SVM 和集成学习算法的改进和实现①

魏仕轩, 王未央

(上海海事大学 信息工程学院, 上海 201306)

摘 要: 支持向量机(SVM)算法的主要缺点是当它处理大规模训练数据集时需要较大内存和较长的训练时间. 为了加快训练速度和提高分类准确率, 提出了一种融合了 Bagging, SVM 和 Adaboost 三种算法的二分类模型, 并提出了一种去噪的算法. 通过实验对比 SVM, SVM-Adaboost 以及本文提出的分类模型. 随着训练数据规模不断扩大, 该分类模型在提高准确率的前提下, 明显提高了训练速度.

关键词: Bagging; SVM; Adaboost; 集成学习; 噪声处理; 分类

Improvement and Implementation of SVM and Integrated Learning Algorithm

WEI Shi-Xuan, WANG Wei-Yang

(College of Information Engineering, Shanghai Maritime University, Shanghai 201306, China)

Abstract: The main drawback of support vector machine (SVM) algorithm is that it needs large memory and long training time while handling large training data set. In order to speed up the training and improve classification accuracy, this paper proposes a binary classification model, which fuses the Bagging, SVM and Adaboost algorithm. And a kind of denoising algorithm is proposed. Contrast SVM, the SVM-Adaboost and classification model proposed in this paper by experiment. With the expanding of training data, this classification model has improved training speed significantly under the premise of improving accuracy.

Key words: Bagging; SVM; Adaboost; integrated learning; noise processing; classification

SVM 在解决非线性高维模式时有一些特殊的优势, 但是对于大训练样本, SVM 需要花费大量的时间在计算上, SVM 是借助二次规划来求解支持向量[1], 而求解二次规划将涉及 m 阶矩阵的计算(m 为样本个数), 当 m 数目很大时, 该矩阵的存储和计算将耗费大量的机器内存和运算时间, 因此SVM算法不适合解决大样本问题. 张晓龙, 任芳利用 SVM 和 Adaboost 结合的想法, 虽然提高了预测精确度[2,3] , 但是数据的计算规模并没有减少, 这导致对大数据集预测将花费大量的时间. 而本文利用Bagging算法, 每训练一个弱分类器只是从原来的整个训练数据集中选择一小部分用 SVM-Adaboost 进行训练, 这样不仅能提高预测精确度, 计算时间也大大降低.

由于Adaboost方法对于噪声数据和异常数据很敏感, 而训练数据一般都含有一些噪声, 这些数据将对分类结果产生重要的影响. 一般有 2 种噪声: 属性噪声和样本噪声. 所以本文针对这 2 个方面, 提出了一种去噪算法, 该算法不是去除掉噪声数据, 而是去发现噪声数据, 并且根据它所在的类别的统计数据去修改它的类标签或者它的某些属性值. 因此这种方法不仅能够减弱噪声对分类的影响, 同时还保持了训练集原有的规模.

本文Bagging算法利用SVM-Adaboost作为基分类模型,虽然随着基分类器个数的增加, 训练误差会逐渐趋向与0, 但是随着基分类器个数的增加, 样本的训练时间会大大增加, 同时学习难度的增加, 容易造成过度拟合, 导致分类效率下降, 稳定性变差. 为了能够提高集成学习系统的泛化性能, 构建泛化能力强的分类器集合, 我们通过控制基分类器的数量(SVM-Adaboost), 使得在训练误差足够小的情况下,

① 收稿时间:2014-10-24;收到修改稿时间:2015-03-12

提高学习效率. 这就要求选择的基分类器具有一定的质量. 本文提出了一种筛选基分类器的算法, 最后得到一组预测准确率高的基分类器.

1 相关算法和流程

1.1 算法流程

① 输入数据集 S;

② 对 S 数据集随机抽样, 用 SVM-Adaboost 算法进行仿真实验, 确定弱分类器数范围, 并在这个范围内选择弱分类器个数(既考虑精确率, 又考虑时间的效率);

③从 S 中按照某种分布抽取子训练集 $(S_1, S_2, ..., S_k)$;

④ 对每一个子训练集进行如下操作;

1) 用本文设计的去噪算法进行噪声处理;

2) 用 SVM-Adaboost 算法对子数据集进行训练;

3) 通过本文筛选方法筛选弱分类器个数为步骤②选择的个数;

4) 整合弱分类器得到每个子数据集的最终分类模型, 作为 Bagging 算法的一个基本分类器;

⑤ 根据下文改进的 Bagging 算法, 整合 K 个基本分类器(SVM-Adaboost)得到最终的分类决策函数(本文仿真实验 K 取 5).

1.2 SVM-Adaboost[4](作为 Bagging 算法中基分类器)

SVM 是一种二类分类模型, 其基本模型定义为特征空间上的间隔最大的线性分类器, 其学习策略便是间隔最大化, 最终可转化为一个凸二次规划问题的求解.

SVM 的主要思想:

(1) 它是针对线性可分情况进行分析, 对于线性不可分的情况, 通过使用非线性映射算法将低维输入空间线性不可分的样本转化为高维特征空间使其线性可分, 从而使得高维特征空间采用线性算法对 样本的非线性特征进行线性分析成为可能.

(2) 它基于结构风险最小化理论之上在特征空间中建构最优分割超平面, 使得学习器得到全局最优化, 并且在整个样本空间的期望风险以某个概率满足一定上界. 流程如下:

设给定训练样本 $(x_1, y_1), (x_2, y_2)$ ……, 其中, $x_i \in R_n, y_i \in \{-1,1\}$, 对于 x 作非线性变换, 使 x 成为线性可分, 则可找到向量 w 和常量 b 满足:

$$y_i(\mathrm{wx}+\mathrm{b}) \geq 1$$

$\mathrm{i} \in \{1,2,,....,\mathrm{m}\}$ 是 m 个训练样本. 则 f(x)=wx+b 就是能够线性分开的超平面.

要使分类距离最大, 即求函数 $\oint(x) = |w|^2$ 的最小值, 可以定义如下的 Lagrange 函数:

$$L(w,b,c) = \frac{1}{2}|w|^2 - \sum_{i=1}^{m} c_i[y_i(wx_i + b) - 1]$$

其中, c_i 为 Lagrange 系数, 于是就将问题转化成了对 w 和 b 求 Lagrange 函数的极小值. 每一个 Lagrange 系数对应于一个训练样本 x_i. 最后得到的分类函数为:

$$f(x) = \mathrm{sgn}(wx+b) = \mathrm{sgn}(\sum_{i=1}^{N_s} c_i y_i x + b)$$

AdaBoost 算法是目前 Boosting 算法中最常用的方法, 特别在人脸识别中得到了广泛的应用[5,6].

该算法针对相同训练集的不同分布训练同一个基本分类器(弱分类器), 然后把这些在不同分布训练集上得到的分类器集合起来, 构成一个更强的最终的分类器(强分类器). 理论证明, 只要每个弱分类器分类能力比随机猜测要好, 当其个数趋向于无穷个数时, 强分类器的错误率将趋向于零. 而 SVM 作为精确度很高的分类器, 把它作为 AdaBoost 算法的弱分类器可以达到更好的分类效果.

但是 SVM 算法处理大规模训练数据集时需要较大内存和较长的训练时间. 为了加快训练速度和提高分类准确率, 因此本文融合了 Bagging 算法.

SVM-Adaboost 具体实现过程:

输入: 一组样本集 $\{(x_1, y_1), (x_2, y_2), ..., (x_n, y_n)\}$, 其中 x_i 表示样本的属性向量, y_i 表示类别标签, $y_i \in Y = \{1,2,.....,k\}$ 识别算法迭代次数为 T;

初始化: 对于所有 $1 \leq i \leq n, D_1(i) = 1/n$,($D_1(i)$ 表示在第 1 次迭代时, 样本 (x_i, y_i) 占的权重, 样本权重将影响样本的分布)

训练过程: For t ={1,2,..,T}:

根据 D_t, 训练弱识别器(本文基于 SVM 算法来训练)

根据上一步的分类器, 输入样本的属性向量集合 $X = \{x_1, x_2, ..., x_n\}$, 得到一个分类器

$$h_t : X \to Y = \{1,2,...,n\}$$

计算 h_t 的错误率: $\varepsilon_t = \sum_{i:h_t(x_i) \neq y_i} D_t(i)$, 同时令 $\beta_t = \varepsilon_t / (1-\varepsilon_t)$

改变样本的权重和分布:

$$D_{t+1}(i)=\begin{cases}\beta_t\times\dfrac{D_t(i)}{Z_t}, h_t(x_i)=y_i\\ \dfrac{D_t(\mathrm{i})}{Z_t}, h_t(x_i)\neq y_i\end{cases}\qquad Z_t=\sum_{i=1}^{l}D_t(i)$$

输出: 强分类器 $H(x)=\sum_{t=1}^{T}\beta_t h_t(x)$(作为 Bagging 算法的弱分类器)

1.3 Bagging[7,8]

本文利用 Bagging 算法把大数据集分成许多小数据集(因为 SVM 适合于小数据集). 算法流程如下:

从S中按照某种分布独立抽取训练集 S_1;

do for t=1,2,...,T

① 用弱学习算法SVM-Adaboost, 在训练样本子集 S_t 上训练得到弱分类器:

$$h_t(x):x\to\{-1,+1\}$$

② 计算h(x)的错误率:

$$\varepsilon_t=\sum_{(x_i,y_i)\in S_t}[h_t(x_i)\neq y_i]/|S_t|$$

($|S_t|$代表 S_t 中的数量)令:

$$\beta_t=\varepsilon_t/(1-\varepsilon_t),\alpha_t=\frac{1}{2}\log(1/\beta_t);$$

③ 按照某种分布再次独立抽取训练集 S_{t+1};

循环结束后, 最终的决策函数为(强分类器):

$$H(x)=sign(f(x)),f(x)=\sum_{t=1}^{T}\alpha_t h_t(x).$$

1.4 去噪处理

由于基分类器中用到 Adaboost, 它对噪声很敏感, 容易产生过适应现象, 因此对于每一个小数据集, 先进行去噪处理. 本文设计的去噪算法针对二类数据集, 此算法不改变数据集的规模大小.

该去噪方法针对以下 2 种情况:

1) 训练样本中有贴错标签的样本, 找出来予以改正;

2) 训练样本中某一个属性的值出现偏差, 找出并改为均值;

具体的去噪算法如下:

对于二分类数据集的噪声处理: (设正类为 1, 负类为 0)

① 对所有正类样本, 计算每个属性的均值为 C_{1j}

对所有负类样本, 计算每个属性的均值为 C_{0j}

(其中j=1,2.....m, m为输入属性的个数)

② 设置一个计数器n 初始值为 0,

α_{1j} (表示正样本的第j个属性值)

对每个正样本进行以下操作

For (j=1; j<=m;j++)

{

If ($|\alpha_{1j}-c_{1j}|>|\alpha_{1j}-c_{0j}|$) //表示该样本点在属性j上更偏向于负类

n++;

}

If(n>m/2) //表示该样本点更偏向于负类

{

(1) 改变该样本的标签为负类

(2) 并且改变所有偏向于正类的属性值(即 $|\alpha_{1j}-c_{1j}|>|\alpha_{0j}-c_{1j}|$ 的这些属性值)为负类对应属性值的均值

}

Else //表明标签没有错, 即还是更偏向于正类的

{

改变那些偏向负类的属性值(即 $|\alpha_{1j}-c_{1j}|>|\alpha_{0j}-c_{1j}|$ 的这些属性值)为正类对应属性值的平均值

}

③ 同样对每个负样本 α_{0j} 进行同样的操作

④ 所有的样本都操作过一次, 查看是否有样本需要改动, 如果有的话, 那么返回步骤①继续迭代, 直到没有样本需要修改标签或者属性为止

1.5 弱分类器的筛选

设弱分类器(SVM-Adaboost)数目为 L, 集成规模为 S

1) 去掉正确率小于 0.6 的弱分类器, 去掉之后, 如果弱分类器个数还是大于S的话, 进行第二步筛选[9];

2) 对于某一个弱分类器(SVM-Adaboost)最终的决策函数 h(x), 是通过 h(x)的符号来判断样本属于正样本还是负样本的, 并且|h(x)|越小表明分类的越没用把握, 所以统计每个弱分类器, 计算|h(x)<e| (e 是一个很小的数)样本的个数, 去掉最大的前几个分类器使得最后剩余 S 个弱分类器.

2 实验结果及讨论

2.1 实验说明

本实验在 matlab R2010 环境下仿真实现. 个人电

脑: Intel Core i5 CPU 主频 2.53GHZ, 内存 2GB.

2.2 结果与讨论

实验一. 用本文模型对 UCI 3 种数据集(a1a, a7a, a8a)进行抽样实验

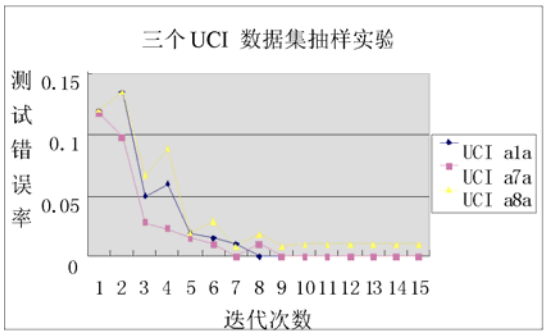

图 1　抽样实验

从图 1 可以看出, 3 种数据集的测试错误率大部分情况下都是先随弱分类器个数增加而减少, 随着个数的不断增加, 错误率趋向于平稳, 甚至会出现上升的可能. 由图可以看出当弱分类器个数在 5-10 次时能到达或接近错误率的最小值. 当弱分类器个数很少时, 达不到充分学习的效果, 而当弱分类器个数过多时, 用大量的时间可能只换得微小的准确率, 并且在噪声数据比较多的情况下, 也可能会出现过学习的情况. 所以, 我们可以选取 5-10 个作为 SVM-Adaboost 算法的弱分类器个数, 如 7 次. 所以本文可以训练 10 个弱分类器, 然后用筛选算法进行筛选最终保留 7 个高质量的弱分类器集成为强分类器[10].

实验二. 分别对 UCI 数据集 a1a,a7a,a8a 用 SVM, SVM-Adaboost 和本文方法进行仿真实验, 比较它们的准确率和运行时间:

表 1　三种算法性能比较

UCI 数据集	SVM	SVM-Adaboost	本文方法
	准确率(%)	准确率(%)	准确率(%)
	运行时间	运行时间	运行时间
a1a	87.129	92.350	99.600
1-2000 行	3.736s	51.352s	10.056s
a7a	86.112	87.441	92.480
1-1000 行	71.523s	1376.795s	254.251s
a8a	85.933	87.324	90.773
1-1500 行	159.687s	3107.410s	594.247s
均值	86.391	89.038	94.284
	78.315s	1511.852s	286.185s

从表 1 可以看出, SVM 运行时间较其它两种算法的运行时间较少, 但它的准确率最低[11]. SVM-Adaboost 算法利用集成学习方法[12,13], 虽然提高了预测准确率, 但是运行时间较长. 而且随着数据量的增加, 效果会下降. 针对这 2 种算法的缺点, 本文提出来一种方法, 它利用 Bagging 算法思想, 每次训练的时候只是抽取大数据集中的一部分, 大大降低了运行时间. 另一方面, 该方法运用了特殊的去噪方法, 对弱分类器进行了筛选优化, 以及对最终的分类决策函数的改善(不再是普通的加权, 而是根据弱分类器的精度来确定它的系数), 从而提高了预测精确度. 从表中可以看出, 随着数据量的增加, 本文方法的效果越来越明显.

3　结语

本文主要运用集成学习思想, 针对二分类问题, 提出了一种学习算法. 由于 SVM 算法对大数据集的预测效果不明显, 本文利用 Bagging 算法, 每次从原数据集中按照某种分布选择一部分数据, 用 SVM-Adaboost 算法进行学习. Adaboost 算法对噪声数据比较敏感, 本文也提出来一种噪声处理算法, 该算法不是去除数据, 而是纠正数据, 从而保证了数据集的规模. 通过实验仿真和筛选算法, 选出 SVM-Adaboost 算法中高质量的弱分类器, 并确定个数. 综上所述, 本文的方法在提高了预测准确率的前提下, 也降低了运行时间. 最后要说明的是, 本文所提出来的方法对小数据集不是很明显, 但随着数据量的增大, 效果会越来越明显.

参考文献

1 Xiang AL, Yang XT. Pedestrian detection of based on Adaboost-SVM cascade classifier. Computer Engineering and Design, 2013, 34(7): 2547–2550.

2 Zhang XL, Ren F. The research of combination of support vector machine (SVM) and AdaBoost algorithm. Computer Application Research, 2009, 26(1): 77–100.

3 Liu WH. The comparison and research of MK-LSSVM and AdaBoost-SVM in classification. Automation Instrument, 2013, 34(5): 13–18.

4 Quan LT. The solution of real-time traffic identification based on SVM and Adaboost. Softwear, 2013, 34(9): 61–64.

5 Viol P, Jones M. Rapid object detection using Aboosted cascade of simple features. Proc. of IEEE Conf. Computer Vision and Pattern Recognition. Pisscatway. IEEE. 2001. 511–518.

6 Wu B, Huang C, Ai HZ, et al. A multi view face detection based on real AdaBoost algorithm. Journal of Computer Research and Development, 2005, 42(9): 1612–1621.

7 Fu ZL. About AdaBoost effectiveness analysis. Computer Research and Development, 2008, 45(10): 1747–1754.

8 Breiman L. Bagging predicators. Machine Learning, 1996, 24(2): 123–140.

9 Yu X, Wang XD, Yao X, Bi K. Multi-polarization HRRP recognition of dynamic integration based on Bagging-SVM. Systems Engineering and Electronics, 2012, 34(7): 1367–1370.

10 Zhao S, Li X, Liu WH. Bagging text categorization algorithm based on classifier performance evaluation. Computer Engineering, 2008, 34(1): 61–63.

11 Chen L, Gunduz S, Ozsu MT. Mixed type audio classificationwith support vector machine. IEEE International Conference on Multimedia and Expo. 2006. 781–784.

12 Chen D, Wang JL, Zhou Y. Face detection method research and implementation based on AdaBoost. Proc. of the 2010 International Symposium on Intelligence Information Processing and Trusted Computing. 2010. 643–646.

13 Chang TT, Liu HW. Largescale classification with local diversity AdaBoost SVM algorithm. Journal of Systems Engineering and Electronics, 2009, 20(6): 1344–13.