

支持向量机算法解决多类分类问题的方法研究

孟繁鹏 U201914689

(华中科技大学 人工智能与自动化学院 自动化 1905 班)

摘要: 支持向量机算法 (Support Vector Machine, 下文简称 SVM) 建立在统计学习理论基础上, 是一种解决小样本问题机器学习方法。传统支持向量机算法只能处理二分类问题, 然而在实际应用中的分类问题大多为多分类问题。本文分类介绍了几种常见的运用支持向量机算法解决多分类问题的方法, 并进行了简单的分类延伸。

关键词: 机器学习; 支持向量机; 多类分类; 二叉树;

1 引言

SVM 算法是一种是为进行小样本机器学习而提出的方法, 在 20 世纪 60 年代由前苏联数学家 Vapnik 提出。该算法牢固地扎根于统计学习理论 (也称 VC 理论) 的框架中, 该理论是 Vapnik 和 Chervonenkis 在二十世纪末的三十年中发展起来的, 描述了学习机的特性, 使学习机拥有了很强的泛化能力。迄今为止, SVM 已经被广泛的应用。最初, SVM 主要被用于 OCR (光学字符识别) 上。很快, SVM 算法就可以与用于 OCR 和对象识别任务的最佳系统竞争^[1]。如今的 SVM 算法已经被应用到各行各业^[2-4]。作为机器学习界最早提出, 也是最经典的分类算法之一, 对 SVM 算法的研究与推广从未停止。

设计 SVM 算法的最初是用来解决二分类问题的。不能直接用于解决多分类问题, 然而实际上我们遇到的往往是多分类问题。因此, 改良 SVM 算法, 使其能够准确高效地解决多类分类问题是很有必要的。本文首先介绍了 SVM 算法的基本原理和实现方法, 随后介绍了两类多种多分类方法, 其一是改进传统的二分类 SVM 构造多分类分类器, 其二是构造多个二分类器实现解决多分类问题, 包括“一对多”方法, “一对一”方法, 纠错编码方法三种方法。特别地, 本文同时介绍了几种基于二叉树从数据结构角度对构造多个分类器方法的改良, 极大程度地提高了第二类方法的效率。

2 SVM 的算法原理

支持向量机的主要思想是找到一个最优的划分超平面, 使得两个异类支持向量的间隔最大化。如果最终所有训练数据都可以被这个超平面正确分开, 并且离超平面最近的向量与超平面之间的距离最大, 则这个超平面就是我们待求的最优超平面^[5]。我们称前文中提到的离超平面最近的向量为支持向量 (support vector), 称两个异类支持向量的距离为分类间隔 (margin), 称这个超平面为最优划分超平面 (Optimal Separating Hyperplane, OSH)。如图 2.1 所示, 在一个线性可分的样本中, 实心点和圆圈分别代表两类样本, 其中的实线为一条分类线, 代表一个划分超平面, 将两类样本完全分开。

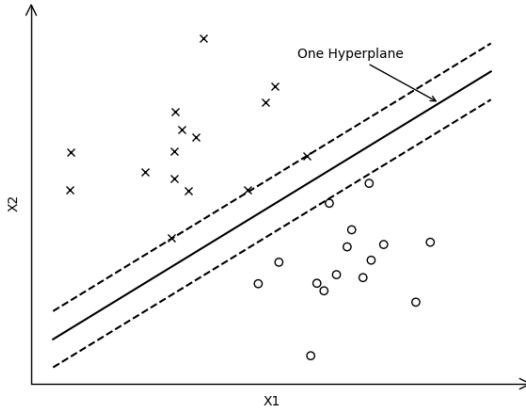


图 2.1 线性可分的 SVM 图示

而实际情况中我们常常会遇到线性可分和线性不可分两种情况的数据集，线性可分 (linear separable)，表示存在一个超平面，能够在特征空间内将两个类别的特征点完全分开。面对这两种不同的数据集，我们也常常会采取不同的模型进行处理。

2.1 线性可分的 SVM 模型

线性可分 SVM 问题的基本形如下：

$$\begin{aligned} & \min_{\omega, b} \frac{1}{2} \|\omega\|^2 \\ & \text{s.t. } y_i(\omega^T \mathbf{x}_i + b) \geq 1, \quad i = 1, 2, \dots, n. \end{aligned} \tag{2.1.1}$$

此时两个类的分类间隔为 $\frac{2}{\|\omega\|}$ ，问题转化为一个二次优化问题，可利用凸优化理论 (Convex Optimization) 进行解决。但实际上我们通常利用 Lagrange 乘子法将基本形问题转化为对偶问题：

$$\begin{aligned} & \max_{\alpha} \sum_{i=1}^m \alpha_i - \frac{1}{2} \sum_{i=1}^m \sum_{j=1}^m \alpha_i \alpha_j y_i y_j \mathbf{x}_i^T \mathbf{x}_j \\ & \text{s.t. } \sum_{i=1}^m \alpha_i y_i = 0 \\ & \alpha_i \geq 0, \quad i = 1, 2, \dots, n. \end{aligned} \tag{2.1.2}$$

公式中的 α 为每个支持向量对应的 Lagrange 乘子，当满足 KKT 条件时，可认为对偶问题与原问题距离为 0，可以等价转换。求解上式的训练方法多采用块算法或 SMO 算法^[6]，解出 α 后，求出 ω 和 b 即可得到模型，得到分类函数为：

$$\begin{aligned} f(x) &= \omega^T \mathbf{x} + b \\ &= \sum_{i=1}^m \alpha_i y_i \mathbf{x}_i^T \mathbf{x} + b \end{aligned} \tag{2.1.3}$$

2.2 线性不可分的 SVM 模型

在解决非线性问题时，我们通常使用映射将数据映射到高维度的欧几里德空间，这个空间可能是无限维，如果存在一个核函数（kernel function） K ，使得（ $\varphi(\mathbf{x})$ 为向量到高维空间的映射）：

$$K(\mathbf{x}_i, \mathbf{x}_j) = \langle \varphi(\mathbf{x}_i), \varphi(\mathbf{x}_j) \rangle = \varphi(\mathbf{x}_i)^T \cdot \varphi(\mathbf{x}_j) \quad (2.2.1)$$

则我们只需直接使用核函数，无需明确地知道映射关系^[6]。当选择合适的核函数满足 Mercer 条件，目标函数表示为：

$$\begin{aligned} & \max_{\alpha} \sum_{i=1}^m \alpha_i - \frac{1}{2} \sum_{i=1}^m \sum_{j=1}^m \alpha_i \alpha_j y_i y_j K(\mathbf{x}_i^T, \mathbf{x}_j) \\ & \text{s.t. } \sum_{i=1}^m \alpha_i y_i = 0 \\ & \alpha_i \geq 0, \quad i = 1, 2, \dots, n. \end{aligned} \quad (2.2.2)$$

对应的分类函数变为：

$$\begin{aligned} f(x) &= \omega^T \mathbf{x} + b \\ &= \sum_{i=1}^m \alpha_i y_i K(\mathbf{x}, \mathbf{x}_i) + b \end{aligned} \quad (2.2.3)$$

2.3 SVM 的主要特点

支持向量机的理论与算法研究支持向量机具有以下显著特征^[7]：

- (1) 结构简单。
- (2) 运用凸优化原理。优化问题无局部极小值点。
- (3) 稀疏表示。最优分割超平面的法向量 ω 是训练样本向量的线性组合，每个样本向量的系数反映了该样本对于最终划定分割超平面影响的权重。对分类问题的起到作用的向量系数均不为零，这些向量即为支持向量。也就是说问题的解仅与支持向量有关。
- (4) 模块化。问题清楚地分成两个模块：一个通用的学习机和与具体问题有关的核函数。这使得我们能把两个问题分开来研究，便于理论分析和工程实现。
- (5) 本质上是线性学习机。它是核函数所表示的映射指向的高维特征空间上的线性函数，因而便于理论分析。

3 SVM 的多类分类方法

常用的基于 SVM 解决多分类的方法主要可以分为两类：第一类是改变自身的约束和目标函数，将传统的二分类 SVM 模型改造成能够解决多分类问题的 SVM 模型；第二种方法是构造多个二分类 SVM 模型，进而解决多分类问题，其构造的多个二分类器不同的组合方式是这一大类中不同的方法的主要区别。特别地，一些解决方法基于二叉树这种数据结构进行改造，很大程度地提高了构造多个分类器方法的效率。

3.1 改造传统 SVM 分类器方法

这种方法通过修改传统 SVM 模型的目标函数和约束条件，依然是解决单个优化问题，但是使得结果可以输出多个子分类，从而建立 k 分类支持向量机模型^[8]。优化问题形式由传统支持向量机推广可得：

$$\begin{aligned} \min_{\omega, b} \quad & \frac{1}{2} \sum_{k \in K} \|\omega\|^2 \\ \text{s.t. } \quad & y_i \left((\omega_{y_i}^T \mathbf{x}_i + b_{y_i}) - (\omega_k^T \mathbf{x}_i + b_k) \right) \geq 1, \\ & i = 1, 2, \dots, n, k \in K. \end{aligned} \quad (3.1.1)$$

与传统 SVM 问题同理，利用 Lagrange 乘子法将问题转换为其对偶问题最终得以解决。得到的决策函数为：

$$f(x) = \arg \max_{k=1 \rightarrow K} (\omega_k^T \mathbf{x} + b_k) \quad (3.1.2)$$

3.2 朴素的构造多个分类器方法

3.2.1 “一对多”方法 (one-against-the rest, O-v-R)

一对多方法是最简单，也是最基本的实现多分类 SVM 的方案。对于 k ($k \geq 2$) 类 SVM 分类问题，把第一类作为一类，其余所有类视为另一类。很自然地将 k 分类问题转化为多个二分类问题。其分类函数为：

$$\arg \max_{k=1 \rightarrow K} \left(\sum_{i=1}^n a_i^k y_i K(\mathbf{x}, \mathbf{x}_i) + b^k \right) \quad (3.2.1)$$

其中上标 k 表示第 k 个 SVM 分类器的决策函数，此公式的直观解释就是找到某一分类器，使其异类间隔最大，这一模型中所对应的类别即为测试样本所属类别^[9]。这一方法容易造成训练样本不平衡的问题，即两类样本中一类数目远大于另一类，此时分类器往往会偏向样本数目较多的一侧^[10]。图 3.1 给出了简化的 O-v-R 方法图示，将每个分类器的间隔均视为 1：

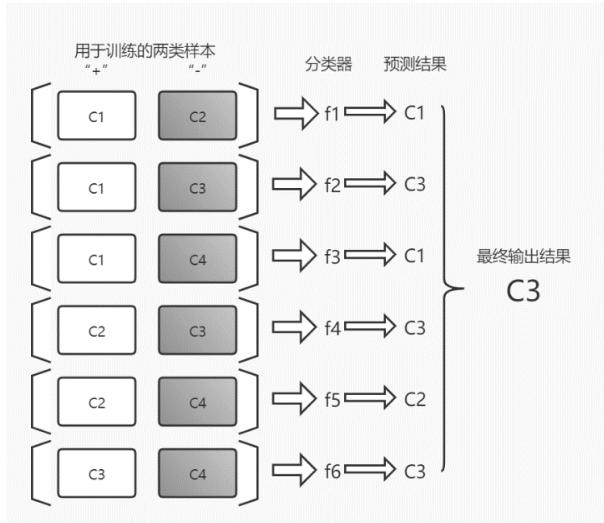


图 3.1 “一对多”方法 (O-v-R) 图示

3.2.2 “一对一”方法 (one-against-one, O-v-O)

一对二方法是在 k 类问题中对于 k 个子分类进行两两组合，构造 C_k^2 个分类器，该方法也被称为 Pairwise Method。其分类函数为：

$$\arg \max_{p=1 \rightarrow P} \sum_{k=1}^{C_k^2} S(p) \left(\sum_{i=1}^n a_i^k y_i K(\mathbf{x}, \mathbf{x}_i^k) + b^k \right) \quad (3.2.2)$$

其中 p 代表某一类别，其中 $S(p)$ 是一个关于类别 p 的映射，当 p 不包含于第 k 个分类器处理的两个类时值为 0，当 p 是第 k 个分类器所求的正类时值为 1，当 p 是第 k 个分类器所求的负类时值为 -1。这种方法规避了 0-v-R 中的样本不平衡问题，但是其缺点在于当总类别数 k 过大时，产生的子分类器过多，训练时间更长，训练速度随着类别的增加成指数降低^[11]。图 3.2 给出了简化的 0-v-0 方法图示，将每个分类器的间隔均视为 1：

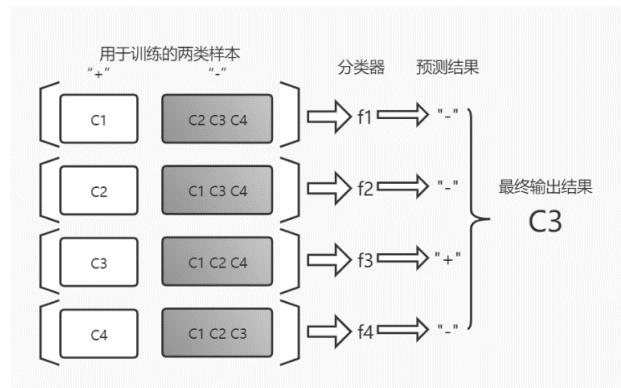


图 3.2 “一对多”方法 (0-v-R) 图示

3.2.3 纠错编码 SVM (ECOC 码)

纠错编码方法通过构造编码矩阵来实现多分类，构造 m 个分类器，每一个子分类都对应着一串编码，当输入一个测试样例时，同样可以得到一串编码，找到与该编码距离最近的编码所对应的子分类，即为该测试样例所属的子分类。常用的 ECOC 主要有两种形式，二元码和三元码：二元的表示包括正类和负类，后者在二元码的基础上，引进了第三类“停用类”。这种方法的优点在于该方法有很强的容错性，而且 ECOC 码越长，容错性越强，但增加长度的同时也会导致资源占用增加和效率降低，所以 ECOC 码长度的选取要是具体问题而分析^[12]。图 3.3 给出了 ECOC 码的图示，其中距离的选取包括但不限于海明距离和欧氏距离：

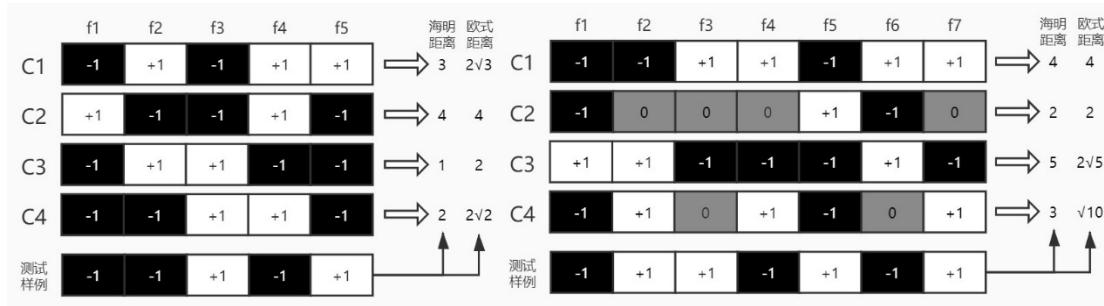


图 3.3 ECOC 编码图示，左图为二元码，右图为三元码

3.3 基于二叉树的构造多个分类器方法

构造多个分类器的方法是一种很好的解决 SVM 多分类问题的方法，但是朴素的构造多个分类器方法往往效率低，占用过多时间和空间资源，以下几种优化，基于二叉树的数据结构，通过不同的排列组合和剪枝方式，更加高效实现了 SVM 多分类问题。

3.3.1 构造完全二叉树的有向无环图方法（DAG-SVM）

有向无环图方法是基于二叉树改造多分类 SVM 的最简单基础的一个方法，该方法采用的是一对一 SVM 的思路中子分类两两任意组合训练方式，也需构造 C_K^2 个分类器。但是可以通过构造有向无环图来提高效率，该图是一个完全二叉树，由 C_K^2 个节点和 k 个叶子节点组成，其中每个节点是一个二分类器，每个叶子节点是一个子分类。其核心方法为排除法，首先将全集中首尾两个子分类进行对比，将负类排除，再在完成依次删除后新的子分类集合中执行此操作，直到该集合中只剩下最后一个子分类。对未知样本训练时，从根节点开始进行比较分裂，只需 $k-1$ 步即可完成分类，和传统的一对一方法相比进行了大幅度的剪枝，极大提高了分类速度^[13]。图 3.4 给出了 DAG-SVM 方法图示：

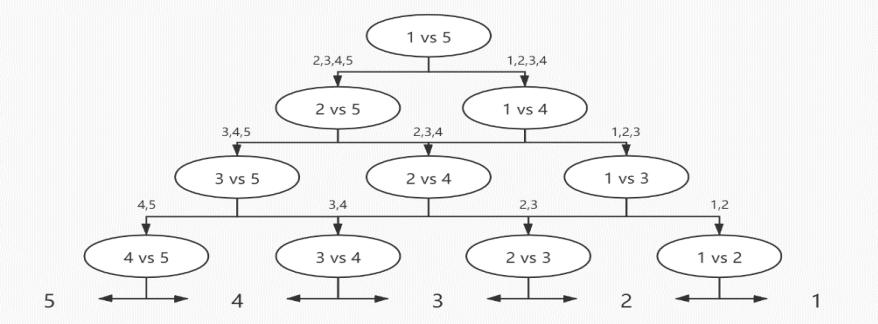


图 3.4 有向无环图方法（DAG-SVM）图示

3.3.2 按权值排序的二叉树多分类方法

按权值排序的二叉树的核心思想就是首先将区分度最大的子类分出来，一旦找到一个正类即可得到答案，否则删除该子类，再去寻找其它类中区分度最大的子分类。在这种方法中，我们首先将 k 个子分类按照某种权值排序，然后构造如图 3.5 所示的二叉树，从根节点开始进行二分类，直到找到一个正类停止。这种方法好处在于大大减少了无用步骤，大幅度提高了算法效率，但是容错性较差，错误会沿二叉树结构向后传导。下面介绍两种常见的子分类权值取法：类距离法和类分布体积法。

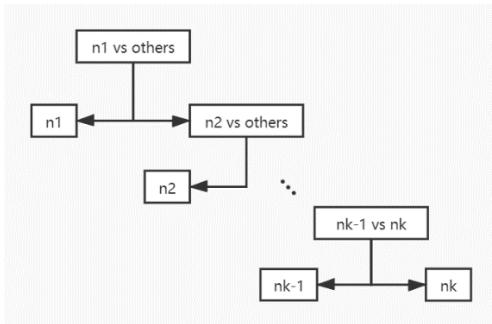


图 3.5 排序二叉树方法图示

第一种是按类距离排序，类距离排序法的核心思想求出某一子分类到其他类距离的均值，并让与其他类相隔最远的类最先分割出来。表示两个类之间的距离有很多方式，在此我们采用最具代表性的重心法：

定义 3.1（重心法） 定义类 S_k 的重心 \mathbf{x}_k 为类 S_k 中所有样本向量的均值，重心的计算公式为：

$$\mathbf{x}_k = \frac{1}{n_k} \sum_{\mathbf{x}_i \in S_k} \mathbf{x}_i \quad (3.3.1)$$

式中 n_k 为类 S_k 的样本数，两类的重心之间的欧氏距离则为两类之间的重心距离：

$$d_{ij} = \|\mathbf{x}_i - \mathbf{x}_j\| \quad (3.3.2)$$

求得类类距离后，求出每一类到其他所有子类的距离的平均值，表示这一类到其他子类的平均距离，然后将这个距离由大到小排序，距离相同则标号小的在前，（这种情况很少见）再用图 3.3 所示的二叉树结构进行分类^[14]。

第二种是按类分布体积分布，类分布体积排序核心思想是让与分布体积最大的类最先分割出来，表示分布体积有很多种方式，在此我们采取最具代表性的超球体最小包含法：

定义 3.2（超球体最小包含法） 设类 S_k 中样本为 \mathbf{x}_i ， $\mathbf{x}_i \in R^k$ ，其中 R^k 表示第 k 类样本的样本空间， \mathbf{x}_k 表示样本集的重心，则包含类全部 S_k 样本的最小超球体半径为：

$$r_k = \max_{\mathbf{x}_i \in R^k} \{\|\mathbf{x}_k - \mathbf{x}_i\|\} \quad (3.3.3)$$

由于球体的体积与半径的立方成正比，所以我们可以利用对半径大小的排序来实现分布体积大小的排序。图 3.6 即为超球体包含的图示。

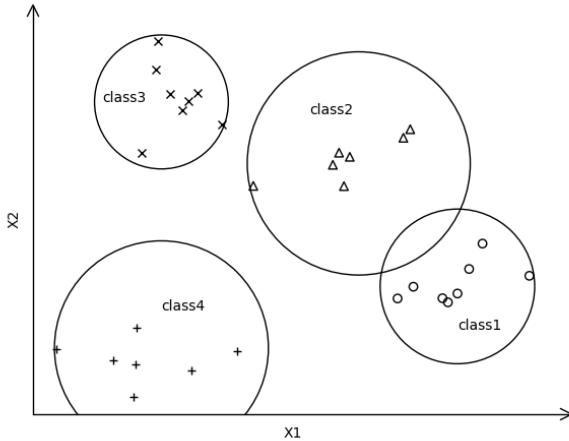


图 3.6 超球体最小包含法图示

求得分布体积后，将得到的分布体积由大到小排序，体积相同则标号小的在前，（这种情况很少见），再用图 3.3 所示的二叉树结构进行分类^[14]。

4 总结

本文中讨论了多种关于 SVM 多分类的方法，下面将集中对将本文中提到的几种方法进行总结和对比，以供读者在不同的实际情况下进行方法的选择，详情如表 4.1 所示：

改造二分类器	效率高，但是准确率欠佳	
构造多个分类器	“一对多”方法	容易出现训练样本不平衡的问题
	“一对一”方法	当类别较多时，占用时间空间资源较多
	纠错编码法	容错能力强，但占用时间空间资源较多
	基于二叉树构造	
	构造完全二叉树	效率高，但是容错性差
	按权值排序构造	在前一方法基础上进一步提高效率，但是容错性差

表 4.1 不同多分类 SVM 方法的对比

致谢

首先，本文是本人的第一篇学术论文，在撰写本文的过程中，向自阜 1901 班屈文杰同学请教了很多问题，得到了很多方面的指点和帮助，对我完成此篇论文有着非常大的帮助，在此特表感谢。

此外，感谢谭山老师和自动化学院开设了这门课程，让我对科技论文的写作和各种工具的使用有了一个初步的了解，并且能够加以实践，虽然文章内容十分简陋，但是在撰写文章的过程中，但是关于科技论文的写作我学到了很多东西，也为接下来的学习和研究打下了一点。

参考文献

- [1] Smola A J, Scholkopf B. A tutorial on support vector regression[J]. Statistics and Computing, 2004, 14(3): 199-222.
- [2] 叶林, 刘鹏. 基于经验模态分解和支持向量机的短期风电功率组合预测模型[J]. 中国电机工程学报, 2011, 31(31): 102-108.
- [3] 张强强, 苏变萍, 李敏. 基于卷积神经网络的 SVM 疾病症状分类模型[J]. 传感器与微系统, 2020, 39(3): 64-67,71.
- [4] 胡海青, 张琅, 张道宏, et al. 基于支持向量机的供应链金融信用风险评估研究[J]. 软科学, 2011, 25(5): 26-30,36.
- [5] 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.
- [6] Burges C J C. A Tutorial on Support Vector Machines for Pattern Recognition[J]. Data Mining and Knowledge Discovery, 1998, 2(2): 121-167.
- [7] 王国胜 钟 北 信 博. 支持向量机的理论与算法研究[J], 2007.

- [8] Franc V, Hlaváć V. Multi-class support vector machine[C]. Object recognition supported by user interaction for service robots, 2002: 236-239.
- [9] Hsu C, Lin C. A comparison of methods for multiclass support vector machines[J]. IEEE Transactions on Neural Networks, 2002, 13(2): 415-425.
- [10] 杨智明 彭 哈 仪 博. 面向不平衡数据的支持向量机分类方法研究[J], 2009.
- [11] 郑勇涛, 刘玉树. 支持向量机解决多分类问题研究[J]. 计算机工程与应用, 2005, 41(23): 190-192.
- [12] Rocha A, Goldenstein S K. Multiclass from binary: expanding one-versus-all, one-versus-one and ECOC-based approaches[J]. IEEE Transactions on Neural Networks and Learning Systems, 2013, 25(2): 289-302.
- [13] Platt J C, Cristianini N, Shawe-Taylor J. Large margin DAGs for multiclass classification[C]. Advances in neural information processing systems, 2000: 547-553.
- [14] 唐发明 华 控 博. 基于统计学习理论的支持向量机算法研究[J], 2005.