考虑局部均值和类全局信息的快速近邻原型选择算法

李 娟 [1, 2] 王宇平 [1]

摘 要 压缩近邻法是一种简单的非参数原型选择算法, 其原型选取易受样本读取序列、异常样本等干扰. 为克服上述问题, 提出了一个基于局部均值与类全局信息的近邻原型选择方法. 该方法既在原型选取过程中, 充分利用了待学习样本在原型集中 k 个同异类近邻局部均值和类全局信息的知识, 又设定原型集更新策略实现对原型集的动态更新. 该方法不仅能较好克服读取序列、异常样本对原型选取的影响, 降低了原型集规模, 而且在保持高分类精度的同时, 实现了对数据集的高压缩效应. 图像识别及 UCI (University of California Irvine) 基准数据集实验结果表明, 所提出算法集具有较比较算法更有效的分类性能.

关键词 数据分类, 原型选择, 局部均值, 类全局信息, 自适应学习

引用格式 李娟, 王宇平. 考虑局部均值和类全局信息的快速近邻原型选择算法. 自动化学报, 2014, **40**(6): 1116−1125

DOI 10.3724/SP.J.1004.2014.01116

A Fast Neighbor Prototype Selection Algorithm Based on Local Mean and Class Global Information

LI Juan[1, 2] WANG Yu-Ping[1]

Abstract The condensed nearest neighbor (CNN) algorithm is a simple non-parametric prototype selection method, but its prototype selection process is susceptible to pattern read sequence, abnormal patterns and so on. To deal with the above problems, a new prototype selection method based on local mean and class global information is proposed. Firstly, the proposed method makes full use of those local means of the k heterogeneous and homogeneous nearest neighbors to each be-learning pattern and the class global information. Secondly, an updating process is introduced to the proposed method. Lastly, updating strategies are adopted in order to realize dynamic update of the prototype set. The proposed method can not only better lessen the influence of the pattern selected sequence and abnormal patterns on prototype selection, but also reduce the scale of the prototype set. The proposed method can achieve a higher compression efficiency that can guarantee the higher classification accuracy synchronously for original data set. Two image recognition data sets and University of California Irvine (UCI) benchmark data sets are selected as experimental data sets. The experiments show that the proposed method based on the classification performance is more effective than the compared algorithms.

Key words Data classification, prototype selection, local mean, global class information, adaptive learning

Citation Li Juan, Wang Yu-Ping. A fast neighbor prototype selection algorithm based on local mean and class global information. *Acta Automatica Sinica*, 2014, **40**(6): 1116−1125

在机器学习和数据挖掘任务中, 作为一种简单成熟的分类算法 KNN (k-nearest neighbors algorithm)[1] 获得广泛的应用. 作为数据挖掘领域的十大经典算法之一, KNN 算法具有理论简单、易于实现、无需预先训练分类器、可适用各种数据分布环境等优势, 然而尤其处理大规模数据集时, 由于其简单的处理策略而导致产生难以接受的时间和空间消耗. 故在分类算法中, 如何对大规模数据集去除冗余节点, 保留高效分类贡献的代表点, 进而降低数据规模、提高分类速度, 成为了研究热点.

为此, 一种有效的处理策略即原型选择就是对原始数据集进行必要缩减, 即在保证不降低甚至提高分类精度等性能的基础上, 对原始训练集处理从中获取能够反映原始数据集分布及分类特性的代表样本集即原型集, 进而降低数据规模和噪音的敏感度, 提高分类算法执行效率.

收稿日期 2013-06-19 录用日期 2013-11-11
Manuscript received June 19, 2013; accepted November 11, 2013
国家自然科学基金 (61272119) 资助
Supported by National Natural Science Foundation of China (61272119)
本文责任编委 章毓晋
Recommended by Associate Editor ZHANG Yu-Jin
1. 西安电子科技大学计算机学院 西安 710071 2. 陕西师范大学远程教育学院 西安 710062
1. School of Computer Science and Technology, Xidian University, Xi′an 710071 2. School of Distance Education, Shaanxi Normal University, Xi′an 710062

1 相关技术

1.1 原型选择算法

原型选择算法的重要应用之一是作为某个分类算法的预处理步骤, 可与各种分类算法相结合, 降低分类算法的数据规模. 本文选定原型选择算法与近邻算法相结合, 通过分类精度比较所提出算法的执行效率. 原型选择算法目标为在不降低分类性能的基础上, 去除噪音等异常节点, 降低训练集规模, 进而提高算法执行效率. 其常见模型[2] 为: 设 TR (Training set) 为训练集 (包含一些无用信息, 如噪音、冗余信息等), 寻求选择子集 TP (Training prototype set), $TP \subset TR$ 使得 TP 不包含多余原型, 且 $Acc(TP) \cong Acc(TR)$ ($Acc(X)$ 表示 X 作为训练集所获得的分类精度). 而在分类过程中, 使用 TP 代替 TR 作为分类判断基准数据, 从而降低了运算的数据规模.

原型选择算法一经提出, 就获得了长足的发展, 产生了诸多的研究成果. 其中剪辑近邻法 (Edited nearest neighbor, ENN)[3] 与压缩近邻法 (Condensed nearest neighbour, CNN)[4] 是较早提出的样本选择算法. CNN 算法的缺点是对原样本集样本的排列顺序敏感, 而且压缩集中含有较多的冗余样本. 围绕着 CNN 算法, 产生了一系列改进算法: 如 FCNN (Fast condensed nearest neighbor)[5] 侧重降低样本读取序列敏感性和尽可能获取类决策边界原型; GCNN (Generalized condensed nearest neighbor)[6] 引入了同异类近邻, 克服了 CNN 仅使用同类近邻的不足; MNV (Mutual neighborhood value)[7] 使用互近邻值降低算法样本读取序列敏感性; RNN (Reduced nearest neighbor rule)[7] 侧重于改进 CNN 算法原型集不能删除的缺陷等; 基于聚类策略的类边界样本选择算法, 如 IKNN (Improved k-nearest neighbor classification)[8] 和 PSC (Prototype selection by clustering)[9] 等. 上述算法仍然具备算法对噪音的敏感性. 通常压缩近邻法即剪辑法, 通过去除噪声点和清理不同类别重叠区的样本点来达到代表点选择的目的. 编辑法主要采取剔除原始样本集中的噪音等策略, 是一种非增量算法, 不适用于大规模数据集处理.

为此, 如何降低传统增量原型选择算法对样本读取序列、异常点敏感, 成为增量原型选择算法的研究热点, 同时也是本文研究的主要问题.

1.2 局部均值或类均值分类算法

针对 KNN 算法的噪音敏感性及传统只关注近邻样本忽略其样本分布等弊端, 很多研究者考虑了近邻局部均值或类均值信息与样本分布的关系, 将近邻局部信息和类统计或均值信息纳入到近邻分类算法中. 其中 Mitani 等[10] 提出了一种基于局部均值的非参数分类方法, 克服离群点对分类性能的影响, 尤其在小样本情形下分类性能较好. Brown 等[11] 使用了各自类近邻类样本距离加权信息进行分类, 区别于文献 [10] 中样本集距离加权信息; Han 等[12] 引入了类中心思想, 充分利用训练样本的整体信息分类; 在此基础上, Zeng 等[13] 提出了基于局部均值和类均值的分类算法, 既利用未分类样本在每类里的近邻局部均值信息, 又利用类均值的整体知识进行分类; 而 Brighton 等[14] 则定义了待学习样本的 Reachable 和 Coverage 概念, 在此基础上同 ENN 算法相结合, 提出了迭代样本过滤算法 (The iterative case filtering, ICF), Wang 等[15] 对其进行改进, 提出了 ISSARC (An iterative algorithm for sample selection based on the reachable and coverage) 算法.

设置不同的参数, 基于均值的分类方法可退化传统最近邻方法. 当选择待分类样本在每类训练样本集里的近邻数为 1 时, 则该局部均值方法等价于最近邻分类; 当选取近邻数等于对应类的训练样本数时, 则等价于欧几里得距离分类[7].

综上, 在传统原型选择算法中, 借鉴样本局部近邻均值和全局均值等信息, 可进一步贴合原型集分布状态, 降低了异常原型的干扰. 本文在传统 CNN 算法基础上, 利用近邻局部均值和类全局信息, 同时借鉴 RNN 样本删除思想, 提出一种新的原型选择算法 (An improved nearest neighbor prototype selection algorithm based on local-mean and class global information, LCNN), 可在保障不降低甚至提高分类效率基础上, 较好克服 CNN 及其改进算法对样本读取序列的依赖性, 提升原型集的动态更新能力、降低算法的噪音敏感性.

2 考虑局部均值和类全局信息的近邻原型选择算法

为便于描述, 本文使用以下符号: 记任意数据集 $D = \{x_i = (x_{i1}, x_{i2}, \cdots, x_{id}) \,|i = 1, 2, \cdots\}$, 类标记集 $C = \{c_1, c_2, \cdots, c_m\}$, d 为样本维度, m 为类别数. 记 $TR = \{(x_i, y_i)|x_i \in D,\ y_i \in C,\ i = 1, 2, \cdots, n\}$ 为训练集; $TP \subset TR$ 为训练所得原型集; TS (Testing set) 与 TR 同构, 为若干样本的测试集.

设 $TP \neq \emptyset$, 记任一待扫描学习样本 $x \in TR$, 任一原型 $p \in TP$; $s_{kx} = S_k(x) \subset TP$ 为 x 同类别的 k 个近邻原型; $h_{kx} = H_k(x) \subset TP$ 为 x 异类别的 k 个近邻原型; $d(x, y)$ 为 x 与 y 间欧氏距离; $label\,(x)$ 表示样本 x 的类别; $D(x) = \sum_{i=1}^{k} w_i d(x, x^i)$ (其中 w_i 为 x 的第 i 近邻原型的距离加权系数, x^i 表示 x 的第 i 近邻原型) 为 x 的 k 近邻加权距离和, 即本文所定义的局部均值信息; $Ind\,(p)$ 表示原型 p 在 TP 中对应索引; PS 为四元组结构, 用以表示 p 及其同异类最近邻原型关系, 其中 $PS\,(1)$、$PS\,(2)$、$PS\,(3)$ 分别表示 TP 中 p 索引、p 同类最近邻索引和 p 异类最近邻索引, $PS\,(4)$ 表示 p 是否被删除的标识.

2.1 LCNN 算法策略

在 CNN 算法基础上产生了诸多的改进算法. 但这些算法仅利用所筛选的 k 近邻样本类别信息, 未考虑到样本分布等数据集的局部或全局信息, 易受近邻样本偏好影响; 同时仍保留着 CNN 算法的样本读取序列及噪音的敏感性; 且较少涉足对原型集样本的动态增删操作, 使噪音和孤立点等样本得以延续保存.

为此, 本文对 CNN 算法进行必要改进, 其处理策略如下: 1) 去除 CNN 算法无指导的新类别原型获取策略, 新增初始化操作, 主要完成类全局信息的获取和所有类别初始原型的获取, 以类全局信息调控噪音和孤立点能否成为原型; 2) 针对 CNN 算法中仅使用最近邻样本而导致易受样本读取序列和噪音干扰的情况, 扩充最近邻样本为 k 同类近邻样本和 k 异类近邻样本, 使用同类近邻均值和异类近邻均值信息作为原型初步判断条件, 可有效降低 CNN 算法噪音敏感度; 3) 在预设的更新周期内, 使用局部均值及类全局信息完成对孤立点原型、类中心区域原型等删除操作, 进而对原型集信息进行针对性更新.

图 1 显示了 LCNN 算法运行框图, 其中虚线框部分为本文研究的主要内容, 即实现原型集选择功能; 而分类算法构造分类器部分, 本文选择了最近邻分类用以检验 LCNN 所产生原型集的性能.

令 $N_1, N_2, \cdots, N_m$ 表示 TP 中对应于类别 $c_1, c_2, \cdots, c_m$ 的原型个数. 设 x 有 k 个有效可取的同异类原型, s_{kx} 为测试样本 x 获取 TP 中 k 个近邻同类原型, 那么 x 同类局部均值为:

$$D_s = \sum_{i=1}^{k} w_i d(x, s_{kx}^i) \tag{1}$$

同理, h_{kx} 为测试样本 x 获取 TP 中 k 个近邻异类原型, x 的异类原型局部均值为:

$$D_h = \sum_{i=1}^{k} w_i d(x, h_{kx}^i) \tag{2}$$

对于 TP 中属于类别 $c_j (j = 1, 2, \cdots, m)$ 的原型表示为 $TP_j = \{p_j^i | i = 1, 2, \cdots, N_j\}$, 那么 c_j 类的全局均值原型为:

$$G_j = \frac{1}{N_j} \sum_{i=1}^{N_j} p_j^i \tag{3}$$

对于类别 c_j 原型与类均值原型的平均距离为:

$$D_j = \frac{1}{N_j} \sum_{i=1}^{N_j} d(p_j^i, G_j) \tag{4}$$

综上, 以类别为整体的均值原型及平均距离都属于 c_j 的全局信息, 故定义了 $GD =< GD_1, GD_2, \cdots, GD_m >$ 为 TP 各类的全局信息结构, 其中 $GD_j(1)$ 表示类均值原型向量, 用来存储 TP 各类别的动态中心, 即存储 G_j; $GD_j(2)$ 表示类原型间平均距离, 用来存储 TP 各类别原型间的动态平均距离, 即存储 D_j. 本文中 GD 被称为 TP 类全局信息.

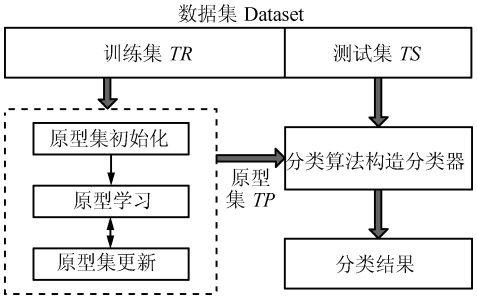

图 1 LCNN 算法运行框图

Fig. 1 Running diagram of LCNN algorithm

2.2 算法主要处理过程

原型集初始化过程、原型学习过程和原型更新过程是 LCNN 算法的核心内容. 其中, 初始化过程采取随机比例的样本读取获取类全局信息, 根据类全局信息有指导性选取原型集初始化, 降低 CNN 算法原型无指导选取的随机性影响; 学习过程在一定学习策略下, 实现对原型集有效增添; 更新过程, 设置了不同的更新阀值, 通过周期性删除 TP 中不符合条件的原型, 完成 TP 集的动态更新, 进而去除类中心原型、孤立点及噪音, 较好克服传统 CNN 算法只增加、不删除原型的弊端.

2.2.1 原型集初始化过程

原型集初始化过程包含两个功能: 1) 通过随机提取各类训练样本, 获取类样本的平均距离、类均值中心节点的全局信息, 作为原型集初始原型选择依据; 2) 在类全局信息指导下, 为每类样本随机选取 f (本文一般设置 $f=2$, 当不平衡数据集时, $f=1$) 个初始原型加入 TP, 降低了 CNN 算法新类别原型选取的随机性; 同时获取并填充 TP 中各原型的同异类近邻节点, 更新类均值中心节点.

LCNN 的原型集初始化过程描述如下:

输入. 训练样本集 TR.

输出. GD、PS.

步骤 1. 初始化 $GD=\emptyset$, $PS=\emptyset$.

步骤 2. 随机从 TR 中读取一定比例的训练样本, 完成 GD 信息的填充.

步骤 3. $i=1$.

步骤 4. $j=1$.

步骤 5. 从第 i 类样本中读取任一样本 x, 若其满足 $GD\,(i,2)<d\,(x,GD(i,1))<3\times GD\,(i,2)$, 加入到原型集, $j=j+1$.

步骤 6. 若 $j<f$, 转到步骤 5.

步骤 7. $i=i+1$, 若 $i<m$, 转到步骤 4.

步骤 8. 逐类别逐原型完成对 GD 和 PS 的数据填充.

步骤 9. 输出 GD、PS.

2.2.2 原型学习过程

LCNN 算法是个增量学习算法, 整个算法单遍扫描训练样本集, 从读取第一个未被扫描样本开始, 直至所有待学习样本学习完毕, 获取最终原型集. 当一个样本的同类近邻局部均值大于样本的异类近邻局部均值时, 该样本被选作原型加入到原型集; 同时, 判断样本与其类中心点距离是否大于最近邻与类中心点距离, 如大于则将其选作原型加入到原型集.

LCNN 的原型学习过程描述如下:

输入. GD、PS、λ 及 TR.

输出. TP.

步骤 1. 如 TR 不存在未被扫描样本, 则输出 TP, 结束算法.

步骤 2. 任取一未被扫描样本 x.

步骤 3. 根据 x 的类别信息 c, 获取 x 的 s_{kx}、h_{kx}、$GD\,(c,:)$.

步骤 4. 若 $d\,(x,GD(c,1))<GD\,(c,2)$, 转到步骤 8.

步骤 5. 使用式 (1) 和式 (2) 分别计算 x 的同异类近邻局部均值 D_s 和 D_h.

步骤 6. 若 $D_s>D_h$, x 被选作原型加入 TP, 同步设置 $PS\,(x,:)$ 数据, 转到步骤 8.

步骤 7. 若 $d\,(x,GD(c,1))>d\,(s_{kx}^{1},GD(c,1))$, x 被选作原型加入 TP, 同步设置 $PS(x,:)$ 数据.

步骤 8. 若已学习样本数是 λ 的整数倍, 则调用更新过程.

步骤 9. 否则, 转到步骤 1.

LCNN 算法突破了 CNN 算法仅使用最近邻判别原型的简单方式, 考虑到原型样本分布等因素, 引入训练样本 x 的同异类局部均值, 通过局部均值信息、类均值中心间关系作为判断原型的依据, 即克服最近邻原型判别准则的偏好, 在一定程度上实现了类边界原型的选取. 同时减少了与类中心距离过近原型的添加, 稀疏化类中心区域原型个数.

2.2.3 原型集更新过程

更新过程引入了原型删减思想, 每 λ 个样本学习后, 调用原型集更新过程, 定期删除不符合规则的原型, 减少原型数目. 本文依托类全局信息和原型的最近同异类近邻设定不同的更新阀值, 用以处理不同情况的原型删除操作. 对于任一 $p_i\in TP$, c_j、c_s 分别为 p_i 与最近邻异类原型类别, 执行两步骤更新操作; 待 TP 原型扫描完毕, 执行局部均值及类全局信息更新操作.

LCNN 的原型集更新过程描述如下:

步骤 1 (孤立原型更新). 当 $d\,(p_i,GD(c_j,1))\geq 3\times GD(c_j,2)$ 且 $d\,(p_i,TP(PS(Ind(p_i),2)))>d\,(p_i,TP(PS(Ind(p_i),3)))$ 表明该类原型为孤立点, 删除此类原型, 可降低孤立原型影响, 则设置 $PS(Ind(p_i),4)=1$.

步骤 2 (噪音等异常原型更新). 当 $GD(c_s,2)\leq d\,(TP(PS(Ind(p_i)),3)),GD(c_s,1))$ 且 $3\times GD(c_s,2)>d\,(TP(PS(Ind(p_i)),3)),GD(c_s,1))$ 时, 利用 p_i 相关局部均值和类全局信息进行判断. 若 p_i 的同类局部均值小于它的异类局部均值 $(D_s<D_h)$, 同时 p_i 异类原型处于非类边缘区域 (即 $d\,(TP(PS(Ind(p_i),3)),GD(c_j,1))>d\,(TP(PS(Ind(p_i),3)),GD(c_s,1))$ 或 $d\,(TP(PS(Ind(p_i),3)),GD(c_j,1))>d\,(TP(PS(Ind(p_i),1)),GD(c_s,1))$, 则表示 p_i 为噪音, 则设置 $PS\,(Ind(p_i),4)=1$.

步骤 3 (局部均值及类全局信息更新). 原型扫描完毕, 对所有更新标识的原型进行删除; 更新原型集 TP 的 PS 结构信息; 最后分类别计算类均值中心 $\widetilde{G}_j$ 和类原型标准差距离 $\widetilde{D}_j$, 更新 $GD(c_j,1)=\widetilde{G}_j$

和 $GD\left(c_j,2\right)=\widetilde{D}_j$.

2.3 关键概念及参数界定

1) 孤立原型界定: 本文采用文献 [16] 的定义, 把孤立原型定义为与类原型均值的距离超过 3 倍标准差距离的原型.

2) 近邻权重选取: 本文选取了最简单的倒数距离加权参数, 即 $w_i=1/i$, w_i 随着 i 的增加而减小, 对应的原型对新原型选取的影响越小. 在未考虑全局信息情况, 若近邻数为 1, 即对待学习样本只选取一个同类近邻和一个异类近邻, 则局部均值学习退化为传统 CNN 学习.

LCNN 运行必须两个参数支撑: 一是原型近邻数 k; 二是更新周期 λ. 两种参数选取有预设、交叉验证和动态调整三种方式. 其中预设和动态调整方式简单便捷, 交叉验证方式需要多次验证运行才能获取较好的参数配置. 因此, 结合原型集增量生成方式, 选择了动态调整设置方式, 即伴随着原型集的动态变化, 动态调整 k 和 λ. 为简化问题, 本文在一个更新周期 λ 内, 将不同类别样本 x 在 TP 中各同异类近邻数设置为相同, 且 $k\leq\min(\overline{N}_1,\overline{N}_2,\cdots,\overline{N}_m)$. 本文分别选取 $k=\left\lceil\sqrt{\min\limits_{j=1}^{m}\overline{N}_j}\right\rceil$, $\lambda=\left\lceil\sqrt{\sum\limits_{j=1}^{m}\overline{N}_j}\right\rceil$ ($\overline{N}_j$ 表示更新周期开始时 TP 中类别 $c_j(j=1,2,\cdots,m)$ 原型数, $\lceil\cdot\rceil$ 表示向上取整).

3 算法评估

为了更好评估 LCNN 算法的性能指标, 本文选择了 KNN、CNN、GCNN、PSC、ISSARC 以及 ILVQ (Incremental learning vector quantization)[17] 作为比较算法. 其中 LCNN 与 GCNN 处理策略相似, 均采取了同异类近邻思想, 本文中 GCNN 选取了 $\rho=0,0.1,0.25,0.5,0.75,0.99$ 下的平均运行效率; PSC 主要思想是以空间划分策略尽可能获取类边界原型, 本文选取文献 [9] 中获取最佳运行效率的 $r=6\,\mathrm{m}$ 和 $r=8\,\mathrm{m}$; ISSARC 算法是在 ICF 算法基础上进行改进的非增量原型选择算法, 主要思想是考虑同异类近邻距离的限定, 同时通过去除噪音的非增量 ENN 算法的预先处理, 降低了算法噪音敏感度, 其 ENN 算法运行采取了文献 [15] 中的参数设置; 而 LCNN 也以获取类边界原型为处理目标; ILVQ 是目前高压缩性的快速的增量原型生成算法之一 (为简化 ILVQ 运行, 本文对 λ 和 $Ageold$ 采取简单预设 $\lambda=Ageold=\lceil\sqrt{n}\rceil$); LCNN 单遍扫描训练集, 也体现了快速增量原型选择思想; 而选择 KNN 和 CNN 则作为比较算法分类性能的参照, 其中 KNN 算法预设 5 个常见的 k 值, 分别为 3、5、7、9、11.

为了验证 LCNN 算法的有效性, 选择了两个图像识别数据集以及其他 12 个 UCI 数据集 (见表 1) 和 3 个大规模数据集[18], 采用 5 次 5 折交叉验证获得对比算法的平均分类效率及分类速度. 本文在奔腾 IV Intel (R) Core (TM)2 Du CPU E 8300 2.83 GHz 1G 的 PC 硬件支撑, Windows XP 32 位及 Matlab 7 运行环境下获取实验数据. 本文中采用分类精度 $=\frac{|TS_{correct}|}{|TS|}\times100\%$、压缩比率 $=\frac{|TP|}{|TR|}\times100\%$、运行时间 (单位: 秒) 作为比较算法的评价指标. 其中 $|TS_{correct}|$ 表示 TS 在 TR 或 TP 作为训练集下被正确分类的样本数, $|TR|$、$|TP|$、$|TS|$ 分别表示 TR、TP、TS 所包含的样本或原型数.

表 1 UCI 基准数据集信息

Table 1 The information of UCI benchmark data sets

数据集	特征数	类别数	样本数
Iris	4	3	150
Wine	13	3	178
Glass	9	6	214
Ionosphere	34	2	351
Cancer	9	2	699
Zoo	16	7	101
Heart	13	2	270
TAE	5	3	151
Liver disorders	6	2	345
Spectf	44	2	267
Ecoli	7	8	336
Ctg	20	3	2 126

3.1 理论分析

分析比较算法, 其中 KNN 算法的时间复杂度为 $\mathrm{O}(dn^2n_i)$, CNN 算法时间复杂度为 $\mathrm{O}(nN^2d+n_1N^2d)$, GCNN 算法时间复杂度为 $\mathrm{O}(n^2Nd+n_1N^2d)$, PSC 算法时间复杂度为 $\mathrm{O}(\tau rnd+n_1N^2d)$, ISSARC 算法时间复杂度为 $\mathrm{O}(n^3d+d\sum\limits_{i=1}^{t}M_i^2+n_1N^2d)$, ILVQ 算法时间复杂度为 $\mathrm{O}(dnN+n_1N^2d)$. LCNN 算法是增量学习算法, 主要分为两部分: 增量原型生成时间 $\mathrm{O}(dnN)$ 和原型分类时间 $\mathrm{O}(n_1N^2d)$, 其整体时间复杂度为 $\mathrm{O}(dnN+n_1N^2d)$. 上述公式中, n 为训练样本数, d 为样本维度, N 为最终原型数, r 为聚类数, τ 为聚

类迭代次数, n_1 为测试样本数, t 为 ISSARC 算法迭代周期, $\sum_{i=1}^{t} M_i$ 为 ENN 算法运行所得原型集规模. LCNN 算法对于所有的训练样本而言是近线性的, 但后续原型分类所需时间复杂度为传统的近邻分类算法时间复杂度. LCNN 算法是一种增量算法, 仅在原型生成过程中执行单遍样本扫描, 并不需对训练集进行存储, 因此, LCNN 具有处理大规模数据集的能力.

3.2　人工数据实验

为验证 LCNN 算法在大规模数据集的处理性能, 本文选择文献 [17] 实验所使用的人工数据集进行增量的原型分类比较. 图 2 和图 3 均为 2 维人工数据集: 图 2 中包含 5 类数据, 类别 1 和类别 2 满足 2 维高斯分布, 类别 3 和类别 4 数据分布为 2 个同心圆, 类别 5 满足正弦分布; 图 3 在图 2 有效数据分布的基础上, 加入了 20 % 的均匀分布噪音将其随机分布到 5 个类别中.

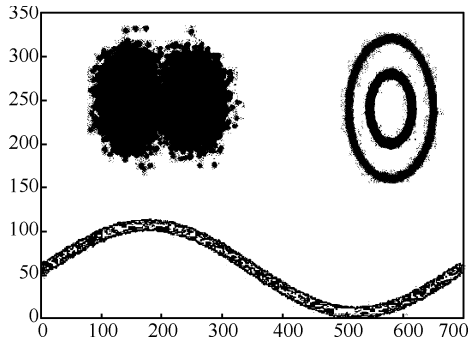

图 2　无噪音人工数据集

Fig. 2　No noise artificial data set

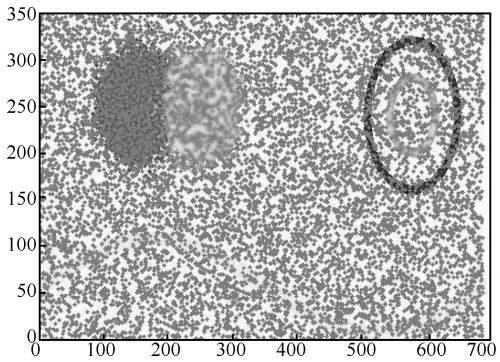

图 3　含噪音的人工数据集

Fig. 3　Noise artificial data set

区别于文献 [17] 实验中多种样本读取序列和不同迭代次数, 本文采取单遍随机样本读取序列的简单方式. 除选择三种增量算法外, 由于 ISSARC 算法通过 ENN 算法对噪音等异常数据进行了预先处理, 提高了算法的抗噪能力, 故而选择其作为对照算法. 图 4、图 6、图 8、图 10 为四种算法在图 2 数据集上原型生成情况, 可以看出, LCNN 算法在保持原始样本集分布的情况, 对其进行必要缩减, 其结果同 ISSARC 和 ILVQ 算法结果具有可相较性. 图 5、图 7、图 9、图 11 为 4 种算法在图 3 数据集上原型生成情况, LCNN 算法除原型个数明显少于 ILVQ 算法结果外, 相对于 ISSARC 算法而言, 在一定程度上降低了噪音的敏感性, 其噪音数据数量明显少于比较算法.

其中, ISSARC 算法属于非增量算法, 在人工数据实验中的运行时间消耗达 10 小时以上.

3.3　图像识别对比

1) 医学图像诊断识别

为验证 LCNN 的实用性能, 本文选取了 569 幅乳腺癌症图像数据进行实验, 该数据将每幅乳腺癌症图像提取 30 个维度详细描述, 其中 212 个异常图像, 357 个正常图像. 通过 5 次 5 折交叉验证得到比较算法的平均运行数据. 表 2 数据表明 LCNN 在癌症图像识别中有着明显的压缩、分类精度及运行时间优势, 是一种可行性的原型选择算法.

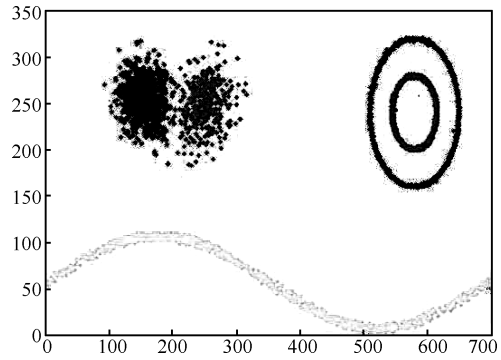

图 4　CNN 在无噪音数据集的原型集

Fig. 4　The prototype set obtained by CNN on no noise data set

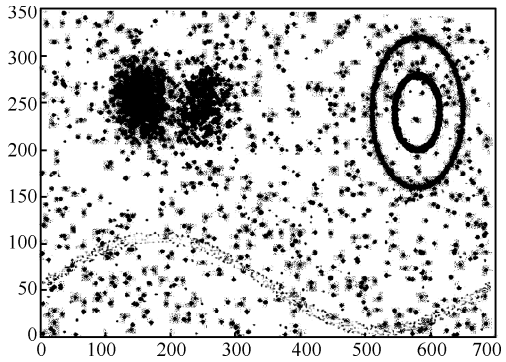

图 5　CNN 在噪音数据集的原型集

Fig. 5　The prototype set obtained by CNN on noise data set

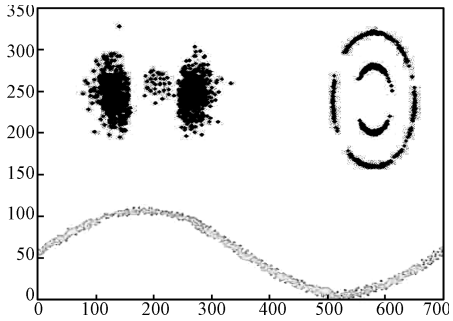

图 6 ISSARC 在无噪音数据集的原型集

Fig. 6 The prototype set obtained by ISSARC on no noise data set

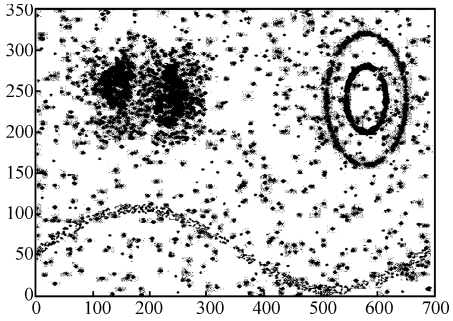

图 7 ISSARC 在噪音数据集的原型集

Fig. 7 The prototype set obtained by ISSARC on noise data set

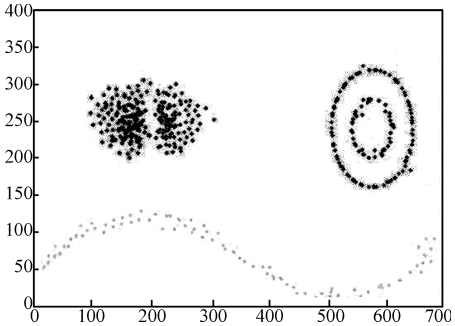

图 8 ILVQ 在无噪音数据集的原型集

Fig. 8 The prototype set obtained by ILVQ on no noise data set

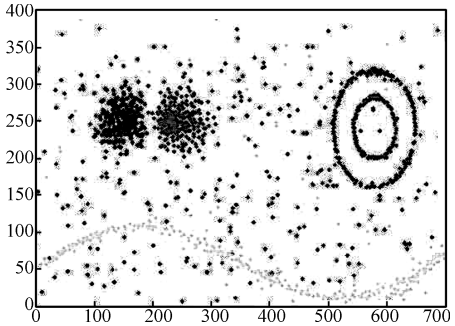

图 9 ILVQ 在噪音数据集的原型集

Fig. 9 The prototype set obtained by ILVQ on noise data set

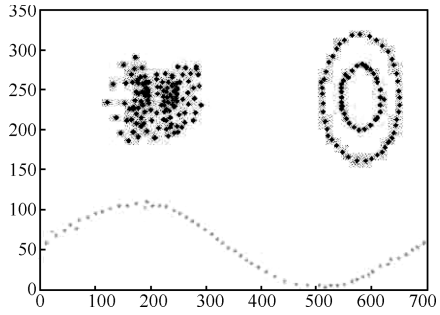

图 10 LCNN 在无噪音数据集的原型集

Fig. 10 The prototype set obtained by LCNN on no noise data set

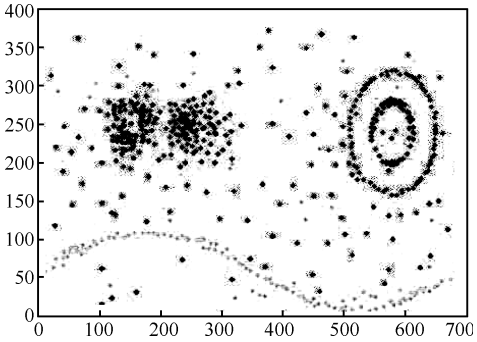

图 11 LCNN 在噪音数据集的原型集

Fig. 11 The prototype set obtained by LCNN on noise data set

2) 数字手写体识别

为进一步验证 LCNN 实际问题解决能力, 特选择了研究文献中常用的手写体数字光学数据集进行算法的比较. 该数据集含 0 到 9 阿拉伯手写体数字的 3 823 个训练图像信息和 1 797 个测试图像信息. 表 3 数据获取环境同表 2.

表 3 数据显示 LCNN 较其他比较算法有着一致好的运行效率. 其中 CNN 算法因运算简单且无需进行原型集的删除等操作, 所以运行时间较少; PSC 需要较大的运行开销来完成初始的聚类操作; GCNN 因需要动态计算 δ 而增加一定运行时间消耗; ILVQ 算法增加了原型周期动态更新操作, 运行消耗较大. ISSARC 算法虽然保持最好的压缩比率, 然而由于其自身调用 ENN 算法的预先处理策略, 增加了 ISSARC 算法的运行时间消耗. 综上, 采取 LCNN 算法解决实际问题, 可有效降低数据规模, 可配合其他高效分类算法更好地发挥其优势.

3.4 UCI 基准数据集实验

除上述图像识别数据集外, 为更全面验证算法有效性, 本文选择的 12 个中小规模和 3 个大规模 UCI 基准数据集, 较全面涵盖了数据集的维度规模和样本规模多样化分布, 实验环境同上.

表 2 比较算法在乳腺癌数据集上的运行效率

Table 2 Operational efficiency results obtained by compared algorithms on breast cancer data set

算法	KNN	CNN	GCNN	PSC	ISSARC	ILVQ	LCNN
分类精度	93.67	81.55	78.27	89.27	73.81	90.61	92.14
压缩比率	100	60.16	21.24	46.27	11.35	35.52	15.98
运行时间	2.409	6.202	3.5193	3.872	4.726	9.641	2.752

表 3 比较算法在数字手写体集上的运行效率

Table 3 Operational efficiency results obtained by compared algorithms on handwritten digits dataset

算法	KNN	CNN	GCNN	PSC	ISSARC	ILVQ	LCNN
分类精度	97.99	92.07	94.57	93.25	92.48	95.59	97.08
压缩比率	100	41.34	25.72	33.94	19.96	31.58	22.57
运行时间	756.39	214.34	595.28	456.92	612.58	372.35	247.47

表 4 比较算法分类精度与压缩比的实验数据

Table 4 Operational efficiency results obtained by compared algorithms on breast cancer data set

算法	KNN		CNN		GCNN		PSC		ISSARC		ILVQ		LCNN	
	分类精度	压缩比率	分类精度	压缩比率	分类精度	压缩比率	分类精度	压缩比率	分类精度	压缩比率	分类精度	压缩比率	分类精度	压缩比率
Iris	96.67	100	95.50	59.71	95.78	12.32	92.89	64.83	94.54	23.67	93.07	45.04	93.33	28.63
Wine	70.80	100	71.23	67.94	67.32	23.54	62.21	73.74	65.57	18.54	67.64	41.12	69.65	15.82
Glass	65.08	100	62.64	66.59	68.27	49.26	60.69	72.78	62.12	22.40	64.69	28.74	65.43	26.43
Ionosphere	88.68	100	85.89	48.33	84.32	22.17	86.18	45.19	87.45	8.76	89.29	19.91	86.16	18.79
Cancer	96.50	100	88.12	7.153	94.61	16.92	78.05	10.55	84.55	14.56	78.05	10.55	95.14	25.35
Zoo	83.22	100	88.14	57.67	88.73	31.52	78.26	57.19	76.43	23.51	87.10	35.19	92.62	34.36
Heart	76.21	100	67.27	43.40	75.49	46.24	79.54	31.23	68.31	10.60	80.34	37.38	76.57	37.01
TAE	77.72	100	56.34	36.75	64.48	44.61	72.25	37.18	57.57	36.92	70.22	23.33	76.69	21.69
Liver disorders	67.61	100	55.80	16.67	65.26	42.31	61.88	67.21	55.36	20.87	60.87	15.89	67.51	14.06
Spectf	71.95	100	63.05	20.36	73.35	52.07	79.41	24.56	72.33	16.48	77.93	35.74	80.14	35.09
Ecoli	86.45	100	77.44	53.71	76.79	33.85	74.93	42.89	78.69	15.70	82.22	42.96	80.17	29.73
Ctg	82.09	100	62.68	42.91	64.49	18.73	74.86	41.66	65.05	10.09	69.18	12.23	76.76	12.48
Average	82.87	100	72.84	43.43	79.21	32.80	75.11	47.42	72.33	18.51	76.72	29.01	80.01	24.95

基于表 4 数据, 可以得到如下结论: 对比 CNN 和 GCNN, LCNN 在保持明显的数据优势情况下, 有着较高比例数据集的分类精度优势; 对比 ILVQ 算法, 除 Ionosphere 外, LCNN 分类效率优势明显, 同时保证了 11 个数据集上的高压缩比; 对比 PSC 快速原型算法, LCNN 在保持 11 个数据集的高分类效率之外, 仍保持 9 个数据集的高压缩比率, 体现了较好的分类效率和较高的压缩比率. 相对于其他对比算法而言, ISSARC 算法保持着明显的平均压缩优势; 而 LCNN 算法仅有 2 个数据集的压缩率高于 ISSARC 算法. 通过表 5 运行时间数据, 可得出在小规模数据集下, 对于 KNN 和 CNN 而言, LCNN 时间优势不足, 而在较大规模数据集 Ctg 下, LCNN 时间优势明显; 此外, LCNN 相对于 GCNN、ISSARC、ILVQ 算法而言, 有着显著的运行时间优势; 而相对于目前快速原型 PSC 算法, LCNN 也有着明显的 12 取 7 和平均的两项时间优势.

表 5 比较算法的运行时间数据

Table 5 Running time results obtain by compared algorithms on UCI benchmark data sets

算法	KNN	CNN	GCNN	PSC	ISSARC	ILVQ	LCNN
Iris	0.48	0.57	0.76	0.74	2.47	0.67	0.61
Wine	0.64	1.30	0.94	1.28	3.23	2.11	0.58
Glass	0.94	1.12	1.42	3.10	2.43	2.50	2.19
Ionosphere	2.53	2.58	5.45	3.02	4.21	3.93	3.37
Cancer	11.47	3.06	3.29	22.25	3.74	7.64	5.83
Zoo	0.53	0.75	0.86	0.57	0.77	0.63	0.33
Heart	0.68	0.74	0.97	0.78	0.83	0.56	0.51
TAE	0.51	0.63	1.01	0.71	0.85	0.49	0.45
Liver disorders	2.35	1.37	2.77	2.43	5.46	3.49	3.34
Spectf	2.89	2.42	5.79	2.67	3.36	2.32	2.25
Ecoli	4.01	3.61	4.53	1.88	4.27	3.42	2.88
Ctg	90.92	42.9	74.56	41.82	62.11	37.64	37.33
Average	9.83	5.09	8.53	6.77	7.81	5.45	4.97

综合表 4 和表 5 结论, 可得出 LCNN 是一种快速的原型选择算法, 能在较好压缩比的情况, 保持较高的分类精度和较快的运行时间.

由于 PSC 和 ISSARC 属于非增量学习算法, 运行过程中需多次扫描数据集, 故较增量算法有过多的时空消耗, 故不是适用于大规模数据约简任务; GCNN 算法虽为增量算法, 但原型吸收等操作过程中存在依赖参数、运算量大的不足, 故受限于中小规模数据集处理.

综上, 针对表 6 所列数据集, 本文选用了 CNN、ILVQ 和 LCNN 三种算法进行比较, 其中 KNN 算法作用同上, 仅作为分类精度及运行时间的基准.

表 6 大规模数据集信息

Table 6 The information of large scale data sets

数据集	特征数	类别数	样本数
Magic	11	2	19 020
Shuttle	10	7	58 000
Skin	4	2	245 057

通过表 7 得出, KNN 未采取任何约简操作, 无法实现数据集约简, 所需运行时间代价最高. 而 LCNN 算法时间代价最小; 此外, CNN 算法的分类精度及压缩比率均差于其他两种算法, 而后两者在这两个评价指标上具有可比性. 故相比 ILVQ 高效增量算法而言, LCNN 算法同样具有处理大规模数据集的分类精度、压缩比率以及运行时间的综合优势.

综合上述各类对比实验的结论, LCNN 算法是一种快速的原型选择算法, 能获取到较好的原型集, 且在保持较好的分类精度和压缩比综合性能基础上, 提高了运行速度, 展现了其对大规模数据处理的较好能力.

表 7 大规模数据集实验数据

Table 7 The experimental results obtained by compared algorithms on large scale data sets

算法		Magic	Shuttle	Skin
KNN	分类精度	81.34	99.8	97.28
	压缩比率	100	100	100
	运行时间	1 275.6	19 024	212 837
CNN	分类精度	75.13	68.89	95.78
	压缩比率	45.69	31.18	33.16
	运行时间	913.72	7 659	94 267
ILVQ	分类精度	78.13	98.31	98.81
	压缩比率	19.67	13.26	18.41
	运行时间	482.6	2 356.8	10 319
LCNN	分类精度	79.84	99.59	97.95
	压缩比率	17.32	7.53	9.34
	运行时间	315.8	751.4	3 753.7

4 结论

LCNN 在 CNN 基础上引入了原型更新策略, 通过近邻局部均值和类全局信息设定不同的阀值, 指导原型集学习、动态更新, 从而实现原型集的动态增长. LCNN 算法的核心思想是在局部均值和类全

局信息的指导下, 调整原型学习策略降低读取序列对原型集选择的影响, 同时纳入原型集更新策略, 保留类边界原型等有效信息, 进而较好克服噪音等异常节点的影响. 实验数据显示, 该算法具有较好压缩比和分类精度的优势, 且可保障原型集的快速获取, 能较好地应用到大规模数据集分类中.

References

1 Wu X D, Kumar V, Quinlan J R, Ghosh J, Yang Q, Motoda H. Top 10 algorithms in data mining. *Knowledge and Information Systems*, 2008, **14**(1): 1−37

2 López J A O, Ochoa J A C, Trinidad J F M. Prototype selection methods. *Computacióny Sistemas*, 2010, **13**(4): 449−462

3 Verbiesta N, Cornelisa C, Herrerab F. FRPS: a fuzzy rough prototype selection method. *Pattern Recognition*, 2013, **46**(10): 2770−2782

4 Rico J R, Iñesta J M. New rank methods for reducing the size of the training set using the nearest neighbor rule. *Pattern Recognition Letters*, 2012, **33**(5): 654−660

5 Angiulli F. Fast nearest neighbor condensation for large data sets classification. *IEEE Transactions on Knowledge and Data Engineering*, 2007, **19**(11): 1450−1464

6 Chang F, Lin C C, Lu C J. Adaptive prototype learning algorithms: theoretical and experimental studies. *Journal of Machine Learning Research*, 2006, **7**: 2125−2148

7 García S, Derrac J, Cano J R. Prototype selection for nearest neighbor classification: taxonomy and empirical study. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2012, **34**(2): 417−435

8 Wu Y Q, Lanakiev K, Govindaraju V. Improved *k*-nearest neighbor classification. *Pattern Recognition*, 2002, **35**(10): 2311−2318

9 Olvera-López J A, Carrasco-Ochoa J A, Martínez-Trinidad J F. A new fast prototype selection method based on clustering. *Pattern Analysis and Applications*, 2010, **13**(2): 131−141

10 Mitani Y, Hamamoto Y. A local mean-based nonparametric classifier. *Pattern Recognition Letters*, 2006, **27**(10): 1151−1159

11 Brown T A, Koplowitz J. The weighted nearest neighbour rule for class dependent sample size. *IEEE Transaction on Information Theory*, 1979, **25**(5): 617−619

12 Han E H, Karypis G. Centroid-Based Document Classification: analysis & Experimental Results. Technical Report 00-017, Computer Science, University of Minnesota, 2000

13 Zeng Y, Yang Y P, Zhao L. Nonparametric classification based on local mean and class statistics. *Expert Systems with Applications*, 2009, **36**(4): 8443−8448

14 Brighton H, Mellish C. Advances in instance selection for instance-based learning algorithms. *Data Mining and Knowledge Discovery*, 2002, **6**(2): 153−172

15 Wang X Z, Wu B, He Y L. An iterative algorithm for sample selection based on the reachable and coverage. In: Proceedings of the IEEE International Conference on Communications Technology and Applications. Beijing, China: IEEE, 2009. 521−526

16 Theodoridis S, Koutroumbas K. *Pattern Recognition* (Third Edition). New York: Elsevier, chapter 5, 2006

17 Xu Y, Shen F R, Zhao J X. An incremental learning vector quantization algorithm for pattern classification. *Neural Computing and Applications*, 2012, **21**(6): 1205−1215

18 Newman A D. UCI machine learning repository [Online], available: http://www.ics.uci.edu/ mlearn, March 12, 2013

李 娟 西安电子科技大学计算机学院博士研究生, 陕西师范大学远程教育学院讲师. 主要研究方向为数据挖掘，模式识别. E-mail: ally_2004@126.com

(**LI Juan** Ph. D. candidate at the School of Computer Science and Technology, Xidian University, and also lecturer at the School of Distance Education, Shaanxi Normal University. Her research interest covers data mining and pattern recognition.)

王宇平 西安电子科技大学计算机学院教授. 主要研究方向为进化计算, 运筹学, 模式识别和机器学习. 本文通信作者. E-mail: ywang@xidian.edu.cn

(**WANG Yu-Ping** Professor at the School of Computer Science and Technology, Xidian University. His research interest covers evolutionary computation, operational research, pattern recognition, and machine learning. Corresponding author of this paper.)