基于聚类和类重叠分析的近邻分类①

刘杜钢

(福建师范大学 数学与计算机科学学院, 福州 350100)

摘 要: k 近邻分类(kNN)是一种简单而有效的非参数分类算法, 但存在着参数需要人工确定, 没有显式构建分类模型造成存储空间大、分类效率低, 且易受到“维灾”效应影响等缺点. 针对这些缺点, 提出一种高效的近邻分类新方法, 构造了两个新的近邻分类器. 新方法使用由 K 均值聚类产生的优化的簇原型集合为分类模型, 减少了存储空间的同时提高了分类效率; 提出三种类重叠分析策略并引入模糊基准度量以减轻维灾影响. 以该分类模型学习方法为基础, 提出一种新的kNN分类器和组合朴素贝叶斯的新分类器, 算法涉及的参数都可以自动确定. 在人工和现实数据集上进行的实验表明, 新分类器具有良好的分类效率和分类准确率.

关键词: 近邻分类; K 均值聚类; 簇原型; 类重叠分析; 模糊基准度量

Neighbor Classification Based on Clustering and Class Overlapping Analysis

LIU Du-Gang

(School of Mathematics and Computer Science, Fujian Normal University, Fuzhou 350100, China)

Abstract: K-nearest neighbor classifier (kNN) is a simple and effective non-parametric classification algorithm. The major drawbacks of the kNN include parameters to be determined manually, its low efficiency in testing phase and suffered effect of “curse of dimensionality”. An efficient method is proposed that constructing a new kNN classifier and Naive Bayes combination classifier. K-means clustering is used to build an optimal set of cluster prototype, reducing storage space while improving the classification efficiency and determining automatically parameters. Using three types of overlapping analysis strategies and fuzzy norms measure are to relieve impacts of “curse of dimensionality”. Experimental results on both synthetic and real-world data sets show that the new classifier has good classification efficiency and classification accuracy.

Key words: neighbor classification; K-means clustering; cluster prototype; class overlapping analysis; fuzzy norms measure

分类是一类有监督的学习任务, 其主要思想是在训练阶段构建分类模型, 在分类阶段根据分类模型预测新数据的类标号. 分类技术已广泛应用于欺诈检测、目标营销、性能预测和文本分类等方面[1], 是模式识别和数据挖掘等领域重要的组成部分.

当前, 分类主要通过决策树、贝叶斯分类器、支持向量机(SVM)、神经网络和基于实例学习等方法实现. k 近邻(kNN)分类[2]属于基于实例学习的方法, 由于其简单性和有效性, 是数据挖掘领域十大经典算法之一[3]. kNN 存在的缺点包括参数需要人工确定; 由于没有显式构建分类模型, 导致存储空间大和分类效率低下. 此外, kNN 易受到“维灾”效应影响, 具体表现在: 由于距离度量的 concentration effect, 随着数据维数的增长, 对于一个查询点, 以欧几里德距离等常用距离衡量的最远点和最近点之间的差异趋向于零, 导致“最近邻”失去意义; 某些点容易出现在其他点的最近邻列表中, 而某些点几乎从未出现, 即高维空间中的最近邻变得严重倾斜, 易导致 “虚假”最近邻的出现[4].

针对上述缺点, 研究人员提出了许多的解决方法. 对于参数的自动获取, 由 Guo 等人提出的 kNNModel

① 收稿时间:2014-12-30;收到修改稿时间:2015-03-02

算法[5], 使用贪婪型的搜索算法, 根据样本的分布情况自动确定构造模型簇的近邻数目, 但算法时间复杂度较高. 由 Chen 等人提出的 e-kNNModel 算法[6], 通过引入模糊聚类和试探性地簇分裂操作, 有效降低了 kNNModel 的时间复杂度, 但两者都未优化噪声或类重叠部分, 易在这些区域产生“碎片化”的模型簇, 导致模型簇数目剧增或部分样本未覆盖, 降低了分类精度; 对于减少存储空间和获取分类模型, 数据减少(DR)是一个成功的技术代表, 其中以原型选择(PS)[7]和原型产生(PG)[8]最为突出. 通过引入聚类提高原型产生效果的方法包括, 由 J. Arturo 等人提出的 PSC 算法[9]、由 Venmann CJ 等人提出的 NSB 算法[10]和由 Mollineda RA 等人提出的 GCM 算法[11]等. 它们都获得了较好的减少存储空间效果和分类效率, 但存在重要参数需要人工确定, 高度依赖聚类效果的问题; 对于减轻维灾影响, 由 Pasi Luukka 提出的模糊基准分类[12], 通过引入模糊集基准度量和理想类矢量, 在高维医疗数据集上获得较高的分类准确率, 但算法只考虑用单一理想类矢量表示一个类别的所有样本, 在任意数据集上不能始终保持高代表性.

在实践中为获得更好的分类效率和分类准确率, kNN 组合分类器的使用也很普遍, 但这些组合分类器都需忍受 kNN 的上述缺点. 例如在图像分类上取得良好效果的贝叶斯最近邻(NBNN)[13], 需要大量开销存储所有的描述符、进行最近邻检索和比较. 由 Manuel Montes 等人提出的 NBNNPG 算法[14], 通过引入原型产生处理数据集, 在新原型集上实施 NBNN 算法, 一定程度上减少了存储空间和计算开销, 但算法本身并未提出一个通用的原型产生技术, 只是评估多种已有的原型产生技术, 具有较大局限性.

本文提出一种基于 K 均值聚类和类重叠分析的近邻分类方法(Base –Kmeans-and-Overlap-Analyze NN, KaOANN), 构造了两个新的近邻分类器—KaOANN 原始分类器和组合朴素贝叶斯分类器. 与 kNN 相比, 该算法利用簇原型显式构造分类模型, 自动确定参数, 且减轻维灾影响; 与 e-kNNModel 等相比, 该算法可以从复杂类别结构的数据中获得优化模型簇; 和 PSC 等相比, 该算法自动确定参数, 依赖聚类效果程度降低; 和模糊基准分类等相比, 该算法使用获得的簇原型作为理想矢量, 在任意数据集上始终保持高代表性; 和 NBSSPG 等相比, 该算法给出了一种通用的原型产生技术. 在多个数据集上的实验结果表明, 新分类器具有较高的分类准确率和分类效率, 且有效减轻维灾影响.

文章结构如下: 在第 1 节, 简要说明背景知识和相关概念; 在第 2 节, 结合图形化示例描述提出的 KaOANN 方法和说明 KaOANN 组合贝叶斯分类器; 在第 3 节, 给出并说明提出的方法和多种相关联的算法在一系列数据集上的实验对比结果; 最后, 我们在第 4 节中给出结论以及下一步的工作展望.

1 背景知识和相关概念

给定一个训练集 $Tr=\{(X_1,y_1),(X_2,y_2),...,(X_n,y_n)\}$, 其中第 i 个样本表示为 $X_i=(x_{i1},x_{i2},...,x_{id})$, 类标号 $y_i\in\{1,2,\ldots,m\}$, $m(m\geq 2)$是类别数. 分类问题是从 Tr 上学习一个映射函数 $y=f(X)$, 预测任何给定的测试样本 X_t 的类标号 y_t, f 通常也称为分类模型.

kNN 利用训练样本间的相似性指标进行分类, 相似性指标通常使用 L_2 范式, $D(X_i,X_j)=\sqrt{\sum_{z=1}^{d}(x_{iz}-x_{jz})^2}$. 对于给定的测试样本 X_t, kNN 在 Tr 中搜索 k 个与 X_t 最相似的样本组成最近邻集合 $kNN(X_t)$, 依据多数表决的原则对 y_t 赋值, 即

$$y_t=\arg\max_{l}\sum_{(x_i,y_i)\in kNN(X_t)}I(l=y_t)$$

其中, I 是类标号, $I(.)$是指示函数, 如果其参数为真, 则返回 1, 否则返回 0.

在提出的方法中, K 均值聚类用于获取簇原型和确定近邻个数. K 均值聚类首先选择 K 个样本为初始质心, 指派每个样本到最相似的质心形成 K 个簇, 相似性度量通常使用 L_2 范式. 簇的新质心是所有簇成员的均值, 重新指派样本直到质心几乎不发生变化. 通过 K 均值聚类, 可以给出以下两个定义并在算法过程中使用.

定义 1. 在 Tr 上应用 K 均值聚类, 每个样本获得一个聚类的簇标号. 此时, 训练集的表示可以从原始的 $Tr=\{(X_1,y_1),(X_2,y_2),...,(X_n,y_n)\}$增加簇标号, 变换为 $Tr=\{(X_1,y_1,C_1),(X_2,y_2,C_2),...,(X_n,y_n,C_n)\}$, 簇标号 $C_i\in\{1,2,\ldots,K\}$, $K(K>1)$是簇的数目.

定义 2. 对于相同簇标号 C 的样本, 如果类标签 y 也一致, 则称这个簇为纯簇, 即

$$PC(Pure-Cluster)=\{X_i,X_j\mid\forall(i,j)C_i=C_j\rightarrow y_i=y_j\}$$

同理, 如果类标签 y 不一致, 则称它为非纯簇, 即

$$DPC(disPure-Cluster)=\{X_i,X_j\mid\exists(i,j)C_i=C_j\rightarrow y_i\neq y_j\}$$

模糊基准度量在提出的方法中作为“二次距离”, 在 L 范式距离提供大致的近邻后, 对近邻进行核查, 弥补高维时 L 范式距离的脆弱性. 模糊基准度量利用由模糊集的模糊逻辑结构构造的一系列基准来计算两个对象间的模糊相似度, 本文中使用的模糊基准度量定义如下:

$$S(X_i,V)=\sum_{j=1}^{d}\max(0,(1+\lambda)(Sn(x'_{i,j},v_j)+Sn(x_{i,j},v'_j)-1-\lambda Sn(x'_{i,j},v_j)Sn(x_{i,j},v'_j)))$$

其中, $Sn(x,v)=\min(1,x+v+\lambda xv)$, $x'=1-x$, $v'=1-v$, $\lambda>-1$, $x,v\in[0,1]^d$, X 是样本, λ 是比例参数, V 是理想类矢量, 通常可取每个类别所有样本的均值, $S(x,y)$越大表示两者相似度越高. 有关这个度量的更多信息可以参阅[11].

在下一节中, 我们将结合图形化示例描述提出的 KaOANN 方法和说明 KaOANN 组合贝叶斯分类器.

2 基于聚类和类重叠分析的近邻分类方法

在本节中, 我们定义基于 K 均值聚类获取簇原型和辅助簇的策略, 结合图形化示例描述并分析 KaOANN 方法的具体过程, 并对 KaOANN 组合贝叶斯分离器进行介绍. 为了便于描述, 下文使用 L_2 范式 $D(X_i,X_j)$作为样本间的相似度度量.

定义 3. 如果一个簇 C_i 属于纯簇集 PC, 以所有簇成员和的均值作为原型的圆心, $v=\frac{\sum_{X\in C_i}X}{|C_i|}$, 其中 $|C_i|$ 表示簇 C_i 包含的样本数; 以簇成员到圆心的最远距离, $r=\arg\max_{X\in C_i}D(X,v)$ 为半径; 并以类标签 y 作为原型的类标签, C_i 作为原型的簇标号, 建立簇原型, 即

$$P(P\ rototype)=\{C_i\in PC\mid(v,r,y,C_i)\}$$

定义 4. 如果一个簇 C_i 属于非纯簇集 DPC, 首先找到簇中的多数类, 即属于该类标签的样本数目最多, $y_m=\arg\max(|y_i|)$, 其中 $|y_i|$ 表示属于类标签 y_i 的样本数. 此时, 认为该簇 C_i 的类标签便为 y_m. 对于其余类标签属于 y_i, $y_i\neq y_m$ 的样本, 把它们从 C_i 移除, 放到统一的辅助簇 C_{k+1}, 且在辅助簇以原簇标号 C_i 区分它们. 最后, 建立辅助簇 C_{k+1}, 即

$$C_{k+1}=\{X\in C_i y_i\neq y_m\mid(X,y_i,C_i)\}$$

其中, 纯簇原型作为训练阶段获得的分类模型, 而辅助簇作为训练阶段构建簇原型的辅助空间, 以及在重叠分析中使用.

2.1 训练阶段

2.1.1 确定 K 值和获取原型的低依赖性策略

为了在 Tr 上应用 K 均值聚类, 需要事先给定划分的簇数目 K, 且参数 K 关系到获取的簇原型集合的质量. 在搜寻过程中, 我们利用如下指标来确定最佳 K 值.

指标 1 对于每一个类标签 y_i, 属于它的簇原型数目至少为 1. 即 $|\{PP\mid y\in y_i\}|\geq 1$

这个指标的作用在于, 簇原型作为最后获得的分类模型, 必须避免某个类标签没有任何分类模型进行分类预测, 且由于 K 均值聚类的特性, 此时可以认为同一类别划分成较好的有代表性的子类别. 从直观上看, 获得簇原型越多, 覆盖率越高, 可以有效提高分类精度. 但在实验中我们发现, 指标成立的效果高度依赖于 K 均值. 例如大部分数据集难以产生大量的纯簇, 造成簇原型的覆盖率较低; 即使可以产生大量的纯簇, 也需要一个很大的 K 值. 为此, 需要一个灵活的策略, 既确定 K 均值的参数和获得簇原型, 又降低对 K 均值的依赖程度.

我们的策略是在 Tr 上应用 K 均值, 令 K=2 并往上递增, 在满足指标时输出 K 为最佳参数值. 依次对 K 个簇进行判断, 如果簇 C_i 为纯簇, 则根据定义 1 建立簇原型; 如果 C_i 为非纯簇, 则根据定义 2, 令簇 C_i 为多数类标签 y_m 一致的纯簇, 在辅助簇 C_{k+1} 中存放其余类别的样本, 此时可以获得 K 个簇原型和一个辅助簇 C_{k+1}. 现在依次重新放置辅助簇 C_{k+1} 中的样本到 K 个纯簇, 放置的条件依照定义 5.

定义5. 对于辅助簇 C_{k+1} 的每一个样本 X_j, 重新放置它到类标签与它一致, 且与它距离最小的簇原型 C_i 中, 将 X_j 的 K 均值簇标号修改为 C_i, 即

$$\{X_j\rightarrow C_i\mid y_j=y_i,D(X,v_i)=\min D(X,v)\}$$

最后获得调整后的 K 个簇原型, 且都为纯簇. 由于只需要 K 均值满足指标 1, 降低了对 K 均值的依赖性. 纯簇建立过程充分考虑了整个数据集, 获得的簇原型具有高代表性.

由于我们的簇原型定义, 不同类别的簇原型之间可能出现覆盖区域重叠的情况. 我们并未避免出现重叠情况, 是因为现实数据集上类重叠现象十分常见, 且重叠区域是影响分类准确率的关键因素. 在下一节中, 我们分别介绍三种策略用于类重叠分析, 正面去处理簇原型覆盖区域重叠的情况, 进一步提高分类准确率.

2.1.2 类重叠分析的三种策略

类重叠问题分为概念重叠和样本重叠两个层次[15].

概念重叠是指样本空间中不同类别出现的区域出现重叠的情况, 而样本重叠是指不同类别样本位置接近甚至重叠的情况, 两者分别从宏观和微观刻画重叠. 易看出要处理的簇原型覆盖区域重叠属于概念重叠, 且现实数据集通常存在一定程度的概念重叠, 而少有样本重叠, 所以下文所提的类重叠默认是概念重叠. 我们将分别说明用于类重叠分析的三种策略: 丢弃、合并检查和分离.

2.1.1.1 丢弃

在上一步获取的辅助簇中, 类标签属于 y_i 的样本被划分到类标签 y_m 一致的簇, 这些样本可能属于噪声或者类别边界重叠的区域, 易导致分类器出现错判. 在原始样本集移除它们, 使得剩余样本集具有较好的分类准确率, 且缩小关联的簇原型覆盖区域, 减轻类重叠现象.

2.1.1.2 合并检查

考虑整个原始样本集, 为获得较好的分类准确率, 需要核查重叠区域的每个样本. 利用模糊基准度量计算样本和簇原型的相似性, 比较最大相似性的簇原型的类标签和样本类标签是否相同. 如果相同则放回, 否则移除样本. 最后获得一个在重叠区域拥有较好分类准确率的样本集.

2.1.1.3 分离

不同于子空间等方法对于同一类别的所有样本赋予相同的权重系数, 易导致非重叠区域的高准确率样本受到影响; 也不同于传统分类器训练所有样本, 易造成决策边界的偏移. 划分整个原始样本集为重叠集和非重叠集两部分, 重叠集通过簇原型覆盖判断:

$$OA(Overlap-Aaeas)=\{X\in Tr\mid D(X,v_i)\le r_i, D(X,v_j)\le r_j\}$$

非重叠集的簇原型与原样本集的簇原型一致, 获取重叠集簇原型的简单策略为: 在重叠集中, 辅助簇处于不同类别的簇原型覆盖重叠区域的核心位置, 且周围关联着少量不同类别的样本. 因为同类别的簇原型吸引力不及其他类别的簇原型(即样本错误聚类到的簇), 在原样本集上它们易导致错判, 所以在重叠集中, 对辅助簇中同一个错误聚类到的簇 C_i, 类标签 y_i 相同的样本, 依照定义 1 建立重叠簇的簇原型. 注意, 这里簇原型只需保留圆心和类标签, 即:

$$OP(Overlap-P\ rototype)=\{(X,y_i,C_i)\mid(v,y)\}$$

排除单样本的情况, 避免原型计算的失效, 且如果聚类后簇只包含一个其他类别的样本, 该样本可以视为噪声排除. 同时, 原样本集的簇原型继承到重叠集, 即重叠集的簇原型为 $P+OP$.

2.2 分类阶段

在分离和合并检查策略下, 训练阶段获取了 Tr 上优化的簇原型 P. 簇原型 P 作为分类阶段的分类模型, 可以对一个新样本 X_t 预测其类标签为 y_t. 执行的分类策略为:

如果测试样本 X_t 未落入任何一个簇原型中, 计算测试样本 X_t 与每个簇原型 P 边界的距离, 以最小距离的簇原型的类标签赋予 X_t.

如果测试样本 X_t 与某个簇原型 P 圆心的距离小于簇原型的半径, 则称 X_t 落入簇原型 P, X_t 的类标签可能与簇原型 P 的类标签一致. 分别统计可能类标签的数目, 如果所有类标签是一致的, 以该类标签赋予 X_t.

如果可能类标签不一致(两类别以上), 计算测试样本 X_t 与这些簇原型 P 的模糊基准度量, 以最大相似性的簇原型的类标签赋予 X_t.

对于分离策略, 前两种情况同上. 在第三种情况下样本落入重叠区域, 计算测试样本 X_t 与所有簇原型 $P+OP$ 的模糊基准度量. 如果最大相似性的簇原型属于 OP, 以该簇原型关联的属于 P 的簇原型中, 有较大相似性的簇原型的类标签赋予 X_t, 否则直接以最大相似性的簇原型的类标签赋予 X_t.

由于只需对覆盖的簇原型圆心额外进行模糊基准度量的计算, 计算复杂度不会有太大提高.

2.3 图形化示例

以 UCI-banana 数据集为例, 通过图形化演示 KaOANN 的具体过程, 数据集的可视化如图 1 所示. 图 2 展示了该数据集最佳 K 均值聚类的结果, K 确定为 7, 三角形表示辅助簇的样本.

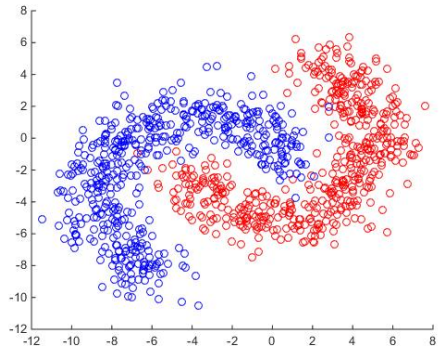

图 1 UCI-banana 数据集的可视化

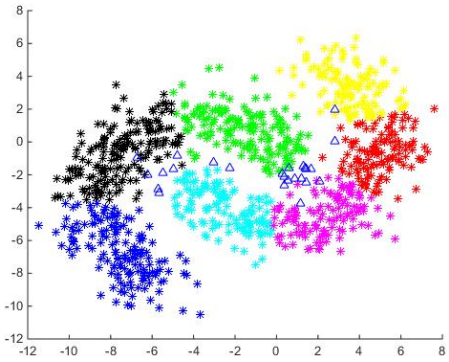

图 2 最佳 K 均值聚类结果

图 3 展示了通过低依赖性策略调整后的 K 均值聚类结果, 辅助簇样本依次放置到最近的同类标签簇中. 图 4 展示了原始方法获取的簇原型, 圆圈表示覆盖区域, 易发现不同类别的簇原型覆盖重叠的区域.

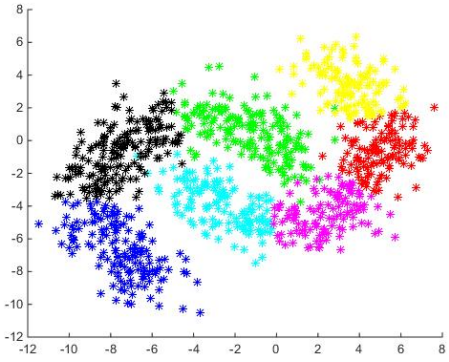

图 3 调整后的 K 均值聚类结果

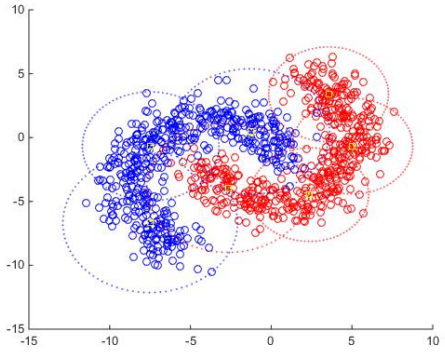

图 4 原始算法的簇原型

图 5 展示了结合丢弃策略的簇原型结果, 三角形表示丢弃的辅助簇样本, 易看出重叠区域的缩小. 图 6 展示了结合合并检查策略的簇原型结果, 绿色表示未通过检查而排除的样本.

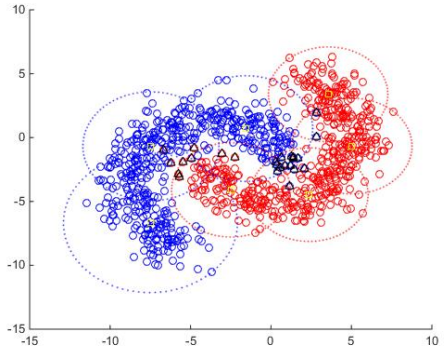

图 5 丢弃策略的簇原型

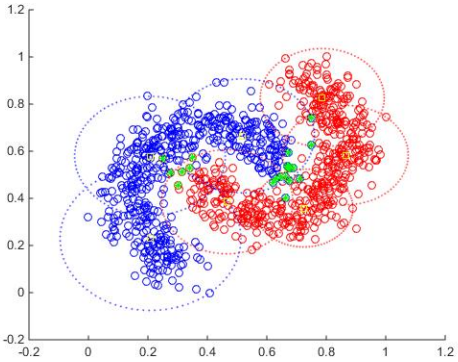

图 6 合并检测的簇原型

图 7 展示了结合分离策略的簇原型结果, 紫色表示重叠集, 其他样本为非重叠集. 绿色表示辅助簇样本, 黑色三角形表示重叠簇原型, 右上角的辅助簇单样本作为噪声排除.

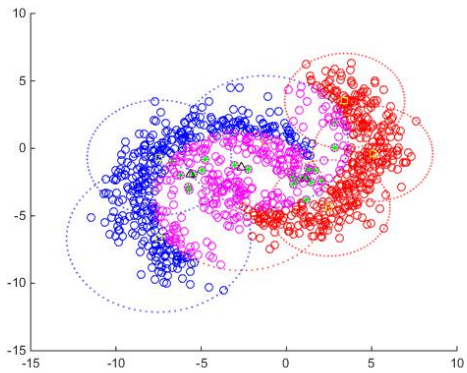

图 7 分离策略的簇原型

2.4 KaOANN 组合贝叶斯分类器

在 3.1.2.3 节中, 目标是分别构建非重叠集和重叠集的分类模型. 组合分类器是一个好的选择, 在非重叠集和重叠集采用不同的分类算法获取分类模型. 为

进一步提高分类准确率, 相对于 3.1.2.3 节的方案, 利用朴素贝叶斯算法构造重叠集的分类模型. 之所以选择朴素贝叶斯算法, 因为它往往提供更好的分类准确率.

考虑简单的情况, 按照 3.1.2.3 节的步骤, 通过簇原型覆盖判断划分出非重叠集和重叠集. 在非重叠集使用 KaOANN 方法构建分类模型, 簇原型为 P. 对所有重叠集样本使用朴素贝叶斯算法, 假定属性服从高斯分布. 对于每个样本 X, 朴素贝叶斯算法通过下式计算样本属于类别 C_i 的后验概率, 并将最大后验概率类别对应的类标签赋予样本.

$$P(x_k|C_i)=\frac{1}{\sqrt{2\pi}\sigma_{c_i}}e^{-\frac{(x_k-\mu_{c_i})^2}{2\sigma_{c_i}^2}}$$

$$P(X|C_i)=\prod_{k=1}^{d}P(x_k|C_i)=P(x_1|C_i)P(x_2|C_i)\ldots P(x_d|C_i)$$

其中, μ_{C_i} 和 σ_{C_i} 分别是 C_i 类别训练样本第 k 个属性的均值和标准差, 这便是朴素贝叶斯算法构建的重叠集的分类模型.

此时, 组合分类器需要存储重叠集所有样本和非重叠集的$|P|$个簇原型. 朴素贝叶斯算法根据样本的结构特征作出分类决策, 我们的划分方式没有改变原始的数据结构, 相反地, 划分出的重叠集包含一系列复杂区域让算法学习. 由于非重叠集和重叠集互不影响, 重叠集的贝叶斯模型和非重叠集的 KaOANN 模型都可以获得较高的分类准确率, KaOANN 组合贝叶斯分类器性能相比原始近邻贝叶斯分类器有较大提升.

在某些情况下, 存储重叠集所有样本需要很大开销, 可以对重叠集也使用原型生成得到一个保持较高贝叶斯分类准确率的重叠集子集. 首先利用贝叶斯模型判断重叠集每个样本的类别, 如果与真实类别一致则放回, 否则移除样本. 在新重叠集上找到样本数最多的类别, 设类标签为 y_m, 固定 y_m 的每个样本, 依次确定其余类别中离其最近的样本, 放置到 m 集合; 固定 m 集合的每个样本, 相反地确定离其最近的类标签 y_m 的样本, 放置到 m 集合, m 集合便是目标的重叠集. 由于 m 集合存放不同类别最相邻的样本, 刻画了类边界区域的特征, 使贝叶斯模型能学习到足够的类别区分信息, 且由于之前的核查步骤, 噪声不会对模型产生太大的影响.

3　实验结果与分析

在本节中, 主要选取有相关联的代表技术, 经典 kNN 算法, e-kNNModel 算法、PSC 算法和模糊基准分类, 在 8 个数据集上分别针对分类准确率和数据减少率两个方面进行实验对比. 其中, 经典 kNN 的 k=3, PSC 采用文献的最佳参数 $K=6c$, 其中 c 是 Tr 的类标签数目, 即 $|y|$, 模糊基准度量参数 $\lambda=0$.

3.1 实验数据

实验数据分别来源于两个人工数据集和六个 UCI 公众数据集[16], 详细的数据信息如表 1 所示. 其中, Dataset1 是协方差为 0.5 和 0.5, 均值为[3,0]和[6,0]的高斯正态分布人工数据集, 两个类别大小几乎相同, 重叠情况较少; Dataset2 是协方差为 0.5 和 0.05, 均值为[0,0]和[1,0]的高斯正态分布人工数据集, 两个类别大小有较大差异, 重叠情况显著. 它们用来评估算法对重叠情况的适应性, 两者都在 Matlab 上生成. 重叠率的计算公式如下,

$$\text{重叠率}=\frac{|OA.num|}{|Tr|}$$

其中, $|OA.num|$表示重叠区域样本数, $|Tr|$表示总样本数, 重叠区域根据 3.1.2.3 节定义计算.

表 1　实验数据的详细信息

数据集	样本数	属性数	类别数	重叠率
Iris	150	4	3	0.2
Heart	270	13	2	0.704
Wine	178	13	3	0.722
Diabetes	768	8	2	0.961
Aust	690	14	2	0.710
Banana	1000	2	2	0.263
Dataset1	400	2	2	0.275
Dataset2	1000	2	2	0.6

3.2 实验结果

实验使用 10 折交叉验证方法, 即数据集被划分为 10 份, 每次选取其中的 9 份作为训练集, 剩余的 1 份作为测试集. 实验选择准确率作为评价指标, 准确率的计算公式如下:

$$\text{准确率}=\frac{|Ts_r|}{|Ts|}$$

其中$|Ts|$是测试集样本数, $|Ts_r|$是正确分类的样本数, 即分类算法赋予的类标签 $y=y_t$, y_t 是测试集的真实类标签. 我们统计 100 次运行的结果, 由于 e-kNNModel 和 PSC 算法对聚类算法的依赖性, 可能出现实验失败的情况, 我们统计它们 100 次成功实验的结果.

分类准确率实验使用提出的方法和经典 kNN 算法[2], e-kNNModel 算法[7]、PSC 算法[11]和模糊基准分类[14]. 我们对比算法改进 kNN 的效果, 获取的簇原型好坏程度和对重叠现象的适应性. KaOANN1、KaOANN2 和 KaOANN3 分别表示丢弃策略、合并检查策略和分离策略.

详细的准确率实验结果如表 2 和表 3 所示. 表 2 展示了不同实验数据集上各个算法获得的最高准确率, 表 3 展示了各个算法的平均准确率.

其次, 分别统计 3 种算法的平均数据减少率(kNN 和模糊基准分类除外), 数据减少率的计算公式为:

$$\text{数据减少率} = \frac{|P_{tol}|}{|Tr|},$$

其中, $|Tr|$是训练集的样本数, $|P_{tol}|$是最后获得的原型数. 数据减少率越高, 表示需要存储的数据越少, 分类效率越高. 详细的数据减少率如表 4 所示, KoOANN 左侧表示丢弃和合并检查策略, 右侧表示分离策略.

从表 2 和表 3 可以看出, 所有算法对 kNN 都有较好的改进, 当重叠率越大时, 改进效果越好. KaOANN 在大部分数据集上持平甚至超过其他算法, 说明 KaOANN 能获得更好的簇原型. 随着重叠率的增加, 只有模糊基准和 KaOANN 具有较好的适应性. 从表 4 可以看出 KaOANN 相比其他算法拥有较好的数据减少率.

表 2 实验结果-最高准确率

数据集	kNN	e-kNNModel	PSC	模糊基准	KaOANN1	KaOANN2	KaOANN3
Iris	0.960	0.960	0.970	0.960	0.971	0.973	0.973
Heart	0.807	0.814	0.789	0.874	0.815	0.826	0.826
Wine	0.967	0.972	0.960	0.921	0.972	0.978	0.977
Diabetes	0.746	0.760	0.741	0.794	0.750	0.746	0.742
Aust	0.840	0.840	0.826	0.876	0.855	0.858	0.850
Banana	0.987	0.987	0.973	0.824	0.980	0.982	0.985
Dataset1	0.988	0.980	0.978	0.970	0.985	0.983	0.980
Dataset2	0.978	0.972	0.956	0.972	0.966	0.965	0.972

表 3 实验结果-平均准确率

数据集	kNN	e-kNNModel	PSC	模糊基准	KaOANN1	KaOANN2	KaOANN3
Iris	0.953	0.952	0.952	0.947	0.957	0.954	0.955
Heart	0.790	0.778	0.770	0.844	0.806	0.807	0.804
Wine	0.963	0.960	0.947	0.886	0.951	0.965	0.963
Diabetes	0.738	0.745	0.710	0.773	0.739	0.737	0.723
Aust	0.837	0.826	0.815	0.856	0.847	0.848	0.840
Banana	0.986	0.985	0.963	0.790	0.975	0.976	0.981
Dataset1	0.982	0.974	0.968	0.940	0.975	0.976	0.977
Dataset2	0.974	0.968	0.945	0.960	0.961	0.962	0.970

表 4 数据减少率

数据集	e-kNNModel	PSC	KoOANN
Iris	0.913	0.823	0.947/0.933
Heart	0.903	0.450	0.956/0.919
Wine	0.899	0.819	0.961/0.949
Diabetes	0.907	0.728	0.969/0.944
Aust	0.970	0.950	0.977/0.961
Banana	0.967	0.955	0.992/0.988
Dataset1	0.975	0.932	0.988/0.983
Dataset2	0.977	0.941	0.993/0.990

结合分类准确率和数据减少率结果, 我们发现三种重叠分析的策略中, 分离策略得到更稳定和更好的分类效果. 因为样本集划分为不相交的重叠集和非重叠集, 彼此之间不会互相影响, 非重叠集分类模型的分类精度较高, 算法总分类精度很大程度上取决于重叠集的分类模型. 为了说明这一点, 也为了进一步说明分离策略的优越性, 我们使用3.4节说明的KaOANN组合贝叶斯分类器简单情况, 在相同实验数据集上测试, 平均准确率的实验结果如表 5 所示. 此时, 在所有数据集上都获得较好的分类效果, 包括原来简单重叠簇原型下分类准确率不高的 Diabetes 数据集.

表 5 KoOANN 贝叶斯分类器的平均准确率

数据集	KaOANN + Naive Bayes
Iris	0.960
Heart	0.819
Wine	0.966
Diabetes	0.756
Aust	0.823
Banana	0.977
Dataset1	0.983
Dataset2	0.973

4 结论

本文提出了一种高效的基于 K 均值聚类和类重叠分析的近邻分类算法(KoOANN), 构造了两个新的近邻分类器—KaOANN 原始分类器和组合朴素贝叶斯分类器. 该算法自动确定近邻参数, 显式构造分类模型提高分类效率和减少存储空间, 且有效减轻维灾影响. 此外, 可以有效构造两个新的近邻分类器. 算法通过 K 均值聚类获取的簇原型来显式构造分类模型, 结合三种类重叠分析的策略对簇原型进行优化, 引入模糊范式度量去适应“维灾效应”, 且参数可以自动确定. 在人工和现实数据集上的实验结果表明, 该算法和分类器具有良好的性能. 我们下一步的工作重点将集中在如何制定一个更好的 K 均值聚类效果指标, 以及进一步探索重叠簇原型的优化.

参考文献

1 Han JW, Pei J, Micheline K. Data Mining: Concepts and Techniques, 3rd Ed. 北京:机械工业出版社, 2013.

2 Cover T, Hart P. Nearest neighbor pattern classification. IEEE Trans. on Information Theory, 1967, 13(1): 21–27.

3 Yang Q, Wu X. 10 Challenging problems in data mining research. International Journal of Information Technology & Decision Making, 2006, 5(4): 597–604.

4 Aggarwal CC, Hinneburg A, Keim DA. On the surprising behavior of distance metrics in high dimensional space. Berlin. Springer Berlin Heidelberg. 2001.420–434.

5 Gou G, Wang H, Bell D, Bi Y, Greer K. KNN model-based approach in classification. In: Robert M, ed. On The Move to Meaningful Internet Systems 2003: CoopIS, DOA, and ODBASE. Berlin. Springer Berlin Heidelberg. 2003. 986–996.

6 Chen L, Gou G, Wang S. Nearest neighbor classification by partially fuzzy clustering. Advanced Information Networking and Applications Workshops (WAINA), 2012 26th International Conference on. New York. IEEE. 2012. 789–794.

7 Garcia S, Derrac J, Cano JR, Herrera F. Prototype selection for nearest neighbor classification: Taxonomy and empirical study. IEEE Trans. on Pattern Analysis and Machine Intelligence, 2012, 34(3): 417–435.

8 Triguero I, Derrac J, Garcia S, Herrera F. A taxonomy and experimental study on prototype generation for nearest neighbor classification. IEEE Trans. on Systems, Man, and Cybernetics, Part C: Applications and Reviews, 2012, 42(1): 86–100.

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

10 Venmann CJ, Reinders MJT. The nearest sub-class classifier: a compromise between the nearest mean and nearest neighbor classifier. IEEE Trans. on Pattern Anal Match Intelligence, 2005, 27(9): 1417–1429.

11 Mollineda RA, Ferri FJ, Vidal E. An efficient prototype merging strategy for the condensed 1-NN rule through class-conditional hierarchical clustering. Pattern Recognition, 2002, 35(12): 2771–2782.

12 Luukka P. Similarity classifier using similarity measure derived from Yu’s norms in classification of medical data sets. Computers in Biology and Medicine, 2007, 37(8): 1133–1140.

13 Boiman O, Shechtman E, Irani M. In defense of nearest-neighbor based image classification. IEEE Conference on Computer Vision and Pattern Recognition, 2008. CVPR 2008. New York. IEEE. 2008. 1–8.

14 Escalante HJ, Sotomayor M, Montes M, Lopez-Monroy AP. Object recognition with näive bayes-nn via prototype generation. In: Ching YS, ed. Pattern Recognition. Berlin: Springer International Publishing, 2014: 162–171.

15 熊海涛,吴俊杰,刘洪甫,刘鲁.分类中的类重叠问题及其处理方法研究.管理科学学报, 2013, 16(4):8–21.

16 https://archive.ics.uci.edu/ml/datasets.html.