循环。一个可行的解决方案是从某个节点位置切断其与代表点的连接，使圈变为路径，同时修改路径上除最末代表点外所有其他节点的代表点。图 1 中的圈结构从节点 1 处切断后重新调整的结果如图 2 所示。

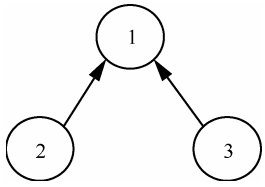


图 2　消除圈结构后的代表点

4.2　考虑用户特征的相似度

由于社交网络规模庞大且变化频繁，在进行数据采样时通常只能得到网络的部分信息。这样，在进行数据分析时，由于部分节点及边信息的缺失，单纯依赖于网络拓扑计算节点相似度的方法可能造成分析结果的不准确。借鉴文献[23]的思想，利用社交网络通常包含用户特征描述信息的优点，将基于网络拓扑的相似度与基于用户特征的相似度相结合，设计一种新的组合相似度，其计算公式如下

$$s_t(i,j) = \frac{NB(i) \bigcap NB(j)}{NB(i) \bigcup NB(j)} \tag{6}$$

$$s_p(i,j) = \frac{\boldsymbol{r}_i \boldsymbol{r}_j}{|\boldsymbol{r}_i||\boldsymbol{r}_j|} \tag{7}$$

$$s(i,j) = \alpha s_t + (1-\alpha) s_p \tag{8}$$

其中，$s_t(i,j)$描述节点的拓扑相似度，其实质上是计算 2 个节点的近邻集的 Jaccard 相似度。$s_p(i,j)$描述 2 个节点的特征向量 $\boldsymbol{r}_i$ 和 $\boldsymbol{r}_j$ 的相似度，这里采用余弦公式计算。节点的特征向量是一个二值向量，$\boldsymbol{r}_i=(r_{ik})$，当节点 i 具有特征 k 时，$r_{ik}=1$，否则 $r_{ik}=0$。最后，节点的组合相似度根据式(8)计算，其中，加权系数 $\alpha \in [0,1]$ 调整拓扑相似度和特征相似度在组合相似度中所占的比重。

4.3　算法总体框架

在网络规模较大且消息传播范围仅限于节点直接近邻时，即使有参数 p 的约束，一次局部近邻传播仍可能产生过多的代表点，使网络被划分成大量规模极小的社区。此时，通过将生成的代表点及代表点间的连接边看作一个新的超网（super network），在这个网络上再次运行局部近邻传播生成新的代表点，可以实现社区数量的进一步压缩。这个过程可以迭代进行，直到生成的代表点不再发生

变化。

综合前面的分析，提出一种结合局部近邻传播及考虑用户特征的相似度的新算法 LAP（local affinity propagation），其总体框架如下。

1) 应用近邻传播求网络 G 的代表点集，消息更新公式采用修改后的式(4)和式(5)。

2) 若当前得到的代表点与上次得到的代表点不一致，则建立由新的代表点及代表点间的连接边组成的超网 G'，返回步骤 1)继续执行。

3) 否则，消除代表点中可能存在的圈结构，输出得到的社区，算法结束。

4.4 算法实现

LAP 算法的具体实现如下。

算法 1 LAP 算法

输入：网络 $G=(V,E,R)$，V 为节点集，E 为边集，R 为节点特征向量集，最大迭代次数 $maxIter$，稳定迭代次数 $convIter$，p，α

输出：社区集 $Set_c=\{C_1,\cdots,C_k\}$，k 为社区数

1.根据式(8)计算节点与其近邻节点的相似度，为每个节点生成相似度向量 $\boldsymbol{s}_i=(s_{ij})$；

2.WHILE true

3. FOR i=1 to $maxIter$

4. FOR j=1 to $|V|$

5. FOR k=1 to $|NB(j)|$

6. 根据式(4)更新节点 j 的支持度向量 $\boldsymbol{r}_j$ 关于近邻 k 的分量 r_{jk}；

7. 根据式(5)更新节点 j 的适选度向量 $\boldsymbol{a}_j$ 关于近邻 k 的分量 a_{jk}；

8. END FOR

9. END FOR

10. 计算决策向量 $\boldsymbol{e}_i=\boldsymbol{r}_i+\boldsymbol{a}_i$，选择节点 i 的代表点；

11. IF 所有节点的代表点在最近 $convIter$ 次迭代中均没有变化 THEN

12. BREAK；

13. END IF

14. END FOR

15. 创建指示向量 $\boldsymbol{v}=(v_i)$，v_i 为节点 i 的代表点；

16. 调用过程 RemoveCircle 消除指示向量 $\boldsymbol{v}$ 中可能存在的圈结构；

17. IF 当前循环的代表点与上次循环得到的代表点完全相同 THEN

18. BREAK；

19. END IF

20. 修剪节点集 V 和边集 E，仅保留代表点及代表点间的连接边

21.END WHILE

22.根据指示向量 $\boldsymbol{v}$ 将节点划分至各个社区，建立社区集 $Set_c=\{C_1,\cdots,C_k\}$，k 为社区数；

23.return Set_c。

算法最外层的 WHILE 循环反复执行局部近邻传播，直至生成的代表点不再变化。步骤 3 至步骤 14 的 FOR 循环是局部近邻传播的实现。这里，由于消息传播的范围限制为节点的直接近邻，不再需要保留支持度矩阵 $\boldsymbol{R}$ 和适选度矩阵 $\boldsymbol{A}$，而只需要为每个节点 i 保留一个支持度向量 $\boldsymbol{r}_i$ 和一个适选度向量 $\boldsymbol{a}_i$，以存储节点接收到和准备发送的消息。这样，算法的空间复杂度可以大幅度下降。步骤 17 的判断决定外层的 WHILE 循环何时可以结束，若生成的代表点仍变化，步骤 20 通过修剪原始网络 G 得到只是包含代表点及其连接边的超网 G'，为下一次迭代做准备。在根据指示向量 $\boldsymbol{v}$ 确定节点所属的社区时，需要先消除向量 $\boldsymbol{v}$ 中可能存在的圈结构，步骤 16 调用过程 RemoveCircle 完成这一操作。算法最后返回根据指示向量 $\boldsymbol{v}$ 建立的以各个代表点为中心的社区集。

过程 RemoveCircle 的具体实现如下。

RemoveCircle 过程

输入：节点指示向量 $\boldsymbol{v}=(v_i)$

1.FOR 节点 $i\in V$

2. 初始化散列表 H；

3. $j=i$；

4. WHILE $j\neq v_i$

5. IF H 中已经存在节点 j THEN

6. BREAK；

7. END IF

8. 将节点 j 加入 H；

9. $j=v_j$；

10. END WHILE

11. IF H 非空 THEN

12. FOR 节点 $k\in H$

13. $v_k=j$；

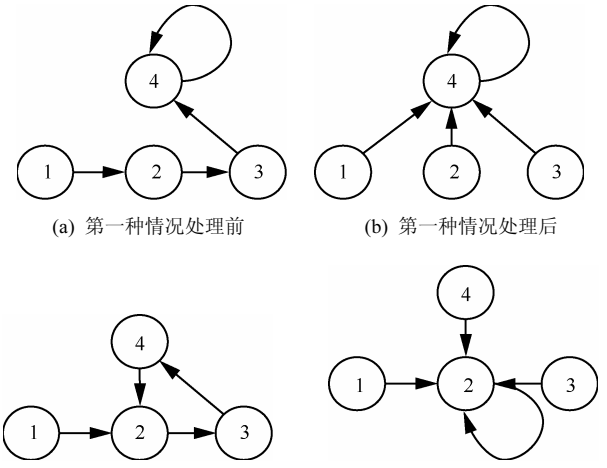
14. END FOR

15. END IF

16.END FOR

针对每个节点 i，过程 RemoveCircle 首先判

断节点 i 的代表点是否是其自身的代表，如果是，则散列表 H 为空，不存在圈结构。否则，从节点 i 到其真实代表点存在一条路径。这里，真实代表点有 2 种情况：一是最终找到以一个以其自身为代表的代表点，此时不存在圈结构，只需要执行步骤 11 至步骤 15 将路径上的所有节点的代表点都修改为该代表点即可；二是找到一个已经在路径中出现过的代表点，此时出现圈结构，在步骤 11 至步骤 15 中必须将该节点连同路径上的所有结构的代表点都修改为一个指定的代表点（过程 RemoveCircle 中设为发现的重复节点）。2 种情况的处理如图 3 所示。



(a) 第一种情况处理前　(b) 第一种情况处理后

(c) 第二种情况处理前　(d) 第二种情况处理后

图 3　圈结构的处理

可以证明，过程 RemoveCircle 能够消除代表点中存在的圈结构。

定理 1　经过过程 RemoveCircle 处理后的代表点集合不存在圈结构。

证明　设代表点集合为 $E=\{e_1,\cdots,e_n\}$

不失一般性，设子集 $P=\{p_1,\cdots,p_{n_p}\}\subseteq E, n_p=|P|$ 存在圈结构。经过过程 RemoveCircle 的处理，P 中所有节点的代表点都设为其真实代表点 e_s，圈结构被消除。此时考虑任一未处理的代表点 $e_x\in E-P$，e_x 的代表点有以下 3 种可能的情况。

1) e_x 的代表点为其自身。此时，不存在圈结构，也不需要过程 RemoveCircle 处理。

2) e_x 的代表点 $e_y\in P$。此时，由于 P 中的代表点统一指向一个代表点 e_k，不存在圈结构，过程 RemoveCircle 按第一种情况处理，将 e_x 的代表点修改为 e_k。

3) e_x 的代表点 $e_y\in E-P$ 且 $e_x\neq e_y$。此时，设从 e_x 开始到其最终代表点的路径上的节点构成集合 $Q=\{e_x,e_y,\cdots,e_q\}$。若集合 Q 不存在圈结构，则过程 RemoveCircle 按第一种情况处理，将所有节点的代表点修改为 e_q。若集合 Q 存在圈结构，根据其真实代表点 e_t 是否在集合 P 中又可以分为 2 种情况。

a) $e_t\notin P$。此时，集合 Q 的处理与集合 P 的处理无关，过程 RemoveCircle 按第二种情况处理集合 Q，将所有节点的代表点修改为 e_t，2 个集合都不再存在圈结构。

b) $e_t\in P$。此时，过程 RemoveCircle 仍按第二种情况处理集合 Q，只是真实代表点变为 e_t 在集合 P 中的真实代表点 e_s，即将集合 Q 中所有节点的代表点修改为 e_s，集合 Q 的圈结构消除，且不会影响集合 P 中的节点。

由上述分析可知，在消除新的圈结构时不会造成已经处理过的节点重新产生圈结构。这样，当依次处理完剩余的所有代表点后，代表点集合 E 不再存在圈结构。

证毕。

4.5　算法复杂度分析

性质 1　LAP 算法具有近似线性的时间复杂度和线性的空间复杂度。

说明　首先分析 LAP 算法的时间复杂度。设 $n=|V|$，$m=|E|$。步骤 1 中的相似度向量的计算的时间复杂度为 $O(d_{\max}n)$。算法最外层的 WHILE 循环的次数取决于代表点需要多长时间达到稳定。在极端情况下，每个节点都选择自身作为其代表点，从而得到 n 个单节点社区。通过设置参数 p 为远小于 0 的值可以避免这种情况的发生。此时，每次生成的社区至少包含 2 个节点，因而每次 WHILE 循环生成的代表点数均减少一半。这样，最外层的 WHILE 循环的次数为 $O(\log n)$。步骤 3 至步骤 14 的第一层 FOR 循环的次数为 $O(maxIter)$。步骤 4 至步骤 9 的第二层 FOR 循环的次数为 $O(n)$。步骤 5 至步骤 8 的最内层 FOR 循环的次数与节点的最大度成正比，设节点最大度为 $d_{\max}$，则该循环的次数为 $O(d_{\max})$。步骤 10 和步骤 11 的时间复杂度分别为 $O(d_{\max})$和 $O(convIter)$。由于采用了散列表，过程 RemoveCircle 的时间复杂度为 $O(nl_p)$，l_p 为节点到其真实代表点的最大路径长度。步骤 20 和步骤 22 的时间开销分别为 $O(m)$和 $O(n)$。综上可得，LAP 算法总的时间复

杂度 $O(d_{\max}n+\log n maxIter(n+d_{\max}+convIter)+nl_p+m)$。社交网络具有稀疏图的特征，因此 $d_{\max}<<n$，$convIter \leqslant maxIter$，$n \leqslant m$，$l_p<<n$，从而可以将 LAP 算法的时间复杂度简化为 $O(maxIter n\log n+m)$。由此可知，LAP 算法具有近似线性的时间复杂度。

接下来分析算法的空间复杂度。由于采用近邻传播，步骤 1 中只需要存储每个节点的相似度向量，其空间开销为 $O(d_{\max}n)$。同样，在向近邻传播消息过程中只需要为每个节点保存其支持度向量和适选度向量，其空间代价为 $O(d_{\max}n)$。步骤 11 要求保存最近 *convIter* 次代表点集，需要的空间为 $O(convIter n)$。指示向量需要的空间为 $O(n)$。过程 RemoveCircle 中存储散列表和路径需要的空间为 $O(n)$。因此，LAP 算法总的空间复杂度为 $O(d_{\max}n+convIter n)$。由于 $d_{\max}$，$convIter<<n$，LAP 算法的空间复杂度可以简化为 $O(n)$。由此可知，LAP 算法具有线性空间复杂度。

5 实验及分析

为了验证本文提出的新算法的性能，分别选择人工生成的数据集和真实数据集，与 AP 算法和 Yoshida 提出的算法（以下简称为 NEM 算法）进行实验对比。人工数据集利用 Lancichinetti 等提出的 LFR 基准程序生成[28]。LFR 能够根据输入的参数生成不同数据量、不同度分布及不同簇数的仿真数据集。真实数据集采用斯坦福大学的网络数据集 SNAP 中的 ego-Facebook 数据集[24]。ego-Facebook 数据集提供了社交网络 Facebook 中以用户为中心的好友圈子及好友的特征信息。实验数据集的详细描述如表 1 所示。

表 1 实验数据集

数据集	参数
人工数据集	N = 100~1 000 μ = 0.1 k = 15~50 c = 10~50 N_r=100
真实数据集	
348	N = 227 c = 14 N_r=161
3 437	N = 547 c = 32 N_r=262
107	N = 1 045 c = 9 N_r=576

表 1 中，参数 N、μ、k、c 和 N_r 分别表示数据集大小、节点平均外部度与内部度之比、节点度、真实社区数和特征数。真实数据集中的 348、3 437 和 107 代表相应编号节点的近邻组成的社交圈子数据集。在人工数据集方面，由于 LFR 不能生成节点的特征信息，采用以下步骤生成每个节点的仿真特征。

步骤 1 将 N_r 个特征组成的特征集 R 划分成 N_r / c 个互不相交的特征子集，每个特征子集对应一个社区。

步骤 2 设与社区 C_i 对应的特征子集为 R_i，社区 C_i 中各节点的特征以概率 0.9 从 R_i 中选择，以概率 0.1 从 $R-R_i$ 中选择。

通过这种方式，既保证各社区节点的特征具有较高的一致性，又允许不同社区节点的特征存在一定程度的重叠。所有算法均采用 Java 语言实现，并在硬件配置为 Intel i5-2520M 2.50 GHz CPU，8 GB RAM，软件配置为 Microsoft Windows7, JDK 7.0 的平台上进行测试。

实验比较了 LAP 算法与 AP 算法和 NEM 算法在不同数据量及不同边保留比例时的模块度、NMI (normalized mutual information)[28]、运行时间和社区数等指标的数值。这里，通过保留原始数据中不同比例的边来模拟真实采样中难以获得全部关联边信息的情况。为了克服随机性对算法测试的影响，所有实验结果均取 20 次运行结果的平均值。各算法的参数设置如表 2 所示。

表 2 算法参数的设置

算法	参数
AP	*maxIter* = 500, *convIter* = 50, p = best p
LAP	*maxIter* = 500, *convIter* = 50, p = −1.0, α = 0.5
NEM	α = 0.5, l = 10, k = true k

AP 算法对参数 p 非常敏感，需要根据不同的数据集调整参数 p 的值以达到最优聚类效果。实验中根据不同数据集取使聚类质量达到最高的 p 值。而 LAP 算法对 p 值变化不敏感，在一个较大范围内改变 p 值得到的社区数基本保持不变，因此实验时采用固定值 p=−1.0。NEM 算法需要输入社区数 k，实验中将 k 设为数据集的真实社区数，以使算法达到最佳性能。其他参数均取原作者论文中的推荐值。

5.1 人工数据集上的实验结果

5.1.1 边完整时的实验结果

本小节描述数据集中的边是完整时的实验结果。图 4 显示了算法的模块度随数据量的变化。

从图 4 可以看出，AP 算法和 LAP 算法得到的社区划分的模块度随着数据量的增加而逐渐提高。

由于 AP 算法和 LAP 算法均通过节点之间的关联边来传播近邻消息，而当数据量增大时网络的边数及每个节点的平均度也相应增加，使消息能够得到更充分地传播，从而使社区计算得到的模块度数值更大。LAP 算法得到的模块度在数据量较小时略低于 AP 算法，表明局部近邻传播策略在数据量较小时对算法性能有一定影响。但随着网络节点及边的增加，这种影响迅速减小，而 LAP 采用的更有效的代表点选择策略的优势则逐渐显露。因此，当数据量超过 400 时，LAP 算法得到的模块度已经高于 AP 算法。图 4 也反映了 NEM 算法得到的社区划分的模块度低于 AP 算法和 LAP 算法，且不同数据量的模块度的波动较明显，这可能与其采用具有随机性的 k 均值算法对映射后的子空间向量进行聚类有关。由于 LFR 基准程序生成的仿真数据分组含真实社区信息，也对不同算法的 NMI 指标进行了实验，结果如图 5 所示。

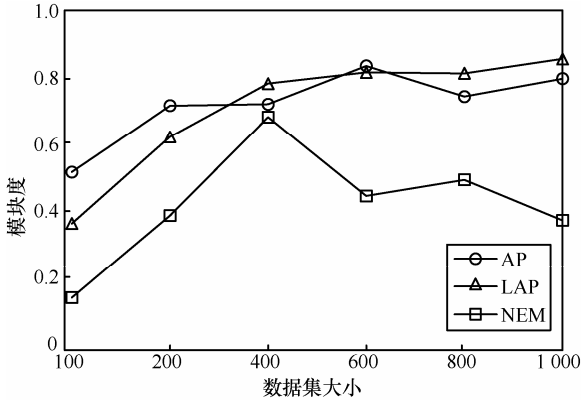


图 4　模块度随数据集大小的变化（人工数据集）

图 5 的实验结果与图 4 基本相似，但也存在一些不同之处。从图 5 可以发现，AP 算法得到的社区划分在所有数据量下均非常接近于真实的社区结构。LAP 算法在数据量大于 100 时也能够得到与真实社区的相近程度大于 80%的社区划分。特别地，当数据量达到 1 000 时，LAP 算法得到的社区划分的质量与 AP 算法已经基本相当。这同样可以由前述 LAP 算法采用的局部近邻传播与代表点约束放松策略的影响来解释。NEM 算法的 NMI 值较低，反映其得到的社区划分与真实社区存在较显著的差别，而曲线较大幅度的波动亦反映了 k 均值算法对其性能的影响。

不同算法的运行时间随数据集大小的变化反映在图 6 中。

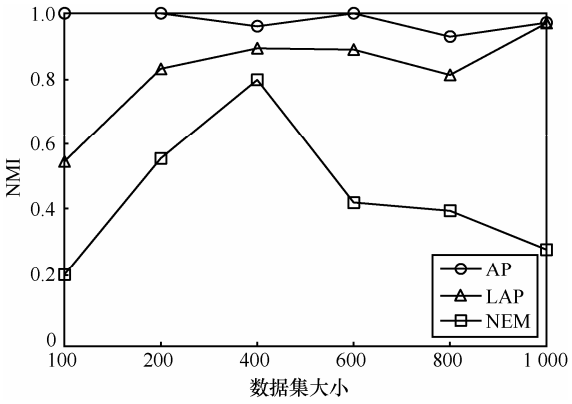


图 5　NMI 随数据集大小的变化（人工数据集）

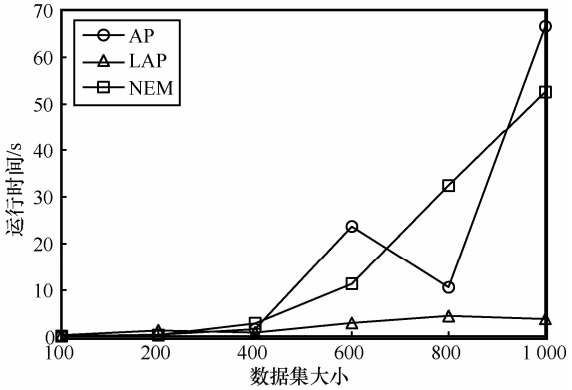


图 6　运行时间随数据集大小的变化（人工数据集）

图 6 显示，与 AP 算法和 NEM 算法相比，LAP 算法的运行时间随数据量的增加的幅度非常平缓。这主要有 2 个方面的原因：一是由第 4 节的分析可知 LAP 算法具有近似线性的时间复杂度，而 AP 算法和 NEM 算法的时间复杂度分别达到 $O(n^2)$和 $O(n^3)$，因此 LAP 算法随数据量增加的时间代价显著小于 2 个对比算法；二是 LAP 算法采用的局部近邻传播及代表点约束放松策略使算法在数据量较大时仍保持较快的收敛速率，而 AP 算法在数据量增大时易出现不收敛现象，使算法的迭代次数大大增加，从而增加了算法的运行时间。

由于 AP 算法和 LAP 算法的社区均由聚类过程自动涌现，还将它们得到的社区数与真实社区数进行了比较，结果如图 7 所示。其中，由于 NEM 算法的社区数参数 k 设为真实社区数，将其作为比较的基准。

由图 7 可知，尽管 AP 算法和 LAP 算法的社区是在聚类过程中自动涌现的，最终得到的社区数与真实社区数仍比较接近。这表明在网络中的边结构完整时，AP 算法和 LAP 算法的具有较强的社区识别能力。由于 LAP 算法在不同数据集上均采用相同

的 p 值，而 AP 算法需要调整 p 值以适应不同数据集，LAP 算法的适应性显然强于 AP 算法。

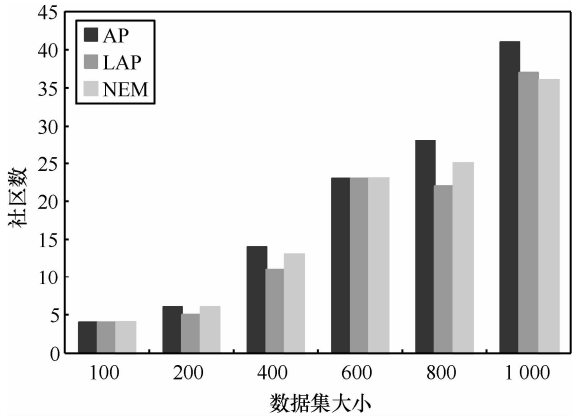


图 7 社区数随数据集大小的变化（人工数据集）

5.1.2 边不完整时的实验结果

现实中采样得到的数据通常无法涵盖网络中所有边。通过按不同比例保留测试数据集中的边，比较了各个算法在不同边缺失情况下的性能，数据集取大小为 400 的人工数据集，其他参数设置同表 2。模块度的实验结果如图 8 所示。

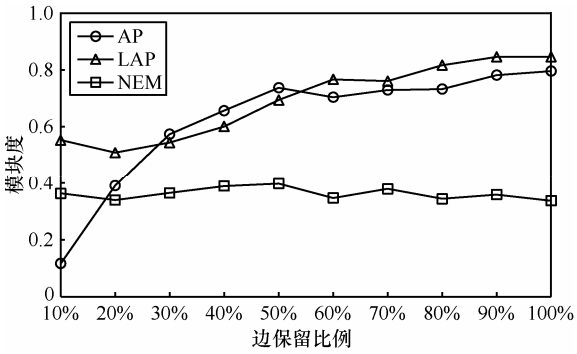


图 8 不同边保留比例时的模块度（人工数据集）

图 8 反映边的减小对 LAP 算法和 NEM 算法的影响较小。这主要是由于 2 个算法在计算节点相似度时均同时考虑网络拓扑和节点的特征相似度。当网络中的边减小时，节点特征相似度对节点之间相似性的判断起到主要作用。而 LAP 算法采用的新策略使其得到的社区的模块度显著高于 NEM 算法。AP 算法需要根据网络中的边信息计算节点相似度矩阵。当网络中的边大量减小时，其相似度矩阵将存在大量 0 值，从而对算法性能造成较大影响。边的删除导致网络拓扑结构发生变化，因此 LFR 基准程序提供的真实社区信息不再适用，故未对 NMI 指标进行测试。但从图 5 和图 4 的相似性可以推测在不同边保留比例下的 NMI 指标值应与图 8 有相似结果。不同边保留比例下运行时间的实验结果如图 9 所示。

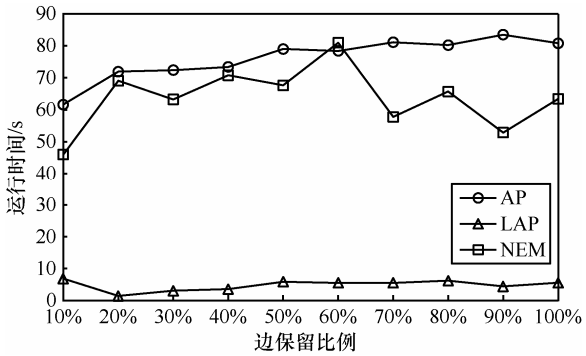


图 9 不同边保留比例的运行时间

图 9 显示算法的运行时间受边数变化的影响较小，这主要是因为算法的时间复杂度主要与数据集规模相关。在数据集大小不变时，各算法的运行时间基本保持稳定。只有 NEM 算法受其采用的 k 均值算法影响，运行时间的波动略大。与图 6 中的实验结果相一致，LAP 算法由于具有近似线性的时间复杂度，运行速度最快。AP 算法和 NEM 算法由于时间复杂度较高，运行速度显著慢于 LAP 算法。

5.2 真实数据集上的实验结果

5.2.1 边完整时的实验结果

图 10 显示了不同算法在 3 个真实数据集上的得到的社区划分的模块度。

真实数据集中通常存在噪声数据，这增加了社区识别的难度，因此图 10 中各算法的模块度总体上要低于图 4 中的结果。除了在 348 数据集上的模块度略低于 NEM 算法外，LAP 算法在所有其他数据集上的性能均优于对比算法，再一次表明 LAP 算法采用的局部近邻传播及代表点约束放松策略在数据集较大时具有较好的效果。由表 1 可知，348 数据集的网络规模较小，每个真实社区的规模也不大，且经过计算该数据集的节点平均度为 28，因此该数据集的社区结构不显著，导致各算法在该数据集上的识别效果均不理想。此外，虽然 107 数据集大于 3 437 数据集，但算法在 3 437 数据集上得到的模块度却高于 107 数据集。由于 107 数据集可能具有更复杂的内部结构（其节点具有更多的特征），这表明了在真实的应用环境中，不仅网络规模，网络内部结构的复杂程度也对算法的性能产生影响。

ego-Facebook 数据集中的真实社区信息不完整，故无法测试 NMI 指标。图 11 显示了不同算法在 3 个真实数据集上的运行时间。

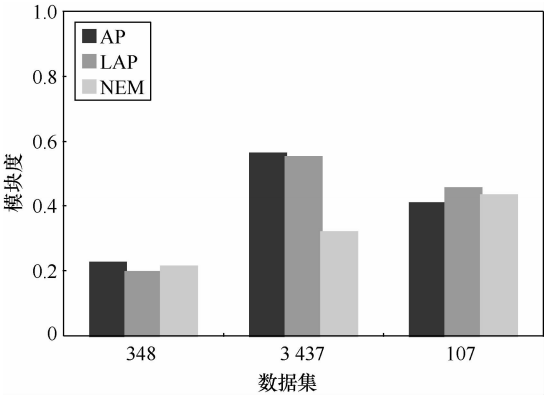


图 10　不同真实数据集的模块度

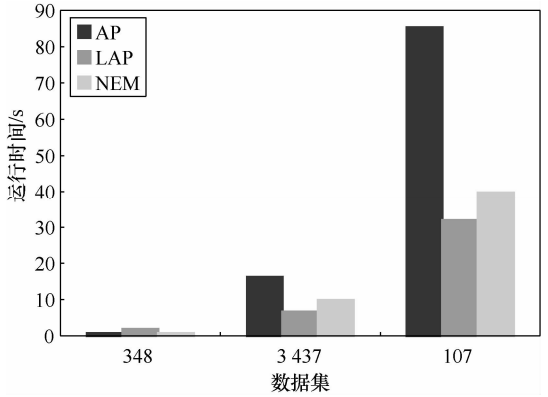


图 11　不同真实数据集的运行时间

与图 6 的结果相似，图 11 同样表明 AP 算法和 NEM 算法的时间复杂度随数据量的增加而快速增加。与之相反，LAP 算法的运行时间随数据量的增加呈近似线性增长，且总是低于 2 个对比算法。其中，由于 107 数据集的复杂度较高，AP 算法不能在 *maxIter* 次迭代内收敛，大大增加了其运行时间。图 12 显示了各算法在不同数据集上得到的社区数。

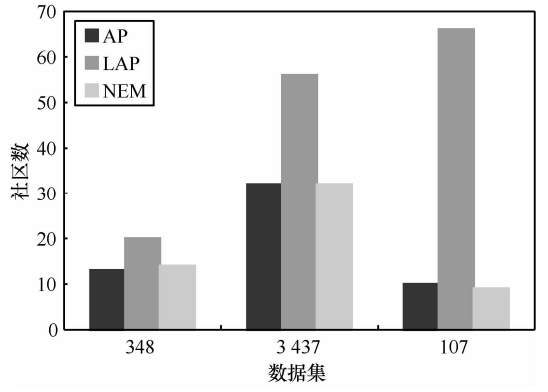


图 12　不同数据集的社区数

图 12 表明， LAP 算法倾向于生成较多小规模的社区，这主要是由于当 p 值固定时，LAP 算法采用的局部近邻传播及代表点约束放松策略使算法的收敛速度加快。而 AP 算法和 NEM 算法由于可以分别通过调整参数 p 和参数 k 的值来匹配真实社区数，不存在这个问题。但是，由图 13 的结果可知，即使 AP 算法和 NEM 算法能够生成正确的社区数，其生成的社区的模块度仍然不如 LAP 算法。

5.2.2　边不完整时的实验结果

当真实网络中的边存在缺失时，不同算法得到的模块度如图 13 所示。采用 107 数据集作为测试数据集，其他参数设置同表 2。

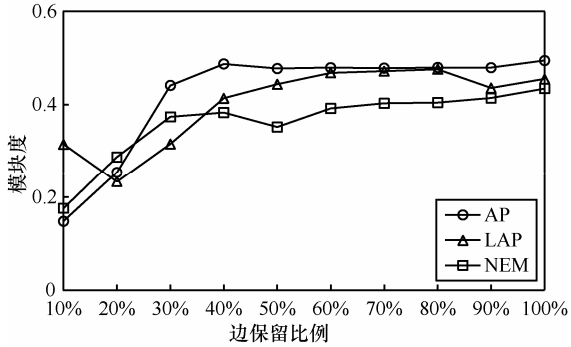


图 13　不同边保留比例时的模块度（真实数据集）

图 13 中的实验结果与图 8 存在相似性，AP 算法和 LAP 算法的模块度仍然高于 NEM 算法。但可以发现，在边保留比例较低时，LAP 算法和 NEM 算法的性能受到一定影响。当保留的边数减少到一定程度时，网络将分隔为多个独立的连通分支，大量节点间的拓扑相似度接近 0，不能有效描述节点间的真实相似度。当 α=0.5 时对算法性能产生一定影响。结合后面对参数 α 的实验结果，此时可以通过减小 α 值使节点特征相似度发挥主导作用，起到一定弥补作用。不同边保留比例时算法的运行时间如图 14 所示。

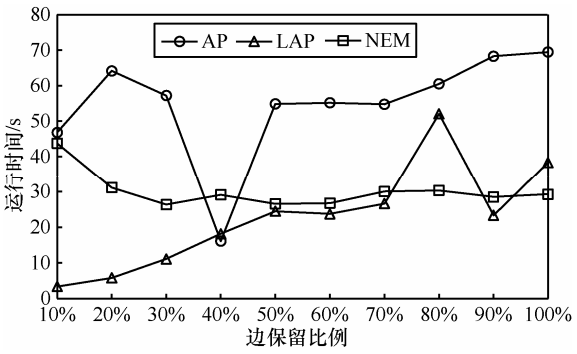


图 14　不同边保留比例的运行时间

图 14 显示的实验结果与图 9 基本一致，LAP 算法具有最快的运行速度，其次是 NEM 算法，AP 算法的运行速度最慢。但 LAP 算法在边保留比例为 80%处存在一个尖峰，这主要是由于此时 LAP 使用了较多的迭代次数才达到收敛，从而增加了运行时间。而在边保留比例为 40%处，AP 算法也存在一个低谷，这主要是由于此时 AP 算法在 *stableIter* 次迭代时即达到收敛，从而减小了运行时间，再次反映了真实数据集内部结构的复杂性对算法性能存在影响。

5.3 参数 α 的实验结果

参数 α 决定了在计算节点相似度时拓扑相似度与特征相似度的相对比例。为了研究 α 值的变化对 LAP 算法性能的影响，在大小为 400 的人工数据集上测试了不同边保留比例下不同 α 值的模块度，实验结果如图 15 所示。

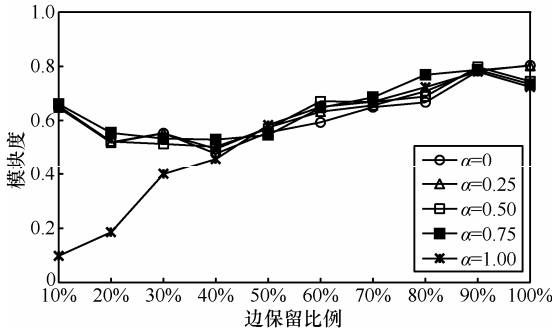


图 15 不同边保留比例及 α 值的模块度

从图 15 可以看出，当 α=1.0 时，边保留比例的降低将导致模块度的显著下降，表明当完全依赖拓扑相似度进行社区识别时，算法性能受网络边缺失的影响较明显。当 α 取小于 1.0 的其他值时，节点的特征相似度参与到节点组合相似度的计算中，此时模块度随边缺失程度变化的幅度较小，表明节点特征相似度对提高社区识别精度具有显著作用。与图 13 中的实验结果类似，图 15 也反映出算法的模块度随着边保留比例的增大而逐渐提高。

6 结束语

针对社交网络规模庞大、动态变化对社交网络上的社区识别带来的挑战，本文提出一种新的将局部传播近邻消息、放松代表点约束，以及在结合网络拓扑及用户特征度量节点相似性等策略相结合的社区识别算法。通过理论分析和实验检验，证明了提出的算法不仅具有较低的时间和空间复杂度，而且在采样得到的网络存在边缺失时仍具有较好的识别精度，具有一定的实用意义。

在接下来的工作中，将进一步考虑更多的节点近邻选择方法，比较不同方法对社区识别精度的影响，并考虑引入 MapReduce、MPI 等并行计算框架，实现算法的并行化，使算法具有更高的实用价值。

参考文献：

[1] 杨博, 刘大有, LIU J M 等. 复杂网络聚类方法[J]. 软件学报, 2009, 20(1): 54-66.
YANG B, LIU D Y, LIU J M, *et al*. Complex network clustering algorithms[J]. Journal of Software, 2009, 20(1): 54-66.

[2] FORTUNATO S. Community detection in graphs[J]. Physics Reports, 2010, 486(3-5): 75-174.

[3] SHIGA M, TAKIGAWA I, MAMITSUKA H. A spectral approach to clustering numerical vectors as nodes in a network[J]. Pattern Recognition, 2011, 44(2): 236-251.

[4] NEWMAN M E J. Detecting community structure in networks[J]. The European Physical Journal B - Condensed Matter, 2004, 38(2): 321-330.

[5] GUIMERA R, SALES-PARDO M, AMARAL L A N. Modularity from fluctuations in random graphs and complex networks[J]. Physical Review E, 2004, 70(2): 025101.

[6] GUIMERA R. Functional cartography of complex metabolic networks[J]. Nature, 2005, 433(7028): 895-900.

[7] DUCH J, ARENAS A. Community detection in complex networks using extremal optimization[J]. Physical Review E, 2005,72(2): 027104.

[8] SON S W, JEONG H, NOH J D. Random field ising model and community structure in complex networks[J]. The European Physical Journal B, 2006, 50(3): 431-437.

[9] 淦文燕, 赫南, 李德毅等. 一种基于拓扑势的网络社区发现方法[J]. 软件学报, 2009, 20(8): 2241-2254.
GAN W Y, HAO N, LI Y D, *et al*. Community discovery method in networks based on topological potential[J]. Journal of Software, 2009, 20(8): 2241-2254.

[10] 何东晓, 周栩, 王佐等. 复杂网络社区挖掘—基于聚类融合的遗传算法[J]. 自动化学报, 2010, 36(8): 1160-1170.
HE D X, ZHOU X, WANG Z, *et al*. Community mining in complex networks — clustering combination based genetic algorithm[J]. Acta Automatica Sinica, 2010, 36(8): 1160-1170.

[11] YE Z Q, ZHANG K, HU S N, *et al*. A new definition of modularity for community detection in complex networks[J]. Chinese Physics Letters, 2012, 29(9): 098901.

[12] 林旺群, 卢风顺, 丁兆云等. 基于带权图的层次化社区并行计算方法[J]. 软件学报, 2012, 23(6): 1517-1530.
LIN W Q, LU F S, DING Z Y, *et al*. Parallel computing hierachical community approach based on weighted-graph[J]. Journal of Software, 2012, 23(6): 1517-1530.

[13] 杨博, 刘杰, 刘大有. 基于随机网络集成模型的广义网络社区挖掘算法[J]. 自动化学报, 2012, 38(5): 812-822.
YANG B, LIU J, LIU D Y. A random network ensemble model based generalized network community mining algorithm[J]. Acta Automatica

Sinica, 2012, 38(5): 812-822.

[14] 韩毅, 方滨兴, 贾焰等. 基于密度估计的社会网络特征簇挖掘方法[J]. 通信学报, 2012, 33(5): 38-48.
HAN Y, FANG B X, JIA Y, *et al*. Mining characteristic clusters: a density estimation approach[J]. Journal on Communications, 2012, 33(5): 38-48.

[15] GIRVAN M., NEWMAN M E J. Community structure in social and biological networks[J]. Proceedings of the National Academy of Sciences of the United States of America, 2002, 99(12): 7821-7826.

[16] WU F, HUBERMAN B A. Finding communities in linear time: a physics approach[J]. The European Physical Journal B-Condensed Matter, 2004,38(2): 331-338.

[17] PALLA G, DERENYI I, FARKAS I, *et al*. Uncovering the overlapping community structure of complex networks in nature and society[J]. Nature, 2005,435(7043): 814-818.

[18] WU Z H, LIN Y F, GREGORY S, *et al*. Balanced multi-label propagation for overlapping community detection in social networks[J]. Journal of Computer Science and Technology, 2012, 27(3): 468-479.

[19] 金弟, 杨博, 刘杰等. 复杂网络簇结构探测—基于随机游走的蚁群算法[J]. 软件学报, 2012, 23(3): 451-464.
JIN D, YANG B, LIU J, *et al*. Ant colony optimization based on random walk for community detection in complex networks[J]. Journal of Software, 2012, 23(3): 451-464.

[20] WANG X, MOHANTY N, MCCALLUM A. Group and topic discovery from relations and their attributes[J]. Advances in Neural Information Processing Systems, 2006, 18:1449.

[21] MOSER F, GE R, ESTER M. Joint cluster analysis of attribute and relationship data without a-priori specification of the number of clusters[A]. Proceedings of the 13th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining(KDD'07)[C]. 2007. 510-519.

[22] YAN L, ALEXANDRU N M, WOJCIECH G. Topic-link LDA: joint models of topic and author community[A]. Proceedings of the 26th Annual International Conference on Machine Learning[C]. 2009. 665-672.

[23] YOSHIDA T. Toward finding hidden communities based on user profiles[A]. Proceedings of the 2010 IEEE International Conference on Data Mining Workshops (ICDE'10)[C]. 2010. 380-387.

[24] MCAULEY J., LESKOVEC J. Learning to discover social circles in ego networks[A]. Proceedings of the 26th Annual Conference on Neural Information Processing Systems 2012[C]. 2012. 548-556.

[25] BRENDAN J F, DELBERT D. Clustering by passing messages between data points[J]. Science, 2007, 315(5814): 972-976.

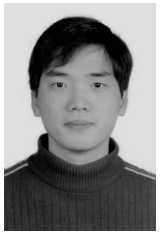
[26] JUDEA P. Probabilistic Reasoning in Intelligent Systems: Networks of Plausible Inference[M]. Morgan Kaufmann, 1988.

[27] SUMEDHA M L, WEIGT M. Unsupervised and semi-supervised clustering by message passing: soft-constraint affinity propagation[J]. The European Physical Journal B, 2008, 66(1):125-135.

[28] LANCICHINETTI A, FORTUNATO S. Benchmarks for testing community detection algorithms on directed and weighted graphs with overlapping communities[J]. Physical Review E, 2009, 80(1):1-8.

[29] LANCICHINETTI A., FORTUNATO S., KERTÉSZ J. Detecting the overlapping and hierarchical community structure in complex networks[J]. New Journal of Physics, 2009, 11(3): 033015.

作者简介：



郭昆（1979-），男，福建福州人，博士，福州大学讲师、硕士生导师，主要研究方向为社交网络数据挖掘、灰色系统理论、决策支持系统等。

郭文忠（1979-），男，福建泉港人，博士，福州大学教授、博士生导师，主要研究方向为智能信息处理、网络计算。

邱启荣（1981-），男，福建仙游人，福州大学博士生，主要研究方向为社会网络。

张岐山（1962-），男，黑龙江绥化人，博士，福州大学教授、博士生导师，主要研究方向为灰色系统、商务智能与系统工程等。