[SCAN：一种基于密度的社团发现算法](http://blog.csdn.net/dawnranger/article/details/51108433)

标签： [clustering](http://www.csdn.net/tag/clustering)[社团发现](http://www.csdn.net/tag/%e7%a4%be%e5%9b%a2%e5%8f%91%e7%8e%b0)

2016-04-09 23:32 595人阅读 [评论](http://blog.csdn.net/DawnRanger/article/details/51108433#comments)(1) [收藏](javascript:void(0);) [举报](http://blog.csdn.net/DawnRanger/article/details/51108433#report)

http://static.blog.csdn.net/images/category_icon.jpg 分类：

社团发现

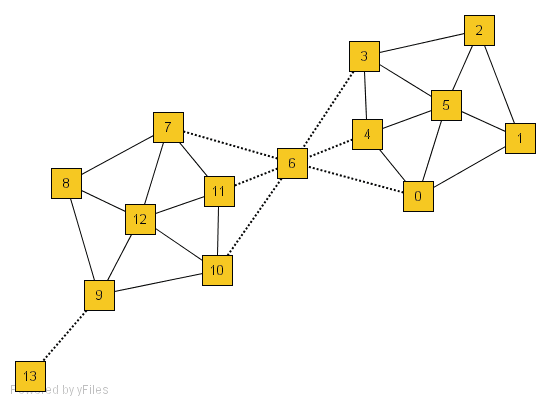
版权声明：本文为博主原创文章，未经博主允许不得转载。

目录[(?)[+]](http://blog.csdn.net/DawnRanger/article/details/51108433)

Paper: 《SCAN: A Structural Clustering Algorithm for Networks》   
Auther: Xiaowei Xu, Nurcan Yuruk, Zhidan Feng, Thomas A. J. Schweiger   
Conference: SIGKDD 2007

**一：SCAN算法简介**

SCAN[**算法**](http://lib.csdn.net/base/datastructure)是由[**机器学习**](http://lib.csdn.net/base/machinelearning)里的基于密度的聚类算法DBSCAN改进而来的一种非重叠社团发现算法，具有线性时间复杂度。其一大亮点在于能发现社团中桥节点(hub)和离群点(outlier)。

主要思想在于，在考虑两点之间的关系的时候，不仅考虑它们的直接链接，而是利用它们的邻居节点来作为聚类的标准。也就是说，节点根据它们共享邻居方式而聚类。   


由图可知，节点0、5共享了4个节点，节点9、13只共享了2个节点，显然它们在聚类是应采取不同的聚类方式。

**二、主要概念介绍**

**1. 节点相似度**

节点相似度定义为**两个节点共同邻居的数目**与**两个节点邻居数目的几何平均数**的比值（这里的邻居均包含节点自身）。 

σ(v,w)=|Γ(v)∩Γ(w)||Γ(v)||Γ(w)|−−−−−−−−−−√

其中 Γ(x) 表示节点 x 及其相邻节点所组成的集合。

**2.**ϵ**- 邻居**

节点的 ϵ - 邻居定义为与其相似度不小于 ϵ 的节点所组成的集合。 

Nϵ={w∈Γ(v)|σ(v,w)≥ϵ}

**3. 核节点**

核节点是指 ϵ 邻居的数目大于 μ 的节点。 

COREϵ,μ(v)⇔|Nϵ(v)|≥μ

**4. 直接可达**

节点 w 是核节点 v 的 ϵ 邻居，那么称从 v 直接可达 w. 

DirREACHϵ,μ(v,w)⇔COREϵ,μ(v)∧w∈Nϵ(v)

**5. 可达**

节点 v 可达 w ，当且仅当存在一个节点链 v1,…,vn∈V,v1=v,vn=w,使得 vi+1 是 从vi 直接可达的。 

REACHϵ,μ(v,w)⇔∃v1,…,vn∈V:v1=v∧vn=w∧∀i∈{1,…,vn}:DirREACHϵ,μ(vi,vi+1)

**6. 相连**

若核节点u可达节点v和节点w，则称节点v和节点w相连. 

CONNECTϵ,μ(v,w)⇔∃u∈V:REACHϵ,μ(u,v)∧REACHϵ,μ(u,w)

**7.相连聚类**

如果一个非空子图C中的所有节点是相连的，并且C是满足可达的最大子图，那么称C是一个相连聚类。 

CLUSTERϵ,μ(C)⇔(1)Connectivity:∀v,w∈C:CONNECTϵ,μ(v,w)(2)Maximality:∀v,w∈V:v∈C∧REACHϵ,μ(v,w)⇒w∈C

**8. 桥节点(hub)**

与至少两个聚类相邻的孤立节点.

**9. 离群点(outlier)**

只与一个聚类相邻或不与任何聚类相邻的孤立节点.

**10. 引理：**

如果 v 是一个核节点，那么从 v 可达的节点集是一个结构相连聚类

**11. 引理：**

C 是一个结构相连聚类， p 是 C 中的一个核节点。那么 C 等于从 p 结构可达的节点集。

**三、具体算法**

1、对于每个未分配社团的节点 v ，检查 v 是否是核节点，是核节点则将其直接可达节点分配到一个社团中（社团标号记为该节点），并将其 ϵ-邻居放进队列中，重复进行1步骤（类似于对直接可达节点进行DFS）。   
2、若 v 不是核节点则将其标志为non-member。   
3、最后检查所有的non-menber节点，若其相邻节点存在于两个及以上的社团中，则将其标为hub节点，否则标为outlier。

ALGORITHM SCAN(G=<V, E>, ε, μ)

// all vertices in V are labeled as unclassified;

for each unclassified vertex v ∈ V do

// STEP 1. check whether v is a core;

if COREε,μ(v) then

// STEP 2.1. if v is a core, a new cluster is expanded;

generate new clusterID;

insert all x ∈ Nε (v) into queue Q;

while Q ≠ 0 do

y = first vertex in Q;

R = {x ∈ V | DirREACHε,μ(y, x)};

for each x ∈ R do

if x is unclassified or non-member then

assign current clusterID to x;

if x is unclassified then

insert x into queue Q;

remove y from Q;

else

// STEP 2.2. if v is not a core, it is labeled as non-member

label v as non-member;

end for.

// STEP 3. further classifies non-members

for each non-member vertex v do

if (∃ x, y ∈ Γ(v) ( x.clusterID ≠ y.clusterID) then

label v as hub

else

label v as outlier;

end for.

end SCAN.

顶