基于局部聚类与图方法的半监督学习算法

李 明 [1,2] 杨艳屏 [1,2] 占惠融 [1,2]

摘 要 基于图的算法已经成为半监督学习中的一种流行方法, 该方法把数据定义为图的节点, 用图的边表示数据之间的关系, 在各种数据分布情况下都具有很高的分类准确度. 然而图方法的计算复杂度比较高, 当图的规模比较大时, 计算所需要的时间和存储都非常大, 这在一定程度上限制了图方法的使用. 因此, 如何控制图的大小是基于图的半监督学习算法中的一个重要问题. 本文提出了一种基于密度估计的快速聚类方法, 可以在局部范围对数据点进行聚类, 以聚类形成的子集作为构图的节点, 从而大大降低了图的复杂度. 新的聚类方法计算量较小, 通过推导得到的距离函数能较好地保持原有数据分布. 实验结果表明, 通过局部聚类后构建的小图在分类效果上与在原图上的结果相当, 同时在计算速度上有极大的提高.

关键词 半监督学习, 图方法, 密度估计, 局部聚类

DOI 10.3724/SP.J.1004.2010.01655

Semi-supervised Learning Based on Graph and Local Quick Shift

LI Ming[1,2] YANG Yan-Ping[1,2] ZHAN Hui-Rong[1,2]

Abstract Graph-based semi-supervised methods define a graph where the nodes are labeled and unlabeled examples in the dataset, and edges reflect the similarity of examples. These methods usually assume label smoothness over the graph. Graph methods are nonparametric, discriminative, and transductive in nature. These methods take high classification accuracy on variant data distributions. But the computation complexity is very high. As the size of dataset grows, the graph will be too large to compute and this limits the extension of its usage. In this paper, we propose a novel method for fast computation based on local clustering, which is very efficient for reduction of graph size and can maintain the accuracy at the same time. The local clustering method is of low computation complexity and the data structure can be preserved by a newly designed distance function. Experimental results show that this approach can preserve the accuracy of purely graph-based methods and significantly reduce computational cost.

Key words Semi-supervised learning, graph-based methods, density estimate, local clustering

图方法是半监督学习算法中一个颇具吸引力的方法, 它把标记数据与未标记数据都表示成图的节点, 根据数据之间的距离关系构建连接节点的边的权重, 然后从图谱的角度出发设计算法对节点进行分类. 自 Blum 等[1] 的工作以来, 基于图的方法在半监督学习领域获得了不断的发展[2−4], 更多细节可以参见文献 [5]. 图方法的种类很多, 但基本结构变化不大, Zhou 等在文献 [6] 中给出的图方法的正则化框架可以作为理解图方法的基础, 它还从随机游走和贝叶斯估计的角度对图方法进行解释. 由于图方法的计算速度依赖于图的节点数目, 当数据量很大时直接把每个数据点作为一个图节点的办法并不适合. 针对这一问题目前已有若干研究. 其中, Zhu 等[4] 使用高斯混合模型的方法减少图节点数目, 通过混合模型生成原始数据的骨架图, 骨架图的每个节点是一个混合分量, 当混合模型与数据的分布模型相适应时, 该方法能取得较好的分类效果. Delalleau 等[7] 使用数据采样的方法获得原始数据集合上的子集合, 使用子集合上的数据作为构图的节点数据, 从而降低图的规模. Pfahringer 等[8] 通过使用稀疏图降低图的存储空间和计算时间. Zhou 等[9] 则研究了在资源受限情况下的图方法. 构图的另外一些研究则关注于数据的结构关系, Wang 等[10] 使用节点之间的线性组合关系构建权重, 取得了较好的分类效果. Cheng 等[11] 在文献 [10] 的基础上使用 L_1 范数代替 L_2 范数作为约束项, 获得节点的稀疏权重表示, 实验显示了很好的分类效果. 在本文中, 我们主要研究使用聚类的方法缩小图规模. 结合图方法所依据的聚类假设, 我们使用基于密度估计的方法对数据进行聚类. 聚类后的每一个簇作为构图的一个节点, 聚类后获得的图远小于在原始数据上直接构造的图, 因此在很大程度上减少了图方法的计算量. 而且新设计的簇间距离能很好地保持原图节点之间的距离关系, 使得分类效果与在原始图上进行的分类相当.

收稿日期 2009-12-28 录用日期 2010-05-20
Manuscript received December 28, 2009; accepted May 20, 2010
中国国际科技合作项目 (2009DFA12290) 资助
Supported by the China International Science and Technology Cooperation Project (2009DFA12290)
1. 华中科技大学计算机学院 武汉 430074 2. 教育部图像处理与智能控制国家重点实验室 武汉 430074
1. School of Computer Science and Technology, Huazhong University of Science and Technology, Wuhan 430074 2. Key Laboratory of Ministry of Education for Image Processing and Intelligent Control, Wuhan 430074

1 算法

半监督学习领域的两个常用假设是聚类假设和流形假设[12]. 聚类假设是指在相同聚类中的样本有较大的可能拥有相同的标记, 这说明不同标记点的分界面不应该出现在样本密度较大的区域; 而流形假设是指相邻的样本具有相似的性质, 其标记也应该相似. 由于图方法与基于密度的聚类方法都遵守这两个假设, 所以在一定程度上综合使用这两种方法, 其分类效果应该与单独使用图方法相当. 具体来说, 图方法的计算复杂度较高, 而局部密度估计的计算复杂度要低得多; 我们一方面利用局部聚类在计算时间上的优势缩小构图的节点数目, 另一方面利用图方法对数据结构整体的刻画能力获得较好的分类效果.

1.1 基于聚类图的半监督学习算法

设数据样本集合为 $X = \{\boldsymbol{x}_1, \cdots, \boldsymbol{x}_l, \cdots, \boldsymbol{x}_n\} \subset \mathbf{R}^m$, 标记集合为 $L = \{1, \cdots, s\}$, 样本集合中的前 l 个点 $\boldsymbol{x}_i (i \leq l)$ 为已标记样本点, 各点对应的标记为 $y_i \in L$, 数据集合中剩下的点为未标记点. 令 Γ 为 $n \times s$ 维度的非负矩阵集合, 矩阵 $F = [\boldsymbol{F}_1^{\mathrm{T}}, \cdots, \boldsymbol{F}_n^{\mathrm{T}}]^{\mathrm{T}} \in \Gamma$ 中的向量 $\boldsymbol{F}_i$ 表示元素 $\boldsymbol{x}_i$ 针对各个标号的归属度, 显然 $y_i = \arg\max_{j \leq s} F_{ij}$ 表示元素 $\boldsymbol{x}_i$ 的分类标号. 令初始标号矩阵 $Y \in \Gamma$, 当 $\boldsymbol{x}_i$ 的标号为 j 时 $Y_{ij} = 1$, 否则 $Y_{ij} = 0$. 目前已经有许多基于谱图理论的半监督学习算法, 在此只结合文献 [3] 中介绍的 LLGC (Learning with local and global consistency) 算法描述聚类与图方法的结合, 相同的结合方法也同样适用于其他基于谱图理论的方法.

根据聚类假设, LLGC 的正则化理论模型如下:

$$E(F) = \frac{1}{2}\left(\sum_{i,j=1}^{n} W_{ij} \left\| \frac{1}{\sqrt{D_{ii}}}\boldsymbol{F}_i - \frac{1}{\sqrt{D_{jj}}}\boldsymbol{F}_j \right\|^2 + \mu \sum_{i=1}^{n} \|\boldsymbol{F}_i - \boldsymbol{Y}_i\|^2 \right) \tag{1}$$

其中, W 为相似矩阵, D 为对角矩阵, $D_{ii} = \sum_j W_{ij}$, $\mu > 0$ 为正则化参数. 则分类函数为

$$F^* = \arg\min_{F \in \Gamma} E(F) \tag{2}$$

能量函数 $E(F)$ 的第一项代表了聚类约束, 它说明距离接近的元素的分类也应该接近; 第二项表示标号约束, 说明元素的分类不能偏离已有的标号, 该项使得已标号元素在分类中具有锚定作用.

LLGC 的一个主要问题是计算量比较大, W 随分类的样本点以平方的速度增加, 当数据量比较大的时候, 存储和计算都很复杂. 显然, 通过减少图的节点数目可以有效地缓解该问题. 但是如何在减少图节点数目的同时保持原有的分类准确性, 则是一个值得考虑的问题.

假设缩减后的节点集合为 $C = \{c_1, \cdots, c_h, c_{h+1}, \cdots, c_v\}$, 其中每个元素都是原始数据集的一个簇, $c_i = \{\boldsymbol{x}_{i1}, \cdots, \boldsymbol{x}_{ik_i}\}, k_i \geq 1$. 簇集合中的前 h 个簇 c_i $(i \leq h)$ 为已标记的簇. 如果要把 $E(F)$ 的第一项表示为簇集合的形式. 则

$$\sum_{i,j=1}^{n} g(W_{ij}, \boldsymbol{F}_i, \boldsymbol{F}_j) = \sum_{\substack{c_p, c_q \in C \\ p \neq q}} \sum_{\substack{\boldsymbol{x}_i \in c_p \\ \boldsymbol{x}_j \in c_q}} g(W_{ij}, \boldsymbol{F}_i, \boldsymbol{F}_j) + \sum_{c_p \in C} \sum_{\boldsymbol{x}_i, \boldsymbol{x}_j \in c_p} g(W_{ij}, \boldsymbol{F}_i, \boldsymbol{F}_j) \tag{3}$$

其中

$$g(W_{ij}, \boldsymbol{F}_i, \boldsymbol{F}_j) = W_{ij} \left\| \frac{1}{\sqrt{D_{ii}}}\boldsymbol{F}_i - \frac{1}{\sqrt{D_{jj}}}\boldsymbol{F}_j \right\|^2 \tag{4}$$

令 $\bar{W}_{pq} = \sum_{\boldsymbol{x}_i \in c_p, \boldsymbol{x}_j \in c_q} W_{ij}$, 则 $\sum_{c_p, c_q \in C, p \neq q} \sum_{\boldsymbol{x}_i \in c_p, \boldsymbol{x}_j \in c_q} g(W_{ij}, \boldsymbol{F}_i, \boldsymbol{F}_j) = \sum_{c_p, c_q \in C, p \neq q} g(\bar{W}_{pq}, \boldsymbol{F}_p, \boldsymbol{F}_q)$, 符合聚类约束的形式. 如果能令 $\sum_{c_p \in C} \sum_{\boldsymbol{x}_i, \boldsymbol{x}_j \in c_p} g(W_{ij}, \boldsymbol{F}_i, \boldsymbol{F}_j) \to 0$ 并舍弃该项, 则可以获得簇图上的 LLGC 能量函数. 为了使 $\sum_{c_p \in C} \sum_{\boldsymbol{x}_i, \boldsymbol{x}_j \in c_p} g(W_{ij}, \boldsymbol{F}_i, \boldsymbol{F}_j)$ 尽可能小, 需要 $\boldsymbol{F}_i$ 与 $\boldsymbol{F}_j$ 尽量接近, 这意味着 c_p 中的元素要尽量集中. 综合上述分析, 本文算法流程如下:

1) 计算样本点的相似矩阵 W, 其中

$$W_{ij} = \begin{cases} \exp\left(\dfrac{-\|\boldsymbol{x}_i - \boldsymbol{x}_j\|^2}{2\sigma^2}\right), & i \neq j \\ 0, & i = j \end{cases}$$

2) 对样本集合 X 进行聚类, 得到簇集合 C;

3) 计算簇集合 C 的相似矩阵 $\bar{W}$, 其中

$$\bar{W}_{pq} = \sum_{\boldsymbol{x}_i \in c_p, \boldsymbol{x}_j \in c_q} W_{ij}$$

4) 令传播矩阵 $\bar{S} = D^{-\frac{1}{2}} \bar{W} D^{-\frac{1}{2}}$;

5) 迭代计算 $F(t+1) = \alpha \bar{S} F(t) + (1-\alpha)\bar{Y}$, 其中, $\alpha \in (0, 1)$, t 表示迭代次数;

6) 则 $\boldsymbol{x}_i \in c_p$ 的标号为 $y_p = \arg\max_{q \leq c} F^*_{pq}$, 其中 F^* 表示序列 $\{F(t)\}$ 的极限.

对样本集合 X 进行聚类的算法将在下一节中描述.

1.2 局部快速聚类

搜索样本空间中局部极值点的算法有很多, 如 Mean shift[13], Quick shift[14] 等, 结合半监督学习的特点, 我们提出了可以控制邻域大小的聚类算法. 该算法称为 Local quick shift, 具体描述如下:

设数据集 $X = \{\boldsymbol{x}_1, \boldsymbol{x}_2, \cdots, \boldsymbol{x}_n\}$, 其中, $\boldsymbol{x}_i \in \mathbf{R}^d$, 为了得到局部极值点, 我们计算

$$\boldsymbol{x}_i(1) = \arg \min_{j:P_j > P_i, \boldsymbol{x}_j \in \delta(\boldsymbol{x}_i, d)} D_{ij} \tag{5}$$

其中

$$P_i = \frac{1}{N} \sum_{j=1}^{N} \phi(D_{ij}) \tag{6}$$

D_{ij} 是样本之间的距离函数, ϕ 是密度估计的核函数, d 是邻域半径, $\boldsymbol{x}_i(1)$ 表示 $\boldsymbol{x}_i$ 所在簇的极值元素. 由于不需要计算梯度, 对 D_{ij} 与 ϕ 在形式上有多种选择, 本文使用欧氏距离和高斯核函数进行密度估计. 本质上该算法是一个局部区域内的搜索算法, 聚类后的每一簇都是一个树结构, 整个样本集合则被组织成森林的结构.

半监督学习与无监督学习的一个根本差异是标记样本点的使用. 在无监督的聚类分析中, 所有数据按照相同的方式进行处理, 数据是否处于相同的簇完全由聚类假设或者先验的混合分布决定; 然而在半监督学习中, 不同标记的已标记点不能处于同一簇中. 本文算法先对样本数据进行聚类的处理, 然后在聚类获得的簇的基础上使用基于图的半监督学习算法. 所以上面的 Local quick shift 虽然按照聚类假设生成了许多小的数据簇, 但是这些簇放在半监督学习的框架中, 其合理性还有待进一步确认. 在最基本的情况下, 它要求同一簇中至多包含一种标记点. 通过聚类后, 获得的簇按照它们包含的点是否标记的情况, 可以分为三种类型:

1) 不包含标记点;

2) 包含一种标记点;

3) 包含多种标记点.

对于第 1) 和 2) 两种情况, 本文认为是合法情况, 第 3) 种情况则为非法情况, 在聚类过程中形成的小簇必须保证其成员至多拥有单一的标记. 下面所列的算法以一种统一的方法处理上述三种情况, 在保证聚类假设的同时, 也保证半监督学习的附加要求. 图 1 和图 2 给出了拆分处理的图示.

算法 1. 用于半监督学习的 Local quick shift **算法**

初始化根节点集合 R 为空.

1) 估计密度. 对每个样本点 $\boldsymbol{x}$, 按式 (6) 估计其密度 p_i.

2) 建立森林. 对每个样本点 $\boldsymbol{x}_i$ 在其 ε 邻域内, 寻找一个最近的密度大于 p_i 的 $\boldsymbol{x}_j$. 若存在该点, 则连接 $\boldsymbol{x}_j\boldsymbol{x}_i$; 否则, 将点 $\boldsymbol{x}_i$ 加入根集合 R 中.

3) 树的拆分和标记. 对根集合 R 中的每个根节点 $\boldsymbol{x}_i$ 对应的树 T_i 按以下步骤遍历:

a) 设变量 B_{Ti} 为树 T_i 的标记. 若节点 $\boldsymbol{x}_i$ 已标记, 则 B_{Ti} 等于 $\boldsymbol{x}_i$ 的标记 b_i, 否则, $B_{Ti} = \text{NULL}$. 初始化当前节点 k 为 $\boldsymbol{x}_i$.

b) 对树 T_i 的当前节点 k

i) 当 $B_{Ti} = \text{NULL}$ 时

• 若 k 的子节点中不存在标记节点, 如图 2 (a), 则跳至 c);

• 若 k 的子节点中不存在一种标记节点 b, 如图 2 (b), 则 $B_{Ti} = b$;

• 若 k 的子节点中存在多种标记节点 b, 如图 2 (c), 则将所有已标记的子节点对应的子树从树 T_i 中拆分出来, 并将这些子节点依次加入根集合 K 中.

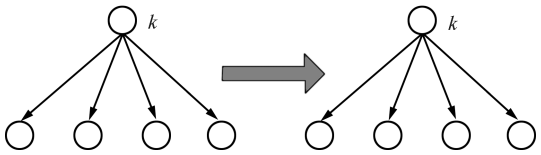
ii) 当 $B_{Ti} \neq \text{NULL}$ 时

• 若 k 的子节点中不存在标记节点, 如图 2 (a), 则跳至 c);

• 若 k 的子节点中不存在一种标记节点 b, 如图 2 (e) 和 2 (f), 则将标记不等于 B_{Ti} 的子节点对应的子树从树 T_i 中拆分出来, 并将这些节点依次加入根集合 R 中.

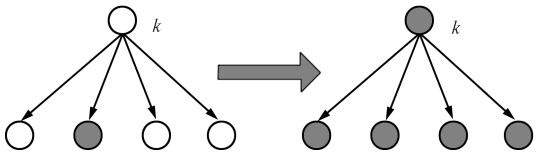
c) 按树层次遍历的方式将当前节点 k 更新为其下一个节点, 跳至 b); 若树 T_i 已遍历完毕, 则跳至 d).

d) 将当前树 T 更新为 R 中的下一棵树, 跳至 a). 若树 T_i 为 R 中的最后一棵树, 则算法停止.



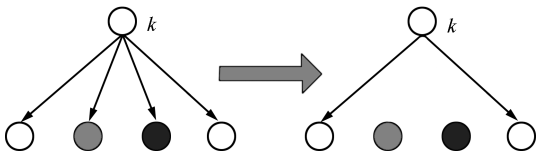
(a) 树节点未标记, 子节点未标记

(a) Unlabeled root and unlabeled leaves



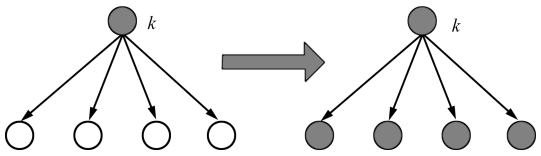
(b) 树节点未标记, 子节点存在一种标记

(b) Unlabeled root and one labeled leaf



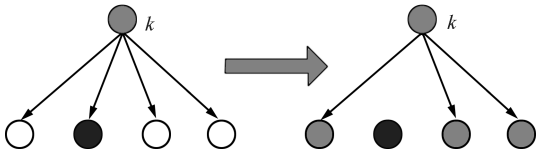
(c) 树节点未标记, 子节点存在多种标记

(c) Unlabeled root and multi-labeled leaves



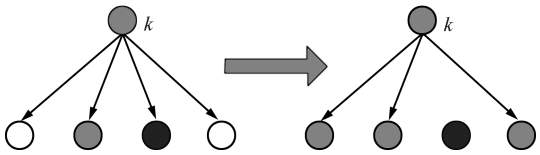
(d) 树节点已标记, 子节点不存在标记

(d) Labeled root and unlabeled leaf



(e) 树节点已标记, 子节点存在一种标记

(e) Labeled root and one leaf with other label



(f) 树节点已标记, 子节点存在多种标记

(f) Labeled root and multi-labeled leaves

图 1 树的拆分和标记过程中, 树标记和子节点标记的六种情况对应的处理

Fig. 1 Splitting and labeling process of six different kinds of trees

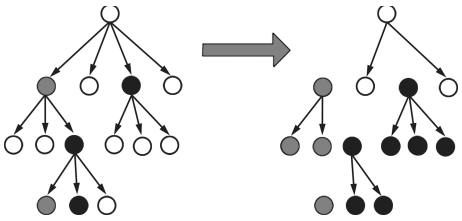


图 2 对树进行拆分和标记后的结果示意图

Fig. 2 Results of tree splitting and labeling

2 实验与分析

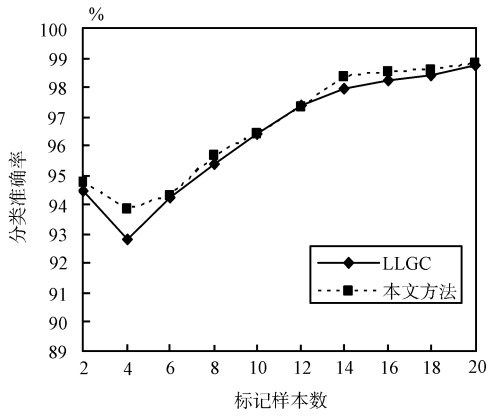
本节在不同的数据集上比较了 LLGC 和本文方法的分类准确率和速度. 为了使实验数据更具有说服力, 对于每个样本数目下的分类, 都独立重复 50 次实验, 取其平均准确率和平均计算时间作为算法的分类准确率与计算时间. 为了量化比较 LLGC 算法和本文算法的计算速度, 我们引入加速比的概念如下:

$$\text{加速比(Ratio)} = \frac{\text{LLGC算法时间}}{\text{本文算法时间}}$$

加速比表示了本文算法相对于 LLGC 算法的速度. 在只能获取少数标记样本的情况下, 并没有一个可靠的方法能寻找到最优的模型参数[3], 故本实验中 LLGC 算法和本文算法均分别取其最优参数. 本节所有实验使用 Weka 3.6.0 在 Intel Pentium 3.06 GHz, 512 MB 系统上进行.

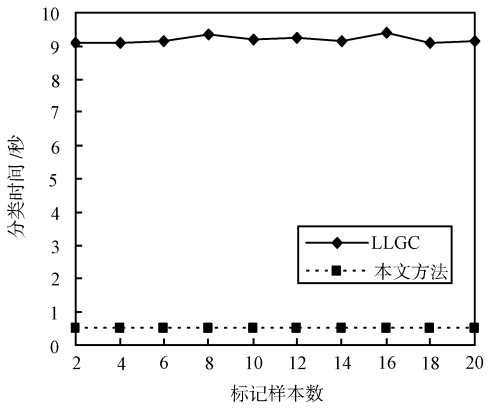
2.1 双月数据

本小节对双月数据[3] 进行分类, 该数据集中两类的样本数分别为 111 和 389, 共 500 个. 算法 LLGC 的参数设置为 $\sigma = 0.01$, $\alpha = 0.99$, 本文算法的参数为邻域半径 $\varepsilon = 0.18$, 聚类高斯函数方差 $\sigma_1 = 0.01$, 图方法距离函数方差 $\sigma_2 = 0.03$. 每次实验时, 从 365 个样本中随机采样 L 个样本作为标记实例, 剩下的样本为未标记实例; 采样应保证每类至少存在一个标记实例, 否则重采样. 在图 3 中给出了分类准确率和计算时间在各个标记样本数目下的曲线图, 其中横坐标表示标记样本数, 纵坐标分别表示分类准确率和分类时间.



(a) 分类准确率

(a) Accuracy



(b) 分类速度

(b) Time consumption

图 3 双月数据集上各算法的分类准确率和分类速度

Fig. 3 Accuracy and time consumption of the two algorithms on the double-moon data

从图 3 中可以看出, 在分类准确率方面, 本文算法略微高于 LLGC; 在分类速度方面, 本文算法明显优于 LLGC 算法, 其计算速度约为 LLGC 的 18 倍. 在表 1 中列出了该实验中本文算法的加速比.

表 1　本文算法的加速比 (L: 标记样本数目, Ratio: 加速比)

Table 1　Speedup ratio of the proposed algorithm (L: number of labeled data, Ratio: speedup ratio)

双月数据	L	2	4	6	8	10	12	14	16
	Ratio	17.17	17.46	17.88	18.33	17.63	18.16	17.56	18.06
UCI 字符数据	L	4	20	40	60	80	100	120	140
	Ratio	31.71	30.73	29.07	28.97	28.46	31.67	30.62	30.05

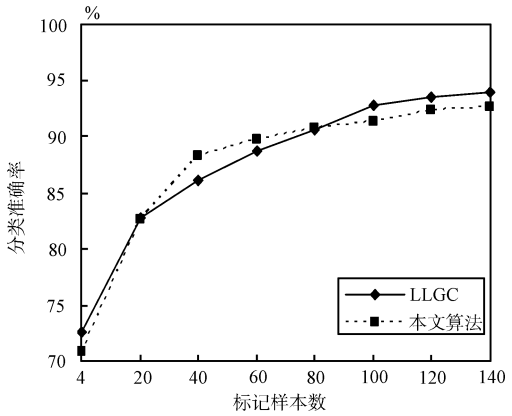
2.2　UCI 英文字符数据集

本小节实验选用 UCI 英文字符数据集[15]. 该数据集中包含了 26 个大写英文字符. 每个字符有 20 种字体. 该数据集中共有 20 000 个样本. 为了便于分类程序的处理, 对这 20 000 幅图像再次扫描, 每幅图像提取了 16 个数值型特征. 这些特征代表了每幅图像中像素分布的相关信息. 各特征维的详细描述可参考文献 [12]. 为了使数据紧密, 将每个属性维的数值范围压缩为 $[0, 15]$. 从样本集中选取 “A”、“B”、“C” 和 “D” 这 4 个字符, 其样本数分别为 789、766、736、805, 总共 3 096 个样本. LLGC 的参数为 $\sigma = 0.01$, $\alpha = 0.99$; 本文算法的参数为 $\varepsilon = 0.227$, $\sigma_1 = 0.11$, $\sigma_2 = 0.082$. 图 4 给出了分类准确率和分类时间在各标记样本数目下的曲线图.

从图 4 中可以发现, 在 “A”、“B”、“C”、“D” 这 4 类数据上, 本文算法在部分标记样本数目上的分类准确率略低于 LLGC 算法, 但是分类时间明显优于 LLGC 算法, 加速比达到了 30 倍左右. 在表 1 中列出了该实验中本文算法的加速比.

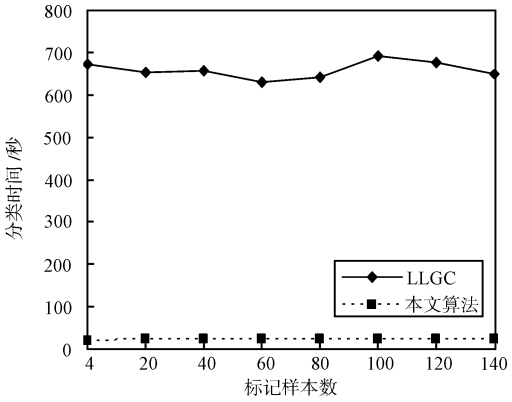
2.3　分析

LLGC 算法的计算时间复杂度为 $\mathrm{O}(n^3)$, 而局部聚类算法的计算时间复杂度为 $\mathrm{O}(n^2)$, 通过聚类原来的图规模能有效缩小, 所以本文方法的复杂度为 $\mathrm{O}\left((n/c)^3\right) + \mathrm{O}(n^2)$, 其中, c 表示图规模缩小的倍数, 也可以作为平均聚类程度, n 表示样本数目. 本文算法的一个优点是, 对于同一分布, 数据量越大, 局部聚类中每一簇包含的数据点就越多, 因此, c 的值越大, 加速效果越好. 这也意味着当 n 不断增加时, n/c 有可能趋于固定的取值范围, 从而使得样本点的增加只影响到局部聚类, 对图方法而言, 所要处理的簇的数量可以维持在一个较稳定的水平. 另一方面, 本文提出的局部聚类方法对分类准确率的影响非常稳定. 从分类准确率对比图上可以看到, 在不同标记样本数目下两条曲线的距离都很接近. 这是因为局部聚类和图方法遵守同样的聚类假设, 所以聚类后的分类效果始终与原始的 LLGC 算法相当.



(a) 字符 “A”, “B”, “C”, “D” 的分类准确率

(a) Accuracy on “A”, “B”, “C”, “D”



(b) 字符 “A”, “B”, “C”, “D” 的分类时间

(b) Time-consuming on “A”, “B”, “C”, “D”

图 4　字符 “A”、“B”、“C” 和 “D” 上各算法的分类准确率和分类时间

Fig. 4　Accuracy and time consumption of the two algorithms on letters “A”, “B”, “C”, and “D”

3　结论

计算复杂度较高是图方法最重要的缺点之一. 随着数据量的增加, 如果采用原始的样本数据与图节点一对一的方法构图, 建立的图将过于庞大, 以至于难以计算. 本文通过分析图方法的正则化理论模型, 提出了使用局部聚类进行图缩减的构图方法. 该方法产生的图通过新定义的距离函数保持了原始数据的流型结构, 使得分类效果与在原图上进行的分

类相当. 缩减过的图中一个节点可以代表多个数据, 因此极大地降低了图方法的计算量. 在后续工作中, 我们将进一步研究局部聚类与图方法相容性的理论分析, 以及随着数据量的增加, 缩略图的增长规模问题.

References

1 Blum A, Chawla S. Learning from labeled and unlabeled data using graph mincuts. In: Proceedings of the 18th International Conference on Machine Learning. Williamstorn, USA: Morgan Kaufmann Publisher, 2001. 19−26

2 Zhu X J, Ghahramani Z, Lafferty J. Semi-supervised learning using Gaussian fields and harmonic functions. In: Proceedings of the 20th International Conference on Machine Learning. Washington D. C., USA: Morgan Kaufmann Publisher, 2003. 912−919

3 Zhou D, Bousquen O, Lal T N, Weston J, Scholkoph B. Learning with local and global consistency. In: Proceedings of the Conference on Neural Information Processing Systems. Cambridge, USA: MIT Press, 2004. 321−328

4 Zhu X J, Lafferty J. Harmonic mixtures: combining mixture models and graph-based methods for inductive and scalable semi-supervised learning. In: Proceedings of the 22nd International Conference on Machine Learning. Bonn, Germany: ACM, 2005. 1052−1059

5 Zhu X J. Semi-Supervised Learning Literature Survey, Technical Report 1530, Computer Sciences, University of Wisconsin-Madison, USA, 2005

6 Zhou D, Scholkopf B. A regularization framework for learning from graph data. In: Proceedings of the 21st International Conference on Machine Learning. Banff, Canada: Morgan Kaufmann Publisher, 2004. 132−137

7 Delalleau O, Bengio Y, Roux N L. *Semi-Supervised Learning*. Cambridge: MIT Press, 2006. 87−96

8 Pfahringer B, Leschi C, Reutemann P. Scaling up semi-supervised learning: an efficient and effective LLGC variant. In: Proceedings of the 11th Pacific-Asia Conference on Knowledge Discovery and Data Mining. Nanjing, China: Springer, 2007. 236−247

9 Zhou Z H, Ng M, She Q Q, Jiang Y. Budget semi-supervised learning. In: Proceedings of the 13th Pacific-Asia Conference on Advances in Knowledge Discovery and Data Mining. Bangkok, Thailand: Springer, 2009. 588−595

10 Wang F, Zhang C S. Label propagation through linear neighborhoods. *IEEE Transactions on Knowledge and Data Engineering*, 2008, **20**(1): 55−67

11 Cheng H, Liu Z C, Yang L. Sparsity induced similarity measure for label propagation. In: Proceedings of the 12th IEEE International Conference on Computer Vision. Kyoto, Japan: IEEE, 2009. 317−324

12 Zhou Zhi-Hua, Wang Jue. *Machine Learning and Its Applications*. Beijing: Tsinghua University Press, 2007. 259−275 (周志华, 王珏. 机器学习及其应用. 北京: 清华大学出版社, 2007. 259−275)

13 Cheng Y Z. Mean shift, mode seeking, and clustering. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 1995, **17**(8): 790−799

14 Vedaldi A, Soatto S. Quick shift and kernel methods for mode seeking. In: Proceedings of the 10th European Conference on Computer Vision. Marseille, France: Springer, 2008. 705−718

15 Frey P W, Slate D J. Letter recognition using Holland-style adaptive classifiers. *Machine Learning*, 1991, **6**(2): 161−182

李 明 华中科技大学计算机学院博士研究生. 主要研究方向为图像处理和机器学习. 本文通信作者.
E-mail: liming_lm_ing@sina.com
(**LI Ming** Ph. D. candidate at the School of Computer Science and Technology, Huazhong University of Science and Technology. His research interest covers image processing and machine learning. Corresponding author of this paper.)

杨艳屏 华中科技大学计算机学院博士研究生. 主要研究方向为图像处理和机器学习.
E-mail: yangyanping07@gmail.com
(**YANG Yan-Ping** Ph. D. candidate at the School of Computer Science and Technology, Huazhong University of Science and Technology. Her research interest covers image processing and machine learning.)

占惠融 华中科技大学计算机学院硕士研究生. 主要研究方向为机器学习和网页搜索.
E-mail: zhanhuirong@baidu.com
(**ZHAN Hui-Rong** Master student at the School of Computer Science and Technology, Huazhong University of Science and Technology. Her research interest covers machine learning and page searching.)