

基于零阶减小方差方法的鲁棒支持向量机

鲁淑霞^{1,2} 蔡莲香¹ 张罗幻¹

(河北大学数学与信息科学学院 河北 保定 071002)¹

(河北省机器学习与计算智能重点实验室 河北 保定 071002)²

摘 要 采用传统的支持向量机方法对含有噪声的数据进行分类时会产生较大的损失,使得分类超平面严重偏离最优超平面,从而导致分类性能较差。为了解决此问题,文中提出了一种鲁棒的支持向量机(Robust Support Vector Machine, RSVM)方法,该方法给出了一种正弦平方形式的损失函数,根据正弦函数的特点,即使对于噪声数据,其损失函数的值也会被限制在 $[0,1]$ 区间,从而提高了支持向量机的抗噪性。另外,在求解支持向量机时,传统的随机梯度下降方法在每次迭代中利用单个样本梯度近似代替全梯度,这样必然会产生方差,而随着迭代次数的增加,方差也不断累积,从而严重影响算法的分类性能。为了减小方差的影响,引入零阶减小方差的随机梯度下降(Zeroth Order-Stochastic Variance Reduced Gradient, ZO-SVRG)算法。该算法使用坐标梯度估计方法近似代替梯度,通过在每轮迭代中引入梯度修正项来减小方差的影响;同时,采取加权平均的输出形式进行内外循环的输出,加快了优化问题的收敛速度。实验结果表明,提出的基于零阶减小方差方法的鲁棒支持向量机算法对噪声数据具有更好的鲁棒性,且有效降低了方差的影响;为了进一步提高算法的性能,对实验中主要参数 λ, k 对算法精度的影响进行了分析。对于线性和非线性两种情况,当其参数对 (λ, k) 分别满足 $(\lambda=1, k=5)$ 和 $(\lambda=10, k=3)$ 时,可以达到各自的最高精度。

关键词 方差约简, 零阶优化, 支持向量机, 损失函数, 噪声

中图分类号 TP181 文献标识码 A DOI 10.11896/jsjcx.181001840

Robust SVM Based on Zeroth Order Variance Reduction

LU Shu-xia^{1,2} CAI Lian-xiang¹ ZHANG Luo-huan¹

(College of Mathematics and Information Science, Hebei University, Baoding, Hebei 071002, China)¹

(Hebei Province Key Laboratory of Machine Learning and Computational Intelligence, Baoding, Hebei 071002, China)²

Abstract Great losses will be produced when traditional SVM methods are used to deal with the classification problem with noisy data, which makes the classification hyperplane seriously deviates from the optimal hyperplane, resulting in poor classification performance. In order to solve this problem, this paper proposed a robust support vector machine (RSVM) and gave a loss function in sinusoidal square form. According to the characteristics of sinusoidal function, the value of loss function is limited to the range of $[0,1]$, even for noise data, which improves the anti-noise ability of SVM. In addition, when the traditional stochastic gradient descent method is used to solve the SVM, a single sample gradient is used to approximately replace the full gradient in each iteration, which will inevitably produce variance. As the number of iterations increases, the variance also accumulates, which seriously affects the classification performance of the algorithm. In order to reduce the influence of variance, this paper introduced a zeroth order-stochastic variance reduced gradient (ZO-SVRG) algorithm. This algorithm uses coordinate gradient estimation method to replace gradient approximately, and reduces the influence of variance by introducing the gradient correction term in each iteration. Besides, in the output of the internal and external loop, the weighted average output form is adopted, and then the convergence speed of the optimization problem is accelerated. The experimental results show that the robust support vector machine based on zeroth-order variance reduction algorithm has better robustness to noise data and effectively reduces the influence of variance. In order to further improve the performance of the algorithm, the influence of the main parameters λ and k on the accuracy of algorithm were analyzed. For both linear and nonlinear cases, when its parameter pairs (λ, k) are satisfied $(\lambda=1, k=5)$ and $(\lambda=10, k=3)$, respectively, the highest accuracy of each can be achieved.

Keywords Variance reduction, Zeroth order optimization, Support vector machine, Loss function, Noise

1 引言

在实际应用中,人为或其他因素会给数据带来误差或错误,在机器学习领域中通常称该现象为数据存在噪声^[1]。经典的支持向量机的表达形式为“正则化项+损失函数”,正则化项通常指最大间隔项,损失函数一般为铰链损失。支持向量机往往对噪声数据不具有很好的鲁棒性,原因在于铰链损失函数的无界性,对于噪声或异常点,该函数对应的值很大,从而带来了较大的损失,使 SVM 的决策超平面严重偏离最优决策面,导致分类性能较差。为了提高支持向量机对噪声数据的鲁棒性,许多学者提出了改进 SVM 的方法。Liu 等^[2]引入了 L_q 范数惩罚项的思想,基于范数 L_q 可以对数据进行自动的数值选择,实现了自适应惩罚的效果。相比一般的支持向量机,该方法有效地提高了 SVM 的抗噪性。Lin 等提出了模糊支持向量机^[3] (Fuzzy Support Vector Machine, FS-VM),该算法根据训练样本对分类的贡献程度,对其赋予不同的隶属度,对于噪声或野点则给出较小的隶属度。隶属度的计算是基于样本和类中心之间的距离实现的。但这种计算方式存在一个弊端,即没有考虑样本之间的联系,这样对于两个距离比较近的样本,有可能无法准确识别出哪一个有效样本。Wu 等提出加权的支持向量机^[4] (Weighted Support Vector Machine, WSVM),这种算法与传统的支持向量机的区别在于,该算法对不同的训练样本赋予不同的权重,对训练分类器比较重要的样本赋予较大的权重,而对重要程度低和含噪声的样本赋予较小的权重,使用加权的方式可以有效地降低噪声对学习过程的影响。此外,采用不同的损失函数也会提高 SVM 的抗噪性。Sun 等^[5]将铰链损失改为 ϵ -不灵敏损失函数,利用光滑方法去掉了原始优化问题的约束条件,实验显示这种方式不仅提高了分类器的鲁棒性,还提高了分类器的学习效率。Huang 等^[6]提出使用斜坡损失代替传统支持向量机中的铰链损失。斜坡损失的形式为 $l_{\text{ramp}}(z) = \max(1-z) - \max(0, r-z)$,其中 r 为一个小于 0 的常数,这样即使对于噪声点较大的 z 值,也会使函数值最终限制在 $[0, 1]$ 区间,降低了噪声的影响。Xu 等^[7]提出了一种单调有界的非凸损失函数,并将其命名为 rhinge 损失。该损失函数以铰链损失为上界,随着给定的 η 值的增大,损失函数的值单调减小。对于噪声点,若给出适当的 η 值, rhinge 损失的函数值将远小于铰链损失的函数值,弥补了传统支持向量机抗噪性差的缺点。

如何对建立的分类模型进行有效的求解是目前的一个主要研究课题。Pegasos^[8]算法是由 Shalev-Shwartz 等提出的,该算法的原理不同于一般的把原始问题转换成对偶问题的求解方式,Pegasos 使用随机梯度下降的方法对支持向量机的原始问题进行求解,提高了 SVM 的收敛速度。但该算法的不足之处在于存在方差,在每次迭代中该算法随机选择一个样本进行梯度估计,并用该样本的梯度代替全梯度,随着迭代的进行方差随之累积,最后则会产生一个较大的方差。为了降低方差的影响,Zhang 提出了减小方差的随机梯度下降算法^[9] (Stochastic Variance Reduced Gradient, SVRG),该算法在每次的内层循环中使用单个样本梯度与全梯度之间的差值,使方差有一个可以不断减少的上界,减弱了方差对算法性

能的影响。同时该算法分为内外两层循环,仅在外层循环进行全梯度估计,在内层循环只进行单个样本的梯度估计,由于外层的轮数很少,因此只需要计算几次全梯度就可以实现收敛,提高了学习速度。但该算法在实际求解中仍存在一个问题:对于复杂的学习模型,很难甚至无法求出梯度的显式表达式,若依然采用求梯度的方式,显然会增加算法的复杂度。于是, Nesterov^[10]提出了无梯度优化的方法,也可称其为零阶优化方法。零阶优化方法通过随机选取样本邻域中的一个方向,利用差分的思想求解目标函数值,并用目标函数值近似代替梯度,有效地解决了上述问题。随后, Liu 把零阶优化方法和减小方差策略相结合,提出了零阶减小方差的随机梯度下降方法^[11] (Zeroth Order-Stochastic Variance Reduced Gradient, ZO-SVRG)。ZO-SVRG 不需要再进行梯度估计,仅用函数值近似代替全梯度和单个样本的梯度。经过理论和实验验证,其可以实现与 SVRG 相同数量级的收敛速度,同时避免了梯度的计算。Liu 等^[12]提出了 SZVR-G 算法。SZVR-G 算法迭代地从生成的高斯分布中随机抽取零阶优化的方向,经过理论推导可知,该算法的复杂度和数据的维数 d 线性相关,算法的性能得到提升。

为了解决使用传统支持向量机对噪声敏感和随机梯度下降求解的方法中存在较大方差的问题,本文提出了一种零阶减小方差的鲁棒支持向量机 (RSVM-ZOVR) 方法。通过引入正弦平方形式的损失函数来抑制噪声的影响,并利用基于零阶减小方差的方法求解所建立的优化问题,实验验证了方法的有效性。

本文的主要贡献有:

1) 提出了一种正弦平方形式的损失函数,对损失函数进行改进可使其具有明显的优势。正弦函数具有周期性,通过合理的参数设置可以将定义域缩小到一个周期内,配合对函数取平方的方式,可将损失函数值限制到 $[0, 1]$ 区间,降低了噪声数据对分类结果的影响。

2) 使用零阶减小方差算法 (ZO-SVRG) 求解所建立的模型,使用函数估计值代替梯度,避免了重复的梯度计算;引入梯度修正项以减弱方差的影响,提升算法的性能。此外,依据加速策略对算法的每次内层和外层输出分别进行加权平均计算,提高了优化问题的收敛速度。

2 减小方差的随机梯度下降算法

梯度下降 (Gradient Descent, GD) 是求解优化模型中最基础的一种方法。它的主要更新公式为:

$$\omega_{t+1} = \omega_t - \eta_t g(\omega_t) \quad (1)$$

其中, $g(\omega_t)$ 是优化问题中的目标函数在 ω_t 处的全梯度, η_t 为学习速率。梯度下降算法在每次迭代中需要遍历全部训练样本,并计算全梯度。当数据规模较大时这种算法的时间开销非常大。

随机梯度下降算法 (Stochastic Gradient Descent, SGD) 基于无偏估计^[13]的思想,使用单个样本的梯度代替全梯度,减少了计算量,弥补了 GD 算法的不足。然而,在更新时并不是使用梯度的精确值,因此在实际运行中单个样本 i 的梯度和全梯度之间存在方差,即:

$$E[\|g_i(\omega_t) - g(\omega_t)\|^2] \quad (2)$$

随着迭代的进行,方差也会逐次累加,最终降低算法的收敛速率,因此出现了一系列减小方差的算法。这些算法具有相似的框架,均分为内外两层循环并引入梯度修正项,其不同之处在于如何对梯度进行估计,本文将其分为一阶估计和零阶估计分别进行讨论。

2.1 一阶优化方法

SVRG^[9]是一种经典的一阶优化减小方差算法,SVRG分为内外两层循环,在每轮的外层循环中定义一个向量 $\tilde{\omega}$,这个向量是上一轮进行 T 次内层迭代后的最后一个值,同时仅在每轮的外层循环中计算所有样本在 $\tilde{\omega}$ 处的全梯度 $g(\tilde{\omega})$ 。在SVRG的 T 次内层循环中,每次随机抽取一个样本 i ,并计算单个样本梯度 $g_i(\omega_t)$,同时计算单个样本在 $\tilde{\omega}$ 处的梯度 $g_i(\tilde{\omega})$,并进行如下的梯度修正:

$$G(\omega_t) = g_i(\omega_t) - g_i(\tilde{\omega}) + g(\tilde{\omega}) \quad (3)$$

其中, $G(\omega_t)$ 称为梯度修正项。

一阶优化方法的迭代公式为:

$$\omega_{t+1} = \omega_t - \eta G(\omega_t) \quad (4)$$

SVRG算法如算法1所示。

算法1 SVRG(Stochastic Variance Reduced Gradient)算法

输入:外层迭代轮数 S ,内层迭代次数 T ,学习率 η

输出: $\tilde{\omega}_S$

初始化: $\tilde{\omega}_0$

1. for $s=1, 2, \dots, S$

2. $\tilde{\omega} = \tilde{\omega}_{s-1}$

3. 计算全梯度 $g(\tilde{\omega})$

4. $\omega_0 = \tilde{\omega}$

5. for $t=0, 1, \dots, T-1$

6. 随机抽取一个样本 i ,并进行梯度更新

7. $\omega_{t+1} = \omega_t - \eta(g_i(\omega_t) - g_i(\tilde{\omega}) + g(\tilde{\omega}))$

8. end

9. $\tilde{\omega}_s = \omega_T$

10. end

11. return $\tilde{\omega}_S$

2.2 零阶优化方法

对于一般的优化问题,我们定义如下的样例空间:给定 n 个输入数据, $S = \{(x_1, y_1), (x_2, y_2), \dots, (x_n, y_n)\}$, $x_i \in R^d$, d 表示样本的维数,样本标签 $y = \{+1, -1\}$ 。零阶优化方法与一阶优化方法的框架基本相同,均分为内外两层循环和相同形式的梯度修正项。不同于一阶优化方法,零阶优化方法不再计算样本的梯度,仅仅用函数值近似代替梯度,避免了重复的梯度计算。零阶优化的代表性算法是Liu提出的ZO-SVRG^[11]算法。ZO-SVRG算法在SVRG算法的基础上,计算全部样本在 $\tilde{\omega}$ 处的全梯度估计 $\hat{\nabla} F(\tilde{\omega})$,以及单个样本的梯度估计 $\hat{\nabla} F_i(\tilde{\omega})$ 和 $\hat{\nabla} F_i(\omega_t)$ 。对于零阶梯度估计,目前主要有以下3种方法:

1)随机梯度估计^[10]

$$\hat{\nabla} F_i(\tilde{\omega}) = (d/\mu)[F_i(\omega + \mu u_i) - F_i(\omega)]u_i \quad (5)$$

其中, d 表示样本的维数, $\mu > 0$ 是一个光滑参数, u_i 是单位圆上均匀分布的随机方向。

2)平均随机梯度估计^[14]

$$\hat{\nabla} F_i(\omega) = (d/(\mu q)) \sum_{j=1}^q [F_i(\omega + \mu u_{i,j}) - F_i(\omega)] u_{i,j} \quad (6)$$

平均随机梯度估计是随机梯度估计的一个变种,随机选取 q 个方向进行梯度估计求和,最后取其平均值。

3)坐标梯度估计^[15]

$$\hat{\nabla} F_i(\omega) = \sum_{l=1}^d (1/(2\mu_l)) [F_i(\omega + \mu_l e_l) - F_i(\omega - \mu_l e_l)] e_l \quad (7)$$

其中, e_l 表示一个标准基向量,在第 l 个坐标处为1,在其他坐标处为0。本文使用坐标梯度估计进行优化。

3 零阶减小方差的鲁棒支持向量机(RSVM-ZOVR)

本节对本文提出的算法进行讨论。基于2.2节中定义的样例空间,传统的支持向量机的表示形式如下:

$$\min F(\omega) = \frac{\lambda}{2} \|\omega\|^2 + \frac{1}{n} \sum_{i=1}^n \max(0, 1 - y_i \omega^T x_i) \quad (8)$$

其中, $\frac{\lambda}{2} \|\omega\|^2$ 是目标函数的正则化项; $x_i \in R^d$,表示第 i 个样本; $y_i \in \{+1, -1\}$,代表第 i 个样本标签。传统的支持向量机采用铰链损失,铰链损失函数的一个特点是无界性,对于噪声点,铰链损失函数会达到一个较大的值,影响目标函数的收敛速度和算法的分类性能。为了弥补传统支持向量机损失函数的不足,本文提出了一种新的损失函数:

$$\sin^2\left(\frac{1 - y_i \omega^T x_i}{k}\right) \quad (9)$$

本文提出的损失函数采用正弦函数的形式,给定一个比较大的 k 值,可以将 $\frac{1 - y_i \omega^T x_i}{k}$ 限制在 $[-\frac{\pi}{2}, \frac{\pi}{2}]$ 区间。根据正弦函数的性质,对正弦函数取平方,可使损失函数的值限制在 $[0, 1]$ 区间,即使对于噪声数据也不会出现损失函数值过大的情况。

3.1 非线性RSVM-ZOVR算法

在非线性输入空间,我们定义目标函数形式如下:

$$\min F(\omega) = \frac{\lambda}{2} \|\omega\|^2 + \frac{1}{n} \sum_{i=1}^n \sin^2\left(\frac{1 - y_i \omega^T \varphi(x_i)}{k}\right) \quad (10)$$

对于非线性的支持向量机, $\varphi(x_i)$ 的维数很高,用传统的求解方法求解目标函数有一定的困难。本文使用表示定理^[16]:

$$\omega = \sum_{i=1}^n \alpha_i \varphi(x_i) = X\alpha \quad (11)$$

其中, $\alpha = [\alpha_1, \alpha_2, \dots, \alpha_n]^T$, $X = [\varphi(x_1), \varphi(x_2), \dots, \varphi(x_n)]$ 。根据式(11)可得:

$$X^T \omega = X^T X \alpha = G\alpha, \omega^T \omega = \alpha^T X^T X \alpha = \alpha^T G \alpha$$

其中, $G = X^T X$ 表示核矩阵, G_i 表示 G 的第 i 列。式(10)可以表示为如下形式:

$$\min F(\alpha) = \frac{\lambda}{2} \alpha^T G \alpha + \frac{1}{n} \sum_{i=1}^n \sin^2\left(\frac{1 - y_i \alpha^T G_i}{k}\right) \quad (12)$$

对于式(12)的求解,不再引入拉格朗日函数求其对偶形式,而是直接采用零阶减小方差的方法。

零阶减小方差的方法首先对梯度进行估计。本文中零阶梯度估计使用坐标梯度估计^[15]方法:

$$\hat{\nabla} F_i(\alpha) = \sum_{l=1}^n (1/(2\mu_l)) [F_i(\alpha + \mu_l e_l) - F_i(\alpha - \mu_l e_l)] e_l \quad (13)$$

其次,为了减小方差,使用下面形式的梯度修正项:

$$G(\alpha_i) = \hat{g}_i(\alpha_i) - \hat{g}_i(\tilde{\alpha}) + \hat{g}(\tilde{\alpha}) \quad (14)$$

其中, $\hat{g}_i(\alpha_i)$, $\hat{g}_i(\tilde{\alpha})$ 分别代表单个样本在 α_i 和 $\tilde{\alpha}$ 处的梯度估计:

$$\hat{g}_i(\alpha_i) = \hat{\nabla} F_i(\alpha_i), \hat{g}_i(\tilde{\alpha}) = \hat{\nabla} F_i(\tilde{\alpha}) \quad (15)$$

$\hat{g}(\tilde{\alpha})$ 表示全部样本在 $\tilde{\alpha}$ 处的全梯度估计:

$$\hat{g}(\tilde{\alpha}) = \hat{\nabla} F(\tilde{\alpha}) \quad (16)$$

非线性 RSVM-ZOVR 算法如算法 2 所示。

算法 2 非线性 RSVM-ZOVR 算法

输入: 外层迭代轮数 S, 内层迭代次数 T, 学习率 η , 光滑参数 μ , 正则化参数 λ , 损失函数参数 k

输出: α

初始化 $\tilde{\alpha}_0$

1. for $s=1, 2, \dots, S$

2. $\tilde{\alpha} = \tilde{\alpha}_{s-1}$

3. 计算全梯度 $\hat{g}(\tilde{\alpha})$

4. $\alpha_0 = \tilde{\alpha}$

5. for $t=0, 1, \dots, T-1$

6. 随机抽取一个样本 i , 并进行梯度更新

7. $\alpha_{t+1} = \alpha_t - \eta(\hat{g}_i(\alpha_t) - \hat{g}_i(\tilde{\alpha}) + \hat{g}(\tilde{\alpha}))$

8. end

9. $\alpha_T = \frac{2}{(T+1)(T+2)} \sum_{t=0}^T (t+1) \alpha_t$

10. $\tilde{\alpha}_s = \alpha_T$

11. end

12. $\alpha = \frac{2}{(S+1)(S+2)} \sum_{s=0}^S (s+1) \tilde{\alpha}_s$

13. return α

非线性 RSVM-ZOVR 算法不再计算目标函数的梯度,而是用目标函数值进行逼近,降低了计算量。另外,我们引入梯度修正项,随机抽取单个样本后产生的方差:

$$E[\| \hat{g}_i(\alpha_i) - \hat{g}_i(\tilde{\alpha}) + \hat{g}(\tilde{\alpha}) - \hat{g}_i(\alpha_i) \|^2] \quad (17)$$

与式(2)相比,引入梯度修正项的式(17)具有更小的方差,另外在随后的实验中也证实了减小方差的效果。同时本文依据 Lacoste^[17]提出了加速策略,在算法的第 9 行和第 11 行中依次对内部迭代和外部迭代的输出分别进行了加权平均:

$$\alpha_T = \frac{2}{(T+1)(T+2)} \sum_{t=0}^T (t+1) \alpha_t \quad (18)$$

与未加权的算法相比,对输出进行加权平均后可以加速优化问题的收敛。

3.2 线性 RSVM-ZOVR 算法

实验表明,在 3.1 节提出的非线性 RSVM-ZOVR 算法,改善了传统支持向量机对噪声数据敏感的情况,提高了对噪

声数据的分类准确率。但当数据规模变大时,非线性算法往往需要大量时间来进行核矩阵运算,因此,对于较大规模的数据,本文提出下述算法。

对于线性情况,优化问题可以表示为:

$$\min F(\omega) = \frac{\lambda}{2} \|\omega\|^2 + \frac{1}{n} \sum_{i=1}^n \sin^2\left(\frac{1 - y_i \omega^T x_i}{k}\right) \quad (19)$$

不同于非线性算法,线性情况无需引入映射 φ ,也不涉及核矩阵的存储。因此,我们无需引入表示定理^[16],而是直接通过式(20)对 ω 进行迭代更新:

$$\omega_{t+1} = \omega_t - \eta(\hat{g}_i(\omega_t) - \hat{g}_i(\tilde{\omega}) + \hat{g}(\tilde{\omega})) \quad (20)$$

同样地, $\hat{g}_i(\omega_t) - \hat{g}_i(\tilde{\omega}) + \hat{g}(\tilde{\omega})$ 是我们引入的梯度修正项, $\hat{g}_i(\omega_t)$ 和 $\hat{g}_i(\tilde{\omega})$ 分别表示单个样本在 ω_t 和 $\tilde{\omega}$ 处的梯度估计, $\hat{g}(\tilde{\omega})$ 为全部样本的全梯度估计。线性 RSVM-ZOVR 算法和非线性 RSVM-ZOVR 算法有相同的框架。

线性 RSVM-ZOVR 算法如算法 3 所示。

算法 3 线性 RSVM-ZOVR 算法

输入: 外层迭代轮数 S, 内层迭代次数 T, 学习率 η , 光滑参数 μ , 正则化参数 λ , 损失函数参数 k

输出: ω

初始化: $\tilde{\omega}_0$

1. for $s=1, 2, \dots, S$

2. $\tilde{\omega} = \tilde{\omega}_{s-1}$

3. 计算全梯度 $\hat{g}(\tilde{\omega})$

4. $\omega_0 = \tilde{\omega}$

5. for $t=0, 1, \dots, T-1$

6. 随机抽取一个样本 i , 并进行梯度更新

7. $\omega_{t+1} = \omega_t - \eta(\hat{g}_i(\omega_t) - \hat{g}_i(\tilde{\omega}) + \hat{g}(\tilde{\omega}))$

8. end

9. $\omega_T = \frac{2}{(T+1)(T+2)} \sum_{t=0}^T (t+1) \omega_t$

10. $\tilde{\omega}_s = \omega_T$

11. end

12. $\omega = \frac{2}{(S+1)(S+2)} \sum_{s=0}^S (s+1) \tilde{\omega}_s$

13. return ω

3.3 算法的复杂度分析

本节依次对提出的线性 RSVM-ZOVR 算法和非线性 RSVM-ZOVR 算法进行时间复杂度分析。

基于上述的样例空间和算法设定,设样本个数为 n , 样本的维数为 d , 算法的外层循环轮数为 S , 内层迭代次数为 T 。

线性 RSVM-ZOVR 算法的主要工作量集中在内外双重循环的梯度计算部分。本文对梯度的计算不同于传统的方法,这里通过式(7)的坐标梯度估计法近似代替梯度,则坐标梯度估计(函数查询次数)的复杂度为 $O(d)$ ^[11]。又因为算法的内外双重循环,计算梯度的总次数为 $nS + TS$, 所以线性 RSVM-ZOVR 算法的时间复杂度为:

$$O(d(nS + TS)) = O(dS(n + T)) \quad (21)$$

同样地,按照上述相同的方式对非线性 RSVM-ZOVR 算法进行时间复杂度分析。非线性 RSVM-ZOVR 算法经过核函数的作用映射到高维特征空间,利用式(13)对梯度进行坐

标梯度估计,则坐标梯度估计(函数查询次数)的复杂度为 $O(n)$ 。同样地,根据算法的内外双重循环可知计算梯度的总次数为 $nS+TS$,因此非线性 RSVM-ZOVR 算法的时间复杂度为:

$$O(n(nS+TS))=O(S(n^2+nT)) \quad (22)$$

4 实验与结果

本节通过实验验证提出的零阶减小方差的鲁棒支持向量机(RSVM-ZOVR)算法的性能,实验主要分为以下 4 个部分。

1)抗噪性实验:进行线性 RSVM-ZOVR 算法和非线性 RSVM-ZOVR 算法的抗噪性实验。

2)方差分析实验:针对非线性情况,对比本文提出的算法和标准随机梯度下降算法(SGD),验证算法减小方差策略的有效性,并进行线性 RSVM-ZOVR 算法和非线性 RSVM-ZOVR 算法的方差对比分析实验。

3)针对非线性情况,对比本文的加权输出的 RSVM-ZOVR 算法和正常输出的 RSVM-ZOVR 算法,从而验证算法收敛速度的加速。

4)分析参数值 λ, k 对实验精度的影响。

实验程序的运行环境为 Matlab R2016a。实验数据来源于 KEEL¹⁾网站(训练集和测试集的比例均为 4:1),其分为两部分:1)常规噪声数据集,我们依次选取含有 0.5%,10%,15%属性噪声的数据进行实验;2)相对较大规模的标准数据集,为了验证算法的抗噪性,依次对这几个数据集加入 0.5%,10%,15%的均值为 0.2、方差为 0.5 的高斯噪声。实验数据集的具体情况如表 1 所列。

表 1 实验数据集

Table 1 Experimental datasets

数据名称	样例个数	特征数
sonar	208	60
ionosphere	351	33
wdbc	569	14
pima	768	8
spambase	4597	57
ring	7400	5
twonorm	7400	2
penbased	10992	16
magic	19020	10
shuttle	58000	9

4.1 抗噪性实验

本节进行线性和非线性 RSVM-ZOVR 算法的抗噪性实验。为了使实验效果更明显,本文进行两个对比实验:1)将新提出的算法分别与 SVM-ZOVR 算法和传统的 SVM 算法进行对比,这里的 SVM-ZOVR 算法是指使用本文提出的零阶减小方差的方法求解一般的支持向量机模型,传统 SVM 是指文献[8]中的 Pegasos 算法;2)将线性 RSVM-ZOVR 算法和非线性 RSVM-ZOVR 算法进行抗噪性对比。

4.1.1 RSVM-ZOVR 算法、SVM-ZOVR 算法和传统 SVM 算法的对比实验

本节分别将线性 RSVM-ZOVR 算法和非线性 RSVM-ZOVR 算法与 SVM-ZOVR 算法和传统 SVM 算法进行对比

实验。0.5%,10%,15%噪声数据的分类准确率如表 2—表 5 所列(实验结果为 5 折交叉验证的平均准确率)。

表 2 线性 RSVM-ZOVR 算法的分类准确率(0 噪声)

Table 2 Accuracy of linear RSVM-ZOVR algorithm (0 noise)
(单位:%)

数据名称	RSVM-ZOVR	SVM-ZOVR	传统 SVM
sonar	69.05±1.10	64.29±3.05	61.90±3.23
ionosphere	74.65±1.34	70.42±2.98	70.03±2.45
wdbc	81.58±1.32	79.09±3.07	78.82±2.97
pima	70.13±2.14	68.08±2.95	66.88±3.05
spambase	61.30±2.08	60.54±3.01	59.89±2.95
ring	70.61±1.38	73.24±3.23	70.47±2.08
twonorm	96.95±1.05	96.78±3.15	95.95±3.12
penbased	71.17±1.38	68.06±2.42	66.67±2.68
magic	67.06±1.75	64.54±3.01	63.04±2.08
shuttle	93.87±0.96	92.31±1.99	92.19±2.98

表 3 线性 RSVM-ZOVR 算法的分类准确率(5%噪声)

Table 3 Accuracy of linear RSVM-ZOVR algorithm (5% noise)
(单位:%)

数据名称	RSVM-ZOVR	SVM-ZOVR	传统 SVM
sonar	69.05±1.60	61.90±3.50	59.52±3.18
ionosphere	71.83±1.58	70.42±3.21	69.01±3.77
wdbc	80.70±1.67	77.19±2.97	76.95±2.58
pima	69.48±2.01	66.88±3.18	65.58±2.95
spambase	60.76±2.12	59.46±4.01	58.59±3.21
ring	67.09±1.35	71.69±3.05	66.01±2.38
twonorm	96.22±1.65	95.74±3.65	92.36±2.68
penbased	70.53±1.68	66.35±2.68	65.76±3.54
magic	65.27±1.99	62.35±2.08	61.54±2.95
shuttle	92.19±1.03	90.74±2.65	89.84±2.03

表 4 线性 RSVM-ZOVR 算法的分类准确率(10%噪声)

Table 4 Accuracy of linear RSVM-ZOVR algorithm (10% noise)
(单位:%)

数据名称	RSVM-ZOVR	SVM-ZOVR	传统 SVM
sonar	64.29±1.32	57.14±3.26	57.05±2.28
ionosphere	70.42±1.65	69.01±3.31	67.61±3.18
wdbc	79.82±1.48	75.44±3.68	73.68±2.89
pima	64.29±1.93	61.69±4.23	60.39±3.26
spambase	60.33±2.35	57.07±4.56	56.74±5.08
ring	63.04±1.97	68.24±3.58	62.09±3.96
twonorm	93.65±1.66	92.24±3.98	90.68±2.97
penbased	66.58±1.48	63.01±3.31	62.89±4.38
magic	64.17±1.38	61.07±2.31	58.52±2.86
shuttle	90.99±1.12	87.97±2.56	89.47±3.58

表 5 线性 RSVM-ZOVR 算法的分类准确率(15%噪声)

Table 5 Accuracy of linear RSVM-ZOVR algorithm (15% noise)
(单位:%)

数据名称	RSVM-ZOVR	SVM-ZOVR	传统 SVM
sonar	61.90±1.67	54.76±4.25	54.06±3.95
ionosphere	69.01±1.34	67.61±2.98	66.20±3.92
wdbc	73.16±1.69	72.81±9.87	69.30±4.05
pima	62.99±2.03	60.39±2.65	59.74±3.59
spambase	60.11±1.57	55.11±3.58	54.67±4.08
ring	59.12±1.59	67.30±6.67	57.97±3.95
twonorm	90.95±1.69	90.07±1.94	87.30±3.91
penbased	64.53±1.98	60.99±2.05	60.71±4.05
magic	63.83±1.65	60.08±2.48	54.65±2.93
shuttle	86.03±1.21	85.16±1.91	84.34±3.91

¹⁾ <http://www.keel.es/>

由表2—表5的运行结果可知,对于给定的10组数据集,与SVM-ZOVR算法相比,除ring数据之外,新提出的线性RSVM-ZOVR算法在其余数据集上均有相对较高的准确率;与传统的SVM算法相比,新提出的RSVM-ZOVR算法在全部10组数据集中均具有相对高的准确率。这说明本文算法对噪声数据有一定的鲁棒性,弥补了传统支持向量机对噪声数据分类的不足。同时从运行结果可以看出,随着噪声百分比的增加,分类的准确率随之降低。

在给出非线性RSVM-ZOVR算法的分类精度之前,我们首先比较线性RSVM-ZOVR算法和非线性RSVM-ZOVR算法在较大规模数据上的运行时间差异,以便进行后续的验证实验。表6列出了线性RSVM-ZOVR算法和非线性RSVM-ZOVR算法在较大规模数据集上迭代800次的运行时间。

表6 运行时间

Table 6 Running time

(单位:s)

数据名称	运行时间	
	线性算法	非线性算法
penbased	3.67	438.67
magic	3.26	771.48
shuttle	6.40	6263.05

由表6的结果可知,当对较大规模数据进行分类时,线性算法的运行时间远远短于非线性算法的运行时间,这有效地验证了当对较大规模数据进行分类时,线性RSVM-ZOVR算法在运行时间方面明显优于非线性RSVM-ZOVR算法。因此,为了高效地进行后续的验证实验,我们在常规的数据上进行了非线性RSVM-ZOVR算法的对比实验。非线性RSVM-ZOVR算法的对比实验结果如表7—表10所列(实验结果为5折交叉验证的平均准确率)。

表7 非线性RSVM-ZOVR算法的准确率(0噪声)

Table 7 Accuracy of no-linear RSVM-ZOVR algorithm (0 noise)

(单位:%)

数据名称	RSVM-ZOVR	SVM-ZOVR	传统 SVM
sonar	74.67±1.12	64.37±3.21	64.28±4.01
ionosphere	79.24±1.31	78.01±3.56	77.46±3.08
wdbc	73.05±1.28	64.91±4.21	64.79±2.27
pima	78.18±2.05	64.99±3.04	64.93±4.24
spambase	61.54±2.24	60.94±2.01	60.33±2.88
ring	71.91±0.92	68.01±2.50	67.32±3.64
twonorm	96.99±1.36	96.01±2.68	95.95±3.58

表8 非线性RSVM-ZOVR算法的准确率(5%噪声)

Table 8 Accuracy of no-linear RSVM-ZOVR algorithm (5% noise)

(单位:%)

数据名称	RSVM-ZOVR	SVM-ZOVR	传统 SVM
sonar	73.81±1.03	64.29±3.68	61.90±3.54
ionosphere	78.87±1.21	74.65±3.99	73.24±4.09
wdbc	72.81±1.68	63.16±4.03	63.15±5.23
pima	74.65±2.30	64.29±3.56	61.03±3.58
spambase	60.11±1.68	59.94±2.06	59.56±3.06
ring	70.13±1.63	66.99±2.58	66.52±1.58
twonorm	96.69±0.96	95.31±2.97	93.92±3.95

表9 非线性RSVM-ZOVR算法的准确率(10%噪声)

Table 9 Accuracy of no-linear RSVM-ZOVR algorithm (10% noise)

(单位:%)

数据名称	RSVM-ZOVR	SVM-ZOVR	传统 SVM
sonar	66.88±1.65	61.90±3.85	59.52±3.77
ionosphere	74.65±1.59	70.42±3.69	69.01±2.96
wdbc	68.42±2.34	61.40±4.21	59.65±3.59
pima	70.42±1.77	63.64±3.25	60.39±3.85
spambase	59.46±1.89	58.95±1.59	57.39±4.19
ring	68.24±2.65	64.14±2.95	64.09±3.75
twonorm	94.08±1.32	93.24±2.93	92.91±3.03

表10 非线性RSVM-ZOVR算法的准确率(15%噪声)

Table 10 Accuracy of no-linear RSVM-ZOVR algorithm (15% noise)

(单位:%)

数据名称	RSVM-ZOVR	SVM-ZOVR	传统 SVM
sonar	64.29±1.89	59.52±3.98	57.14±3.89
ionosphere	73.81±0.92	67.61±4.35	67.61±4.86
wdbc	67.54±1.26	59.65±3.28	57.89±3.74
pima	67.65±1.34	62.34±3.65	59.74±4.02
spambase	58.35±1.59	57.94±2.26	56.09±3.56
ring	66.99±1.95	62.81±3.25	61.82±2.93
twonorm	91.81±1.52	90.81±1.48	89.73±3.88

实验中选取 sonar, ionosphere, wdbc, pima, spambase, ring, twonorm 7组数据集对非线性RSVM-ZOVR算法的抗噪性进行对比实验。实验中,RSVM-ZOVR算法、SVM-ZOVR算法以及传统的SVM算法均采用高斯核函数。同样地,与SVM-ZOVR算法、传统的SVM算法相比,本文提出的算法在这几组数据中均具有相对较高的分类准确率。同时,通过表7—表10的竖向比较可以看出,随着数据噪声百分比的增加,各种算法的分类准确率都会随之降低。

4.1.2 线性RSVM-ZOVR算法和非线性RSVM-ZOVR算法的对比实验

本节对线性RSVM-ZOVR算法和非线性RSVM-ZOVR算法进行抗噪性对比实验。0,5%,10%,15%噪声数据的分类准确率如表11—表14所列(实验结果为5折交叉验证的平均准确率)。由表11—表14可知,在给定的7组不同噪声比的实验数据集上,除wdbc和spambase数据集外,非线性RSVM-ZOVR算法的分类精度均高于线性RSVM-ZOVR算法的分类精度。这说明与线性RSVM-ZOVR算法相比,非线性RSVM-ZOVR算法有着较高的分类准确率,分类性能更好。

表11 线性RSVM-ZOVR与非线性RSVM-ZOVR的准确率对比(0噪声)

Table 11 Accuracy comparison of linear RSVM-ZOVR and no-linear RSVM-ZOVR (0 noise)

(单位:%)

数据名称	线性 RSVM-ZOVR	非线性 RSVM-ZOVR
sonar	69.05±1.10	74.67±1.12
ionosphere	74.65±1.34	79.24±1.31
wdbc	81.58±1.32	73.05±1.28
pima	70.13±2.14	78.18±2.05
spambase	61.30±2.08	61.54±2.24
ring	70.61±1.38	71.91±0.92
twonorm	96.95±1.05	96.99±1.36

表 12 线性 RSVM-ZOVR 与非线性 RSVM-ZOVR 的
准确率对比(5%噪声)

Table 12 Accuracy comparison of linear RSVM-ZOVR and
no-linear RSVM-ZOVR (5% noise)
(单位:%)

数据名称	线性 RSVM-ZOVR	非线性 RSVM-ZOVR
sonar	69.05±1.60	73.81±1.03
ionosphere	71.83±1.58	78.87±1.21
wdbc	80.70±1.67	72.81±1.68
pima	69.48±2.01	74.65±2.30
spambase	60.76±2.12	60.11±1.68
ring	67.09±1.35	70.13±1.63
twonorm	96.22±1.65	96.69±0.96

表 13 线性 RSVM-ZOVR 与非线性 RSVM-ZOVR 的
准确率对比(10%噪声)

Table 13 Accuracy comparison of linear RSVM-ZOVR and
no-linear RSVM-ZOVR (10% noise)
(单位:%)

数据名称	线性 RSVM-ZOVR	非线性 RSVM-ZOVR
sonar	64.29±1.32	66.88±1.65
ionosphere	70.42±1.65	74.65±1.59
wdbc	79.82±1.48	68.42±2.34
pima	64.29±1.93	70.42±1.77
spambase	60.33±2.35	59.46±1.89
ring	63.04±1.97	68.24±2.65
twonorm	93.65±1.66	94.08±1.32

表 14 线性 RSVM-ZOVR 与非线性 RSVM-ZOVR 的
准确率对比(15%噪声)

Table 14 Accuracy comparison of linear RSVM-ZOVR and
no-linear RSVM-ZOVR (15% noise)
(单位:%)

数据名称	线性 RSVM-ZOVR	非线性 RSVM-ZOVR
sonar	61.90±1.67	64.29±1.89
ionosphere	69.01±1.34	73.81±0.92
wdbc	73.16±1.69	67.54±1.26
pima	62.99±2.03	67.65±1.34
spambase	60.11±1.57	58.35±1.59
ring	59.12±1.59	66.99±1.95
twonorm	90.95±1.69	91.81±1.52

4.2 方差分析实验

本节进行方差分析实验,该实验主要包括两部分:1)非线性 RSVM-ZOVR 算法与传统的随机梯度下降算法的对比实验,以对减小方差进行验证;2)进行线性 RSVM-ZOVR 算法和非线性 RSVM-ZOVR 算法的方差分析实验。

4.2.1 与随机梯度下降算法的方差对比实验

图 1—图 4 为非线性 RSVM-ZOVR 算法和传统随机梯度下降算法对 pima 数据进行分类的方差对比图。其中,横坐标为迭代次数,纵坐标为每次迭代中产生的方差。

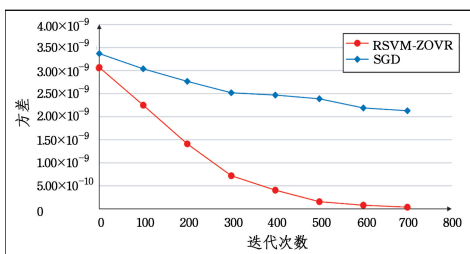


图 1 0 噪声时 RSVM-ZOVR 与 SGD 的方差对比

Fig. 1 Variance comparison of RSVM-ZOVR and SGD with 0 noise

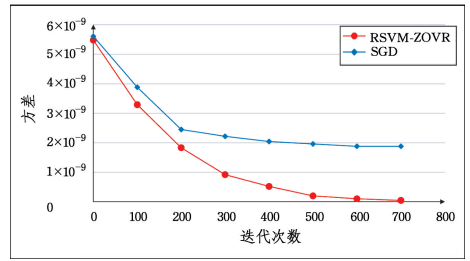


图 2 5%噪声时 RSVM-ZOVR 与 SGD 的方差对比

Fig. 2 Variance comparison of RSVM-ZOVR and SGD with 5% noise

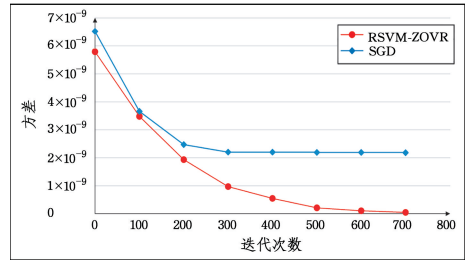


图 3 10%噪声时 RSVM-ZOVR 与 SGD 的方差对比

Fig. 3 Variance comparison of RSVM-ZOVR and SGD with 10% noise

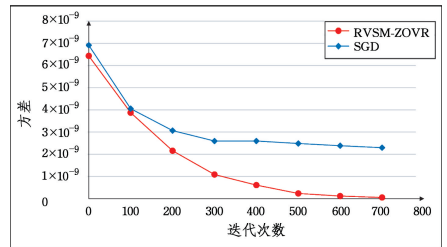


图 4 15%噪声时 RSVM-ZOVR 与 SGD 的方差对比

Fig. 4 Variance comparison of RSVM-ZOVR and SGD with 15% noise

由图 1—图 4 可以看出,对于不同噪声百分比的数据,在随机梯度下降算法前 300 次的迭代中,方差逐步减小,但在 300 次以后方差基本维持在一个较大的值,变化很小(方差维持在一个较大的值,并不能说明算法收敛)。相比之下,随着迭代的进行,本文提出的算法的方差逐步减小,在 700 次迭代后趋于一个接近 0 的定值。并且在整个迭代过程中,本文提出的算法的方差一直小于随机梯度下降算法的方差。这表明,从减小方差的角度考虑,本文提出的算法可以对方差进行有效的修正。

4.2.2 线性 RSVM-ZOVR 和非线性 RSVM-ZOVR 的方差分析

本节分别记录线性 RSVM-ZOVR 算法和非线性 RSVM-ZOVR 算法对 pima 数据集进行分类时方差的变化情况。图 5—图 8 依次为 pima 数据含有 0,5%,10%,15%属性噪声时的方差变化情况。由图 5—图 8 可以看出,对于不同噪声百分比的数据,随着迭代过程的进行,线性 RSVM-ZOVR 算法和非线性 RSVM-ZOVR 算法的方差均随之减小。从整体对比分析来看,对于不同噪声比的实验数据,在算法的迭代过程中,非线性 RSVM-ZOVR 算法的方差始终比线性 RSVM-

ZOVR算法的方差小。从方差的下降速度来看,对于不同噪声比的实验数据,相比线性RSVM-ZOVR算法,非线性RSVM-ZOVR算法的方差下降速度快且稳定,这说明了非线性RSVM-ZOVR算法在降低方差方面优于线性RSVM-ZOVR算法。

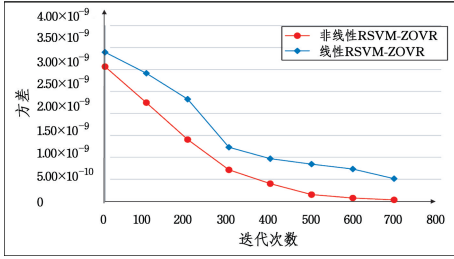


图5 0噪声时线性RSVM-ZOVR与非线性RSVM-ZOVR的方差对比

Fig. 5 Variance comparison of linear and no-linear RSVM-ZOVR with 0 noise

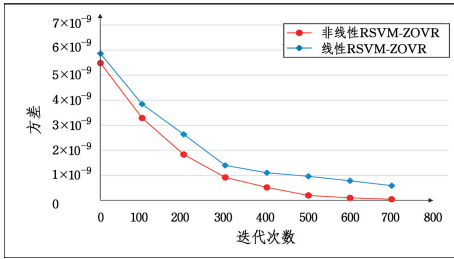


图6 5%噪声时线性RSVM-ZOVR与非线性RSVM-ZOVR的方差对比

Fig. 6 Variance comparison of linear and no-linear RSVM-ZOVR with 5% noise

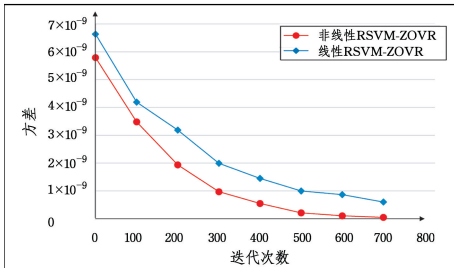


图7 10%噪声时线性RSVM-ZOVR与非线性RSVM-ZOVR的方差对比

Fig. 7 Variance comparison of linear and no-linear RSVM-ZOVR with 10% noise

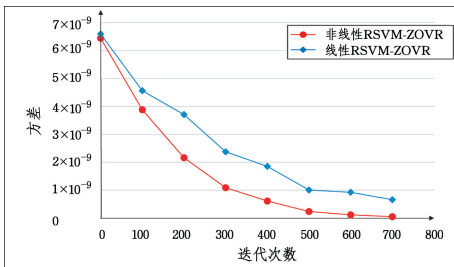


图8 15%噪声时线性RSVM-ZOVR与非线性RSVM-ZOVR的方差对比

Fig. 8 Variance comparison of linear and no-linear RSVM-ZOVR with 15% noise

4.3 收敛速度验证实验

本节进行非线性加权输出的RSVM-ZOVR算法和非线性正常输出的RSVM-ZOVR算法的收敛速度的对比实验。图9—图12为两种方法对wdbc数据进行分类的函数收敛对比图,其中,横坐标为迭代次数,纵坐标为目标函数值的变化。由图9—图12可以看出,对于不同百分比的噪声数据,当采用提出的加权输出RSVM-ZOVR算法进行求解时,在前400次迭代中函数值逐步减小,在400次迭代后函数值逐步趋于一个定值,算法达到收敛。对于正常输出的RSVM-ZOVR算法,从开始到第600次迭代的过程中,函数值处于逐次减小的状态,直到600次之后函数值才趋于稳定。这说明提出的加权输出策略可以加速算法的收敛。同时,在整个迭代过程中,提出的加权输出算法相比于正常输出的算法有着较小的函数值,性能更优。

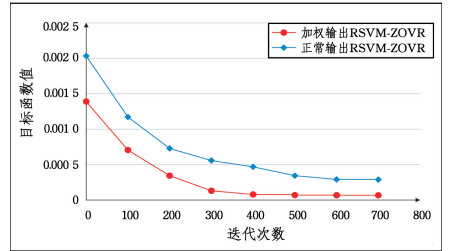


图9 0噪声时的函数收敛对比图

Fig. 9 Comparison of function convergence with 0 noise

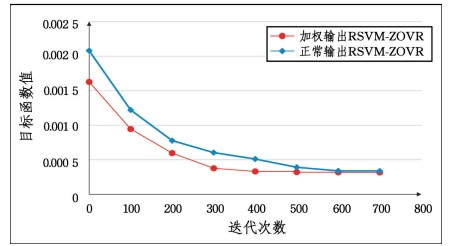


图10 5%噪声时的函数收敛对比图

Fig. 10 Comparison of function convergence with 5% noise

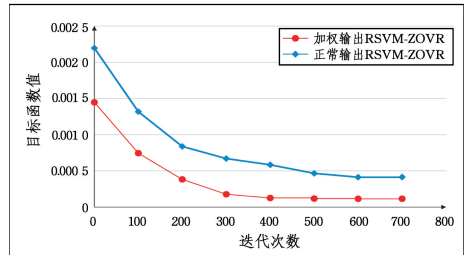


图11 10%噪声时的函数收敛对比图

Fig. 11 Comparison of function convergence with 10% noise

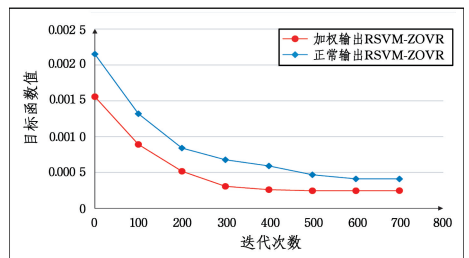


图12 15%噪声时的函数收敛对比图

Fig. 12 Comparison of function convergence with 15% noise

4.4 参数分析实验

本节分析算法的主要参数 λ, k 对实验精度的影响。其中, λ 为正则化参数, k 为损失函数参数。我们依照网格搜索的方法, 分别对线性 RSVM-ZOVR 算法和非线性 RSVM-ZOVR 算法使用不同参数对, 并记录其分类精度。表 15 和表 16 列出了线性 RSVM-ZOVR 算法和非线性 RSVM-ZOVR 算法下 pima 数据 5 折交叉验证的平均准确率。

表 15 线性 RSVM-ZOVR 算法的准确率

Table 15 Accuracy of linear RSVM-ZOVR algorithm

(单位: %)

噪声比/%	k 值	$\lambda=0.1$	$\lambda=1$	$\lambda=10$
0	1	44.81±1.36	64.94±0.89	51.95±1.39
	3	66.23±1.69	68.69±1.12	66.23±1.95
	5	61.04±1.89	70.13±2.14	57.79±1.87
5	1	36.36±2.56	65.58±1.65	55.84±2.01
	3	56.49±2.98	62.99±1.32	59.74±2.65
	5	57.14±1.75	69.48±2.01	55.19±1.99
10	1	54.55±1.93	58.44±2.01	57.14±2.14
	3	53.64±1.10	61.69±1.32	59.74±2.32
	5	57.14±2.69	64.29±1.93	59.09±2.04
15	1	50.00±3.54	59.74±2.63	44.16±1.96
	3	54.74±2.14	57.79±2.14	52.34±2.15
	5	56.49±2.58	62.99±2.03	44.81±3.65

表 16 非线性 RSVM-ZOVR 算法的准确率

Table 16 Accuracy of no-linear RSVM-ZOVR algorithm

(单位: %)

噪声比/%	k 值	$\lambda=0.1$	$\lambda=1$	$\lambda=10$
0	1	51.09±1.73	59.68±1.96	61.69±1.89
	3	64.29±1.12	63.64±1.38	78.18±2.05
	5	64.94±1.93	61.69±1.82	64.99±1.68
5	1	58.64±3.56	59.74±2.45	60.39±2.63
	3	59.74±3.25	63.64±3.12	74.65±2.30
	5	57.09±2.93	60.94±2.98	62.14±1.99
10	1	62.99±1.99	57.79±1.96	63.64±1.14
	3	56.49±2.38	62.01±2.56	70.42±1.77
	5	58.44±3.01	58.44±1.85	62.99±1.68
15	1	53.64±1.35	58.44±2.93	59.09±2.41
	3	62.99±3.11	64.29±3.02	67.65±1.34
	5	61.04±2.88	64.29±2.21	64.29±2.87

首先对表 15 中关于线性 RSVM-ZOVR 算法的参数进行分析。先进行横向对比: 在固定噪声比和 k 值的条件下, 根据分类精度可以得出, 当 $\lambda=1$ 时分类精度较高。其次进行纵向对比: 在固定噪声比和 $\lambda=1$ 的条件下, 根据结果可得当 k 为 5 时分类精度较高。因此, 当参数对满足 ($\lambda=1, k=5$) 时, 线性 RSVM-ZOVR 算法的性能最优。按照同样的方式, 对表 16 中关于非线性 RSVM-ZOVR 算法的参数进行分析得出, 当参数对满足 ($\lambda=10, k=3$) 时, 算法的性能最优。

结束语 针对传统支持向量机对噪声数据敏感的情况, 本文提出了一种基于零阶减小方差方法的鲁棒支持向量机。其用一种正弦平方形式的损失函数代替传统支持向量机的铰链损失, 使用坐标梯度估计方法计算目标函数值并以此近似代替梯度, 结合减小方差算法中梯度修正项的思想, 提出了 RSVM-ZOVR 算法。实验结果表明, 与 SVM-ZOVR 算法和传统的 SVM 算法相比, 本文提出的算法对噪声数据有更好的鲁棒性并且具有较小的方差。最后, 本文引入了对输出加权的策略, 分别对内外层迭代的输出结果进行加权平均, 与未

加权的算法相比, 提出的算法的收敛速度得到提高。

参考文献

- [1] FRENAY B, VERLEYSEN M. Classification in the Presence of Label Noise; A Survey [C] // IEEE Transactions on Neural Networks and Learning Systems, 2014; 845-869.
- [2] LIU Y F, ZHANG H H. Support Vector Machines with Adaptive Lq Penalty [J]. Computational Statistic & Data Analysis, 2007, 51(12): 6380-6394.
- [3] LIN C F, WANG S D. Fuzzy support vector machines [J]. IEEE Transactions on Neural Networks, 2002, 13(2): 464-471.
- [4] WU Y, LIU Y. Adaptively Weighted Large Margin Classifiers [J]. Journal of Computational and Graphical Statistics, 2013, 22(2): 416-432.
- [5] SUN S C, HUANG D. A Novel Robust Smooth Support Vector Machine [J]. Applied Mechanics and Materials, 2012, 1603 (297): 1438-1441.
- [6] HUANG X L, SHI L. Ramp Loss Linear Programming Support Vector Machine [J]. Machine Learning, 2014, 15 (1): 2185-2211.
- [7] XU G B, CAO Z, HU B G. Robust Support Vector Machines Based on the Rescaled Hinge loss function [J]. Pattern Recognition, 2017, 63: 139-148.
- [8] SHALEV-SHWARTZ S, SINGER Y. Pegasos: Primal Estimated Sub-Gradient Solver for SVM [J]. Mathematical Programming, 2011, 127(1): 3-30.
- [9] ZHANG J R. Accelerating Stochastic Gradient Descent using Predictive Variance Reduction [C] // Advances in Neural Information Processing Systems, 2013: 315-323.
- [10] NESTEROV Y. Random gradient-free minimization of convex functions [J]. Foundations of Computational Mathematics, 2015, 2(17): 527-566.
- [11] LIU S, KAILKHURA B. Zeroth-Order Stochastic Variance Reduction for Nonconvex Optimization [J]. arXiv:1805.10367v2.
- [12] LIU L, CHENG M. Stochastic Zeroth-order Optimization via Variance Reduction Method [J]. arXiv:1805.11811v1.
- [13] NEMIROVSKI A, JUDITSKY A. Robust Stochastic Approximation Approach to Stochastic Programming [J]. SIAM Journal on Optimization: A Publication of the Society for Industrial and Applied Mathematics, 2009, 19(4): 1574-1609.
- [14] DUCHII J C, JORDAN M I. Optimal rates for zeroth-order convex optimization: The power of two function evaluations [J]. IEEE Transactions on Information Theory, 2015, 61(5): 2788-2806.
- [15] GU B, HUO Z. Zeroth-Order Asynchronous Doubly Stochastic Algorithm with Variance Reduction [J]. arXiv:1612.01425v1.
- [16] CHENG F, ZHANG J. Large Cost-Sensitive Margin Distribution Machine for Imbalanced Data Classification [J]. Neurocomputing, 2017, 224(8): 45-57.
- [17] LACOSTE-JULIEN S, SCHMIDT M. A Simpler Approach to Obtaining an $O(1/t)$ Convergence Rate for Projected Stochastic sub-gradient Method [J]. arXiv:1212.2002v2.