

DOI:10.13232/j.cnki.jnju.2020.01.011

求解非凸截断 L_1 -SVM 的多阶段非精确线搜割平面方法

袁友宏, 刘 欣, 鲍 蕾*

(中国人民解放军陆军炮兵防空兵学院, 合肥, 230031)

摘 要: 截断 Hinge 损失能够获得更为稀疏的支持向量, 因此在鲁棒性上有显著的优点, 但却由此导致了难以求解的非凸问题. MM (Majorization-Minimization) 是一种求解非凸问题的一般框架, 多阶段 MM 策略已经在稀疏性上取得了很好的效果, 但是计算复杂度较高. 另一方面, 非精确线搜割平面方法可以高效求解线性支持向量机问题. 针对截断 L_1 -SVM (L_1 Support Vector Machine) 这一非凸非光滑问题, 提出一种基于非精确线性搜索的多阶段割平面方法, 避免每个阶段都进行批处理求解, 克服了计算复杂度高的缺点, 具有每个阶段求解速度快的优点. 该算法适用于大规模问题的求解, 也从理论上保证了其收敛性. 最后, 与其他多阶段算法进行了实验对比, 验证了该方法的有效性.

关键词: 截断 Hinge 损失, 非凸优化, 多阶段策略, 非精确线性搜索

中图分类号: TP301

文献标识码: A

A multi-stage cutting plane method with inexact line search for solving non-convex truncated L_1 -SVMs

Yuan Youhong, Liu Xin, Bao Lei*

(Army Academy of Artillery and Air Defense of PLA, Hefei, 230031, China)

Abstract: The truncated Hinge loss can get more sparse support vectors, thus it has significant advantages in robustness. But it leads to a non-convex problem which is difficult to solve. MM (Majorization-Minimization) is a general framework and is suitable for solving non-convex problems. The specific multi-stage MM strategy has a good effect in sparsity for the non-convex problem considering the structure of the objective function, but higher computational complexity is caused. On the other hand, a cutting plane method with inexact line search can efficiently solve linear support vector machines, and the theoretical analysis shows that it has the same optimal convergence bound as the exact line search method with a higher speed and lower cost. In this paper, for the truncated L_1 -SVM (L_1 Support Vector Machine) which is non-convex and non-smooth, we propose a multi-stage cutting plane method with inexact line search. It avoids solving the problem by a batch method at each stage, which overcomes the shortcoming of high computational complexity. Meanwhile, there is an advantage that a faster solution can be got at each stage. The algorithm is suitable for solving large-scale problems and it is convergent in theory. Finally, comparison experiments with other multi-stage algorithm demonstrate that our method is effective.

Key words: truncated Hinge loss, non-convex optimization, multi-stage strategy, inexact line search

支持向量机 (Supporting Vector Machine, SVM) 是 Vapnik^[1] 在 1995 年提出的一种标准机器

学习算法, 在过去的二十多年里, SVM 的研究已经取得非常显著的成果. 截断 Hinge 损失能得到

基金项目: 国家自然科学基金 (61673394), 安徽省自然科学基金 (1908085MF193)

收稿日期: 2019-08-01

* 通讯联系人, E-mail: baolei1219@sina.cn

更稀疏的支持向量,但会使原问题变为非凸问题,增加了计算复杂度.割平面算法(Cutting Plane Algorithm, CPA)最早由 Kelley^[2]于1960年提出,后来被用来解决线性SVM问题,和传统SVM方法相比,CPA的求解速度大大加快.

SVM优化问题中,Hinge损失^[3](又称 L_1 -Loss)的引入是其取得成功的关键性因素^[4-6].一方面,在无限样本的情况下,Hinge损失所具有的统计一致性保证SVM的解能收敛到贝叶斯最优分类器^[5];另一方面,Hinge损失的特点使分类器只利用了非零向量,这些非零向量被称为支持向量.然而,支持向量的数量会随着样本数线性增加^[7],导致在解决大规模问题的时候,时间消耗过大.针对这类问题,文献[8-10]通过截断损失函数把SVM的一些不必要的误差样本点剔除,这些误差样本点称为outlier点,但却由此导致了棘手的非凸问题.光滑条件下的非凸问题比较好解决,传统的解决方案是直接利用凹凸优化技巧,例如DCA(Difference Convex Algorithm)算法^[11]和CCCP(Concave-Convex Procedure)算法^[12],通过分解目标非凸函数为一系列凸的子问题来得到一个局部解.Mairal^[13]将DCA和CCCP这类处理方法都归结为一个统一的MM(Majorization-Minimization)框架.MM框架特别适用于处理有限和形式的目标函数,在非凸领域中被广泛应用.Mairal^[14]于2015年提出一种增量MM算法,增量计算比传统批处理方法的计算复杂度低.Fang et al^[15]利用随机估计对非凸光滑问题进行最优化,也适用于有限和形式的目标函数.Carmon et al^[16]利用梯度计算对非凸问题进行加速,非常适用于大规模问题的求解.此外,Lan and Yang^[17]于2019年提出一种针对上述有限和目标函数的加速随机算法,在速度上取得了很好的效果.但这些方法前提是要满足一阶稳定点条件,即目标函数可以是非凸的但必须是光滑的,满足Lipschitz条件.而截断Hinge损失函数不满足这一条件.2018年Tao et al^[18]为得到稀疏的支持向量并减少计算复杂度,提出多阶段支持向量机(Multi-stage Supporting Vector Machine, MS-SVM)算法,其原理是在每次迭代之前将满足outlier点的样本点删去,目标函数被截断,再用新的替代函数

逼近原始目标函数,然后优化新的替代函数得到最优解.MS-SVM算法可以看作MM框架下的一种考虑损失函数结构的特殊方法,但因为在每个阶段都要批处理求解一个SVM问题的精确解,计算复杂度很高,在大规模数据上效果不理想.

还有很多可以高效求解线性SVM的算法,如割平面算法、Pegasos算法^[19].Joachims^[20]在2006年引入割平面算法来解决线性SVM问题,并证明了算法的收敛性.后来Teo et al^[21]将其推广为求解一般正则化损失函数的方法,但在处理实际问题时,割平面算法的性能未能得到充分发挥,并且不能保证主问题单调下降.2008年,Franc and Sonnenburg^[22]在此基础上提出优化割平面算法(Optimized Cutting Plane Algorithm, OCAS),在每次迭代过程中,对优化变量 w 进行精确线性搜索,使目标函数单调下降且保证算法收敛.最近,Chu et al^[23]提出一种加速精确线性搜索割平面算法,比Pegasos算法的求解速度更快.2014年储德军等^[24]提出一种非精确线性搜索割平面算法,对非光滑目标函数进行了收敛性分析,能保证目标函数单调下降,并且每次迭代都采用非精确线性搜索,使时间消耗大大降低.本文分析表明,这种方法的计算复杂度低于加速精确线性搜索割平面算法.

多阶段策略在处理非凸非光滑问题时依然能保证支持向量的稀疏性,而非精确线性搜索则具有快速求解的优点,可以迅速进入下一阶段.鉴于此,本文提出一种基于非精确线性搜索的多阶段方法,在降低时间消耗的同时保证算法的单调性,并且得到了稀疏的支持向量.

1 截断损失和稀疏性分析

定义集合 $S = \{(x_1, y_1), (x_2, y_2), \dots, (x_m, y_m)\}$, 其中, $(x_i, y_i) \in \mathbb{R}^N \times \{-1, 1\}$, $i = 1, 2, \dots, m$, 且样本为独立同分布.本文优化的目标函数如下:

$$\min_w F(w) \quad (1)$$

其中,

$$F(w) = \frac{1}{2} \|w\|^2 + C \sum_{i=1}^m L_1[y_i(\langle w, x_i \rangle + b)]$$

$\mathbf{w} \in \mathbb{R}^N$ 是优化问题的解向量. L_1 函数的具体形式如式(2)所示.

首先介绍 MM 算法. 在当前最优值 \mathbf{w}_{t-1} 附近找到一个便于优化的替代函数 g_t , g_t 满足一阶替代函数条件, 如定义 1 所示. 通过优化函数 g_t 得到最优解 \mathbf{w}_t . 对 MM 框架的描述如算法 1 所示.

算法 1 基本 MM 算法

Input: $\mathbf{w}_0 \in \Theta$ (初始估计), T (迭代次数)

Repeat

$t = 1$

1. 在 \mathbf{w}_{t-1} 附近找一个 F 的替代函数 g_t

2. 最小化替代函数 g_t , 求得最优解 \mathbf{w}_t

$t = t + 1$

Until: $t = T$

Output: \mathbf{w}_T

定义 1 替代函数 函数 $g: \mathbb{R}^N \rightarrow \mathbb{R}$ 满足如下两个条件, 则为函数 F 在点 \mathbf{w} 处的替代函数:

(1) 对于函数 g 的所有最小值 \mathbf{w}' , 有 $g(\mathbf{w}') > F(\mathbf{w}')$ 成立, 若更一般的条件 $g > F$ 成立, 可以称函数 g 是一个 majorizing 项.

(2) 近似误差 $h \triangleq g - F$ 是光滑误差, 而且 $h(\mathbf{w}') = 0, \nabla h(\mathbf{w}') = 0$.

下面对截断 Hinge 损失函数进行分析. 令 $L_1(u) = (1 - u)_+$, 其中,

$$(u)_+ = \begin{cases} u, & u \geq 0 \\ 0, & u < 0 \end{cases} \quad (2)$$

显然, L_1 -SVM 需要最小化如下目标函数:

$$F(\mathbf{w}) = \frac{1}{2} \|\mathbf{w}\|^2 + C \sum_{i=1}^m L_1[y_i(\langle \mathbf{w}, \mathbf{x}_i \rangle + b)] \quad (3)$$

其中, 参数 $C > 0$.

截断形式的 Hinge 损失定义如下:

$$\hat{L}_1(u) = (1 - u)_+ - (\delta - u)_+ \quad (4)$$

显然, 函数 $\hat{L}_1(u)$ 是非光滑函数. 其中参数 $\delta (\delta \leq 0)$ 的值代表截断点的位置, 如图 1 所示. 这样的截断策略能保证 $u < \delta$ 的样本点不成为支持向量. 那么截断 L_1 -SVM 需要优化如下目标函数:

$$\hat{F}(\mathbf{w}) = \frac{1}{2} \|\mathbf{w}\|^2 + C \sum_{i=1}^m \hat{L}_1[y_i(\langle \mathbf{w}, \mathbf{x}_i \rangle + b)] \quad (5)$$

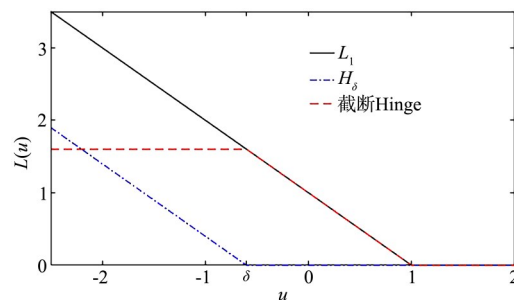


图 1 截断损失函数

Fig. 1 Truncated-Loss function

令 $H_\delta(u) = (\delta - u)_+$, 显然 H_δ 是一个凸函数, 所以截断 Hinge 损失函数可以由两个凸函数之差构成, 其中两个凸函数分别为 L_1 和 H_δ . 此外, 根据 MM 框架原理, 显然可以得知 $\hat{F}(\mathbf{w})$ 就是 $F(\mathbf{w})$ 的替代函数. 但是在这里 $\hat{L}_1(u)$ 不是光滑误差, 不满足 Lipschitz 条件, 因此不能直接套用 MM 框架来解决此问题. 然而可以通过多阶段策略, 避免 MM 框架必须要满足 Lipschitz 条件的约束, 巧妙地解决这类非凸非光滑问题.

MS-SVM 算法表明, 浮点运算已经严重影响了 L_1 正则化在随机学习和在线学习中的稀疏性^[25]. 有学者通过分别处理正则化和损失函数, 得到一个子问题的闭式解, 使这个问题得以解决^[26-27]. MS-SVM 算法也延续了这个优点, 因此使用多阶段策略同样可以继承这些优点. MS-SVM 算法是在每个阶段之前首先剔除一部分 outlier 点, 然后求解 SVM 原问题或对偶问题, 得到解 \mathbf{w}_t , 进行多个阶段的批处理直到算法收敛, 停止迭代. 这种方法能得到稀疏的支持向量, 但时间消耗较大, 因为每个阶段都必须求解一个批处理问题, 计算复杂度相当高, 不适于大规模问题求解. 其实每个阶段不需要批处理精确求解 \mathbf{w}_t , 只需要得到满足 outlier 点条件的解 \mathbf{w}_t 就可以进入下一阶段, 将 outlier 点剔除之后, 对样本进行更新, 继续进行训练. 因此可以用一种快速的方法来替代原有的批处理 SVM 方法以提升其性能.

2 非精确线搜割平面算法

在上一节定义的样本集中,式(3)可以写成如下优化问题形式:

$$\min_{\mathbf{w}} F(\mathbf{w}) = \frac{1}{2} \|\mathbf{w}\|^2 + CR(\mathbf{w}) \quad (6)$$

其中,

$$R(\mathbf{w}) = \sum_{i=1}^m L_1[y_i(\langle \mathbf{w}, \mathbf{x}_i \rangle + b)]$$

称为Hinge损失函数. 实际应用中,有时SVM会带有偏置项 b ,常用的处理技巧是将其放入权重 \mathbf{w} 中统一处理:

$$\mathbf{x}_i^T \leftarrow [\mathbf{x}_i^T, 1], \mathbf{w}^T \leftarrow [\mathbf{w}^T, b]$$

在CPA算法中原始问题式(6)被称为主问题,使用Teo et al^[21]的方法可以定义一个子问题:

$$\mathbf{w}_t = \underset{\mathbf{w}}{\operatorname{argmin}} F_t(\mathbf{w}) = \left[\frac{1}{2} \|\mathbf{w}\|^2 + CR_t(\mathbf{w}) \right] \quad (7)$$

因为 $R(\mathbf{w})$ 在 S 上为凸损失函数,若在 \mathbf{w}' 处的次梯度为 \mathbf{a}' ,则有不等式:

$$R(\mathbf{w}) \geq R(\mathbf{w}') + \langle \mathbf{w} - \mathbf{w}', \mathbf{a}' \rangle, \forall \mathbf{w} \in S$$

成立. 推广开,设 $R(\mathbf{w})$ 在点 $\mathbf{w}_1, \mathbf{w}_2, \dots, \mathbf{w}_t$ 的次梯度分别为 $\mathbf{a}_1, \mathbf{a}_2, \dots, \mathbf{a}_t$,则 $R(\mathbf{w})$ 的分段线性近似函数可表示为:

$$R_t(\mathbf{w}) = \max_{i=1,2,\dots,t} \{0, R(\mathbf{w}_i) + \langle \mathbf{w} - \mathbf{w}_i, \mathbf{a}_i \rangle\} \quad (8)$$

其中, $R(\mathbf{w}_i) + \langle \mathbf{w} - \mathbf{w}_i, \mathbf{a}_i \rangle = 0$ 被称为点 \mathbf{w}_i 处的割平面. 显然 $R_t(\mathbf{w})$ 也是凸函数,而且随着迭代次数的增加,分段线性逼近更加精确,如图2所示.

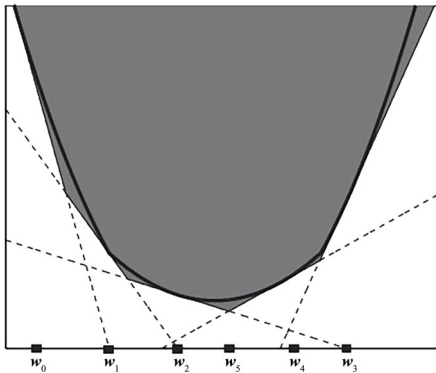


图2 凸函数的分段线性近似

Fig. 2 Piecewise linear approximation for convex function

OCAS算法在每次求得 \mathbf{w}_t 后,关键是进行一次精确线性搜索,如式(9):

$$\lambda_t = \underset{\lambda \geq 0}{\operatorname{argmin}} J((1-\lambda)\mathbf{w}_{t-1}^b + \lambda\mathbf{w}_t) \quad (9)$$

在精确线性搜索求解时需要通过排序所有的 λ_i 值来得到 λ_t 值. 容易知道,排序算法的时间复杂度为 $O(m \lg m)$. 在处理大规模机器学习问题的时候,一方面,精确线性搜索式(9)不仅受到特征维数的影响,并且关键的排序算法会受到数据规模的限制;另一方面,优化算法只是一种手段,其目的是使机器学习有更好的泛化能力,有时为了使机器学习算法获得良好的鲁棒性,无需求得模型的最优精确解. 储德军等^[24]在Franc and Sonnenburg^[22]工作的基础上(算法2),提出一种非精确线性搜索的优化割平面算法(INexact-Line-Search OCAS, INOCAS, 算法3),克服了上述缺点,并且保持了OCAS算法的优点,能够保证目标函数值 $F(\mathbf{w}_t)$ 单调下降,并且比OCAS算法的效率更高. 加速效果如图3所示.

算法2 非精确线性搜索算法

Input: $\gamma = 0.01, \lambda_{\text{new}} = 0, \lambda_{\text{old}} = 0, a_{\text{new}} = 0, a_{\text{old}} = 0, k = 0$

Repeat

$k = k + 1$

计算梯度 $[a_{\text{new}} \leftarrow \max\{\partial f(\lambda_{\text{new}})\}]$

如果 $a_{\text{new}} < 0, \lambda_{\text{new}} \leftarrow \lambda_{\text{old}} + 2^k \gamma$, 则

$\lambda_{\text{old}} \leftarrow \lambda_{\text{new}}, a_{\text{old}} \leftarrow a_{\text{new}}$

否则,根据二点二次插值方法:

$$\lambda^* \leftarrow \lambda_{\text{new}} - \frac{\lambda_{\text{new}} - \lambda_{\text{old}}}{a_{\text{new}} - a_{\text{old}}} a_{\text{new}}$$

Until: $a_{\text{new}} > 0$

Output: λ^*

算法3 基于非精确线性搜索的加速割平面算法(INOCAS)

Input: $\mathbf{w}_0, \mathbf{w}_0^* \leftarrow \mathbf{w}_0, t \leftarrow 0, \epsilon \geq 0$

Repeat

$t \leftarrow t + 1$

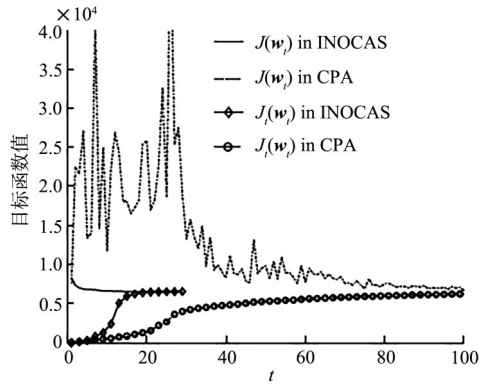
1. 求解子问题式(7), 得到 \mathbf{w}_t

2. 利用非精确线性搜索(算法2), 求得 λ_t

3. 更新 $\mathbf{w}_t^* \leftarrow (1 - \lambda_t)\mathbf{w}_{t-1}^* + \lambda_t\mathbf{w}_t$, 并在 \mathbf{w}_t^* 处添加新的割平面, 更新分段近似函数 $R_t(\mathbf{w})$

Until: $F(\mathbf{w}_t^b) - F_t(\mathbf{w}_t) \leq \epsilon$

Output: \mathbf{w}_t^*

图 3 非精确线性搜索割平面方法的加速效果^[24]Fig. 3 Speedup effect of INOCAS^[24]

多阶段策略不是一次批处理求解得到最优值,因此在每个阶段都利用批处理精确求解会降低算法的效率.非精确线性搜索能在保证单调性的条件下求得满足要求的解,同时计算复杂度很低.下一节将阐述这样求解的优点.

3 多阶段非精确线搜割平面方法

本节提出一个简单有效的多阶段非精确线性搜索的优化割平面方法(Multistage Inexact-Line-Search Ocas, MILSO)来解决关于截断 Hinge 损失的线性 SVM 问题,如算法 4 所示.

算法 4 MILSO

Input: w_0

Repeat

1. 计算 $S_t = S - O^t$

2. 利用 INOCAS 算法,在样本集 S_t 上求得 w_t

Until: $O^{t+1} = O^t$

Output: w_t

定义 2 outlier 点 满足式(10)的点称为 outlier 点.其中 O^t 代表第 t 阶段的 outlier 点:

$$O^t = \{(x_i, y_i) \in S: y_i [\langle w_t, x_i \rangle + b^t] < \delta\} \quad (10)$$

算法第一步首先要计算得到提前删去的 outlier 点;第二步对于样本集合 S_t ,利用 INOCAS 算法非精确解搜索 100 次,得到一个不是最优解但趋于最优解的解 w_t .以上为一个阶段.然后利用式(10)继续计算得到集合 O^t ,反复迭代最终得到最优解.

直观地说,多阶段程序的目的是不断从之前的支持向量集合中剔除所有当前异常值(outlier

点),并通过截断的损失函数提供更鲁棒的分类器,朝一个最佳的优化子集移动.特别的,算法 2 是一种非常直观的方法,每次迭代的时间消耗为 $O(m)$,主要用来计算梯度,因此整个算法 2 的时间复杂度为 $O(km)$,其中 k 为迭代次数, m 为样本个数.在大规模数据问题中,通常 $k \ll \lg m$,所以该算法的时间复杂度较算法 1 中的排序算法更小.对比算法 1 和算法 3 可以发现,两种算法都能保证目标函数单调下降,但算法 3 将算法 1 的精确排序问题转化为求解次梯度函数值的问题,减少了计算复杂度.在 MS-SVM 算法里,每个阶段都需要用一个批处理算法求解一个精确解,调用的 LIBSVM 算法的时间复杂度为 $O(m^3)$,远远高于算法 2.理论分析表明, MILSO 算法比 MS-SVM 算法的计算复杂度更低.下面给出算法 4 的收敛性分析.

定理 1 $\forall w_0 \in \mathbb{R}^N$,假设 w_t 是由 MILSO 算法得到,则有:

(1) $\hat{F}(w_t)$ 是单调递减的.若 $S_{t+1} \neq S_t$, 则 $\hat{F}(w_{t+1}) < \hat{F}(w_t)$.

(2) 存在一个正常数 t_0 使 $S_{t_0+1} = S_{t_0}$, 则 $S_t = S_{t_0}, \forall t \geq t_0$.

(3) 存在一个向量 $w^* \in \mathbb{R}^{N+1}$, 使 w_t 在有限步骤收敛到 w^* .

(4) w^* 是 \hat{F} 的一个局部最小点.

证明 因为算法 MILSO 在每次迭代过程中都会进行非精确线性搜索,保证目标函数值 $F(w_t)$ 序列是单调下降的(见文献[22]中的 Theorem 1),即定理 1 第(1)条得证.

定理 1 第(2)条很显然.

注意到每个原始 SVM 子问题在指定集合 S_t 上的解是唯一的,而最多存在有 2^m 种集合 S_t , 所以定理 1 第(3)条得证.

令:

$$S^* = S - \{(x_i, y_i) \in S: y_i [\langle w^*, x_i \rangle + b^*] < \delta\}$$

显然,存在一个数 $\delta > 0$,使得如果 $\|w - w^*\| < \delta_0$,那么与 w 相关的异常值包含 S^* ,即对于所有满足 $\|w - w^*\| < \delta_0$ 的点 w , 有 $\hat{F}(w^*) \leq \hat{F}(w)$,定理 1 第(4)条得证.

下面简单将本文算法与相关参考算法进行对比分析.

(1)与标准的CCCP算法^[8-9]相比较,MILSO算法不是另一种简单的寻找局部最小值的方法,除了纯凹凸优化技巧外,它应被视作一种实用的学习策略.

(2)与MS-SVM算法相比,多阶段策略能在保证稀疏性的同时,以更少的时间消耗来解决问题.

(3)多阶段优化策略已被用于解决一些非凸问题.如Zhang^[28]面对截断正则化项导致的非凸性提出一种有效的多级凸松弛策略来解决稀疏学习中的非凸问题,目标是改进文献[9-10]中求解支持向量稀疏性的方法,其中支持向量的稀疏性是由非凸损失导致的.

4 多阶段非精确线搜割平面方法

本节对本文提出的方法进行对比验证.实验采用Mac Pro工作站(2×2.8 GHz Quad-Core Intel Xeon处理器,4 GB 667 MHz DDR2内存,Mac OS X版本10.5.4).C语言编译器gcc 4.2.1.在常用的大规模数据库(表1)上进行实验.标准数据库均来自林智仁小组(<https://www.csie.ntu.edu.tw/~cjlin/libsvmtools/datasets/>).

表1 数据库描述

Table 1 Description of databases

数据集	训练集大小	测试集大小	维数
Covtype	116202	464810	54
Ijcnn1	49990	91701	22
A9a	32561	16281	123
Rcv1	20242	677399	47236

在常用的大规模数据库上,分别用三种算法对其进行训练测试,设置参数 $C=1, \delta=0$,且均不含偏置 b ,实验结果如表2至表5所示.

对比三种算法在四个数据库上的实验效果可以发现,MILSO算法得到的支持向量数是远低于传统SVM算法的,也是低于MS-SVM算法的,因此MILSO算法在稀疏性上有很好的表现.此外,CPU时间消耗也大大减少.从图4可以看出,

表2 三种算法在数据集Covtype上的线性分类结果

Table 2 Linear classification of three algorithms on the Covtype dataset

算法	阶段	支持向量数	准确度(%)	CPU时间(s)
SVM	1	257123	75.5231	3.96
MS-SVM	331	14127	77.0152	122.61
MILSO	266	11578	77.1022	76.32

表3 三种算法在数据集Ijcnn1上的线性分类结果

Table 3 Linear classification of three algorithms on the Ijcnn1 dataset

算法	阶段	支持向量数	准确度(%)	CPU时间(s)
SVM	1	21915	91.7896	0.21
MS-SVM	49	2057	93.4568	2.05
MILSO	42	1983	93.4886	1.13

表4 三种算法在数据集A9a上的线性分类结果

Table 4 Linear classification of three algorithms on the A9a dataset

算法	阶段	支持向量数	准确度(%)	CPU时间(s)
SVM	1	19793	84.9886	0.45
MS-SVM	47	1102	84.8634	1.98
MILSO	39	1078	84.9126	1.09

表5 三种算法在数据集Rcv1上的线性分类结果

Table 5 Linear classification of three algorithms on the Rcv1 dataset

算法	阶段	支持向量数	准确度(%)	CPU时间(s)
SVM	1	7017	96.1493	0.06
MS-SVM	4	6735	96.2216	0.23
MILSO	4	6653	96.2354	0.14

MILSO算法的CPU时间比MS-SVM几乎少了一半.因为MILSO算法在每个阶段采用的都是非精确线性搜索算法,计算复杂度低于传统SVM,实验还表明,在每个阶段非精确线性搜索100次就可以达到对解的要求,在多阶段策略中很快就能从当前阶段跳到下一阶段,每个阶段用

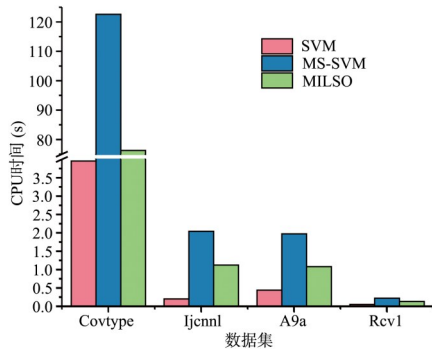


图 4 不同算法的 CPU 时间

Fig. 4 The CPU time of difference algorithms

的时间也是少于 MS-SVM 的,因而减少了 CPU 总时间,如图 5 和图 6 所示. 综上所述,在常用的大规模数据库上,和传统算法相比, MILSO 算法利用多阶段准则对损失函数进行截断,可以得到更稀疏的支持向量,并且能够大大减少时间消耗,验证了 MILSO 方法的有效性.

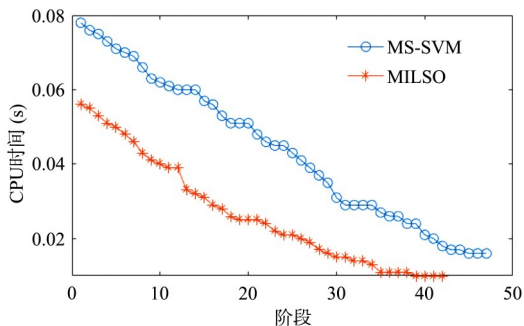


图 5 MILSO 和 MS-SVM 在数据集 A9a 上的每个阶段所用的 CPU 时间

Fig. 5 The CPU time of each stage of MILSO and MS-SVM on the A9a dataset

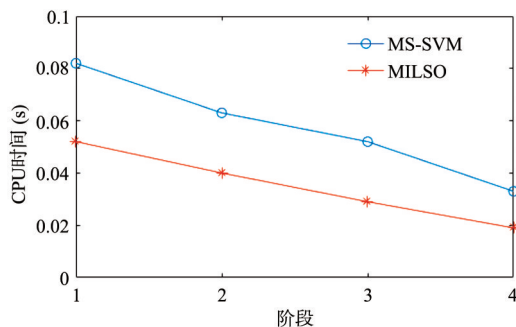


图 6 MILSO 和 MS-SVM 在数据集 Rcv1 上的每个阶段所用的 CPU 时间

Fig. 6 The CPU time of each stage of MILSO and MS-SVM on the Rcv1 dataset

SVM 在处理大规模数据时,线性增长的支持向量是限制其效率的主要原因,因此采用截断策略可以得到更稀疏的支持向量. 在处理相同的非凸截断问题时,和传统 MS-SVM 算法相比, MILSO 算法得到的支持向量数更少,并在每个阶段都使用处理效率更高的 INOCAS 算法,因此在 CPU 时间上几乎快了一倍,如前文的表 2 至表 5 所示.

为验证算法的收敛性,对比三种算法的目标函数值变化情况,如图 7 至图 10 所示,可以看出:

(1) MILSO 算法优化的目标函数在有限时间内收敛到稳定值,验证了算法具有收敛性.

(2) 对比不同数据集,多阶段准则对目标函数的优化大大优于传统的 SVM 算法,收敛时目标函数值远低于 SVM 收敛时的目标函数值.

(3) 和 MS-SVM 算法相比,由于 MILSO 算法使用了非精确线搜索,在 CPU 时间上的表现更优.

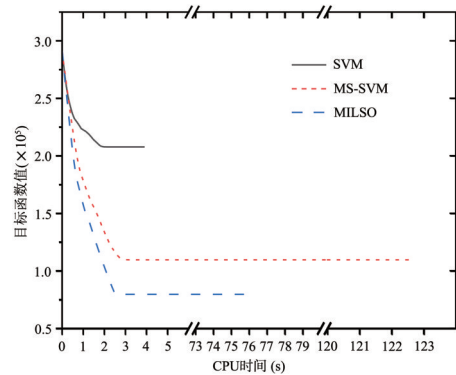


图 7 三种算法在 Covtype 上的收敛性

Fig. 7 Convergence of three algorithms on the Covtype dataset

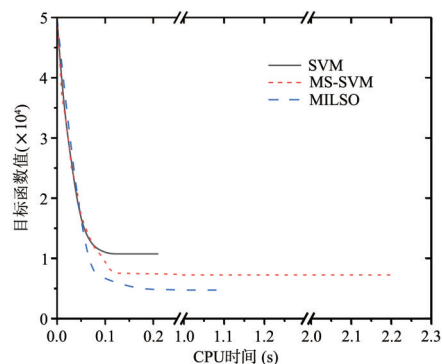


图 8 三种算法在 Ijcnn1 上的收敛性

Fig. 8 Convergence of three algorithms on the Ijcnn1 dataset

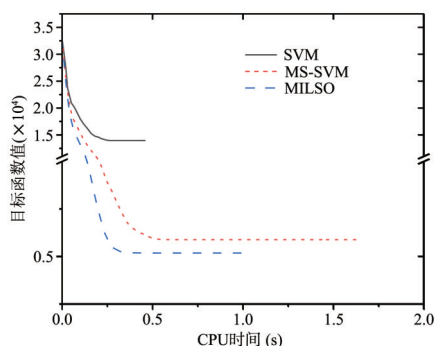


图9 三种算法在A9a上的收敛性

Fig. 9 Convergence of three algorithms on the A9a dataset

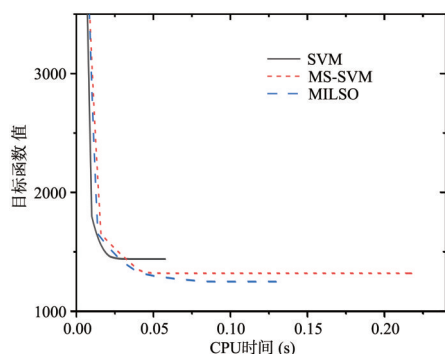


图10 三种算法在Rcv1上的收敛性

Fig. 10 Convergence of three algorithms on the Rcv1 dataset

5 总结

本文提出一种基于非精确线性搜索的多阶段策略MILSO,继承了非精确线性搜索的优点.与MS-SVM算法里的每个阶段都要调用批处理方法相比较,MILSO算法在每个阶段进行非精确求解,能迅速进入下一阶段,计算复杂度低,且能保证目标函数的单调性,保证了模型的稳定性.另一方面,多阶段策略在每个阶段剔除一部分 outlier点,得到了稀疏的支持向量.理论分析该方法具有收敛性,实验也证明MILSO算法的性能优于MS-SVM算法.下一步的主要工作是将此方法扩展到随机或增量形式.

参考文献

- [1] Vapnik V. The nature of statistical learning theory. New York:Springer,1995,314.
- [2] Kelley J E. The cutting plane method for solving convex problems. Journal of the Society for Industrial

&. Applied Mathematics,1960,8(4):703—712.

- [3] Chang K W, Hsieh C J, Lin C J. Coordinate descent method for large-scale L2-loss linear support vector machines. Journal of Machine Learning Research, 2008,9(3):1369—1398.
- [4] Hastie T, Zhu J. Comment on "Support vector machines with applications". Statistical Science, 2006,21(3):352—357.
- [5] Zhang T. Statistical behavior and consistency of classification methods based on convex risk minimization. The Annals of Statistics, 2004, 32(1): 56—85.
- [6] Cheung Y M, Lou J. Efficient generalized conditional gradient with gradient sliding for composite optimization//Proceedings of the 24th International Conference on Artificial Intelligence. Buenos Aires, Argentina: AAAI Press, 2015: 3409—3415.
- [7] Steinwart I. Sparseness of support vector machines. Journal of Machine Learning Research, 2003, 4(6): 1071—1105.
- [8] Collobert R, Sinz F, Weston J, et al. Trading convexity for scalability//Proceedings of the 23rd International Conference on Machine Learning. Pittsburgh, PA, USA: ACM, 2006: 201—208.
- [9] Liu S J, Shen X T, Wong W H. Computational developments of ψ -learning//Proceedings of the 5th SIAM International Conference on Data Mining. Newport Beach, CA, USA: Society for Industrial and Applied Mathematics, 2005: 1—11.
- [10] Wu Y C, Liu Y F. Robust truncated hinge loss support vector machines. Journal of the American Statistical Association, 2007, 102(479): 974—983.
- [11] An L T H, Tao P D. Solving a class of linearly constrained indefinite quadratic problems by D. C. algorithms. Journal of Global Optimization, 1997, 11 (3): 253—285.
- [12] Yuille A L, Rangarajan A. The concave - convex procedure. Neural computation, 2003, 15(4): 915—936.
- [13] Mairal J. Stochastic majorization - minimization algorithms for large-scale optimization//Proceedings of the 26th International Conference on Neural Information Processing Systems. Lake Tahoe, NV, USA: Curran Associates Inc., 2013: 2283—2291.

- [14] Mairal J. Incremental majorization - minimization optimization with application to large-scale machine learning. *SIAM Journal on Optimization*, 2015, 25(2):829—855.
- [15] Fang C, Li C J, Lin Z C, et al. Spider: near-optimal non - convex optimization via stochastic path integrated differential estimator. 2018, arXiv: 1807.01695.
- [16] Carmon Y, Duchi J C, Hinder O, et al. Accelerated methods for nonconvex optimization. *SIAM Journal on Optimization*, 2018, 28(2):1751—1772.
- [17] Lan G H, Yang Y. Accelerated stochastic algorithms for nonconvex finite - sum and multi - block optimization. 2018, arXiv:1805.05411.
- [18] Tao Q, Wu G W, Chu D J. Improving sparsity and scalability in regularized nonconvex truncated - loss learning problems. *IEEE Transactions on Neural Networks and Learning Systems*, 2018, 29(7):2782—2793.
- [19] Shalev - Shwartz S, Singer Y, Srebro N, et al. Pegasos: primal estimated sub - gradient solver for SVM. *Mathematical Programming*, 2011, 127(1): 3—30.
- [20] Joachims T. Training linear SVMs in linear time// *Proceedings of the 12th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*. Philadelphia, PA, USA: ACM, 2006: 217—226.
- [21] Teo C H, Smola A, Vishwanathan S V N, et al. A scalable modular convex solver for regularized risk minimization//*Proceedings of the 13th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*. San Jose, CA, USA: ACM, 2007:727—736.
- [22] Franc V, Sonnenburg S. Optimized cutting plane algorithm for support vector machines//*Proceedings of the 25th International Conference on Machine Learning*. Helsinki, Finland: ACM, 2008:320—327.
- [23] Chu D J, Zhang C S, Tao Q. A faster cutting plane algorithm with accelerated line search for linear SVM. *Pattern Recognition*, 2017, 67:127—138.
- [24] 储德军, 陶安, 高乾坤等. 求解线性 SVM 的非精确步长搜索割平面方法. *模式识别与人工智能*, 2014, 27(8):692—700. (Chu D J, Tao A, Gao Q K, et al. Optimized cutting plane method for linear SVM via inexact step - length search. *Pattern Recognition and Artificial Intelligence*, 2014, 27(8):692—700.)
- [25] Langford J, Li L H, Zhang T. Sparse online learning via truncated gradient. *Journal of Machine Learning Research*, 2009, 10:777—801.
- [26] Duchi J C, Shalev - Shwartz S, Singer Y, et al. Composite objective mirror descent//23rd *International Conference on Learning Theory*. Haifa, Israel: COLT, 2010:14—26.
- [27] Xiao L. Dual averaging methods for regularized stochastic learning and online optimization. *The Journal of Machine Learning Research*, 2010, 11: 2543—2596.
- [28] Zhang T. Analysis of multi-stage convex relaxation for sparse regularization. *Journal of Machine Learning Research*, 2010, 11:1081—1107.

(责任编辑 杨可盛)