基于混合测度的并行仿射传播聚类算法

张建朋 陈福才 李邵梅 于洪涛

(国家数字交换系统工程技术研究中心 郑州 450002)

摘 要 针对仿射传播聚类(AP)算法应用于流形结构复杂、密度不均匀的数据集存在的不足,通过学习数据集的低维流形结构,提出了密度自适应的"流形距离核"(ad-MDK)的概念。该距离测度既考虑了数据点的局部密度信息,又包含了数据集全局结构信息,从而提高了算法对这类数据集的处理能力。同时,针对引入流形距离所带来的计算复杂问题,提出了算法的并行化设计方法,有效提高了算法处理效率。通过在多个数据集上的实验验证了所提算法在处理大规模多尺度数据集上的性能优于传统AP算法。

关键词 仿射传播聚类,流形距离核,共享最近邻,并行计算

中图法分类号 TP181 **文献标识码** A

Parallel Affinity Propagation Clustering Algorithm Based on Hybrid Measure

ZHANG Jian-peng CHEN Fu-cai LI Shao-mei YU Hong-tao

(China National Digital Switching System Engineering & Technological R&D Center, Zhengzhou 450002, China)

Abstract Affinity propagation clustering (AP) algorithm has a difficult to get ideal clustering results in a complex manifold structure and non-uniform density datasets. Through studying the low-dimensional manifold structure, this paper drawed out the density auto-adapted "manifold distance kernel" concept(ad-MDK), which takes into account the local density information of data points, also contains the overall structure of the data set globally, making the algorithm can sovle complex distributed data clustering problem. Meanwhile, in order to reduce the manifold distance calculated, parallel algorithm for the proposed algorithm was introduced to the affinity propagation clustering to effectively improve the speed of the algorithm. Experiments on several data sets verify that the proposed algorithm is superior to the traditional AP algorithm performance in dealing with large-scale multi-scale data set.

Keywords Affinity propagation, Manifold distance kernel, Shared nearest neighbor, Parallel computation

仿射传播(Affinity Propagation,AP)算法[1]自2007年提出以来,因其能在较短的时间内对大规模数据集进行处理得到较理想的结果而成为聚类分析领域的研究热点。与多数聚类算法(如K-means[2],谱聚类[3]等)不同,AP算法将每个数据点都作为候选的类代表点,避免了聚类结果受限于初始类代表点选择的影响,同时该算法对于数据集生成的相似度矩阵的对称性没有要求,并且在处理大规模多类数据时运算速度快,所以能够很好地解决非欧空间问题以及大规模稀疏矩阵计算问题等。但是,由于仿射传播算法只能处理紧凑的具有超球形分布的数据集合,对于一些本身具有复杂结构的数据集难以得到合理的聚类结果,因此AP算法对这些数据集处理的改进成为当前研究的热点。

为了提高AP算法的聚类精度,KJ. Wang等人[4]提出了自适应仿射传播聚类算法,讨论了如何选择迭代和阻尼系数,使算法在任何情况下都能自适应收敛,但其应用只局限于超球面的紧凑型数据。肖宇等人[5]提出的半监督的仿射传播聚类算法,利用成对约束信息调整相似度矩阵,一定程度改进了聚类性能,但它没有考虑无标记样本数据所隐含的背景信息,当约束信息较少或者包含噪声时,反而可能误导聚类过程。董俊等人[6]提出了可变相似性度量的仿射传播聚类算法,其在紧致和部分重叠的数据集上均得到了较好的聚类结果,但在非均匀密度或结构分散的数据集上效果并不令人满意。

本文通过分析数据的分布特性,提出了一种基于混合测度的仿射聚类算法,其通过同时描述数据点的局部邻域信息和数据集全局结构信息,提高了对复杂数据集的处理能力。同时,针对引入流形距离所带来的计算复杂问题,提出了算法的并行化设计方法,有效提高了算法的处理效率。实验结果表明,本文所提算法的性能与原始AP聚类算法相比有明显的提高。

本文第1节简要叙述AP聚类算法的思想;第2节设计出一种结合数据点局部信息和整体信息的流形距离核测度,并基于此提出基于混合测度的仿射传播聚类算法(Affinity Propagation Clustering based on Kernel Manifold Distance, APMDK);第3节在APMDK算法的基础上,对其进行并行

到稿日期:2012-09-21 返修日期:2012-12-13 本文受国家高技术研究发展计划("863"计划),国家重点基础研究发展计划("973"计划)基金资助项目(2012CB315901,2012CB315906)资助。

张建朋(1988—),男,硕士生,主要研究领域为电信网关防及数据挖掘,Email:feilong0309@sina.com;陈福才(1974—),男,研究员,硕士生导师,主要研究领域为电信网关防;李邵梅(1982—),女,博士,讲师,主要研究领域为数据挖掘;于洪涛(1970—),男,教授,硕士生导师,主要研究领域为电信网关防。

化改进，提出了并行 APMDK 聚类方法(A Parallel Affinity Propagation Clustering based on Kernel Manifold Distance，P-APMDK)；第 4 节是算法的有效性论证和实验；最后对全文进行小结。

1 基本的仿射传播算法

AP 算法是一种基于近邻信息传播的聚类算法，该算法的目的是找到最优的类代表点集合，使得所有数据点到最近的类代表点的相似度之和最大。算法的输入是所有 N 个数据点两两之间的相似度组成的相似性矩阵 $S_{N\times N}$。起始阶段假设所有的样本被选中成为类代表点的可能性相同，即设定所有 $s(i,i)$ 为相同值 p，p 值越大，则最终输出的聚类数目越大。同时，计算每个样本点与其他样本点的吸引程度，吸引程度主要由下面两个指标来描述。

吸引度(Responsibility)：$r(i,k)$ 用来描述点 k 作为数据点 i 的聚类中心的适合程度。

归属度(Availability)：$a(i,k)$ 用来描述点 i 选择点 k 作为其聚类中心的适合程度。

为了找到合适的聚类中心 x_k，AP 算法不断地从数据样本中搜集证据 $r(i,k)$ 和 $a(i,k)$。$r(i,k)$ 和 $a(i,k)$ 的迭代公式为：

$$r(i,k)\leftarrow s(i,k)-\max_{k' \text{ s.t. } k'\neq k}[a(i,k')+s(i,k')] \tag{1}$$

$$a(i,k)\leftarrow \begin{cases}\min\{0,r(k,k)+\sum_{i' \text{ s.t. } i'\notin\{i,k\}}\max[0,r(i',k)]\}, & i\neq k\\ \sum_{i' \text{ s.t. } i'\neq k}\max[0,r(i',k)], & i=k\end{cases} \tag{2}$$

AP 算法中主要根据式(1)和式(2)不断循环迭代从而更新证据，迭代更新的快慢可以通过调节阻尼系数 λ 实现。$r(i,k)$ 与 $a(i,k)$ 越大，则 k 点作为聚类中心的可能性就越大，并且 i 点隶属于以 k 点为聚类中心的聚类的可能性也越大。AP 算法通过迭代过程不断更新每一个点的吸引度和归属度值，直到产生 m 个高质量类代表点(exemplar)，同时将其余的数据点分配到相应的类簇中。

2 基于混合测度的仿射传播聚类算法

Zhou 等人[7]同时考虑数据集的局部和整体特性，认为对象相似必须满足如下的“一致性”条件：

局部一致性——如果两个点在空间位置上相邻，它们很可能来源于同一类；

全局一致性——处在同一流形上的数据点属于同一类的可能性比较大。

通常数据的分布具有不可预期的复杂结构，AP 算法在处理超球形紧密分布的数据集时表现优异，但其由于只关注数据的局部一致性，难以适用于具有任意形状和多重尺度的数据集。如图 1 所示，在只能表征数据点的局部一致性的欧式距离中 $dist(a,b)=dist(a,c)$，根本无法反映图中数据的全局一致性(即位于同一流形上的数据点具有较高的相似性)，严重影响了聚类算法的性能。鉴于此，本文综合考虑数据集的全局结构和局部信息，提出了以具有全局一致性的流形距离核为基础、核参数根据簇局部密度变化而自适应调整的“流形距离核测度”的计算方法，并将其作为数据间的相似性度量准则，从而改善其聚类性能，使其能更好地处理复杂结构的聚类问题。下面首先给出本文改进的流形距离核全局测度的计算方法。

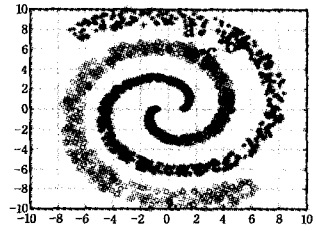


图 1 螺旋臂示意图

2.1 基于流形距离核的全局测度

定义 1(流形上的线段长度) 空间两点 x_i 与 x_j 之间流形上的线段长度 $d_X(x_i,x_j)$ 按下式计算：

$$d_X(x_i,x_j)=\rho^{dist(x_i,x_j)}-1$$

式中，$dist(x_i,x_j)$ 为 x_i 与 x_j 之间的欧氏距离；$\rho>1$ 为伸缩因子，调节 ρ 可放大或缩短两点间线段长度。

根据流形上的线段长度，将数据点看作是一个无向加权图 $G=(V,E)$ 的顶点 V，边集合 $E=\{W_{ij}\}$ 表示数据点间连接权值，则流形距离核测度计算步骤如下：

步骤 1 根据数据集 $X=\{x_1,x_2,\cdots,x_N\}$ 构建无向加权图 $G=(V,E)$，若点 x_j 是 x_i 的 k-近邻点 $x_j\in V_k(x_i)$，则用流行上的线段长度 $d_X(x_i,x_j)$ 连接两点，否者断开连接。

步骤 2 利用 Floyd's 算法计算邻接图 $G=(V,E)$ 两点间的最短路径 $d_G(x_i,x_j)$，使其近似等于流形的测地线距离，令 P_{ij} 表示图 G 上连接顶点 x_i 和 x_j 的所有路径的集合，则

$$d_G(x_i,x_j)=\min_{p\in P_{ij}}\sum_{k=1}^{|p|-1}d_X(p_k,p_{k+1}) \tag{3}$$

步骤 3 计算邻接图 G 中任意两顶点间的流形距离核测度 d_{MD}。

$$d_{MD}(i,j)=1-\exp(-\frac{d_G(x_i,x_j)^2}{\sigma^2}) \tag{4}$$

则，点 x_i、x_j 的流形相似度为：

$$S_{MD}(i,j)=\begin{cases}\exp(-\dfrac{\min\limits_{p\in P_{ij}}\sum\limits_{k=1}^{|p|-1}d_X(p_k,p_{k+1})^2}{\sigma^2})-1, & x_j\notin V_k(x_i)\\ \exp(-\dfrac{d_X(x_i,x_j)^2}{\sigma^2})-1, & x_j\in V_k(x_i)\end{cases} \tag{5}$$

式中，σ 为高斯核的半径。可以看出，该距离测度满足自反性、对称性、三角不等式的约束条件。

该距离测度可以度量流形的最短路径，能够很好地反映数据集内在的流形结构，这使得位于同一流形上的两点可以用许多较短的边相连，位于不同流形上的点要用较长的边相连接，而流行距离的核映射进一步放大位于不同流形上的数据点间距离，缩短位于同一流形上的数据点间距离。

2.2 基于局部密度信息的核参数调整

在流形距离核的全局测度中，如何建立满足学习目标的核函数是影响算法效果的关键。高斯核的分布参数值 σ 会极大地影响核函数的泛化性能。σ 值过大或过小均会导致泛化能力的降低。怎样选择合适的 σ 值使得数据在核空间中线性可分或者近似线性可分，是本小节讨论的重点。

流形距离测度虽然能够反映数据的流形结构特征，但采用的相似度函数仍是基于核空间的欧氏距离度量方法，其处理分布密度不均匀的数据时仍然存在困难。如图 2 所示，数据集具有两个密度不同的簇，如果相似度参数选择得足够小，则容易将两个簇并入同一类；反之，如果相似度参数足够高，

则不能发现低密度簇。因此基于欧氏距离的相似度度量在此类问题上不能产生理想的聚类结果。

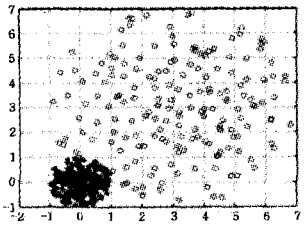


图 2　密度不均匀的数据集

为了解决这种多重尺度的数据集，Manor 等[8]提出将数据点的邻域信息加入相似度计算中，给出了一种自动选择尺度参数(Local Scaling)的方法，其并不是为整个样本集选择一个尺度参数 σ，而是为每一个样本点 x_i 选择一个 σ_i，称之为自调节的高斯核函数，定义如下：

$$S_M(i,j)=\exp(-\frac{d(x_i,x_j)^2}{\sigma_i\sigma_j}) \tag{6}$$

$$\sigma_i=d_M(x_i,x_k),i=1,\cdots,N \tag{7}$$

式中，σ_i 等于点 x_i 到它的第 p 个近邻的欧式距离(p 通常取 $[N/2k]$，k 为类数[11])。利用这一函数重新计算图 3 所示的数据集中各点的相似度，由于点 a 位于较稀疏的簇中，点 b 位于较稠密的簇中，因此，分别计算这两点到它们第 p 个近邻的欧式距离，有 $\sigma_a>\sigma_b$，可得 $S_M(a,c)>S_M(b,c)$。这样的相似度关系有利于 AP 算法将点 c 和点 a 聚到一个类中，而将点 b 聚到另一个簇中，从而发现真实的簇结构。

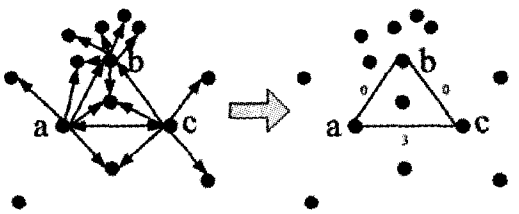


图 3　SNN 计算示意图

上述方法虽然考虑了数据点各自的邻域，但是并没有充分利用邻域提供的全部密度信息，如图 1 所示的螺旋臂数据集，即使考虑了每个数据点自身邻域的数据分布，但由于在欧式空间中 $dist(a,b)=dist(a,c)$，而且数据点沿两个螺旋的走向均匀分布得到 $\sigma_b=\sigma_c$，因此只能得到这样的相似度信息，即 $S_M(a,b)=S_M(a,c)$，点 a 很可能受距离很近的点 c 的影响，从而被误分到 c 所属的类。如果能赋予点 a 和点 b 更高的相似度，则有利于将点 a 分到正确的簇中。为此，本文借鉴 Jarvis[9]提出的共享近邻(SNN)的概念来进一步表征两点间的局部密度，一对点之间的 SNN 依赖于两个对象共享的最近邻个数，可以根据数据点的局部密度进行自动缩放，因此其对数据集密度不敏感，既能够解决稀疏数据集聚类问题，又能够解决密度不均匀的问题，更容易发现低密度区域的簇。

从前面关于流形距离核测度以及共享近邻的讨论可知，定义一个新的混合测度的相似度函数，首先它应该以具有全局一致性的流形距离核测度为基础，其次核参数应该根据簇的局部密度的变化而自适应调整。本文结合 SNN 给出了一种新的混合测度——局部密度自适应的“流形距离核”(ad-MDK)的概念，其定义如下所示：

$$S_{new}(i,j)=$$

$$\begin{cases}\exp(-\dfrac{\min\limits_{p\in P_{ij}}\sum\limits_{k=1}^{|p|-1}d_X(p_k,p_{k+1})^2}{\sigma_i\sigma_j(SNN(x_i,x_j)+1)})-1, & x_j\notin V_k(x_i)\\ \exp(-\dfrac{d_X(x_i,x_j)^2}{\sigma_i\sigma_j(SNN(x_i,x_j)+1)})-1, & x_j\in V_k(x_i)\end{cases} \tag{8}$$

式中，σ_i 表示点 x_i 到它的第 p 个近邻的欧式距离，$d_X(x_i,x_j)$ 表示流形上的线段长度。不难看出，该距离核同样满足正定条件。

2.3　基于混合测度的仿射传播聚类算法描述

根据数据集的流形结构特点，并利用本文所提的混合测度(ad-MDK)计算流形上的数据点间的相似度函数，提出了基于混合测度的近邻传播算法(APMDK)。

算法具体描述如下。

输入：数据集 $X=\{x_1,x_2,\cdots,x_N\}$；最近邻点数 L；伸缩因子 ρ；偏向参数 p；

输出：最优类代表点集合，X 划分成的 m 个聚类。

步骤：

1. 初始化：r(i,j)=0，a(i,j)=0，p_0=median(s(i,j))；
2. 计算数据集 X 的流形上的距离矩阵 D_X；
3. 根据距离矩阵 D_X 构建邻接图 G=(V,E)；
4. 根据式(8)计算邻接图中数据点的相似度矩阵 $S_{new}(i,j)$；
5. 以 $S_{new}(i,j)$ 为输入，采用 AP 原理进行聚类；
6. 判断算法是否收敛，如果收敛，判断得到的类中心个数是否满足要求，如果不满足，则可根据文献[4]设定步长 p_{step} 值 $p\leftarrow p_0+p_{step}$，重复步骤 5，直至聚类个数满足要求为止，输出最终聚类结果。

2.4　算法分析

影响仿射传播聚类算法的一个重要因素是相似度矩阵是否准确。理想的相似度矩阵应该是块对角矩阵：即对于一个聚类 $\{C_1,\cdots,C_K\}$，当数据点 x_i,x_j 属于不同类时 $S_{ij}=0$，以图 1 螺旋臂数据集为例，图 4、图 5 分别显示在欧氏空间计算的欧氏距离度量的相似度矩阵和本文所提混合测度下的相似性矩阵。对矩阵按照聚类属性重新排序后可以发现，以欧氏距离为测度的距离矩阵没有任何规律，完全无法体现不同类数据之间的区别。而经过核流形距离自适应映射后得到的相似度矩阵显示了明显的块对角模式。这说明该方法可以增大类内数据点之间的相似度，同时缩小类间数据点间的相似度。因此，相对于原有的 AP 算法，APMDK 算法能够更好地识别数据本身的聚类结构。

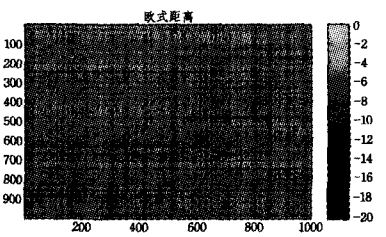


图 4　欧氏距离相似度矩阵

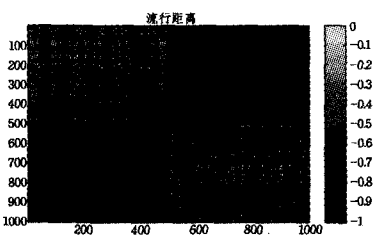


图 5　本文混合测度下相似性矩阵

3 并行仿射传播聚类算法

APMDK算法在计算任意两点间的测地线距离时，算法的复杂度为 $O(N^2\ln N+N^2L)$（L 为近邻点数），对于较大的 N 值，这一代价将会很高，使得 AP 算法处理大规模数据集的快速有效的优势无法显现出来，在很大程度上限制了算法在实际中的应用。因此为了简化算法复杂度，我们提出了一种采用“分而治之”思想[11]的并行 AP 聚类算法（P-APMDK），其基本思想是将待测数据集随机分成规模相近的 K 个子集，初始化各子集内数据点的相似度后，并行地进行 APMDK 聚类，生成各自类代表集合 $Et=\{e^j,n^j\}$，其中 e^j 是某个类的中心点，n^j 是某类中元素的个数；最后将 K 个子集的类代表点集合 Et 合并成新的数据集，并运行带权重的 AP（WAP）聚类[10]得到聚类中心，重复此过程，直至满足聚类条件。

3.1 并行仿射聚类算法（P-APMDK）

当数据规模非常大时，将待测数据集随机分成规模相近的子集，并对各数据集进行 APMDK 聚类得到类代表 $Et=\{e^j,n^j\}$。在对类代表点 $\{e^j,n^j\}$ 再次进行聚类时，数据点间的相似度和偏好度被定义为：

$$S'(i,j)=n_iS(i,j) \tag{9}$$

$$S'(i,i)=P(i)+(n_i-1)\varepsilon_i,\varepsilon_i\geqslant 0 \tag{10}$$

n_i 是以 i 点为代表点的子集所含的数据点个数，ε_i 是以 i 点为代表点的子集的平均相似度。从本质上看，相当于在做带权重 AP（WAP）聚类时，每个中心点代表的点的个数 n_i 被考虑，并且影响该点传递信息的量。因此在这些类代表点上用 WAP 聚类就等价于用 AP 对所有点聚类，并得到表达整个子集的聚类代表。由于采样的过程是在子集上进行的，因此可以并行执行，使其聚类效率大大提高，从而极大地改善对大规模数据集的处理能力。

3.2 并行仿射聚类算法实现步骤

输入：数据集 $X=\{x_1,x_2,\cdots,x_N\}$；子块数目 K；伸缩因子 ρ；聚类参数 p；

输出：最优类代表点集合，X 划分成的 m 个聚类。

步骤：

1. 对 X 中所有 N 个数据点进行随机划分，得到 K 个近似相等的划分 $A_1,A_2,\cdots,A_k$；
2. 初始化各划分 A_i 中数据的距离矩阵 D_X；
3. 对每个划分 $A_1,A_2,\cdots,A_k$ 并行进行 APMDK 聚类，得到一系列聚类代表点的集合 interExs；
4. 利用式(9)、式(10)初始化 interExs 中数据的相似度矩阵 S'(i,k)及偏向参数 S'(i,i)；
5. 在 interExS 的数据上进行 WAP 聚类，得到聚类代表点集合 Exemplars；
6. 判断 Exemplars 集合中的元素数是否满足要求，如果不满足，则可根据文献[4]设定步长 p_{step} 值 $p\leftarrow p_0+p_{step}$，重复步骤 5，直至聚类个数满足要求为止，若满足要求，则执行步骤 7。
7. 为 A 中数据点 j 指派 Exemplars 中聚类代表点作为中心，得到 m 个聚类。

3.3 算法复杂度分析

假设 1　N 个数据点的集合，随机划为 $K\subset[2,N]$ 份，每个划分的数据规模近似相等，约为 $M=N/K$。

假设 2　M 个点的聚类期望个数为 $\sqrt{\lambda M}$，$\lambda\subset[1/M,M]$。其中，λ 与偏向参数 p 相关，当 p 减小时，λ 减小，聚类数目随之减少，反之亦然。

基于并行处理的聚类方法主要分两个阶段，首先计算基于随机划分并行进行 APMDK 聚类的时间，然后利用上一阶段的结果再次进行 WAP 聚类得到最终结果，所以该算法的时间复杂度如下：

（1）基于随机划分并行进行 APMDK 聚类的时间

各个子集根据本文提出的混合测度来计算样本间的相似度矩阵，其时间复杂度包括：

计算测地线的距离的时间复杂度为 $O(M^2\ln M+M^2L)$；

计算样本的最近邻 SNN 的时间复杂度为 $O(M^2)$；

各个子集并行 AP 聚类时间复杂度近似为 $O(M^2)$，于是有：

$$T_1=O(M^2\ln M+M^2L+2M^2) \tag{11}$$

（2）利用上一阶段结果再次进行 WAP 聚类的时间

每个子集上采样得到的聚类代表数量近似相同，均为 $\sqrt{\lambda M}$，则在所有子集上得到的数据点的规模为 $\sqrt{\lambda M}$，所以一次采样之后 AP 聚类的时间为：

$$T_2=O(\lambda K^2M) \tag{12}$$

综合上述两个阶段，时间复杂度为：

$$T=O(M^2\ln M+M^2L+2M^2+\lambda K^2M)$$

划分数 K 太大或 λ 太小会导致聚类代表质量下降，聚类精度退化；反之，K 太小或 λ 太大会使得时间复杂度上升，甚至使其退化为 APMDK 算法。因此实际中要对两者进行权衡，此外子集点数不易过小，否则当类簇结构变得不明显时，聚类的精度显然会不可避免地下降。

4 实验与验证

为了测试本文所提算法的聚类效果，将本文 APMDK 算法、AP 算法[1]与数据挖掘领域最常用的 K-均值算法（K-means）[2]和能够处理流形结构的标准谱聚类（SC）[3]算法进行性能比较，分别应用到 3 种不同类型的数据集上：人工数据集、UCI 标准数据集、手写体数字数据集 USPS。另外，在后两种数据集上进行了 APMDK 算法和 P-APMDK 算法的对比实验，分析了这两种算法相比于 AP 算法的优点与代价。实验选用 Fowlkes-Mallows 指标和 Silhouette 指标以及算法运行时间对聚类结果进行评价。

实验的计算机环境为：处理器 Core 1.6GHz，内存 1GB，硬盘 250G，操作系统为 Windows XP 专业版，编程语言为 matlab2008b。

实验参数设定：其中 AP 的参数设置如下，最大迭代次数设置 $maxits=1000$，收敛迭代次数 $conv=50$，阻尼因子初始值 $l_{am}=0.85$，$P(i)$ 初始设定为相似度的中值；

标准 SC 算法核宽度 σ 都是通过反复实验得出的，可根据文献[3]采用网格寻优策略从 $\{4^{-3}\sigma_0,4^{-2}\sigma_0,4^{-1}\sigma_0,\sigma_0,4^1\sigma_0,4^2\sigma_0,4^3\sigma_0\}$ 中得到指标最高值作为结果，其中 σ_0 为相似度的平均值。

APMDK 算法需要设定伸缩因子 ρ 和最近邻点数 L 两个参数，伸缩因子通常设为 $\rho=2$，而 SNN 参数 L 则可借鉴文献[9]给出的经验值 $L=20$，这里 L 取[10,30]范围内的每个整数值，分别执行 APMDK，然后取算法的平均值；而 P-APMDK 算法的并行度 K 根据实验所取数据集大小而定，通常取 10。

实验中采用 10 重交叉验证（cross validation）来测试聚类的性能，每次从原始数据集中抽取 90%作为训练数据集，剩余的 10%作为测试数据集。每个算法进行 10 次 10 重交叉验证，最后取 10 次实验结果的均值代表该算法的聚类性能。

4.1 实验数据

1）人工数据集

图 6 中显示了 4 类业界公认比较具有“挑战性”的人工数据集，分别为：两个半环、三个圆环、人脸数据集、稀疏背景下的两个块状数据集。

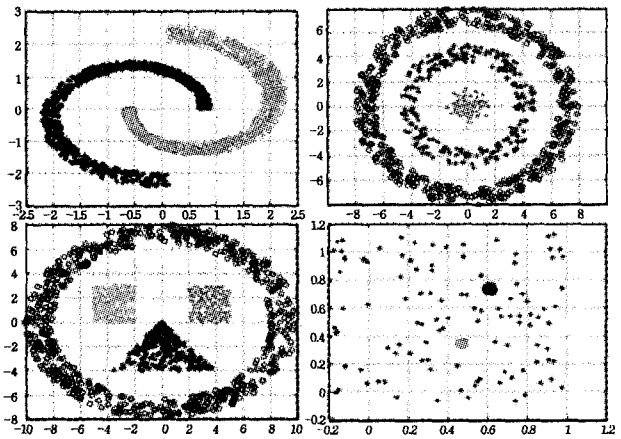


图 6　人工数据集

2）UCI 标准数据集

从 UCI 机器学习数据库[12]中选出 5 个标准数据集：Iris、Ionosphere、Wine、Glass 和 Image-segment 作为本文实验数据。数据集描述如表 1 所列。

表 1　UCI 标准数据集属性

数据集	Iris	Ionosphere	Wine	Glass	Image-segment
样本数	150	351	178	214	2310
类数	3	2	3	6	7
维数	4	34	13	9	19

3）手写体数字集

实验选用的是美国邮政手写体数字图像集（USPS）[12]，USPS 数据集是由 9298 个 16×16 维的灰度图像构成的，其中包含 7291 个训练样本、2007 个测试样本。取全部训练样本作为待聚类数据集，从中挑选 3 组较难识别和 1 组相对容易识别的数字集合进行识别，即数字{0,6}，{3,8,9}，{3,5,8}，{1,2,3,4}，数据集描述如表 2 所列。

表 2　USPS 手写体数字集属性

数据集	{0,6}	{3,5,8}	{3,8,9}	{1,2,3,4}
样本数	1736	1756	1844	3046
类数	2	3	3	4
维数	16×16	16×16	16×16	16×16

4.2　评价准则

聚类的评价标准分为有监督的聚类评价和无监督的聚类评价。其中有监督的聚类评价是将聚类结果与真实的类标号进行比较，来评价聚类结果的好坏；而无监督的聚类评价只能通过衡量聚类结果中各个类簇的凝聚度和离散度等信息来衡量聚类结果。具体而言，本文采用 3 种评价指标对聚类结果进行评价：Fowlkes-Mallows 指标、Silhouette 指标以及算法运行时间。

Fowlkes-Mallows 指标是有监督的评价标准，直接将聚类结果与真实的类标号进行比较，计算方法为：

$$FM(C,C')=\sqrt{\frac{TP}{TP+FP}\cdot\frac{TP}{TP+FN}} \tag{13}$$

式中，C_i，C_j 是 k 个聚类中两个不同的类，TP 表示在 C_i，C_j 上的同一类数据对的数量；FN 表示在 C_j 上但不在 C_i 上的同一类数据对的数量，FP 表示在 C_i 上但不在 C_j 上的同一类数据对的数量，TN 表示不在 C_i，C_j 上的同一类数据对的数量，这里：$TP+FP+FN+TN=n(n-1)/2$ 为数据集中所有数据对的最大数。通常 FM 值处于 0 与 1 之间，且越大表示一致性越好，当聚类结果与正确类标完全一致时，$FM=1$。

Silhouette 指标是能够反映聚类结构的类内紧密性和类间可分性的无监督评价标准，对于 n 个样本 k 个聚类 $C_i(i=1,\cdots,N)$，一个样本 t 的 Silhouette 指标为

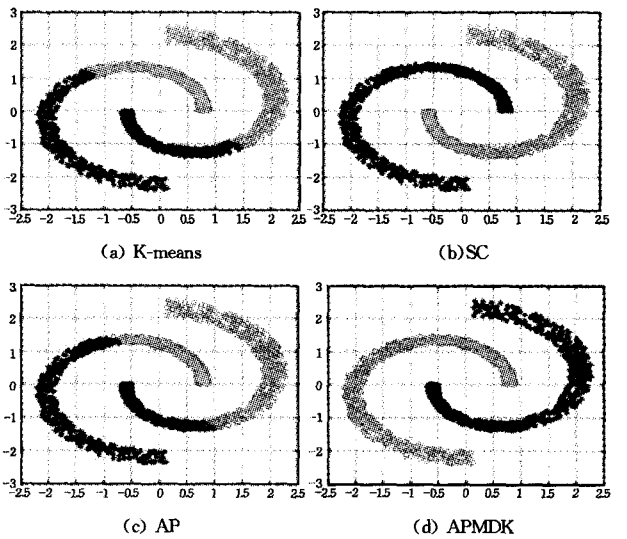
$$S_{il}(t)=\frac{b(t)-a(t)}{\max\{a(t),b(t)\}} \tag{14}$$

式中，$a(t)$ 为聚类 C_j 中的样本 t 与 C_i 内所有其他样本的平均不相似度或距离。$d(t,C_i)$ 为 C_j 的样本 t 到另一个类 C_i 的所有样本的平均不相似度或距离，记 $Silav(C_i)$ 为一个聚类 C_i 的所有样本的 $S_{il}(t)$ 平均值，它反映了类 C_i 的紧密性和可分性，而一个数据集的所有样本的 $S_{il}(t)$ 平均值 $Silav(C_i)$ 则可以反映聚类结果的质量，聚类结果的 Silhouette 值超过 0.5 说明各个类簇能明显地分开（好的分类），低于 0.5 表明一些类簇有重叠的情况，而 0.2 以下是缺乏实质的聚类结构。

4.3　结果与分析

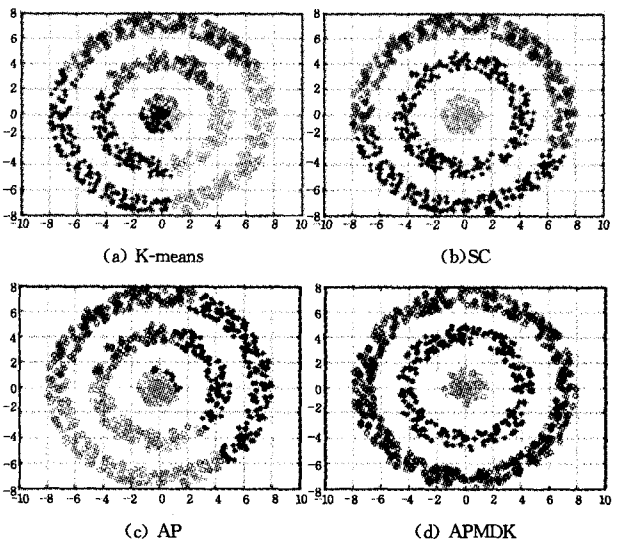
4.3.1　人工数据集

图 7－图 10 分别显示了 4 种方法分别在 4 个人工数据集上的聚类结果。



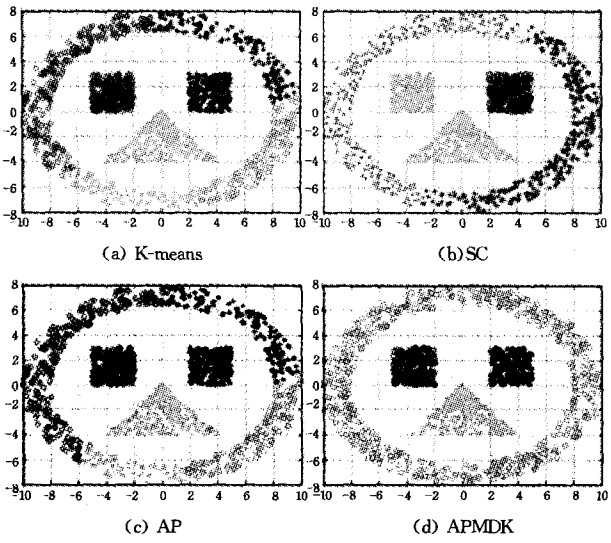
(a) K-means　(b) SC　(c) AP　(d) APMDK

图 7　两个半环数据集上的聚类结果比较



(a) K-means　(b) SC　(c) AP　(d) APMDK

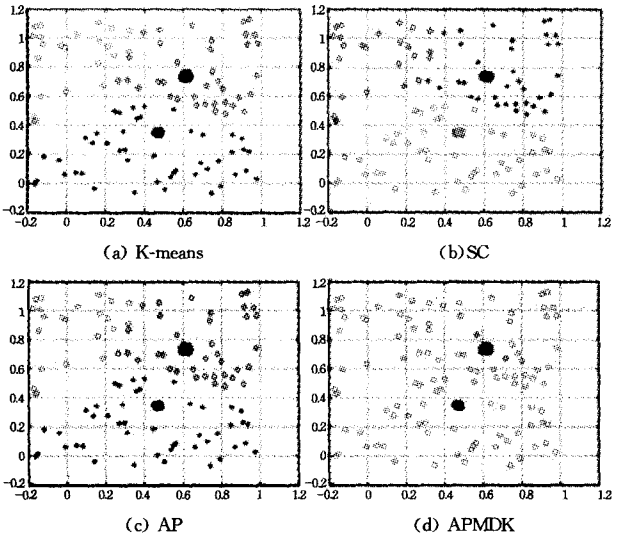
图 8　3 个圆环数据集上的聚类结果比较



(a) K-means (b) SC

(c) AP (d) APMDK

图 9 人脸数据集数据集上的聚类结果比较



(a) K-means (b) SC

(c) AP (d) APMDK

图 10 稀疏背景下两个块状数据集上的聚类比较

从实验结果可以看出，基于欧氏距离的 K-means 和 AP 算法完全无法处理流形结构的聚类问题，而本文所提 APMDK 算法成功地找出了 4 种数据的流形结构。同时发现，标准 SC 算法虽然也具备发现流形的能力，但聚类结果对核参数选取非常敏感，且仅能够聚类明显分离的流形，当不同流形之间相距很近或者有部分重叠时，SC 算法不能给出满意的结果，算法运行时间随着样本数量的增长而变得非常缓慢。

4.3.2 UCI 标准数据集

实验对 APMDK、P-APMDK、AP 算法以及 SC 算法在 4 个数据集上进行了测试，表 3 给出了 APMDK、P-APMDK、AP 和 SC 算法在 5 个数据集上取得的 FM 指标和 Silhouette 指标以及算法运行时间。

从实验结果可以看出，基于欧氏距离的 K-means 和 AP 算法完全无法处理流形结构的聚类问题，而本文所提 APMDK 算法成功地找出了 4 种数据的流形结构。同时发现，标准 SC 算法虽然也具备发现流形的能力，但聚类结果对核参数选取非常敏感，且仅能够聚类明显分离的流形，当不同流形之间相距很近或者有部分重叠时，SC 算法不能给出满意的结果，算法运行时间随着样本数量的增长而变得非常缓慢。从表中可以看出：

1）本文所提出的两个算法的聚类性能均优于原始 AP 算法。说明基于流形距离核测度的相似度矩阵能够更好地表征数据集潜在的结构信息，显著提高了反映类簇内部结构的 Silhouette 指标，进而提高了算法的聚类性能。

2）APMDK 算法的聚类性能最好，尤其是在 Wine 和 Glass 数据集上，该算法相比于其它 3 种算法，性能有明显提高，这是因为 APMDK 算法对规模较小的数据集可以得到很好的聚类效果，使用并行算法 P-APMDK 反而会降低聚类准确精度。

3）如在 Image-segment 数据集上的聚类结果所示，对较大规模数据进行聚类时，P-APMDK 算法可以获得与 APMDK 算法几乎同样好的聚类性能，而前者的运行速度明显高于后者，且聚类精度没有太大变化。因此，当用户对聚类效率有较高要求的情况下，可以采用 P-APMDK 算法进行聚类分析。

表 3 UCI 数据集上的聚类结果和运行时间

数据集	SC			AP			APMDK			P-APMDK		
	FM	Sil	Time	FM	Sil	Time	FM	Sil	Time	FM	Sil	Time
Iris	0.87	0.51	3.09	0.88	0.53	1.62	0.93	0.56	2.98	0.93	0.56	1.03
Ionosphere	0.73	0.50	9.78	0.56	0.49	7.11	0.87	0.53	10.10	0.86	0.53	5.38
Wine	0.68	0.46	3.22	0.64	0.45	2.42	0.89	0.54	3.15	0.85	0.52	1.81
Glass	0.64	0.44	4.79	0.60	0.43	2.89	0.81	0.52	4.67	0.76	0.52	2.01
segment	0.61	0.48	25.8	0.57	0.45	15.1	0.84	0.53	22.1	0.84	0.53	9.84

4.3.3 USPS 手写体数字集

图 11 和图 12 分别为 5 种算法在 USPS 数据集上的聚类结果和运行时间对比示意图。从图 11 可看出，APMDK 算法的 *FM* 指标均大于 84%，SC 和 AP 算法次之，分别为78.45%和 73.25%，K-means 算法为 69.60%，说明本文所提流形距离核测度更容易发现流形结构，而且 P-APMDK 算法更适合大规模复杂的数据集，在得到理想聚类结果的同时大大降低了算法计算复杂度。从图 11 和图 12 可以看出，APMDK 和 SC 算法的运行时间最长，明显大于其他算法。相反，P-APMDK 算法运行速度最快，AP 算法次之，当数据规模较大时 SC 算法变得非常耗时。

综合两图的实验结果发现，P-APMDK 算法能够在比 APMDK 算法小得多的运行时间内取得和 APMDK 算法相近的聚类结果，从而验证了本文并行算法的优点。

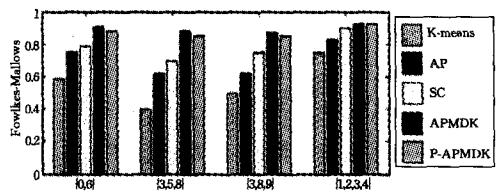


图 11 不同算法聚类效果对比图

（下转第 195 页）

decision systems based on information theory[J]. Information Sciences,2009,179:1694-1704

[7] 洪晓蕾,王燕,莫执文,等.集值不完备信息系统上的一种知识约简方法[J].四川师范大学学报:自然科学版,2007,30(3):266-269

[8] 陈子春,秦克云.集值信息系统在相容关系下的属性约简[J].模糊系统与数学,2009,23(1):150-154

[9] 陈子春.集值信息系统的知识发现与属性约简研究[D].成都:西南交通大学,2011

[10] Tsumoto S. Mining diagnostic tules from clinical databases using rough sets and medical diagnostic model[J]. Information Siences,2004,162:65-80

[11] Tsai Y C, Cheng C H, Chang J R. Entropy-based fuzzy rough claasification approach for extracting rules[J]. Expert Systems with Application,2006,31(2):436-443

[12] Hu Q H, Yu D R, Xie Z X. Neighborhood classifiers[J]. Expert Systems with Application,2008,34:866-876

[13] Li Y, Shiu S C K, Pal S K. Combining feature reduction and case selection in building CBR classifiers[J]. IEEE Transactions on Knowledge and Data Engineering,2006,18:415-429

[14] 宋笑雪,解争龙,张文修.集值决策信息系统的知识约简与规则提取[J].计算机科学,2007,34(4):182-184

[15] 宋笑雪,张文修.基于集值决策属性的集值信息系统[J].计算机工程与应用,2007,43(17):8-10

[16] Leung Y, Wu W Z, Zhang W X. Knowledge acquisition in incomplete information systems: a rough set approach[J]. European Journal of Operational Research,2006,168(1):164-180

[17] Yang X, Xie J, Song X, et al. Credible rules in incomplete decision system based on descriptors[J]. Knowledge-Based Systems,2009(22):8-17

[18] 吴鹏,杨勇,张阿红.基于集值的 Rough 集扩充模型[J].计算机工程与应用,2008,44(32):134-136

[19] 鲍忠奎,杨善林.集值信息系统的粗糙集扩展模型[J].计算机工程与应用,2011,47(35):22-24

[20] 乔全喜,秦克云.集值信息系统基于对称限制相容关系的属性约简[J].计算机工程与应用,2011,47(35)

[21] 张文修,梁怡,吴伟志.信息系统与知识发现[M].北京:科学出版社,2003

(上接第 172 页)

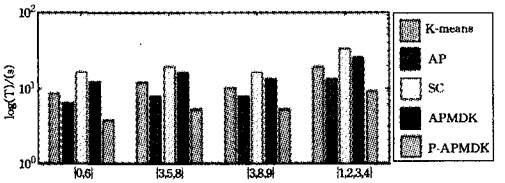


图 12 不同算法聚类时间对比图

从以上的实验结果可以得出如下结论:

1. 基于流形距离核的混合测度能够揭示数据集的整体结构信息,能够根据数据间局部密度自适应调整聚类参数的特点,能够灵活适用于各种复杂结构的数据集,拓宽了 AP 算法的应用范围。

2. 对于流形结构比较单一或者相互分离的数据集,APMDK 和 P-APMDK 算法都能获得比较理想的聚类效果,同时,对于大规模复杂的数据集,采用 P-APMDK 算法能够在改善聚类性能的同时明显提高算法的运行效率。

3. 对同一数据集,P-APMK 算法的聚类效果稍差于 APMDK 算法,在对运行速度有较高要求的场合,可以使用 P-APMDK 算法牺牲部分精度,提高处理速度。

结束语 本文针对仿射传播算法只能应用于超球形数据集的不足,提出了基于混合测度的仿射传播算法(APMDK),该算法能够同时满足聚类的局部一致性和全局一致性假设,克服了原有算法不能处理非凸形结构聚类的缺陷。在此基础上,为了进一步降低计算复杂度,能够在保持聚类性能的同时大大提高算法运算速度,提出了一种基于"分治"思想的并行算法——P-APMDK 算法。实验结果验证了本文算法在处理大规模复杂数据上的优越性。研究如何改进当前算法的参数设置问题,从而进一步提高算法的精度,是我们今后重要的研究方向。

参 考 文 献

[1] Frey B J, Dueck D. Clustering by passing messages between data points[J]. Science,2007,315(5814):972-976

[2] Jain A K. Data clustering:50 years beyond K-means[J]. Pattern Recognition lett,2009,9(11)

[3] von Luxburg U. A tutorial on spectral clustering[R]. Technical report. Max Planck Institute for Biological Cybernetics,2006

[4] Wang K, Zhang J, Li D, et al. Adaptive Affinity Propagation Clustering[J]. Acta Automatica Sinica,2007,33(12):1242-1246

[5] 肖宇,于剑.基于近邻传播算法的半监督聚类[J].软件学报,2008,19(11):2803-2813

[6] 董俊,王锁萍,熊范纶.可变相似性度量的近邻传播聚类[J].电子与信息学报,2010,32(3):509-514

[7] Zhou D, Bousquet O. Learning with Local and Global Consistency [A]// Proceeding of advances in Neural Information Processing Systems [C]. Cambridge: MIT Press,2004:321-328

[8] Zelnik-Manor L, Perona P. Self-tuning spectral clustering[M]. Advances in Neural Information Processing Systems (NIPS), Cambridge, MA: MIT Press,2004

[9] Ertoz L, Steinbach M, Kumar V. A new shared nearest neighbor clustering algorithm and its applications[C]// Workshop on Clustering High Dimensional Data and its Applications, Second SIAM International Conference on Data Mining. Arlington, VA, USA,2002

[10] Zhang X L, Furtlehner C. Toward autonomic grids: Analyzing the job flow with affinity streaming[C]//KDD '09: Proceedings of the 15th ACM SIGKDD international conference on Knowledge discovery and data mining. 2009

[11] Song Y, Chen W-Y, Bai H, et al. Parallel spectral clustering[C]// Proceedings of Learning and Principles and Practice of Knowledge Discovery in Databases (ECML/PKDD). 2008:374-389

[12] Asuncion A, Newman D J. UCI machine learning repository. University of California, Irvine, School of Information and Computer Sciences[C/OL]. http://archive.ics.uci.edu/ml/datasets.html,2007