基于时间加权的重叠社区检测算法研究

李 慧[1,4] 马小平[2] 张 舒[3] 施 珺[1] 李存华[1] 仲兆满[1,4]

摘 要 随着网络结构的不断扩大和日益复杂, 重叠社区发现技术对挖掘复杂网络深层潜在结构具有重要意义. 本文提出一种基于时间加权的重叠社区检测算法. 该方法考虑了用户兴趣的时间因素, 构建带有时间加权链接的用户–用户图. 接着, 基于网络节点的影响力计算用户全局相似度, 在此基础上通过计算节点的中心度作为度量节点对社区结构影响力的重要性指标, 从而提出一种社区中心点的选取方法. 最后, 通过效用函数的迭代计算实现重叠社区检测. 利用人工网络和真实网络对提出的算法进行验证, 实验结果表明: 相对于传统的社区发现方法, 该算法在社区发现质量和计算效率方面都优于许多已有重叠社区发现算法.

关键词 社会网络, 加权, 重叠, 社区, 检测

引用格式 李慧, 马小平, 张舒, 施珺, 李存华, 仲兆满. 基于时间加权的重叠社区检测算法研究. 自动化学报, 2021, **47**(4): 933−942

DOI 10.16383/j.aas.c180559

Research of Overlap Community Detection Algorithm Based on Time-Weighted

LI Hui[1,4] MA Xiao-Ping[2] ZHANG Shu[3] SHI Jun[1] LI Cun-Hua[1] ZHONG Zhao-Man[1,4]

Abstract With the continuous expansion and complexity of network structure, overlapping community discovery technology is of great significance to excavate the deep potential structure of complex network. This article presents an overlapping community detection algorithm based on time time-weighted. Considering the time factor of user interest, this method constructs a user-user graph with time-weighted links. Then, the global similarity of users is calculated based on the influence of network nodes. On this basis, the centrality of nodes is calculated as an important index to measure the impact of nodes on community structure, and a method to select community centers is proposed. Finally, overlapping community detection is realized by iteration of utility function. The proposed algorithm is validated by artificial network and real network. The experimental results show that compared with traditional community discovery methods, the proposed algorithm outperforms many existing overlapping community discovery algorithms in terms of community discovery quality and computational efficiency.

Key words Social network, time-weighted, overlap, community, detection

Citation Li Hui, Ma Xiao-Ping, Zhang Shu, Shi Jun, Li Cun-Hua, Zhong Zhao-Man. Research of overlap community detection algorithm based on time-weighted. *Acta Automatica Sinica*, 2021, **47**(4): 933−942

社区结构是复杂网络的重要特性, 在网络中发现社区就是把相似节点划分为一个集合, 使得集合内节点之间的相互作用比它们与集合外节点的相互作用更强, 即同一社区内部节点间的链接较为稠密, 不同社区之间的链接较为稀疏[1]. 但是社会化网络中用户的多重社会属性导致用户可以同时从属于多个社区, 因此基于可重叠聚类的社区发现算法效果更佳. 发现高质量的社区有助于理解真实的复杂网络, 尤其是动态地分析社区重叠结构, 对社区管理和演化具有重要意义[2−4].

在传统的社区发现方法中, 网络可以作为静态

收稿日期 2018-10-30 录用日期 2019-05-19
Manuscript received October 30, 2018; accepted May 19, 2019
国家自然科学基金 (61873105), 江苏省 “333 工程” 培养对象, 连云港市第六期 “521 工程” 培养对象, 江苏省 “333 工程” 项目 (BRA2020261), 教育部协同育人项目 (201902159041), 江苏省教改项目 (JGX2019011ZZ), 江苏省高等学校自然科学研究项目 (19KJB520004), 连云港高新区科技计划项目 (ZD201912) 资助
Supported by National Natural Science Foundation of China (61873105), “333 Project” cultivation object of Jiangsu province, “521 Project” Cultivation Object of Lianyungang, “333 Project” of Jiangsu Province (BRA2020261), Ministry of Education Collaborative Education Project (201902159041), Education Reform Project of Jiangsu Province (JGX2019011ZZ), Natural Science Research Project of Higher Education of Jiangsu Province (19KJB520004), and Lianyungang High-tech Zone Science and Technology Project (ZD201912)
本文责任编委 段书凯
Recommended by Associate Editor DUAN Shu-Kai
1. 江苏海洋大学计算机工程学院 连云港 222005 2. 中国矿业大学信息与控制工程学院 徐州 221008 3. 江苏海洋大学商学院 连云港 222005 4. 江苏省海洋资源开发研究院 连云港 222005
1. School of Computer Engineering, Jiangsu Ocean University, Lianyungang 222005 2. School of Information and Control Engineering, China University of Mining and Technology, Xuzhou 221008 3. School of Business, Jiangsu Ocean University, Lianyungang 222005 4. Jiangsu Institute of Marine Resources Development, Lianyungang 222005

拓扑图处理而不用考虑节点间的信息交互因素, 在微博等社交网络中已经不再适用. 在微博及其应用所构成的社交网络中频繁地使用不同节点间的信息交互; 拓扑结构仅代表用户之间交互的可能性, 而实际交互的程度则由节点之间的信息流动情况决定. 这种社区划分方法由于仅仅依赖拓扑结构, 却忽略了社交网络中的信息流动, 因此表现出明显的局限性, 这已经与现代社交网络的特征相背离, 除此之外, 社区划分结果在这种体系下也无法得到较高的准确性.

本文的重叠社区检测算法是针对传统的社区发现方法在解决社交网络中社区划分时所面临的问题所提出的, 称为基于时间加权关联规则的时域重叠社区检测算法 (Time-weighted overlapping community detection, TWOCD). TWOCD 算法的主要创新点在于重叠社区检测时充分考虑了用户兴趣的时间因素, 根据带有时间加权链接的用户 – 用户图实现重叠社区检测.

本文第 1 节介绍了重叠社区检测的相关工作, 并描述了一些主流的重叠社区发现算法; 第 2 节具体地阐述了重叠社区的检测算法及社区合并方案; 第 3 节是算法性能验证实验; 第 4 节是我们的工作的总结以及对未来研究工作的展望.

1 相关工作

目前已出现 5 类重叠社区发现算法, 即派系过滤算法、局部扩展社区发现算法、模糊重叠社区发现算法、边社区发现算法、标签传播算法.

1.1 派系过滤算法

2005 年, Gergely 等提出了派系过滤 (Clique percolation method, CPM) 算法[5]. 其核心思想是发现基于 k 极大团的重叠社区. 由 k 个节点构成的完全连通子图称为 k 极大团. Gergely 等引入一种新的概念, 即将具有 $k-1$ 个相同节点的两个 k 极大团称为邻接的 k 极大团. 派系过滤算法 (Cluster porcdation method, CPM) 旨在寻找邻接的 k 极大团. 由于极大团的内部节点之间的全连通性可以形成一种内部紧密而外部稀疏的社区结构, 这是一种理想的社区结构. 邻接的 k 极大团就是派系过滤算法 (CPM) 寻找的重叠社区结构. 但是 CPM 算法具有只能够发现基于 k 极大团的重叠社区结构的缺陷. Farkas 等对 CPM 算法进行改进, 将其扩展应用到有权图上, 提出子图密度的概念实现对 k 极大团的搜索[6]. 2015 年, Zhang 等提出一种新的重叠社区发现方法, 称为 MOHCC 算法[7]. 该算法在寻找图中极大团的基础上结合 Wang 提出的 Coupling Strength[8] 作为目标函数进行极大团的合并, 从而得到最佳的层次划分.

1.2 局部扩展社区发现算法

基于局部扩展的重叠社区发现算法, 通常从不同种子节点开始, 根据设定的某优化函数, 探索种子所在的局部社区结构, 各个局部社区结构融合形成网络整体的重叠社区结构[9]. 代表算法有 LFM 算法[10] 和 GCE 算法[11]. LFM 算法的基本思想是每次在网络中随机选取一个尚无社区标签的节点作为种子, 然后采用一种贪心的策略将种子扩展为一个局部自然社区, 直到网络中所有节点都有社区标签为止. 在局部扩展的过程中, LFM 算法通过不断对当前子图增加或者删除节点使得适应度函数值达到局部最大值. GCE 算法在整个算法执行的初始阶段, 在网络中找出所有节点规模不小于 k 的最大团 (全连通子图) 作为种子; 然后同样采用贪心的策略对种子进行扩展得到局部自然社区, 其设定的适应度函数与 LFM 算法相同. 该算法在扩展的每一次迭代中仅添加使得适应度函数最大的节点, 得到新的社区之后重复执行直到适应度函数不再增大, 然后将此时的社区同之前已检测到的所有社区计算二者的距离, 根据设定的阈值决定是否保留该社区.

2011 年, Lancichinetti 等又提出了 OSLOM 算法[12], 该算法提出了一种带有随机扰动的用于表达社区的统计学重要性局部优化适应度函数. 根据该适应度函数寻找重要的社区, 直至收敛. 2017 年, Yang 等[13] 提出了一种种子节点选择策略, 并基于节点影响力和模块度定义目标函数, 从而实现社区的初始化和社区优化. Su 等[14] 根据节点的中心性从网络中选取种子节点, 并计算其与邻居节点的局部簇系数来决定是否和邻居节点进行合并, 从而实现社区结构的发现.

1.3 模糊重叠社区发现算法

模糊重叠社区发现算法通过确定节点与社区之间的隶属度来确定节点与社区的从属关系, 为重叠社区发现中的另一类重要算法. 2011 年 Gregory 针对社交网络的社区检测首次提出了 “模糊重叠划分 (Fuzzy overlapping partition)” 的概念[15]. 模糊重叠社区检测与传统离散重叠社区检测的区别在于: 允许重叠节点对所属社区具有不完全且不一致的隶属关系, 利用 $[0,1]$ 连续区间内分布的模糊隶属度量化重叠节点对不同社区的相对隶属程度.

2015 年, Eustace 等的邻居比例矩阵模型结合非负矩阵分解算法, 使用 Perron clusters 进行网络中的社区数目的求解, 并将其应用到重叠社区发现中[16], 实现了将网络中低于平均邻居节点数目的节点之间关系的过滤功能. 文献 [17] 提出了一种在社交网络下基于模糊自适应推理理论的重叠社区发现

算法, 该算法包含比较和预测两个阶段, 通过两个阶段的循环迭代较好地解决社区发现问题. 文献 [18] 提出了一种模糊模块度最大化方法, 利用模块度优化模型确定节点的最优隶属度. 此外, 还有一些研究以非负矩阵分解为工具, 提出一些节点隶属度的计算方法[19−20].

1.4 边社区发现算法

重叠社区发现的焦点问题可以归结到节点的社区结构研究上, 忽略了边对于重叠社区发现问题研究的重要性. 边聚类算法的核心思想是在将边转换为聚类算法能够处理的模型的基础上, 利用聚类算法对边进行聚类, 从而实现边社区的发现. 相继产生了一些边聚类算法中的代表性算法, 如 Ahn 等[21] 提出的经典的边聚类 (Link clustering, LC) 算法的核心思想是将 Jaccard 方法应用到边的相似度计算中, 从而得到边的相似度矩阵. Shi 等在经典的边聚类算法基础上又提出了将遗传算法应用到边聚类的方法, 称为 GaoCD 算法[22], 该算法将分割密度作为目标函数, 基于一种新的基因表达方法实现边社区到节点社区的转换. 2014 年, Lim 等提出的 LinkScan 算法[23] 用于边社区发现. Li 等[24] 提出了一种以线图模型为基础的加权模型, 对模块密度函数进行优化识别, 设计一种新的基因表示模型将链路社区映射为节点社区, 从而实现重叠社区的检测. 目前边社区发现算法已经成为一类重要的重叠社区发现算法.

1.5 标签传播算法

标签传播算法的核心思想为节点通过与邻域节点之间交互社区归属标签信息, 更新节点自身的社区归属标签, 使网络中所有节点对应的标签分布达到动态平衡, 具有相同标签的节点构成社区, 而具有多个社区标签的节点为重叠节点, 由此得到重叠社区结构. 这类方法的典型代表是基于多标签的 COPRA 算法[25] 和基于 Speaker-listener 模型的 SLPA 算法[26]. COPRA 是 Gregory 于 2010 年提出的首个基于标签传播的模糊重叠社区检测算法, 节点标签对中不仅含有社区名称, 而且包含节点对该社区的归属系数. SLPA 算法是由 Xie 等于 2011 年提出的, 该算法为每个节点提供存储信息 (标签) 的记忆空间, 将从记忆空间中获取标签的概率作为节点隶属度, 无需社区数目等先验信息. Gaiter 等[27] 于 2015 年提出了一种 SpeakEasy 聚类方法, 根据节点的局部连接性和网络全局信息将节点加入社区, 该方法在社区结构稳定性上给出了定量分析与评价.

上述介绍的 5 种方法是一些经典的重叠社区发现算法, 每种算法适用于不同的场合. 本文所提出的算法是对边社区发现算法的扩充, 通过加入用户相似度和社区中心点提升重叠社区发现算法的准确率.

2 重叠社区的生成

已知一组用户和一组对象, 这些用户和对象间的交互关系可表示为一个用户 – 对象关系图, 该图中的用户节点只与其感兴趣的对象相连. 然后, 可将用户 – 对象关系图转化为用户 – 用户关系图, 且用户 – 用户关系图中两个用户间的链接表示这两个用户共同喜欢某些对象, 且链接权重表示这些共同对象的数量. 考虑到用户兴趣会随着时间的变化而变化, 我们假设两个用户发生交互的时间越近, 则这两个用户具有共同兴趣的概率越大, 越会在用户 – 用户图中形成相应的时间加权链接.

2.1 利用时间加权链接构建用户 – 用户图

根据数据的时间标签, 将训练数据集分成不同时间段的数据子集. 假设第 i 个用户在第 t 段时间对第 j 个对象打分, 其中 $i = 1, \cdots, n.\ j = 1, \cdots, m.\ t \in \{1, \cdots, TL\}$, 用 n 表示用户数量, m 表示对象数量, TL 表示训练数据集中所有交互的总时间. 如果没有交互信息或者打分低于预期阈值, 则将 t 设置为 0.

于是, 利用如下类似于遗忘曲线的函数, 将交互情况表示为时间加权用户 – 对象矩阵 $G = [g_{ij}]_{n\times m}$:

$$g_{ij} = \begin{cases} \mathrm{e}^{-\frac{TL-t}{\theta_1}}, & t > 0 \\ 0, & t = 0 \end{cases} \tag{1}$$

其中, $\theta_1 > 0$ 表示预先指定的实数, 用于反映交互的时间效应. θ_1 数值越大, 时间对用户和对象间交互的影响越少. 例如, 当 $\theta_1 \to +\infty$ 时, $g_{ij} = 1$, 时间加权用户 – 对象图转化为没有考虑时间效应的传统用户评分矩阵. 然后, 对具有相同对象喜好的用户间添加链路, 将用户 – 对象图转化为时间加权用户 – 用户图. 从矩阵角度讲, 可将用户 – 用户图描述为用户 – 用户矩阵.

$$U = G \times G^{\mathrm{T}} = [u_{il}]_{n\times n} \tag{2}$$

因此, 用户间的链接反映了用户兴趣的相似度, 且用户兴趣的相似度主要取决于用户共同喜欢的对象数量以及喜欢这些对象的时间. 矩阵 U 中的元素 u_{il} 表示第 i 个和第 l 个用户间的兴趣相似度, 且 $u_{il} = \sum_{i=1}^{m} g_{ij} \times g_{lj}, i = 1, \cdots, n, l = 1, \cdots, n.$ 但是这个相似度只能反映节点的局部相似性, 要想真实反映网络中节点间的相似度必须从全局角度计算用户的全局相似度.

2.2 用户全局相似度的计算

网络中节点间的相似度计算大多基于节点的局部信息. 如果两个节点共享更多的邻居, 它们就会被认为更加相似. 但是, 该方法没有考虑到网络中节点的全局重要性. 在本节中, 我们融合了网络的全局结构来计算用户间全局相似度. 首先基于原始 PageRank 算法定义节点影响度, 以测量网络中节点的影响程度. 节点的影响程度越大, 节点在网络中的全局重要性就越大.

2.2.1 节点的影响度

我们使用 PageRank 算法[28] 来计算网络中的节点影响程度. PageRank 算法的主要思想是网页中节点的 PageRank 值等于指向它的所有节点 PageRank 值的总和. 同样, 网络中节点的影响度是指向它的所有节点的影响度总和. 节点 i 影响度 $Inf(i)$ 计算方法如下:

$$Inf(i) = c \sum_{l \in F(i)} \frac{Inf(l)}{N(l)} + \frac{1-c}{N} \tag{3}$$

其中, $Inf(i)$ 代表节点 l 的影响度, 即网络中节点 l 的度数. $F(i)$ 是节点 i 的一个邻居集合, $N(l)$ 是节点 l 的邻居数量, N 是图中节点的总数. 为了便于计算, 在方程中加入常数 c, $c \in (0,1)$ 为阻尼因子, 一般设为 0.85. 阻尼因子的取值是基于原 PageRank 算法的经验分析.

2.2.2 用户全局相似度

为了计算用户间的全局相似度, 我们将由式 (2) 计算出的局部相似度与节点的结构聚合度相结合, 计算用户全局相似度. 在网络中, 可以用公共邻居的个数来计算两个节点的结构聚合度. 两个节点共享的公共邻居越多, 它们就越相似. 如果一个节点具有较大的影响力, 那么它将与其他节点更加聚集. 节点结构聚合度 (SCD) 定义为:

$$SCD(i,l) = \frac{\sum_{k \in (F_{(i)} \cap F_{(l)})} Inf(k)}{\sqrt{\sum_{k \in F_{(i)}} Inf(k)} \sqrt{\sum_{k \in \cap F_{(l)}} Inf(k)}} \tag{4}$$

节点的全局相似度考虑了基于局部相似度 u_{il} 与节点的结构聚合度 SCD, 利用加权和将这两个因素相结合, 即可得到用户全局相似度 Sim, 其定义如下:

$$Sim(i,l) = \alpha u_{il} + (1-\alpha) SCD(i,l) \tag{5}$$

其中, 参数 $\alpha \in [0,1]$ 是根据实际情况设置的权重因子, 用以控制两个因素的比例大小, 具体的取值在实验部分给出.

2.3 社区中心点的计算

在进行重叠检测之前, 首先要选择一个初始节点, 最简单的方法是根据节点度排序选择节点度最大的节点为初始节点, 但这种方法并不可取, 因为节点度最大的节点并不能保证是最重要的. 在一个网络社区中, 其中心节点是社区的核心, 应该与其他节点有着较为密切的联接, 从而中心节点通常会具有较高的度. 同时, 由社区中心节点关联的节点间应该具有较高的相似性. 本节通过计算节点的内聚度和分离度作为度量节点对社区结构影响力的重要性指标, 从而提出了一种社区中心点的选取方法.

定义 1. 节点内聚度. 网络中节点 i 的内聚度是指该节点的时间加权用户 – 用户矩阵及其与邻居节点的最大全局相似度之积, 形式化表示为:

$$I_i = U_i \times \max_{l \in F(i)} Sim(i,l) \tag{6}$$

由上式可知节点 i 的内聚度 I_i 同时考虑了节点的连接数量和全局相似度两个因素, 节点的内聚度越高, 表示该节点对社区内其他节点的聚合能力会越强.

由于网络社区的外部连接通常是相对稀疏的, 因此社区中心节点与其他内聚度较高的节点应该具有较低的相似性. 这一特征可以用节点的分离度来表示.

定义 2. 节点分离度. 网络中节点 i 的分离度是内聚度高于 i 的节点与该节点之间的最大全局相似度的倒数, 形式化表示为:

$$O_i = \frac{1}{1 + \max\limits_{I_l > I_i} Sim(i,l)} \tag{7}$$

其中, O_i 为节点 i 的分离度, 其取值越大表明节点 i 与内聚度更大的节点之间具有较低的相似性.

社区的中心点是对社区结构具有最大影响力、与内部具有较高的内聚度以及与其他内聚度较高的节间具有较低的相似性的节点. 因此, 可以用节点的中心度来表示其影响力.

定义 3. 节点中心度. 网络中节点 i 的中心度是该节点的内聚度与分离度的乘积, 形式化表示为:

$$R_i = I_i \times O_i \tag{8}$$

其中, R_i 为节点 i 的中心度, 节点的中心度越高, 则该节点成为社区中心的可能性就越大.

2.4 重叠社区检测

我们检测时间加权用户 – 用户图中的重叠社区之前, 首先根据节点的中心度排序来选择初始节点. 其次, 我们规定社区中的节点停止增长了才能进行

节点删除操作. 再次, 为了避免死循环, 我们规定初始节点不得删除. 利用节点中心度的概念来衡量节点的重要性, 选择节点中心度最大的节点为初始节点, 通过使如下效用函数最大化便可实现社区检测:

$$f_k = \frac{w_k^{\text{in}}}{w_k^{\text{in}} + w_k^{\text{out}}} \tag{9}$$

其中, w_k^{in} 表示第 k 个社区的总体内部度, 且等于第 k 个社区所有链接权重的两倍; w_k^{out} 表示第 k 个社区总体外部度, 且等于第 k 个社区内部节点和外部节点间的链路权重之和; $k \in \{1, \cdots, K\}$ 且 K 表示重叠社区的总体数量.

重叠社区的检测步骤如下所示.

算法 1. Overlapping community detection algorithm

步骤 1. 选择整个节点中心度最高的节点 A 作为起始节点;

步骤 2. 通过如下步骤检测出这个节点的自然社区:

1) 利用被选节点对社区 C 初始化, 将社区的初始适应度设置为 0;

2) 确定社区 C 有哪些相邻节点没有包含在 C 中但与 C 中节点具有直接联系;

3) 确定每个相邻节点对于社区 C 的适应度, 即存在和不存在相邻节点时社区 C 的适应度变化. 从所有相邻节点中选择正值适应度最大的节点纳入社区 C, 然后再次计算社区的适应度.

4) 重复步骤 2) 和 3), 直到没有相邻节点对社区 C 的适应度为正;

5) 计算 C 中各个节点的适应度, 即包含和不包含该节点时社区 C 的适应度变化. 删除与社区 C 的适应度为负且数值最大的节点 (该社区的起始节点例外), 然后再次计算社区的适应度;

6) 重复步骤 5), 直到社区 C 中没有节点的适应度为负.

步骤 3. 如果存在部分节点未被分配到任何当前社区, 则从这些节点中选择节点中心度最高的节点, 然后跳到步骤 2); 否则, 输出最终社区.

2.5　社区融合

如果利用社区检测算法获得的两个社区中包含了太多重叠节点, 则应该将这些节点融入到一个社区中. 通过计算重叠比例可以确定这两个社区是否应该融合. 当两个社区重叠节点的比例均较高时, 则可将这两个社区进行合并.

$$\delta_{pq} = \frac{|C_p \cap C_q|}{\min\left(|C_p|, |C_q|\right)} \tag{10}$$

其中, C_p 和 C_q 表示第 p 个和第 q 个重叠社区的用户集合, $\min\left(|C_p|, |C_q|\right)$ 表示社区 p 或 q 中节点最少的某个社区的节点数目. $|\cdot|$ 表示社区集或节点集中的节点数量, 设置融合阈值 $\beta \in [0, 1]$. 如果 $\delta_{pq} > \beta$, 则将两个社区进行合并. 融合阈值具体的取值在实验部分给出.

3　实验与性能分析

3.1　实验数据集

1) 人工网络数据集

LFR 基准程序是近年来广泛使用的人工基准网络生成工具, 因为其生成的网络可以很好地表示出节点度和社区规模分布的异质性. 通过设置不同的参数可以生成不同的网络结构, 表 1 给出了 LFR 基准网络生成参数的说明, 表 2 给出了根据 LFR 中参数的不同取值所生成的三个数据集信息, 分别记为 S1, S2 和 S3.

表 1　LFR 基准网络生成参数说明

Table 1　Parameter setting of LFR benchmark network generation

参数	说明
N	网络的节点数目
k	网络中节点的平均度数
$C_{\min}$	最小社区的节点数目
$C_{\max}$	最大社区的节点数目
on	重叠节点的个数
om	重叠节点所从属的社区个数
mu	社区混合参数

表 2　人工网络数据集

Table 2　Artificial network datasets

编号	N	k	$C_{\min}$	$C_{\max}$	mu	on	om
S_1	10 000	20	50	100	0.1 ∼ 0.7	1 000	3
S_2	100 000	20	100	200	0.1 ∼ 0.7	5 000	2
S_3	10 000	20	100	200	0.1	1 000	2 ∼ 7

2) 真实网络数据集

为了检测算法在真实网络上的性能, 选用 6 个真实网络数据集对本章提出算法进行验证, 包括 Zachary 空手道俱乐部成员关系网络 (Karate)、海豚社会网络 (Dolphins)、美国政治书网络 (Polbooks) 和美国大学足球网络 (Football) 等. 本文选取了两个具有代表性的真实数据集: Polblogs 和 DBLP. 数据集如表 3 所示.

3.2　实验方法与评价指标

为了对比本文提出的 TWOCD 算法性能, 选取目前重叠社区发现的主流算法 CPM[5]、COPRA[25]、LFM[10] 对比实验, 对比实验将在不同的人工数据集

和真实数据集上进行验证, 从而对 TWOCD 算法的性能进行分析. 对比算法的简介如下:

CPM: 由 Palla 等提出的基于派系过滤的算法, 基于 K 极大团发现重叠社区.

LFM: 由 Lancichinetti 等提出的一种基于局部扩展的重叠社区发现算法, 通过局部适应度函数决定是否加入社区.

COPRA: 由 Gregory 等提出的一种基于标签传播的重叠社区发现算法, 为每个节点保留了多个标签, 根据标签进行重叠社区的发现.

表 3 真实数据集

Table 3 Real datasets

编号	名称	节点数	边数	平均度
R1	Karate	34	78	4.75
R2	Doplphins	62	159	5.13
R3	Polbooks	105	441	8.40
R4	Football	115	613	10.66
R5	Folbogs	1 490	16 715	22.44
R6	DBLP	4 000	8 301	2.52

本节介绍算法性能评估指标, 包括标准化互信息 (NMI) 和模块度 (Q). 当时效网络的社区结构真实情况已知时, 采用 NMI 和错误率指标; 否则, 使用模块度指标.

标准化互信息 (NMI) 指标定义为:

$$NMI=\frac{\sum_{i=1}^{K^t}\sum_{j=1}^{K^s}n_{ij}\log_2\left(\frac{n\cdot n_{i,j}}{n_i^T\cdot n_j^s}\right)}{\sqrt{\left(\sum_{i=1}^{K^t}n_i^r\log_2\frac{n_i^r}{n}\right)\left(\sum_{j=1}^{K^s}n_j^s\log_2\frac{n_j^s}{n}\right)}} \tag{11}$$

其中, n 表示网络节点的数量, K^r 和 K^s 分别表示真实网络结构的社区数量及本文算法获得的社区数量; n_i^r、n_j^s 和 n_{ij} 分别表示真实网络结构第 i 个社区的节点数量, 本文算法获得的第 j 个社区的节点数量, 以及第 i 和第 j 个社区的共同节点数量; NMI 的数值范围在 0∼1 之间. 数值越接近于 1, 表明社区发现结果越接近真实值.

模块度 (Q) 定义为:

$$Q=\frac{1}{2m}\sum_{ij}\left(A_{ij}-\frac{d_id_j}{2m}\right)\delta(C_i,C_j) \tag{12}$$

其中, m 表示网络中边缘总量, A_{ij} 表示网络邻接矩阵的元素, d_i 表示节点 i 的度, C_i 表示节点 i 所隶属的社区. 如果节点 i 和第 j 属于同一社区, 则 $\delta(C_i,C_j)=1$; 否则, $\delta(C_i,C_j)=0$. 总体来说, Q 值越接近 1, 社区划分的结果越好.

3.3 参数 α 和 β 取值的验证实验

参数 α 是用户全局相似度计算的权重因子, 用以控制局部相似度与结构聚合度在全局相似度计算时的比例大小. 为分析参数 α 对本文算法社区发现结果产生的影响, 我们在社区取不同数量下计算参数 α 的取值对社区发现结果的 Q 值影响情况. 图 1 显示 Polblogs 数据集中各种 α 值对 Q 值的影响值. 通过比较图 1 中 K 取不同值时 Q 值的结果, 可以看到当 $K=2$ 时社会结构最优. 这是由于 Polblogs 数据集中的包括了保守主义和自由主义两类不同政治倾向的节点, 因此该数据集很自然地分为两个社区. 这也说明我们的社区发现算法结果与实际情况一致.

图 2 显示了各种 α 值对 DBLP 数据集 Q 值的影响情况. 与政治观点是社区重要特征的 Polblogs 数据不同, DBLP 社区考虑了合作者关系. 因此, 由图 2 可以看出, 通过重复实验, 当 $K=50$ 时, Q 的平均值最佳. 由图 1 和图 2 的验证结果可知, 当参数 $\alpha=0.4$ 时, 社区模块度 Q 达到了最佳值, 因此本算法中的参数 α 最终取值为 0.4.

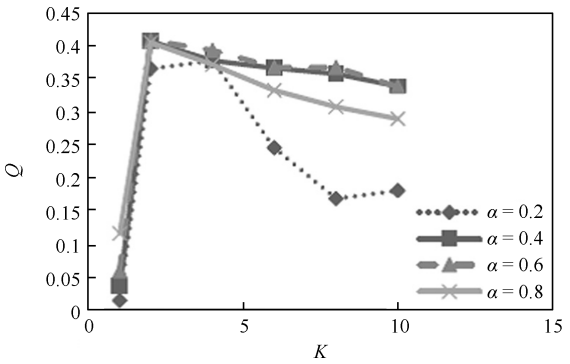


图 1 Polblogs 数据集中参数 α 对 Q 值的影响结果

Fig. 1 The influence of different α on the Q in Polblogs data set

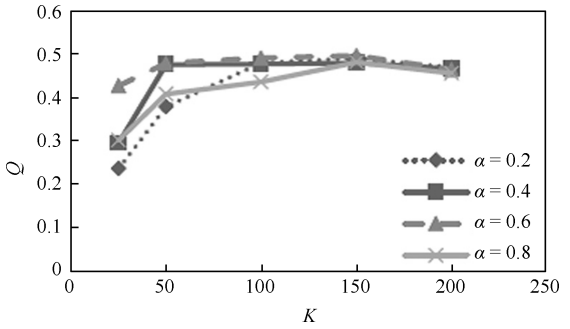


图 2 DBLP 数据集中参数 α 对 Q 值的影响结果

Fig. 2 The influence of different α on the Q in DBLP data set

参数 β 是社区融合阈值, 用以控制两个相似社区是否应该合并, 因此控制重叠度的阈值 β 直接影

响了最终社区数量. 本节实验用模块度 Q、社区数量对 β 的最佳取值进行验证. 根据不同社区数 K 下的模块度 Q 进行对比, 结果如图 3 所示.

一般社区模块度在 [0.3, 0.7] 之间被认为是一个好的社区发现算法. 图 3 中 $S1$ 的模块度在社区数量为 12 时达到最优, $S2$ 和 $S3$ 则分别在社区数量为 15 和 18 时达到最优. 其次, 再将不同 β 下的社区个数进行对比, 结果如图 4 所示, 可以发现 β 取值为 0.8 时社区数量在 15～18 之间, 说明此时划分结果中的重叠比例最接近真实情况, 因此社区融合阈值 β 的值最终设置为 0.8.

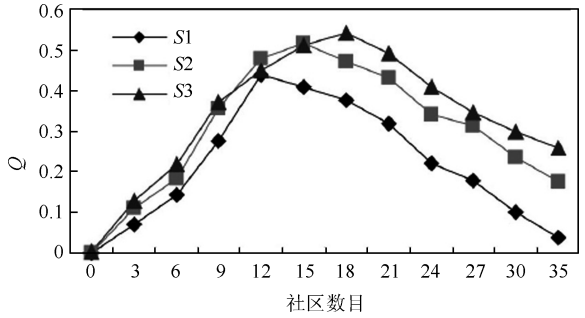


图 3　重叠社区的模块度随社区数量的变化情况

Fig. 3　The influence of different community number on the K

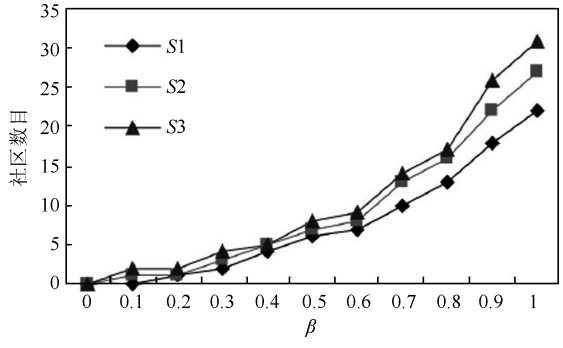


图 4　社区数量随阈值 β 的变化情况

Fig. 4　The influence of different β on different community number K

3.4　算法性能验证实验

本节分别在人工数据集和 6 个真实网络数据集上进行算法性能的验证实验, 以此检验本文所提出的重叠社区检测算法在检测性能和检测效率上的正确性和高效性.

3.4.1　人工数据集上的实验结果

表 4 给出了人工网络参数 mu 在不同取值时各算法在人工网络 S1 上的 NMI 实验结果. 从表 4 中数据可以看出, 随着 mu 逐渐增大, 各算法的 NMI 值在逐渐减小, 当 mu 值增大到一定程度时, 社区识别算法将会失效. 本文提出的 TWOCD 算法在 mu 取不同值时均具有较好的 NMI 值, 这主要是由于 TWOCD 算法在执行过程中构建了更合理的用户–用户关系图, 使得用户的边集合更高效, 即便在处理复杂的网络时, 也能保证算法具有较高的精度和社区发现的稳定性.

表 4　mu 在不同取值时各算法在人工网络 S1 上的 NMI 实验结果

Table 4　NMI experimental results of different algorithms on S1 under different mu value

mu	0.1	0.2	0.3	0.4	0.5	0.6	0.7
TWOCD	0.53	0.45	0.34	0.24	0.23	0.22	0.22
COPRA	0.85	0.79	0.71	0.62	0.42	0.22	0.22
CPM	0.82	0.8	0.62	0.48	0.21	0.22	0.22
LFM	0.52	0.42	0.35	0.22	0.22	0.22	0.22

表 5 给出了人工网络参数 om 取不同值时, 各算法在人工网络 S2 上的 NMI 实验结果. 从表中的结果可以观察到, 当参数 om 增大时, 即网络中每个重叠节点隶属的社区数增加时, 各算法的 NMI 值随之减小. 尽管如此, 本文提出的 TWOCD 算法在 om 取不同值时均具有较好的 NMI 值, 这主要是由于 TWOCD 算法在最后通过社区重叠度进行判断, 将重叠度高的社区进行了合并, 有效缓解社区结构过度重叠的问题, 提高算法的识别效率与社区发现的稳定性.

表 5　om 在不同取值时各算法在人工网络 S2 上的 NMI 实验结果

Table 5　NMI experimental results of different algorithms on S2 under different om value

om	2	3	4	5	6	7	8
TWOCD	0.92	0.95	0.88	0.84	0.78	0.72	0.75
COPRA	0.93	0.9	0.83	0.78	0.71	0.65	0.6
CPM	0.92	0.87	0.81	0.78	0.69	0.62	0.62
LFM	0.76	0.68	0.66	0.67	0.65	0.66	0.5

表 6 给出了人工网络参数 on 取不同值时各算法在人工网络 S3 上的 NMI 实验结果. 参数 on 的增大意味着网络中更多的节点隶属于重叠社区. 由表中的结果可以看出, 随着参数 on 的增大, 各算法的 NMI 值都不断减小. 但是, TWOCD 的 NMI 值下降趋势较其他算法较慢. 而且本文提出的 TWOCD

表 6　on 在不同取值时各算法在人工网络 S3 上的 NMI 实验结果

Table 6　NMI experimental results of different algorithms on S3 under different on value

on	1 000	2 000	3 000	4 000	5 000	6 000	7 000
TWOCD	0.95	0.95	0.89	0.76	0.67	0.56	0.38
COPRA	0.89	0.83	0.78	0.58	0.32	0.21	0.21
CPM	0.82	0.84	0.79	0.7	0.6	0.47	0.28
LFM	0.43	0.33	0.23	0.23	0.24	0.24	0.24

算法在 on 取不同值时均具有较好的 NMI 值, 这主要是由于 TWOCD 算法在社区发现时的初始种子节点选取时选择了中心度最大的节点, 中心度大说明该节点的影响力强, 因此将这样的节点作为起始节点将更加合理.

综上, 在不同人工数据集上本文算法获得了优于其他算法的重叠社区发现结果.

3.4.2 真实数据集上的实验结果

在真实网络上对将本文算法与各种重叠社区发现算法的性能进行对比, 各算法的参数选取均使用最优参数配置, 图 5 给出了各算法在真实网络上社区发现的对比结果.

几种算法的参数均根据文献建议进行设置, 实验中各算法的参数取值设置如下: COPRA 中参数 v 表示节点携带的最大标签数, 参数 v 的取值在 $2\sim15$ 之间; LFM 中的参数 α 用于控制社区规模, 参数 α 的取值在 $0.5\sim1.5$ 之间; CPM 的参数 K 在 $1\sim10$ 之间. 通过实验结果可以发现相对于其他 4 种算法, 由于考虑到了用户全局相似度和时效因素, TWOCD 算法在多数网络上取得了最好的重叠模块度值.

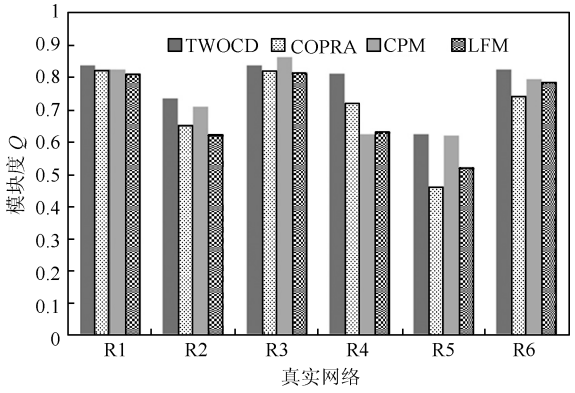


图 5 真实数据集上各算法性能对比实验

Fig. 5 Comparison results of different algorithms on real networks

表 7 给出了各算法在真实网络上社区发现结果及最优参数值, 在不同数据集下, 计算出了参数取不同值时算法的模块度指标性能. 由于本文算法在用户相似度计算及中心度计算上都较对比算法有所改进, 因此在这些真实网络中, 本文算法 TWOCD 在大部分情况下都取了最高的模块度 Q. 并且, 本文算法在不同网络上获得最大模块度时对应的参数 α 和 β 取值变化不大, 这也验证了这两个参数最优取值的有效性和通用性.

3.5 算法运行时间性能分析

本节将通过对比不同算法在 LFR 基准数据集上的实验效果来验证本文所提算法的时间性能优势. 在 S1 网络上, 固定 $mu = 0.1$, N 取 $10\,000\sim70\,000$, 保持其他参数不变. 各算法在不同规模人工网络数据集上的运行性能如图 6 所示. 由图 6 可知, CFinder 算法运行效率最低, 由于该算法以派系为单位计算社区的重叠度, 因此计算量过大, 当网络数量增加到一定值后算法失效; CPM 算法的时间复杂度为非多项式级; COPRA 算法的计算量与算法的

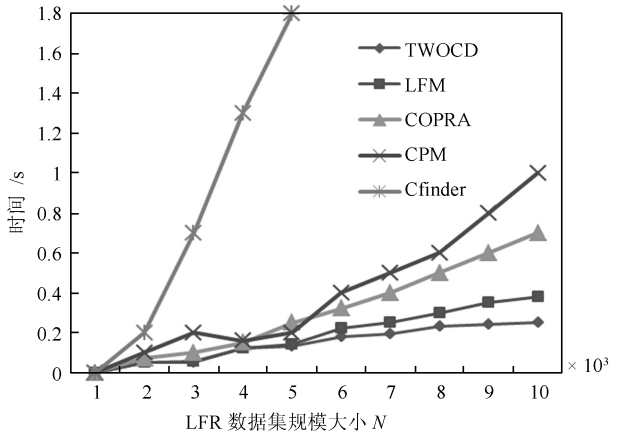


图 6 不同算法运行时间比较

Fig. 6 Execution time comparison of different algorithms

表 7 真实数据集上各算法在不同参数取值下性能对比结果

Table 7 Comparison results of different algorithms on different parameter in real networks

Data set	LFM		COPRA		CPM		TWOCD	
	α	Q	v	Q	k	Q	α, β	Q
Karate	0.8	0.813	2	0.825	2	0.823	0.4, 0.8	**0.8463**
Football	1.1	0.645	2	0.656	4	0.707	0.4, 0.8	**0.7246**
Doplphins	0.8	0.812	6	0.821	5	**0.924**	0.3, 0.7	0.9005
Polbooks	0.9	0.634	8	0.717	8	0.795	0.4, 0.9	**0.7342**
Folbogs	1.4	0.122	2	0.466	6	0.625	0.4, 0.8	**0.6257**
DBLP	0.8	0.787	9	0.745	3	0.797	0.4, 0.8	**0.8143**

迭代次数有关, 因此当网络规模较小时算法性能具有较大的优势; LFM 算法是随机选择种子节点进行扩展, 其局部最优化的思想使得算法具有较高的计算效率. 本文算法 TWOCD 在社区发现算法中的初始节点选择上, 优化了社区中心度的计算方法, 使得初始种子节点的选取更有价值, 因此较好地降低了算法的计算复杂度.

4 结论

本文提出一种新颖的重叠社区发现算法 TWOCD, 该算法充分考虑了用户兴趣的时间因素, 根据带有时间加权链接的用户 – 用户图实现重叠社区检测. 在社区发现迭代计算时选择中心度最大的节点为种子节点, 提高了社区发现在精准度. 最后通过重叠度计算将重叠过多的社区进行合并, 从而提高了算法执行的效率. 在仿真实验中, 利用人工网络数据和真实网络数据进行有效性验证, 实验结果表明, 本文提出的算法在社区发现质量和计算效率上优于已有算法. 未来的工作计划将该算法应用于为各类复杂网络提供社区识别服务, 进而为用户提供更加个性化的社区服务.

References

1 Newman M E J. The Structure and function of complex networks. *SIAM Review*, 2003, **45**(2): 167−256

2 Pan Jian-Fei, Dong Yi-Hong, Chen Hua-Hui, et al. The overlapping community discovery algorithm based on compact structure. *Acta Electronic Sinica*, 2019, **47**(001): 145−152
(潘剑飞, 董一鸿, 陈华辉, 等. 基于结构紧密性的重叠社区发现算法. 电子学报, 2019, **47**(001): 145−152)

3 Li Hui, Ma Xiao-Ping, Shi Jun, Li Cun-Hua, Zhong Zhaoman, Cai Hong. A recommendation model by means of trust transition in complex network environment. *Acta Automatica Sinica*, 2018, **44**(2): 363−376
(李慧, 马小平, 施珺, 李存华, 仲兆满, 蔡虹. 复杂网络环境下基于信任传递的推荐模型研究. 自动化学报, 2018, **44**(2): 363−376)

4 Xu M, Li Y, Li R, et al. EADP: An extended adaptive density peaks clustering for overlapping community detection in social networks. *Neurocomputing*, 2019, **337**(14): 287−302

5 Gergely P, Imre D, Illés F, Tamás V. Uncovering the overlapping community structure of complex networks in nature and society. *Nature*, 2005, **435**: 814−818

6 Farkas I, ábel D, Palla G, et al. Weighted network modules. *New Journal of Physics*, 2007, **9**(6): 180

7 Zhang Z, Wang Z. Mining overlapping and hierarchical communities in complex networks. *Physica A: Statistical Mechanics and Its Applications*, 2015, **421**: 25−33

8 Wang L. Using the relationship of shared neighbors to find hierarchical overlapping communities for effective connectivity in IoT. In: Proceedings of the 6th International Conference on Pervasive Computing and Applications. New York, USA: IEEE, 2011. 400−406

9 Chen Jun-Yu, Zhou Gang, Nan Yu, Zeng Qi. Semi-supervised local expansion method for overlapping community detection. *Journal of Computer Research and Development*, 2016, **53**(6): 1376−1388
(陈俊宇, 周刚, 南煜, 曾琦. 一种半监督的局部扩展式重叠社区发现方法. 计算机研究与发展, 2016, **53**(6): 1376−1388)

10 Lancichinetti A, Fortunato S, Kertész J. Detecting the overlapping and hierarchical community structure in complex networks. *New Journal of Physics*, 2009, **11**(3): 15−33

11 Murphy R J L. Reliability of marking in eight GCE examinations. *British Journal of Educational Psychology*, 2011, **48**(2): 196−200

12 Lancichinetti A, Radicchi F, Ramasco J J, et al. Finding statistically significant communities in networks. *PloS One*, 2011, **6**(4): e18961

13 Yang J, Zhang X D. Finding overlapping communities using seed set. *Physica A: Statistical Mechanics and Its Applications*, 2017, **467**(1): 96−106

14 Su Y J, LeeC C. Overlapping community detection with seed set expansion by local cluster coefficient. In: Proceedings of the International Conference on Consumer Electronics Piscataway, New York, USA: IEEE, 2017. 73−74

15 Gregory S. Fuzzy overlapping communities in networks. *Journal of Statistical Mechanics-Theory and Experiment*, 2011, **2**: P02017

16 Eustace J, Wang X, Cui Y. Overlapping community detection using neighborhood ratio matrix. *Physica A: Statistical Mechanics and its Applications*, 2015, **421**: 510−521

17 Raj E D, Babu L D D. A fuzzy adaptive resonance theory inspired overlapping community detection method for online social networks. *Knowledge-Based Systems*, 2016, **112**(1): 75−87

18 Su Jianhan, Havens T C. Quadratic program-based modularity maximization for fuzzy community deterction in social network. *IEEE Trans. on Fuzzy Systems*, 2015, **23**(5): 1356−1371

19 Javed M A, Younis M S, Latif S, et al. Community detection in networks: A multidisciplinary review. *Journal of Network and Computer Applications*, 2018, **108**(15): 87−111

20 Chen Naiyue, Liu Yun, Chao H C. Overlapping community detection using non-negative matrix factorization with orthogonal and sparseness constraints. *IEEE Access*, 2018, **6**: 21266−21274

21 Ahn Y Y, Lehmann S, Bagrow J, et al. Hierarchical link clustering in complex networks. *American Physical Society*, 2009, **3**(2009): 1−10.

22 Shi C, Cai Y, Fu D, et al. A link clustering based overlapping community detection algorithm. *Data & Knowledge Engineering*, 2013, **87**: 394−404

23 Lim S, Ryu S, Kwon S, et al. LinkSCAN: Overlapping community detection using the link-space transformation. In: Proceedings of the 30th International Conference on Data Engineering. New York, USA: IEEE, 2014. 292−303

24 Li M M, Liu J. A link clustering based memetic algorithm for overlapping community detection. *Physica A: Statistical Mechanics and Its Application*, 2018, **503**(1): 410−423

25 Gregory S. Finding overlapping communities in networks by label propagation. *New Journal of Physics*, 2009, **12**(10): 2011−2024

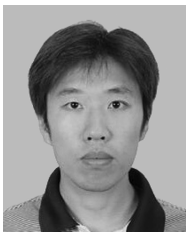
26 Xie J, Szymanski B K. Towards linear time overlapping community detection in social networks. In: Proceedings of the Pacific-Asia Conference on Knowledge Discovery and Data Mining, New York, USA: IEEE, 2012. 25−36

27 Gaiter C, Chen M M, Szymanski B, et al. Identifying robust communities and multi-community nodes by combining top-down and bottom-up approaches to clustering. *Scientific Reports*, 2015, **5**: 16361

28 Brin S, Page L. Reprint of: The anatomy of a large-scale hypertextual web search engine. *Computer Networks*, 2012, **56**(18): 3825−3833

李 慧 博士, 江苏海洋大学计算机工程学院副教授. 主要研究方向为个性化推荐, 社会网络分析.
E-mail: shufanzs@126.com
(**LI Hui** Ph. D., associate professor at the School of Computer Engineering, Jiangsu Ocean University. Her research interest covers personality recommendation and social network analysis.)

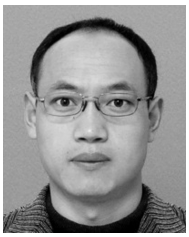
马小平 博士, 中国矿业大学信电学院教授. 主要研究方向为智能计算. 本文通信作者. E-mail: xpma@cumt.edu.cn
(**MA Xiao-Ping** Ph. D., professor at the School of Information and Electrical Engineering, China University of Mining & Technology. His main research interest is intelligent computing. Corresponding author of this paper.)



张 舒 江苏海洋大学商学院讲师. 主要研究方向为智能信息处理.
E-mail: lih@cumt.edu.cn
(**ZHANG Shu** Lecturer at the School of Business, Jiangsu Ocean University. Her main research interest is information processing.)

施 珺 江苏海洋大学计算机工程学院教授. 主要研究方向为智能信息处理.
E-mail: sj_lfg@hotmail.com
(**SHI Jun** Professor in the Department of Computer Science, Jiangsu Ocean University. Her main research interest is information processing.)

李存华 博士, 江苏海洋大学计算机工程学院教授. 主要研究方向为数据挖掘.
E-mail: cli2000@126.com
(**LI Cun-Hua** Ph. D., professor in the Department of Computer Science, Jiangsu Ocean University. His main research interest is data mining.)



仲兆满 博士, 江苏海洋大学计算机工程学院副教授. 主要研究方向为中文信息处理.
E-mail: zhongzhaoman@163.com
(**ZHONG Zhao-Man** Ph. D., associate professor in the Department of Computer Science, Jiangsu Ocean University. His main research interest is Chinese information processing.)