基于最佳距离度量的两层最近邻分类算法

崔宾阁,庄仲杰
(山东科技大学 信息科学与工程学院,山东 青岛 266590)

摘 要:两层最近邻(TLNN)分类算法通过在有限训练样本条件下最小化错误率的平均绝对误差,能够产生比 k-最近邻(kNN)算法更好的分类结果,但其精度易受噪声影响。针对这个问题,提出一种基于最佳距离度量的两层最近邻分类算法(ODM-TLNN),提高对噪声数据的鲁棒性。算法分为两层,下层使用最佳距离度量来确定一个未标记样本的局部子空间,上层采用 AdaBoost 在子空间进行信息提取。基于 UCI 数据集的实验结果表明,该算法能充分降低分类错误率,并且在噪声数据下具有较好的稳定性。

关键词:距离度量;k-最近邻算法;噪声数据;分类;错误率

中图分类号:TP181 **文献标志码**:A **文章编号**:1672-3767(2015)03-0020-08

Two-level Nearest Neighbor Classification Algorithm Based on Optimal Distance Metric

Cui Binge, Zhuang Zhongjie
(College of Information Science and Engineering, Shandong University of Science and Technology, Qingdao, Shandong 266590, China)

Abstract: Two-level nearest neighbor (TLNN) classification algorithm can obtain better results than k-nearest neighbor (kNN) classification algorithm by minimizing the mean-absolute error of the misclassification rate with finite number of training samples. However, the classification accuracy of TLNN is susceptible to noise. In order to solve this problem, this paper proposed a two-level nearest neighbor classification algorithm based on optimal distance metric (ODM-TLNN) which can improve the robustness on noise. This proposed algorithm is divided into two levels. At the low-level, the optimal distance metric to determine a local subspace of an unlabeled test sample; at the high-level, AdaBoost is used as guidance for local information extraction. The experiment results based on UCI data sets show that the proposed algorithm is able to achieve a lower classification error rate and it has more stable performance in noisy cases.

Key words: distance metric; k-nearest neighbor classification; noisy case; classification; error rate

k-最近邻(k-nearest neighbor, kNN)算法是一种经典的非线性分类算法,常用于确定决策边界。给定一个未知数据,kNN 搜索模式空间,找出最接近未知数据的 k 个训练数据。研究表明,kNN 分类算法的渐进错误率,不论使用哪一种距离度量,最多是贝叶斯分类错误率的两倍。当训练集的样本数量为无限多个时,在无穷小的未标记测试样本域中类条件概率是常量。因为在输入空间中欧氏距离具有各向同性的性质,因此,最近邻样本使用欧式距离度量作为距离度量方式是自然的选择[1]。然而,无限数量的训练样本在现实情况下是不存在的。因此,局部的类条件概率在有限样本下假设为常量是不成立的,这样,在训练数据中使用欧氏距离不能充分利用统计规律。为了提高 kNN 分类的性能,选择一个合适的距离度量至关重要。

研究表明,可以选择合适的距离度量来提升 kNN 算法的分类精度[2]。Domeniconi 等[3]提出利用支持向量机作为指导来定义一个局部灵活度量(local flexible metric-support vector machine, LFM-SVM)的方

收稿日期:2014-12-29
基金项目:国家自然科学基金项目(41406200);山东省自然科学基金项目(ZR2014DQ030)
作者简介:崔宾阁(1979—),男,山东烟台人,副教授,博士后,主要从事智能信息处理方面的研究. E-mail: cuibinge@qq.com

法。然而,将转换数据用核函数的方式从输入空间转换到高维特征空间时,难以维持数据的不变性。卢伟胜[4]和 Gou[5]利用加权方式进行距离学习,取得了较好的效果。林耀进等[6]考虑样本邻域对分类的影响,提出一种融合邻域信息的 kNN 分类算法。此外,Yang 等[7]提出从训练数据集中通过知识嵌入的方式学习距离度量的算法。由于局部特征的相似性加上位置接近,可以使用新的距离度量来确定未标记测试样本的近邻。张兴福[8]提出局部线性嵌入算法,该算法从样本中学习到一个马氏度量,然后在局部线性嵌入算法的近邻选择、现有样本及新样本降维过程中用马氏度量作为相似性度量。Gao 等[9]提出一种基于自适应距离度量的两层最近邻分类算法(two-level nearest neighbor,TLNN),以最小化平均绝对误差为原则定义一个自适应距离度量,该算法分两层,下层使用欧氏距离确定一个未标记样本的局部子空间,上层采用 AdaBoost 算法在子空间进行信息提取。然而,该算法在上层子空间进行信息提取时定义的距离度量最终演化为类标签的比较,且对在下层建立子空间方式重视不足,这对算法的分类精度以及抗噪性都有很大影响。为此,本研究提出基于最佳距离度量的两层最近邻(optimal distance metric-two level nearest neighbor,ODM-TLNN)分类算法,该方法在均方意义下可将有限样本的错误率降到与无限样本的错误率相近。

1 TLNN 算法

1.1 最小绝对误差距离度量

设 $\boldsymbol{x}$ 是一个未知样本点,$\boldsymbol{x}'$ 是 $\boldsymbol{x}$ 的最近邻点。由于多分类问题可以转换为多级二分类问题,因此,为简化起见,只考虑二分类的情况。对于最近邻算法,未知样本 $\boldsymbol{x}$ 被错分的概率是[10]:

$$\begin{aligned}P_N(e\mid\boldsymbol{x})=P(w_1\mid\boldsymbol{x})P(w_2\mid\boldsymbol{x}')+P(w_2\mid\boldsymbol{x})P(w_1\mid\boldsymbol{x}')=\\P(w_1\mid\boldsymbol{x})[1-P(w_1\mid\boldsymbol{x}')]+P(w_2\mid\boldsymbol{x})P(w_1\mid\boldsymbol{x}')=\\2P(w_1\mid\boldsymbol{x})P(w_2\mid\boldsymbol{x})+[P(w_1\mid\boldsymbol{x})-P(w_2\mid\boldsymbol{x}')]\cdot[P(w_1\mid\boldsymbol{x})-P(w_1\mid\boldsymbol{x}')]。\end{aligned}\tag{1}$$

其中:N— 训练样本数量;w_1,w_2— 类标签。当训练样本的数量 $N\to\infty$ 时,渐进条件错误率为:

$$P(e\mid\boldsymbol{x})=\lim_{N\to\infty}P_N(e\mid\boldsymbol{x})=2P(w_1\mid\boldsymbol{x})P(w_2\mid\boldsymbol{x})。\tag{2}$$

这样,式(1)和式(2)都表示最近邻法的条件错误率,式(1)是在有限样本情况下,式(2)是在无限样本情况下的计算结果,计算两者之间的误差,可得

$$P_N(e\mid\boldsymbol{x})-P(e\mid\boldsymbol{x})=[P(w_1\mid\boldsymbol{x})-P(w_2\mid\boldsymbol{x})]\cdot[P(w_1\mid\boldsymbol{x})-P(w_1\mid\boldsymbol{x}')],\tag{3}$$

则在有限和无限数量训练样本下的条件错误率的平均绝对误差为 $E\{[\,|P_N(e\mid\boldsymbol{x})-P(e\mid\boldsymbol{x})|\,]\mid\boldsymbol{x}\}$。

定义 $\varepsilon(\boldsymbol{x},\boldsymbol{x}')=|P(w_1\mid\boldsymbol{x})-P(w_1\mid\boldsymbol{x}')|$,则 $\varepsilon(\boldsymbol{x},\boldsymbol{x}')$ 和 $P(w_1\mid x)-P(w_1\mid\boldsymbol{x}')$ 之间呈线性对应关系,如图 1(a)所示。由于$[P(w_1\mid\boldsymbol{x})-P(w_2\mid\boldsymbol{x})]$在式(3)中是一个常数项,那么最小化 $|P_N(e\mid\boldsymbol{x})-P(e\mid\boldsymbol{x})|$ 相当于最小化 $\varepsilon(\boldsymbol{x},\boldsymbol{x}')$。通常,可以通过增加训练样本数量来最小化 $\varepsilon(\boldsymbol{x},\boldsymbol{x}')$。文献[9]中试图不增加样本数量而定义一个等价距离度量:

$$\begin{aligned}\varepsilon'(\boldsymbol{x},\boldsymbol{x}')=\left|\ln\frac{P(w_1\mid\boldsymbol{x})}{P(w_2\mid\boldsymbol{x})}-\ln\frac{P(w_1\mid\boldsymbol{x}')}{P(w_2\mid\boldsymbol{x}')}\right|=\left|\ln\frac{P(w_1\mid\boldsymbol{x})}{P(w_1\mid\boldsymbol{x}')}\cdot\frac{P(w_2\mid\boldsymbol{x}')}{P(w_2\mid\boldsymbol{x})}\right|=\\\left|\ln\frac{P(w_1\mid\boldsymbol{x})}{P(w_1\mid\boldsymbol{x}')}\cdot\frac{1-P(w_1\mid\boldsymbol{x}')}{1-P(w_1\mid\boldsymbol{x})}\right|,\end{aligned}\tag{4}$$

此处 $P(w_2\mid\boldsymbol{x})=1-P(w_1\mid\boldsymbol{x})$,$P(w_2\mid\boldsymbol{x}')=1-P(w_1\mid\boldsymbol{x}')$,则 $\varepsilon'(\boldsymbol{x},\boldsymbol{x}')$ 与 $\dfrac{P(w_1\mid\boldsymbol{x})}{P(w_1\mid\boldsymbol{x}')}\cdot\dfrac{1-P(w_1\mid\boldsymbol{x}')}{1-P(w_1\mid\boldsymbol{x})}$ 之间的关系如图 1(b)所示。

当 $P(w_1\mid\boldsymbol{x})=P(w_1\mid\boldsymbol{x}')$ 时,$\varepsilon(\boldsymbol{x},\boldsymbol{x}')$ 与 $\varepsilon'(\boldsymbol{x},\boldsymbol{x}')$ 同时取到最小值 0;$P(w_1\mid\boldsymbol{x})\leqslant P(w_1\mid\boldsymbol{x}')$ 时,$\varepsilon(\boldsymbol{x},\boldsymbol{x}')$ 与 $\varepsilon'(\boldsymbol{x},\boldsymbol{x}')$ 同时下降;$P(w_1\mid\boldsymbol{x})>P(w_1\mid\boldsymbol{x}')$ 时,$\varepsilon(\boldsymbol{x},\boldsymbol{x}')$ 与 $\varepsilon'(\boldsymbol{x},\boldsymbol{x}')$ 同时上升,$\varepsilon(\boldsymbol{x},\boldsymbol{x}')$ 与 $\varepsilon'(\boldsymbol{x},\boldsymbol{x}')$ 有相同的变化趋势。因此,在判定最近邻点时,可以用 $\varepsilon'(\boldsymbol{x},\boldsymbol{x}')$ 代替 $\varepsilon(\boldsymbol{x},\boldsymbol{x}')$。TLNN 算法利用统计规律来最小化 $\varepsilon'(\boldsymbol{x},\boldsymbol{x}')$,利用 AdaBoost 算法构建分类器 $f(\boldsymbol{x})$,其经验风险函数为:$\dfrac{1}{N}\sum_{i=1}^{N}\exp(-y_if(\boldsymbol{x}_i))$,这里 y_i 表示输入样本点 $\boldsymbol{x}_i$ 的类标签。令 $\boldsymbol{P}(y\mid\boldsymbol{x})$ 表示给定输入值 $\boldsymbol{x}$ 的类标签的条件概率。假设样本是独立同分布的,经验风险的损失

函数期望在样本数量趋于无穷大时($N\to\infty$),满足大数定理

$$\lim_{N\to\infty}\frac{1}{N}\sum_{i=1}^{N}\exp(-y_i f(\boldsymbol{x}_i))=\int_{\boldsymbol{x}}\sum_{y=\pm1}\boldsymbol{P}(y\mid\boldsymbol{x})\exp(-yf(\boldsymbol{x}))\mathrm{d}\boldsymbol{x}=E[\exp(-yf(\boldsymbol{x}))]。\quad(5)$$

分类器 $f(\boldsymbol{x})$ 可以通过最小化风险函数 $E[\exp(-yf(\boldsymbol{x}))]$ 来推导,式(5)中的被积函数在 $f(\boldsymbol{x})$ 处取得最小值,满足

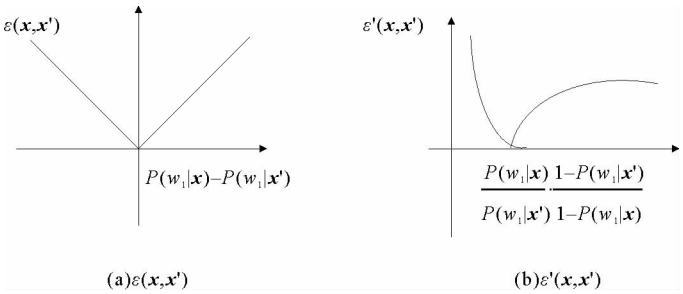

图 1 $\varepsilon(\boldsymbol{x},\boldsymbol{x}')$和$\varepsilon'(\boldsymbol{x},\boldsymbol{x}')$

Fig. 1 $\varepsilon(\boldsymbol{x},\boldsymbol{x}')$ and $\varepsilon'(\boldsymbol{x},\boldsymbol{x}')$

$$\frac{\partial\sum_{y=\pm1}P(y\mid\boldsymbol{x})\exp(-yf(\boldsymbol{x}))}{\partial f(\boldsymbol{x})}=0。$$

不失一般性,若 $\boldsymbol{x}_i\in w_1$,$y_i=1$,否则 $y_i=-1$。由上式可得:

$$\frac{P(1\mid\boldsymbol{x})}{P(-1\mid\boldsymbol{x})}=\frac{\exp(f(\boldsymbol{x}))}{\exp(-f(\boldsymbol{x}))}=\exp(2f(\boldsymbol{x}))\Rightarrow f(\boldsymbol{x})=\frac{1}{2}\ln\frac{P(1\mid\boldsymbol{x})}{P(-1\mid\boldsymbol{x})},$$

那么

$$\varepsilon'(\boldsymbol{x},\boldsymbol{x}')=\left|\ln\frac{P(w_1\mid\boldsymbol{x})}{P(w_2\mid\boldsymbol{x})}-\ln\frac{P(w_1\mid\boldsymbol{x}')}{P(w_2\mid\boldsymbol{x}')}\right|=2\mid f(\boldsymbol{x})-f(\boldsymbol{x}')\mid。$$

继而定义距离度量:

$$D_{\mathrm{TLNN}}(\boldsymbol{x},\boldsymbol{x}')=\frac{1}{2}\varepsilon'(\boldsymbol{x},\boldsymbol{x}')=\mid f(\boldsymbol{x})-f(\boldsymbol{x}')\mid。\quad(6)$$

1.2 TLNN 算法处理流程

AdaBoost 算法[11] 是一种经典分类方法,应用广泛,在实际应用中取得了较好的效果,但在有噪声的情况下,容易出现过拟合现象。kNN 算法不易受噪声数据的影响,具有很好的稳定性。

为了减少 AdaBoost 算法过拟合现象的发生,TLNN 算法考虑用 kNN 算法进行初步筛选,建立一个含有较少噪声数据的局部子空间。TLNN 算法的处理流程如下:首先,利用 AdaBoost 算法建立强分类器 $f(\boldsymbol{x})$;然后,在下层用 kNN 算法为每个测试数据选定 k_1 个近邻,建立局部子空间,在上层用距离度量 $D_{\mathrm{TLNN}}(\boldsymbol{x},\boldsymbol{x}')$ 从局部子空间选定 k_2 个近邻;最后,对给定的观测值 $\boldsymbol{x}$ 判定其标签。

1.3 TLNN 算法不足之处

TLNN 算法存在以下不足:首先,kNN 假设类条件概率密度在局部近邻区域相同,当这一条件不满足时,kNN 错分的风险较大,而 AdaBoost 算法本身对样本的密度分布是不敏感的;其次,TLNN 算法利用 kNN 算法建立局部子空间限制 AdaBoost 算法过度延伸,所用的样本集较小,而 kNN 在样本趋于无穷多时具有较好的分类能力,样本较少时 kNN 错分的风险很大,这是算法的主要不足。

以图 2 为例,图中黑点为测试数据,白点为近邻的正负样本。5 个白点样本组成测试数据下层局部子空间,假设当测试样本用 AdaBoost 算法得出黑点标签为 -1 时,由于下层局部子空间中有负样本,当 $k=1$ 时用距离度量 $D_{\mathrm{TLNN}}(\boldsymbol{x},\boldsymbol{x}')$ 判定黑点的最终标签为 -1。然而,由图 2 可见,组成局部子空间的 5 个白点样本中,负样本可能为异常样本,黑点的真实标签为 $+1$ 的概率更大。因此,在样本有限情况下用最近邻算法建立下层子空间时,合适的距离度量选择对算法结果有很大影响。

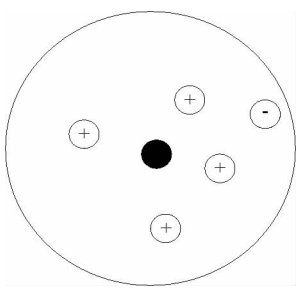

图 2 TLNN 算法错分的情况

Fig. 2 Error classification of TLNN algorithm

2 基于最佳距离度量的两层最近邻分类算法

针对TLNN算法的不足，考虑到样本集是有限的，需要研究如何在样本 $\boldsymbol{x}$ 的局部近邻区域中选择 $\boldsymbol{x}$ 的最近邻 $\boldsymbol{x}'$，在均方意义下将有限样本的错误率降到最小。文献[10]提出的最佳距离度量可以用来解决这个问题。最佳距离度量给出了一个新的距离度量函数，可以用来建立分类器的局部子空间，最大限度地降低噪声数据的影响。最佳距离度量推导过程如下：

在 $\boldsymbol{x}$ 的一个局部近邻区域，$P(w_1 \mid \boldsymbol{x}')$ 可通过截尾的级数展开成线性近似为

$$P(w_1 \mid \boldsymbol{x}') \doteq P(w_1 \mid \boldsymbol{x}) + \nabla P(w_1 \mid \boldsymbol{x})^{\mathrm{T}}(\boldsymbol{x}' - \boldsymbol{x})。 \tag{7}$$

其中，$\nabla P(w_1 \mid \boldsymbol{x})^{\mathrm{T}}$ 表示 $P(w_1 \mid \boldsymbol{x})$ 的梯度转置。

根据 $\varepsilon(\boldsymbol{x},\boldsymbol{x}')$ 的定义和式(7)，有

$$\varepsilon(\boldsymbol{x},\boldsymbol{x}') \doteq \mid \nabla P(w_1 \mid \boldsymbol{x})^{\mathrm{T}}(\boldsymbol{x} - \boldsymbol{x}') \mid, \tag{8}$$

则 $$E[\varepsilon(\boldsymbol{x},\boldsymbol{x}') \mid \boldsymbol{x}] = E\{[\mid \nabla P(w_1 \mid \boldsymbol{x})^{\mathrm{T}}(\boldsymbol{x} - \boldsymbol{x}') \mid] \mid \boldsymbol{x}\}。$$

可以得到：在 $\boldsymbol{x}$ 的局部近邻区域中，利用 $\varepsilon(\boldsymbol{x},\boldsymbol{x}')$ 的近似表达式(8)，最近邻 $\boldsymbol{x}'$ 应使 $\mid \nabla P(w_1 \mid \boldsymbol{x})^{\mathrm{T}}(\boldsymbol{x} - \boldsymbol{x}') \mid$ 为最小。即新定义的距离度量为

$$D(\boldsymbol{x},\boldsymbol{x}^l) = \mid \nabla P(w_1 \mid \boldsymbol{x})^{\mathrm{T}}(\boldsymbol{x} - \boldsymbol{x}^l) \mid。 \tag{9}$$

其中，$\boldsymbol{x}^l$ 为 $\boldsymbol{x}$ 的局部近邻区域中的样本，则在均方意义下将有限样本的错误率降到与无限样本下的错误率相近。但是，只能在 $\mathbf{R}^N$ 中与 $\boldsymbol{x}$ 较为邻近的那些样本中使用上述距离度量，因为它是通过在 $\boldsymbol{x}$ 的局部近邻区域中对 $P(w_1 \mid \boldsymbol{x}')$ 做线性近似得到的。

计算 $D(\boldsymbol{x},\cdot)$ 的关键步骤是求 $\nabla P(w_1 \mid \boldsymbol{x})$，或者求 $\nabla P(w_1 \mid \boldsymbol{x})$ 的估计值。令 A 表示一个局部近邻区域，A 以 $\boldsymbol{x}$ 为中心，以 r_A 为欧氏意义下的半径，则有：若 $\| \boldsymbol{x}^l - \boldsymbol{x} \| \leqslant r_A$，则 $\boldsymbol{x}^l \in A$。

定义 $\boldsymbol{x}$ 处的一个局部期望向量为

$$\mu_i(\boldsymbol{x}) = E\{(\boldsymbol{x}^l - \boldsymbol{x}) \mid \boldsymbol{x}^l \in A, y^l \in \{w_1, w_2\}\} = \int_A (\boldsymbol{x}^l - \boldsymbol{x}) \frac{p(\boldsymbol{x}^l \mid w_i)}{u_i(\boldsymbol{x})} \mathrm{d}\boldsymbol{x}^l。 \tag{10}$$

其中：y^l 是 $\boldsymbol{x}^l$ 的类标签，$u_i(\boldsymbol{x}) = \int_A p(\boldsymbol{x}^l \mid w_i)\mathrm{d}\boldsymbol{x}^l$，$p(\boldsymbol{x}^l \mid w_i)$ 为 $\boldsymbol{x}^l$ 的类条件概率密度。将 $p(\boldsymbol{x}^l \mid w_i)$ 作泰勒级数展开为

$$p(\boldsymbol{x}^l \mid w_i) = p(\boldsymbol{x} \mid w_i) + \nabla p(\boldsymbol{x} \mid w_i)^{\mathrm{T}}(\boldsymbol{x}^l - \boldsymbol{x}) + \frac{1}{2}(\boldsymbol{x}^l - \boldsymbol{x})^{\mathrm{T}} \nabla^2 p(\boldsymbol{x} \mid w_i)(\boldsymbol{x}^l - \boldsymbol{x}) + \cdots$$

根据上式，式(10)可改写为

$$\mu_i(\boldsymbol{x}) = \frac{p(\boldsymbol{x} \mid w_i)}{u_i(\boldsymbol{x})} \int_A (\boldsymbol{x}^l - \boldsymbol{x})\mathrm{d}\boldsymbol{x}^l + \frac{\nabla p(\boldsymbol{x} \mid w_i)}{u_i(\boldsymbol{x})} \int_A (\boldsymbol{x}^l - \boldsymbol{x})(\boldsymbol{x}^l - \boldsymbol{x})^{\mathrm{T}}\mathrm{d}\boldsymbol{x}^l +$$

$$\frac{1}{2u_i(\boldsymbol{x})} \int_A (\boldsymbol{x}^l - \boldsymbol{x})(\boldsymbol{x}^l - \boldsymbol{x})^{\mathrm{T}} \nabla^2 p(\boldsymbol{x} \mid w_i) \cdot (\boldsymbol{x}^l - \boldsymbol{x})\mathrm{d}\boldsymbol{x}^l + \cdots$$

由于积分限 A 是以 $\boldsymbol{x}$ 点对称，上式中第一项和第三项为零，因此有

$$\mu_i(\boldsymbol{x}) \doteq \frac{\int_A (\boldsymbol{x}^l - \boldsymbol{x})(\boldsymbol{x}^l - \boldsymbol{x})^{\mathrm{T}}\mathrm{d}\boldsymbol{x}^l}{u_i(\boldsymbol{x})} \nabla p(\boldsymbol{x} \mid w_i)。 \tag{11}$$

因为 A 是以 $\boldsymbol{x}$ 为中心的一个小区域，因此可得近似式 $u_i(\boldsymbol{x}) \doteq V_A \cdot p(\boldsymbol{x} \mid w_i)$，其中，$V_A$ 为区域 A 的体积。推导后，式(10)可写为 $\mu_i(\boldsymbol{x}) \doteq c \dfrac{\nabla p(\boldsymbol{x} \mid w_i)}{p(\boldsymbol{x} \mid w_i)}$。其中，常数 c 为 $c = \dfrac{2r_A^2}{d+2}$，d 为样本空间的维数。

另外，可以在 A 上定义混合均值向量 $\mu_0(\boldsymbol{x}) = E\{(\boldsymbol{x}^l - \boldsymbol{x}) \mid \boldsymbol{x}^l \in A\}$。

经同样推导过程可得 $\mu_0(\boldsymbol{x}) \doteq c \dfrac{\nabla p(\boldsymbol{x})}{p(\boldsymbol{x})}$。

利用上述结果可以计算

$$\nabla P(w_i \mid \boldsymbol{x}) = \nabla[P(w_i) \frac{p(\boldsymbol{x} \mid w_i)}{p(\boldsymbol{x})}] =$$

$$P(w_i)\left[\frac{\nabla p(\boldsymbol{x}\mid w_i)}{p(\boldsymbol{x})}-\frac{p(\boldsymbol{x}\mid w_i)}{p^2(\boldsymbol{x})}\nabla p(\boldsymbol{x})\right]=\frac{1}{c}P(w_i\mid \boldsymbol{x})\left[\mu_i(\boldsymbol{x})-\mu_0(\boldsymbol{x})\right]。$$

用 $\mu_1(\boldsymbol{x})$ 和 $\mu_0(\boldsymbol{x})$ 的估计值 $M_1(\boldsymbol{x})$ 和 $M_0(\boldsymbol{x})$，利用上式可以得到 $\nabla P(w_1\mid \boldsymbol{x})$ 的估计值为

$$\hat{\nabla}P(w_1\mid \boldsymbol{x})=\frac{P(w_1\mid \boldsymbol{x})}{c}\left[M_1(\boldsymbol{x})-M_0(\boldsymbol{x})\right], \tag{12}$$

结合式(9)，不难看出，在用距离

$$D(\boldsymbol{x},\boldsymbol{x}^l)=\left|\ \hat{\nabla}P(w_1\mid \boldsymbol{x})^{\mathrm{T}}(\boldsymbol{x}-\boldsymbol{x}^l)\ \right|。 \tag{13}$$

度量时，寻找 $\boldsymbol{x}$ 的近邻 $\boldsymbol{x}^l$ 不受 $\hat{\nabla}P(w_1\mid \boldsymbol{x})$ 大小的影响，所以，可以忽略式(12) 前面的系数，得

$$\hat{\nabla}P(w_1\mid \boldsymbol{x})=M_1(\boldsymbol{x})-M_0(\boldsymbol{x})。$$

假设在区域 A 中有 N_A 个样本 $\boldsymbol{x}^l$，其中属于 w_i 类的有 N_{A_i} 个，则局部样本均值的近似估计为 $M_i(\boldsymbol{x})=\frac{1}{N_{A_i}}\sum_{N_{A_i}}(\boldsymbol{x}^l-\boldsymbol{x})$ 和 $M_0(\boldsymbol{x})=\frac{1}{N_A}\sum_{N_A}(\boldsymbol{x}^l-\boldsymbol{x})$，因此，定义最佳距离度量的表达式为

$$D_{\mathrm{opt}}(\boldsymbol{x},\boldsymbol{x}^l)=\left|\ (M_1(\boldsymbol{x})-M_0(\boldsymbol{x}))^{\mathrm{T}}(\boldsymbol{x}-\boldsymbol{x}^l)\ \right|。 \tag{14}$$

参照 TLNN 算法的思路，引入最佳距离度量，基于最佳距离度量的两层近邻分类算法描述如下。

1) 利用 AdaBoost 算法建立强分类器 $f(\boldsymbol{x})$。

2) 在下层，按照下面步骤为每个测试样本 $\boldsymbol{x}$ 选出 k_1 个近邻样本，组成子空间 $R(\boldsymbol{x})$。

① 用欧氏距离度量计算每个样本点与测试样本 $\boldsymbol{x}$ 的距离，$\|\boldsymbol{x}-\boldsymbol{x}_1\|,\cdots,\|\boldsymbol{x}-\boldsymbol{x}_N\|$；

② 找出与 $\boldsymbol{x}$ 距离最短的 N_A 个近邻 $\boldsymbol{x}^l$，$l=l_1,l_2,\cdots,l_{N_A}$；

③ 利用 $\boldsymbol{x}^l$ 计算 $M_1(\boldsymbol{x})-M_0(\boldsymbol{x})$；

④ 计算 $\left|\ [M_1(\boldsymbol{x})-M_0(\boldsymbol{x})]^{\mathrm{T}}(\boldsymbol{x}-\boldsymbol{x}^{l_1})\ \right|,\cdots,\left|\ (M_1(\boldsymbol{x})-M_0(\boldsymbol{x}))^{\mathrm{T}}(\boldsymbol{x}-\boldsymbol{x}^{l_{N_A}})\ \right|$，选择距离最小的 k_1 个样本，组成子空间 $R(\boldsymbol{x})$。即 $R(\boldsymbol{x})=\{\boldsymbol{x}^l\mid D_{\mathrm{opt}}(\boldsymbol{x},\boldsymbol{x}^l)\leqslant d_{(k_1)}\}$，其中 $d_{(k_1)}$ 是 $\{D_{\mathrm{opt}}(\boldsymbol{x},\boldsymbol{x}^l)\}_1^N$ 的第 k_1 小的值，$D_{\mathrm{opt}}(\boldsymbol{x},\boldsymbol{x}^l)$ 是式(14) 中的最佳距离度量。

3) 在上层，利用距离度量 $D_{\mathrm{TLNN}}(\boldsymbol{x},\boldsymbol{x}')$ 从子空间 $R(\boldsymbol{x})$ 中选定 k_2 个近邻建立一个新的局部子空间 $R'(\boldsymbol{x})=\{\boldsymbol{x}'\mid D_{\mathrm{TLNN}}(\boldsymbol{x},\boldsymbol{x}')\leqslant d_{k_2}\}$，其中 d_{k_2} 表示 $\{D_{\mathrm{TLNN}}(\boldsymbol{x},\boldsymbol{x}')\}_1^{k_1}$ 的第 k_2 小的值，$D_{\mathrm{TLNN}}(\boldsymbol{x},\boldsymbol{x}')$ 是由式(6) 定义的距离度量。

4) 对于给定的观测值 $\boldsymbol{x}$ 判定其标签：$y=\mathrm{sign}(\mathrm{ave}_{\boldsymbol{x}'\in R'(\boldsymbol{x})}y')$。

一般情况下 $N_A\ll N$，所以，上述算法所增加的计算时间相比 TLNN，是可以忽略的。

算法流程如图 3 所示。

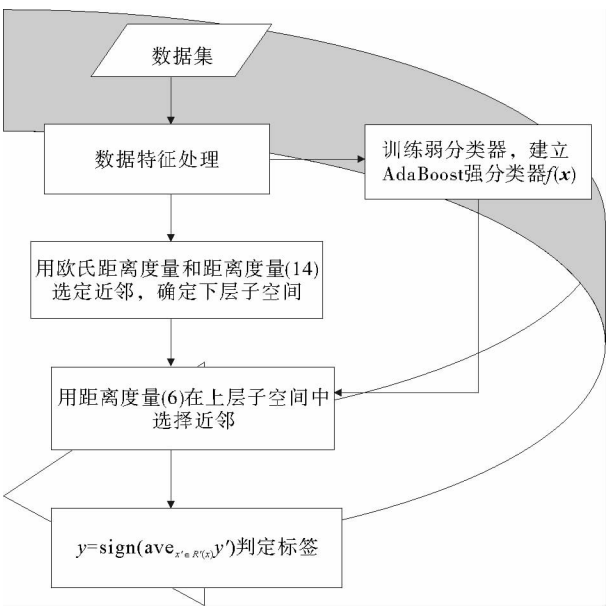

图 3 ODM-TLNN 算法流程图

Fig. 3 ODM-TLNN algorithm flow chart

3 实验结果及分析

本节通过使用 UCI 数据集对 AdaBoost 算法、TLNN 算法和 ODM-TLNN 算法进行比较，验证算法的精度和稳定性。

AdaBoost 算法能够将弱分类算法提升为强分类算法，它与支持向量机(support vector machine，SVM) 是效果最好的两种分类算法。TLNN 算法对 AdaBoost 算法做了一些改进，试图避免 AdaBoost 算法出现过拟合。加州大学欧文分校(University of California Irvine)提出的 UCI 数据库是用于机器学习的数据库。

从 UCI 数据库中挑选二分类数据集，如表 1 所示(属性值缺失的实例已删除)。

首先，将所有训练数据和测试数据的特征进行归一化处理到单位方差和零均值，标记数据类别为+1 和−1，所有实验均在 Matlab 平台进行；然后，将所有数据集都分为两个子集，两个子集随机划分，大小相等，一个为训练集一个为测试集，将过程重复 10 次，交换训练集与测试集，即交叉验证，结果取平均值。近邻数量关系在 ODM-TLNN 和 TLNN 算法中设置为 $k_1=2k_2+1$，此处设置 $k_2=1$。最大训练迭代次数设置为 $T=25$，以单特征值训练泛化型 AdaBoost 的弱分类器。

3.1 分类精度分析

首先将 AdaBoost、TLNN 和 ODM-TLNN 算法进行比较，实验结果如表 2 所示。

这三种算法数据分类的错误率如表 2 所示，表 2 中的加粗数字为最低错误率。从实验结果看，在 10 组数据中，AdaBoost 在 2 个组中取得了最低的错误率，TLNN 算法在 1 个组中取得了最低值，而 ODM-TLNN 算法在 7 组中取得了最优结果。相对于 AdaBoost 和 TLNN 算法，ODM-TLNN 算法的分类精度明显提高。

表 1 UCI 数据集的基本信息

Tab. 1 Basic information of UCI data set

名称	样本数	特征数
breast	699	10
heart	270	14
ionosphere	351	34
contraceptive	1 473	10
Clean1	476	167
banknote	1 372	5
spect	267	23
spectf	267	45
Monks-1(train+ test)	556	7
wpdc	198	34

表 2 实验结果

Tab. 2 Experiment results

名称	Adaboost	TLNN	ODM-TLNN
breast	0.045 2	0.039 6	**0.034 9**
heart	0.193 0	**0.184 8**	0.187 8
ionosphere	0.128 3	0.126 6	**0.124 9**
contraceptive	**0.285 4**	0.305 2	0.315 1
Clean1	0.148 4	0.139 1	**0.134 4**
banknote	0.016 2	0.004 7	**0.003 9**
spect	0.175 2	0.175 2	**0.170 3**
spectf	**0.208 3**	0.220 7	0.227 4
Monks-1(train+ test)	0.267 1	0.188 3	**0.167 6**
wpdc	0.256 2	0.244 8	**0.237 0**

3.2 抗噪声能力分析

抗噪声能力是衡量一个算法的重要指标，如支持向量机的抗噪性在理论和实践中都得到证明。以往研究表明，在噪声情况下，AdaBoost 和 TLNN 算法都容易出现过度拟合现象[12-13]。因此，需要测试 ODM-TLNN 算法在噪声数据影响下的性能。

在表 1 的 10 组数据中引入标签噪声，即随机挑选训练集中的部分数据，然后调换它们的标签，以此作为噪声。通过这种方式，在数据集构造 5%，10%，15%和 20%的随机噪声。表 3～6 显示了 4 种噪声数据情况下 AdaBoost，TLNN 和 ODM-TLNN 算法分类错误率的对比结果，表中将错误率最低的数据加黑。从实验结果可以看出，AdaBoost 算法在 4 组噪声实验结果中取得最低分类错误率的个数分别为 2，1，1，1，而 TLNN 算法在 4 组噪声实验结果中取得最低分类错误率个数分别为 0，1，0，0，而 ODM-TLNN 算法取得最低分类错误率的数据集数量分别为 8，8，9，9。可见，随着噪声数据增加，AdaBoost 和 TLNN 算法分类效果逐渐变差，而 ODM-TLNN 算法效果逐渐变优。表明 ODM-TLNN 算法在噪声情况下具有更好的抗噪性能，稳定性强且有更好的泛化能力。

表 3 5%噪声数据分类错误率

Tab. 3 Wrong classification ratio with 5% noisy data

名称	Adaboost	TLNN	ODM-TLNN
breast	0.073 3	0.064 7	**0.060 0**
heart	0.216 7	0.208 9	**0.207 8**
ionosphere	0.163 4	0.152 6	**0.150 3**
contraceptive	**0.303 7**	0.326 4	0.330 8
Clean1	0.191 0	0.171 1	**0.161 7**
banknote	0.054 5	0.036 9	**0.033 2**
spect	0.206 0	0.206 8	**0.201 1**
spectf	**0.248 1**	0.248 5	0.250 4
Monks-1(train+test)	0.286 9	0.218 5	**0.213 7**
wpdc	0.301 0	0.279 7	**0.276 0**

表 4 10%噪声数据分类错误率

Tab. 4 Wrong classification ratio with 10% noisy data

名称	Adaboost	TLNN	ODM-TLNN
breast	0.123 2	0.126 5	**0.121 6**
heart	0.268 9	0.257 0	**0.250 0**
ionosphere	0.219 1	0.205 7	**0.196 3**
contraceptive	**0.324 7**	0.344 9	0.351 4
Clean1	0.228 9	0.213 7	**0.204 7**
banknote	0.104 1	0.089 9	**0.087 3**
spect	0.249 2	0.249 2	**0.239 1**
spectf	0.295 1	**0.294 7**	0.299 2
Monks-1(train+test)	0.316 2	0.278 7	**0.268 7**
wpdc	0.322 9	0.304 7	**0.300 5**

表 5 15%噪声数据分类错误率

Tab. 5 Wrong classification ratio with 15% noisy data

名称	Adaboost	TLNN	ODM-TLNN
breast	0.196 8	0.194 7	**0.191 5**
heart	0.299 3	0.298 1	**0.293 0**
ionosphere	0.266 3	0.246 3	**0.240 9**
contraceptive	**0.351 0**	0.368 9	0.373 6
Clean1	0.292 2	0.266 0	**0.255 9**
banknote	0.156 9	0.146 0	**0.142 8**
spect	0.3071	0.2932	**0.292 1**
spectf	0.315 0	0.316 5	**0.313 5**
Monks-1(train+test)	0.352 3	0.317 3	**0.312 4**
wpdc	0.369 8	0.362 5	**0.354 2**

表 6 20%噪声数据分类错误率

Tab. 6 Wrong classification ratio with 20% noisy data

名称	Adaboost	TLNN	ODM-TLNN
breast	0.253 6	0.252 1	**0.251 2**
heart	0.348 5	0.342 6	**0.340 0**
ionosphere	0.309 4	0.302 6	**0.294 6**
contraceptive	**0.388 9**	0.402 6	0.405 8
Clean1	0.346 1	0.326 2	**0.309 0**
banknote	0.209 8	0.205 5	**0.201 3**
spect	0.333 8	0.326 7	**0.324 8**
spectf	0.366 2	0.363 9	**0.359 4**
Monks-1(train+test)	0.377 3	0.360 3	**0.355 6**
wpdc	0.409 9	0.395 8	**0.392 2**

在 ODM-TLNN 算法中，由于在下层用最近邻算法建立局部子空间时引入最佳距离度量，在均方意义下将有限样本的错误率降到与无限样本下的错误率相近，可以最大程度地减少噪声数据的影响，防止 AdaBoost在噪声数据存在时过度拟合现象的发生，提高了分类精度和稳定性。

4 结束语

结合 AdaBoost 和 kNN 算法的优势，提出一种基于最佳距离度量的两层近邻分类算法。最佳距离度量可在均方意义下将有限样本的错误率降到与无限样本下的错误率相近，而且所增加的计算时间是可以忽略的。通过在 UCI 数据集上的实验表明，该算法能充分降低分类错误率，并且在噪声数据下有较好的稳定性。

参考文献：

[1]Cover T M, Hart P E. Nearest neighbor pattern classification[J]. IEEE Transactions on Information Theory, 1967, 13(1): 21-27.

[2]Goldberger J, Roweis S, Hinton G, et al. Neighborhood components analysis[J]. Advances in Neural Networks, 2005, 16(4): 899-909.

[3]Domeniconi C, Gunopulos D. Adaptive nearest neighbor classification using support vector machines[M]//Advances in Neu-

ral Information Processing Systems. Cambridge, MA, 2002: 655-672.

[4]卢伟胜,郭躬德,严宣辉,等. 基于类别子空间距离加权的互 k 近邻算法[J]. 计算机科学,2014,41(2):166-169.
Lu Weisheng, Guo Gongde, Yan Xuanhui, et al. Mutual k-nearest neighbours algorithm based on class subspace and distance-weighted[J]. Computer Science, 2014, 41(2): 166-169.

[5]Gou J, Du L, Zhang Y, et al. A new distance-weighted k-nearest neighbor classifier[J]. Journal of Information and Computational Science, 2012(9): 1429-1436.

[6]林耀进,李进金,陈锦坤,等. 融合邻域信息的 k-近邻分类[J]. 智能系统学报,2014,9(2):240-243.
Lin Yaojin, Li Jinjin, Chen Jinkun, et al. k-nearest neighbor classification algorithm fusing neighborhood information[J]. CAAI Transactions on Intelligent System, 2014, 9(2): 240-243.

[7]Yang L, Jin R, Sukthankar R, et al. An efficient algorithm for local distance metric learning[C]//AAAI: The 21st National Conference on Artificial Intelligence. Boston, MA, July, 2006: 543-548.

[8]张兴福,黄少滨. 基于马氏距离的局部线性嵌入算法[J]. 模式识别与人工智能,2012,25(2):318-324.
Zhang Xingfu, Huang Shaobin. Mahalanobis distance measurement based locally linear embedding algorithm[J]. Pattern Recognition and Artificial Intelligence, 2012, 25(2): 318-324.

[9]Gao Y L, Pan J Y, Ji G L, et al. A novel two-level nearest neighbor classification algorithm using an adaptive distance metric [J]. Knowledge Based Systems, 2012, 26: 103-110.

[10]边肇祺,张学工. 模式识别[M]. 2 版. 北京:清华大学出版社,1999:156-158.

[11]Freund Y, Schapire R. A decision-theoretic generalization of on-line learning and application to boosting[J]. Journal of Computer and System Sciences, 1997, 55(1): 119-139.

[12]仝伯兵,王士同. 基于概率的两层最近邻自适应度量分类算法[J]. 计算机工程与应用,2014(8):1-8.
Tong Bobing, Wang Shitong. Probability-based two-level nearest neighbor classification algorithm for adaptive distance[J]. Computer Engineering and Applications, 2014(8): 1-8.

[13]仝伯兵,王士同,梅向东. 稀疏条件下的两层分类算法[J]. 智能系统学报,2015,10(1):27-36.
Tong Bobing, Wang Shitong, Mei Xiangdong. Sparsity-inspired two-level classification algorithm[J]. CAAI Transactions on Intelligent Systems, 2015, 10(1): 27-36.

(责任编辑:吕文红)

(上接第 19 页)

[8]曹春红,王鹏,曹犁歌. 基于 D-tree 分解的欠约束与完备约束的几何约束求解[J]. 东北大学学报:自然科学版,2014,35(5):626-629.
Cao Chunhong, Wang Peng, Cao Lige. Well-constrained and under-constrained geometric constraint solving based on D-tree decomposition[J]. Journal of Northeastern University: Natural Science, 2014, 35(5): 626-629.

[9]Fudos I, Hoffmann C. A graph-constructive approach to solving systems of geometric constraints[J]. ACM Transactions on Graphics, 1997, 16(2): 179-216.

[10]高小山,黄磊东,蒋鲲. 求解几何约束问题的几何变换法[J]. 中国科学:E 辑,2001,31(2):182-192.
Gao Xiaoshan, Huang Leidong, Jiang Kun. Geometry transformation approach to solving systems of geometric constraints [J]. Science in China: Series E, 2001, 31(2): 182-192.

(责任编辑:吕文红)