基于局部和全局信息的正则化迭代聚类

许小龙,王士同

(江南大学数字媒体学院,江苏 无锡 214122)

[**摘要**] 聚类是一种高效的数据分析方法,经典的 K-means 算法只适用于类簇为凸形的数据集,谱聚类算法虽然避免了 K-means 的一些缺点,但相似度中的参数设置问题以及较高的计算、存储复杂度对聚类有所限制. 基于局部和全局信息的正则化迭代聚类,先取部分数据作为一个整体聚类,然后逐渐加入少量数据进行迭代求解. 该方法继承传统谱聚类的优点,充分利用局部正则化和全局正则化信息,通过迭代方式求解使较大规模数据聚类成为可能. 通过实验对比结果显示,该算法有良好的聚类效果.

[**关键词**] 凸形,谱聚类,局部正则化,全局正则化,迭代

[**中图分类号**]TP181 [**文献标志码**]A [**文章编号**]1001-4616(2014)03-0021-08

Iterative Clustering with Local and Global Regularization

Xu Xiaolong, Wang Shitong

(School of Digital Media, Jiangnan University, Wuxi 214122, China)

Abstract: Clustering is an efficient method of data analysis, K-means method is one of the most popular algorithms. The algorithm only works when the cluster of data is convex. Spectral clustering avoids the problems of K-means method, however, parameters settings in similarity calculation, complex calculation and storage complexity all constraint the effectiveness of spectral clustering. In this paper, Iterative clustering with local and global regularization is proposed. In this method, we conduct a cluster with a part of data, and then we add a small amount of remaining data gradually to the iterative calculation. The proposed method has the advantages of traditional spectral clustering, exploring both the local and global regularization, and achieve the effective clustering for Large-scale data by an iteration method. Experimental results on several data sets show the greater performance on the method.

Key words: convex, spectral clustering, local regularization, global regularization, iterative

聚类是数据挖掘和机器学习领域一个最基本的研究课题[1],其目标是把数据划分为一组组相似的对象,从机器学习的角度看,聚类采用无监督的方式来学习数据集的隐藏模式,这些模式被定义为数据元. 从实际应用的角度来看,聚类在数据挖掘应用中扮演了一个重要的角色,比如在科学信息检索、文本挖掘、网络分析、市场营销、计算生物学等其他许多领域的应用[2].

时至今日,前人已经提出了许多的聚类方法,其中最有名的就是 K-means 算法[3],它的目的是使每个数据点到对应聚类中心的距离平方之和最小. K-means 算法能对大型数据集进行高效分类,然而众所周知 K-means 算法存在一些不足之处:①只适用于聚类结果为凸形(即类簇为凸形)的数据集,当样本集空间结构非凸时,算法就会陷入局部最优,从而使得算法聚类的质量无法得到保证. ②目标函数最小化求解得到的结果过于依赖初始值. 过去的几十年,已经提出许多方法[4,5]来克服上述问题.

近年,另一种有效的方法——基于数据图的聚类方法,在机器学习和数据挖掘领域引起了广泛关注,这种方法的基本思想是首先对整个数据集进行建模,从而形成一个加权图,图中的节点表示数据点,边缘的权重对应成对点之间的相似度,然后通过优化图中所定义的一些标准来获得数据集的集群分配,谱聚类是一种最具代表性的基于数据图的聚类方法,目的是优化定义在无向图中的一些切值[6](如 Ratio Cut[7]、

收稿日期:2014-02-15.

基金项目:江苏省自然科学基金(BK2011417).

通讯联系人:王士同,教授,研究方向:人工智能、模式识别、生物信息. E-mail:wxwangst@aliyun.com

min-max Cut[8]),经过一些放宽后,这些标准通常可以通过特征分解进行优化,并能保证这些解全局最优,因而可适用于非凸结构的数据集. 这种情况下谱聚类有效避免了传统 K-means 方法的一些缺点,但是这类方法仍存在一些不足,例如这些方法都是建立在相似度矩阵之上,而相似度矩阵的构造涉及许多参数的精确设置问题,其较高的计算复杂度和存储复杂度也限制它在大规模聚类问题上的应用.

针对上述问题,本文采取一种新的聚类方法. 它继承了谱聚类的优点,即最终的结果可以通过对一个对称矩阵进行特征分解得到,因此在非凸形数据集也能进行优化,并能保证这些解全局最优. 而与传统谱聚类不同的是,这只是在整个数据流型上执行一个平滑约束的数据标签[9]. 该方法先针对一部分数据点的近邻点进行正则化线性标签预测器的构造,然后把局部预测标签的结果和全局光滑正则化预测标签合并起来[10],在此基础上依次添加数据点进行迭代来预测标签,我们称之为局部和全局正则化迭代聚类,它有效避免了谱聚类的一些高计算复杂度和存储复杂度问题,在一些标准数据集上的实验结果表明了所提方法的有效性.

表 1　算法主要用到的符号

Table 1　The main symbols used in the algorithm

n	数据点的个数
$\boldsymbol{X}$	数据矩阵,$\boldsymbol{X}=[\boldsymbol{x}_1,\boldsymbol{x}_2,\cdots,\boldsymbol{x}_N]$
$\boldsymbol{N}_i$	$\boldsymbol{x}_i$ 的近邻点
n_i	$\boldsymbol{N}_i$ 矩阵的基数
$\boldsymbol{X}_i$	$\boldsymbol{x}_i$ 的最近邻点构成的稀疏矩阵
$\boldsymbol{L}$	拉普拉斯算子矩阵

1　问题陈述

在一个聚类问题中,给定 n 个数据点 $\boldsymbol{x}_1,\cdots,\boldsymbol{x}_n$,和一个正整数 C,聚类的目的就是把给定的数据集 $\boldsymbol{\chi}=\{\boldsymbol{x}_i\}_{i=1}^n$ 划分为 C 类,使不同种类的数据在某种意义上有明显的区别.

在数学上,聚类算法的结果可以表示为一个集群分配指示矩阵 $\boldsymbol{P}_{n\times C}$,如果 $\boldsymbol{x}_i$ 属于第 j 类,则 $\boldsymbol{P}_{ij}=1$,否则 $\boldsymbol{P}_{ij}=0$. 这样 $\boldsymbol{P}$ 矩阵的每一行只有一个 1,其余的元素都为 0.

正如多级谱聚类一样[11],我们不能直接地解出矩阵 $\boldsymbol{P}$,在本文中我们解决的是一个按比例缩小的集群分配指示矩阵 $\boldsymbol{Q}_{n\times C}$,使得 $\boldsymbol{Q}_{ij}=\boldsymbol{P}_{ij}/\sqrt{n_j}$,则:

$$\boldsymbol{Q}=\boldsymbol{P}(\boldsymbol{P}^{\mathrm{T}}\boldsymbol{P})^{-\frac{1}{2}}. \tag{1}$$

因此 $\boldsymbol{Q}$ 是一个半正交矩阵:

$$\boldsymbol{Q}^{\mathrm{T}}\boldsymbol{Q}=(\boldsymbol{P}^{\mathrm{T}}\boldsymbol{P})^{-\frac{1}{2}}(\boldsymbol{P}^{\mathrm{T}}\boldsymbol{P})(\boldsymbol{P}^{\mathrm{T}}\boldsymbol{P})^{-\frac{1}{2}}=\boldsymbol{I}. \tag{2}$$

其中 $\boldsymbol{I}$ 是一个 $n\times n$ 的单位矩阵,在下文中 $\boldsymbol{Q}$ 记作:

$$\boldsymbol{Q}=[\boldsymbol{q}^1,\boldsymbol{q}^2,\cdots,\boldsymbol{q}^C]. \tag{3}$$

此处 $\boldsymbol{q}^i(1\leqslant i\leqslant C)$ 对应 $\boldsymbol{Q}$ 矩阵的第 i 行,$\boldsymbol{q}_{ij}$ 可以视为 $\boldsymbol{x}_i$ 属于第 j 类的置信度.

2　局部和全局正则化迭代聚类

2.1　正则化线性分类器

传统的机器学习方法主要可以分为两类:监督学习和无监督学习. 对于无监督学习,我们面对的是不带标签的数据集,目的是用一种合理有效的方式来划分它们. 而监督学习可以描述成一个函数估计问题,旨在从标签训练数据集得到一个较好的分类函数,该分类函数能够在成本最小的情况下利用已标记数据对未知的测试数据进行标签预测[12]. 最小二乘拟合线性分类是一个最简单的监督学习分类方法,目的是学习一个列向量 $\boldsymbol{w}$ 使得它的平方的成本最小化:

$$\boldsymbol{J}'=\frac{1}{n}\sum_i(\boldsymbol{w}^{\mathrm{T}}\boldsymbol{x}_i-\boldsymbol{y}_i)^2. \tag{4}$$

这里 $\boldsymbol{y}_i$ 是点 $\boldsymbol{x}_i$ 的标签,通过求偏导并且使其值为 0,即 $\partial\boldsymbol{J}/\partial\boldsymbol{w}=0$,可以得到解:

$$\boldsymbol{w}^*=(\boldsymbol{X}\boldsymbol{X}^{\mathrm{T}})^{-1}\boldsymbol{X}\boldsymbol{y}. \tag{5}$$

其中 $\boldsymbol{X}=[\boldsymbol{x}_1,\boldsymbol{x}_2,\cdots,\boldsymbol{x}_n]$ 是一个 $m\times n$ 的数据矩阵,$\boldsymbol{y}=[\boldsymbol{y}_1,\boldsymbol{y}_2,\cdots,\boldsymbol{y}_n]^{\mathrm{T}}$ 是标签向量. 对于两类的问题 $\boldsymbol{y}_i\in\{+1,-1\}$,我们可以通过下式得到测试点 $\boldsymbol{x}_u$ 的标签:

$$l=\mathrm{sign}(\boldsymbol{w}^{*T}\boldsymbol{x}_u). \tag{6}$$

此处 sign(·)是一个符号函数. 对于多类(例如 C 类)问题,我们可以通过最小化式(7)为每一类构建

一个分类器:

$$\boldsymbol{J}^{c\prime}=\frac{1}{n}\sum_{i}((\boldsymbol{w}^{c})^{\mathrm{T}}\boldsymbol{x}_{i}-(\boldsymbol{y}^{c})_{i})^{2}. \tag{7}$$

其中 $1\leqslant c\leqslant C$,如果 $\boldsymbol{x}_i$ 属于第 c 类,则 $(\boldsymbol{y}^c)_i=1$,否则 $(\boldsymbol{y}^c)_i=0$. 则一个总数为 c 类的分类器的法向量变为:

$$\boldsymbol{w}^{c*}=(\boldsymbol{X}\boldsymbol{X}^{\mathrm{T}})^{-1}\boldsymbol{X}\boldsymbol{y}^{c}. \tag{8}$$

测试点 $\boldsymbol{x}_u$ 的标签可以通过下面的式子得到:

$$c=\mathrm{argmax}_{c}((\boldsymbol{w}^{c*})^{\mathrm{T}}\boldsymbol{x}_{u}). \tag{9}$$

为了避免 $\boldsymbol{X}\boldsymbol{X}^{\mathrm{T}}$ 的奇异性,我们可以为 c 类分类器标准添加一个正则化项并且使其最小化:

$$\boldsymbol{J}^{c}=\frac{1}{n}\sum_{i=1}^{n}((\boldsymbol{w}^{c})^{\mathrm{T}}\boldsymbol{x}_{i}-\boldsymbol{y}_{i})^{2}+\lambda_{c}\parallel\boldsymbol{w}^{c}\parallel^{2}. \tag{10}$$

这里的 λ_c 是一个正则化参数,此时 $\boldsymbol{J}_c$ 的最优解就变成了:

$$\boldsymbol{w}^{c*}=(\boldsymbol{X}\boldsymbol{X}^{\mathrm{T}}+\lambda n\boldsymbol{I})^{-1}\boldsymbol{X}\boldsymbol{y}^{c}. \tag{11}$$

其中 $\boldsymbol{I}$ 是一个 $m\times m$ 的单位矩阵. 这就是我们通常见到的正则化线性分类器.

类似于绝大部分监督学习方法,正则化线性分类器是一种全局分类,它使用整个训练集进行训练分类. 然而,文献[12]指出,有时候很难找出一个足够好的线性分类器来预测整个输入数据空间的标签. 为了获得更好的预测标签,文献[13]发现对于特定的任务,局部训练的分类器对于测试数据集可以得到一个较好的预测效果.

2.2 局部正则化

我们将局部学习算法引入到聚类问题中,其基本的思想是,基于它的近邻点 N_i(k-nearest 近邻或者 ε 近邻)[13] 为每个数据点 $\boldsymbol{x}_i(1\leqslant i\leqslant n)$ 训练一个局部标签[14],并用它来预测 $\boldsymbol{x}_i$ 的标签,然后通过最小化预测误差之和来合并局部预测.

由于其简单及有效性,我们选用正则化线性分类器来预测局部标签,对于每个数据点 $\boldsymbol{x}_i$,目的是得到一个 $\boldsymbol{w}_i$ 使得下列式子最小化:

$$\boldsymbol{J}_{i}^{c}=\frac{1}{n_{i}}\sum_{\boldsymbol{x}_{j}\in\boldsymbol{N}_{i}}\parallel(\boldsymbol{w}_{i}^{c})^{\mathrm{T}}\boldsymbol{x}_{j}-(\boldsymbol{q}^{c})_{j}\parallel^{2}+\lambda_{i}\parallel\boldsymbol{w}_{i}^{c}\parallel^{2}. \tag{12}$$

此处的 $n_i=|\boldsymbol{N}_i|$ 是 $\boldsymbol{N}_i$ 的基数,$(\boldsymbol{q}^c)_j$ 是 $\boldsymbol{x}_j$ 属于第 c 类的置信度. 从等式(11)可以得到一个优化解:

$$\boldsymbol{w}_{i}^{c*}=(\boldsymbol{X}_{i}\boldsymbol{X}_{i}^{\mathrm{T}}+\lambda_{i}n_{i}\boldsymbol{I})^{-1}\boldsymbol{X}_{i}\boldsymbol{q}_{i}^{c}(1\leqslant c\leqslant C). \tag{13}$$

其中 $\boldsymbol{X}_i=[\boldsymbol{x}_{i1},\boldsymbol{x}_{i2},\cdots,\boldsymbol{x}_{in_i}]$,$\boldsymbol{x}_{ik}$ 是 $\boldsymbol{x}_i$ 的第 k 个最近邻点,每个近邻点都在输入矩阵 $\boldsymbol{X}$ 中原本的位置,其他位置的点均为 0,它是 $\boldsymbol{x}_i$ 的 k 个最近邻点组成的和输入数据 $\boldsymbol{X}$ 等大小的稀疏矩阵,$\boldsymbol{q}_i^c=[\boldsymbol{q}_{i1}^c,\boldsymbol{q}_{i2}^c,\cdots,\boldsymbol{q}_{in_i}^c]^{\mathrm{T}}$,其中 $\boldsymbol{q}_{ik}^c=\boldsymbol{q}^c(x_{ik})$. 等式(13)可以通过 Woodbury formula[15] 得到:

$$\boldsymbol{w}_{i}^{c*}=\boldsymbol{X}_{i}(\boldsymbol{X}_{i}^{\mathrm{T}}\boldsymbol{X}_{i}+\lambda_{i}n_{i}\boldsymbol{I}_{i})^{-1}\boldsymbol{q}_{i}^{c}(1\leqslant c\leqslant C). \tag{14}$$

对于一个落入 $\boldsymbol{N}_i$ 中的新的测试点 $\boldsymbol{u}$,我们可以通过预测它属于第 c 类的置信度:

$$\boldsymbol{q}_{u}^{c}=(\boldsymbol{w}_{i}^{c*})^{\mathrm{T}}\boldsymbol{u}=\boldsymbol{u}^{\mathrm{T}}\boldsymbol{w}_{i}^{c*}=\boldsymbol{u}^{\mathrm{T}}\boldsymbol{X}_{i}(\boldsymbol{X}_{i}^{\mathrm{T}}\boldsymbol{X}_{i}+\lambda_{i}n_{i}\boldsymbol{I}_{i})^{-1}\boldsymbol{q}_{i}^{c}.$$

所有局部预测构成之后,我们可以通过最小化预测误差之和来合并它们:

$$\boldsymbol{J}_{l}=\sum_{c=1}^{C}\sum_{i=1}^{n}((\boldsymbol{w}_{i}^{c*})^{\mathrm{T}}\boldsymbol{x}_{i}-\boldsymbol{q}_{i}^{c})^{2}. \tag{15}$$

合并(11)和(15)得到:

$$\boldsymbol{J}_{l}=\sum_{c=1}^{C}\sum_{i=1}^{n}((\boldsymbol{w}_{i}^{c*})^{\mathrm{T}}\boldsymbol{x}_{i}-\boldsymbol{q}_{i}^{c})^{2}=\sum_{c=1}^{C}\sum_{i=1}^{n}(\boldsymbol{x}_{i}^{\mathrm{T}}\boldsymbol{X}_{i}(\boldsymbol{X}_{i}^{\mathrm{T}}\boldsymbol{X}_{i}+\lambda_{i}n_{i}\boldsymbol{I}_{i})^{-1}\boldsymbol{q}_{i}^{c}-\boldsymbol{q}_{i}^{c})^{2}=\sum_{c=1}^{C}\parallel\boldsymbol{G}\boldsymbol{q}^{c}-\boldsymbol{q}^{c}\parallel^{2}. \tag{16}$$

这里 $\boldsymbol{q}^c=[\boldsymbol{q}_1^c,\boldsymbol{q}_2^c,\cdots,\boldsymbol{q}_n^c]^{\mathrm{T}}$,而 $\boldsymbol{G}$ 是一个 $n\times n$ 的矩阵,它的第(i,j)项为:

$$\boldsymbol{G}_{ij}=\begin{cases}\boldsymbol{\alpha}_{j}^{i}, & \text{if } x_{j}\in\boldsymbol{N}_{i},\\ 0, & \text{otherwise.}\end{cases} \tag{17}$$

此处 $\boldsymbol{\alpha}_j^i$ 代表 $\boldsymbol{\alpha}^i$ 第 j 项:

$$\boldsymbol{\alpha}^{i}=\boldsymbol{x}_{i}^{\mathrm{T}}\boldsymbol{X}_{i}(\boldsymbol{X}_{i}^{\mathrm{T}}\boldsymbol{X}_{i}+\lambda_{i}n_{i}\boldsymbol{I}_{i})^{-1}. \tag{18}$$

到目前为止,我们构建所有局部正则化线性标签预测并将它们结合在一个可以写成明确数学表达式

的成本函数中,使用优化技术可以有效地使该函数最小化. 然而,因为我们只使用了数据集的局部信息,结果可能不是足够好,在下面的小节我们介绍一个全局正则化标准,并把它和 J_l 结合起来,目的是从局部和全局的方法中得到一个较好的聚类效果.

2.3 全局正则化

一个通常可以引导学习过程的假设是集群假设[10],它表明:

(1)邻近的点往往具有相似的集群分配;

(2)相同结构的点(如流型或集群)常常有类似的集群分配.

换句话说,群集中的假设意味着,就数据集的内在结构而言,数据集的标签应该平稳地变化. 数据标签向量的平滑性可以通过下式测量:

$$\boldsymbol{J}_g=\sum_{c=1}^{C}(\boldsymbol{q}^c)^{\mathrm{T}}\boldsymbol{L}\boldsymbol{q}^c=\sum_{c=1}^{C}\sum_{i=1}^{n}(\boldsymbol{q}_i^c-\boldsymbol{q}_j^c)^2\boldsymbol{w}_{ij}. \tag{19}$$

$\boldsymbol{L}$ 是一个 $n\times n$ 的矩阵,它的第(i,j)项是

$$\boldsymbol{L}_{ij}=\begin{cases}\boldsymbol{d}_i-\boldsymbol{w}_{ii}, & i=j,\\ -\boldsymbol{w}_{ij}, & i\neq j.\end{cases} \tag{20}$$

其中 $\boldsymbol{d}_i=\sum \boldsymbol{w}_{ij}$,$\boldsymbol{w}_{ij}$是 x_i 和 x_j 之间的相似度,有很多不同的方法求 $\boldsymbol{w}_{ij}$,下面列出几种:

(1)未加权 k 最近邻相似[16]:如果 $\boldsymbol{x}_i$ 是 $\boldsymbol{x}_j$ 的 k 最近邻或者 $\boldsymbol{x}_j$ 是 $\boldsymbol{x}_i$ 的 k 最近邻时,它们之间的相似度为 1,否则为 0,k 是控制这种相似度的唯一超参数,这种相似度具有很好的“自适应尺度”属性[17],因为在低密度和高密度区域成对点之间的相似度是相同的.

(2)未加权 ε-Ball 近邻相似度[17]:对于距离函数 $d(\cdot)$,如果 $d(\boldsymbol{x}_i,\boldsymbol{x}_j)\leqslant\varepsilon$,则它们之间的相似度为 1,$\varepsilon$ 是控制这种连续相似度的唯一超参数.

(3)加权双曲正切相似[17]:$\boldsymbol{d}_{ij}$是 $\boldsymbol{x}_i$ 和 $\boldsymbol{x}_j$ 之间的距离,点 $\boldsymbol{x}_i$ 点 $\boldsymbol{x}_j$ 之间的双曲正切相似度为:$\boldsymbol{w}_{ij}=\frac{1}{2}(\tanh(\boldsymbol{\alpha}_1(\boldsymbol{d}_{ij}-\boldsymbol{\alpha}_2))+1)$.

直接创建一个软截止周长 $\boldsymbol{\alpha}_2$,这样相似的样本(大概来自同一类)具有较高的相似度,而不同的样本(可能来自不同类)具有较低的相似度,$\boldsymbol{\alpha}_1$ 和 $\boldsymbol{\alpha}_2$ 分别控制双曲正切相似的斜率和截值.

(4)加权指数相似[6,9,17]:$\boldsymbol{d}_{ij}$是 $\boldsymbol{x}_i$ 和 $\boldsymbol{x}_j$ 之间的距离,则 $\boldsymbol{x}_i$ 和 $\boldsymbol{x}_j$ 之间的相似度为:

$$\boldsymbol{w}_{ij}=\exp\left(-\frac{\boldsymbol{d}_{ij}^2}{\sigma}\right). \tag{21}$$

这是由 σ 控制衰减率的一个连续加权方案.

本文中,我们选用加权指数相似度,由于:(1)该方法简单易行并广泛应用于许多领域. (2)在一定条件下,$\boldsymbol{w}_{ij}$的形式决定了拉普拉斯算子对拉普拉斯贝尔特拉米算子加权图边缘收敛[16,18]. 使用欧几里德距离计算 $\boldsymbol{d}_{ij}$.

2.4 局部和全局正则化

将上面介绍到的局部和全局正则化标准结合到一起[19],可以得到下式:

$$\begin{aligned}&\min_q\boldsymbol{J}=\boldsymbol{J}_l+\lambda\boldsymbol{J}_g=\sum_{c=1}^{C}(\|\boldsymbol{G}\boldsymbol{q}^c-\boldsymbol{q}^c\|^2+\lambda(\boldsymbol{q}^c)^{\mathrm{T}}\boldsymbol{L}\boldsymbol{q}^c),\\ &\text{s.t.}\quad \boldsymbol{Q}^{\mathrm{T}}\boldsymbol{Q}=\boldsymbol{I}.\end{aligned} \tag{22}$$

其中 $\boldsymbol{G}$ 在(17)中已经定义过,λ 是权衡 $\boldsymbol{J}_l$ 和 $\boldsymbol{J}_g$ 的一个正实参数. $\boldsymbol{Q}=[q^1,q^2,\cdots,q^C]$,注意我们已经放宽了对 $\boldsymbol{Q}$ 的限制,使得它仅需要满足半正交性的约束关系. 然后使得下式最小化:

$$\begin{aligned}\boldsymbol{J}=\boldsymbol{J}_l+\lambda\boldsymbol{J}_g&=\sum_{c=1}^{C}[\|\boldsymbol{G}\boldsymbol{q}^c-\boldsymbol{q}^c\|^2+\lambda(\boldsymbol{q}^c)^{\mathrm{T}}\boldsymbol{L}\boldsymbol{q}^c]=\sum_{c=1}^{C}[(\boldsymbol{q}^c)^{\mathrm{T}}((\boldsymbol{G}-\boldsymbol{I})^{\mathrm{T}}(\boldsymbol{G}-\boldsymbol{I})+\lambda\boldsymbol{L})\boldsymbol{q}^c]=\\ &\quad\operatorname{trace}[\boldsymbol{Q}^{\mathrm{T}}((\boldsymbol{G}-\boldsymbol{I})^{\mathrm{T}}(\boldsymbol{G}-\boldsymbol{I})+\lambda\boldsymbol{L})\boldsymbol{Q}].\end{aligned} \tag{23}$$

因此我们只需要解决下列优化问题

$$\begin{aligned}&\min_Q\boldsymbol{J}=\operatorname{trace}[\boldsymbol{Q}^{\mathrm{T}}((\boldsymbol{G}-\boldsymbol{I})^{\mathrm{T}}(\boldsymbol{G}-\boldsymbol{I})+\lambda\boldsymbol{L})\boldsymbol{Q}],\\ &\text{s.t.}\quad \boldsymbol{Q}^{\mathrm{T}}\boldsymbol{Q}=\boldsymbol{I}.\end{aligned} \tag{24}$$

通过 KyFan 定理[5],可以得到上述问题的最优解:

$$\boldsymbol{Q}^{*}=[\boldsymbol{q}_1^{*},\boldsymbol{q}_2^{*},\cdots,\boldsymbol{q}_C^{*}]\boldsymbol{R}. \tag{25}$$

此处的 $\boldsymbol{q}_k^{*}$ $(1\leqslant k\leqslant C)$ 是对称矩阵 $(\boldsymbol{G}-\boldsymbol{I})^{\mathrm{T}}(\boldsymbol{G}-\boldsymbol{I})+\lambda\boldsymbol{L}$ 的第 k 个最小特征值对应的特征向量,$\boldsymbol{R}$ 是一个任意的 $C\times C$ 矩阵,因此上述问题的最优解是不唯一的,这是通常被称为格拉斯曼流型的一个子空间[20],我们需要找到一个按比例缩小的集群分配指示矩阵 $\boldsymbol{Q}^{*}$ 和一个旋转矩阵 $\boldsymbol{R}$,使得 $\boldsymbol{Q}^{*}\boldsymbol{R}$ 接近一个真正的离散扩展分配指示矩阵[21],在这种条件下,由此产生的集群分配指示矩阵 $\boldsymbol{P}$ 接近真正的离散分配指示矩阵,为了达到这种效果,在我们实验中参照多级谱聚类[11]中采用的方法.

2.5 局部和全局正则化迭代聚类

针对大规模数据的聚类问题,我们可以通过逐渐增加样本点的方法来解决. 一个较大的数据集当中,在样本点逐渐增加的情况下,可视作先用局部和全局聚类的方法预测一个 $m\times u$(维数为 m,点的个数为 u)的数据矩阵,得到近邻矩阵和 $\boldsymbol{G}$ 矩阵,通过和全局正则化结合预测出聚类的标签得到聚类效果. 同一种分布中在前 u 个数据点的基础上增加了 v 个点的时候,我们假定 u 个点的 k 个最近邻点保持不变,新增加的 v 个点在总共 $u+v$ 个点中求得每个点的 k 个最近邻点组成的矩阵. 由(17)可知 $\boldsymbol{G}$ 矩阵求法为 $\boldsymbol{\alpha}^{i}=\boldsymbol{x}_i^{\mathrm{T}}\boldsymbol{X}_i(\boldsymbol{X}_i^{\mathrm{T}}\boldsymbol{X}_i+\lambda_i n_i\boldsymbol{I}_i)^{-1}$. 当点的个数从 u 增加到 v 的时候,新的聚类中 $\boldsymbol{G}'$矩阵为$(u+v)\times(u+v)$的方阵. 此时的 $\boldsymbol{G}'$ 求法如下:

因为假定增加 v 个点之后,前 u 个点的近邻点不发生变化,即此时前 u 个点中某一个点的 k 近邻矩阵为 $\boldsymbol{X}_i'$,矩阵 $\boldsymbol{X}_i'=[\boldsymbol{X}_i\quad\boldsymbol{O}]$($\boldsymbol{X}_i'$是由 $\boldsymbol{X}_i$ 和 $\boldsymbol{O}$ 组成的矩阵),$\boldsymbol{O}$ 是 $m\times u$ 的全 0 矩阵,此时

$$\boldsymbol{X}'^{\mathrm{T}}{}_i\boldsymbol{X}_i'=\begin{bmatrix}\boldsymbol{X}_i^{\mathrm{T}}\\ \boldsymbol{O}^{\mathrm{T}}\end{bmatrix}\cdot[\boldsymbol{X}_i\quad\boldsymbol{O}]=\begin{bmatrix}\boldsymbol{X}_i^{\mathrm{T}}\boldsymbol{X}_i & \boldsymbol{X}_i^{\mathrm{T}}*\boldsymbol{O}\\ \boldsymbol{O}^{\mathrm{T}}*\boldsymbol{X}_i & \boldsymbol{O}^{\mathrm{T}}*\boldsymbol{O}\end{bmatrix}=\begin{bmatrix}\boldsymbol{X}_i^{\mathrm{T}}\boldsymbol{X}_i & \boldsymbol{O}\\ \boldsymbol{O}^{\mathrm{T}} & \boldsymbol{O}\end{bmatrix},$$

$$\boldsymbol{\alpha}^{i}=\boldsymbol{x}_i^{\mathrm{T}}\boldsymbol{X}_i'(\boldsymbol{X}'^{\mathrm{T}}{}_i\boldsymbol{X}_i'+\lambda_i n_i\boldsymbol{I}_i')^{-1}=\boldsymbol{x}_i^{\mathrm{T}}\boldsymbol{X}_i'\begin{bmatrix}\boldsymbol{X}_i^{\mathrm{T}}\boldsymbol{X}_i+\lambda_i n_i\boldsymbol{I}_i & \boldsymbol{O}\\ \boldsymbol{O}^{\mathrm{T}} & \lambda_i n_i\boldsymbol{I}\end{bmatrix}^{-1}=\boldsymbol{x}_i^{\mathrm{T}}[\boldsymbol{X}_i\quad\boldsymbol{O}]\begin{bmatrix}(\boldsymbol{X}_i^{\mathrm{T}}\boldsymbol{X}_i+\lambda_i n_i\boldsymbol{I}_i)^{-1} & \boldsymbol{O}\\ \boldsymbol{O}^{\mathrm{T}} & (\lambda_i n_i\boldsymbol{I})^{-1}\end{bmatrix}=$$

$$\boldsymbol{x}_i^{\mathrm{T}}[\boldsymbol{X}_i(\boldsymbol{X}_i^{\mathrm{T}}\boldsymbol{X}_i+\lambda_i n_i\boldsymbol{I}_i)^{-1}\quad\boldsymbol{O}]=[\boldsymbol{\alpha}^{i}\quad\boldsymbol{O}]. \tag{26}$$

上面式子中的 $\boldsymbol{O}$ 均为满足运算的全 0 矩阵或向量,结果 $\boldsymbol{\alpha}^{i\prime}=[\boldsymbol{\alpha}^{i}\quad\boldsymbol{O}]$ 中的 $\boldsymbol{O}$ 是 $1\times v$ 的全 0 向量. 因此对于增加了 v 个点的矩阵,计算 $\boldsymbol{G}'$的方法为,前 u 个点直接调用它们的 $\boldsymbol{G}(u\times u)$,后 v 个点在总共 $u+v$ 个点中计算:

$$\boldsymbol{\alpha}^{i}=\boldsymbol{x}_i^{\mathrm{T}}\boldsymbol{X}_i(\boldsymbol{X}_i^{\mathrm{T}}\boldsymbol{X}_i+\lambda_i n_i\boldsymbol{I}_i)^{-1}\ (u<i\leqslant u+v),$$

$$\boldsymbol{G}'=\begin{bmatrix}\boldsymbol{G} & \boldsymbol{O}\\ \boldsymbol{x}_i^{\mathrm{T}} & \boldsymbol{X}_i(\boldsymbol{X}_i^{\mathrm{T}}\boldsymbol{X}_i+\lambda_i n_i\boldsymbol{I}_i')^{-1}\end{bmatrix}. \tag{27}$$

$\boldsymbol{O}$ 是 $u\times v$ 的全 0 矩阵,此时构建一个和 $\boldsymbol{G}'$相同大小$(u+v)\times(u+v)$的拉普拉斯算子 $\boldsymbol{L}'$.

$$\min_{Q}\quad \boldsymbol{J}'=\mathrm{trace}[\boldsymbol{Q}'^{\mathrm{T}}((\boldsymbol{G}'-\boldsymbol{I}')^{\mathrm{T}}(\boldsymbol{G}'-\boldsymbol{I}')+\lambda\boldsymbol{L}')\boldsymbol{Q}'],$$

$$\text{s. t.}\quad \boldsymbol{Q}'^{\mathrm{T}}\boldsymbol{Q}'=\boldsymbol{I}. \tag{28}$$

然后通过对矩阵$(\boldsymbol{G}'-\boldsymbol{I}')^{\mathrm{T}}(\boldsymbol{G}'-\boldsymbol{I}')+\lambda\boldsymbol{L}'$进行特征分解,得到它的 k 个最小特征值,结合等式(25)中的 $\boldsymbol{R}$ 矩阵,可以得到 $u+v$ 个点的预测标签. 在一个较大的数据集中,可以视为先预测出 u 个较多部分点的标签,然后每次叠加进去 v 个点(不足 v 个点在最后一次全部叠加),可求得它们在 u 个点以及之后每次叠加 v 个点的每一个预测标签,再比较其聚类效果,即新得到的 u、$u+v$、$u+2v\cdots u+\tau v$ 个数据点的预测标签聚类效果变化都可以清楚地观察到.

表 2 本文算法主要步骤

Table 2 The main steps of the algorithm

输入:

1. 数据集 $\boldsymbol{\chi}=\{x_i\}_{i=1}^{n}$.
2. 聚类的数目 C.
3. 近邻的大小 K.
4. 局部正则化参数 $\{\lambda_i\}_{i=1}^{n}$.
5. 全局正则化参数 λ.

输出:

每个数据点对应的聚类关系

过程:

1. 为每个数据点找出它距离最近的 K 个近邻点.
2. 用等式(17)求出矩阵 $\boldsymbol{G}$.
3. 构建式(20)中的拉普拉斯矩阵.
4. 特征矩阵 $\boldsymbol{M}=(\boldsymbol{G}-\boldsymbol{I})^{\mathrm{T}}(\boldsymbol{G}-\boldsymbol{I})+\lambda\boldsymbol{L}$.
5. 通过(27)及 $\boldsymbol{G}$ 得出迭代的 $\boldsymbol{G}'$.
6. 迭代后 $\boldsymbol{M}'=(\boldsymbol{G}'-\boldsymbol{I}')^{\mathrm{T}}(\boldsymbol{G}'-\boldsymbol{I}')+\lambda\boldsymbol{L}'$.
7. 对 $\boldsymbol{M}$ 及迭代后 $\boldsymbol{M}'$进行特征分解,并用式(25)求得矩阵 $\boldsymbol{Q}^{*}$.
8. 正确地离散 $\boldsymbol{Q}^{*}$ 矩阵得到每个数据点的聚类分配.

从 $\boldsymbol{G}'$矩阵的求法中可以看出,进行迭代的次数越多,求逆矩阵节省的计算复杂度就越多,整个算法的计算复杂度和存储复杂度相对大大降低,相比于谱聚类算法中异常复杂的一些参数设置问题,也较好地体

现其有效性,同时保证了这些解全局最优,这样对于较大的数据集,也能进行比较良好的聚类.

3　实验与分析

3.1 实验条件与数据介绍

本次实验均在 MATLAB 7.10.0 平台下完成,实验环境为 CPU Intel Core(TM)i3-3240 3.40 GHz,内存 4 G,这一小节对局部和全局信息正则化迭代聚类算法和其他一些聚类方法在几个数据集上进行比较,先介绍一下数据集的基本信息.

我们使用 3 个真实数据集来测评这种方法的有效性,表 3 介绍了数据集的特点和大小.

表 3　数据集的大小

Table 3　Description of the datasets

数据集	大小	维度	类数
USPS	9 298	256	10
letter	5 000	16	26
waveform	5 000	21	3

USPS 是著名的手写数字数据集,它是常用的数据挖掘测试数据集,包含了 0 ~9 总共 10 个不同的数字. Letter 是 UCI 中的数据集,它的 26 个类别对应了从 A ~Z 的 26 个不同的字母. Waveform 也是 UCI 数据库中的数据集,它的 3 类代表了 3 种不同的波形. 几个数据集的大小在表 3 中列出.

3.2　评价标准

实验中,我们在所有的聚类算法设置集群的数目等于真实数目 C 类,为了评价它们的性能,通过计算下列 2 个测试来比较聚类和真实的类之间的效果.

聚类精度:第一个评价指标是聚类精度,它找出了聚类和真实类之间一一对应的关系,并测出了每个聚类对应真实的类包含的数据点,它总结了所有聚类之间的匹配度,聚类精度通过下式计算:

$$Acc=\frac{1}{N}\max\left(\sum_{C_k,L_m}T(\boldsymbol{C}_k,\boldsymbol{L}_m)\right). \tag{29}$$

其中 $\boldsymbol{C}_k$ 表示最终结果的第 k 类,$\boldsymbol{L}_m$ 是真正的第 m 类,$T(\boldsymbol{C}_k,\boldsymbol{L}_m)$ 是属于 m 类被分配到聚类 k 的数据点的数量,精度对所有集群和类计算 $T(\boldsymbol{C}_k,\boldsymbol{L}_m)$ 总和的最大值,所有这些成对集群和类没有重叠,更高的聚类精度意味着更好的聚类效果.

归一化互信息(Normalized Mutual Information,NMI):使用的另外一个聚类评价标准是归一化互信息[17],它被广泛应用于聚类的效果评价,对于两个随机的标签 $\boldsymbol{X}$、$\boldsymbol{Y}$,他们的 NMI 定义为:

$$\mathrm{NMI}(\boldsymbol{X},\boldsymbol{Y})=\frac{I(\boldsymbol{X},\boldsymbol{Y})}{\sqrt{H(\boldsymbol{X})H(\boldsymbol{Y})}}. \tag{30}$$

$I(\boldsymbol{X},\boldsymbol{Y})$ 是 $\boldsymbol{X}$ 和 $\boldsymbol{Y}$ 之间的互信息,$H(\boldsymbol{X})$ 和 $H(\boldsymbol{Y})$ 分别是 $\boldsymbol{X}$ 和 $\boldsymbol{Y}$ 的熵,给定一个聚类结果,由(30)计算其 NMI 为:

$$\mathrm{NMI}(\boldsymbol{X},\boldsymbol{Y})=\frac{\sum_{k=1}^{C}\sum_{m=1}^{C}n_{k,m}\log\left(\frac{n\cdot n_{k,m}}{n_k\hat{n}_m}\right)}{\sqrt{\left(\sum_{k=1}^{C}n_k\log\frac{n_k}{n}\right)\left(\sum_{m=1}^{C}\hat{n}_m\log\frac{\hat{n}_m}{n}\right)}}. \tag{31}$$

其中 n_k 表示 $\boldsymbol{C}_k(1\leqslant k\leqslant C)$ 中所包含的内容,$\hat{n}_m$ 是数据中属于第 $m(1\leqslant m\leqslant C)$ 类的数量,$n_{k,m}$ 表示聚类 C_k 和 m 类交叉点的数目,用式(31)计算出的值来评价聚类结果,这个值越大,聚类效果越好.

对比和参数设置:

本文聚类方法和其他的方法做了一些对比,它们是 K-means(KM),Clustering with Pure Local Regularization(CPLR). 本文算法参数设置为:所有局部正则化参数 $\{\lambda_i\}_{i=1}^{n}$ 都从 $\{0.1,1,10\}$ 中选择,近邻数从 $\{20,40,80\}$ 中选择,全局正则化参数 λ 从 $\{0.1,1,10\}$ 中选择,且采用相同的离散化方法[11]. 正则化参数及近邻数选取不同,聚类的结果会略有浮动,但是变化不大,CPLR 算法选取效果最佳一组参数,本文算法在与 CPLR 算法相同参数的条件下进行聚类,得出各自聚类的效果并进行比较.

3.3　实验结果分析

实验对 USPS 数据集第一次预测 4 000 个点,然后依次迭加 1 000 个点,剩余部分最后一次全部加进去,letter 和 waveform 先预测 2 000 个数据点,然后依次迭加 500 个数据点,A1 和 N1 表示第一次预测数据

点的聚类效果,A7 和 N7 表示数据全部叠加完毕整个数据集的聚类精度和归一化互信息.

实验结果的聚类精度和归一化互信息分别如表 4 和表 5 所示.

表 4　聚类精度

Table 4　Clustering accuracy results

	USPS	letter	waveform
KM	0.633 1	0.272 0	0.509 2
CPLR	0.601 9	0.280 2	0.502 4
本文算法	A1 = 0.676 5	A1 = 0.339 5	A1 = 0.519 0
	A2 = 0.682 6	A2 = 0.330 8	A2 = 0.517 2
	A3 = 0.683 5	A3 = 0.329 3	A3 = 0.510 0
	A4 = 0.681 7	A4 = 0.313 4	A4 = 0.517 1
	A5 = 0.630 5	A5 = 0.310 0	A5 = 0.508 0
	A6 = 0.677 9	A6 = 0.316 0	A6 = 0.506 0
	A7 = 0.675 8	A7 = 0.319 2	A7 = 0.503 0

表 5　归一化互信息

Table 5　Normalized mutual information results

	USPS	letter	waveform
KM	0.608 6	0.370 6	0.364 8
CPLR	0.585 6	0.403 5	0.363 9
本文算法	N1 = 0.653 1	N1 = 0.450 9	N1 = 0.366 6
	N2 = 0.655 4	N2 = 0.438 7	N2 = 0.367 3
	N3 = 0.649 2	N3 = 0.429 0	N3 = 0.367 7
	N4 = 0.644 5	N4 = 0.419 9	N4 = 0.364 6
	N5 = 0.627 9	N5 = 0.416 6	N5 = 0.364 4
	N6 = 0.632 0	N6 = 0.420 1	N6 = 0.363 1
	N7 = 0.630 5	N7 = 0.418 0	N7 = 0.364 6

从上述实验结果可以看出,我们的聚类方法虽然随着样本点不断增加聚类效果会稍微有所降低,但是整个数据集迭代完毕之后的效果较之其他方法还是要优胜一些,而传统谱聚类对较大规模数据集聚类会出现存储瘫痪,我们的方法对于实验中这样大规模的数据集通过迭代能取得较好的聚类效果,且聚类所需时间也相对较少,能比较快地得到聚类的结果,体现本文算法相对传统谱聚类在存储及计算复杂度上面的优势.

4　结论与展望

本文中,我们采用了局部与全局正则化的聚类方法,对其进行了一些改进,并与其他方法进行了比较,实验结果表明,在线性空间,同一种分布中的样本点不大可分的情况下,随着数据点的逐步增加,聚类的效果会略微有所下降,因为在线性不可分的情况下重叠的点比较多的话,随着样本越来越大,重叠部分会越来越密而变得越来越不可分,这时候要找到一个聚类效果较好的线性分类器就更难,因此它的聚类效果也会稍有下降. 未来的期望主要是在核化上面,在线性不可分的情况下,选择一个核函数通过映射到高维空间,来解决在原始空间中线性不可分的问题,在这种情况下,数据点逐步地增加,信息量越来越多,而随着样本点的逐步增多,我们期望能够得到更好的聚类效果,这是我们未来主要的工作方向.

[参考文献]

[1] Jain A, Dubes R. Algorithms for Clustering Data[M]. NJ: Prentice-Hall, 1988.

[2] Han J, Kamber M. Data Mining: Concepts and Techniques[M]. San Francisco: Morgan Kaufmann Publishers, 2001.

[3] Duda R O, Hart P E, Stork D G. Pattern Classication[M]. USA: John Wiley and Sons, 2001.

[4] He J, Lan M, Tan C L, et al. Initialization of cluster renement algorithms: a review and comparative study[C]//Proceedings of IEEE International Joint Conference on Neural Networks. United States: IEEE Computer Society, 2004: 297-302.

[5] Zha H, He X, Ding C, et al. Spectral relaxation for K-means clustering[C]//Dietterich T G, Becker S, Ghahramani Z. Advances in Neural Information Processing Systems. USA: The MIT Press, 2001: 1 057-1 064.

[6] Shi J, Malik J. Normalized cuts and image segmentation[J]. IEEE Trans on Pattern Analysis and Machine Intelligence, 2000, 22(8): 888-905.

[7] Chan P K, Schlag D F, Zien J Y. Spectral K-way ratio-cut partitioning and clustering[J]. IEEE Trans Computer-Aided Design, 1994, 13(9): 1 088-1 096.

[8] Ding C, He X, Zha H, et al. A min-max cut algorithm for graph partitioning and data clustering[C]//Proceedings of the 1st International Conference on Data Mining(ICDM). California, USA: IEEE Computer Society, 2001: 107-114.

[9] Belkin M, Niyogi P. Laplacian eigenmaps for dimensionality reduction and data representation[J]. Neural Computation, 2003, 15(6): 1 373-1 396.

[10] Zhou D, Bousquet O, Lal T N, et al. Learning with local and global consistency[C]//Advances in Neural Information Processing Systems. Cambrige: MIT Press, 2003: 321-328.

[11] Yu S X, Shi J. Multiclass spectral clustering[C]//Proceedings of the International Conference on Computer Vision. USA:

IEEE,2003:313-319.
[12] Vapnik V N. The Nature of Statistical Learning Theory[M]. Berlin:Springer-Verlag,1995.
[13] Bottou L,Vapnik V. Local learning algorithms[J]. Neural Computation,1992,4(6):888-900.
[14] Wu M,Sch¨olkopf B. A local learning approach for clustering[C]//Advances in Neural Information Processing Systems. Germany:NIPS,2006:1 529-1 536.
[15] Golub G H,Van Loan C F. Matrix computations[C]. Baltimore,MD,USA:Johns Hopkins University Press,1996:374-426.
[16] Belkin M,Niyogi P. Semi-supervised learning on riemannian manifolds[J]. Machine Learning,2004:209-239.
[17] Zhu X,Lafferty J,Ghahramani Z. Semi-supervised learning:from Gaussian fields to Gaussian process[R]//Computer Science Technical Report. USA:Carnegie Mellon University,2003.
[18] Hein M,Audibert J Y,Luxburg U von. From graphs to manifolds-weak and strong pointwise consistency of graph laplacians[C]//Proceedings of the 18th Annual Conference on Learning Theory(COLT). Bertinoro,Italy:Springer,2005:470-485.
[19] Wang F, Zhang C,Li T. Clustering with local and global regularization[J]. IEEE Transactions on Knowledge and Data Engineering,2009,21(12):1 665-1 678.
[20] Jiao L,Bo L,Wang L. Fast sparse approximation for least squares support vector machine[J]. IEEE Transactions on Neural Networks,2007,18(3):685-697.
[21] Ng A Y,Jordan M I,Weiss Y. On spectral clustering analysis and an algorithm[C]//Proceedings of Advances in Neural Information Processing Systems. Cambridge,MA:MIT Press,2001,14:849-856.

[责任编辑:黄 敏]