模式识别与人工智能
PR & AI

SVM QP 问题分解算法的研究进展

邱熔胜　　董云杰

（甘肃工业大学 电气工程与信息工程学院 兰州 730050）

摘　要　本文首先简单介绍了支持向量机的基本原理，然后简单分析了 SVM QP 问题的特点，详细介绍了解 SVM QP 问题的一系列分解算法，并对分解算法更进一步的研究方向作了探讨.

关键词　支持向量机，SVM QP 问题，分解算法
中图法分类号　TP181

1 引　言

近几年来，在 Vladimir N. Vapnik 的统计学习理论 (Statistical Learning Theory, SLT)[1]基础上发展起来的支持向量机[2-5]（Support Vector Machines, SVM）正成为机器学习领域中新的研究热点. 求解一个 SVM 可转化为优化一个 QP 问题（Quadratic Programming Problem）. 虽然 QP 问题在数学上早已解决，但由于 SVM QP 问题的特殊性，探寻关于 SVM QP 问题合理且高效的算法便成为 SVM 领域的一个重要研究课题. 目前，分解算法是优化 SVM QP 问题的主流算法，其研究也取得了长足的发展. 本文对 SVM QP 问题分解算法的研究进展作了简单的总结，并对 SVM 本身及 SVM QP 问题更进一步的研究方向作了探讨.

2 SVM 的基本原理及 SVM QP 问题

在解模式识别问题的过程中，由于实现由贝叶斯决策理论导出的期望风险最小化原则必须依赖类先验概率和类条件概率密度，故期望风险无法直接计算并最小化. 在实际应用中，我们只能用经验风险逼近期望风险，并希望通过最小化经验风险来实现期望风险最小[6]. 经验风险最小化(ERM)原则一直是解统计模式识别等统计机器学习问题的基本思想. 在此思想的指导下，人们主要解决如何更好地求取最小经验风险. 但实践证明，一味追求训练误差最小(对应经验风险最小)并不能得到最好泛化能力，有些情况下，训练误差太小反而会导致泛化能力下降，这在神经网络中表现得尤为突出(即过学习问题). 导致出现该问题的一个根本原因就是传统统计学是一种渐进理论，它的许多结论都是在样本数目趋向于无穷大的条件下得出的，而在小样本条件下，以传统渐进统计学为理论基础的经验风险最小化原则并不能很好地实现由贝叶斯决策理论导出的期望风险最小化原则. 为了解决传统渐进统计学应用在小样本统计学习中的不足，Vladimir N. Vapnik 等人建立了统计学习理论. 统计学习理论指出，在小样本条件下，只有同时控制经验风险和学习机的容量(用 VC 维衡量)，才能获得具有良好泛化能力的学习机(此处的学习机，在模式识别问题中即为分类器). 该思想在数学上可由式(2-1)表示的分类器推广能力的界来表达[4]：

$$R(\alpha) \leqslant R_{emp}(\alpha) + \phi\left(\frac{h}{l}, \frac{\ln(\eta)}{l}\right), \qquad (2-1)$$

其中 $\phi\left(\frac{h}{l}, \frac{\ln(\eta)}{l}\right) = \sqrt{\frac{h\left(\ln\frac{2l}{h}+1\right) - \ln\left(\frac{\eta}{4}\right)}{l}}$ 代表 VC 信任，$R(\alpha)$ 为实际风险，$R_{emp}(\alpha)$ 为经验风险，h 为 VC 维，l 为训练集样本数，$1-\eta$ 为置信度. 由该理论导出的结构风险最小化 (Structural Risk

收稿日期：2002－01－30；修回日期：2002－07－23

Minimization, 简称 SRM) 原则很好地实现了同时控制经验风险和学习机容量的思想,它将分类器构成的函数集合划分成若干按 VC 信任(也即按 VC 维)升序排列的子集,然后在每一个子集中寻找最小经验风险.这样,下一步要解决的问题便是构造分类器实现 SRM 原则,于是人们发展出 SVM 来解决这一问题.我们注意到,在众多分类器当中,线性分类器具有最简单的结构,人们便考虑用类似线性判别函数的方法来实现 SRM 原则,SVM 便是从模式类线性可分情况下的最优分类面(Optimal Hyperplane)提出的.它的基本思想是:若在原始特征空间中实现的分类器结构十分复杂,则通过定义适当的核函数诱导出某个非线性变换,用此变换将原始特征空间映射到一个高维空间,然后在这个新的特征空间中求得最优线性分类面,以降低分类器的复杂度.由 RKHS(Reproducing Kernel Hilbert Spaces, 简称 RKHS) 理论[7-9] 可知,当选定的核函数满足一定条件时,由该核函数导出的高维特征空间中两特征向量间的点积可由核函数在低维特征空间中对应两特征向量上定义的计算而得到.这样,我们便可在低维特征空间中处理对应高维特征空间中的数据.由于求解 SVM 只涉及到向量间的点积运算,故我们可不必担心由于引入核函数而引起维数灾难,而可将注意力集中到如何选取恰当的核函数上,以改善特征向量在高维特征空间中的分布,从而使分类器结构更简单.这样,求解 SVM 的过程即为在高维特征空间中求解模式类样本数据之间最优分类面的过程,此处的最优分类面是在控制样本错分率的前提下使两类样本数据间的分类间隔(高维特征空间中)最大的分类面.Δ- 间隔分类超平面集合的 VC 维上界 h 可由式(2 - 2) 给出[1]:

$$h \leqslant \min\left(\left[\frac{R^2}{\Delta^2}\right], n\right) + 1, \qquad (2-2)$$

其中,R 为包含训练数据的球体的半径,

$$\Delta = \frac{1}{\| w^* \|},$$

$$w^* = \sum_{i=1}^{l} y_i \alpha_i \boldsymbol{x}_i, \ \alpha_i \geqslant 0, i = 1,2,\cdots,l,$$

n 为特征空间的维数.这样在选定核函数,训练集确定的情况下,只需最小化 $\| w^* \|$ 便可控制 h,从而达到控制 VC 信任的目的,然后寻找最小经验风险,继而实现 SRM 原则.

求解一个 SVM 可转化为优化一个 QP 问题[13]:

$$\min \frac{1}{2}\boldsymbol{\alpha}^T \boldsymbol{Q} \boldsymbol{\alpha} - \boldsymbol{\alpha}^T \boldsymbol{e}$$

$$subject\ to \begin{cases} 0 \leqslant \boldsymbol{\alpha} \leqslant \boldsymbol{c} \\ \boldsymbol{\alpha}^T \boldsymbol{y} = 0 \end{cases} \qquad (2-3)$$

其中,$Q_{ij} = y_i y_j K(\boldsymbol{x}_i, \boldsymbol{x}_j)$, $\boldsymbol{\alpha} = (\alpha_1\, \alpha_2 \cdots \alpha_n)^T$, $\boldsymbol{y} = (y_1\, y_2 \cdots y_n)^T, y_i \in \{+1, -1\}, i = 1,2,\cdots,n$, $\boldsymbol{c} = (c_1\, c_2 \cdots c_n)^T$, 且 $c_1 = c_2 = \cdots = c_n = c$. $K(\boldsymbol{x}_i, \boldsymbol{x}_j)$ 为我们选定的核函数,$\boldsymbol{x}_i$ 为样本向量,y_i 为样本类别,c 为控制错分样本比例与模型复杂度之间折衷度的常量,$\boldsymbol{\alpha}$ 为解向量,$\boldsymbol{e}$ 为单位向量.我们称问题(2 - 3) 为 SVM QP 问题.解决了 SVM QP 问题后,可很容易得到支持向量机:

$$f(\boldsymbol{x}) = \mathrm{sgn}\left(\sum_{i=1}^{m} \alpha_i^* y_i K(\boldsymbol{x}_i, \boldsymbol{x}) + b^*\right), \qquad (2-4)$$

其中,m 是支持向量的数目.

针对 SVM QP 问题,有许多标准算法[10],但这些算法大都针对小样本或 $\boldsymbol{Q}$ 为稀疏矩阵的问题而设计.在实际应用中,SVM 要处理的数据量很大,维数较高,在 SVM QP 问题中,一般情况下,$\boldsymbol{Q}$ 非稀疏;另一方面,由于矩阵 $\boldsymbol{Q}$ 的规模与样本量间呈平方关系,故在样本数较大时,直接解上述问题会由于存储空间的限制而无法进行.在实际应用中,要求快速准确地解决 SVM QP 问题,故必须针对 SVM QP 问题设计高效可行的算法.在众多算法中,分解算法是得到广泛研究并在实际中得到广泛应用的算法.

3 分解算法

3.1 分解算法基础

3.1.1 SVM QP 问题数学特性分析

SVM QP 问题中的矩阵 $\boldsymbol{Q}$ 是正半定 Hessian 阵,且问题具有线性约束.这类优化问题具有良好的数学特性: 1) 其目标函数是凸函数,且具有全局最优解. 2) 若解满足 KKT 条件,则该解必是全局最优解.从这一点来讲,SVM 从原理上就回避了优化过程中的局部极小问题.SVM 具有如下两个特点: 1) 支持向量的数目远小于训练样本的数目. 2) 有些支持向量是取上界值的支持向量(Bounded Support Vectors).这反映在 SVM QP 问题中就是其全局最优解具有稀疏性.我们可充分利用解的稀疏性来加快 SVM QP 问题的优化速度.

下面我们简单介绍一下 KKT 条件(Karush-Kuhn-Tucker Conditions).

KKT 条件　对 SVM QP 问题应用 Lagrange 优化技术.则其 Lagrange 函数为

$$L(\lambda_{eq}, \boldsymbol{\lambda}_{lo}, \boldsymbol{\lambda}_{up}) = -\boldsymbol{\alpha}^T \boldsymbol{e} + \frac{1}{2}\boldsymbol{\alpha}^T \boldsymbol{Q} \boldsymbol{\alpha} + \lambda_{eq} \boldsymbol{\alpha}^T \boldsymbol{y} - \boldsymbol{\lambda}_{lo} \boldsymbol{\alpha}$$

$+ \boldsymbol{\lambda}_{up}(\boldsymbol{\alpha} - \boldsymbol{c})$,

则 KKT 条件为[13]

$$\begin{cases} g(\boldsymbol{\alpha}) + (\lambda_{eq}\boldsymbol{y} - \boldsymbol{\lambda}_{lo}{}^{T} + \boldsymbol{\lambda}_{up}{}^{T}) = 0 \\ \forall i \in [1,2,\cdots,n], \lambda_{lo\,i}(-\alpha_i) = 0 \\ \forall i \in [1,2,\cdots,n], \lambda_{up\,i}(\alpha_i - c) = 0 \\ \lambda_{lo\,i} \geqslant 0, \lambda_{up\,i} \geqslant 0, \boldsymbol{\alpha}^T\boldsymbol{y} = 0 \\ 0 \leqslant \alpha_i \leqslant c \end{cases} \tag{3-1}$$

其中,$g(\boldsymbol{\alpha}) = -\boldsymbol{e} + \boldsymbol{Q}\boldsymbol{\alpha}$ 是 SVM QP 问题中目标函数对 $\boldsymbol{\alpha}$ 的偏微分,$\lambda_{eq}, \lambda_{up\,i}, \lambda_{lo\,i}$ 为 Lagrange 因子.

3.1.2 分解算法的基本思想

分解算法的基本思想是将 SVM QP 问题分解成一系列规模较小的子问题,希望通过对子问题的迭代求解而得到 SVM QP 问题的全局最优解.这样便解决了存储空间受限的问题.另一方面,要使分解算法可行,还必须解决诸如分解策略,迭代算法的收敛性,收敛条件,优化时间,以及如何充分利用解的稀疏性来加快优化速度等众多问题.目前,针对分解算法的研究,大都集中在这几个方面.

3.2 CHUNKING

CHUNKING 是 Vladimir N. Vapnik 等人提出的一种求解 QP 问题的算法[11],它可看成是一种不完全的分解算法.算法将原先的 QP 问题分解成一系列待优化的子问题.任何一个子问题都维护一个与其相关的变量的子集,在优化过程中保持其它的变量值不变.子问题维护所有上一步优化后得到的非零变量,及其它违反 KKT 条件最严重的 M 个变量.这样经过反复迭代优化,最终将得到 SVM QP 问题的全局最优解.

3.2.1 算法描述

1.选择初始子问题,包括选择问题的规模 m,待优化的变量子集及每次添加进优化子集的违反 KKT 条件的变量数目 M.

2.采用一种 QP 优化器求解上述子问题,得到一个分类器.

3.用剩余训练样本测试上述分类器,采用 KKT 条件评价测试结果.如果所有变量都满足 KKT 条件,则结束训练;否则,选择违反 KKT 条件最剧烈的 M 个变量,并与上一步优化得到的非零值变量构成待优化子问题,重复第 2 步.

3.2.2 算法分析

CHUNKING 中,初始问题规模及待优化的变量子集可随机选取,一般来讲,初始问题规模不宜过大.参数 M 对收敛速度有较大影响,不宜选得过大,也不宜过小,实际应用中可采用交叉验证等模型选择技术优化该参数.CHUNKING 的存储空间需求受所解决问题支持向量数目的限制,若问题支持向量的数目过大,算法仍然有存储空间需过大的缺点,甚至对有些问题,该算法不能胜任.另外,算法需要一个 QP 优化器来全局优化每一个 QP 子问题.

3.3 子问题规模固定的分解算法

针对 CHUNKING 的不足,Osuna 等人在 1997 年提出了一种新的 SVM QP 问题的分解算法[12,13].该算法仍沿用分解问题,迭代求解的思路,与 CHUNKING 不同的是,它每次迭代求解都将 SVM QP 问题中的变量分解成两部分:自由变量集合和固定变量集合.其中自由变量集合是本次迭代优化要操纵的变量的集合,其规模固定;而固定变量集合中变量的值在本次迭代优化过程中保持不变.算法采用最大梯度下降算法选取自由变量集合,并结合缓冲区技术,缩减问题规模技术等方法提高其性能,算法采用 KTT 条件评价优化结果,经过迭代,当解向量满足 KKT 条件时,便得到 SVM QP 问题的全局最优解.

3.3.1 算法描述

1.选择包含 q 个变量的自由变量集合 B,算法收敛评价参数及内核函数等参数.

2.采用一种 QP 优化器优化子问题,固定变量集合中的变量值在优化过程中保持不变.

3.采用 KKT 条件评价解的合理性,若解向量满足 KKT 条件,训练结束;否则,采用最陡梯度下降算法选取自由变量集合,重复第 2 步.

3.3.2 自由变量集合的选择技术

分解算法中,自由变量集合的选择策略对算法的效率有很大的影响.高效的选择策略可减少迭代优化的次数,从而提高算法的效率.算法选择经迭代优化后解向量可最大程度地接近 SVM QP 问题全局最优解的那些变量组成集合 B. Osuna 等人采用 Zoutendijk 在 1970 年提出的方法[13]来实现上述选择思想.Zoutendijk 方法的基本出发点是优化一个由 q 个非零元素组成的具有最大梯度下降方向的矢量,对应于该方向矢量中非零元素的解向量中的变量构成自由变量集合 B.在算法实现中,可通过维护一个由梯度信息组成的有序数组成一个有序堆来完成集合 B 的选择任务.令

$$S_i = \sum_{j=1}^{n} \alpha_j y_j K(\boldsymbol{x}_i, \boldsymbol{x}_j) - y_i, \tag{3-2}$$

则由式(3-1)表示的 KKT 条件等价于:

$$\forall \alpha_i = 0 \quad (S_i - \lambda_{eq}) \geqslant 0, \tag{3-3}$$

$$\forall 0 < \alpha_i < c \quad (S_i - \lambda_{eq}) = 0, \tag{3-4}$$

$\forall \alpha_i = c \quad (S_i - \lambda_{eq}) \leqslant 0.$ (3-5)

对给定的 α_i,定义如下集合:

$\boldsymbol{I}_0 = \{i \mid 0 < \alpha_i < c\}$, $\boldsymbol{I}_1 = \{i \mid y_i = 1,\ \alpha_i = 0\}$, $\boldsymbol{I}_2 = \{i \mid y_i = -1,\ \alpha_i = c\}$, $\boldsymbol{I}_3 = \{i \mid y_i = 1,\ \alpha_i = c\}$, $\boldsymbol{I}_4 = \{i \mid y_i = -1,\ \alpha_i = 0\}$.

令 $\boldsymbol{A}_{up} = \boldsymbol{I}_0 \cup \boldsymbol{I}_1 \cup \boldsymbol{I}_2$, $\boldsymbol{A}_{low} = \boldsymbol{I}_0 \cup \boldsymbol{I}_3 \cup \boldsymbol{I}_4$,则式(3-3)、(3-4)、(3-5)等价于:存在 λ_{eq},使得 $\lambda_{eq} \leqslant S_i$, $\forall i \in \boldsymbol{A}_{up}$; $\lambda_{eq} \geqslant S_i$, $\forall i \in \boldsymbol{A}_{low}$;令 $b_{up} = \min\{S_i \mid i \in \boldsymbol{A}_{up}\}$, $b_{low} = \max\{S_i \mid i \in \boldsymbol{A}_{low}\}$,则 KKT 条件可简单表示为

$b_{low} \leqslant b_{up}.$ (3-6)

可以证明,由Zoutendijk方法得到的变量集合就是违反 KKT 条件最剧烈的变量集合.当未得到最优解向量时,式(3-6)必不满足.此时,可从 $\boldsymbol{A}_{up}$ 中按 S_i 的值从小到大选择 $q/2$ 个变量(假定自由变量集合中元素数 q 为偶数)与从 $\boldsymbol{A}_{low}$ 中按 S_i 值从大到小选出的 $q/2$ 个变量共同组成自由变量集合,此时,集合 $\boldsymbol{B}$ 中的变量违反 KKT 条件最为剧烈.

3.3.3 算法收敛标准

实际求解过程中,引入容许误差 $\tau > 0$,则KKT条件:

$\forall \alpha_i = 0 \quad (S_i - \lambda_{eq}) \geqslant -\tau,$ (3-7)

$\forall 0 < \alpha_i < c \quad |(S_i - \lambda_{eq})| \leqslant \tau,$ (3-8)

$\forall \alpha_i = c \quad (S_i - \lambda_{eq}) \leqslant \tau.$ (3-9)

上面三式为算法执行过程中判断解向量是否达最优及挑选集合 $\boldsymbol{B}$ 的依据.

3.3.4 问题规模收缩策略

由于SVM QP问题的解具有稀疏性,算法希望通过在优化过程中识别出非支持向量或取界值的支持向量,并将其剔除出SVM QP问题,以减小问题规模,提高算法性能.但到目前为止,仍没有很好的方法能够完成上述识别任务.Osuna等人采用一种启发式算法来实现缩减问题规模的目的.原理如下:在拉格朗日优化式中,界约束的拉格朗日因子表明了变量推进到约束值的程度.一个约束的严格为正的拉格朗日因子表示变量在这个界上达到最优.计算拉格朗日因子,如果其始终为正或大于某个设定值的次数达到某个设定值,则此变量的解很可能已达到最优.这意味着变量已固定,不需计算梯度和最优条件.可将这些变量剔除出优化问题.但这种启发式方法并无严格数学论证,可能失效,故每次迭代收敛后,仍需用KKT条件检验解向量.

q,τ 及解向量元素取值上界 c 等参数,在实际应用中,可采用留一法、交叉验正等方法进行优化.算法还采用缓冲区技术,将使用频率较大的核函数的值存于缓冲区中,以备后用,以此来减少核函数的重复计算,这虽然增加了算法的空间需求,但却提高了算法的执行速度.从上分析可知,算法仍需用一个QP问题优化器进行子问题优化.Thorsten Joachims在其软件包 SVM^{light} 中实现了上述算法[14].文献[15]分析了这类算法的收敛性.

3.4 SMO

SMO(Sequential Minimal Optimization)是John C. Platt在1998年提出的一种解决SVM QP问题的算法[16].它可看成是一种特殊的分解算法.SMO将自由变量集合的大小固定为2,由于SVM QP问题具有线性约束和不等式约束,这是使优化可行的最小的变量数.由于每一次迭代只优化两个参数,且SVM QP问题具有线性约束,这使得对两个变量的优化过程不再需要引入一个QP问题优化器来进行数字求解,而是通过分析的方法便可得到两变量的最优解.由于QP问题优化器运算较复杂,占用运算时间较多,而分析求解只需执行一小段逻辑分析程序,运算的时间效率较QP问题优化器要高得多,SMO正是通过这一点来提高算法的性能.

3.4.1 两变量QP问题的分析求解

设 α_1,α_2 为待优化变量.令 $h = y_1 y_2$,则 α_2 的约束可表示如下:

$$H = \min(c, \alpha_2 + h\alpha_1 - \frac{1}{2}(h - 1)c),$$

$$L = \max(0, \alpha_2 + h\alpha_1 - \frac{1}{2}(h + 1)c).$$

正常情况下,目标函数正定,沿线性约束方向具有最小值,则 α_2 计算如下:

$$\alpha_2^{new} = \alpha_2 + \frac{y_2(E_1 - E_2)}{K(\boldsymbol{x}_1, \boldsymbol{x}_1) + K(\boldsymbol{x}_2, \boldsymbol{x}_2) - 2K(\boldsymbol{x}_1, \boldsymbol{x}_2)},$$

其中,

$E_i = y_i^{out} - y_i$ 是SVM上一步输出误差.将 α_2^{new} 经区间(L, H)限定后,得到 α_2 最优解 $\alpha_2^{clipped}$.则有

$\alpha_1^{new} = \alpha_1 + h(\alpha_2 - \alpha_2^{clipped}).$

然后,修正 λ_{eq},以使用式(3-7)、(3-8)、(3-9)表示的KKT条件评价解向量.

3.4.2 待优化变量选择策略

SMO采用启发式方法确定每一步要优化的变量,以提高收敛速度.针对 α_1,α_2 有两个独立的选择策略,选择其中一个变量(如 α_1)的过程构成外层循环,如果外层循环找到一个违反KKT条件的变量,则将该变量确定为优化目标.外层循环单程遍历所有变量,并与多重遍历非界值支持向量对应的变量集交替进行.算法将非界值支持向量集对应的变量

集优化完成后再单程遍历所有变量.这样,算法把训练时间主要集中在违反 KKT 条件概率较大的变量身上.随着算法的进行,界值支持向量对应的变量趋向于保持边界值不动,而其它支持向量对应的变量则得到更多的动态优化.一个变量选好后,算法通过选取使 $|E_1 - E_2|$ 最大的变量作为另外一个待优化的变量.

以上分析了 SMO 区别于其它分解算法的主要特征.同其它分解算法一样,SMO 也使用问题规模收缩策略和缓冲区技术来提高算法的性能.SMO 虽然避免了数字求解 QP 问题,但也带来了选择优化变量次数增大的问题.由于变量选择过程要使用核函数的值,故 SMO 增加了计算核函数的运算量.如何减少核函数评价次数,是提高 SMO 性能的关键.

3.5　改进的 SMO

S.S. Keerthi 等人于 1999 年提出了一种 SMO 的改进算法[17].回顾前文所述,无论 Osuna 的分解算法还是 John C. Plat 的 SMO 算法,都使用式(3-7)、(3-8)、(3-9)表示的 KKT 条件作为选择自由变量集合和评价算法是否收敛的标准.考察(3-7)、(3-8)、(3-9)我们发现,要使用这种形式表示的 KKT 条件,我们必须维护一个 Lagrange 因子 λ_{eq}.可以证明,在 SVM QP 问题解空间的最优点上,λ_{eq} 与式(2-4)中的阈值 b^* 取相同的值[17].对于给定的 SVM QP 问题,其对应于式(2-4)中的阈值 b^* 是个待定常量,当 SVM QP 问题优化未完成时,λ_{eq} 只是对 b^* 的估计.随着优化的不断进行,λ_{eq} 对 b^* 的估计越来越准确,若算法收敛,λ_{eq} 最终完成对 b^* 的较准确的估计.由此可看出采用式(3-7)、(3-8)、(3-9)表示 KKT 条件并不合理,其原因分析如下:λ_{eq} 是当解向量满足 KKT 条件时才能准确估计出的一个量,这时它可作为判定解向量是否满足 KKT 条件的标准;但当解向量还未达最优时,估计出的 λ_{eq} 可能不准确,显然,用其作为选择自由变量集合和评价算法是否收敛的标准并不合适,可能会造成优化变量对不匹配及收敛误判,从而影响算法的效率.虽然计算 λ_{eq} 对解 SVM 必不可少(因为我们必须求得阈值 b^*),但考察式(3-2)～(3-6)便知,我们完全可以不使用 λ_{eq} 而通过维护式(3-6)中的两个变量来表示 KKT 条件.实践证明,采用式(3-6)表示的 KKT 条件能很好地提高算法的性能.

限于篇幅,关于 SMO 算法收敛性的论述,本文不作叙述,感兴趣的读者可参阅文[18]～[20].

4　试验与讨论

文献[16]在几个不同的数据集上分别对 CHUNKING, SMO, 用软件包 SVM^{light} 进行了测试,文献[12]在几个不同的数据集上比较研究了 SMO 与改进的 SMO 性能.但这些研究主要围绕提高算法时间性能展开,为了更全面地研究不同分解算法的性能,我们在 Reuters category "corporate aquisitions"数据集上对子问题规模固定的分解算法和改进的 SMO 进行了比较研究.Reuters category "corporate equistions"可从文献[14]中 softwre 选项下载 mySVM 软件而同时得到.这是一个两类文本分类问题,包括 1 000 个正类数据和 1 000 个反类数据,数据向量的最大维数为 9 947.子问题规模固定的分解算法用 SVM^{light}[14] 实现,改进的 SMO 用 LibSVM[14]实现.试验在 Intel Pentium 4　1.6G/128M RAM 机器上进行.我们从数据集中选出几个规模不等的数据子集进行研究,试验结果如下:

表 1　子问题规模固定的分解算法和改进的 SMO 在不同数据集上的性能比较

算法	正类数据	反类数据	支持向量	训练时间	准确度%
LibSVM	169	196	208	0.5	94.6
SVM^{light}	169	196	215	0.55	94.75
LibSVM	330	603	329	2.25	95.55
SVM^{light}	330	603	330	1.96	95.5
LibSVM	1000	1000	225	4.7	100
SVM^{light}	1000	1000	225	3.18	100

采用参数:kernel = sigmoid, $a = 1.3$, $b = 0.05$, $c = 2000$, $e = 0.001$.其中,a,b 为 sigmoid 内核的两个参量.

由表 1 可知,两个算法在不同训练集上得到的支持向量略微不同,分类器的分类准确度也有些差异.分析 SVM 的导出过程,便可发现,SVM 通过求解 SVM QP 问题来最小化式(2-2)中的 w^*,以此来控制 Δ-间隔分类超平面集合的 VC 维上界,进而实现 SRM 原则的容量控制思想.但这里面存在几点问题:1)要真正实现 SRM 同时控制经验风险和分类器的容量进而最小化实际风险的思想并不容易,这需要在式(2-1)的两个加和项之间进行折衷,反映在 SVM QP 问题中,就是要优化参数 c,到目前为止,仍没有很好的策略来完成这个任务,在实际应用中通常采用试探的方法优化该参数.2)解 SVM QP 问题只是优化了 Δ-间隔分类超平面集合的 VC 维上界,由此并不能得到分类器函数集合的具体 VC

维,而 SVM QP 问题的解可能不唯一(表现为式(2-2)中 w^* 的组合不同,VC 维也可能不同),不同组合对应的分类器的推广能力可能不同[21].在给定样本集上,怎样解 SVM QP 问题,使得到的全局最优解能导出具有最好推广性能的 SVM 还有待进一步研究.3)对具体分类问题,如何选择合适的核函数是个很难解决的问题.4)SVM QP 问题只反映了特征空间中类间数据的分布关系,而没有考虑每一类数据内在的分布关系,对某些问题,如二类模式中,如果一类数据方差很大,另一类数据方差很小,则类间隔中线显然不是最优分类边界.在 SVM 领域中,上述几个问题都值得进一步研究.文献[22]给出了一种避免优化参数 c 的算法来求解 SVM(即 v-SVM),该算法通过引入一个参数 $v\in(0,1]$ 取代传统 SVM QP 问题中的参数 c.该算法的优点是 v 的控制意义较 c 明显,但仍需对 v 进行优化.文献[23]给出了一种表达训练数据类间间隔的新的界.文献[24]又将该界加以改进,并在此基础上提出了线性分类器推广误差的新界,该界基于 PAC-Bayesian 框架[25],它指出在给定训练数据集上,应该最大化一个正则间隔,这样得到的分类器在一定程度上克服了以传统间隔的中线作为最优边界的缺点.文献[26]提出了一种基于估计贝叶斯点的新型分类器,即 BPM(Bayes Point Machine).文献[27]提出了一种大数据量情况下 BPM 的快速算法,文献[28]又提出了一种最小化期望风险的新原则,称作邻域风险最小化(Vicinal Risk Minimization, VRM)原则.

5 结束语

传统的 SVM 最终将问题归结为解一个在给定数据集上的 SVM QP 问题,针对该问题发展出了一系列解法.SVM 发展到今天.针对于解 SVM QP 问题的传统 SVM 的研究已取得了很大的进展,同时也发现了传统 SVM 的一些不足,SVM 本质上是一种基于统计与 PAC 学习的非参数机器学习方法,寻找更好的理论模型来改进 SVM 正成为该领域的研究重点.

参 考 文 献

[1] Vapnik V N, 著;张学工,译.统计学习理论的本质.北京:清华大学出版社,2000

[2] 张学工.关于统计学习理论与支持向量机.自动化学报,2000, 26(1): 32-42

[3] Burges C J C. A Tutorial on Support Vector Machines for Pattern Recognition. Data Mining and Knowledge Discovery, 1998, 2(2): 121-167

[4] Schölkopf B. Support Vector Learning. R. Oldenbourg Verlag, Munich, 1997

[5] Evgeniou T, Pontil M, Poggio T. Regularization Networks and Support Vector Machines. Advances in Computational Mathematics, 2000, 13(1): 1-50

[6] 边肇祺,张学工,等.模式识别.第2版.北京:清华大学出版社,2000

[7] Schölkopf B, *et al*. A Generalized Representer Theorem. NeuroCOLT2 Technical Report Series, NC2-TR-2000-81, 2000. http://www.neurocolt.com.

[8] Schölkopf B. A Short Tutorial on Kernels. Tutorial Given at the NIPS'00 Kernel Workshop, 2000. http://research.microsoft.com/~bsc.

[9] Wahba G. An Introduction to Model Building with Reproducing Kernel Hilbert Spaces (with Applications). Interface 2000 Short Course, New Orleans, 2000

[10] Osuna E, Freund R, Girosi F. Supprot Vector Machines: Training and Applications. Technical Reprot AIM-1602, MIT A. I. Lab, 1996

[11] Boser B E, Guyon I M, Vapnik V N. A Training Algorithm for Optimal Margin Classifiers. In: Haussler D ed. 5th Annual ACM Workshop on COLT, Pittsburge, PA, ACM Press, 1992, 144-152

[12] Osuna E, Freund R, Girosi F. An Improved Training Algorithm for Support Vector Machines. In: Principe J, Gile L, Morgan N, Wilson E, eds. Neural Networks for Signal Processing VII - Proceedings of the 1997 IEEE Workshop, New York, 1997, 276-285

[13] Joachims T. Making Large-Scale SVM Learning Practice. In: Schölkopf B, *et al*, eds. Advances in Kernel Methods - Support Vector Learning, MIT Press, 1998

[14] http://www.kernel-machines.org

[15] Lin C J, *et al*. The Analysis of Decomposition Methods for Support Vector Machines. In: Proceeding of IJCAI 99, SVM Workshop, 1999

[16] Platt J C. Using Analytic QP and Sparseness to Speed Training of Support Vector Machines. In: Advances in Neural Information Processing System 11, 1999, 550-557

[17] Keerthi S S, *et al*. Imprvoments to Platt's SMO Algorithm for SVM Classifier Design. Neural Computation, 2001, 13(3): 637-649

[18] Keerthi S S, Gilbert E G. Convergence of a Generalized SMO Algorithm for SVM Classifier Design. Machine Learning, 2002, 46(1-3): 351-360

[19] Lin C J, *et al*. On the Convergence of the Decomposition Method for Support Vector Machines. IEEE Trans on Neural Networks, 2001, 12(6): 1288-1298

[20] Lin C J. Asymptotic Convergence of an SMO Algorithm without any Assumptions. IEEE Trans on Neural Networks, 2002, 13(1): 248-250

[21] Müuler K R, *et al*. An Introduction to Kernel-Based Learning Algorithms. IEEE Trans on Neural Networks, 2001, 12(2): 181-202

[22] Schölkopf B, *et al*. New Support Vector Algorithms. Neural Computation, 2000, 12(5): 1207-1245
[23] Shawe-Taylor J, *et al*. Structural Risk Minimization over Data-Dependent Hierarchies. IEEE Trans on Information Theory, 1998, 44(5): 1926-1940
[24] Herbrich R, Graepel T. A PAC-Bayesian Margin Bound for Linear Classifiers: Why SVMs Work. In: Advances Neural Information Processing System 13, 2001, 224-230
[25] McAllester D A. Some PAC Bayesian Theorems. In: Proc of the 7th Annual Conference on Computational Learning Theory, Madison, Wis-Consin, 1998, 230-234
[26] Herbrich R, Graepel T, Campbell C. Robust Bayes Point Machines. In: Proc of European Symposium on Artificial Neural Networks, 2000, 45-54
[27] Herbrich R, Graepel T. Large Scale Bayes Point Machines. In: Neural Information Processing System 13, 2001, 217-223
[28] Chapelle O, *et al*. Vicinal Risk Minimization. In: Advances in Neural Information Processing System 13, 2001, 416-422

RESEARCH DEVELOPMENTS OF DECOMPOSITION ALGORITHM FOR SVM QP PROBLEM

Qiu Rongsheng, Dong Yunjie
(*College of Electric & Information Engineering, Gansu University of Technology, Lanzhou* 730050)

ABSTRACT

In this paper, a simple introduction to the principle of SVM is given firstly. Then the character of SVM QP problem as well as a series of decomposition algorithms for SVM QP problem are reviewed. Finally, further research fields are suggested.

Key Words　Support Vector Machines (SVM), SVM QP Problem, Decomposition Algorithm

27. 一种改进的支持向量机E-SVM算法
28. 求解多物流配送中心问题的分解算法
29. 遥感影像解译算法的研究进展
30. QP(质量进展)工具箱
31. 对算法问题的研究
32. N_2O分解催化剂的研究进展
33. SVM算法分析与研究
34. 基于SVM的分类问题的研究
35. 基于SVM的大学生热点问题的研究
36. 基于SVM算法学生毕业的预测
37. 基于ACMSFLA-SVM的人脸识别算法
38. 基于GA-SVM对HHT算法的优化
39. 基于SVM算法癫痫脑电的研究初探
40. 基于Spark的并行SVM算法研究
41. NO_X直接分解催化剂的研究进展
42. 基于SVM算法的局部放电模式识别
43. 脂肪分解关键酶的研究进展
44. 大型过程优化问题SQP算法的研究进展
45. 基于LS-SVM算法的旅游数量预测
46. DMC算法中QP求解的降维算法
47. SVM增量学习算法研究
48. 复分解反应的研究进展
49. SVM的快速分类及其算法
50. SVM-KNN分类算法研究