计算机科学 COMPUTER SCIENCE
http://www.jsjkx.com
DOI:10.11896/jsjkx.181001921

攻击标签信息的堆栈式支持向量机

金　耀[1]　徐丽亚[1]　吕慧琳[1]　顾苏杭[2,3]
1 常州大学信息科学与工程学院　江苏 常州 213164
2 江南大学数字媒体学院　江苏 无锡 214122
3 常州轻工职业技术学院信息工程学院　江苏 常州 213164
(398124657@qq.com)

摘　要　真实数据集中存在的对抗样本易导致分类器取得较差的分类性能,但如果其能够被合理利用,分类器的泛化能力将得到显著提高。针对现有大部分分类器并没有涉及对抗样本信息的问题,提出一种攻击标签信息的堆栈式支持向量机。该方法从给定的初始数据集中选取一定比例的样本,并攻击所选取样本的标签,使之成为对抗样本,即将样本标签替换成其他不同类型的标签,利用支持向量机训练包含对抗样本的数据集,从而生成对抗支持向量机。计算对抗支持向量机的输出误差相对于输入样本的一阶梯度信息,并将其嵌入到输入样本特征中以更新输入样本。将更新后的样本输入到下一个对抗支持向量机中,并重新训练。以堆栈方式级联一定数目的对抗支持向量机,直至取得最好的分类性能。原理分析与实验结果表明,基于对抗样本的一阶梯度信息不仅提供了分类器输出与输入之间的一种正相关关系,而且为堆栈式支持向量机中的子分类器提供了一种新的堆栈方式,并提高了分类器的整体性能。

关键词:堆栈结构;对抗样本;标签攻击;支持向量机
中图法分类号　TP391.4

Stacked Support Vector Machine Based on Attacks on Labels of Data Samples

JIN Yao[1],XU Li-ya[1],LV Hui-lin[1] and GU Su-hang[2,3]
1 School of Information Science and Engineering,Changzhou University,Changzhou,Jiangsu 213164,China
2 School of Digital Media,Jiangnan University,Wuxi,Jiangsu 214122,China
3 College of Information Engineering and Technology,Changzhou Vocational Institute of Light Industry,Changzhou,Jiangsu 213164,China

Abstract　As for the adversarial data samples which indeed exist in real-world datasets,they can mislead data classifiers into correct predictions which results in poor classification. However,reasonable utilization of the adversarial data samples can distinctly improve the generalization of data classifiers. Since most of existing classifiers do not take the information about adversarial data samples into account to build corresponding classification models,a stacked support vector machine called S-SVM based on attacks on the labels of data samples which aims to obtain outperformed classification performance by learning the adversarial data samples was proposed. In a given dataset,a certain percentage of data samples are randomly chosen as adversarial data samples,in other words,the labels of these chosen data samples are substituted by the other labels included in the given dataset which are different from the original labels of the chosen data samples. Adversarial support vector machine (A-SVM) can be consequently generated by using the support vector machine (SVM) to train the given dataset which contains the adversarial data samples. The first-order gradient information on the output error of the generated A-SVM with respect to the input samples can be then computed,and the input samples will be updated by embedding the first-order gradient information into the original feature space of the input samples. Consequently,the updated data samples can be input into next A-SVM to be trained again to gradually improve the classification performance of the current A-SVM. As a result,S-SVM is formulated by stacking some A-SVMs layer by layer,the best classification results can also be obtained by the corresponding S-SVM. In terms of theoretical analysis and experimental results on UCI and KEEL real-world datasets,the mathematically computed first-order gradient information based on learning the adversarial data samples not only provide a positive relation between the outputs and the inputs of a classifier,but also indeed provide a novel way to stack the front and rear sub-classifiers in the proposed S-SVM.

Keywords　Stacked structure,Adversarial data samples,Attacks on labels,Support vector machine (SVM)

到稿日期:2018-10-15　返修日期:2019-04-26　本文已加入开放科学计划(OSID),请扫描上方二维码获取补充信息。
基金项目:国家自然科学基金(81701793);常州市科技计划项目(CJ20190016)
This work was supported by the National Natural Science Foundation of China (81701793) and Science and Technology Project of Changzhou (CJ20190016).
通信作者:顾苏杭(gusuhang09@163.com)

数据分类技术一直是机器学习、人工智能等领域的研究热点，已被广泛应用于图像识别、自然语言处理、语音识别、智能交通及医疗辅助诊断等[1-4]。数据分类技术通过训练或者学习给定的数据样本建立数据分类模型，从而对未知的数据样本进行预测和识别[5-7]。然而，真实数据集中会存在不易被察觉的扰动（被称作对抗样本[8-11]），这些对抗样本易导致所训练的数据分类器在未知样本上出现错误分类，从而大大降低分类器的实际分类性能。因此，如何有效处理并利用数据集中存在的对抗样本来训练数据分类器，已逐步成为数据分类技术的重要研究问题之一。

Mosca 等[8]将包含扰动的样本输入神经网络，并利用输出的结果对当前输入样本进行一阶求导；解得的一阶梯度信息被嵌入到当前输入样本特征中，更新后的样本再次被输入到神经网络进行训练，由此生成的神经网络的泛化能力得到明显提高。文献[9]将微小且合理的扰动加入到样本特征中，人为生成对抗样本，训练包含对抗样本的训练集从而生成的深度神经网络（Deep Neural Network，DNN）可被有效地应用于恶意软件检测。马玉琨等[10]针对 DNN 应用于活体检测时性能易受对抗样本干扰的问题，从样本特征维度的角度考虑，将对抗样本干扰集中在少数几个样本特征维度，从而提出一种最小扰动维度的活体检测对抗样本生成技术。该技术只需要对样本的少数几个特征维度做扰动便可生成对抗样本。Gu 等[11]在研究对抗样本结构的基础上，在 DNN 输入层中将扰动加入到样本特征中使部分样本成为对抗样本，训练生成的深度感知网络可以很好地抑制样本噪声，改善分类性能。一方面，神经网络及其改进技术由输入层、隐藏层及输出层组成，在输出层学习输出误差梯度，并利用反向传播算法（Back Propagation，BP）不断调整隐藏层中每个单元的权重与偏值，以提高神经网络的分类性能[8-11]；另一方面，在实际数据分类的过程中，由于每个真实数据集都会包含对抗样本，因此本文从合理利用对抗样本的落脚点出发，结合支持向量机提出一种攻击标签信息的堆栈式支持向量机。所提分离器由一定数目的子分类器按照栈式结构堆栈而成[12-13]。

不同于文献[8-11]所提的攻击样本特征本身的对抗样本生成方法，即将合理的样本扰动直接嵌入到样本特征中，本文将通过攻击一定比例样本的标签生成对抗样本，即将样本标签随机替换成给定数据集中的其他不同类型的标签，再利用 SVM 训练包含对抗样本的训练集来生成对抗支持向量机（Adversarial Support Vector Machine，A-SVM）。所提堆栈式支持向量机 S-SVM 将 A-SVM 作为子分类器，以堆栈结构的方式级联一定数目的 A-SVM，直至取得最好的分类性能。具体地，计算当前 A-SVM 的输出误差对所有训练样本的一阶梯度信息，并将该信息嵌入到相应训练样本特征中以更新训练样本。同样地，攻击一定比例的更新后的训练样本标签信息，将更新后且包含对抗样本的训练样本输入到下一个 A-SVM 中并重新训练，以堆栈的方式[12-13]逐渐提高每一个子分类器 A-SVM 的分类性能，即逐渐提高 S-SVM 的分类性能。

1　对抗支持向量机

本文所提对抗支持向量机以支持向量机为基础，因此 1.1 节简要介绍支持向量机的基本原理；1.2 节阐述如何利用对抗样本训练 A-SVM 模型；1.3 节从原理上详细分析为什么对抗样本学习能够提高 A-SVM 的分类性能。

1.1　支持向量机

Vapnik 等[14]在统计学习理论的基础上提出 SVM 方法来解决分类问题，其基本原理是通过运用非线性变换将输入样本特征空间变换到一个高维特征空间，使得样本达到线性可分的效果。本文将线性 SVM（Linear SVM）作为基础，以二分类问题为例，即样本标签为 $y\in\{-1,1\}$，当输入一个样本 $\boldsymbol{x}=[x_1,x_2,\cdots,x_d]$ 时，d 为样本特征维数，SVM 的决策函数可表达为：

$$f(\boldsymbol{x};\boldsymbol{w},b)=\mathrm{sign}(\boldsymbol{w}^{\mathrm{T}}\boldsymbol{x}+b)\in\{-1,1\} \tag{1}$$

其中，$\mathrm{sign}(h)$ 为符号函数，即当 $h>0$ 时 $\mathrm{sign}(h)=1$，反之 $\mathrm{sign}(h)=0$；$\boldsymbol{w}=[w_1,w_2,\cdots,w_d]$ 和 b 为 SVM 在样本特征空间中确定决策平面的两个参数。通过训练样本生成的最优 SVM 模型旨在寻找最优决策平面 $(\boldsymbol{w},b)$ 使得支持向量样本之间的距离最大化[14-15]，从而有效地降低 SVM 的泛化误差。最优决策平面 $(\boldsymbol{w},b)$ 可通过学习支持向量样本间的软间隔来确定，即利用二次规划（Quadratic Programming，QP）求解以下凸优化问题。

$$\begin{aligned}&\min_{\boldsymbol{w},b,\varepsilon}\frac{1}{2}\boldsymbol{w}^{\mathrm{T}}\boldsymbol{w}+C\sum_{i=1}^{n}\varepsilon_i\\&\text{s.t. }\ y_i(\boldsymbol{w}^{\mathrm{T}}\boldsymbol{x}_i+b)\geqslant 1-\varepsilon_i,i=1,2,\cdots,n\\&\qquad\varepsilon_i\geqslant 0,i=1,2,\cdots,n\end{aligned} \tag{2}$$

其中，ε_i 为松弛变量，代表样本 $\boldsymbol{x}_i$ 背离最优决策平面 $(\boldsymbol{w},b)$ 的程度；n 为样本的总个数；C 为平衡系数，用于平衡 SVM 模型分类误差和最大间隔之间的关系。由此，求解间隔最大化问题便可转化为最小化 $\frac{1}{2}\boldsymbol{w}^{\mathrm{T}}\boldsymbol{w}$ 问题。图 1 给出了训练 SVM 模型过程中的决策平面、支持向量和间隔等概念。

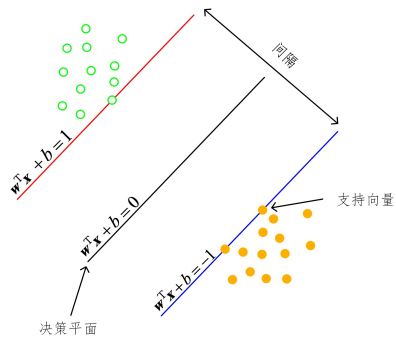

图 1　SVM 中的简单概念

Fig. 1　Simple concept of SVM

1.2　A-SVM 模型

学习对抗样本，利用对抗样本包含的信息训练分类模型，已被证明能够有效地提高分类模型的性能[8-11,16-18]。不同于文献[8-11]生成对抗样本的方法，即将微小且合理的扰动直接加入原有样本特征，本文首先从训练集中随机选取较小比例的训练样本，并对所选取的样本标签进行攻击，即用训练集包含的其他不同类型的标签替换所选取的样本标签。当解决二分类问题 $y\in\{-1,1\}$ 时，某一随机选取的样本 $\boldsymbol{x}_i$ 的标签为 $y_{x_i}=1$，那么该样本标签被攻击后将变为 $y_{x_i}=-1$。依此类推，当解决多分类问题 $y\in\{y_1,y_2,\cdots,y_L\}$ 时，某一随机选取的样本 $\boldsymbol{x}_i$ 的标签为 $y_{x_i}=y_l$，那么该样本标签被攻击后将变为 $y_{x_i}\in\{y_1,y_2\cdots,y_L\}-y_l$。其次，将包含对抗样本的训练集输入 SVM 进行参数学习，生成对抗支持向量机。求解生成

的 A-SVM 输出误差对输入所有训练样本的一阶梯度信息，并将该梯度信息嵌入到训练样本特征中，以此更新所有训练样本。最后，再次利用 SVM 学习更新后的所有训练样本，生成新的 A-SVM。可求解得到在生成 A-SVM 之前 SVM 输出误差对每个输入样本特征的一阶梯度：

$$\nabla(x_{i,j})=\frac{\partial E}{\partial x_{i,j}} \tag{3}$$

其中，E 代表 SVM 输出误差，$x_{i,j}$ 代表样本 $\boldsymbol{x}_i$ 的第 j 维特征，$1\leqslant j\leqslant d$。由此可求解得到 SVM 输出误差对每个输入样本的一阶梯度：

$$\nabla(\boldsymbol{x}_i)=\frac{\partial E}{\partial \boldsymbol{x}_i} \tag{4}$$

将式(4)中求解得到的一阶梯度信息嵌入到输入样本，可得更新后的样本 $\boldsymbol{x}_i'$：

$$\boldsymbol{x}_i'=\boldsymbol{x}_i+\lambda\nabla(\boldsymbol{x}_i) \tag{5}$$

其中，λ 为输入样本的特征学习效率[8]，其取值范围可为 $-1\leqslant\lambda\leqslant 1$。根据大量的实验结果，$\lambda$ 应取一较小值，较大的 λ 值将会严重破坏样本原有的特征空间结构。

为了更清晰地了解本文所提 A-SVM 模型，算法 1 详细描述了 A-SVM 模型的整个训练流程。

算法 1 A-SVM 模型训练流程

输入：数据集 $\mathbf{X}=\{\mathbf{x}_1,\mathbf{x}_2,\cdots,\mathbf{x}_N\}$，$\mathbf{x}_i=[x_1,x_2,\cdots,x_d]$，N 代表样本总数，d 代表特征维数，对应的标签集为 $Y=\{y_{\mathbf{x}_1},y_{\mathbf{x}_2},\cdots,y_{\mathbf{x}_N}\}$；样本特征学习效率为 λ

输出：A-SVM 模型参数

步骤 1 从输入的数据集 $\mathbf{X}$ 中随机抽取 80% 的样本作为训练集，并从训练集中随机抽取较小比例 a 的样本；

步骤 2 攻击从训练集中随机抽取的样本标签信息，使之成为对抗样本；

步骤 3 利用 SVM 算法训练包含对抗样本的训练集，生成 A-SVM 分类器；

步骤 4 利用式(4)求解 SVM 分类器输出对输入样本的一阶梯度；

步骤 5 通过式(5)将求解的一阶梯度信息嵌入到输入样本特征中，以更新输入样本。

1.3 原理分析

针对算法 1，假设步骤 1 中随机抽取形成的训练集在被攻击标签信息前的真实标签集记为 Y，执行步骤 2 后标签集记为 $\overline{Y}$，A-SVM 的输出结果记为 Y'，那么可得 A-SVM 输出误差为：

$$E=(Y'-\overline{Y})^2 \tag{6}$$

由步骤 1 可知，从训练集中随机选取较小比例 a 的样本，其标签信息被攻击，因此式(6)可进一步表达为：

$$E=[Y'-(1-a)Y]^2 \tag{7}$$

针对训练集中的某个样本 $\boldsymbol{x}_i$，可求得式(7)中 A-SVM 输出误差对于该样本特征的偏导：

$$\frac{\partial E}{\partial \boldsymbol{x}_i}=\frac{\partial E}{\partial Y'}\frac{\partial Y'}{\partial \boldsymbol{x}_i} \tag{8}$$

结合式(7)，式(8)可进一步表达为：

$$\frac{\partial E}{\partial \boldsymbol{x}_i}=2[Y'-(1-a)Y]\frac{\partial Y'}{\partial \boldsymbol{x}_i} \tag{9}$$

文献[8]中，DNN 将模型输出误差对输入的一阶梯度信息嵌入到原样本特征中以更新原样本，将更新后的样本输入到 DNN 并再次训练可优化隐藏层中的每个节点权重，且实验结果也证明了该对抗样本学习方式能够较好地提高 DNN 的泛化能力。基于大量的实验结果，并结合式(9)可知：一方面，被嵌入到输入样本特征中的一阶梯度信息 $\nabla(\boldsymbol{x}_i)$ 可提高 A-SVM 的分类性能；另一方面，所计算的 $\nabla(\boldsymbol{x}_i)$ 不仅与 A-SVM 的输出 Y' 相关，还与 A-SVM 的输入 Y 相关，即提供了一种输出与输入之间的正相关关系。式(9)从分类器输出误差的角度分析了对抗样本如何提高 A-SVM 的实际分类性能。

2 堆栈式支持向量机

为了获取最佳分类性能，本文将 A-SVM 作为子分类器，利用栈式结构原理级联一定数目的 A-SVM，生成堆栈式支持向量机。特别地，A-SVM 输出误差的梯度信息为关联前后两个子分类器提供了一种新的级联方式。

2.1 S-SVM 的栈式结构

根据栈式结构原理[12-13]，如图 2 所示，本文将第 1 节中训练包含对抗样本的训练集生成的 A-SVM 作为子分类器，S-SVM 由 K 个 A-SVM 级联而成。根据算法 1 中的步骤 4 和步骤 5，计算当前 A-SVM_k 输出误差对于输入样本特征的一阶梯度信息，并将该信息嵌入到当前输入样本特征中以更新样本。根据式(5)，将更新后的样本输入到下一个 A-SVM 中，即：

$$\boldsymbol{X}_{k+1}=\boldsymbol{X}_k+\boldsymbol{G}_k \tag{10}$$

根据式(4)，S-SVM 的每一个子分类器 A-SVM 输出误差的一阶梯度信息矩阵 $\boldsymbol{G}_k$ 可表示为：

$$\boldsymbol{G}_k=\lambda\nabla\frac{\partial E_k}{\partial \boldsymbol{X}_k} \tag{11}$$

在生成每个子分类器 A-SVM_k 前，都需要从当前输入的样本中随机选取较小比例 a 的样本并攻击其标签，如图 2 中“标签攻击”模块所示，其中，$\boldsymbol{X}$ 代表原始样本集，Y 代表原始样本集 $\boldsymbol{X}$ 的真实标签集，Y_K' 代表 S-SVM 的输出。由图 2 可知，S-SVM 中当前与下一个子分类器 A-SVM 主要通过当前 A-SVM 输出误差上的一阶梯度信息进行级联，与深度神经网络 DNN 相比，即通过 BP 算法在输出层学习输出误差来逐层优化隐藏层中每个单元的权值，基于对抗样本学习的一阶梯度信息为具有栈式结构的分类器提供了一种新的级联方式。另外，当把 S-SVM 中的每个子分类器 A-SVM 看作 DNN 中的每一个隐藏层时，所提 S-SVM 可看作深度学习模型。

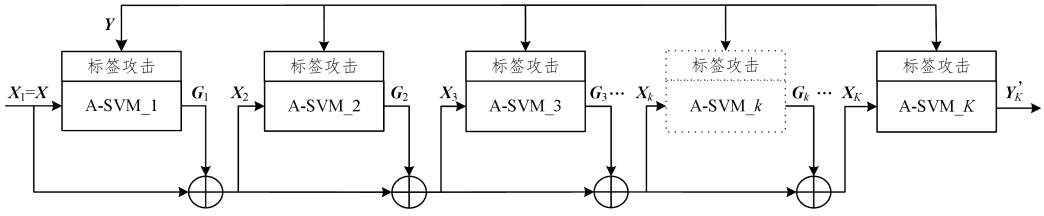

图 2 S-SVM 的栈式结构

Fig. 2 Stacked structure of S-SVM

2.2 S-SVM 算法及其复杂度分析

根据 2.1 节 S-SVM 的具体栈式结构，算法 2 给出 S-SVM 算法的具体描述。

算法 2 S-SVM 算法

输入：原始数据集 $\mathbf{X}=\{\mathbf{x}_1,\mathbf{x}_2,\cdots,\mathbf{x}_N\}$，$\mathbf{x}_i=[x_1,x_2,\cdots,x_d]$，N 代表样本总数，d 代表特征维数，对应的标签集为 $Y=\{y_{\mathbf{x}_1},y_{\mathbf{x}_2},\cdots,y_{\mathbf{x}_N}\}$；样本特征学习效率 λ；S-SVM 包含的子分类器数 K

输出：S-SVM 模型参数

步骤 1　设置 $\mathbf{X}_1=\mathbf{X}$，k=1；

步骤 2　将数据集 $\mathbf{X}_1$ 及对应的标签集 Y 作为 A-SVM_k 的输入；

步骤 3　调用算法 1 训练 A-SVM_k 模型并计算输出误差上的一阶梯度信息矩阵 $\mathbf{G}_k$；

步骤 4　对于输出集 Y_k'，如果 S-SVM 已获得最佳分类结果或已达到最大迭代次数 K，算法停止迭代，否则 $\mathbf{X}_{k+1}=\mathbf{X}_k+\lambda\mathbf{G}_k$；

步骤 5　k=k+1，返回步骤 2。

根据算法 2 及栈式结构原理[12-13]，S-SVM 中的每个子分类器输出为根据式(4)和式(11)计算的输出误差上的一阶梯度信息矩阵，S-SVM 栈式结构中的中间变量具有透明性，且每个子分类器 A-SVM 基于基本的支持向量机经学习对抗样本训练而成。因此，所提 S-SVM 易于理解和实现。

根据算法 2，S-SVM 由多个子分类器 A-SVM 根据栈式结构原理级联而成，因此在分析算法 2 复杂度时应首先分析 A-SVM 模型的复杂度。针对算法 1，由于步骤 1、步骤 2 中选取较小比例 a 的样本并攻击其标签使之成为对抗样本，因此可以忽略步骤 1、步骤 2 的时间复杂度；步骤 3 采用基本的 SVM 训练包含 N 个样本的数据集生成 A-SVM，根据文献[19]，步骤 3 所需的最大时间复杂度为 $O(dN^2)$；由于需要计算 A-SVM 输出误差对每一个样本的一阶梯度信息，且样本特征维度为 d，因此步骤 4 的时间复杂度为 $O(dN)$；步骤 5 将计算的一阶梯度信息嵌入每一个样本特征中，与之对应的时间复杂度为 $O(dN)$。综上所述，训练 A-SVM 模型所需的时间复杂度为 $O(dN^2+dN+dN)$。针对算法 2，因 S-SVM 由 K 个子分类器 A-SVM 级联而成，且算法 2 所需的时间复杂度主要集中于步骤 3，根据上述对算法 1 的时间复杂度分析，算法 2 的时间复杂度为 $O[K(dN^2+dN+dN)]$。取最高阶时，算法 2 的时间复杂度可进一步演变为 $O(KdN^2)$，因此本文所提 S-SVM 可有效解决包含样本数适中的数据集分类问题。

3 实验结果与分析

3.1 实验配置

本文从实验对比的角度来分析所提分类模型 S-SVM 的实际分类性能。在构建对抗支持向量机 A-SVM 的过程中，以线性型 SVM(Linear-SVM)和高斯型 SVM(Gaussian-SVM)为基础，由此形成的堆栈式支持向量机分别为 L-S-SVM 和 G-S-SVM。对比算法除了选择 Linear-SVM、Gaussian-SVM、随机森林及决策树外，由于 S-SVM 的栈式结构决定其具有深度学习的性质，因此本文还选择深度学习方法，即基于 wake-sleep 算法与受限制玻尔兹曼机(Restricted Boltzmann Machine，RBM)的深层信念网(Deep Belief Networks，DBN)。对比算法的参数设置分别如下：SVM 来源于 LIBSVM[20]，其中，对于 Linear-SVM，其参数惩罚系数 c 的搜索范围为 $\{10^{-5},10^{-4},\cdots,10^4,10^5\}$；对于 Gaussian-SVM，其参数惩罚系数 c 的搜索范围为 $\{10^{-5},10^{-4},\cdots,10^4,10^5\}$，高斯核函数中核宽度 δ 的搜索范围为 $\{10^{-5},10^{-4},\cdots,10^4,10^5\}$；对于随机森林(Random Forest，RF)[21]，其叶子节点数 T 设定为文献[22]中的推荐值，即 $T=2^7$，此时的 RF 能够较好地平衡算法的分类精度与算法时间复杂度；决策树(C4.5)算法[23]及朴素贝叶斯(Naïve Bayes，NB)[24]的参数设置均采用默认值；深层信念网[25-27]的层数为 3，每一层包含的隐藏节点数的搜索范围取决于数据集包含的样本数，其他参数分别为 $maxepoch=50$，$epsilon=0.1$，$epsilonvb=0.1$，$epsilonhb=0.1$，$initialmomentum=0.5$，$finalmomentum=0.9$，$weightcost=0.0002$。所提 S-SVM 涉及模型的参数如下：惩罚系数 c，或惩罚系数 c 与高斯核宽度 σ，子分类器总数 K(即 S-SVM 包含的层数)，式(7)中的比例 a 以及式(5)中的输入样本更新学习效率 λ。其中，惩罚系数 c 与高斯核宽度的搜索范围与 Gaussian-SVM 相同，参数 K、参数 a 以及参数 λ 分别设置为 5，0.05(5%)，0.001，其具体设置规则将在以下实验结果中介绍。实验中选取的真实数据集均来源于 UCI machine learning repository[28] 和 the KEEL-dataset repository[29]。表 1 详细列出了所选真实数据集的详细配置情况。数据集的样本数范围为 210～12960，样本特征维数的范围为 3～21，样本类别数的范围为 2～54，数据集配置一方面符合验证所提 S-SVM 有效性的需求，另一方面也符合算法 2 的复杂度分析，即 S-SVM 适合包含样本数适中的数据集的分类问题。实验中所有算法的最优参数均经网格搜索结合十折交叉验证的方法获得[5,30]。所有实验结果均在 matlab 2015b 软件平台上运行程序获得，电脑配置为 64 位的 Windows 10 操作系统，CPU 为 Inter(R) Core(TM) i7-4790，CPU 主频为 3.6 GHz，内存大小为 8 GB。

表 1　真实数据集的详细配置

Table 1　Detailed configurations of real-world datasets

数据集	样本数	特征维数	类别数
Haberman(HAB)	306	3	2
Liver(LIV)	345	6	2
Phoneme(PHO)	5404	5	2
Wisconsin(WIS)	683	9	2
Monks-1(MON)	432	6	2
Seeds(SEE)	210	7	3
Thyroid(THY)	7200	21	3
Vehicle(VEH)	846	18	4
Glass(GLA)	214	9	6
WhiteWineQuality(WHI)	4898	11	7
Yeast(YEA)	1484	8	10
Nursery(NUR)	12960	8	54

3.2 实验结果集分析

表 2 详细列出了所有对比算法在所有真实数据集上取得的实际分类性能。其中，“*Acc*”代表运行程序 10 次后取得的平均分类精度，“*Opt*”代表算法经网格搜索结合十折交叉验证方法获得的最优参数，“—”代表算法参数采用默认配置，最高的分类精度采用加粗标识。由表 2 可得出以下结果：1) L-S-SVM 与 G-S-SVM 在 Haberman，Liver，Seeds，Thyroid，Vehicle 以及 Yeast 6 个数据集上取得最高的分类精度，尤其在数据集 Haberman，Liver，Seeds，Vehicle 以及 Yeast 上的分类结果明显优于其他对比算法；2) 作为性能优异的算法，RF 与 DBN 在 Phoneme，Wisconsin 数据集上表现出了很好的分类

性能;3)与 Linear-SVM 和 Gaussian-SVM 相比,L-S-SVM 和 G-S-SVM 的分类性能均得到了提高,这有效说明了在栈式结构原理框架下,基于对抗样本学习的一阶梯度信息确实能够提高分类模型的性能;4)在栈式结构原理框架下,虽然设置 L-S-SVM 和 G-S-SVM 的子分类器总数 $K=5$,但所提 S-SVM 基本在第 2、第 3 或第 4 层取得最佳分类精度,即使增加层数(即增加子分类器个数),也不会提高 S-SVM 的分类性能;5)对于大部分真实数据集,如 LIV,WIS,SEE 等,由于考虑了对抗样本学习和栈式结构,L-S-SVM 与 G-S-SVM 的分类性能相当。

表 2 所有算法的实际分类性能

Table 2 Actual classification performance of all comparative algorithms on all real-world datasets

数据集	指标	Linear-SVM (c)	Gaussian-SVM (c,δ)	RF	C4.5	NB	DBN	L-S-SVM (c,K)	G-S-SVM (c,δ,K)
HAB	*Acc*	71.80±4.20	74.82±3.09	69.84±4.76	70.16±6.91	72.30±6.79	74.16±4.60	73.20±3.38	**76.56** ± 2.67
	Opt	(10^4)	(10^4,10)	—	—	—	—	(10^4,4)	(10^5,10^5,3)
LIV	*Acc*	73.04±4.36	70.87±3.75	69.71±6.03	64.93±5.30	53.77±7.68	54.72±5.91	**76.54** ± 3.91	75.83±3.24
	Opt	(10^5)	(10^3,10^2)	—	—	—	—	(10^3,3)	(10^4,10^4,3)
PHO	*Acc*	84.24±0.82	84.93±0.61	**91.17** ± 0.92	85.98±1.17	76.26±1.10	74.27±7.13	87.46±1.34	86.30±0.72
	Opt	(10^3)	(10^5,10^5)	—	—	—	—	(10^5,5)	(10^5,10^3,4)
WIS	*Acc*	94.45±1.82	92.92±1.96	97.35±1.15	94.73±1.75	96.32±1.19	**97.54** ± 1.19	96.02±1.29	95.19±2.48
	Opt	(10^5)	(10^5,10^5)	—	—	—	—	(10^4,3)	(10^3,10^4,4)
MON	*Acc*	23.55±3.64	42.94±2.83	11.01±2.27	15.92±2.64	42.03±1.72	**49.52** ± 4.21	30.54±4.83	45.83±2.85
	Opt	(10^5)	(10^{-2},10^{-1})	—	—	—	—	(10^4,2)	(10^3,10^5,2)
SEE	*Acc*	94.01±2.35	93.52±3.48	93.20±3.62	94.33±3.18	91.67±2.43	89.88±4.82	**96.57** ± 1.92	96.18±3.25
	Opt	(10^4)	(10^3,10^4)	—	—	—	—	(10^5,4)	(10^5,10^5,3)
THY	*Acc*	98.79±0.36	98.80±0.19	99.67±0.19	99.58±0.13	95.78±0.53	93.56±1.11	**99.72** ± 0.44	99.33±0.75
	Opt	(10^5)	(10^5,10^5)	—	—	—	—	(10^5,3)	(10^5,10^5,2)
VEH	*Acc*	84.76±2.37	85.00±2.72	71.94±2.78	71.25±2.93	44.53±3.69	49.18±7.28	85.60±3.01	**86.03** ± 1.85
	Opt	(10^5)	(10^4,10^3)	—	—	—	—	(10^5,2)	(10^3,10^4,2)
GLA	*Acc*	73.95±7.84	69.76±8.44	**78.57** ± 5.93	68.10±7.24	46.73±7.54	68.02±9.29	75.26±5.98	73.64±6.77
	Opt	(10^5)	(10^3,10^2)	—	—	—	—	(10^4,3)	(10^4,10^5,4)
WHI	*Acc*	56.14±1.06	55.09±0.64	**68.12** ± 1.25	56.42±1.43	43.65±1.44	53.08±5.07	56.82±0.93	56.78±0.42
	Opt	(10^5)	(10^5,10^5)	—	—	—	—	(10^5,2)	(10^5,10^5,2)
YEA	*Acc*	61.55±2.16	57.62±2.79	61.81±2.48	53.15±2.24	57.82±2.54	58.49±4.80	**63.04** ± 1.85	62.12±2.29
	Opt	(10^5)	(10^3,10^4)	—	—	—	—	(10^4,3)	(10^4,10^5,2)
NUR	*Acc*	96.74±0.19	96.94±0.26	**99.54** ± 0.14	98.17±0.25	74.51±0.16	94.87±0.28	98.61±0.29	97.92±0.23
	Opt	(10^2)	(10^2,10^5)	—	—	—	—	(10^5,3)	(10^5,10^5,4)

以 G-S-SVM 为例,表 3 给出了当式(7)中的参数 a 取不同比例值时,G-S-SVM 在 6 个真实数据集上的实际分类性能,此时固定式(5)中的 λ 值为 0.001。由表 3 可知,绝大多数情况下取 $a=0.05$(5%)时,G-S-SVM 能够取得最佳分类结果。即从训练集中学习更多的对抗样本信息,并不能提高 S-SVM 的分类性能,只会降低 S-SVM 的整体性能,这也符合分类模型学习更多错误样本信息导致更多错误分类的结果。另外,增加比例 a 并没有大幅降低 S-SVM 的整体分类性能,大量的实验结果表明这种现象得益于在 S-SVM 中运用了栈式结构原理。在随机抽取原始样本生成对抗样本的过程中,本文推荐抽取样本的最佳比例为 $a=0.05$(5%)。

表 3 参数 a 取不同值时的分类结果($\lambda=0.001$)

Table 3 Classification results with different values of a($\lambda=0.001$)

数据集	指标	0.05(5%)	0.10(10%)	0.15(15%)
HAB	*Acc*	76.56±2.67	75.31±2.36	73.18±2.94
	Opt	(10^5,10^5,3)	(10^4,10^5,4)	(10^4,10^4,4)
LIV	*Acc*	75.83±3.24	76.32±3.80	73.17±3.13
	Opt	(10^4,10^4,3)	(10^3,10^5,3)	(10^4,10^3,4)
WIS	*Acc*	95.19±2.48	94.16±2.29	92.40±3.09
	Opt	(10^3,10^4,4)	(10^3,10^4,3)	(10^4,10^4,4)
THY	*Acc*	99.33±0.75	98.45±0.57	96.72±0.16
	Opt	(10^5,10^5,2)	(10^5,10^5,3)	(10^5,10^5,2)
VEH	*Acc*	86.03±1.85	85.33±2.31	84.20±2.14
	Opt	(10^3,10^4,2)	(10^4,10^4,3)	(10^4,10^5,4)
YEA	*Acc*	62.12±2.29	60.63±2.24	59.18±2.86
	Opt	(10^4,10^5,2)	(10^5,10^5,4)	(10^5,10^5,3)

以 G-S-SVM 为例,表 4 给出了当式(5)中的参数 λ 取不同值时,G-S-SVM 在 6 个真实数据集上的实际分类性能,此时固定式(7)中 a 的值为 0.05。

表 4 参数 λ 取不同值时的分类结果($a=0.05$)

Table 4 Classification results with different values of λ($a=0.05$)

数据集	指标	0.001	0.01	0.1
HAB	*Acc*	76.56±2.67	73.64±3.28	72.70±2.95
	Opt	(10^5,10^5,3)	(10^5,10^4,4)	(10^4,10^4,4)
LIV	*Acc*	75.83±3.24	71.48±3.48	69.53±2.31
	Opt	(10^4,10^4,3)	(10^3,10^4,4)	(10^4,10^4,5)
WIS	*Acc*	95.19±2.48	92.59±2.93	90.56±2.40
	Opt	(10^3,10^4,4)	(10^4,10^4,4)	(10^4,10^5,4)
THY	*Acc*	99.33±0.75	97.37±0.31	93.72±0.47
	Opt	(10^5,10^5,2)	(10^5,10^5,3)	(10^5,10^5,4)
VEH	*Acc*	86.03±1.85	82.66±2.23	79.63±3.63
	Opt	(10^3,10^4,2)	(10^3,10^3,3)	(10^3,10^4,4)
YEA	*Acc*	62.12±2.29	58.72±2.76	55.29±2.64
	Opt	(10^4,10^5,2)	(10^5,10^5,2)	(10^4,10^5,4)

从表 4 可得出以下几点结论:1)输入样本更新学习效率 λ 的最佳取值为 0.001,此时 G-S-SVM 能够取得最佳的分类结果;2)当 λ 取更大值时,逐层向原始样本特征中嵌入的子分类器 A-SVM 输出误差上的一阶梯度信息已破坏了原始样本的特征空间,强大的栈式结构原理已不能适用于逐层提高 S-SVM 的子分类器的性能,增加 S-SVM 层数只会严重降低 S-SVM 的实际分类性能;3)如在数据集 HAB,LIV,THY,VEH 以及 YEA 上,随着 λ 取值的进一步增大,S-SVM 需要

更多的子分类器（增加 S-SVM 层数）来提高其整体分类性能，表明了将栈式结构原理运用于具有栈式结构的分类模型时确实能够有效提高分类模型的性能。

3.3　统计分析

本节引用文献[31]所提出的在多个数据集上的多分类器性能对比方法，从统计分析的角度进一步分析所提 S-SVM 与其他对比算法的区别。该统计分析方法主要包含 3 个步骤。

(1)根据分类精度为每一个对比算法在每一个数据集上进行排名（*rank* 值）。以数据集 Haberman 为例，由于 G-S-SVM 和 Gaussian- SVM 取得的分类精度分别为 76.56%和 74.82%，因此其 *rank* 值分别为 1 和 2，依次类推，赋予其他对比算法在每一个数据集上的 *rank* 值。由此可得 *rank* 值表，如表 5 所列，其中“*Ave*”代表每个算法在所有数据集上的平均 *rank* 值，假设 N_c 和 N_d 分别代表对比算法和数据集的数量，则计算公式如下：

$$rank_i=\frac{1}{N_d}\sum_{j=1}^{N_d}rank_i^j \tag{12}$$

其中，$rank_i^j$ 代表第 i 个算法在第 j 个数据集上的 *rank* 值。

(2)执行 Friedman 测试以确定是否能够否定空假设，即假设所有对比算法在所有数据集上的分类性能一样。根据 F 分布表[31]，选择置信度 $\alpha=0.05$，根据 $F((N_c-1),(N_c-1)(N_d-1))$计算关键值为 $F(7,77)\approx2.13$，只要执行 Friedman 测试得到的 F_F 大于该关键值，空假设即被否定。其中，F_F 的计算公式如下：

$$F_F=\frac{(N_d-1)\chi_F^2}{N_d(N_c-1)-\chi_F^2} \tag{13}$$

其中，Friedman 统计量 χ_F^2 的计算式如下：

$$\chi_F^2=\frac{12N_d}{N_c(N_c+1)}\left[\sum_{i=1}^{N_c}rank_i^2-\frac{N_c(N_c+1)^2}{4}\right] \tag{14}$$

由式(13)与式(14)可得表 2 上的 $F_F\approx6.63$。由于 $F_F>2.13$，因此空假设被否定，即所有对比算法在所有数据集上的分类性能并不相同。

(3)随着步骤(2)中的空假设被否定，可进一步执行 Bonferroni-dunn 测试[32-33]来探索 S-SVM 与其他对比算法是否存在本质的区别。根据表 2，执行 Bonferroni-dunn 测试后可根据下式计算 *CD*(Critical Difference)值：

$$CD=q_\alpha\sqrt{\frac{N_c(N_c+1)}{6N_d}} \tag{15}$$

由于置信度 $\alpha=0.05$，根据文献[31]可查得 $q_{0.05}=2.690$。因此，根据表 2 和 Bonferroni-dunn 测试可计算 $CD=2.69$。任何两种对比算法的平均 *rank* 值差大于该 *CD* 值时，表明这两种算法之间存在本质的区别。

根据表 5 可得出以下几点结论：1)L-S-SVM 及 G-S-SVM 相比其他对比算法具有最小的平均 *rank* 值，表明所提 S-SVM 在所选真实数据集上的分类性能优于对比算法或者至少与对比算法的分类性能相当；2)L-S-SVM 与 Linear-SVM，C4.5，NB 以及 DBN 之间和 G-S-SVM 与 NB 以及 DBN 之间的平均 *rank* 值的差均大于 $CD=2.69$，因此，所提 S-SVM 与 Linear-SVM，C4.5，NB 以及 DBN 之间存在本质的不同；3)虽然 L-S-SVM 及 G-S-SVM 在大规模数据集 NUR 上的分类效果不是最好的，但由于 L-S-SVM 及 G-S-SVM 与 RF 之间的平均 *rank* 值的差分别为 1.33 和 0.83，平均 *rank* 值差均小于 $CD=2.69$，结合 1)可知 L-S-SVM 及 G-S-SVM 在大规模数据集上至少能够取得与其他对比算法相当的分类性能；4)L-S-SVM 与 G-S-SVM 之间的平均 *rank* 值的差为 0.50，从而表明 L-S-SVM 与 G-S-SVM 在所有真实数据集上有着相当的分类性能，这与 3.2 节实验分析中的第 5)点一致；5)由 1)和 2)可知，在栈式结构原理框架下，基于对抗样本学习的一阶梯度信息逐层更新输入样本的策略能够有效地提高分类器的实际分类性能。

表 5　依据表 2 的分类结果得出的 *rank* 值

Table 5　Value of *rank* based on classification results in Table 2

数据集	Linear-SVM	Gaussian-SVM	RF	C4.5	NB	DBN	L-S-SVM	G-S-SVM
HAB	6	2	8	7	5	3	4	1
LIV	3	4	5	6	8	7	1	2
PHO	6	5	1	4	7	8	2	3
WIS	7	8	2	6	3	1	4	5
MON	6	3	8	7	4	1	5	2
SEE	4	5	6	3	7	8	1	2
THY	6	5	2	3	7	8	1	4
VEH	4	3	5	6	8	7	2	1
GLA	3	5	1	6	8	7	2	4
WHI	5	6	1	4	8	7	2	3
YEA	4	7	3	8	6	5	1	2
NUR	6	5	1	3	8	7	2	4
Ave	5.00	4.83	3.58	5.25	6.58	5.75	2.25	2.75

结束语　针对真实数据集中存在对抗样本的问题，本文从攻击标签信息的角度人为生成对抗样本，将合理利用对抗样本演算出的一阶梯度信息嵌入到原始样本特征中以更新原始样本，并利用栈式结构原理逐层更新并训练更新后的原始样本生成具有栈式结构的分类模型。在真实数据集上的实验结果以及统计分析结果表明：本文利用对抗样本学习和栈式结构原理生成的 S-SVM 具有优异的实际分类性能。对于算法 1 中的步骤 1，由于从训练样本中随机选取较小比例样本以生成对抗样本的过程带有随机性，因此，接下来的工作将重点研究如何合理地从数据集中选取样本并生成对抗样本[9]。另外，根据式(5)，本文将子分类器 A-SVM 输出误差上的一阶梯度信息直接嵌入到原有样本特征中以更新样本，如何优化样本特征更新的过程也是未来工作的重点研究内容之一[8]。

参 考 文 献

[1] WAN Y, LI H H, WU K F, et al. Fusion with layered features of LBP and HOG for face recognition[J]. Journal of Computer-Aided Design & Computer Graphics, 2015, 27(4): 640-650.

[2] XI X F, ZHOU G D. A survey on deep learning for natural language processing[J]. ACTA Automatica Sinica, 2016, 42(10): 1445-1465.

[3] WANG D, MIAO D Q, WANG R Z. A new method of EEG classification with feature extraction based on wavelet packet decomposition[J]. ACTA Electronica Sinica, 2013, 41(1): 193-198.

[4] ZENG Z, WU C G, TANG Q H, et al. Classification of commodi-

ty image based on multi-feature fusion and depth learning[J]. Computer Engineering and Design, 2017, 38(11): 3093-3098.
[5] ZHOU T, CHUNG F-L, WANG S T. Deep TSK fuzzy classifier with stacked generalization and triplely concise interpretability guarantee for large data[J]. IEEE Transactions on Fuzzy Systems, 2017, 25(5): 1207-1221.
[6] DONG A, CHUNG F L, DENG Z, et al. Semi-supervised SVM with extended hidden features[J]. IEEE Transactions on Cybernetics, 2016, 46(12): 2924-2937.
[7] HE X, ZHANG C, ZHANG L, et al. A optimal projection for image representation[J]. IEEE Transactions on Pattern Analysis and Machine Intelligence, 2016, 38(5): 1009-1015.
[8] MOSCA A, MAGOULAS G D. Hardening against adversarial examples with the smooth gradient method[J]. Soft Computing, 2018, 22(10): 3203-3213.
[9] KATHRIN G, NICOLAS P, PRAVEEN M, et al. Adversarial perturbations against deep neural networks for malware classification[J]. arXiv: 1511.04508.
[10] MA Y K, WU L F, JIAN M, et al. Approach to generate adversarial examples for face-spoofing detection[J]. Journal of Software, 2019, 30(2): 279-290.
[11] GU S X, RIGAZIO L. Towards deep neural network architectures robust to adversarial examples[C]// International Conference on Learning Representation(ICLR). Banff, Canada, 2014.
[12] ZHOU T, CHUNG F L, WANG S T. Deep TSK fuzzy classifier with stacked generalization and triplely concise interpretability guarantee for large data[J]. IEEE Transactions on Fuzzy Systems, 2017, 25(5): 1207-1221.
[13] ZHANG Y P, ISHIBUCHI H, WANG S T. Deep Takagi-Sugeno-Kang fuzzy classifier with shared linguistic fuzzy rules [J]. IEEE Transactions on Fuzzy Systems, 2018, 26(3): 1535-1549.
[14] VAPNIK V N. Statistical learning theory [M]. New York: Wiley, 1998.
[15] XU Y T. Maximum margin of twin spheres support vector machine for imbalanced data classification[J]. IEEE Transactions on Cybernetics, 2017, 47(6): 1540-1550.
[16] WANG Z R, WANG J, WANG Y R. An intelligent diagnosis scheme based on generative adversarial learning deep neural networks and its application to planetary gearbox fault pattern recognition[J]. Neurocomputing, 2018, 310(8): 213-222.
[17] OZSOY M, KHASAWNEH K N, DONOVICK C, et al. Hardware-based malware detection using low-level architectural features[J]. IEEE Transactions on Computers, 2016, 65(11): 3332-3344.
[18] TANG J J, LEU G, ABBASS H A. Networking the boids is more robust against adversarial learning[J]. IEEE Transactions on Network Science and Engineering, 2018, 5(2): 141-155.
[19] BURGES C J C. A tutorial on support vector machines for pattern recognition[J]. Data Mining and Knowledge Discovery, 1998, 2(2): 121-167.
[20] CHANG C C, LIN C J. LIBSVM: A library for support vector machines[J]. ACM Transactions on Intelligent Systems and Technology, 2011, 2(3): 27: 1-27: 27.
[21] WANG Y S, XIA S T, TANG Q T, et al. A novel consistent random forest framework: Bernoulli random forests[J]. IEEE Transactions on Neural Networks and Learning Systems, 2018, 29(8): 3510-3523.
[22] OSHIRO T M, PEREZ P S, BARANAUSKAS J A. How many trees in a random forest? [C]// International Conference on Machine Learning and Data Mining in Pattern Recognition. 2012.
[23] QUINLAN J R. Induction of Decision Trees [J]. Machine Learning, 1986, 1(1): 81-106.
[24] RUSSEL S, NORVIG P. Artificial intelligence: A modern approach (2nd ed.)[M]. Prentice Hall, 2003: 597.
[25] HINTON G E, OSINDERO S, TEH Y W. A faster learning algorithm for deep belief nets[J]. Neural Computation, 2006, 1(7): 1527-1544.
[26] SON N T, ARTUR S D, AVILA G. Deep logic networks: Inserting and extracting knowledge from deep belief networks[J]. IEEE Transactions on Neural Networks and Learning Systems, 2018, 29(2): 246-258.
[27] CHONG Z, PIN L, QIN A K, et al. Multiobjective deep belief networks ensemble for remaining useful life estimation in prognostics[J]. IEEE Transactions on Neural Networks and Learning Systems, 2017, 28(10): 2306-2318.
[28] FRANK A, ASUNCION A. (2010) UCI Machine Learning Repository [OL]. http://archive.ics.uci.edu/ml.
[29] ALCALÁ-FDEZ J, FERNÁNDEZ A, LUENGO J, et al. KEEL data-mining software tool: Data set repository, integration of algorithms and experimental analysis framework[J]. Journal of Multiple-Valued Logic and Soft Computing, 2011, 17(2/3): 255-287.
[30] ITO K, NAKANO R. Optimizing support vector regression hyper-parameters based on cross-validation [C]// International Joint Conference on Neural Networks(IJCNN). Istanbul, Turkey, 2003: 2077-2082.
[31] DEMSAR J. Statistical comparisons of classifiers over multiple data sets[J]. Journal of Machine Learning Research, 2006, 7: 1-30.
[32] ZAR J H. Biostatistical Analysis(4th ed)[M]. Prentice Hall, Englewood Clifs, New Jersey, 1998.
[33] SHESKIN D J. Handbook of parametric and nonparametric statistical procedures[M]. Chapman and Hall/CRC, 2000.

JIN Yao, born in 1971, master, associate professor. His main research interests include the computer application technology, and library and information science.

GU Su-hang, born in 1989, doctoral student. His main research interests include artificial intelligence and machine learning.