计算机科学2003Vol. 30№. 8

一种改进的SMO算法

张 召 黄国兴 鲍 钰
(华东师范大学计算机系 上海200062)

An Improved SMO Algorithm

ZHANG Zhao HUANG Guo-Xing BAO Yu
(Department of computer science, East China Normal University, Shanghai 200062)

Abstract In this paper we have pointed out an important source of inefficiency in SMO algorithm that is caused by the operation with a single threshod value. We have suggested modifications of SMO algorithm that overcome the problem by efficiently maintaining and updating two threshold parameters. Our experiments show that these modifications speed up the SMO algorithm.

Keywords SVM, SMO, KKT condition

1. 引言

支持向量机(Support Vector Machine，简称SVM)是Vapnik等人提出的一类新型机器学习算法。由于其出色的学习性能，该技术已成为机器学习界的研究热点，并在很多领域得到了成功的应用。在最近几年里已经提出了很多种关于SVM的训练算法，包括"块算法"、"固定工作样本集算法"等。其中以John C. Platt提出的SMO(Sequential Minimal Optimization)算法应用最为广泛。SMO算法和其他算法相比训练速度快，扩展能力好而且容易实现。

但是我们研究发现SMO算法仍然存在可以进一步改进的地方。比如，SMO算法在使用KKT条件判优的时候使用单一的阈值就是导致SMO算法低效的原因之一。针对这一问题，提出在使用KKT条件判优时用两个阈值参数。用UCI的标准数据库和我们自己的肾病诊断数据集对Platt的SMO算法和我们改进的SMO算法测试的结果表明，我们的算法比Platt的SMO算法有更好的性能。

2. SVM的理论基础

支持向量机的理论最初来自于对数据二值分类问题的处理。其机理可以简单地描述为：寻找一个满足分类要求的最优分割超平面，使其在保证分类精度的同时最大化超平面两侧的空白区域(如图1所示)。这使得SVM分类器的结果不仅在训练集上得到优化，而且在整个样本集上的风险也拥有上界，这就是SVM的结构风险最小化的思想。在线性不可分的问题上，Vapnik等人成功地引入了核空间理论，将低维的输入空间数据通过非线性映射函数映射到高维特征空间，从而把分类问题转化到高维特征空间进行。可以证明，如果选用适当的映射函数，大多数输入空间线性不可分问题在特征空间可以转化为线性可分问题。这一问题的解决，使得SVM分类器正式成为通用的分类器之一。

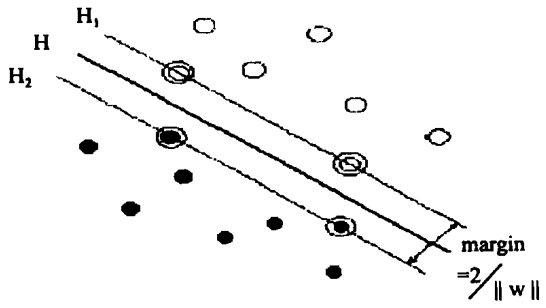

图1 数据点集的超平面分割

3. SVM方法介绍

支持向量机的原理是用分类超平面将空间中两类样本点正确分离，并取得最大边缘(正样本与负样本到超平面的最小距离)，这样原问题为一个有约束的非线性规划问题：

$$\min \frac{1}{2}\|w\|^2+C\sum_i \xi_i \tag{1}$$

约束：$y_i(w\cdot z_i-b)\geqslant 1-\xi_i$

$\xi_i\geqslant 0, i=1,2,\cdots,n$

可见目标函数是严格上凹的二次型，约束函数是下凹的，这是一个严格的凸规划。按照最优化理论中凸二次规划的解法，我们可以把原问题转化为Wolfe对偶问题：

$$\text{最大值：}Q(\alpha)=\sum_{i=1}^{n}\alpha_i-\frac{1}{2}\sum_{i,j=1}^{n}\alpha_i\alpha_j y_i y_j K(x_i,x_j) \tag{2}$$

约束：$\sum_{i=1}^{n} y_i\alpha_i=0$

$0\leqslant\alpha_i\leqslant C \quad i=1,\cdots,n$

这就是SVM方法的最一般的表述。为了方便后面的陈述，这里对对偶问题的最优解做一些推导。

定义1 $$\omega(\alpha)=\sum_i \alpha_i y_i \Phi(x_i) \tag{3}$$

$$F_i=\omega(\alpha)\cdot\Phi(x_i)-y_i=\sum_j \alpha_j y_j K(x_i,x_j)-y_i \tag{4}$$

对偶问题的Lagrange函数可以写成：

$$L=\frac{1}{2}\omega(\alpha)\cdot\omega(\alpha)-\sum_i\alpha_i-\sum_i\delta_i\alpha_i+\sum_i\mu_i(\alpha_i-C)-\beta\sum_i\alpha_i y_i \tag{5}$$

张 召 硕士研究生，研究方向：智能信息系统，数据挖掘。黄国兴 教授，研究方向：智能信息系统，数据挖掘。鲍 钰 硕士研究生，研究方向：智能信息系统，数据挖掘。

KKT 条件为：

$$\frac{\partial L}{\partial \alpha_i}=(F_i-\beta)y_i-\delta_i+\mu_i=0$$

$$\delta_i\alpha_i=0 \quad 且\ \delta_i\geqslant 0$$

$$\mu_i(\alpha_i-C)=0\ \forall\ ,$$

由此，我们可以推导出如下关系式：

若 $\alpha_i=0$ 则 $\delta_i\geqslant 0,\mu_i=0\Rightarrow(F_i-\beta_i)y_i\geqslant 0$ (6a)

若 $0<\alpha_i<C$ 则 $\delta_i=0,\mu_i=0\Rightarrow(F_i-\beta_i)y_i=0$ (6b)

若 $\alpha_i=C$ 则 $\delta_i=0,\mu_i\geqslant 0\Rightarrow(F_i-\beta_i)y_i\leqslant 0$ (6c)

由于 KKT 条件是最优解应满足的充要条件，因此目前提出的一些算法几乎都是以是否违反 KKT 条件作为迭代策略的准则，当然 SMO 算法也不例外。

4. SMO 算法及其存在的问题

4.1 早期的 SVM 训练算法

人们针对 SVM 本身的特点提出了很多算法来解决对偶寻优问题。根据子问题的划分和迭代策略的不同，可以大致分为两类。

第一类是“块算法”。所谓“块算法”就是选择一部分样本构成工作样本集进行训练，剔除其中的非支持向量，并用训练结果对剩余样本进行检验，将不符合训练结果（一般指违反 KKT 条件）的样本（或其中的一部分）与本次结果的支持向量合并成为一个新的工作样本集，然后重新训练。如此重复下去直到获得最优结果。

第二类是“固定工作样本集”方法，它把工作样本集的大小固定在算法速度可以容忍的限度内，迭代过程中只是将剩余样本中部分“情况最糟的样本”与工作样本集的样本进行等量交换，即使支持向量机的个数超过工作样本集的大小，也不改变工作样本集的规模，而只对支持向量中的一部分进行优化。

4.2 SMO 算法

SMO 算法的全称是 Sequential Minimal Optimization，即序列最小优化。它是“固定工作样本集”的一个特例，将工作样本集的规模减到最小一两个样本。我们把每一步选择出来用来优化的两个 Lagrange 乘子所对应的样本表示为 i_1 和 i_2。

在每一步选择需要优化的样本 i_1 和 i_2 的方法对算法的性能至关重要。SMO 算法使用两层循环：外层循环选择 i_2，对于给定的 i_2，内层循环选择 i_1。外层循环首先遍历非边界样本并选出它们当中违反 KKT 条件的样本进行调整，直到非边界样本全部满足 KKT 条件为止。当某一次遍历发现没有非边界样本被调整，就遍历所有样本，以检验是否整个集合都满足 KKT 条件。如果都满足的话，算法终止，否则，继续优化并再次遍历边界样本。内层循环针对违反 KKT 条件的样本选择另一个样本与它配对优化（指优化它们的 Lagrange 乘子），选择的依据是尽量使这样一对样本能取得最大优化步长，一般选使得 $|E_1-E_2|$（E_i 为第 i 个样本的输出错误）最大的样本作为第二个样本。

SMO 算法每次在对选出的需要优化的两个样本对应的 Lagrange 乘子优化后，都要更新 β 值，但是该值有可能是无法确定的（例如不存在 $0<\alpha_i<C$ 的样本，即样本都在边界上），这时 SMO 采用的方法是确定出 β 的上下界然后取平均值；另外，每一次迭代过程中的 β 值仅取决于上次迭代结果的两个变量的最优值，用这个 β 值判断样本是否满足迭代结果，这就可能存在某些达到最优值的样本却不满足 KKT 条件的情况，从而影响了 SMO 算法的效率。

下面我们拿一个具体的例子来说明由于 SMO 算法使用了一个阈值参数 β 而引起的问题。

4.3 举例说明 SMO 算法存在的问题

例1 考虑如下的三个样本的情况：

$$y_1=-1,y_2=y_3=+1,C=\frac{1}{4}$$

$$KernalMatrix=\begin{bmatrix}1&0&0\\0&1&2\\0&2&6\end{bmatrix}$$

设初始时 $\alpha_1=\alpha_2=\alpha_3=0,\beta=0$（SMO 算法的初始点），根据前面的知识计算得到 $F_1=1,F_2=F_3=-1$，可见这三个样本都违反了优化条件(6a)。设 SMO 选择样本1和样本2来做优化，样本3对应的 Lagrange 乘子固定为0不变。优化以后的 Lagrange 乘子为 $\alpha_1=\alpha_2=C,\alpha_3=0$。在该点上我们计算得到 $F_1=3/4,F_2=-3/4,F_3=-1/2$，显然优化条件是满足的。这时 SMO 选 $\beta=(F_1+F_2)/2=0$。假如用这个 β 值来判断优化条件(6a)，第三个训练样本就违反了 KKT 优化条件，但在下面的讨论中我们将看到这里对应的 $b_{low}=-3/4,b_{up}=-1/2$，而 $b_{low}<b_{up}$，实际上是没有违反 KKT 优化条件的。

从上面的例子，我们可以清楚地看到由于 SMO 使用了一个特殊的阈值参数 β，致使算法在该终止时不能正常终止，从而带来许多额外的开销。

5. 改进的 SMO 算法

5.1 两个阈值参数 b_{low} 和 b_{up}

针对由于 SMO 算法使用了一个阈值参数而引起的问题，我们设想可以用两个阈值参数来代替。对(6)式做进一步的推导。

根据(6)式中 α 不同的值可以定义下面的样本集：$I_0=\{i:0<\alpha_i<C\}$；$I_1=\{i:y_i=1,\alpha_i=0\}$；$I_2=\{i:y_i=-1,\alpha_i=C\}$；$I_3=\{i:y_i=1,\alpha_i=C\}$；$I_4=\{i:y_i=1,\alpha_i=0\}$，所以上面的(6a)，(6b)，(6c)式可以重写为：

$$\beta\leqslant F_i\ \forall\ i\in I_0\cup I_1\cup I_2;$$

$$\beta\geqslant F_i\ \forall\ i\in I_0\cup I_3\cup I_4$$

定义2 $b_{up}=\min\{F_i:i\in I_0\cup I_1\cup I_2\}$

$b_{low}=\max\{F_i:i\in I_0\cup I_3\cup I_4\}$

那么 KKT 优化条件转化为：

$$b_{low}\leqslant b_{up} \tag{7}$$

注意，原问题中的阈值参数 b 和对偶问题中的乘子 β 在优化点是相等的。因此，在本文中我们认为 b 和 β 是一样的。

如果样本对(i，j)所对应的 Lagrange 乘子满足以下两式中的任何一个，我们认为样本对(i，j)定义了一个冲突。

$$F_i>F_j \quad i\in I_0\cup I_3\cup I_4 \quad j\in I_0\cup I_1\cup I_2$$

$$F_i<F_j \quad i\in I_0\cup I_1\cup I_2 \quad j\in I_0\cup I_3\cup I_4$$

注意，当在整个样本集上不存在任何的样本对(i，j)能够定义一个冲突的时候，我们认为所有的 α 都满足了优化条件。

5.2 对阈值参数 b_{low} 和 b_{up} 的维护和更新

假设，在任何时候，F_i 对所有的样本 i 都是有效的。让 i_{low} 和 i_{up} 表示如下的样本。

（下转第133页）

@)，则 W_{i_0} 为 $R@$ 的关键策略。

3.5 关键策略的生成过程

针对最优策略或满意策略，寻找关键策略，对系统进行重点管理。

(1)建立系统矛盾蕴含系统[2,5,6]。

(2)根据共轭性分析 N 的结构。

$N=hrN\otimes sfN=reN\otimes inN=apN\otimes ltN=ps(c)N\otimes ng(c)N$

(3)相关分析：根据 N 的结构，利用相关网，列出与目标特征 c 相关的关键特征 $c_1,c_2,\cdots,c_m$。

(4)蕴含分析：利用蕴含系方法，寻找关键特征 $c_1,c_2,\cdots,c_m$ 的最上位特征 $c_{i_1},c_{i_2},\cdots,c_{i_p}$。

(5)根据问题现状，确定研究对象 N 关于最上位特征的量值，写出相应的物元 $R_1,R_2,\cdots,R_r$。

(6)进行物元变换，得到若干个关键策略。

$T_jR_i=R'_j(j=1,2,\cdots,p;i_j=i_1,i_2,\cdots,i_p)$

(7)针对扰动物元，制定防范策略(前馈控制)与补偿策略(反馈控制)进行协调，请参见文[5]。

结束语 策略模型与生成的形式化研究对提高基于知识的系统的性能起到重要的作用，对知识管理系统的研究也有着指导意义。可拓集是近年来提出的一种解决不相容问题的集合理论，本文利用可拓集理论在策略的模型、形式化生成方法等方面作了探索，初步的研究表明可拓集和物元理论可为策略的生成方法及策略协调等的研究提供了新的思路。在"CRM 中客户本体的建立方法及共享机制"课题中，我们利用基于可拓集的策略模型分析了领域客户本体模型的建立方法，探讨了本体中术语语义不一致问题，取得了一定的效果。目前这些决策模型的可操作性还不太强，需要大量领域专家的参与，这需要今后进一步的研究。

参 考 文 献

1 Cai Wen, He Bin, Zhang Yingli, Yang Chunyan. Key Strategy and Coordination Problem [J]. Shanghai: In: The 3rd Intl. Conf. On Management, (ICM'98)

2 蔡文，杨春燕，林伟初. 可拓工程方法[M]. 北京：科学出版社，1997

3 王洪伟. 关键策略的物元模型[J]. 系统工程理论与实践，1998，18(2)：106～109

4 李习彬. 一般决策模型与概念性决策模型[J]. 论文集：科学决策与系统工程. 中国科学技术出版社，1990

5 王洪伟. 策略协调问题的研究[D]：[广东工业大学硕士学位论文]. 1999

6 杨益民，魏仲俊. 关键矛盾法及其在 CSE 中的应用[J]. 系统工程理论与实践，1998，18(1)：99～97

(上接第129页)

定义3 $F_{i\_low}=b_{low}=\max\{F_i:i\in I_0\cup I_3\cup I_4\}$

$F_{i\_up}=b_{up}=\min\{F_i:i\in I_0\cup I_1\cup I_2\}$

这样，检查样本 i 是否满足优化条件就很容易了。例如，假设 $i\in I_1\cup I_2$，我们只需要检查条件：$F_i<F_{i\_low}$。假如这个条件满足，那么就存在一对违反 KKT 条件的样本对(i,i_{low})。

这样，在我们循环遍历所有样本时，对于一个给定的样本 i，首先计算 F_i，然后应用当前的(b_{low},i_{low})，(b_{up},i_{up})来检查样本 i 是否满足优化条件。如果满足，F_i 被用来更新(b_{low},i_{low})或(b_{up},i_{up})。如果不满足，我们就选用(i,i_{low})或(i,i_{up})作为本次优化的样本对。例如，如果 $i\in I_1\cup I_2$并且 $F_i<b_{low}$，即样本 i 违反了优化条件，这时我们选择样本对(i,i_{low})作为本次优化的两个样本。如果，样本 i 满足优化条件的话，那么我们就用 F_i 来更新(i_{up},b_{up})。即，如果 $F_i<b_{up}$，那么 $i_{up}:=i$ 并且 $b_{up}:=F_i$。

这样在 SMO 算法中每次外层循环选择 i_2时我们不再用(6)式作为判优条件，而是用(7)式作为判优条件，这样就避免了由于不正确的值 β 值而引起的不必要的开销。

6. 实验结果

本部分我们比较传统的 SMO 算法和我们改进的 SMO 算法之间的性能。我们实验的硬件环境是128兆内存，Pentium Ⅱ 450MHz 的机器。软件环境是 Win2000 Professional+Matlab6.1。我们选用一阶多项式函数作为核函数，误差控制参数 C=0.10。

我们选用的数据集是 UCI 的 votes 和我们自己的 IGA 肾病诊断数据。

表1 数据集属性

数据集	样本维数	样本数
Votes	16	435
IGA 肾病	14	153

表2 votes 数据集运行效果

算法	运行时间(s)
SMO 算法	23.7
改进的 SMO 算法	15.6

表3 IGA 肾病数据集运行效果

算法	运行时间(s)
SMO 算法	16.9
改进的 SMO 算法	9.8

考虑到我们的 SMO 程序是用 Matlab 实现的，Matlab 是一种解释型的语言，执行效率比较低，如果改用 C 语言编程，速度会更快一点。

结束语 本文通过对 SVM 分类算法的分析，以及对其中最常用的 SMO 算法的深入研究，给出一种改进的 SMO 算法。用标准的 UCI 数据以及我们的肾病诊断数据测试的结果表明，改进的 SMO 算法在性能上优于传统的 SMO 算法。

参 考 文 献

1 Vapnik V. Nature of Statistical Learning Theory. John Wiley and Sons, Inc., New York, in preparation

2 Burges J C. A Tutorial on Support Vector Machines for Pattern Recognition. Bell Laboratories, Lucent Technologies. 1997

3 Keerthi S S, et al. Improvements to Platt's SMO Algorithm for SVM Classifier Design

4 Keerthi S S, et al. A Fast iterative Nearest Point Algorithm for Support Vector Machine Classifier Design

5 Platt J. Sequential minimal optimization: A fastalgorithm for training support vector machines. Advances in Kernel Methods-Support Vector learning. Cambridge, MA: MIT Press, 1999. 185～208

6 边肇祺，等. 模式识别. 清华大学出版社，1988