基于支持向量机的分解合作的加权算法及其应用*)

奉国和[1] 黄榕波[2] 罗泽举[1] 朱思铭[1]

(中山大学数学与计算科学学院 广州510275)[1] (广东药学院数学教研室 广州510224)[2]

摘 要 支持向量机是基于统计学习理论的新一代学习机器。它使用结构风险最小化原则,运用核技巧,较好地解决了学习问题。当训练数据相当大时,其训练速度是制约其应用的瓶颈。本文提出了一种基于支持向量机的分解合作的加权算法并将其应用于股票指数预测,与标准算法相比较,分解合作加权算法表现出了良好的性能。

关键词 支持向量回归,分解合作加权支持向量机,股票指数

Decomposition-Cooperation Weighted Support Vector Machine and it's Application

FENG Guo-He[1] HUANG Rong-Bo[2] LUO Ze-Ju[1] ZHU Si-Ming[1]

(School of Mathematics and Computational Science, Zhongshan University, Guangzhou 510275)[1]

(Department of Mathematics, Guangdong Pharmaceutical College, Guangzhou 510224)[2]

Abstract Support vector machines (SVM) are a new-generation machine learning based on the statistical learning theory. They uses Structural Risk Minimization and the kernel trick to solve the learning problems. In this paper, we use the decomposition- cooperation weighted SVM to predict the stock index and it performs well.

Keywords Support vector machine, Decomposition-cooperation weighted SVM, Stock index

1 支持向量机理论

统计学习理论系统地研究了各种类型函数集的经验风险和实际风险之间的关系,即推广性的界。实际风险 $R(\omega)$ 和经验风险 $R_{emp}(\omega)$ 之间至少以概率1-η 满足下列关系[1]:

$$R(\omega)\leqslant R_{emp}(\omega)+\psi(n/h) \tag{1.1}$$

其中,h 是函数集的 VC 维,n 是样本数,η 是满足$0<\eta<1$的参数,$\psi(n/h)$称作置信范围。统计学习理论把函数集构造为一个函数子集序列,使各个子集按照 VC 维的大小排列;在每个子集中寻找最小经验风险,在子集间折衷考虑经验风险和置信界限,使得实际风险最小。这种思想称为结构风险最小化。而支持向量机是基于统计学习理论的。

给定训练集 $G=\{(x_i,y_i)\}_{i=1}^{n}$,其中 $x_i\in R^d$,$y\in R$。目的是确定一个基于训练集 G 的函数 $f(x)$来逼近未知的函数 $g(x)$,SVM 考虑如下形式的逼近函数:

$$y=f(x)=(\omega\cdot\phi(x))+b \tag{1.2}$$

其中$(.)$表示在高维特征空间 Ω 中的内积,$\phi:x\to\Omega$,$b\in R$,由数据集 G 来确定系数 ω,b 是通过最小化如下回归风险函数,

$$R_{reg}(f)=C\sum_{i=1}^{n}L_\varepsilon(f(x_i)-y_i)+\frac{1}{2}(\omega\cdot\omega) \tag{1.3}$$

其中 C 是常量,$L_\varepsilon(.)$是损失函数并定义为:

$$L_\varepsilon(f(x)-y)=\max\{0,|f(x)-y|-\varepsilon\} \tag{1.4}$$

为了求出 ω,b,引入松弛变量 $\xi_i^+\geqslant 0$,$\xi_i^-\geqslant 0$,即化为求函数

$$\Phi(\omega,\xi)=\frac{1}{2}\|\omega\|^2+C\sum_{i}(\xi_i^++\xi_i^-) \tag{1.5}$$

在约束条件下

$$\begin{cases} y_i-\langle\omega\cdot x\rangle-b\leqslant\varepsilon+\xi_i^+ \\ \langle\omega\cdot x\rangle+b-y_i\leqslant\varepsilon+\xi_i^- \\ \xi_i^+\geqslant 0 \\ \xi_i^-\geqslant 0 \end{cases} \tag{1.6}$$

的最小值。通过引入拉格郎日乘子,得到如下拉格郎日泛函[2,3]:

$$\min_{\omega,b}\max_{\alpha,\beta}L(\omega,b,\alpha,\alpha^*)=\frac{1}{2}\|\omega\|^2+C\sum_{i}(\xi_i^++\xi_i^-)-\sum_{i=1}^{n}\alpha_i(\langle\omega\cdot\phi(x)\rangle+b-1+\xi_i^+)-\sum_{i=1}^{n}\alpha_i^*(\langle\omega\cdot\phi(x)\rangle+b-1+\xi_i^-)-\sum_{i=1}^{n}(\beta_i\xi_i^++\beta_i^*\xi_i^-) \tag{1.7}$$

化为在约束条件

$$\sum_{i=1}^{n}(\alpha_i-\alpha_i^*)=0,0\leqslant\alpha_i\leqslant C,\ 0\leqslant\alpha_i^*\leqslant C,i=1,2,3,\cdots,n \tag{1.8}$$

下的对偶问题(Dual Problem)[2,3],

$$\max_{\alpha,\alpha^*}W(\alpha,\alpha^*)=\max_{\alpha,\alpha^*}-\frac{1}{2}\sum_{i=1}^{n}\sum_{j=1}^{n}(\alpha_i-\alpha_i^*)(\alpha_j-\alpha_j^*)k(x_i,x_j)+\sum_{i=1}^{n}\alpha_i(y_i-\varepsilon)-\alpha_i^*(y_i+\varepsilon) \tag{1.9}$$

所要求的回归方程为:

$$f(x)=\langle\omega\cdot\phi(x)\rangle+b=\sum_{i=1}^{n}(\alpha_i^*-\alpha_i)k(x_i,x)+b^* \tag{1.10}$$

其中 $k(x,y)=(\phi(x)\cdot\phi(y))$称作核函数,文中我们采用 RBF 核函数:$k(x,y)=\exp(-1/\sigma^2(x-y)^2)$,$\sigma$ 为 RBF 核的宽度。

由于所求的问题最后化为一个凸 QP 的求解问题,具有全局最优性,其中 $\alpha_i-\alpha_i^*\neq 0$所对应的 x_i 称为支持向量。

该算法在样本数较多时,计算时间和占用的内存是求解 SVM 的主要瓶颈,基于此原因我们在保证泛化精度的前提下,提出了分解合作加权支持向量机算法(Decomposition-cooperation weighted Support Vector Machines,DCWSVM),该算法大大地降低了系统的复杂性,明显地提高了计算速度和减少了内存的开销。

*)国家自然科学基金资助项目(基金编号:10371135)。奉国和 博士生,主要研究领域为人工智能,机器学习;黄榕波 博士,研究领域为人工智能,机器学习;罗泽举 博士生,主要研究领域为人工智能,机器学习;朱思铭 教授,博导,主要研究领域为动力系统,人工智能。

2 DCWSVM 模型及其算法

2.1 DCWSVM 模型

该模型分两步：第一，将训练集分解成子集，在子集上产生支持向量；第二步，集合第一步挑选的支持向量进行加权支持向量回归，产生决策函数，可以用图表来描述该模型。

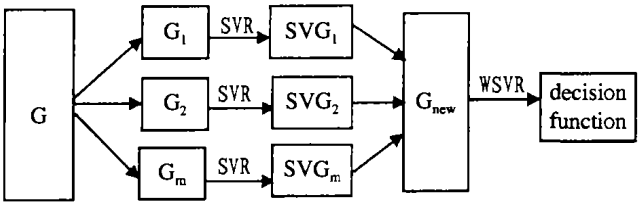

图1 DCWSVM 算法模型

图中训练样本集 $G=\{(x_i,y_i)\}_{i=1}^{n}$，其中 $x_i\in X^d$，$y\in R$，$G_1,G_2,\cdots,G_m$ 为训练子集，且 $G_1\cup G_2\cup\cdots\cup G_m=G$，$G_i\cap G_j=\varnothing$，$i\neq j$，$SVG_1,SVG_2,\cdots,SVG_m$ 为在各个 G_i 上产生的支持向量集，其元素个数分别为 $L_1,L_2,\cdots,L_m$，再将 $L_1+L_2+\cdots+L_m$ 个数据顺次并成一个新的训练集 G_{new}。在新的训练集上进行加权支持向量回归，得到所需要的预测模型。该算法的具体实现如下。

step1：分解数据集合 G 为 m 个工作子集 $G_1,G_2,\cdots,G_m$；

step2：对每一个工作子集合 G_i 进行支持向量机回归，选出支持向量 SVG_i；

step3：集合每个工作子集合的支持向量得到 G_{new}；

step4：在 G_{new} 上加权支持向量回归，产生决策函数。

2.2 加权支持向量回归

在式(1.5)中对超出 ε 管道的样本实行相同的惩罚 C，但在金融时间序列数据分析中，近期数据的重要性要大于远期数据的重要性，重要数据点要求比较小的误差，这样在上述优化问题中，可以采取不同的惩罚因子 C，以得到更准确的回归估计。我们将式(1.5)改写为

$$\Phi(\omega,\xi)=\frac{1}{2}\|\omega\|^2+C\sum_i s_i(\xi_i^+ +\xi_i^-) \tag{2.1}$$

s_i 为加权系数，式(2.1)的约束条件与式(1.6)相同，得到对偶最优化问题

$$\max_{\alpha,\alpha^*}W(\alpha,\alpha^*)=\max_{\alpha,\alpha^*}-\frac{1}{2}\sum_{i=1}^{n}\sum_{j=1}^{n}(\alpha_i-\alpha_i^*)(\alpha_j-\alpha_j^*)k(x_i,x_j)+\sum_{i=1}^{n}\alpha_i(y_i-\varepsilon)-\alpha_i^*(y_i+\varepsilon)] \tag{2.2}$$

约束条件 $\sum_{i=1}^{n}(\alpha_i-\alpha_i^*)=0,0\leqslant\alpha_i\leqslant Cs_i,0\leqslant\alpha_i^*\leqslant Cs_i,i=1,2,\cdots,n$ (2.3)

在本文中我们采用指数加权[4]，即 $s_i=\frac{1}{1+\exp(r-2r_i/n)}$，其中 r 为控制上升速率的参数。

2.3 计算时间的比较

由于 SVM 的计算主要涉及到海森矩阵的计算，而海森矩阵的每个元素是 $y_iy_jk(x_i,x_j)$，我们将它看作一个基本计算单元。由于每个工作子集产生的支持向量数只占工作子集样本数目的少部分，我们假定每个工作子集产生的支持向量个数占工作子集的比例为 $r_i(i=1,2,\cdots,m)$，并设 $r=\max\{r_1,r_2,\cdots,r_m\}$，于是有如下的定理。

定理1 设 DCWSVM 的计算时间为 T_{DCWSVM}，SSVM 的计算时间为 T_{SSVM}，假定 $r<0.7$，则

$T_{DCWSVM}<T_{SSVM}$。

证明：由于每个工作子集 G_i 的样本数为 l 个，则其计算单元数为 l^2，这样 m 个工作子集共有计算单元数为 $l^2\times m$，有在 G_{new} 上 $(\sum_{i=1}^{m}l\times r_i)^2$ 个计算单元，这样分解合作算法总共要计算的单元数为 $l^2\times m+(\sum_{i=1}^{m}l\times r_i)^2$，而 SSVM 算法的计算单元数为 $(l\times m)^2$，又因为有不等式 $l^2\times m+(\sum_{i=1}^{m}l\times r_i)^2<l^2\times m+l^2\times r^2\times m$ 和 $r^2<\frac{m-1}{m}$（因为 $r<0.7$）成立，从而 $(l^2\times m+l^2\times r^2\times m^2)<(l\times m)^2$ 成立，即 $T_{DCWSVM}<T_{SSVM}$。

3 实验结果

3.1 数据采集

实验中我们采用2002年1月4日至2004年4月13日的上证180指数共543个数据点，通过某日前一段时间收盘价的历史数据来预测某日的收盘价格，也就是估计下面的动态系统：

$$x_t=f(x_{t-1},x_{t-2},\cdots,x_{t-k})$$

其中 x_t 为在时间 t 的股票指数，k 为滞后时间值，这样一来，整个系统的输入就是时间 t 前 k 个历史数据，而输出则是时间 t 时的值。在本文中我们取 $k=5$，与一周的开市时间一样。

3.2 性能评价指标

本文中我们采用的评测标准是以下2个指标[5]：

NMSE(normalized mean squared error)：

$$NMSE=1/(\delta^2 n)*\sum_{i=1}^{n}(a_i-p_i)^2$$

$$\delta^2=1/(n-1)*\sum_{i=1}^{n}(a_i-a-)^2$$

MAE(mean absolute error)

$$MAE=1/n*\sum_{i=1}^{n}|a_i-p_i|$$

其中 a_i,p_i 分别表示实际值和预测值。

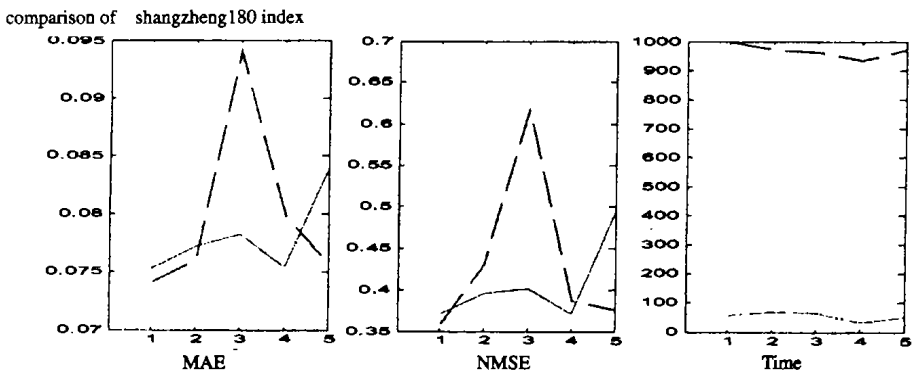

图2 上证180指数实验结果

3.3 实验结果

在数据进行模拟前先对数据预处理，数据归一化，归一化后的数据分为两部分，第一部分为训练数据，第二部分为测试数据，测试数据约占整个数据的10%左右。在模型建立中，我们使用的核函数为径向基函数(RBF)，同时将标准的算法简称为SSVM，分解合作加权算法简称为DCWSVM，实验结果如图2所示，图中横坐标的1至5个点表示(σ,c,ε)参数的不同组合，σ表示核函数RBF的参数，c表示调整参数，ε表示误差。图中实线表示DCWSVM算法结果，虚线表示SSVM算法结果，左图是MAE的比较，中图是NMSE的比较，右图为计算时间的比较。

从图2数据可以看出，在预测误差保持大体一致的情况下，DCWSVM算法在运行速度方面要大大优于SSVM算法；这也说明本文中提出的DCWSVM算法是有效的。

结论 运用基于支持向量机的分解合作加权算法对股票指数时间序列数据作预测，实验和理论都证明在保证泛化能力的情况下，其运算速度比标准算法要快，同时也大大地减少了内存的开销。

参考文献

1 Vapnik V N. Statistical learning theory [M]. New York: Wiley, 1998

2 Vapnik V N. The Nature of Statistical Learning Theory [M]. New York: Springer, 1999

3 Cristianini N, Tayor J S. An introduction to support vector machines and other kernel-based learning methods [M]. New York: Cambridge Press, 2000

4 Tay F E H, Cao L J. Modified support vector machines in financial time series forecasting [J]. Neurocomputing, 2002, 48: 847～861

5 Tay F E H, Cao L J. Application of support vector machines in financial time series forecasting [J]. omega, 2001, 29: 309～317

(上接第78页)

所有顶点的颜色调整，由此造成NP难(NP-Hardness)问题，此算法不会引起该问题而在多项式时间内得出可行解或无解的结果。

例：如图6所示，求图G是否为三色图。

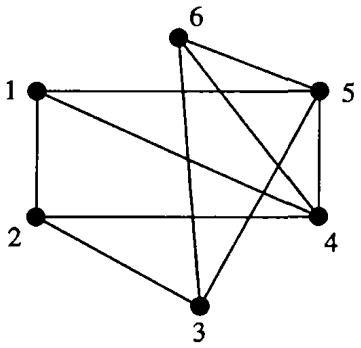

图6 无向图G

求解过程如下：

Step1 输入图G的邻接表如图7所示，并初始化集合$G=(V,E)$，$T_\triangle=\varphi$，$E_\triangle=\varphi$，$\rightarrow E_\triangle=E$。

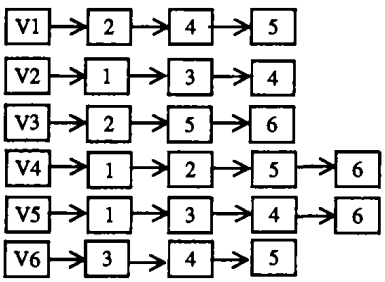

图7 图G的邻接表

Step2 首先从顶点$V1$开始遍历$V1\rightarrow V2$，然后再从$V2$开始遍历第二个结点$V3$(因为链表$V2>V1$，所以跳过$V1$到下一结点)，最后遍历以$V3$开头的链表，查找是否有$V3\rightarrow V1$的边。此时形成三角形的长度为3。表中可以看出不存在$V1$到$V3$的边，因此返回结点$V2$的下一邻边$V4$即$V2\rightarrow V4$。再从$V4$出发，可以找到$V4\rightarrow V1$，也就形成了$V1\rightarrow V2\rightarrow V4\rightarrow V1$的三角环。这样$T_\triangle=\{\triangle 124\}$，$E_\triangle=\{(1,2),(1,4),(2,4)\}$，$\rightarrow E_\triangle=E-E_\triangle$。其次再继续从$V1$的下一个邻边$V1\rightarrow V4$，找到$\triangle 145$，因此$T_\triangle=\{\triangle 124,\triangle 145\}$，$E_\triangle=\{(1,2),(1,4),(2,4),(1,5),(4,5)\}$，$\rightarrow E_\triangle=E-E_\triangle$。然后用同样的方法依次遍历以$V2,V3,V4,V5,V6$开头的链表。实际上只需到结点$V4$就可终止了，因为以$V5$和$V6$开头的结点构成的三角环长度$<3$了，它们所包含的三角环已在前面结点找到了。这样最终结果集为：$T_\triangle=\{\triangle 124,\triangle 145,\triangle 356,\triangle 456\}$，$E_\triangle=\{(1,2),(1,4),(2,4),(1,5),(4,5),(3,5),(3,6),(5,6),(4,6)\}$，$\rightarrow E_\triangle=E-E_\triangle=\{(2,3)\}$。

Step3 判断$T_\triangle$，得出集合中的三角形不构成四面体，所以输出给定的图G是3-Colorable的。

Step4 给出可满足条件的着色方案(进行3-Coloring)。首先，从$T_\triangle$中取出$\triangle 124$，假设$C_1=\{C_0\}$，$C_2=\{C_1\}$，$C_4=\{C_2\}$，$\because (2,3)\in\rightarrow E_\triangle$，$\therefore C_3=C-\{C_1\}=\{C_0,C_2\}$。然后取出$\triangle 145$，$\because C_1$和$C_4$已取色，$C_1\cap C_4\cap C_5=\phi$，$\therefore C_5=\{C_1\}$。再取出$\triangle 356$，又$\because C_5$已取色，$(V_5,V_6)\in E_\triangle$，$C_5\cap C_6=\varphi$，$\therefore C_6=C-C_5=\{C_0,C_2\}$。最后取出$\triangle 456$，$\because C_4$和$C_5$已取色，$C_4\cap C_5\cap C_6=\varphi$，$\therefore C_6=\{C_0\}$，又$\because C_3\cap C_6=\phi$，$\therefore C_3=\{C_2\}$。所以得出其中符合条件的一种着色方案为：$C_1=\{C_0\}$，$C_2=\{C_1\}$，$C_3=\{C_2\}$，$C_4=\{C_2\}$，$C_5=\{C_1\}$，$C_6=\{C_0\}$。

参考文献

1 Karp R M. Reducibility among combinatorial problems. In: Raymond E. Miller and James W. Thather, eds. Complexity of Computer Computations, Plenum Press, 1972. 85～103

2 Khot S. Hardness results for coloring 3-colorable 3-uniform hypergraphs. In: Proc. of the 43rd Annual IEEE Symposium on Foundations of Computer Science, 2002

3 Guruswami V, Khanna S. On the hardness of 4-coloring a 3-colorable graph. IEEE, 2000

4 Bogdanov A, Obata K, Trevisan L. A lower bound for testing 3-colorability in bounded-degree graphs. In: Proc. of the 43rd Annual IEEE Symposium on Foundations of Computer Science, 2002

5 Guruswami V, Hastad J, Sudan M. Hardness of approximate hypergraph coloring. IEEE, 2000

6 Guruswami V, Hastad J, Sudan M. Hardness of approximate hypergraph coloring. IEEE, 2000

7 Beigel R, Eppstein D. 3-coloring in time O(1.3446^n): a no-MIS algorithm. IEEE, 1995