文章编号：1674－9057(2015)01－0181－06 doi:10.3969/j.issn.1674－9057.2015.01.029

基于MCL与KNN的混合聚类算法

牛秦洲，陈 艳

（桂林理工大学 信息科学与工程学院，广西 桂林 541004）

摘 要：MCL是一种图聚类算法，针对MCL计算过程会产生小聚类及边缘节点从团中脱离出来的问题，提出了一种基于MCL与KNN相结合的混合聚类算法。该算法利用KNN的分类特点，以MCL聚类得到的聚类表为依据，通过KNN对小聚类中的元素进行再分类，以提高聚类的质量。实验证明此方法是可行的，改进后的算法能使聚类质量有所提高。

关键词：MCL；聚类；KNN；小聚类；再分类

中图分类号：TP301.6 **文献标志码**：A

聚类分析是数据挖掘领域中一种重要的数据处理方法，主要是对数据按某种相似性度量标准来进行聚类。

聚类操作中比较常用的几种算法有k-means聚类[1－2]、基于密度[3]的聚类和基于网格聚类[4]等，但这些算法都存在以下几个方面的缺陷：1）它们都是以经典的相似性度量作为聚类的基础，聚类而成的概念不能完全反映该类概念抽象的实质，进而导致求解精度下降；2）当样本属性较多时，往往很难通过简单计算得到的相似度值来全面地反映样本之间的相关或相似程度，从而导致样本聚类概念的区分度较低。

与传统的聚类算法相比，MCL（Markov clustering，马尔可夫聚类）能够很好地避免以上问题的发生。MCL聚类是图聚类的一种，其核心思想是基于模拟随机流（使用图的转移概念矩阵）进行图聚类[5]。这种方法已被应用到很多领域并取得显著成功（如蛋白质的Tribe-MCL聚类）[6－7]。在MCL聚类的过程中，不是简单地求出样本之间的关系来进行聚类判断，而是在这种关系的基础上通过膨胀和扩展操作来获得关于样本之间的更深层的知识来进行聚类。因此在聚类的过程中从全局的角度出发，对样本间的关系分析的更全面、更彻底。但由于在MCL聚类过程中过度拟合使得容易产生小聚类（只有一两个节点的聚类）从而影响聚类质量，为了解决这个问题，笔者采用MCL结合KNN[8]的混合算法，对MCL进行改进以提高聚类质量。

1 MCL聚类算法介绍

1.1 MCL聚类算法概述

马尔可夫聚类算法（MCL）是van Dongen[5]开发的用于图形聚类的算法，已成功应用于蛋白质家族分析等领域。MCL算法是基于随机流的一种方法，不需预先设定聚类数目，通过概率改变和反复修改矩阵以实现随机流模拟。

MCL聚类过程主要由在随机矩阵（马尔可夫矩阵）上的两个操作构成：膨胀操作和扩展操作。膨胀操作主要是对矩阵进行的幂操作；扩展操作是对随机矩阵中每个元素进行的幂操作，即$M_{ij}=(M_{ij})^{r}$。

1.2 相关定义及操作介绍

图的相关矩阵：设G为一个有n个顶点的带有权值的图，则G的相关矩阵被定义为$\boldsymbol{M}'(G)$，其中$\boldsymbol{M}'(G)$中的元素M_{ij}'的值与图G中相应边上的权值W_{ij}相等。

收稿日期：2014－03－24

基金项目：国家自然科学基金项目（51365010）；广西科学研究与技术开发计划项目（桂科合1346011－10）

作者简介：牛秦洲（1956—），男，教授，研究方向：制造业信息化、人工智能。

引文格式：牛秦洲，陈艳. 基于MCL与KNN的混合聚类算法［J］. 桂林理工大学学报，2015，35（1）：181－186.

图的马尔可夫矩阵：令 $\boldsymbol{M}'$ 是图 G 的相关矩阵，$\boldsymbol{M}$ 为图的马尔可夫矩阵，则 $\boldsymbol{M}$ 的第 j 列是 $\boldsymbol{M}'$ 的第 j 列的归一化，即 $M_{lj} = M_{lj}' / \sum_{i} M_{ij}'$。其中：$l$，$j$ 为常数。

令图 G 的相关矩阵表示为

$$\boldsymbol{M}'(G) = \begin{pmatrix} 2 & 2 & 1 \\ 3 & 6 & 3 \\ 1 & 4 & 1 \end{pmatrix},$$

则图 G 的马尔可夫矩阵可表示为

$$\boldsymbol{M} = \begin{pmatrix} 1/3 & 1/6 & 1/5 \\ 1/2 & 1/2 & 3/5 \\ 1/6 & 1/3 & 1/5 \end{pmatrix}。$$

扩展操作：设图 G 为一个有 n 个顶点的图，图的马尔可夫矩阵可表示为 $\boldsymbol{M}(G)$。扩展操作的作用是通过对图的马尔可夫矩阵进行幂乘 $\boldsymbol{M} = \boldsymbol{M} \cdot \boldsymbol{M}$，使两个顶点间的相似度转化为两个顶点与共同邻接点的相似度的乘积之和。当扩展值 $e = 2$ 时，可表示为 $\boldsymbol{M} = \boldsymbol{M} \cdot \boldsymbol{M}$，$M_{ij}$ 可表示顶点 i 经过一步到达顶点 j 的概率；当 $e = 3$ 时，$\boldsymbol{M} = \boldsymbol{M}^3$，此时 M_{ij} 表示顶点 i 经过两步流向顶点 j 的概率；当经历的步数越多，则流出所在簇而进入另外一个簇的可能性变大。所以扩展操作数 e 越大，会使同一个簇之间的紧密性作用优势减弱。

膨胀操作：膨胀操作通过膨胀参数 R 来对矩阵 $\boldsymbol{M}$ 中的每一列元素进行幂乘操作，然后对每一列进行归一化操作，这样可以拉大同一列元素之间的差距。膨胀操作是一个非线性操作，目的是增强类内的流，削弱类间的流。

$$\begin{pmatrix} 1 & 2 & 1 \\ 3 & 5 & 2 \\ 1 & 4 & 3 \end{pmatrix} \xrightarrow{R=2} \begin{pmatrix} 1 & 4 & 1 \\ 9 & 25 & 4 \\ 1 & 16 & 9 \end{pmatrix} \xrightarrow{\text{normalization}}$$

$$\begin{pmatrix} 1/11 & 4/45 & 1/14 \\ 9/11 & 25/45 & 4/14 \\ 1/11 & 16/45 & 9/14 \end{pmatrix}。$$

MCL 的聚类过程如图 1 所示。可以看出，MCL 的聚类过程主要通过膨胀和扩展两个操作在马尔可夫矩阵上的交替使用来完成矩阵的修改，从而使马尔可夫矩阵达到稳定状态实现聚类。

由于上述聚类特点，经过一定步骤的操作后导致节点间关系的放大和缩小，从而使节点之间的关系出现偏差。由于偏差不断扩大，致使一些边缘节点从团中分离出来形成单个或只有两三个节点的小聚类。

MCL算法：

输入：G,M(G),膨胀参数 R

//G为一个图

M(G)是图G的关系矩阵

1.令M=M(G)//初始化M

2.while M没有达到稳定状态, do

3. 对M进行扩展操作: M:=expand(M)

4. 对M进行膨胀操作: M=inflate:(M,R)

5. 对M进行归一化操作: $M_{ij}:=M_{ij}/\sum_{i=1}^{n} M_{ij}$

6.输出:聚类信息

图 1 MCL 聚类算法

Fig. 1 MCL clustering algorithm

为了减少小聚类的产生，提高聚类的质量，笔者提出了一种改进的 MCL 算法。考虑到在聚类过程中由于不断的膨胀和扩展操作导致节点关系的偏差放大，图中处在同一个团中的边缘节点容易被分离出来。然而即使被分离出来，边缘形成的小聚类与原来一个团体形成的聚类的联系依然是很紧密的。因此，提出了一种 MCL 与 KNN 相结合的混合聚类算法，算法的主要思想如下：在进行聚类后，对小聚类与其他聚类的类间距离关系进行判断，从而决定是否需要进一步合并。

当小聚类与其他聚类质心的最小距离小于某一阈值，可认为它们之间关系比较紧密，小聚类的存在不合理，需要进一步利用 KNN 对小聚类中的元素进行重新分配；当质心的距离大于阈值时，则认为小聚类与其他类存在明显的区别，其存在是合理的。

2 混合聚类算法

2.1 算法思想

采用 MCL 与 KNN 相结合的混合算法来避免上述问题的发生。KNN 算法是一种分类算法，其基本思想是：给定一个待分类的数据，通过以一个已分类的样本空间为依据，计算离自己最近的 K 个样本来判断这个待分类数据的类别情况。因此，可以考虑利用 KNN 来对分离出来的小聚类进行再次的分类，从而解决小聚类产生的问题。

2.2 马尔可夫矩阵计算

在进行 MCL 聚类前，需要先求出数据之间的关系，得到一个关系矩阵。

定义相似度函数

$$\mathrm{sim}_G(X,Y) = \sum_{i=1}^{k} w_i \cdot \left(1 - \frac{d_n(X_i,Y_i)}{\mathrm{range}(D_i)}\right); \quad (1)$$

$$\mathrm{range}(D_i) = \max(D_i) - \min(D_i)。 \quad (2)$$

式中：$\mathrm{sim}_G(X,Y)$ 表示实例 X,Y 之间的相似度；k 表示样本的属性个数；$d_n(X_i,Y_i)$ 为两个实例的属性 i 的距离；w_i 为属性 i 的权值；D_i 表示属性 i 的取值范围；$\mathrm{range}(D_i)$ 为 D_i 中的最大差值。利用上面的相似度计算模型来计算样本之间的关系，得到关系矩阵即马尔可夫矩阵。

2.3　质心判断

在对聚类进行质量评价时，类间距离常会被作为一个重要因素考虑。采用同样的思想，对于产生的小聚类，利用类间的距离来衡量两个类是否聚类合理。

由于同一个聚类中的样本相似程度高，因此可以利用质心来简化整个类。

质心公式

$$\overline{C_i} = \{\bar{x}_{i1},\bar{x}_{i2},\bar{x}_{i3},\cdots,\bar{x}_{ik}\}; \quad (3)$$

$$\bar{x}_{ij} = \frac{1}{N_i}\sum_{j=1}^{N_i} x_{ij}。 \quad (4)$$

式中：$\overline{C_i}$ 表示类 i 的质心；由 $\bar{x}_{i1}$，$\bar{x}_{i2}$，$\bar{x}_{i3}$，…，$\bar{x}_{ik}$ 即 k 个属性组成，其中 $\bar{x}_{ij}$ 表示类 i 中属性 j 的平均值；N_i 表示类 i 中实例的总数。

质心代表的是聚类中各实例的平均化，通过质心间的距离从整体上判断聚类之间分布的合理性，这里引入阈值 T_d 进行判断，若两个质心的距离小于 T_d，则认为它们属于同一个聚类，利用 KNN 对小聚类中的元素进行精确再分配；若两个质心的距离大于 T_d，则认为它们不属于同一个聚类，不需要对小聚类作任何处理。

2.4　改进算法描述

通过将 MCL 算法结合 KNN 算法的思想控制小聚类的生成，首先对 MCL 产生的聚类进行检测，若产生小聚类，判断是否需要对小聚类进行处理，若需要，则对小聚类中的对象采用 KNN 算法，利用原始的马尔可夫关系矩阵，找出小聚类中每个对象的 K 个最相似的对象，并计算所属类的权值，通过 KNN 将小聚类中的对象合并到权值最大的一类中去，这样可以有效地控制小聚类的形成，满足用户的需求，同时也提高了聚类的质量。K 的取值涉及到再分类的精确性：当 K 取值过大，考虑的周围点多，影响因素多，分类的精度在一定程度上降低；K 取值过小，则由于考虑的周围点太少，使得分类过于片面，如何选择一个合适的 K 值，需要根据具体情况而定，如聚类数量、规模、分类精度要求等。

定义 K 的取值

$$K_i = v_{s_i} + m。 \quad (5)$$

其中：K_i 表示小聚类 i 对应的 K 值；v_{s_i} 表示小聚类 i 的元素个数；m 是一个常数，主要用于调节分类精度，$m = \max(v_{s_i},N_\mathrm{c})$；$N_\mathrm{c}$ 为 MCL 聚类后得到的聚类数，m 取 v_{s_i} 与 N_c 之间的最大值，这样的取值，既避免了小聚类本身元素的干扰，又充分考虑到了训练样本的分类信息，使待分类元素能够在一个合适的 K 值下更加合理、可靠地进行再分类。

算法具体过程如下。

需要建立一张辅助表 Tb_ad，用于存放经过验证可以单独存在的小聚类。

输入 R 值、距离阈值 T_d。

① 对所有训练样本求关系矩阵

$$G_{ij} = \mathrm{sim}_G(X_i,X_j)。 \quad (6)$$

其中：$\boldsymbol{G}$ 为关系矩阵，G_{ij} 是关系矩阵的第 i 行第 j 列的值，表示第 i 个样本与第 j 个样本的关系值。

② 对数据进行 MCL 聚类，聚类结束，形成一张聚类信息表 Tb_inf。

③ 查看 Tb_inf 中是否有小聚类产生，若 Tb_inf 中有小聚类，则查看 Tb_ad 中是否有此聚类的信息：若 Tb_ad 中有此聚类的信息，则在 Tb_inf 中继续往下寻找；若 Tb_ad 中没有此聚类的信息，则进入 ④；若 Tb_inf 中没有小聚类存在，则输出。

④ 计算各聚类的质心，利用质心关系来判断小聚类的分布是否合理，即小聚类的质心与其他聚类的质心距离是否小于给定值 T_d：若小于 T_d 则说明两个聚类很相似，小聚类产生不合理，需要进行再分配，进入 ⑤；若大于 T_d 则说明小聚类与其他聚类区别比较明显，可以单独存在，将小聚类的信息写入 Tb_ad 表中，返回 ③。

⑤ 在基础关系表中找出与小聚类元素最接近的 k 个元素，以 MCL 聚类信息表 Tb_inf 为基础，计算 k 个元素的类别信息，从而确定小聚类元素的最终类别信息，并更新 Tb_inf 表，返回 ③。

计算公式如下：

$$W(x,C_i) = \sum_{j=1}^{K} B(x_j,C_i); \quad (7)$$

$$B(x_j,C_i) = \begin{cases} 1, & x_j \in C_i; \\ 0, & x_j \notin C_i。 \end{cases} \quad (8)$$

$$C(x) = \{C_i \mid \max(W(x,C_i))\}。 \quad (9)$$

其中：$W(x,C_i)$ 表示 x 属于类别 i 的权值；x_j 为 x 的 K 个最相似样本的第 j 个样本；$C(x)$ 表示待分类样本的类别信息，待分类样本的类别为 $W(x,C_i)$ 最大时的类别。

输出：聚类得到的各个簇信息。

其中，Tb_{ad} 结构如下：

```
typedef struct Tb_ad
  {
  int id;//序列号
  int cluster-id;//聚类号
  string inf;//描述信息
  } Tb_ad
```

Tb_{inf} 结构如下：

```
typedef struct Tb_inf
  {
  int id;//序列号
  int cluster-id;//聚类号
  int cluster-num;//聚类中的实例数
} Tb_inf;
```

3 算法检验

3.1 检验标准

3.1.1 模块度指标 模块度[9]指标是由 Newman 等提出来的一种聚类评价标准，也是最广泛被使用的度量聚类质量好坏的指标之一。

公式定义如下：

$$Q = \sum_i (e_{ii} - a_i^2) = \mathrm{tr}(\boldsymbol{E}) - \| \boldsymbol{e}^2 \|。 \quad (10)$$

其中：$\boldsymbol{E}$ 为一个 $k \cdot k$ 的矩阵；e_{ij} 表示第 i 行第 j 列的一个值，也表示聚类 i 与聚类 j 的所有边占整个图中所有边的比例；$\mathrm{tr}(\boldsymbol{E})$ 为 $\sum_{i=1}^{k} e_{ii}$，表示矩阵的迹；a_i 为 $\sum_{j=1}^{k} e_{ij}$，即矩阵 $\boldsymbol{E}$ 中的第 i 行的所有值相加，表示聚类 i 与其他聚类相连的边之和占整个图中的边的比例。

3.1.2 DB Index 指标 DB Index[10]准则作为一种评判准则，同时使用了类间距离和类内离散度，基本内容如下。

类内平均离散度

$$S_i = \frac{1}{|C_i|} \sum_{X \in C_i} \| X - Z_i \|, \quad (11)$$

其中：Z_i 是 C_i 类的类中心；$|C_i|$ 表示 C_i 类样本数。

类间距离

$$d_{ij} = \| Z_i - Z_j \|, \quad (12)$$

即用两个类中心的距离表示类间距离。

DB Index 定义为

$$DB_k = \frac{1}{k} \sum_{i=1}^{k} R_i, \quad (13)$$

其中：$R_i = \max_{j=1,\cdots,k,j\neq i} \frac{S_i + S_j}{d_{ij}}$，$k$ 是分类数目。DB Index 准则是 DB 的值越小，说明分类的效果越好。

3.2 实验分析

通过采用 UCI 机器学习数据库[11]中的 Seeds 数据集和 Iris 数据集对本文提出的算法与 MCL 算法、K 均值算法、K 调和均值（KHM）算法和模糊 C 均值算法分别进行聚类质量的比较。Seeds 数据集和 Iris 数据集都是用于检验分类算法的标准数据，因此选用这些标准数据，对算法的验证具有一定的说服力。

首先以模块度指标对 MCL 聚类与本文提出的混合聚类算法进行比较。在 Matlab 下进行仿真，结果如图 2 所示。表 1 给出了仿真得到的具体数据。

当膨胀参数 R 值为 4 ~ 6 时，MCL 和混合算法聚类效果是一样的，这是由于 R 的取值比较小没有产生小聚类所致。随着 R 值的增大，MCL 算法产生的小聚类个数增多，由于这些小聚类与其他聚类间的联系比较密切，因此使类之间的区分度

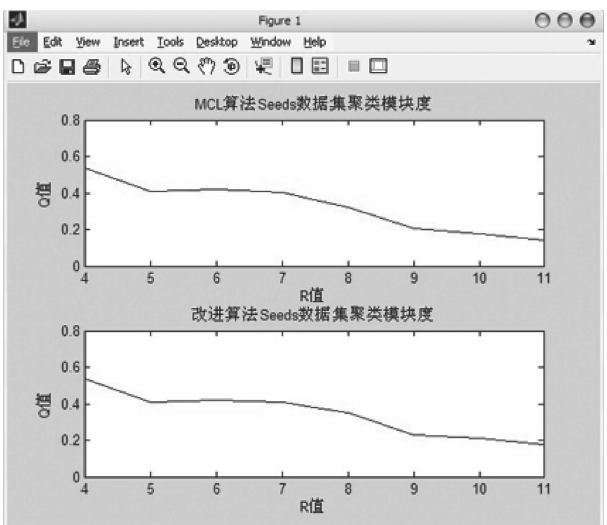


图 2 Seeds 数据集聚类的模块度仿真比较

Fig. 2 Module simulations comparison of Seeds data aggregation

表 1　Seeds 数据集聚类的模块度仿真数据

Table 1　Module comparison of Seeds data aggregation

膨胀参数 R 值	MCL	本文算法
4	0. 534	0. 534
5	0. 408	0. 408
6	0. 420	0. 420
7	0. 401	0. 410
8	0. 320	0. 352
9	0. 205	0. 231
10	0. 174	0. 210
11	0. 143	0. 175

比改进后算法的聚类区分度低，由表 1 可以看出，在相同 R 值下改进后的算法能得到质量更好的聚类。

R 值的取值：当 R 取值太小，聚类粒度过大，整个数据集都被聚为一类，起不到区分的作用；当 R 取值过大时，聚类粒度过细，可能导致整个数据集中的每个实例被单独作为一类。膨胀参数 R 值的选取标准，主要根据聚类评价标准及用户对数据的聚类规模的要求来选择 R 值。

时间复杂度：由于改进算法是在 MCL 基础上结合 KNN 的算法，因此，改进算法在时间复杂度上比 MCL 时间复杂度多出了 KNN 部分，即 $O(N)$。其中 N 为聚类数据的规模，当 N 的数量只有几百个案例时，改进算法与原算法在时间消耗上没有太大差别；随着 N 的增大，改进算法与原算法在时间消耗上差距逐渐拉大。

利用 DB Index 准则对 K 均值聚类、K 调和均值（KHM）算法、模糊 C 均值聚类算法和本文提出的 MCL 混合聚类算法进行聚类质量比较，DB Index 准则利用对聚类后各聚类之间的距离和聚类内部的离散度来评定聚类的质量，在 Matlab 下的仿真结果如图 3 所示，表 2 为仿真得到的具体数据。

从表 2 可以看出，当聚类数为 2、3 时，本文算法与 FCM 算法比较接近，可能是由于聚类数过少所致。随着聚类数的加大，FCM 算法与本文算法得到的聚类质量差距开始变大；随着聚类个数的增加，K 均值算法、KHM 算法、FCM 算法和本文算法的 DB Index 值都呈上升趋势，即聚类质量都在下降，但其他算法与本文算法相比聚类质量下降速度快，聚类质量比本文算法的聚类质量差。这说明了本文提出的改进算法效果更好，可以得到质量更好的聚类结果。

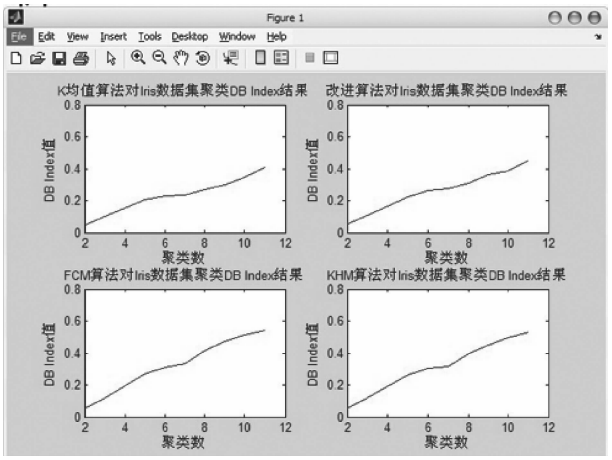


图 3　4 种聚类算法对 Iris 数据集的 DB Index 指标仿真

Fig. 3　Simulations of DB Index of Iris data aggregation with 4 clustering algorithms

表 2　Iris 数据集聚类的 DB Index 指标仿真结果

Table 2　DB Index of Iris data aggregation

聚类数	K 均值算法	本文算法	FCM 算法	KHM 算法
2	0. 055	0. 049	0. 055	0. 055
3	0. 118	0. 100	0. 105	0. 120
5	0. 266	0. 205	0. 222	0. 269
6	0. 304	0. 229	0. 261	0. 312
7	0. 314	0. 237	0. 273	0. 335
8	0. 400	0. 268	0. 310	0. 413
9	0. 450	0. 300	0. 361	0. 470
10	0. 493	0. 346	0. 385	0. 512
11	0. 530	0. 410	0. 447	0. 541

从以上实验看，基于 MCL 与 KNN 相结合的混合聚类算法，能够增强聚类的效果，提高类与类之间的区分度。与几种经典的聚类算法进行实验比较，通过对数据的聚类粒度的不断调整分析几种聚类算法在此过程中的聚类性能。对于同一数据样本集，随着聚类粒度的不断细化，聚类得到的数目变大，由于基于 MCL 与 KNN 的聚类算法，结合了 MCL 的全局判断性，即在进行聚类时，是在对全局数据之间的距离关系进行分析的基础上获得的知识为依据进行聚类的，避免了聚类过程中只考虑部分数据的距离关系而导致的片面性，通过膨胀因子拉大差距，从而使实例之间的区分度得到提高，另外通过结合 KNN 的分类特性，克服了因膨胀因子而导致的过度分离现象（小聚类），在整个聚类过程中，基于 MCL 的聚类算法既保证了聚类结果的区分度，又防止了过度

分离的现象，使聚类结果在保持实例的区分度和相似度上得到了较好的平衡。与其他算法在整个实验中进行比较，得到了较好的聚类结果，具有一定的典型性。

4 结束语

本文针对 MCL 自身存在的不足，考虑到 KNN 算法的分类特性，提出了一种 MCL 与 KNN 相结合的混合聚类算法。该混合聚类算法结合了 MCL 聚类和 KNN 算法自身的特点，以 MCL 聚类信息为指导，利用 KNN 对小聚类进行再分配，从而优化聚类结果，混合聚类算法很好地弥补了 MCL 聚类过程中容易产生小聚类的不足。与 MCL 算法相比，在产生小聚类时，混合算法能很好地将小聚类进行再分配，从而提高了聚类的质量。利用传统聚类算法与改进算法进行比较实验，结果证明此方法是有效的、可行的，具有较好的实际应用价值。由于本算法是在 MCL 基础上结合了 KNN 分类算法，改进算法的时间复杂度比 MCL 的时间复杂度多出 $O(N)$，即 KNN 的时间复杂度。随着数据规模的增大或者小聚类数量的增多，改进算法所消耗的时间将逐渐增大。此算法比较适合数量中等的数据进行聚类，既能保持比 MCL 算法更好的聚类性能，又能避免消耗过多的时间。

参考文献：

[1] 乔丽，姜慧霖，贾世杰. 基于改进 *k*-means 聚类的案例检索策略 [J]. 计算机工程，2011，37 (5)：193 – 195.

[2] 乔丽，姜慧霖. 一种 *k*-means 聚类的案例检索算法 [J]. 计算机工程与应用，2011，47 (4)：185 – 187.

[3] Chen Y X, Tu L. Density-based clustering for real-time stream data[C]//Proceedings of the 13th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, ACM, 2007：133 – 142.

[4] 赵慧，刘希玉，崔海青. 网格聚类算法 [J]. 计算机技术与发展，2010，20 (9)：83 – 85.

[5] van Dongen S. Performance criteria for graph clustering and Markov cluster experiments [R]. Amsterdam：National Research Institute for Mathematics and Computer Science in the Netherlands, 2000.

[6] 温菊屏,钟勇. 图聚类的算法及其在社会关系网络中的应用[J]. 计算机应用与软件,2012,29(2):161 – 163.

[7] Akama H, Miyake M, Jung J. How to take advantage of the limitations with Markov clustering—the foundations of branching Markov clustering (BMCL) [C] //The Third International Joint Conference on Natural Language Processing. Hyderabad：the Asian Federation of Natural Language Processing, 2008：901 – 906.

[8] 孙荣宗,苗夺谦,卫志华,等. 基于粗糙集的快速 KNN 文本分类算法[J]. 计算机工程,2010,36(24):175 – 177.

[9] Newman M E J, Girvan M. Finding and evaluating community structure in networks [J]. Physical Review E, 2004, 69 (2)：026113.

[10] 罗兰，曾斌. 基于时序向量聚类的周期关联规则发现算法 [J]. 计算机工程，2010，36 (19)：110 – 112.

[11] Frank A, Asunclon A. UCI machine learning repository [EB/OL]. [2011 – 11 – 02]. http：//archive. ics. uci. edu/ml.

Hybrid clustering algorithm based on KNN and MCL

NIU Qin-zhou, CHEN Yan

(College of Information Science and Engineering, Guilin University of Technology, Guilin 541004, China)

Abstract: MCL is a graph clustering algorithm. With the characteristics of MCL computational process, MCL is prone to producing small clustering and separating edge nodes from the groups. A hybrid clustering based on MCL combined with KNN algorithm is proposed. Hybrid algorithm improves the quality of clustering by reclassification of elements in small clustering by KNN classification characteristics and clustering tables required by MCL clustering. Experiments prove that this method is feasible and the improved algorithm can enhance the quality of clustering.

Key words: MCL; clustering; KNN; small clustering; reclassification