计算机研究与发展
Journal of Computer Research and Development
DOI:10.7544/issn1000-1239.2016.20148362
53(7):1605-1611,2016

基于分布式低秩表示的子空间聚类算法

许 凯 吴小俊 尹贺峰
(江南大学物联网工程学院 江苏无锡 214122)
(xukai347@sina.com)

Distributed Low Rank Representation-Based Subspace Clustering Algorithm

Xu Kai, Wu Xiaojun, and Yin Hefeng
(*School of Internet of Things Engineering, Jiangnan University, Wuxi, Jiangsu* 214122)

Abstract Vision problem ranging from image clustering to motion segmentation can naturally be framed as subspace segmentation problem, in which one aims to recover multiple low dimensional subspaces from noisy and corrupted input data. Low rank representation-based subspace segmentation algorithm (LRR) formulates the problem as a convex optimization and achieves impressive results. However, it needs to take a long time to solve the convex problem, and the clustering accuracy is not high enough. Therefore, this paper proposes a distributed low rank representation-based sparse subspace clustering algorithm (DLRRS). DLRRS adopts the distributed parallel computing to get the coefficient matrix, then take the absolute value of each element of the coefficient matrix, and retain the k largest coefficients per column and set the other elements to 0 to get a new coefficient matrix. Finally, DLRRS performs spectral clustering over the new coefficient matrix. But it doesn't have incremental learning function, so there is a scalable distributed low rank representation-based sparse subspace clustering algorithm (SDLRRS) here. If new samples are brought in, SDLRRS can use the former clustering result to classify the new samples to get the final result. Experimental results on AR and Extended Yale B datasets show that the improved algorithms can not only obviously reduce the running time, but also achieve higher accuracy, which verifies that the proposed algorithms are efficient and feasible.

Key words low rank representation; subspace clustering; parallel computing; incremental learning; coefficients reconstruction

摘 要 针对基于低秩表示的子空间分割算法运算时间较长、聚类的准确率也不够高,提出一种基于分布式低秩表示的稀疏子空间聚类算法(distributed low rank representation-based sparse subspace clustering algorithm, DLRRS),该算法采用分布式并行计算来得到低秩表示的系数矩阵,然后保留系数矩阵每列的前k个绝对值最大系数,其他系数置为0,用此系数矩阵构造一个稀疏的样本关系更突出的相似度矩阵,接着用谱聚类得到聚类结果.但是其不具备增量学习功能,为此再提出一种基于分布式低秩表示的增量式稀疏子空间聚类算法(scalable distributed low rank representation based sparse subspace clustering algorithm, SDLRRS),如果有新增样本,可以利用前面的聚类结果对新增样本进行

收稿日期:2014-12-09;**修回日期**:2015-06-09
基金项目:国家自然科学基金项目(61373055);江苏省自然科学基金项目(BK20140419);江苏省高校自然科学研究计划重大项目(14KJB520001)
This work was supported by the National Natural Science Foundation of China (61373055), the Natural Science Foundation of Jiangsu Province of China (BK20140419), and the Major Program of Research Plan of the Natural Science in Higher Education of Jiangsu Province of China (14KJB520001).
通信作者:吴小俊(wu_xiaojun@jiangnan.edu.cn)

分类得到最后的结果. 实验结果表明:所提 2 种子空间聚类算法不仅有效减少算法的运算时间,还提高了聚类的准确率,从而验证算法是有效可行的.

关键词 低秩表示;子空间聚类;并行计算;增量学习;系数重建

中图法分类号 TP18; TP391.4

高维数据在信息技术高速发展的今天变得越来越普遍,它们通常分布在不同的子空间,这不仅增加了计算机内存的需求量和算法的执行时间,还会对算法[1]的性能产生不利影响,使得很多传统的聚类算法不再适用.

最近几年,子空间聚类技术已经吸引了很多学者的关注,它基于高维数据固有的维数通常要比外围空间的维数低很多的思想,用多个子空间对高维数据进行聚类,并且发现适合每一组数据的低维子空间. 这在计算机视觉、机器学习和模式识别等方面已经有很多的应用,尤其在图像表示[2]、聚类[3]、运动分割[4]这 3 个应用上的性能优异.

可以将存在的子空间聚类算法分成主要的 4 类:代数方法[5]、迭代方法[6-7]、统计方法[8]和基于谱聚类的方法[9-10]. 在这些方法中,基于谱聚类的方法已经显示出其在计算机视觉等方面的优越性能[11-12].

谱聚类算法[13]的核心是构建一个合适的相似度矩阵. 通常用 2 种方法来构造相似度矩阵,即距离的倒数和重建系数. 1)通过计算 2 个数据点间的距离倒数来得到相似度,例如欧氏距离. 基于距离倒数的方法可以得到数据集的局部结构,但它的值仅仅取决于 2 个数据点之间的距离,所以对噪声和异常值很敏感. 2)基于表示系数的方法,假设每个数据点可以被其他数据点的线性组合进行表示,并且表示系数可以被认为是一种度量. 这种度量对噪声和异常值是鲁棒的,因为系数的值不仅取决于 2 个相连的数据点,还取决于其他的所有数据点. 最近的几篇文章已经说明在子空间聚类中表示系数的性能是优于距离倒数的. 例如基于低秩表示的子空间分割算法(low rank representation, LRR)[14]和基于稀疏表示的稀疏子空间聚类算法(sparse subspace clustering, SSC)[3].

虽然 LRR 子空间聚类算法已经取得了不错的聚类效果,但是此算法仍有很大的改进空间. 我们将文献[15]中的并行计算思想和文献[16]中的增量式学习框架相结合,这样不仅能充分利用当前的多核计算机资源,还能直接处理新增的样本,不需要重新聚类,达到充分利用资源节省运算时间的目的. 最主要地,相似度矩阵中的元素衡量的是对应样本的相似程度,是谱聚类算法的核心,构造一个合适的相似度矩阵可以有效地提高算法的准确率. LRR 子空间聚类算法直接用低秩表示所得的系数矩阵来构造相似度矩阵,这样会包含过多的冗余关系. 本文通过保留系数矩阵每列的前 k 个绝对值最大系数、其他位置置 0,得到一个新的系数矩阵,再用此系数矩阵构造一个稀疏的样本关系更突出的相似度矩阵. 在 AR 数据集和 Extended Yale B 人脸库上的实验结果表明本文所提 DLRRS(distributed low rank representation-based sparse subspace clustering algorithm)和 SDLRRS(scalable distributed low rank representation based sparse subspace clustering algorithm)这 2 种算法不仅有效减少运算时间,还提高了聚类的准确率. SDLRRS 算法还具备增量式学习功能.

1 基于低秩表示的子空间分割算法

研究数据空间的结构在很多领域都是一个非常具有挑战性的任务,这通常涉及到秩最小化问题. LRR 算法通过求解式(1)来得到秩最小化问题的近似解:

$$\begin{aligned}&\min \|\boldsymbol{C}\|_*+\lambda\|\boldsymbol{E}\|_L,\\&\text{s.t. } \boldsymbol{Y}=\boldsymbol{Y}\boldsymbol{C}+\boldsymbol{E},\end{aligned}\tag{1}$$

其中,$\|\cdot\|_*$ 表示核范数,是奇异值的和;$\boldsymbol{C}\in\mathbb{R}^{n\times n}$ 就是数据集矩阵 $\boldsymbol{Y}\in\mathbb{R}^{m\times n}$ 的低秩表示;$\boldsymbol{E}$ 对应稀疏的干扰矩阵;$\|\cdot\|_L$ 可以表示 $L_{2,1}$ 范数、L_1 范数或者 Frobenius 范数,它们的选择取决于在数据集中假设存在哪种误差. 具体就是,$L_{2,1}$ 范数常被用来描述特定样本的污损和异常值,L_1 范数更适合用来描述随机的稀疏异常值,Frobenius 范数通常用来描述小的高斯噪声. Liu 等人[14]应用增广拉格朗日乘子法来解决核范数正规化优化问题可以得到式(1)的解. 在算法 1 中,我们概述了 LRR 算法的具体实现.

算法 1. LRR 算法[14].

输入:数据集矩阵 $\boldsymbol{Y}\in\mathbb{R}^{m\times n}$ 和类别数 u.

① 解决核范数最小化式(1)得到 $\boldsymbol{C}=[\boldsymbol{c}_1,\boldsymbol{c}_2,\cdots,\boldsymbol{c}_n]$;

② 得到相似度矩阵 $\boldsymbol{W}=|\boldsymbol{C}|+|\boldsymbol{C}|^{\mathrm{T}}$;

③ 对相似度矩阵 $\boldsymbol{W}$ 使用谱聚类;

④ 输出数据集矩阵 $\boldsymbol{Y}$ 的类分配.

2　基于分布式低秩表示的子空间聚类算法

2.1　基于分布式低秩表示的稀疏子空间聚类

低秩子空间分割算法可以很精确地处理小规模的数据集,但不能有效处理大规模数据集.为此,文献[15]中提出了一种分布式低秩子空间分割算法,该算法将大规模数据集矩阵 $\boldsymbol{Y}$ 按列分割成 t 个小规模的数据矩阵$\{\boldsymbol{Z}_1,\boldsymbol{Z}_2,\cdots,\boldsymbol{Z}_t\}$,然后再对这 t 个小规模数据矩阵进行并行处理.其中第 i 个 LRR 子问题的处理形式为

$$\begin{aligned}&\min\ \|\boldsymbol{C}_i\|_*+\lambda\|\boldsymbol{E}_i\|_L,\\&\text{s.t.}\ \ \boldsymbol{Z}_i=\boldsymbol{Y}\boldsymbol{C}_i+\boldsymbol{E}_i.\end{aligned}\tag{2}$$

运用此分而治之的思想,不仅保证了算法所得结果的准确率,还充分利用计算机的多核硬件资源,极大地降低算法的运算时间.在分别得到 t 个子系数矩阵后,本文不采用文献[15]中的投影方式来得到最后的系数矩阵,而是直接按列排成最后的系数矩阵.

另外,基于低秩表示的子空间分割和分布式低秩子空间分割这 2 个算法中,都是在得到系数矩阵 $\boldsymbol{C}$ 后,直接用此系数矩阵来构造相似度矩阵,这样会产生大量冗余的关系,降低算法所得结果的准确率.为此,本文在得到系数矩阵后,先对系数矩阵中的每个元素取绝对值;然后保留每列的前 k 个最大值,其他位置的元素置为 0;再次用新得到的系数矩阵来构造相似度矩阵;最后用谱聚类来得到聚类结果.具体实现过程如算法 2 所示.

算法 2. DLRRS 算法.

输入:数据集矩阵 $\boldsymbol{Y}\in\mathbb{R}^{m\times n}$、类别数 u、每列保留的系数个数 k 和并行计算分割数 t.

① 将数据集矩阵 $\boldsymbol{Y}$ 按列分割成 t 个子数据矩阵$\{\boldsymbol{Z}_1,\boldsymbol{Z}_2,\cdots,\boldsymbol{Z}_t\}$;

② 进行并行计算

$$\begin{aligned}\boldsymbol{C}_1^*&=\mathrm{LRR}(\boldsymbol{Z}_1,\boldsymbol{Y});\\\boldsymbol{C}_2^*&=\mathrm{LRR}(\boldsymbol{Z}_2,\boldsymbol{Y});\\&\vdots\\\boldsymbol{C}_t^*&=\mathrm{LRR}(\boldsymbol{Z}_t,\boldsymbol{Y});\end{aligned}$$

③ 得到系数矩阵 $\boldsymbol{C}^*=[\boldsymbol{C}_1^*,\boldsymbol{C}_2^*,\cdots,\boldsymbol{C}_t^*]$;

④ 对系数矩阵 $\boldsymbol{C}^*$ 中的每个元素取绝对值,然后保留每列最大的 k 个元素,其他元素置为 0,得到一个新的系数矩阵 $\hat{\boldsymbol{C}}$;

⑤ 得到相似度矩阵 $\boldsymbol{W}=|\hat{\boldsymbol{C}}|+|\hat{\boldsymbol{C}}|^{\mathrm{T}}$;

⑥ 对相似度矩阵 $\boldsymbol{W}$ 使用谱聚类;

⑦ 输出数据集矩阵 $\boldsymbol{Y}$ 的类分配.

2.2　分布式低秩增量式稀疏子空间聚类

在我们已经完成聚类得到聚类结果后,如果此时有新的样本加入,传统的聚类算法只有重新聚类所有样本,不具备增量学习的功能,会导致计算资源的浪费.在文献[16]中,提出了一种先聚类后分类的增量式聚类算法.本文参考此结构,先进行聚类,然后再用协同表示分类算法对新增的样本进行分类.协同表示分类需要求解的目标函数为

$$\min_{\boldsymbol{c}}\|\boldsymbol{y}-\boldsymbol{Y}\boldsymbol{c}\|_2^2+\lambda\|\boldsymbol{c}\|_2^2,\tag{3}$$

其中,$\boldsymbol{y}$ 是数据集矩阵 $\boldsymbol{Y}\in\mathbb{R}^{m\times n}$ 中的一个样本,$\boldsymbol{c}$ 是经过数据集矩阵 $\boldsymbol{Y}$ 对样本 $\boldsymbol{y}$ 进行协同表示的系数列向量.在得到最优的系数列向量后,通过计算式(4)得到属于所有类的标准化残差:

$$r_j(\boldsymbol{y})=\frac{\|\boldsymbol{y}-\boldsymbol{Y}\delta_j(\boldsymbol{c}^*)\|_2}{\|\delta_j(\boldsymbol{c}^*)\|_2},\tag{4}$$

$$identity(\boldsymbol{y})=\arg\min_j\{r_j(\boldsymbol{y})\},\tag{5}$$

其中,$\delta_j(\boldsymbol{c}^*)$表示保留系数列向量 $\boldsymbol{c}^*$ 中对应第 j 类的元素,其他元素置为 0;$r_j(\boldsymbol{y})$表示样本 $\boldsymbol{y}$ 属于第 j 类的标准化残差.

最后通过式(5)得到最终的分类结果.基于分布式低秩表示的可拓展稀疏子空间聚类算法的实现过程如算法 3 所示.

算法 3. SDLRRS 算法.

输入:数据集矩阵 $\boldsymbol{Y}\in\mathbb{R}^{m\times n}$、类别数 u、每列保留的系数个数 k 和并行计算分割数 t.

① 使用随机抽样或其他方法从数据集矩阵 $\boldsymbol{Y}$ 中选出 p 个数据点,表示为 $\boldsymbol{X}=[\boldsymbol{x}_1,\boldsymbol{x}_2,\cdots,\boldsymbol{x}_p]$,没有被抽到的数据点组成数据矩阵 $\bar{\boldsymbol{X}}$;

② 在数据矩阵 $\boldsymbol{X}$ 上运行 DLRRS 算法,得到聚类结果;

③ 将已经具有类标签的数据矩阵 $\boldsymbol{X}$ 作为训练集,$\bar{\boldsymbol{X}}$ 作为测试集,进行协同表示,可以得到系数矩阵:

$$\bar{\boldsymbol{C}}^*=(\boldsymbol{X}^{\mathrm{T}}\boldsymbol{X}+\lambda\boldsymbol{I})^{-1}\boldsymbol{X}^{\mathrm{T}}\bar{\boldsymbol{X}};$$

④ 通过下式计算 $\bar{\boldsymbol{x}}_i\in\bar{\boldsymbol{X}}$ 到所有类的标准化残差:

$$r_j(\bar{\boldsymbol{x}}_i)=\frac{\|\bar{\boldsymbol{x}}_i-\boldsymbol{X}\delta_j(\bar{\boldsymbol{c}}_i^*)\|_2}{\|\delta_j(\bar{\boldsymbol{c}}_i^*)\|_2};$$

⑤ 通过下式将 $\bar{\boldsymbol{x}}_i$ 归属于第 j 类:

$$identity(\bar{\boldsymbol{x}}_i)=\arg\min_j\{r_j(\bar{\boldsymbol{x}}_i)\};$$

⑥ 输出数据矩阵 $\boldsymbol{Y}$ 的类分配.

3 实验结果与分析

$$聚类准确率=\frac{正确聚类的样本个数}{样本的总数}. \quad (6)$$

本节我们使用子空间聚类准确率(式(6))和归一化互信息(normalized mutual information, NMI)来评估本文基于分布式低秩表示的子空间聚类算法的性能.同时,为了验证本文算法的有效性,实验通过3方面来进行比较分析:1)通过实验将本文算法的参数调到最佳;2)讨论并行计算分割数 t 对DLRRS算法的影响;3)讨论SDLRRS算法增量学习功能的有效性.

其中用到的参考算法有分布式低秩子空间分割算法(distributed low-rank subspace segmentation, DFC-LRR)[15]、基于低秩表示的子空间分割算法(low rank representation, LRR)[14]、稀疏子空间聚类算法(sparse subspace clustering, SSC)[3]、可拓展的基于低秩表示的子空间分割算法(scalable low rank representation, SLRR)[16]和可拓展的稀疏子空间聚类算法(scalable sparse subspace clustering, SSSC)[16].后2种算法分别用LRR和SSC算法先进行聚类,当有新样本加入时再用分类的方法得到结果.实验在同一台PC机(CPU:3.20 GHz,内存:8 GB)上进行,操作系统版本为64位Windows 8,实验工具为MATLAB R2013a.

实验选用2个常用的人脸数据集:AR数据集和Extended Yale B数据集.其中AR数据集包含超过4 000幅126个人(70个男性、56个女性)的人脸图片,这些图片是在不同的表情、不同光照和伪装(戴墨镜或围巾)下得到的.每个人有26幅图片,其中14幅"干净"图片、6幅戴墨镜、6幅戴围巾.这里我们参照文献[17],从50个男性和50个女性的图片中随机选出1 400幅"干净"的人脸图片.Extended Yale B人脸库中有38个人,每个人在不同光照条件下得到64张正面人脸图像,每个人脸图像经过裁剪后有192×168个像素.为了降低所有算法的计算复杂度和对内存的需求量,我们将AR数据集中的图片下采样到55×40,Extended Yale B人脸库中的图片都下采样到48×42个像素,并且对它们进行PCA保留98%的信息.各个数据集的详细信息如表1所示.

Table 1 Datasets Used in Our Experiments

表1 实验中用到的数据集

Attribute	Datasets	
	AR	Extended Yale B
#Samples	1 400	2 414
dimension	19 800	32 256
#Features	167	114
#Classses	100	38

3.1 参数对本文算法的影响

本文所提的2种子空间聚类算法包含3个参数:平衡参数 λ、每列保留的系数个数 k 和并行计算的分割数 t.本节只讨论平衡参数 λ 和每列保留的系数个数 k 对DLRRS和SDLRRS这2种算法聚类质量的影响,先设置 $t=1$,3.2节再详细讨论参数 t 对算法的影响.

图1(a)(b)展示了在AR数据集上参数 λ 和 k 对DLRRS算法的影响.当 λ 逐渐增大的时候,对应的聚类准确率和NMI也逐渐升高,然后趋于稳定.当 k 从3变到8时,对应的聚类准确率从65.36%变到85.93%,NMI从81.78%变到93.66%;当 k 继续增大时,对应的聚类准确率和NMI呈现出缓慢下降的趋势.所以DLRRS算法在AR数据集上的参数选择为平衡参数 $\lambda=2.2$ 和保留的系数个数 $k=8$.

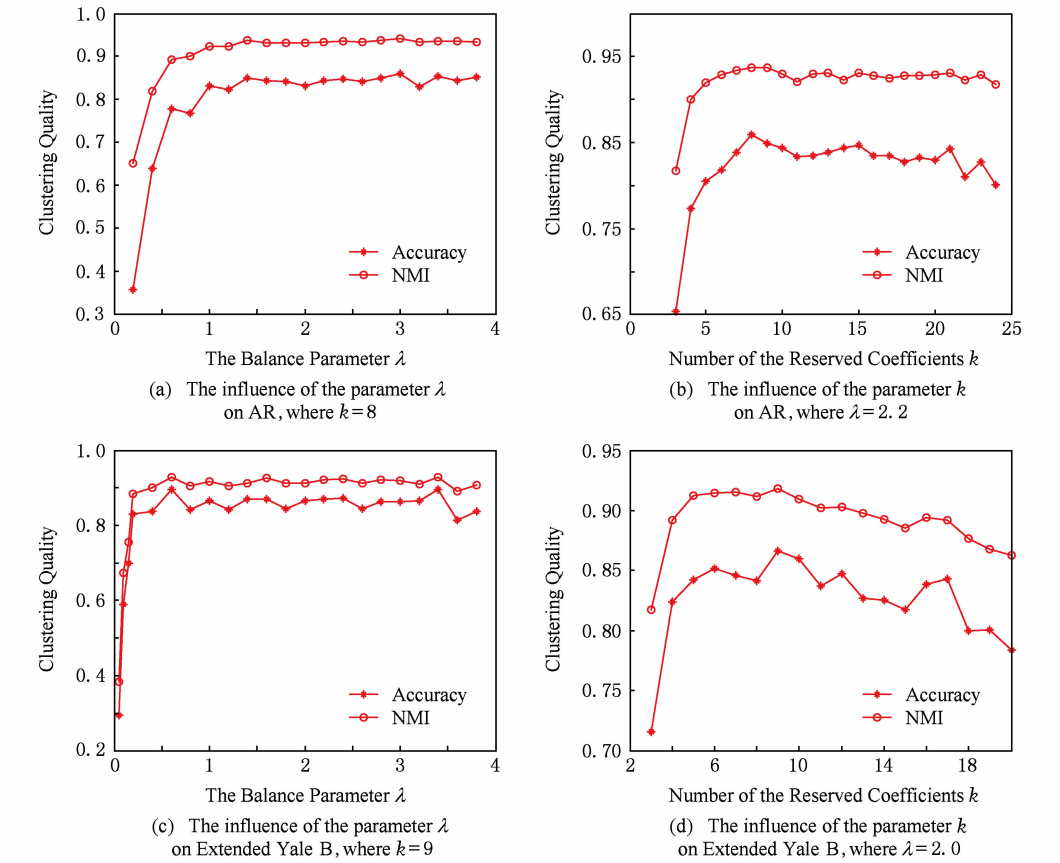
图1(c)(d)展示了在Extended Yale B数据集上参数 λ 和 k 对本文算法的影响.当 λ 从0.05变到2时,对应的聚类准确率从29.41%变到86.45%,NMI从38.37%变到91.15%;当 λ 从2变到3.8时,对应的聚类准确率和NMI基本保持不变.当 k 从3变到9时,对应的聚类准确率从71.58%变到86.62%,NMI从81.70%变到91.84%;在 $k=9$ 时DLRRS算法取得最好的聚类质量;当 k 从9变到20时,对应的聚类准确率从86.62%一直下降到78.38%,NMI从91.84%下降到86.27%.所以DLRRS算法在Extended Yale B数据集上的参数选择为平衡参数 $\lambda=2$ 和保留的系数个数 $k=9$.

由于篇幅所限,在此直接给出SDLRRS算法的参数设置,在AR数据集上为平衡参数 $\lambda=3.1$ 和保留的系数个数 $k=6$,在Extended Yale B数据集上为平衡参数 $\lambda=2.9$ 和保留的系数个数 $k=5$.

3.2 分割数 t 对算法质量的影响

由于实验室只有4核处理器,所以分割数 t 取1~4,然后在AR和Extended Yale B数据集上进行DLRRS和DFC-LRR这2个算法的对比实验.

1) 横向比较.从表2可以看出,在AR数据集上,本文DLRRS算法的聚类准确率较DFC-LRR



(a) The influence of the parameter λ on AR, where $k=8$　(b) The influence of the parameter k on AR, where $\lambda=2.2$

(c) The influence of the parameter λ on Extended Yale B, where $k=9$　(d) The influence of the parameter k on Extended Yale B, where $\lambda=2.0$

Fig. 1　Clustering quality of DLRRS with varying parameters.

图 1　参数对本文 DLRRS 算法聚类质量的影响

Table 2　Clustering Quality of Algorithms with Different Parameter t

表 2　分割数 t 对算法的影响

Parameter t	Quality	AR		Extended Yale B	
		DLRRS	DFC-LRR[15]	DLRRS	DFC-LRR[15]
1	Accuracy	**0.8530**	0.8001	**0.8459**	0.6860
	NMI	**0.9380**	0.9072	**0.9103**	0.7439
	Running Time/s	**64.17**	64.71	**62.49**	80.16
2	Accuracy	**0.8486**	0.8044	**0.8558**	0.6698
	NMI	**0.9375**	0.9118	**0.9119**	0.7388
	Running Time/s	**52.38**	52.81	**56.32**	64.54
3	Accuracy	**0.8484**	0.8021	**0.8618**	0.6757
	NMI	**0.9396**	0.9140	**0.9180**	0.7451
	Running Time/s	**48.48**	49.84	**53.46**	62.2
4	Accuracy	**0.8499**	0.8040	**0.8500**	0.6764
	NMI	**0.9401**	0.9176	**0.9123**	0.7427
	Running Time/s	**46.19**	46.37	**54.63**	63.42

算法高出 5%左右,两者的运算时间基本一致,DLRRS 算法稍优一点;在 Extended Yale B 数据集上,DLRRS 算法在聚类准确率方面高出 DFC-LRR 算法 18%左右,在运算时间方面可以节省 10 s 左右.主要有 2 方面原因使得本文 DLRRS 算法完全优于 DFC-LRR 算法:①保留系数矩阵每列的前 k 个绝对值最大系数,其他位置置 0,然后再构造稀疏的相似度矩阵是有效提高本文算法准确率的关键;

②在并行计算时,不采用投影的方式,而是直接按列排成最后的系数矩阵,在保证聚类准确率的同时可以减少算法的运算时间.

2) 纵向比较. 表 2 所示为并行计算的分割数 t 对算法的影响. 可以很直观地看出,随着 t 的增大,DLRRS 和 DFC-LRR 这 2 个算法的聚类准确率在 AR 和 Extended Yale B 数据集上几乎不受影响,但却可以大幅降低算法的执行时间;$t=4$ 时较 $t=1$ 时在 AR 数据集上可以节省 28% 左右的时间,在 Extended Yale B 数据集上可以节省 13%左右的时间. 由于实验室的计算机只有 4 核,当 t 从 1 变到 2 时,DLRRS 算法在 2 个数据集上的执行时间降幅最大,分别为 18%和 9.8%;当 t 从 2 变到 3 时,执行时间的降幅会变小;当 t 从 3 变到 4 时,执行时间的降幅变得不是很明显,在 Extended Yale B 数据集上相较 $t=3$ 时还出现了小幅度的上升,这是由于实验室 CPU 只有 4 核,在 $t=4$ 满负荷运算时不可能只执行并行计算的代码,还要执行其他指令,这并不影响本文算法的有效性. 综上,我们可以预见如果计算机的核数变得更多、数据集的规模变大,本文 DLRRS 算法在牺牲有限准确率的同时,节省运算时间的优势会更加明显.

3.3 增量学习功能

对已经聚类好的样本,如果此时有新样本加入,DLRRS 算法需要重新聚类. 为此,本文在 DLRRS 算法的基础上提出 SDLRRS 算法使其具备增量学习功能. 为了验证 SDLRRS 算法的性能,我们分别将 AR 和 Extended Yale B 数据集中的一半样本随机选出作为新加入的样本进行测试,并和同样具备增量学习功能的 SLRR 算法和 SSSC 算法进行对比. 对于 DLRRS,LRR 和 SSC 这 3 种不具备增量学习功能的聚类算法直接使用全部样本进行聚类测试. 表 3 给出了不同算法在 AR 和 Extended Yale B 数据集上的聚类结果,同时列出了各个算法使用的参数,其中 λ 是平衡参数,k 指系数矩阵中每列保留的系数个数,t 是并行计算的分割数,μ 是进行交替方向乘子法计算时的惩罚参数. 3.2 节我们已经知道并行计算分割数 t 对 DLRRS 算法的聚类准确率影响很小,为了方便讨论 SDLRRS 算法增量学习的效果,本节我们设置 $t=1$.

Table 3 Clustering Quality Comparison of Different Algorithms on Datasets

表 3 不同算法在数据集上的聚类质量对比

Algorithms	AR							Extended Yale B						
	Accuracy	λ	k	t	μ	NMI	Running Time/s	Accuracy	λ	k	t	μ	NMI	Running Time/s
SDLRRS	0.815 3	3.1	6	1		0.913 5	34.58	0.841 4	2.9	5	1		0.875 0	25.45
DLRRS	0.853 0	2.2	8	1		0.938 0	64.17	0.863 3	2	9	1		0.914 6	62.49
SLRR[16]	0.783 9	3.1				0.894 5	34.62	0.672 9	2.9				0.733 0	24.80
LRR[14]	0.800 1	2.2				0.907 2	64.71	0.686 0	2				0.743 9	80.16
SSSC[16]	0.698 5				30	0.861 3	43.43	0.560 0				5	0.602 6	57.18
SSC[3]	0.775 6				30	0.884 0	142.59	0.674 1				5	0.725 8	252.75

从表 3 可以看出,SDLRRS 算法和 DLRRS 算法的聚类准确率分别较 SLRR 算法,LRR 算法在 AR 数据集上有 4%左右的提升,在 Extended Yale B 数据集上有 17%的提升. 当有新的样本加入时,DLRRS,LRR,SSC 这 3 种算法不得不对所有样本重新聚类,导致大量资源浪费. 而可拓展的 3 种聚类算法 SDLRRS,SLRR,SSSC 可以直接处理新加入的样本,不需要对所有样本重新聚类. 在 AR 数据集上的准确率,SDLRRS 算法比 DLRRS 算法低 3.80%,SLRR 算法比 LRR 算法低 1.62%,SSSC 算法比 SSC 算法低 7.71%;在 Extended Yale B 数据集上的准确率,SDLRRS 算法比 DLRRS 算法低 2.19%,SLRR 算法比 LRR 算法低 1.31%,SSSC 算法比 SSC 算法低 11.41%,可以验证可拓展算法的有效性. 尤其是本文的可拓展聚类算法 SDLRRS,比进行了重新聚类的 LRR 算法在 AR 数据集上的准确率还高出 1.52%,在 Extended Yale B 数据集上高出 15.54%;比 SSC 算法在 AR 数据集上高出 3.97%,在 Extended Yale B 数据集上高出 17.73%. 另外,SDLRRS 算法的运算时间相较 LRR 算法和 SSC 算法至少节省一半以上,所以 SDLRRS 算法不仅可以用来处理新增加的样本,必要的时候还可以用来快速聚类整个数据集,足见本文算法是非常有效可行的.

4　结束语

本文首先设计了一种基于分布式低秩表示的稀疏子空间聚类算法,此算法运用并行计算思想,并且通过保留系数矩阵每列的前 k 个绝对值最大系数、其他系数置为 0,达到简化突出样本间相似程度的目的,此算法具有充分利用计算资源节省运算时间和提高聚类准确率的优点.但它不具备增量学习功能,为此,又提出一种基于分布式低秩表示的增量式稀疏子空间聚类算法,在 AR 数据集和 Extended Yale B 人脸库上的聚类效果优异.但是,本文的研究工作还有待进一步深入和扩展,如新增加的样本不属于前面聚类的类,这时就不可以简单地根据前面的聚类结果对新增样本进行分类.

参　考　文　献

[1] Ying Wenhao, Xu Min, Wang Shitong, et al. Fast adaptive clustering by synchronization on large scale datasets [J]. Journal of Computer Research and Development, 2014, 51(4): 707-720 (in Chinese)
(应文豪, 许敏, 王士同, 等. 在大规模数据集上进行快速自适应同步聚类[J]. 计算机研究与发展, 2014, 51(4): 707-720)

[2] Hong W, Wright J, Huang K, et al. Multiscale hybrid linear models for lossy image representation [J]. IEEE Trans on Image Processing, 2006, 15(12): 3655-3671

[3] Elhamifar E, Vidal R. Sparse subspace clustering: Algorithm, theory, and applications [J]. IEEE Trans on Pattern Analysis and Machine Intelligence, 2013, 35(11): 2765-2781

[4] Zhuang L, Gao H, Lin Z, et al. Non-negative low rank and sparse graph for semi-supervised learning [C] //Proc of IEEE CVPR'12. Piscataway, NJ: IEEE, 2012: 2328-2335

[5] Vidal R, Ma Y, Sastry S. Generalized principal component analysis (GPCA) [J]. IEEE Trans on Pattern Analysis and Machine Intelligence, 2005, 27(12): 1945-1959

[6] Zhang T, Szlam A, Lerman G. Median k-flats for hybrid linear modeling with many outliers [C] //Proc of the 12th Int Conf on Computer Vision Workshops. Piscataway, NJ: IEEE, 2009: 234-241

[7] Lu L, Vidal R. Combined central and subspace clustering for computer vision applications [C] //Proc of the 23rd Int Conf on Machine learning. New York: ACM, 2006: 593-600

[8] Rao S, Tron R, Vidal R, et al. Motion segmentation in the presence of outlying, incomplete, or corrupted trajectories [J]. IEEE Trans on Pattern Analysis and Machine Intelligence, 2010, 32(10): 1832-1845

[9] Favaro P, Vidal R, Ravichandran A. A closed form solution to robust subspace estimation and clustering [C] //Proc of IEEE CVPR'11. Piscataway, NJ: IEEE, 2011: 1801-1807

[10] Elhamifar E, Vidal R. Clustering disjoint subspaces via sparse representation [C] //Proc of IEEE ICASSP'10. Piscataway, NJ: IEEE, 2010: 1926-1929

[11] Vidal R. A tutorial on subspace clustering [J]. IEEE Signal Processing Magazine, 2010, 28(2): 52-68

[12] Li Qingyong, Liang Zhengping, Huang Yaping, et al. Sparseness representation model for defect detection and its application [J]. Journal of Computer Research and Development, 2014, 51(9): 1929-1935 (in Chinese)
(李清勇, 梁正平, 黄雅平, 等. 缺陷检测的稀疏表示模型及应用[J]. 计算机研究与发展, 2014, 51(9): 1929-1935)

[13] Von Luxburg U. A tutorial on spectral clustering [J]. Statistics and Computing, 2007, 17(4): 395-416

[14] Liu G, Lin Z, Yan S, et al. Robust recovery of subspace structures by low-rank representation [J]. IEEE Trans on Pattern Analysis and Machine Intelligence, 2013, 35(1): 171-184

[15] Talwalkar A, Mackey L, Mu Y, et al. Distributed low-rank subspace segmentation [C] //Proc of IEEE ICCV'13. Piscataway, NJ: IEEE, 2013: 3543-3550

[16] Peng X, Zhang L, Yi Z. Scalable sparse subspace clustering [C] //Proc of IEEE CVPR'13. Piscataway, NJ: IEEE, 2013: 430-437

[17] Wright J, Yang A Y, Ganesh A, et al. Robust face recognition via sparse representation [J]. IEEE Trans on Pattern Analysis and Machine Intelligence, 2009, 31(2): 210-227

Xu Kai, born in 1989. Master. His main research interests include pattern recognition and data mining.

Wu Xiaojun, born in 1967. Professor and PhD supervisor. Senior member of China Computer Federation. His main research interests include pattern recognition, computer vision, fuzzy systems, neural networks, and intelligent systems.

Yin Hefeng, born in 1989. PhD candidate. Student member of China Computer Federation. His main research interests include feature extraction, sparse representation and low rank representation.