基于核自适应的近邻传播聚类算法*

付迎丁，兰巨龙
（国家数字交换系统工程技术研究中心，郑州 450002）

摘　要：近邻传播聚类(AP)方法是近年来出现的一种广受关注的聚类方法，在处理多类、大规模数据集时，能够在较短的时间得到较理想的结果，因此与传统方法相比具有很大的优势。但是对于一些聚类结构复杂的数据集，往往不能得到很好的聚类结果。通过分析数据的聚类特性，设计了一种可以根据数据结构自动调整参数的核函数，数据集在其映射得到的核空间中线性可分或几乎线性可分，对该核空间中的数据集进行近邻传播聚类，有效提高了 AP 聚类的精确度和速度。算法有效性分析以及仿真实验验证了所提算法在处理大规模复杂结构数据集上的性能优于原始 AP 算法。

关键词：近邻传播聚类；核聚类；核自适应聚类；流形学习
中图分类号：TP18　　**文献标志码**：A　　**文章编号**：1001-3695(2012)05-1644-04
doi:10.3969/j.issn.1001-3695.2012.05.011

Kernel-based adaptation for affinity propagation clustering algorithm

FU Ying-ding, LAN Ju-long
(*China National Digital Switching System Engineering & Technological R&D Center*, *Zhengzhou* 450002, *China*)

Abstract: AP algorithm has become increasingly popular in recent years as an efficient and fast clustering algorithm. AP has better performance on large and multi-class dataset than the existing clustering algorithms. But for the datasets with complex cluster structures, it cannot produce good clustering results. Through analyzing the property of data clusters, this paper proposed a kernel function, optimized that the parameters automatically according to the dataset structure, and the dataset in kernel space were linearly separable or almost linearly. Carried AP on the kernel space, it had a kernel-adaptive affinity propagation clustering algorithm(KA-APC). Compared with the original AP clustering, it had the advantages of effectively dealing with the large multi-scale dataset. The promising experimental results show that this algorithm outperforms the original AP algorithm.

Key words: affinity propagation(AP); kernel clustering; kernel adaptive clustering; manifold learning

0　引言

聚类分析是一种有效的数据分析方法，在识别数据的内在结构上具有极其重要的作用，被广泛地应用于数据挖掘、模式识别、机器学习等领域。聚类分析是根据对象在某些属性上的相似性，将对象划分成群组或类簇的过程。聚类的目标是使得类内的对象尽可能地相似，类间的对象尽可能地相异。根据数据集的聚类规则，聚类算法有多种，而且分类方法也不尽相同。孙吉贵等人[1]将聚类算法大致分为层次方法、划分方法、基于密度和网格的方法及其他聚类方法四类。近期还有一些结合仿生学思想的聚类算法，如蚁群算法和粒子群算法，这些方法都取得了较好的聚类结果。但是目前没有任何一种算法是普遍适用于所有数据集的。众多方法中经典 K-means 算法模型简单，计算过程相对高效，因此，对 K-means 算法的修正和改进引起了持续的关注[2]。

在现有的无监督聚类算法中，普遍使用的 K-means 聚类是一种基于中心的聚类算法，它存在以下几个缺点：

a)在紧凑的超球形分布的数据集上有很好的性能，然而当数据结构是非凸的，或数据点彼此交叠严重时，K-means 算法往往会失效。

b)K-means 算法对初始聚类中心的选择敏感，而且算法利用迭代最优化方法寻找最优解，因而不能保证收敛到全局最优解。

2007 年，Frey 等人[3]首次提出了同属于 K 中心聚类方法的近邻传播算法(AP)，克服了 K-means 算法的缺点，能够在较短的时间内处理大规模数据集，得到较理想的结果。该算法与经典的 K-means 算法具有相同的目标函数，但其在算法原理上与 K-means 算法存在很大的不同。相比于其他传统的聚类算法，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的选择。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，并在处理多类数据时运算速度快，所以性能更好。目前该算法已应用于人脸识别、网络文本挖掘以及图像分类等问题上，均取得了良好的效果[3~6]。但是，由于近邻传播算法是基于中心的聚类算法，因此同样存在这类算法共有的缺点，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，不能得到合理的聚类结果。

本文将核方法[7]与近邻传播算法相结合，通过将原始聚类空间映射到高维特征空间，使得聚类空间适用于近邻传播算

收稿日期：2011-10-08；**修回日期**：2011-11-14　　**基金项目**：国家“863”计划资助项目(2009AA01A346)

作者简介：付迎丁(1987-)，男，北京人，助理工程师，硕士研究生，主要研究方向为宽带信息网络(fydkill1@163.com)；兰巨龙(1962-)，男，教授，博导，主要研究方向为宽带信息网络.

法,同时考虑数据的空间属性,利用数据在原始空间的分布流形来约束和调整核函数,使得数据在高维空间映射的模型更加准确。本文提出一种基于核自适应的近邻传播聚类(KA-APC)算法,取得了较好的聚类效果。实验结果表明,基于核自适应的近邻传播聚类算法性能与原 AP 聚类算法性能相比有明显的提高。

1　基本的近邻传播算法

近邻传播聚类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。该算法的简要流程如下: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 $S_{N\times N}$。算法起始阶段将所有的样本都看做是潜在的聚类中心。同时,每个样本也作为网络中的一个节点,两种分别被称做吸引度(responsibility)和归属度(availability)的消息(统称为吸引力消息)在各个节点之间不断地传递迭代。越靠近聚类中心位置的点,其对其他所有样本点的吸引力之和越大,因此作为聚类中心的可能性就越大;反之,处于聚类边缘的点,对其他点的吸引力较小,成为聚类中心的可能性就越小。AP 算法的核心就是这两个消息不断地更新过程,更新式如下:

$$r(i,j)\leftarrow s(i,j)-\max_{j'\neq j}(a(i,j')+s(i,j'))$$

$$\text{if}\quad i\neq j, a(i,j)\leftarrow \min_{i\neq j}\{0, r(j,j)+\sum_{i'\neq i,i'\neq j}\max(0,r(i',j))\}$$

$$a(j,j)\leftarrow \sum_{i'\neq i,i'\neq j}\max(0,r(i',j))$$

最后,经过大量交替更新之后,得到所有聚类中心以及各中心与样本点之间的关系。

2　基于核自适应的近邻传播聚类算法

AP 算法仅在处理超球形紧密分布的数据集时具有优异的表现,但是对于许多聚类应用领域,如图像识别、空间数据的挖掘等问题,数据集通常具有任意形状和多重尺度,从而给 AP 算法带来了困难。

2.1　基于核函数的近邻传播算法

本文借鉴核聚类方法的思想,采用非线性变换 Φ 将输入数据空间 X 映射到一个高维特征空间 H,在高维特征空间中扩展近邻传播算法,对变换后的特征矢量 $\Phi(x)$ 作 AP 聚类分析。

令 $X=\{x_1,x_2,\cdots,x_N\}$ 为模式空间 R^n 的一个有限数据集,$\boldsymbol{x}_i(i=1,2,\cdots,N)$ 是该空间中的一个向量,变换后的高维空间向量为 $\Phi(\boldsymbol{x}_i)(i=1,2,\cdots,N)$。数据点在特征空间的距离定义为

$$d_H(x_i,x_j)=\sqrt{\|\Phi(x_i)-\Phi(x_j)\|^2}=\sqrt{\Phi(x_i)\cdot\Phi(x_i)-2\Phi(x_i)\cdot\Phi(x_j)+\Phi(x_j)\cdot\Phi(x_j)} \quad (1)$$

输入空间中的点积形式在高维特征空间可以用 Mercer 核来表示,为 $k(x_i,x_j)=\langle\Phi(x_i),\Phi(x_j)\rangle$,记做 k_{ij}。则式(1)变为

$$d_H(x_i,x_j)=\sqrt{k(x_i,x_i)-2k(x_i,x_j)+k(x_j,x_j)}=\sqrt{k_{ii}-2k_{ij}+k_{jj}} \quad (2)$$

常用的核函数有多项式核函数、高斯核函数和双曲正切核函数。由于高斯核函数对应的特征空间是无穷维的,有限的样本集在特征空间中必定可分,因此本文采用高斯核函数 $k(x_i,x_j)=\exp(\frac{-\|x_i-x_j\|^2}{2\sigma^2})$。当取高斯核时,$k(x,x)=1$,所以式(2)可简化为 $d_H(x_i,x_j)=\sqrt{2-2k(x_i,x_j)}$,以该式作为聚类的距离度量函数,则算法中的相似度函数为

$$s(i,j)=2k(x_i,x_j)-2=2\exp(\frac{-x_i-x_j^2}{2\sigma^2})-2 \quad (3)$$

其中:σ 为高斯核宽度,该参数的选择与超球体半径直接相关。

2.2　核参数的自动调整

在基于核的学习算法中,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 会极大地影响核聚类算法的泛化性能,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,是本节讨论的重点。

2.2.1　数据的空间分布特性

在很多聚类应用领域,数据集合通常都存在一定的空间特性,两个点的相近性不是决定性因素,数据分布的潜在意义才是至关重要的。这些都是基于半监督平滑假设[8,9]的:

a)局部一致性假设。如果两个点在空间位置上相邻,它们就很可能来源于同一类。

b)全局一致性假设。如果边缘概率分布函数 $p(x)$ 是在流形中,那么在同一流形上的数据点属于同一类的可能性比较大。

传统的以欧氏距离为测度的相似度度量虽然能够反映数据的局部一致性,却不能反映数据集的流形结构特征。因此本文考虑首先搜索流形边缘,然后将流形的边缘分布特性与核映射目标函数相结合,在求解核空间相似度距离的过程中优化核参数。具体地说,就是通过原始特征空间中数据点的分布属性来调整搜索最佳核参数值。

首先给出聚类数据几个空间分布特征的定义。

定义 1　可达性。设存在数据集 $X=\{x_1,x_2,\cdots,x_N\}$,$x_i,x_j\in X$,如果 $x_j\in V_k(x_i)$,其中 $V_k(x_i)=\min_k\{\|x_1-x_i\|^2,\|x_2-x_i\|^2,\cdots,\|x_N-x_i\|^2\}$,即 x_j 在 x_i 的 k-邻域(离 x_i 最近的 k 个点)中,认为点 x_j 是点 x_i 直接可达到的;反之不成立。如果存在一系列对象 $x_1,x_2,\cdots,x_n$,其中点 $x_{i+1}(1\leqslant i\leqslant n-1)$ 是点 x_i 直接可达到的,则 x_n 是 x_1 可达到的;反之不成立。

定义 2　连通域。设点 x_i 与点 x_j 都和点 x_k 是可达到的,则称对象 x_i 连通对象 x_j,C 为数据集 X 的非空子集,如果任意点 $x_i\in C$ 与 C 内的其他点都是连通或者可达到的,且 x_i 与任意点 $x_k\in\{X-C\}$ 都是不可达到的且非连通的,则称子集 C 为一个连通域。

定义 3　核心点、噪声和边界点。如果点 x_i 的 k-邻域内与 x_i 相邻的点数大于 ε,那么点 x_i 称为核心点;不包含在任何连通域中的数据点称为噪声;既不是核心点又不是噪声点的所有数据点统称为边界点。

如图 1 所示,令 $k=4,\varepsilon=2$,图 1(a)为人工数据集分布机构图;(b)中点 x_1 是点 x_2 直接可达到的,但是反过来不成立;与 x_4 和 x_5 是相邻的,点 x_1 是 x_4 可达到的,因此点 x_5 与点 x_1 相连通,且点 x_5 可达到点 x_1,但是反过来不成立;图中 $x_1x_2x_3x_4x_5$ 在同一流形上,同样以这些点为核心点的数据点也在同一流形上。

流形搜索算法的具体过程如下:

令 $X_{un}=X,m=0$;

初始化 k、ε;

while $X_{un}\neq\varnothing$ do

　任意选择一个 $x\in X_{un}$;

if x 不是核心点,则
把 x 标记为噪声点;
$X_{un} = X_{un} = \{x\}$;
else x 是核心点,则
$m = m + 1$;
搜索 X 中所有与 x 可达到的点;
把 x 及其所有可达点分配到 C_m 簇中;
$X_{un} = X_{un} - C_m$;
end if
end while

该算法执行过程中标记的噪声点不是最终的,算法一开始一些边界点有可能被标记为噪声点,但随着算法的推进,搜索该点所在簇的核心点 x 的可达点时,该点会被作为边界点分配到簇中;如果是噪声点,则因为其与任一流形中的核心点都不可达,所以在聚类过程中它的噪声点标记不变。另外,算法中选择的 k、ε 值将影响算法的结果,参数的选择应根据数据分布密度来调整,使算法能够检测到密度最小的流形。流形搜索算法复杂度为 $O(N \log_2 N)$。

2.2.2 基于流形的核参数调整方法

理想的核函数应该使得位于同一流形分布上的数据点在高维空间中离得很近,而不同流形上的数据点距离相对较远。

基于这一要求给出核映射条件:X 中的向量 $\boldsymbol{x}$ 满足 $\{x: \sqrt{\| \Phi(x) - c \|^2} = R\}$,则该点映射到 H 中以 c 为球心的球面上,这些点在 X 中对应于流形的边界;位于流形内部的点,映射在球体内;而位于不同流形的两个数据点,对应 H 空间连接它们的任意路径上必然存在点 y,使得 $\sqrt{\| \Phi(y) - c \|^2} > R$,$c$ 是这两个数据点中任意一个映射到 H 空间中所在球体的中心,R 是相应的球体半径。

图 2(a)中数据集 circles 包含三个互相嵌套的环形,三类数据在欧氏距离下;(b)是其经过核变换后的数据分布结构,三类数据线性可分,类间距离明显变大。图 2 中 x_1 和 x_2 分别是不同的两个流形上的点,y 是 H 空间中 x_1x_2 连线上的点,y 到任一聚类中心的距离均大于该聚类半径。

(a) 人工数据集 (b) 流形搜索过程 (a) Circles (b) 核变换后的数据分布

图1 人工数据集及流形搜索过程 图2 Circles及核变换后的数据分布

这里的核映射函数中加入了数据集的流形分布约束,核参数 σ 根据不同的流形而作出相应调整。高维空间的相似度目标函数表示为

$$\min(R_j^2 + C\sum_{i=1}^{N_j}\zeta_i) \tag{4}$$

$$\text{s.t. } \| \Phi(x_i) - c_j \|^2 \leqslant R_j^2 + \zeta_i \quad \forall i, \zeta_i \geqslant 0; i = 1,2,\cdots,N$$

$$\sqrt{\| \Phi(x) - \Phi(y) \|^2} \geqslant \max(R_j, R_k) \quad x \in C_j, y \in C_k; j,k = 1,2,\cdots,K, j \neq k$$

其中:$C_j\{j = 1,2,\cdots K\}$ 表示第 j 个流形包含的数据集;R_j 是第 j 个流形映射到高维特征空间对应的超球体半径;c_j 是第 j 个流形对应的超球体球心;ζ_i 是松弛变量,允许软边界的存在,缩小了超球体的半径。核参数自调整的过程即在高维特征空间 H 中寻找最小包络超球体半径 R_j 的过程。引入 Lagrange 函数,即

$$L(R,c,\zeta,\mu,\lambda) = \sum_{j=1}^{K}(R_j + C\sum_{i=1}^{N_j}\zeta_i - \sum_{i=1}^{N_j}\mu_i\zeta_i) - \sum_{j=1}^{K}\sum_{i=1}^{N_j}\lambda_i(R_j^2 + \zeta_i - \| \Phi(x_i) - c_j \|^2) \tag{5}$$

其中:$\zeta_i \geqslant 0, \mu_i \geqslant 0$ 为 Lagrange 乘子;C 为常数。根据 KKT 条件,令式(5)的倒数等于 0,则 Wolfe 对偶形式为

$$\begin{cases} \max\limits_{\lambda_i} \quad J = \sum\limits_{k=1}^{K}\sum\limits_{i=1}^{N_j}\lambda_i k(x_i,x_i) - \sum\limits_{k=1}^{K}\sum\limits_{i,j,i\neq j}\lambda_i\lambda_j k(x_i,x_j) \\ \text{s.t. } 0 \leqslant \lambda_i \leqslant C, \sum\limits_i \lambda_i = 1, i = 1,\cdots,N_j \end{cases} \tag{6}$$

$$\mu_i\zeta_i = 0$$

$$\lambda_i(R_j^2 + \zeta_i - \| \Phi(x_i) - c_j \|^2) = 0$$

$$c_j = \sum_{i=1}^{N_j}\lambda_i x_i$$

$$\lambda_i = C - \mu_i$$

上式是一个二次函数求极值的问题,存在最优解;而且只有 $\lambda_i \neq 0$ 的点被认为是有意义的点而用于计算球心,即支持向量。在实际情况下,大部分数据对应的 λ 都等于 0。因此,只有很少一部分数据参与运算。根据上式,很容易计算最优 Lagrange 乘子,从而求得核参数,并根据式(6)算出超球体半径、球心以及各个支持向量对应的权值系数。

因此,AP 算法中数据点在 H 空间的相似度可以定义为

$$S_H(i,j) = -d_H^2(i,j) = -\| \Phi(x_i) - \Phi(x_j) \|^2 = -2 + 2k(x_i,x_j)$$

其中:

$$k(x_i,x_j) = \begin{cases} 2\exp(\dfrac{-\| x_i - x_j \|^2}{R_k^2}) - 2 & x_i, x_j \in C_k \\ \| c_k - c_{k'} \|^2 - k(x_i,c_k) - k(x_j,c_{k'}) & x_i \in C_k, x_j \in C_{k'}, k \neq k' \end{cases}$$

$$r(i,j) \leftarrow s(i,j) - \max_{j' \neq j}\{a(i,j') + s(i,j')\} =$$

$$-2 + 2k(x_i,x_j) - \max_{j' \neq j}\{a(i,j') + 2k(x_i,x_{j'}) - 2\}$$

$$a(i,j) \leftarrow \min\{0, r(j,j) + \sum_{i' \neq i, i' \neq j}\max\{0, r(i',j)\}\}, i \neq j$$

$$a(j,j) \leftarrow \sum_{i' \neq j}\max\{0, r(i',j)\}$$

2.3 基于核自适应的近邻传播聚类算法

输入:数据集 $X = \{x_i\}_{i=1}^N$,$r(i,j)$,$a(i,i)$,$s(k,k)$,ε,p。

输出:X 被划分成的 M 个聚类。

a)初始化 ε、p,令 $r(i,j) = 0$,$a(i,i) = 0$。

b) $\forall x_i, x_j \in X$,计算两者之间的欧氏距离 $d_X(x_i, x_j) = \sqrt{\| x_i - x_j \|^2}$,得到距离矩阵 D_X。

c)根据距离矩阵 D_X 进行流形搜索,找到所有满足条件的流形分布 $\{C_k\}_{k=1}^K$。

d)根据流形分布建立目标函数 $\min(R_j^2 + C\sum_{i=1}^{N_j}S_i)$,求解合适的参数 σ,使得数据经高斯核函数 $k(x,y) = \exp(\dfrac{-\| x - y \|^2}{2\sigma^2})$ 变换后满足该目标函数。

e)求解核空间相似度矩阵 $[s_H(i,j)]_{N\times N}$。

f)利用 AP 算法原理聚类,$r(i,j)$ 和 $a(i,j)$ 按如下方法更新:

$$r(i,k) \leftarrow \lambda r(i,k) + (1-\lambda)\{s(i,k) - \max_{k' \neq k}[a(i,k') + s(i,k')]\}$$

If $i \neq k$, $a(i,k) \leftarrow \lambda a(i,k) + (1-\lambda)\min_{i \neq k}\{0, r(k,k) + \sum_{i' \neq i, i' \neq k}\max[0, r(i',k)]\}$

$$a(k,k) \leftarrow \lambda a(i,k) + (1-\lambda)\{\sum_{i' \neq i, i' \neq k}\max[0, r(i',k)]\}$$

g)判断迭代过程是否满足停止条件:超过某一迭代最大数目;信息改变量低于某一固定阈值;选择的类中心在连续几步迭代过程中保持稳定;满足其中一个停止条件即可。

h)判断得到的类中心个数是否满足要求,如果不满足,则改变 p 值,重复整个迭代过程,直至聚类个数满足要求为止,输出最终聚类结果。

3　实验与验证

本文选择了六个数据集分别对 KA-APC、AP 和 K-means 算法进行了聚类比较,以检验 KA-AP 算法是否能通过调整相似性度量获得更准确的聚类效果。实验的计算机环境:处理器为 Core2 2.4 GHz,内存为 2 GB,硬盘为 250 GB,操作系统为 Windows 7 专业版,编程语言为 MATLAB 2008b。

3.1　实验数据

实验中所采用的六个数据集分别为 Iris、Wine、Ionosphere、FaceImage、Circles 和 Spirals(图 3)。其中:Iris、Wine 和 Ionosphere 三个数据集均是 UCI 机器学习数据集储存库中较为常用的数据集;FaceImage 数据集是从近邻传播聚类算法作者的个人主页(http://www. psi. toronto. edu/index. php? q = affinity%20propagation)上下载得到的;Circles、Spirals 是文献[10]所给出的比较具有挑战性的两类人工数据集。以上数据集都是已知聚类结果且适合作聚类分析的基准数据集。表 1 给出了数据集的相关信息。

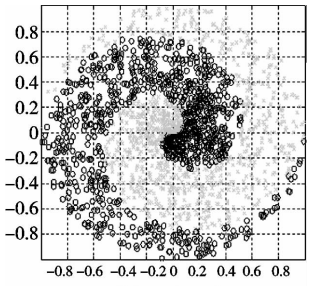


图3　Spirals

表 1　实验中使用的数据集

参数	Iris	Ionosphere	Wine	Circles	Spirals	FaceImage
样本数	150	351	178	600	1 000	900
类数	3	2	3	3	2	100
维数	4	34	13	2	2	5 050
聚类属性	松散	重叠	少量重叠	重叠、非凸	重叠、非凸	重叠、非凸

实验采用 10 倍交叉验证方法(10-fold cross validation),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。每种算法进行 10 次 10 倍交叉验证,取每种算法 10 次交叉验证的均值进行对比。

3.2　评价准则

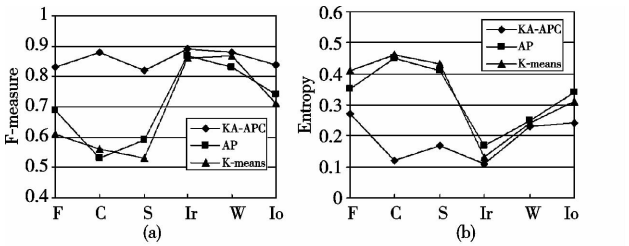
实验采用三种评价指标对聚类结果进行评价,即 F-measure 和 Entropy 指标以及算法运行时间。其中 F-measure 是通过算法的准确率和查全率计算得到。对于类 t 和聚类 C_k 的准确率和查全率分别是:$\mathrm{Prec}(t,C_k)=\frac{N_{tk}}{N_k}$,$\mathrm{Rec}(t,C_k)=\frac{N_{tk}}{N_t}$。其中:$N_{tk}$代表簇类 k 中类别为 t 的样本数;N_k 代表聚类 k 中的样本数;N_t 代表类别 t 中的样本数。相应的 F-measure 为

$$\text{F-measure}(t,C_k)=\frac{\mathrm{Prec}(t,C_k)\cdot\mathrm{Rec}(t,C_k)}{\mathrm{Prec}(t,C_k)+\mathrm{Rec}(t,C_k)}$$

对于整个划分的 FMI 值为 $F(C)=\sum_{t\in T}\frac{N_t}{N}\max_{C_k\in C}(\text{F-measure}(t,C_k))$。FMI 的取值越大则算法越准确,而 Entropy 则相反。Entropy 是一种考察聚类纯度的度量指标。其计算方法如下:对于每一个聚类 C_k,其 Entropy 为 $E_k=-\sum_t p_{tk}\log(p_{tk})$。其中:$p_{tk}$代表聚类 k 中的样本属于类别 t 的概率。整个聚类结果的 Entropy 是所有聚类的 Entropy 的加权平均和:$E=-\sum_k(\frac{N_k}{N}\times E_j)$。其中:$N$ 代表数据集样本总数,N_k 代表聚类 k 中的样本数。

3.3　结果与分析

实验对 KA-APC、AP 和 K-means 三种算法分别在六个数据集上进行了测试,图 4 分别给出了三种算法在六个数据集上 20 次随机实验的平均 F-measure 和 Entropy 指标。表 2 是三种算法的平均运行时间,本文指建立相似度矩阵和聚类过程两个部分运行时间之和的平均。



F: FaceImage　C: Circles　S: Spirals　Ir: Iris　W: Wine　Io: Ionosphere

图4　聚类结果比较

表 2　运行时间列表　/s

算法	Iris	Ionosphere	Wine	Circles	Spirals	FaceImage
KA-APC	0.34	21.29	0.95	2.69	7.37	10.04
AP	0.18	38.76	0.64	2.81	9.33	16.87
K-means	0.28	25.61	0.73	1.13	2.85	19.56

从以上的实验结果可以得出如下结论:

a)KA-APC 算法性能优于其他两种算法。尤其是 Circles 和 Spirals 两个数据集上,KA-APC 算法具有明显优势;而对于数据集 Iris 和 Wine,三种算法的聚类效果差距不是很大。这是由于 Iris 和 Wine 这两个数据集具有较好的聚类结构,因此使用传统的聚类算法也能得到理想的聚类结果。其他四个数据集的结构相对较为复杂,有些数据点在欧氏距离测度下不可分,而本文的 KA-APC 算法根据数据集的特性所设计的核函数能够使得数据集在核空间最大限度可分,从而提高算法的精度。这说明 KA-APC 算法能够适应多种复杂结构数据集的聚类,具有更强的普适性。

b)对于结构单一的数据集,KA-APC 算法的运行时间比其他两种算法略长,但对于结构比较复杂、维度相对较高的数据集,KA-APC 算法要比另两种算法快得多。其主要原因在于 KA-APC 算法进行核变换时耗费的时间较长,但核空间中数据近似线性可分,从而大大减少了算法迭代次数,加快了聚类过程。这种优势在结构单一的数据集上表现得并不明显,甚至反倒增加了运算负荷,但随着数据集结构越来越复杂,当算法迭代所需时间远远大于核变换时间时,KA-APC 算法将表现出显著的优势。

4　结束语

本文对核聚类和聚类中的一致性假设进行了研究,通过核函数的方法将近邻传播聚类在欧氏距离下推广到了核空间的同时,又通过搜索数据集的流形分布来调整核参数,使得核空间的相似度矩阵更加接近数据集本身的聚类结构,提出了基于核自适应的近邻传播聚类算法。该算法克服了原有算法对数据集尺度敏感的缺点,拓宽了算法处理多种数据的能力。实验结果表明,在处理多尺度问题上,该算法的性 (下转第 1650 页)

(上接第 1647 页)能明显优于原有 AP 和 K-means 算法。下一步笔者将研究如何确定聚类的有效终止条件以动态调整 AP 聚类中的参数,从而进一步提高算法的精度。

参考文献:

[1] 孙吉贵, 刘杰, 赵连宇. 聚类算法研究[J]. 软件学报, 2008,19(1):48-61.

[2] GELBARD R, GOLDMAN O, SPIEGLER I. Investigating diversity of clustering methods: an empirical comparison[J]. Data & Knowledge Engineering, 2007, 63(1): 155-166.

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

[4] KANG J H, LERMAN K, PLANGPRASOPCHOK A. Analyzing microblogs with affinity propagation[C]//Proc of the 1st Workshop on Social Media Analytics. New York: ACM Press, 2010: 67-70.

[5] CHEN Yang, LORENZO B, SUN Feng-yue, *et al*. A fuzzy statistics based affinity propagation technique for clustering in multispectral images[J]. IEEE Trans on Geosciences and Remote Sensing, 2010, 48(6): 2647-2659.

[6] GIVONI E, FREY B J. A binary variable model for affinity propagation[J]. Neural Computation, 2009, 21(6): 1589-1600.

[7] 崔鹏, 张汝波. 基于核自调整进行半监督聚类[J]. 计算机应用研究, 2009, 26(5): 1719-1722.

[8] 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.

[9] BRRAND M. Charting a manifold[M]//Advances in Neural Information Processing Systems. Cambridge, MA: MIT Press, 2003.

[10] NG A Y, JORDAN M I, WEISS Y. On spectral clustering: analysis and an algorithm[M]//Advances in Neural Information Processing Systems. Cambridge, MA: MIT Press, 2002: 856-864.