

支持向量机中优化算法^{*}

宋晓峰 陈德钊 俞欢军 胡上序

(浙江大学智能信息工程研究所 杭州 310027)

A Survey of Optimization Algorithms in Support Vector Machine

SONG Xiao-Feng CHEN De-Zhao YU Huan-Jun, HU Shang-Xu

(Laboratory of Intelligent Information Engineering, Zhejiang University, Hangzhou 310027, China)

Abstract Optimization algorithm solving Lagrangian multipliers is the key of training SVM, determining the performance of SVM, affecting practical applications of SVM in various fields widely. Some kinds of optimization algorithms in SVM of overseas are introduced. We classify the optimization algorithms into two kinds: 1. the algorithms based on Osuna's decomposition strategy; 2. The iterative algorithms based on the changes of SVM formulation proposed by O. L. Mangasarian. We also analyze the characteristics of various optimization algorithms in SVM, and predicting the trend of research on optimization algorithm in SVM.

Keywords Support vector machine, Optimization algorithms

1 引言

对基于数据的机器学习研究是目前计算智能技术的基础性工作之一,旨在从观测数据出发基于统计学习理论采用相关技术对研究对象建立可信的模型,利用这一模型对未来实验作出指导性的预测。相关技术包括聚类分析、模式识别、神经网络等等。现有的学习方法多是基于样本数趋于无穷大时的渐进理论。可是在实际科研工作中,有时候很难获得较为充足的实验数据,为了充分利用有限的样本数据提供的信息,就需要利用由 Vapnik 等人提出的支持向量机(Support Vector Machine, SVM),它的理论基础是基于小样本的统计学习理论(Statistical Learning Theory, SLT),由于 SVM 比神经网络具有较好的泛化推广能力,消除了神经网络的过学习现象,在国际上成为计算智能技术研究的又一新的研究热点。

SVM 可广泛应用于文本识别、人脸检测、三维图像识别、非线性回归建模、数据压缩、时间序列预测、生物信息学等各个方面,但所见应用研究的报道较少。对 SVM 中核心算法——求解拉格朗日乘子的优化算法,若采用经典的二次规划方法,一般无法解决实际问题,尤其是大规模问题,这直接影响着 SVM 的实际应用。因此,近年来,国际上对 SVM 中优化算法的研究非常重视,以期加深和扩大 SVM 应用领域的应用。本文将对国际上所开发的 SVM 中优化算法作一详细而系统的介绍,以引起国内众多研究者的注意,尽快将这一机器学习方法应用到各个领域。

2 SVM 简介及基本优化算法

SVM 最初来自对数据分类问题的处理,以两值线性可分情况为例,假定已知观测数据,

$$(x_1, y_1), \dots, (x_l, y_l), x \in R^n, y \in \{+1, -1\}$$

可以被一个超平面 $(w \cdot x) - b = 0$ 分开, SVM 寻求的是这样

一个分割平面,使得训练数据集中的点距离这一分类面尽量地远,亦即使其两侧的空白区域尽量地大。数学描述如下。在满足式(1)

$$y_i[(w \cdot x_i) - b] \geq 1, i = 1, \dots, l \quad (1)$$

的条件下,求解下面的二次规划问题:

$$\Phi(w) = \frac{1}{2}(w \cdot w) \quad (2)$$

这个优化问题的解可通过求解拉格朗日函数的鞍点得出,利用对偶性原理,经过推导转化为求解如下泛函的优化问题:

$$\begin{aligned} \max W(a) &= \sum_{i=1}^l a_i - \frac{1}{2} \sum_{i,j=1}^l a_i a_j y_i y_j (x_i \cdot x_j) \\ s.t. \quad a_i &\geq 0, i = 1, \dots, l; \\ \sum_{i=1}^l a_i y_i &= 0; \end{aligned} \quad (3)$$

设 $a_0 = (a_1^0, \dots, a_l^0)$ 为上面二次优化问题的解,则最优超平面中的向量 w_0 的模为:

$$\|w_0\|^2 = 2W(a_0) = \sum_{i,j=1}^l a_i^0 a_j^0 (x_i \cdot x_j) y_i y_j \quad (4)$$

最后得到的分类规则函数为:

$$f(x) = \text{sgn} \left(\sum_{i=1}^l y_i a_i^0 (x_i \cdot x) - b_0 \right) \quad (5)$$

其中 x_i 是支持向量, a_i^0 是对应的拉格朗日系数, b_0 是常数(可由 w_0 和两类中任一支持向量求得)。

在线性不可分的情况下,在式(1)中增加一松弛变量 $\zeta_i \geq 0$,成为:

$$y_i[(w \cdot x_i) - b] - 1 + \zeta_i \geq 0, i = 1, \dots, l. \quad (6)$$

综合考虑了最小化错分样本数和最优推广能力,则目标函数改为:

$$\Phi(w, \zeta) = \frac{1}{2}(w \cdot w) + c \left(\sum_{i=1}^l \zeta_i \right) \quad (7)$$

其中 $c > 0$ 为一常数,它控制对错分样本的惩罚程度,则求解

^{*}基金项目:国家自然科学基金资助项目(编号:20076041)。宋晓峰 博士研究生,从事建模与优化、信息智能处理等研究。陈德钊 教授,博导,从事神经网络、信息智能处理等研究。俞欢军 副教授,博士,从事计算机网络及应用、软件工程等研究。胡上序 教授,博导,从事过程仿真建模及优化、信息智能处理等研究。

在两值线性不可分情况下的广义最优分类面,就是求解在(6)式约束条件下,式(7)的二次优化问题。最终转化为与两值线性可分情况下相同的问题,即根据求拉格朗日函数的鞍点转化为求解原问题的对偶问题(3)式,只是约束条件 $a_i \geq 0, i = 1, \dots, l$; 改为 $0 \leq a_i \leq c, i = 1, \dots, l$ 。

对于非线性的分类问题,通过引入核函数,将原空间的数据通过非线性变换映射到一高维特征空间,在这一空间求解广义线性最优分类面,定义不同的核函数,则对应不同的算法,较为常用的核函数为:多项式核函数 $K(x, x_i) = [(x \cdot x_i) + 1]^d$, 径向基核函数 $K(x, x_i) = \exp(-\frac{\|x - x_i\|^2}{\sigma^2})$, Sigmoid 核函数 $K(x, x_i) = S[v(x \cdot x_i) + c]$ 。

因此,在广义最优分类面求解中引入核函数 $K(x, x_i)$ 就可以实现某一非线性变换后的线性分类,上面几种情况可统一为求解如下的二次优化问题:

$$\begin{aligned} \max \quad & W(a) = \sum_{i=1}^l a_i - \frac{1}{2} \sum_{i,j=1}^l a_i a_j y_i y_j K(x_i, x_j) \\ \text{s.t.} \quad & 0 \leq a_i \leq c, i = 1, \dots, l; \\ & \sum_{i=1}^l a_i y_i = 0; \end{aligned} \quad (8)$$

其中 a_i 为与每个样本对应的拉格朗日乘子,式(8)存在唯一解,其解中只有一少部分的 a_i 不为零,其对应的样本就是支持向量。则其广义最优分类面决策函数为:

$$f(x) = \text{sgn}(\sum_{s \in V} y_s a_s K(x, x_s) + b) \quad (9)$$

可见 SVM 中最核心、关键的问题也就是求取拉格朗日乘子的二次规划问题,当样本数量不多时,应用传统的二次规划方法(主要有内点法、有效集法、Wolfe 算法、拉格朗日鞍点法、应用 K-T 条件的线性规划法等等)可很方便地求得(8)式,可是在样本数比较多时,求解比较麻烦,计算资源占用较多,这也是 SVM 难以得到广泛应用的原因之一。为了寻求更为有效的算法,人们研究了多种算法来进一步改善 SVM 学习算法,这些算法主要包括早期的分块算法(Chunking Algorithm)、分解算法(Decomposition Algorithm)以及近期由 Joachim 等人提出的 SVMlight 算法和 John C. Platt 开发的 SMO 算法(Sequential Minimal Optimization)、O. L. Mangasarian 提出的 SOR 算法(Successive Overrelaxation)、LSVM 算法(Lagrangian Support Vector Machine)、ASVM 算法(Active Set Strategy Support Vector Machine)和 SSVM 算法(Smooth Support Vector Machine)。这些算法的提出进一步推动了支持向量机的应用,使得以前无法解决的问题得以有效解决。

3 SVM 优化算法的理论基础

3.1 Kuhn-Tucker 最优性必要条件

定理 1 设 x^* 为非线性规划问题 $(\min f(x) \text{ s.t. } g_i(x) \geq 0; h_j(x) = 0)$ 的最优解,如果在 x^* 处梯度向量 $\nabla g_i(x^*)$ 和 $\nabla h_j(x^*)$ 线性无关,则存在向量 λ^*, μ^* 使下述条件成立:

$$\nabla f(x^*) - \sum_{i=1}^m \lambda_i^* \nabla g_i(x^*) - \sum_{j=1}^p \mu_j^* \nabla h_j(x^*) = 0 \quad (10)$$

式(10)通常称为 Kuhn-Tucker 最优性条件,简称 Kuhn-Tucker 条件(K-T 条件),有时也称为 Karush-Kuhn-Tucker 条件(K-K-T 条件)。满足这些条件的点称为 K-T 点。

在一般非线性规划问题中,K-T 条件不是最优解的充分条件,即 K-T 点不一定是最优点。

但对于凸规划问题,其存在唯一最优点,所以 K-T 点一定是最优点。

3.2 Sherman-Morrison-Woodbury 矩阵求逆恒等式

定理 2 如果矩阵 $A \in R^{n \times n}, U, V \in R^{n \times k}$, 并且 A 和 $I + V' A^{-1} U$ 均为正则阵,则存在下列恒等式,称为 Sherman-Morrison-Woodbury 恒等式(SMW 恒等式):

$$(A + UV')^{-1} = A^{-1} - A^{-1} U (1 + V' A^{-1} U)^{-1} V' A^{-1} \quad (11)$$

当 $A = I/c, U = H, V = H'$ 时, I 为 $n \times n$ 实单位阵, c 为正常数, H 为 $n \times k$ 矩阵,则有如下特例成立:

$$(\frac{I}{c} + HH')^{-1} = c(I - H(\frac{I}{c} + H'H)^{-1}H') \quad (12)$$

在 $n \gg k$ 时,那么求解 $n \times n$ 大规模矩阵的逆阵可转化为求解一小规模 $k \times k$ 矩阵的逆阵。

4 SVM 发展初期常用优化算法

在支持向量机技术出现初期,求解拉格朗日乘子的优化算法除了经典的二次规划方法以外,为了解决大规模问题,人们还提出了如下算法。

最初为了解决较大规模的 SVM, Vapnik 采用了将求解矩阵分块的策略,我们知道将矩阵中对应拉格朗日乘子为零的行与列消去,所得矩阵仍为二次型且目标函数值与原矩阵对应的目标函数值相等,由于最终目的是确定那些非零的拉格朗日乘子,因此大型的 QP 问题可化为一系列的小规模的 QP 问题。算法在每步求解的子 QP 问题所包含的样本为上一步中具有非零的拉格朗日乘子所对应的样本以及不满足 K-T 条件的最差 M(可事先确定)个样本(若违反 K-T 条件的样本不足 M 个则全部加入)。算法的每步均以上一步的结果作为初始值进行计算,这样一步步经过若干次的子 QP 问题的求解,最终得到问题的解——所有非零的拉格朗日乘子。分块算法只是降低了求解问题目标函数的 Hessian 阵的规模,仍需使用常规二次规划算法,经常使用的是共轭投影梯度法(PCG 算法)。

在 1997 年 Osuna 等人提出了解决 SVM 中二次规划问题的一种全新的策略——分解算法,并且首次在理论上正式证明了解析算法的收敛性,该定理证明,如果存在不满足 K-T 条件的样本点,那么在把它对应的拉格朗日乘子加入到上一个子问题的集合中后,重新优化这个子问题,则可行点(feasible point)依然满足约束条件,且性能严格地改进。因此,一系列的子 QP 问题(每步至少加入一个不满足 K-T 条件的样本对应的拉格朗日乘子)可保证实现收敛。

分解算法与分块算法类似,都是将矩阵分块计算,只是分解后每一步求解子 QP 问题的 Hessian 阵的大小保持不变。其基本思想是:将 SVM 中大型的 QP 问题化为一系列的求解规模一定(问题的 Hessian 阵保持不变)的子 QP 问题,每一步都要保持问题满足 K-T 条件。下一步将未在上一步子 QP 问题中的不满足 K-T 条件的样本点对应的拉格朗日乘子引入上一步子 QP 问题中同时去掉一个拉格朗日乘子,构成新一步的子 QP 问题。使得目标函数值一步步下降,直至最后所有的拉格朗日乘子都满足 K-T 条件,就得到问题的最优解(对于凸二次规划问题满足 K-T 条件的点一定是最优点)。

5 国际上近期开发的 SVM 常用优化算法

5.1 SVMlight 算法

1998 年由 Joachim 等人提出的求解大规模支持向量机

的 SVMlight 算法^[5],是 Osuna 分解算法的一种推广,其基本思想是,将所求问题的拉格朗日乘子分为活动集和非活动集两部分:如果存在不满足 K-T 条件的样本,则基于某种方式选择 $q(q < l)$ 个样本对应的拉格朗日乘子作为活动集(称为工作集合),其它的样本所对应的拉格朗日乘子保持不变作为非活动集合,在这个工作集合上解决 QP 问题。重复这一过程,直至所有拉格朗日乘子都满足 K-T 条件。

SVMlight 选择工作集合的策略是基于 Zoutendijk 可行方向法,即选择一个最速下降可行方向来得到本次迭代所需的 q 个拉格朗日乘子,亦即求解如下一个优化问题,

$$\begin{aligned} \min \quad & g(a^{(i)})'d \\ \text{s.t.} \quad & y'd = 0 \\ & d_i \geq 0 \text{ for } a_i = 0 \\ & d_i \leq 0 \text{ for } a_i = c \\ & -1 \leq d_i \leq 1 \\ & |\{d_i, d_i \neq 0\}| = q \end{aligned} \quad (13)$$

其中 $g(a^{(i)})$ 为(8)式目标函数在 $a^{(i)}$ 点的一阶偏导数, d 为所求的可行方向。前三个约束条件确定了可行方向的投影在式(8)等式约束上,第四个约束使得可行方向向量标准化,最后一个约束要求所求得的可行方向只包含 q 个拉格朗日乘子。在求解过程中并不需把 d 求出,只需把其所对应的 q 个拉格朗日乘子求出即可。

此外, SVMlight 算法还运用了收缩策略(Shrinking strategy),随着优化的进行,我们可以提前知道有些样本点最终不会成为支持向量,因此将这些样本点可从求解问题中剔除,使得计算量一步步减少。

工作集的选择策略、最优性 K-T 条件、Hessian 阵分解策略以及收缩策略的运用就构成了 SVMlight 算法的主体框架。

5.2 SMO 算法(Sequential Minimal Optimization)

由 MIT 的 John C. Platt 等人在 1998 年提出的 SMO 算法^[3],是目前快速求解 SVM 中 QP 问题的一个较好的方法,它不需要通常的 QP 求解模块嵌入其中。SMO 算法是将问题分解为一系列最小规模的 QP 问题,即每次迭代步只处理含两个拉格朗日乘子的 QP 问题,是 Osuna 算法的一种特殊情况。

通过(8)式我们可以看到 SVM 中的二次规划问题是一类比较特殊的二次规划问题,其约束一个是超立方体约束,一个是超线性约束。因此,从约束入手来求解含两个拉格朗日乘子的 QP 问题。我们知道拉格朗日乘子 a_1, a_2 被定义在 $(0, c)$ 的二维正方形空间中,同时又在 $a_1 \pm a_2 = \gamma$ 所确定的对角线上,根据这一特点,我们可得到 a_1, a_2 的求解范围。不失一般性, a_2 的范围为:

$$\begin{aligned} \text{当 } y_1 \neq y_2 \text{ 时} \\ L = \max(0, a_2^{old} - a_1^{old}), H = \min(c, c + a_2^{old} - a_1^{old}) \end{aligned} \quad (14)$$

$$\begin{aligned} \text{当 } y_1 = y_2 \text{ 时} \\ L = \max(0, a_1^{old} + a_2^{old} - c), H = \min(c, a_1^{old} + a_2^{old}) \end{aligned} \quad (15)$$

根据以上约束,将目标函数化为 a_2 的函数:

$$\begin{aligned} W(a) = \frac{1}{2} (2k(x_1, x_2) - k(x_1, x_1) - k(x_2, x_2))a_2^2 + (y_2) \\ (E_1^{old} - E_2^{old}) - \eta a_2^{old} a_2 + const \end{aligned} \quad (16)$$

令 $\eta = 2k(x_1, x_2) - k(x_1, x_1) - k(x_2, x_2)$, 则关于 a_2 的一次和二次导数分别为:

$$\frac{dW}{da_2} = \eta a_2 + (y_2(E_1^{old} - E_2^{old}) - \eta a_2^{old}) \quad (17)$$

$$\frac{d^2W}{da_2^2} = \eta \quad (18)$$

令一阶导数为零,则得本次求解问题的最优值,即:

$$a_2^{new} = a_2^{old} - \frac{y_2(E_1 - E_2)}{\eta}$$

其中 $E_i = f^{old}(x_i) - y_i$ 为训练误差。

那么,在式(14)、(15)约束下,则:

$$a_2^{new, clipped} = \begin{cases} H, & \text{if } a_2^{new} \geq H \\ a_2^{new}, & \text{if } L < a_2^{new} < H \\ L, & \text{if } a_2^{new} \leq L \end{cases} \quad (19)$$

令 $s = y_1 y_2$, 则:

$$a_1^{new} = a_1^{old} + s(a_2^{old} - a_2^{new, clipped}) \quad (20)$$

至此,得到了本次迭代的满足 K-T 条件的两个拉格朗日乘子。

因此, SMO 算法步骤如下:

1. 给定满足超立方体约束和超线性约束的拉格朗日乘子初值,一般取 $a_i = 0$ 。
2. 应用启发式方法循环计算整个训练样本的 K-T 条件,找到违反 K-T 条件的样本点对应的拉格朗日乘子,将其作为两个拟优化的拉格朗日乘子之一;
3. 第二个拉格朗日乘子的挑选根据最大优化步数来定,即满足 $\max |E_1 - E_2|$ 的样本点对应的拉格朗日乘子。至此,拉格朗日乘子 a_1, a_2 挑选完毕,在保持其余拉格朗日乘子不变情况下,形成一最小规模的二次规划问题(只含两个拉格朗日乘子)。
4. 是否找到违反 K-T 条件的拉格朗日乘子,是,进行下一步;否,则得到整个问题的最优解,转 6。
5. 求解上述最小化规模的二次优化问题,得到一对新的 a_1^{new}, a_2^{new} , 和其余的拉格朗日乘子一起均满足 K-T 条件。返回 2。
6. 根据式(9)得到最优分类规则函数。

5.3 SOR 算法(Successive overrelaxation)

1999 年由美国威斯康星大学 O. L. Mangasarian 等人提出了 Successive overrelaxation(SOR)算法^[4]。对于一个非线性分类问题,我们知道求解式(7)的二次优化问题,写成矩阵形式则为:

$$\begin{aligned} \min \quad & ce'\zeta + \frac{1}{2} \omega' \omega \\ \text{s.t.} \quad & D(A\omega - eb) + \zeta \geq e \\ & \zeta \geq 0 \end{aligned} \quad (21)$$

式中 c 为一正的惩罚系数, e 为一单位列向量, ξ 为松弛变量, A 为由输入向量形成的 $1 \times n$ 矩阵; D 为由输出变量形成的 1×1 对角阵($D_{ii} = 1$ 或 -1 , 其余为 0)。

通常是求其对偶形式,

$$\begin{aligned} \max \quad & -\frac{1}{2} a' D A A' D a + e' a \\ \text{s.t.} \quad & e' D a = 0 \\ & 0 \leq a \leq ce \end{aligned} \quad (22)$$

O. L. Mangasarian 等人将原在 n 维空间求解分类超平面的问题化为在 $n+1$ 维空间进行,即将目标函数式(21)改为:

$$\min ce'\zeta + \frac{1}{2} (\omega' \omega + b^2) \quad (23)$$

其对偶形式则为:

$$\begin{aligned} \max \quad & -\frac{1}{2} a' D A A' D a - \frac{1}{2} a' D e e' D + e' a \\ \text{s.t.} \quad & 0 \leq a \leq ce \\ & (\omega = A' D a, b = -e' D a, \zeta = \max(0, (e - D(A\omega - eb)))) \end{aligned} \quad (24)$$

这样求解式(24)的优化问题中没有了超线性约束,只有一超盒约束,使得我们很方便地直接采用有效的矩阵分解策略,避免了在每次迭代中处理等式约束。这一思想源于 O. L. Mangasarian 等人在处理对称线性互补问题的方法,而且还证明了 SOR 方法的线性收敛性。

令 $H = D[A - e]$, 则 HH' 为一正定对称阵,可分解为 $HH' = L + E + L'$ 。将式(24)转化为求一极小问题,即

$$\min \frac{1}{2} \|H'a\|_2 - e'a \quad (25)$$

$$s. t. \quad 0 \leq a \leq ce$$

我们定义 $\text{box}_L^U(x)$ 函数为:

$$\text{box}_L^U(x) = \begin{cases} L & \text{if } x \leq L \\ x & \text{if } L \leq x \leq U \\ U & \text{if } x \geq U \end{cases} \quad (26)$$

则其最优性条件为:

$$a = \text{box}_0^c(a - rE^{-1}(HH'a - e)) \quad (27)$$

其中 r 为一大于 0 的系数,一般取 $0 < r < 2$ 。

SOR 算法是一矩阵分解迭代过程,每一步均需满足最优性条件(27),经推导^[5],可得下列 SOR 迭代算式,给定初始值 a^0 ,由 a' 计算 a^{+1} ,即:

$$a^{+1} = \text{box}_0^c(a' - rE^{-1}(HH'a' - e + L(a'^{+1} - a'))) \quad (28)$$

式中 $r \in (0, 2)$, $c > 0$ 均需事先设定, L 为一严格下三角阵,在式中作为替代算子,在计算 a_i^{+1} 时可用 $(a_1^{+1}, \dots, a_{i-1}^{+1})$ 替换 $(a_1', \dots, a_{i-1}')$,充分利用已计算出的最新值,加快收敛速度。

5.4 LSVM 算法(Lagrangian support vector machine)

最近, O. L. Mangasarian 等人又提出了具有线性收敛性的 LSVM 算法^[6],这一算法是源于 O. L. Mangasarian 等人在 1993 年所提出的隐拉格朗日公式 (implicit Lagrangian formulation)。

与 SOR 算法类似,同样是对原有的 SVM 优化算式(21)进行改进,注意到如果将式(21)中松弛变量 ξ 的 1 范数改为 2 范数,则式(21)中的约束 $\xi > 0$ 这个条件就可省略,我们同样将 b^2 加入 $\omega'\omega$ 中,则得到如下优化算式:

$$\min c \frac{\xi\xi}{2} + \frac{1}{2} (\omega'\omega + b^2) \quad (29)$$

$$s. t. \quad D(A\omega - eb) + \xi \geq e$$

其对偶问题则为:

$$\min \frac{1}{2} a' \left(\frac{I}{c} + D(AA' + ee')D \right) a - e'a \quad (30)$$

$$s. t. \quad 0 \leq a \in R^n$$

原问题的 ω, ξ, b 可由如下算式求得:

$$\omega = A'Da, \xi = \frac{a}{c}, b = -e'Da \quad (31)$$

如果令

$$H = D[A - e], Q = \frac{I}{c} + HH' \quad (32)$$

则式(30)变为,

$$\min \frac{1}{2} a' Q a - e'a \quad (33)$$

$$s. t. \quad 0 \leq a \in R^n$$

因此其最优解条件为:

$$Qa - e = \max(0, ((Qa - e) - ra)) \quad (34)$$

由此可导出如下 LSVM 算法迭代公式:

$$a^{+1} = Q^{-1}(\max(0, (e + ((Qa' - e) - ra')))), i = 0, 1, \dots, \quad (35)$$

可见它不需要采用通常的线性规划或二次规划算法,其

主要计算量在于求解一矩阵的逆,而且矩阵求逆可利用矩阵理论中的 SMW 恒等式,转化为一小规模的矩阵求逆。算法简单明了,实现起来非常简单。

5.5 其它优化算法

除了上述一些目前常用的 SVM 主要优化算法外,国外研究人员在此基础上又开发了一些优化算法,主要有 O. L. Mangasarian 等人在 SOR 算法和 LSVM 算法的基础上又相继提出的 ASVM 算法(Active set support vector machine)和 SSVM 算法(Smooth support vector machine),有关 ASVM, SSVM 具体算法可参见文[7, 8],还有一些科研人员对前述算法做了进一步的改进,如中国台湾大学的 Chih-Jen Lin 等人将 SMO 算法结合 SVMlight 算法中的工作集的选择策略,开发的 LIBSVM 无论运算速度还是效果上都较好^[13]。目前,鉴于求解拉格朗日乘子的优化算法在支持向量机应用中的重要性,各国科研人员对支持向量机中优化算法的研究与开发仍热情不减,相信今后还会出现在速度和实用上都更加完善的算法。

5.6 各种优化算法的分析比较

通过上面的介绍,我们可以看到目前训练 SVM 的主流算法遵循两条技术路线。一是从 Osuna 的分解策略出发,提出的 Decomposition 算法、SVMlight 算法和 SMO 算法,均是通过求解一系列小规模子 QP 问题来一步步实现对原有问题的解决,只是分解的规模大小和选择工作集的策略有所不同。在这三种算法中工作集的选择是算法实现的关键,关系到算法收敛速度的快慢,若工作集选择不好,算法在迭代过程中还有可能产生循环现象,算法无法收敛。目前在这类算法中,工作集选择策略的研究正成为国内外研究的热点。

二是 O. L. Mangasarian 等人提出的一系列算法,其均是对标准 SVM 进行改造。首先构造最大化分类间隔时不仅仅考虑分类面的方向向量 ω ,还要考虑确定分类面位置的阈值向量 b 。其次在最小化软间隔分类误差时,对松弛变量 ξ 使用 2 范数代替传统的 1 范数。经过这样的改造,使得原有 SVM 优化问题的对偶问题成为一类特殊的二次规划问题,通过利用凸函数的最优性 K-T 条件,得到一求解拉格朗日乘子的迭代公式。同时利用 SMW 恒等式降低问题求解规模。从改造标准 SVM 入手,同时引入二次规划的一些方法,是这类算法的一大特点:将求解对称线性互补问题和二次规划问题的序列超松弛法引入改造后的 SVM,构成了求解 SVM 问题的 SOR 算法;将求解二次规划对偶问题的隐拉格朗日公式引入改造后的 SVM,构成了求解 SVM 问题 LSVM 算法;将有效集方法(Active set methods)引入改造后的 SVM,构成了求解 SVM 问题的 ASVM 算法;将光滑函数方法引入到改造后的 SVM,就构成了求解 SVM 问题的 SSVM 算法。

结束语 优化算法是 SVM 训练中的关键算法,其算法性能的优劣直接影响到 SVM 训练的效果,是影响 SVM 实际应用的主要因素。因此,目前关于 SVM 中优化算法的研究正成为国际上研究 SVM 的热点问题。本文对近期国际上所开发的 SVM 优化算法做了较详细、完整的介绍。这两大类优化算法中,基于 Osuna 分解策略的一类算法以 SMO 算法为其代表,由于 SMO 算法内存资源占用极少,算法简单,易于实施,且比通常 SVM 优化算法速度快,可处理大规模问题(尤其对稀疏阵)。因此,目前 SVM 在工程实际应用中采用 SMO 算法较多,SMO 算法正成为 SVM 中的一种通用算法。但

(下转第 20 页)

- Useful for Computer Supported Coordination. ACM Trans. on Office Information Systems, 1987
- 2 Chess D, Harrison C, Kershbaum A. Mobile Agents: Are They a Good Idea? [IBM Research Report RC 19887(88465)]. December 1994
 - 3 Gray R S. Agent Tcl: A flexible and secure mobile-agent system. In: Proc. of the 1996 Tcl/Tk Workshop, July 1996
 - 4 Wong D, et al. Concordia: An Infrastructure for Collaborating Mobile Agents. Mitsubishi Electric ITA, Horizon Systems Laboratory, 1997
 - 5 Peine H, Stolpmann T. The Architecture of the Ara Platform for Mobile Agents. First Intl. Workshop on Mobile Agents, 1997
 - 6 Cabri G, Leonardi L, Zambonelli F. How to Coordinate Internet Applications based on Mobile Agents. IEEE Seventh Intl. Workshops on Enabling Technologies: Infrastructure for Collaborative Enterprises (WETICE), June 1998
 - 7 Adnan S, Datuin J, Yalamanchili P. A Survey of Mobile Agent Systems. CSE 221 Final Project, June 13, 2000
 - 8 The Mobile Agent List. <http://mole.informatik.uni-stuttgart.de/mal/preview/preview.html>
 - 9 吕建, 陶先平, 董恒, 李新, 张冠群. 移动 agent 系统 Mogent1.0 的设计和实现: [南京大学计算机软件新技术国家重点实验室技术报告]. 1998
 - 10 柳杨. 基于 Internet 的 E-mail Agent 中间件系统: [南京大学硕士学位论文]. 1999
 - 11 陶先平, 吕建等. 流动 agent: 一种未来的分布计算模式. 计算机科学, 1999
 - 12 陶先平, 吕建, 张冠群, 李新, 董恒. 一种新的移动 agent 结构化迁移机制的设计和实现. 软件学报, 2000
 - 13 马勤, 陶先平, 吕建. 一种新的 E-mail 应用模式的研究. 计算机科学, 2001
 - 14 马勤, 张小蕾, 陶先平, 等. 基于 Internet 的 MOON-EAMS 中间件系统: [南京大学计算机软件新技术国家重点实验室技术报告]. 2001
 - 15 陶先平. 基于 Internet 的移动 Agent 技术和应用研究: [南京大学博士学位论文]. 2001

(上接第 15 页)

SMO 算法存在有时收敛速度较慢的现象, 因此, 如何加快 SMO 算法的收敛速度, 采用何种启发式策略选择拉格朗日乘子, 成为今后 SMO 算法的研究方向. 另一大类由 O. L. Mangasarian 提出的一系列算法, 将与样本数有关的矩阵求逆转换为与样本输入空间维数有关的矩阵求逆, 降低了求解问题的复杂性, 算法简单, 易于实施, 也可用于求解经典方法难以胜任的大规模问题, 但 Mangasarian 算法出现比 SMO 算法要晚, 工程实际应用并不多. 但 Mangasarian 算法在处理大规模问题时要比 SMO 算法速度快, 相信今后在某些领域(如文本分类、生物信息学等需要处理海量数据的领域)会得到广泛的应用.

参 考 文 献

- 1 Vapnik V N. The Nature of Statistical Learning Theory. Springer, New York, 1995
- 2 Osuna E, Freund R, Girosi F. An improved training algorithm for support vector machines. In: Proc. of NNSP'97, 1997
- 3 Platt J. Sequential minimal optimization: A fast algorithm for training support vector machine: [Technical Report MSR-TR-98-14]. Microsoft Research, 1998
- 4 Mangasarian O L, Musicant D R. Successive overrelaxation for support vector machines. IEEE Transactions on Neural Networks, 1999, 10: 1032~1037
- 5 Joachims T. Making large-scale support vector machine learning practical. In: B. Scholkopf, C. J. C. Burges, A. J. Smola, eds. Advances in Kernel Methods - Support Vector Learning, Cambridge, MA. MIT Press, 1999
- 6 Mangasarian O L, Musicant D R. Lagrangian support vector machines[J]. Journal of Machine Learning Research, 2001, 1: 161~177
- 7 Mangasarian O L, Musicant D R. Active support vector machine classification. Advances in Neural Information Processing Systems (NIPS 2000), 2000
- 8 Lee Y-J, Mangasarian O L. SSVM: A smooth support vector machine. Computational Optimization and Application, 2000
- 9 Kaufmann L. Solving the quadratic problem arising in support vector classification. In: B. Scholkopf, C. J. C. Burges, A. J. Smola, eds. Advances in Kernel Methods - Support Vector Learning, Cambridge, MA. MIT Press, 1999. 147~168
- 10 Collobert R, Bengio S. SVM-Torch: Support vector machines for large-scale regression problems[J]. Journal of Machine Learning Research, 2001, 1: 143~160
- 11 Flach P A. On the state of the art in machine learning[J]. Artificial Intelligence, 2000, 131: 199~222
- 12 Tong S, Koller D. Support vector machines active learning with applications to text classification[J]. Artificial Intelligence, 2001. 45~66
- 13 Lin C J. On the convergence of the decomposition method for support vector machines. IEEE Transactions on Neural Networks, 2001, 12(6): 1288~1298
- 14 Keerthi S S, Shevade S, Bhattacharyya C, Murthy K. Improvements to Platt's SMO algorithm for SVM classifier design. Neural Computation, 2001, 13: 637~649
- 15 Burges C J C. A tutorial on support vector machines for pattern recognition[J]. Data Mining and Knowledge Discovery, 1998, 2(2): 1~47
- 16 Ferris M C, Munson T S. Interior point methods for massive support vector machines: [Technical Report 00-05]. Computer Sciences Department, University of Wisconsin, Madison, Wisconsin, May 2000
- 17 Scholkopf B, Smola A, Vapnik V N. Prior Knowledge in support vector kernels. NIPS'97, 1997
- 18 Cristianini N, Shawe-Taylor J. An introduction to Support vector machines. Cambridge University Press, Cambridge, 2000
- 19 Osuna E, Freund R, Girosi F. Training support vector machines: An application to face detection. In: Proc. of CVPR'97, 1997
- 20 席少霖, 赵凤治. 最优化计算方法[M]. 上海: 上海科学技术出版社, 1983
- 21 张学工. 关于统计学习理论与支持向量机[J]. 自动化学报, 2000, 26(1): 32~42