文章编号: 1006－4729(2008)04－0381－04

最小二乘支持向量机分类问题的算法实现

周建萍[1,2], 郑应平[2], 王志萍[1]

(1.上海电力学院 电力与自动化工程学院,上海 200090;
2.同济大学 电子与信息工程学院,上海 200092)

摘 要: 介绍了支持向量机理论、常用的支持向量机内积核函数以及最小二乘支持向量机算法.采用最小二乘法实现了支持向量机分类算法.数字仿真结果表明,该算法的识别正确率可达 100%.

关键词: 最小二乘法;支持向量机;核函数;分类
中图分类号: O174.6 文献标识码: A

Arithmetic Realization of Least Square Support Vector Machine Classification

ZHOU Jian-ping[1,2], ZHENG Ying-ping[2], WANG Zhi-ping[1]

(1. School of Electric Power & Automation Engineering, Shanghai University of Electric Power, Shanghai 200090, China;
2. School of Electronics & Information, Tongji University, Shanghai 200092, China)

Abstract: Theories of support vector machine (SVM), the kernel function of commouly-used (SVM), and the least square algorithm support vector are introduced. Support vector machine classification is realized by least square algorithm and numerical simulation is performed. Simulation results show that recognition rate of the proposed method is up to 100%.

Key words: least square algorithm; support vector machine (SVM); kernel function; classification

目前,人工神经网络(NN)已经广泛应用于工业过程建模和控制中,但是 NN存在着一些缺陷:网络结构需要事先指定或应用启发式算法在训练过程中加以修正,而这些启发式算法难以保证网络结构的最优化;网络权值的调整方法存在局限性,如训练可能过早结束,权值衰退等,容易陷入局部最优,甚至无法得到最优解;过分依赖学习样本的数量和质量,学习样本数据仅是实际的高维输入空间中的稀疏分布,而且即使得到了高质量的过大样本也会增加算法的训练时间;优化目标是基于经验风险的最小化,这只能保证学习样本点的估计误差最小,泛化性能不强,存在“过拟合”问题[1].

支持向量机(Support Vector Machine, SVM)是由 Vapnik等人在统计学习理论的基础上建立起来的一种机器学习新方法,着重研究小样本情况下的统计学习规律[2]. SVM通过结构风险最小化原理来提高泛化能力,较好地解决了小样本、非

收稿日期: 2008－02－22

作者简介: 周建萍(1978－),女,在读博士,江西萍乡人.主要研究方向为智能控制与故障诊断等. E-mail: jpzhou820@yahoo.com.cn

基金项目: 上海高校选拔培养优秀青年教师科研专项基金(Z2006-78);上海市重点学科建设项目(P1301).

线性、高维数、局部极小等实际问题，已在模式分类、回归预测、概率估计及控制理论等领域得到应用.在这些应用领域，SVM表现出了比传统学习机器更优秀的分类能力和泛化性能，被公认为是人工神经网络的替代方法.

但是，这种标准 SVM算法的复杂度不依赖于输入空间的维数，而是依赖于样本数据的个数.样本数据越大，求解相应的二次规划问题越复杂，计算速度越慢，存在着鲁棒性、稀疏性和大规模运算问题.目前解决此问题的改进方法有多种，其中比较成功的是 Suykens等提出的最小二乘支持向量机（Least Square Support Vector Machine，LS-SVM）[3]. LS-SVM和标准 SVM的主要区别在于：目标函数中增加误差平方和项，用二次损失函数取代 SVM中的不敏感损失函数；用等式约束代替不等式约束，求解过程变成解一组等式方程，避免了求解耗时的受约束的二次型规划 QP问题，求解速度相对加快.

1 SVM理论及其核函数

1.1 SVM理论

SVM算法采用了一种核函数映射方法，其基本思想就是通过一个非线性映射，把输入空间的数据映射到一个高维特征空间中，在这一高维空间中样本是线性可分的[2].

设有 N个样本，x_i是输入向量，y_i是所属类别，d是输入空间的维数，可表示为 $\{(x_i, y_i)\}$，$x_i \in R^d$，$y_i \in \{1, -1\}$，$i = 1, \cdots, N$.

对于标准的 SVM，其分类间隔为 $2/\|\omega\|$，使分类间隔最大相当于是 $\|\omega\|^2$ 最小，因此，使分类间隔最大的优化问题可表示为二次规划问题：

$$\min_{\omega, b, \zeta} J_{LS}(\omega, \zeta) = \frac{1}{2}\|\omega\|^2 + \frac{1}{2}\gamma\sum_{i=1}^{N}\xi_i^2 \quad (1)$$

Subject to

$$y_i[\omega^T\Phi(x_i) + b] \geqslant 1 - \xi_i, \quad i = 1, \cdots, N \quad (2)$$

$$\xi_i \geqslant 0, \quad i = 1, \cdots, N \quad (3)$$

式中：$\xi_i \geqslant 0$——松弛因子，用来保证在线性不可分情况下分类的正确性（$i = 1, \cdots, N$）；

C——惩罚因子，为某个指定的常数，起到控制对错分样本惩罚程度的作用，实现在错分样本的比例和算法复杂程度之间的折衷；

Φ——非线性变换函数，可通过 Φ将非线性问题转化为某个高维空间中的线性问题，在变换空间求最优分类面.

设有非线性映射 $\Phi: R^d \to H$将输入空间的样本映射到高维的特征空间 H中.当在特征空间 H中构造最优超平面时，训练算法仅使用空间中的内积，即 $\Phi(x_i)\Phi(x_j)$，而 $\Phi(x_i)$没有单独出现.因此，如果能够找到一个函数 K使得

$$K(x_i, x_j) = \Phi(x_i)\Phi(x_j)$$

这样，高维空间的内积运算甚至在没有必要知道其变换形式的情况下，就可以用原空间中的函数实现.

1.2 常用的 SVM内积核函数

根据 Hilbert-Schmidt原理，只要一个函数 $K(x_i, x_j)$满足 Mercer条件[4]，它就对应某一变换空间中的内积.因此，在最优分类面中用适当的内积核函数 $K(x_i, x_j)$就可以实现从低维空间向高维空间的映射，从而实现某一非线性变换后的线性分类，而计算复杂度却没有增加. SVM算法通过求解上述二次规划问题来实现对样本的正确分类.

常用的内积核函数有以下 3个[5].

（1）多项式核函数

$$K(x_i, x_j) = ((x_i x_j) + \theta)^d, \quad d = 1, 2, \cdots$$

此时的 SVM是一个 d阶多项式分类器，d为用户决定的参数.

（2）Gauss核函数

$$K(x_i, x_j) = \exp\left\{\frac{\|x_i - x_j\|^2}{2\sigma^2}\right\}$$

此时的 SVM是一种径向集函数分类器.

（3）Sigmoid核函数

$$K(x_i, x_j) = \tanh(k(x_i x_j) + \theta)$$

此时的 SVM是一个单隐层感知器神经网络.

2 LS-SVM算法

2.1 LS-SVM分类算法

LS-SVM对 SVM进行了改进，它用二次损失函数取代 SVM中的不敏感损失函数，将不等式约束条件变为等式约束，寻优目标函数式应变为

$$\min_{\omega, b, \zeta} J_{LS}(\omega, \zeta) = \frac{1}{2}\|\omega\|^2 + \frac{1}{2}\gamma\sum_{i=1}^{N}\xi_i^2 \quad (4)$$

Subject to

$$y_i[\omega^T\Phi(x_i)+b]=1-\xi_i,\quad i=1,\ \cdots,\ N \tag{5}$$

式中, γ——类似 SVM中的参数 C, 用于对 $J_{LS}(\omega,\ \zeta)$进行控制.

为求解这个优化问题, 引入拉格朗日函数:

$$L_{LS}(w,\ b,\ \zeta,\ a)=J_{LS}(\omega,\ \zeta)-\sum_{i=1}^{N}a_i\{y_i[\omega^T\Phi(x_i)+b]-1+\zeta_i\} \tag{6}$$

式中, a_i—— Lagrangian乘子.

在鞍点 (极值)处, 分别对 ω, b, ζ, a_i求导并令它们等于零, 从而得

$$\frac{\partial L}{\partial\omega}=0\quad \omega=\sum_{i=1}^{N}a_i y_i\Phi(x_i) \tag{7}$$

$$\frac{\partial L}{\partial b}=0\quad \sum_{i=1}^{N}a_i y_i=0 \tag{8}$$

$$\frac{\partial L}{\partial\xi}=0\quad a_i=\gamma\xi_i \tag{9}$$

$$\frac{\partial L}{\partial a_i}=0\quad y_i[\omega^T\Phi(x_i)+b]-1+\xi_i=0 \tag{10}$$

式 (7)至式 (10)可写为一个线性系统

$$\begin{bmatrix}I&0&0&-Z^T\\0&0&0&-Y^T\\0&0&\gamma I&-I\\Z&Y&I&0\end{bmatrix}\begin{bmatrix}\omega\\b\\\xi\\a\end{bmatrix}=\begin{bmatrix}0\\0\\0\\\vec{1}\end{bmatrix} \tag{11}$$

式中: $Z=[\Phi(x_1)^T y_1,\ \cdots,\ \Phi(x_N)^T y_N]^T$

$Y=[y_1,\ \cdots,\ y_N]^T$

$\vec{1}=[1,\ \cdots,\ 1]^T$

$\xi=[\xi_1,\ \cdots,\ \xi_N]^T$

$a=[a_1,\ \cdots,\ a_N]^T$

$I\in R^{N\times N}$——单位矩阵.

消除 ω, ξ后式 (11)简化为

$$\begin{bmatrix}0&Y^T\\Y&ZZ^T+\gamma^{-1}I\end{bmatrix}\begin{bmatrix}b\\a\end{bmatrix}=\begin{bmatrix}0\\\vec{1}\end{bmatrix} \tag{12}$$

定义 $\Omega=ZZ^T=[q_{ij}]_{N\times N}$, 并且应用 Mercer条件, 该矩阵的元素可以表示为

$$q_{ij}=y_i y_j\Phi(x_i)^T\Phi(x_j)=y_i y_j K(x_i,\ x_j) \tag{13}$$

式中, $K(x_i,\ x_j)$——核函数.

式 (12)用最小二乘法即可解.

在 LS-SVM算法中, 将二次规划问题转变为线性方程组的求解, 简化了计算的复杂性. 求解上述问题后得到的最优分类函数

$$y(x)=\mathrm{sign}\left[\sum_{i=1}^{N}a_i y_i\Phi(x,\ x_i)+b\right] \tag{14}$$

2.2 LS-SVM参数的选取

本文采用 Gauss函数作为核函数. 在 LS-SVM算法中, 规则化参数 γ和 Gauss函数的标准化参数 σ一般都是根据经验选取一个固定的值. 但是针对不同的样本集, 最优的参数值是变化的, 因此在一定程度上影响了故障的诊断正确率. 经试验发现, 规则化参数 γ的取值对计算结果影响不明显, 本文取 $\gamma=10$, 而随着 σ的取值不同, 计算的结果波动较大. 根据经验和试凑取 $\sigma=1$. (仿真结果表明 γ和 σ的取值是合理的.)

3 实例仿真

利用 MATLAB7. 0[6]编程如下:

第 1步 构造 K矩阵.

```
X=2. × rand(2, 30)-1;
Y= sign( sin(X(1, :))+X(2, :));
D=2;
C=10;
[ , n ]= size( X);
K= zeros( , l l);
for i=1 :l
for j=1 :l
xi=X( ; i);
xj=X( :, j);
yi=Y( i);
yj=Y( j);
K( ,i j)= exp(-( sum(( xi- xj). 2)/D))×
yi× yj;
end
end
```

第 2步 构造二次规划模型参数 H, F, t Aeq, Beq, lb, ub

```
H=K;
Ft=-1× ones(1, l);
Aeq= Y;
Beq=0;
lb= eps× ones( ,l l);
ub= C× ones( ,l l);
```

第 3步 调用优化工具箱 quadprog函数求解二次规划.

```
OPT= optimset;
```

```
OPT. LargeScale='off';
OPT. Display='off';
[ Gamma, Obj] = quadprog( H, Ft[ ], [ ],
Aeq, Beq, lb, ub[ ], OPT);
```

第 4步 整理输出参数.

```
Alpha=Gamma(1:l1)';
```

第 5步 支持向量的分类.

```
Err=0.0000000000001;
for i=1:l
    AA=Alpha(i);
     if(abs(AA-0)<=Err)
    Flag(i)=0;
    end
    if(AA>Err)&&(AA<C-Err)
    Flag(i)=2;
    end
    if(abs(AA-C)<=Err)
    Flag(i)=1;
    end
end
```

第 6步 计算偏移量 B.

```
B=0;
counter=0;
for i=1:l
  if Flag(i)>0
    SUM=0;
      for j=1:l
        if Flag(j)>0
SUM=SUM+Alpha(j)*Y(j)*exp(-(sum
((X(:,j)-X(:,i)).^2)/D));
        end
      end
    b=Y(i)-SUM;
    B=B+b;
        counter=counter+1;
    end
end
    b=B/counter;
```

第 7步 计算最优分类函数.

```
X_test=X;
for i=1:l
  SUM2=0;
    for j=1:l
      if Flag(j)>0
SUM2=SUM2+Alpha(j)*Y(j)*exp(-
(sum((X(:,j)-X_test(:,i)).^2)/D));
      end
    end
  SUM2=SUM2+b;
  Y_test(i)=sign(SUM2);
end
```

第 8步 判断正确率.

```
Result=~abs(Y_test-Y)
Percent=sum(Result)/length(Result)
```

此程序的运行结果: 分类正确率为 100%.

4 结束语

最小二乘支持向量机是用二次损失函数取代支持向量机中的不敏感损失函数, 将不等式约束条件变为等式约束, 从而将二次规划问题转变为线性方程组的求解. 用最小二乘法实现了支持向量机分类算法, 仿真结果表明, 该算法的识别正确率为 100%.

参考文献:

[1] 程启明, 王勇浩. 基于最小二乘算法的模糊支持向量机控制器及其应用[J]. 中国电机工程学报, 2007, 27(8): 76-80.

[2] Vapnik V N. The nature of statistical learning theory[M]. New York: Springer, 1995: 1-33.

[3] Suykens J A K, Vandewalle J. Least squares support vector machine classifiers[J]. Neural Network Letters, 1999, 9(3): 293-300.

[4] Sebald D J, Buckjew J A. Support vector machine techniques for nonlinear equalization[J]. IEEE Transactions on Signal Processing, 2000, 48(11): 3 217-3 226.

[5] 翟永杰. 基于 SVM的故障智能诊断方法研究[D]. 华北电力大学, 2004.

[6] 求是科技. MATLAB7.0从入门到精通[M]. 北京: 人民邮电出版社, 2006: 71-120.