DOI:10.16208/j.issn1000-7024.2014.01.060

基于密度的半监督复杂网络聚类算法

孟凡荣，张可为+，朱 牧

（中国矿业大学 计算机科学与技术学院，江苏 徐州 221116）

摘 要：针对大多数复杂网络聚类算法不能有效利用先验知识的问题，提出了一种基于密度的半监督复杂网络聚类算法。通过已有的成对约束关系及其传递性质发现网络中所有潜在的约束关系，以更充分地指导聚类过程；在基于密度的聚类算法基础上，综合考虑节点之间的可达性以及成对约束关系，以发现网络中满足连通性和最大性的社区结构。将实验结果与其它算法进行比较，比较结果表明了该算法能更加有效的利用先验知识来提高聚类性能。

关键词：复杂网络；聚类；基于密度；半监督；约束

中图法分类号：TP181 **文献标识号**：A **文章编号**：1000-7024（2014）01-0271-05

Density-based semi-supervised clustering algorithm in complex network

MENG Fan-rong, ZHANG Ke-wei+, ZHU Mu

(School of Computer Science and Technology, China University of Mining and Technology, Xuzhou 221116, China)

Abstract: Aiming at the problem that most of the existing clustering algorithms for complex networks can not make use of the prior information effectively, a density-based semi-supervised clustering algorithm is proposed. Firstly, all the hidden pairs of constraints are found by the algorithm via the existing pairs of ones together with their transitivity to make full use of prior information. Then, the community structure, satisfying connectivity and maximality, is discovered by the reach abilities between nodes and all the pairs of constraints. Experimental results compared with other algorithms demonstrate that the proposed algorithm can utilize the small amount of prior information to improve the clustering performance.

Key words: complex network; clustering; density-based; semi-supervised; constraints

0 引 言

目前，复杂网络聚类算法得到了广泛的研究[1-3]，算法主要包括模块度优化算法[4,5]、启发式算法[6,7]、基于模型的算法[8,9]等，文献[10]给出了详细的综述。然而，大多数已有的算法都是无监督的学习方式，不能够利用少量的先验知识对复杂网络的聚类进行有效指导。而往往在很多实际应用中，可以获取少量有关社区结构的先验知识，例如类标签或者成对约束信息等，如何利用这些知识指导复杂网络的聚类，并提高社区发现的性能，具有非常重要的意义[11]。

针对上述问题，本文提出一种基于密度的半监督复杂网络聚类算法，以利用少量的先验知识来有效提高复杂网络的聚类性能。本文算法首先采用 *must-link* 和 *cannot-link* 两种常用的约束对信息来表示先验知识，并根据 *must-link* 的传递性质计算并记录所有相互之间满足 *must-link* 约束的节点集合；然后，在基于密度的复杂网络聚类算法[12]的基础上，根据所获取的所有的约束关系，在对节点集进行遍历过程中识别并记录节点之间相互连通的关系，并在先验知识的指导下发现网络中隐含的连接信息并进一步修改节点的原有遍历方式，从而发现满足连通性和最大性的社区结构。在若干真实复杂网络上的实验结果表明，新算法相对于其它几种代表性的半监督算法，能够更加有效地提高无监督聚类的性能。

收稿日期：2013-04-19；**修订日期**：2013-06-20

基金项目：教育部高等学校博士学科点专项科研基金项目（20110095110010）；江苏省研究生科研创新计划基金项目（CXZZ12_0934）；国家博士后科学基金项目（20070421041）

作者简介：孟凡荣（1962-），女，辽宁沈阳人，博士，教授，研究方向为数据库技术、数据挖掘；+通讯作者：张可为（1989-），男，湖北松滋人，硕士研究生，研究方向为数据挖掘；朱牧（1985-），男，江苏睢宁人，博士研究生，研究方向为机器学习、数据挖掘。

E-mail: zkw19891007@163.com

1 密度聚类算法

给定复杂网络 $G=(V,E)$，V 表示节点的集合，E 表示边的集合，基于密度的复杂网络聚类算法[12]主要包含如下若干定义：

定义 1 节点 v 的节点结构定义为节点 v 和其所有邻居节点组成的节点集，记为 $N(v)$，如下所示

$$N(v)=\{v\}\bigcup\{w\in V\mid(v,w)\in E\} \tag{1}$$

定义 2 节点 v 和节点 w 的结构相似度为将两个节点的节点结构进行余弦相似度计算得到的结果，记为 $Sim(v,w)$，如下所示

$$Sim(v,w)=\frac{\mid N(v)\bigcap N(w)\mid}{\sqrt{\mid N(v)\mid\mid N(w)\mid}} \tag{2}$$

定义 3 节点 v 的近邻节点为节点 v 的节点结构中所有与 v 的结构相似度不小于参数 ε 的节点构成的集合，记为 $N_\varepsilon(v)$，如下所示

$$N_\varepsilon(v)=\{w\in N(v)\mid Sim(v,w)\geqslant\varepsilon\} \tag{3}$$

定义 4 若节点 v 的近邻节点数目不小于设定的参数 μ，则称节点 v 为核节点，表示为 $C_{\varepsilon,\mu}(v)$，如下所示

$$C_{\varepsilon,\mu}(v)\Leftrightarrow\mid N_\varepsilon(v)\mid\geqslant\mu \tag{4}$$

定义 5 如果节点 w 属于节点 v 的近邻节点，而且节点 v 是核节点，那么 v 到 w 是直接结构联通，记为 $Dirr_{\varepsilon,\mu}(v,w)$，如下所示

$$Dirr_{\varepsilon,\mu}(v,w)\Leftrightarrow C_{\varepsilon,\mu}(v)\wedge w\in N_\varepsilon(v) \tag{5}$$

定义 6 节点 v 到节点 w 结构联通即存在一个节点集，使得节点 v 与这些节点通过直接结构联通的传递性可以到达节点 w。

定义 7 结构社区表示算法对网络进行聚类得到的一个社区，这个社区内部的节点之间是结构联通的，并且这个社区包含了所有内部节点的结构联通节点集，即节点数目达到最大。

密度聚类算法遍历样本节点集，通过直接结构联通的传递关系将所有与核节点联系密切的节点都聚到同一个社区中，保证了社区的联通性和最大性，遍历过程中如果节点不是核节点则标记为特殊节点。遍历结束后即完成了对网络的社区划分。

2 半监督密度算法

在实际聚类研究中，往往可以获取少量先验知识，例如类标签或者成对约束信息等，本文选择以约束对信息作为先验知识。其中，*must-link* 和 *cannot-link* 是两种常用的约束对信息：*must-link* 表示两者必须隶属于同一个社区，*cannot-link* 表示两者必须隶属于不同的社区[13]。针对普通密度聚类算法无法有效利用先验知识的问题，本文提出了基于密度的半监督复杂网络聚类算法。

2.1 先验知识的表示

定义 8 C_m 表示所有已知的满足 *must-link* 约束的节点对组成的集合，即

$$C_m=\{(v,w)\mid must-link(v,w)\} \tag{6}$$

定义 9 C_c 表示所有已知的满足 *cannot-link* 约束的节点对组成的集合，即

$$C_c=\{(v,w)\mid cannot-link(v,w)\} \tag{7}$$

定义 10 *TMS* 表示一个由若干自成集合的元素组成并且每个元素内部的任意两个节点之间满足 *must-link* 约束的集合，即

$$TMS=\{V'\mid\forall v,w\in V',must-link(v,w)\} \tag{8}$$

定义 11 *TCS* 表示在已知 *TMS* 集合的情况下对 C_c 集合进行拓展得到的所有满足 *cannot-link* 约束的节点对组成的集合。

must-link 约束具有对称性和传递性。利用这些性质对 C_m 集合进行计算，可以发现隐含的 *must-link* 约束，整合即可知道所有已知的和隐含的 *must-link* 约束对，最后将所有相互之间满足 *must-link* 约束的节点集都保存在 *TMS* 中。*TMS* 的具体计算步骤见算法 1：

算法 1：CALTMS ($G=(V,E)$, C_m)

输入：V，E，C_m
输出：*TMS*

```
01 M=formMustLinkMatrix();
02 for each vertex v_i not in TMS do
03     generate a new set S';
04     if(v_j∈V) && (M(i,j) ==1) then
05         add v_j not in TMS to S';
06         add v_i not in TMS to S';
07         if (v_x∈V) && (M(x,j) ==1) then
08             add v_x not in TMS to S';
09         end if;
10     end if;
11     add S' to TMS;
12 end for;
```

CALTMS 用于根据 C_m 集合计算 *TMS* 集合，为聚类过程提供输入。其中，算法第 1 行用于生成一个 $n\times n$ 维矩阵 M，如果节点 i 和节点 j 满足 $(i,j)\in C_m$，则矩阵对应位置的元素为 1，否则为 0。算法第 3-10 行用于生成一个集合 S'，里面的元素为所有与节点 i 之间满足 *must-link* 约束的节点。所有的子集 S' 共同构成了 *TMS* 集合。

随后，我们根据计算得到的 *TMS* 集合来对 C_c 集合进行拓展，可以得到 *TCS* 集合。内部元素为所有已知的和计算得到的满足 *cannot-link* 约束的节点对。

2.2 算法描述

算法根据前面获取的所有约束关系，识别并影响节点之间相互连通的关系，从而发现满足连通性和最大性的社区结构，见算法 2，这部分是算法的核心部分。

算法 2：TMSSCAN ($G=(V, E)$, ε, μ, C_c, TMS)

输入：V, E, ε, μ, C_c, TMS
输出：聚类结果
01 for each unmarked vertex $v\in V$ do
02 if $C_{\varepsilon,\mu}(v)$ then
03 generate new cluster U , U. id=$newid$;　Q. add(v);
04　while(Q. size>0) do
05　$y=Q$. poll(); //取出 Q 的第一个元素，并删除
06　if($y\in c_j$) && ($c_j\in TMS$) then
07　for each unmarked or special vertex o in c_j do
08　o. type=$newid$;　Q. add(o);
09　end for;
10　end if;
11　$R=\{x\in V \mid Dirr_{\varepsilon,\mu}(y, x)\}$
12　for each $x\in R$ do
13　if each vertex p in $Q(x, p)\notin(TCS$ then
14　x. type=$newid$;　Q. add(x);
15　end if;
16　end for;
17　end while;
18 else
19　v. type=special;
20　end if;
21 end for;

TMSSCAN 开始时默认所有的节点都是未标记的。算法从第 1 行开始遍历所有的样本节点，第 2 行用于判断当前节点 v 是否为核节点，如果是则检索与节点 v 连接紧密的所有节点，即 TMS 集合中节点 v 所在子集的其它节点（6-10 行）和节点 v 的直接结构联通节点（11 行），分析这些节点与当前划分跟 TCS 集合是否存在矛盾，不矛盾时将节点 v 和这些节点分配到同一个社区中，存在矛盾则分析下一个与节点 v 连接紧密的节点，检索结束即得到一个由节点 v 拓展得到的社区（3-17 行）；如果节点 v 不是核节点，则将其标记为特殊节点（19 行）。如此将样本节点全部遍历一遍，即可得到最终的聚类结果。

聚类过程中先验知识的指导作用主要体现在如下部分：TMSSCAN 的第 6-10 行用于将所有与当前节点满足 *must-link* 约束的节点都加入到队列中，保证最后能聚到同一个社区；第 13 行检测能否将当前节点加入队列，使得不将满足 *cannot-link* 约束的两个节点聚到同一个社区。如此，算法在对节点集进行遍历时充分考虑了 TMS 集合和 TCS 集合，保证了先验知识的有效性。

2.3　复杂度分析

CALTMS 用于计算 TMS 集合，其时间复杂度跟 C_m 集合有关。考虑最坏的情况，即所有的节点标签信息已知，在构造子集 S_1' 时，将遍历标签信息为第一类的节点，设其时间复杂度为 $O(a_1)$，设构造子集数为 b，则总共时间复杂度为 $O(a_1+a_2+a_3+\cdots+a_b)$，其中 $a_1+a_2+a_3+\cdots+a_b=n$，因此在最坏情况下 CALTMS 的时间复杂度为 $O(n)$。

TMSSCAN 在聚类过程中，需遍历网络中的每个节点。对每个节点 v，必须检索其所有邻居节点，其时间复杂度为 $O(deg(v))$。所以最终的时间复杂度为 $O((deg(v_1)+deg(v_2)+deg(v_3)+\cdots+deg(v_n))/2)$，其中 $deg(v_1)+deg(v_2)+\cdots+deg(v_n)=2m$，因此 TMSSCAN 聚类过程的时间复杂度为 $O(m)$。

综上所述，基于密度的半监督复杂网络聚类算法的时间复杂度为 $O(m+n)$。在对实际网络进行分析时，网络中的边数和节点数大致相当，即 m≈n，所以算法时间复杂度可以记为 $O(m)$。可知，本文所提算法的时间复杂度较低，能应用于对大规模复杂网络进行聚类研究。

3　实验分析

实验环境为：处理器 Inter（R）Core（TM）2 Duo 2.8GHz PC，内存 2G，操作系统为 Windows 7，编程环境为 Visual Studio 2008 C++.Net。

3.1　数据

为了验证算法的聚类性能，我们在 *Newman* 数据中的真实网络上进行了聚类分析，网络的具体信息见表 1。

表 1　*Newman* 真实网络数据

名称	节点数	边数	社区数目
football	115	613	12
karate	34	78	2
polbooks	105	441	3
adjnoun	112	425	2

3.2　方法

我们用基于密度的半监督复杂网络聚类算法（*SS-SCAN*）对表 1 中的数据进行聚类分析，将结果与半监督谱聚类算法（*SS-SP*）[13]、基于 *Ncut* 的半监督核 *K-means* 算法（*SS-KK*）[13]的聚类结果进行了对比。

实验中，我们选择被广为使用的 *NMI*[14] 作为算法聚类精度的评价标准，其计算公式如下

$$NMI=\frac{-2\sum_{i,j}N_{ij}\log(\frac{N_{ij}N}{N_{i.}N_{.j}})}{\sum_i N_{i.}\log(\frac{N_{i.}}{N})+\sum_j N_{.j}\log(\frac{N_{.j}}{N})} \tag{9}$$

式中：N——一个矩阵，N_{ij}——既在类 X_i 中又在簇 Y_j 中的节点的个数，$N_{i.}$——矩阵第 i 行求和的结果，$N_{.j}$——矩阵第 j 列求和的结果。

3.3　结果

我们分别用 *SS-SCAN*、*SS-SP* 和 *SS-KK* 算法在 4 个数据集上进行聚类分析，实验结果如图 1、图 2、图 3、图 4 所示。其中横坐标表示数据中已知标签节点的比例，考虑实际情况中已知比例不会太多的问题，其取值范围为[0，50%]；纵坐标表示聚类结果的 *NMI* 值，*NMI* 值越高表示聚类精度越高。

实验表明在使用半监督学习之前（对应图中已知比例

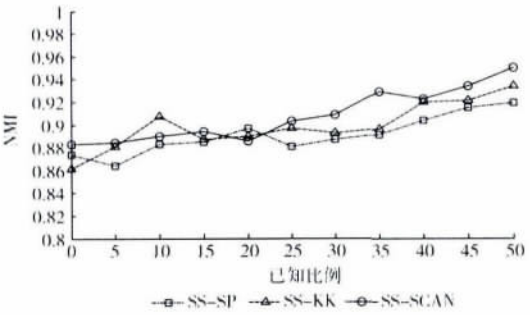

图 1 *football* 数据聚类结果

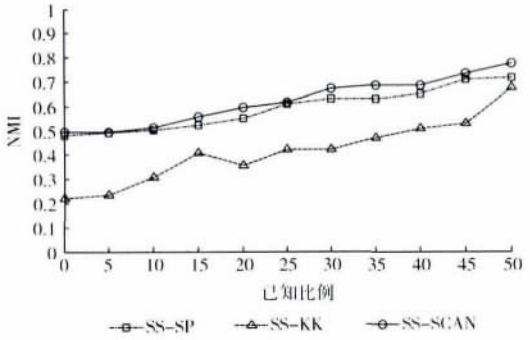

图 2 *karate* 数据聚类结果

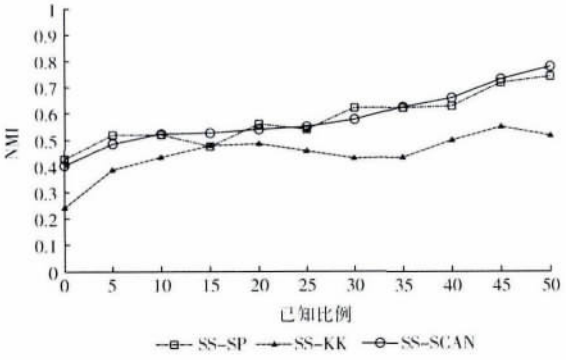

图 3 *polbooks* 数据聚类结果

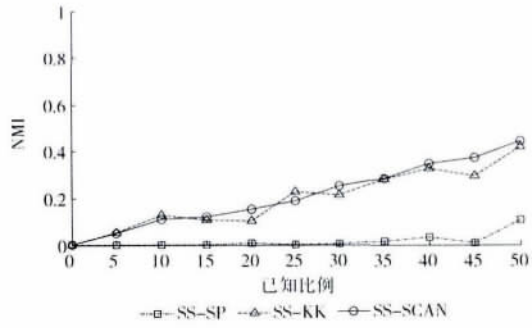

图 4 *adjnoun* 数据聚类结果

为 0 的聚类结果），*SS-SCAN* 算法在 *football* 数据和 *karate* 数据上的聚类精度相比其它两个算法更高，但在其它两个数据上的聚类结果就没有这个现象，说明在使用半监督学习之前，*SS-SCAN* 相比其它两个算法在聚类精度上并没有绝对优势。

使用半监督学习后（对应图中已知比例不为 0 的聚类结果），可以发现，*SS-SP* 算法在前 3 个数据集上的聚类精度比较高，但其波动现象相对比较明显（这是由于算法利用 *K-means* 来对网络提取的特征向量进行聚类，*K-means* 不稳定导致的），且在 *adjnoun* 上的聚类精度是最低的；*SS-KK* 算法在 *football* 和 *adjnoun* 上的聚类精度比较高，在另外两个数据集上的表现相对较差；*SS-SCAN* 算法的聚类精度相比其它两个算法在大多数情况下都是最高的，并且其 *NMI* 值随着已知标签比例的增加大致符合线性增长的趋势，表现相对稳定。

同时，我们统计了 3 个算法对 4 个数据集进行聚类所消耗的平均时间。在实验中我们发现引入半监督策略对算法本身的运行时间并不会有很大程度的改变，在这里我们只给出算法在各种已知比例情况下对 4 个数据进行聚类的平均时间，见表 2。可知 *SS-SCAN* 算法的时间消耗最少。

表 2 算法的聚类时间

	football	*polbooks*	*adjnoun*	*karate*
SS-SP	109.9	84.1	103.4	7.7
SS-KK	242.2	176.1	405.1	11.6
SS-SCAN	71.3	42.2	44.3	7.8

综上所述，*SS-SCAN* 算法的聚类精度更高，聚类结果更稳定，时间消耗更少，能够更有效的利用先验知识指导聚类。

4 结束语

本文提出了一种基于密度的半监督复杂网络聚类算法，以利用少量的先验知识提高复杂网络的聚类质量。该算法主要包含两个重要组成部分：一是利用少量的 *must-link* 和 *cannot-link* 两种常用的成对约束信息，发现网络中所有潜在的成对约束，最大化先验知识，以更好的提高聚类质量；二是在基于密度的方法基础上，判断两个节点是否属于同一个社区时，综合考虑了节点之间的可达性和成对约束两方面的因素，从而更加准确地对复杂网络进行有效划分。在若干真实网络上的实验结果表明，本文所提出的算法相对于其它几种代表性的半监督聚类算法，更能够有效利用少量的先验知识，稳定地提高聚类的质量。

参考文献：

[1] Fortunato Santo. Community detection in graphs [J]. Physics Reports, 2010, 486 (3-5): 75-174.

[2] ZHAO Yunpeng, Levina Elizaveta, ZHU Ji. Community extraction for social networks [J]. Proceedings of the National

Academy of Sciences of the United States of America，2011，108 (18)：7321-7326.

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

[4] Mucha Peter J，Richardson Thomas，Macon Kevin，et al. Community structure in time-dependent，multiscale，and multiplex networks [J]. Science，2010，328 (5980)：876-878.

[5] ZHANG X S，WANG R S，WANG Y，et al. Modularity optimization in community detection of complex networks [J]. Europhysics Letters，2009，87 (3)：38002.

[6] Good Benjamin H，Montjoye Yves Alexandre de，CLAUSET Aaron. Performance of modularity maximization in practical contexts [J]. Physical Review E，2010，81 (4)：046106.

[7] Tantipathananandh Chayant，Tanya Berger Wolf. Constant-factor approximation algorithms for identifying dynamic communities [C] //New York：Proceedings of the 15th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining，2009：827-836.

[8] Hofman J M，Wiggins C H. Bayesian approach to network modularity [J]. Physical Review Letters，2008，100 (25)：258701.

[9] Clauset Aaron，Moore Cristopher，Newman M E J. Hierarchical structure and the prediction of missing links in networks [J]. Nature，2008，453 (7191)：98-101.

[10] Coscia Michele，Giannotti Fosca，Pedreschi Dino. A classification for community discovery methods in complex networks [J]. Statistical Analysis and Data Mining，2011，4 (5)：512-546.

[11] ZHAO Weizhong，MA Huifang，LI Zhiqing，et al. Efficiently active learning for semi-supervised document clustering [J]. Journal of Software，2012，23 (6)：1486-1499 (in Chinese). [**赵卫中，马慧芳，李志清，等. 一种结合主动学习的半监督文档聚类算法** [J]. **软件学报**，2012，23 (6)：1486-1499.]

[12] Xu Xiaowei，Yuruk Nurcan，Feng Zhidan，et al. SCAN：A structural clustering algorithm for networks [C] //New York：Proceedings of the 13th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining，2007：824-833.

[13] MA Xiaoke，GAO Lin，YONG Xuerong，et al. Semi-supervised clustering algorithm for community structure detection in complex networks [J]. Physica A：Statistical Mechanics and its Applications，2010，389 (1)：187-197.

[14] TANG Lei，WANG Xufei，LIU Huan. Community detection via heterogeneous interaction analysis [J]. Data Mining and Knowledge Discovery，2012，25 (1)：1-33.

(**上接第** 106 **页**)

[8] GObject reference manual [EB/OL]. [2013-03-06]. http：//developer. gnome. org/gobject/stable/.

[9] ZHANG Haibin，LI Hui. Design and implementation of embedded high definition player [J]. Computer Engineering and Design，2010，31 (13)：3048-3087 (in Chinese). [**张海滨，李挥. 嵌入式高清播放器的设计与实现** [J]. **计算机工程与设计**，2010，31 (13)：3084-3087.]

[10] HONG Chengyu. Design and implementation of the embedded multimedia system based on GStreamer [D]. Chengdu：Chengdu University of Technology，2009：33-38 (in Chinese). [**洪承煜. 基于** GStreamer **嵌入式多媒体系统的设计与实现** [D]. **成都：成都理工大学**，2009：33-38.]

[11] Summerfield Mark. Advanced Qt programming：Creating great software with C++ and Qt 4 [M]. **北京：电子工业出版社**，2011.

[12] Richard Stevens W. UNIX network programming volume 2：Interprocess communications [M]. **北京：清华大学出版社**，2002：1-395.