

基于统计学习理论的支持向量机算法研究

1 理论背景

基于数据的机器学习是现代智能技术中的重要方面，研究从观测数据（样本）出发寻找规律，利用这些规律对未来数据或无法观测的数据进行预测。迄今为止，关于机器学习还没有一种被共同接受的理论框架，关于其实现方法大致可以分为三种^[3]：

第一种是经典的（参数）统计估计方法。包括模式识别、神经网络等在内，现有机器学习方法共同的重要理论基础之一是统计学。参数方法正是基于传统统计学的，在这种方法中，参数的相关形式是已知的，训练样本用来估计参数的值。这种方法有很大的局限性，首先，它需要已知样本分布形式，这需要花费很大代价，还有，传统统计学研究的是样本数目趋于无穷大时的渐近理论，现有学习方法也多是基于此假设。但在实际问题中，样本数往往是有限的，因此一些理论上很优秀的学习方法实际中表现却可能不尽人意。

第二种方法是经验非线性方法，如人工神经网络（ANN）。这种方法利用已知样本建立非线性模型，克服了传统参数估计方法的困难。但是，这种方法缺乏一种统一的数学理论。

与传统统计学相比，统计学习理论（Statistical Learning Theory 或 SLT）是一种专门研究小样本情况下机器学习规律的理论。该理论针对小样本统计问题建立了一套新的理论体系，在这种体系下的统计推理规则不仅考虑了对渐近性能的要求，而且追求在现有有限信息的条件下得到最优结果。V. Vapnik 等人从六、七十年代开始致力于此方面研究^[1]，到九十年代中期，随着其理论的不断发展和成熟，也由于神经网络等学习方法在理论上缺乏实质性进展，统计学习理论开始受到越来越广泛的重视。

统计学习理论的一个核心概念就是 VC 维(VC Dimension)概念，它是描述函数集或学习机器的复杂性或者说是学习能力(Capacity of the machine)的一个重要指标，在此概念基础上发展出了一系列关于统计学习的一致性(Consistency)、收敛速度、推广性能(Generalization Performance)等的重要结论。

统计学习理论是建立在一套较坚实的理论基础之上的，为解决有限样本学习问题提供了一个统一的框架。它能将很多现有方法纳入其中，有望帮助解决许多原来难以解决的问题（比如神经网络结构选择问题、局部极小点问题等）；同时，这一理论上发展了一种新的通用学习方法——支持向量机（Support Vector Machine 或 SVM），已初步表现出很多优于已有方法的性能。一些学者认为，SLT 和 SVM 正在成为继神经网络研究之后新的研究热点，并将推动机器学习理论和技术有重大的发展。

支持向量机方法是建立在统计学习理论的 VC 维理论和结构风险最小原理基础上的，根据有限的样本信息在模型的复杂性（即对特定训练样本的学习精度，Accuracy）和学习能力（即无错误地识别任意样本的能力）之间寻求最佳折衷，以期获得最好的推广能力(Generalization Ability)。支持向量机方法的几个主要优点有：

1. 它是专门针对有限样本情况的，其目标是得到现有信息下的最优解而不仅仅是样本数趋于无穷大时的最优值；

2. 算法最终将转化成为一个二次型寻优问题，从理论上说，得到的将是全局最优点，解决了在神经网络方法中无法避免的局部极值问题；
3. 算法将实际问题通过非线性变换转换到高维的特征空间(Feature Space)，在高维空间中构造线性判别函数来实现原空间中的非线性判别函数，特殊性质能保证机器有较好的推广能力，同时它巧妙地解决了维数问题，其算法复杂度与样本维数无关；

在 SVM 方法中，只要定义不同的内积函数，就可以实现多项式逼近、贝叶斯分类器、径向基函数(Radial Basic Function 或 RBF)方法、多层感知器网络等许多现有学习算法。

统计学习理论从七十年代末诞生，到九十年代之前都处在初级研究和理论准备阶段，近几年才逐渐得到重视，其本身也趋向完善，并产生了支持向量机这一将这种理论付诸实现的有效的机器学习方法。目前，SVM 算法在模式识别、回归估计、概率密度函数估计等方面都有应用。例如，在模式识别方面，对于手写数字识别、语音识别、人脸图像识别、文章分类等问题，SVM 算法在精度上已经超过传统的学习算法或与之不相上下。

目前，国际上对这一理论的讨论和进一步研究逐渐广泛，而我国国内尚未在此领域开展研究，因此我们需要及时学习掌握有关理论，开展有效的研究工作，使我们在这一有着重要意义的领域中能够尽快赶上国际先进水平。由于 SLT 理论和 SVM 方法尚处在发展阶段，很多方面尚不完善，比如：许多理论目前还只有理论上的意义，尚不能在实际算法中实现；而有关 SVM 算法某些理论解释也并非完美(J.C.Burges 在[2]中就曾提到结构风险最小原理并不能严格证明 SVM 为什么有好的推广能力)；此外，对于一个实际的学习机器的 VC 维的分析尚没有通用的方法；SVM 方法中如何根据具体问题选择适当的内积函数也没有理论依据。因此，在这方面我们可做的事情是很多的。

2 方法介绍

SVM 是从线性可分情况下的最优分类面发展而来的，基本思想可用图 1 的两维情况说明。图中，实心点和空心点代表两类样本，H 为分类线， H_1 、 H_2 分别为过各类中离分类线最近的样本且平行于分类线的直线，它们之间的距离叫做分类间隔 (margin)。所谓最优分类线就是要求分类线不但能将两类正确分开 (训练错误率为 0)，而且使分类间隔最大。分类线方程为 $x \cdot w + b = 0$ ，我们可以对它进行归一化，使得对线性可分的样本集 (x_i, y_i) ， $i = 1, \dots, n$ ， $x \in R^d$ ，

$y \in \{+1, -1\}$ ，满足

$$y_i[(w \cdot x_i) + b] - 1 \geq 0, \quad i = 1, \dots, n \quad (1)$$

此时分类间隔等于 $2/\|w\|$ ，使间隔最大等价于使 $\|w\|^2$ 最小。满足条件(1)且使 $\frac{1}{2}\|w\|^2$ 最小的分类面就叫做最优分类面， H_1 、 H_2 上的训练样本点就称作支持向量。

利用 Lagrange 优化方法可以把上述最优分类面问题转化为其对偶问题[2]，即：在约束条件

$$\sum_{i=1}^n y_i a_i = 0, \quad (2a)$$

和 $\mathbf{a}_i \geq 0 \quad i=1, \dots, n$ (2b)
 下对 \mathbf{a}_i 求解下列函数的最大值：

$$Q(\mathbf{a}) = \sum_{i=1}^n \mathbf{a}_i - \frac{1}{2} \sum_{i,j=1}^n \mathbf{a}_i \mathbf{a}_j y_i y_j (\mathbf{x}_i \cdot \mathbf{x}_j) \quad (3)$$

\mathbf{a}_i 为原问题中与每个约束条件 (1) 对应的 Lagrange 乘子。这是一个不等式约束下二次函数寻优的问题，存在唯一解。容易证明，解中将只有一部分（通常是少部分） \mathbf{a}_i 不为零，对应的样本就是支持向量。解上述问题后得到的最优分类函数是

$$f(\mathbf{x}) = \text{sgn}\{(\mathbf{w} \cdot \mathbf{x}) + b\} = \text{sgn}\left\{\sum_{i=1}^n \mathbf{a}_i^* y_i (\mathbf{x}_i \cdot \mathbf{x}) + b^*\right\}, \quad (4)$$

式中的求和实际上只对支持向量进行。 b^* 是分类阈值，可以用任一个支持向量（满足(1)中的等号）求得，或通过两类中任意一对支持向量取中值求得。

对非线性问题，可以通过非线性变换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。这种变换可能比较复杂，因此这种思路在一般情况下不易实现。但是注意到，在上面的对偶问题中，不论是寻优目标函数(3)还是分类函数(4)都只涉及训练样本之间的内积运算 $(\mathbf{x}_i \cdot \mathbf{x}_j)$ 。设有非线性映射：

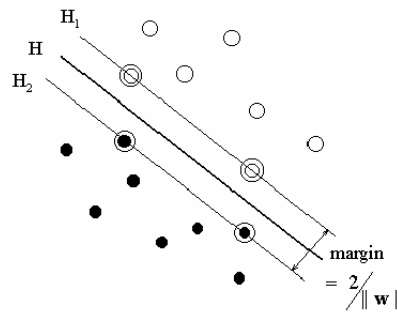


图 1 最优分类面

$\mathbf{R}^d \rightarrow H$ 将输入空间的样本映射到高维

(可能是无穷维)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点积，即 $(\mathbf{x}_i \cdot \mathbf{x}_j)$ ，而没有单独的 (\mathbf{x}_i) 出现。因此，如果能够找到一个函数 K 使得

$K(\mathbf{x}_i, \mathbf{x}_j) = (\mathbf{x}_i \cdot \mathbf{x}_j)$ ，这样，在高维空间实际上只需进行内积运算，而这种内积运算是可以用原空间中的函数实现的，我们甚至没有必要知道变换的形式。根据泛函的有关理论，只要一种核函数 $K(\mathbf{x}_i, \mathbf{x}_j)$ 满足 Mercer 条件，它就对应某一变换空间中的内积。

因此，在最优分类面中采用适当的内积函数 $K(\mathbf{x}_i, \mathbf{x}_j)$ 就可以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，此时目标函数(3)变为：

$$Q(\mathbf{a}) = \sum_{i=1}^n \mathbf{a}_i - \frac{1}{2} \sum_{i,j=1}^n \mathbf{a}_i \mathbf{a}_j y_i y_j K(\mathbf{x}_i, \mathbf{x}_j), \quad (5)$$

而相应的分类函数也变为

$$f(\mathbf{x}) = \text{sgn}\left(\sum_{i=1}^n \mathbf{a}_i^* y_i K(\mathbf{x}_i, \mathbf{x}) + b^*\right), \quad (6)$$

这就是支持向量机。

这一特点提供了解决算法可能导致的“维数灾难”问题的方法：在构造判别函数时，不是对输入空间的样本作非线性变换，然后在特征空间中求解；而是先在输入空间比较向量(例

如求点积或是某种距离), 对结果再作非线性变换[9]。这样, 大的工作量将在输入空间而不是在高维特征空间中完成。SVM 分类函数形式上类似于一个神经网络, 输出是 s 中间节点的线性组合, 每个中间节点对应一个支持向量, 如图 2 所示。

函数 K 称为点积的卷积核函数, 根据[2], 它可以看作在样本之间定义的一种距离。

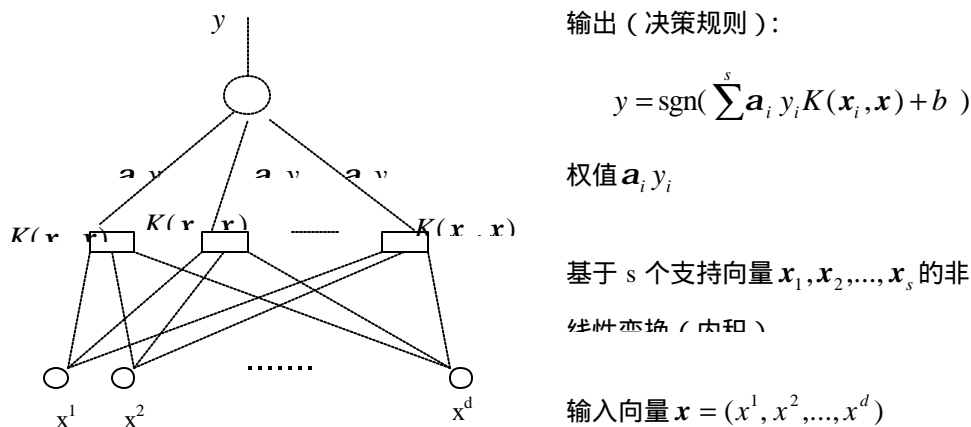


图 2 支持向量机示意图

显然, 上面的方法在保证训练样本全部被正确分类, 即经验风险 R_{emp} 为 0 的前提下, 通过最大化分类间隔来获得最好的推广性能。如果希望在经验风险和推广性能之间求得某种均衡, 可以通过引入正的松弛因子 ξ_i 来允许错分样本的存在。这时, 约束 (1) 变为

$$y_i[(w \cdot x_i) + b] - 1 + \xi_i \geq 0, \quad i = 1, \dots, n \quad (7)$$

而在目标——最小化 $\frac{1}{2} \|w\|^2$ ——中加入惩罚项 $C \sum_{i=1}^n \xi_i$, 这样, Wolfe 对偶问题可以写成:

$$\text{Maximize:} \quad Q(\mathbf{a}) = \sum_{i=1}^n a_i - \frac{1}{2} \sum_{i,j=1}^n a_i a_j y_i y_j K(x_i, x_j) \quad (8)$$

$$\text{s.t.} \quad \sum_{i=1}^n y_i a_i = 0 \quad (9a)$$

$$0 \leq a_i \leq C \quad i = 1, \dots, n \quad (9b)$$

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

定义

$$w(\mathbf{a}) = \sum_i a_i y_i \Phi(x_i) \quad (10)$$

$$F_i = w(\mathbf{a}) \cdot \Phi(x_i) - y_i = \sum_j a_j y_j K(x_i, x_j) - y_i \quad (11)$$

对偶问题的 Lagrange 函数可以写成:

$$L = \frac{1}{2} \mathbf{w}(\mathbf{a}) \cdot \mathbf{w}(\mathbf{a}) - \sum_i \mathbf{a}_i - \sum_i d_i \mathbf{a}_i + \sum_i m_i (\mathbf{a}_i - C) - \mathbf{b} \sum_i \mathbf{a}_i y_i \quad (12)$$

KKT 条件为

$$\frac{\partial L}{\partial \mathbf{a}_i} = (F_i - \mathbf{b}) y_i - d_i + m_i = 0 \quad (13a)$$

$$d_i \mathbf{a}_i = 0 \quad \text{且} \quad d_i \geq 0 \quad (13b)$$

$$m_i (\mathbf{a}_i - C) = 0 \quad \forall i \quad (13c)$$

由此，我们可以推导出如下关系式：

$$\bullet \text{ 若 } \mathbf{a}_i = 0 \quad \text{则 } d_i \geq 0 \quad m_i = 0 \quad \Rightarrow \quad (F_i - \mathbf{b}_i) y_i \geq 0 \quad (14a)$$

$$\bullet \text{ 若 } 0 < \mathbf{a}_i < C \quad \text{则 } d_i = 0 \quad m_i = 0 \quad \Rightarrow \quad (F_i - \mathbf{b}_i) y_i = 0 \quad (14b)$$

$$\bullet \text{ 若 } \mathbf{a}_i = C \quad \text{则 } d_i = 0 \quad m_i \geq 0 \quad \Rightarrow \quad (F_i - \mathbf{b}_i) y_i \leq 0 \quad (14c)$$

由于 KKT 条件是最优解应满足的充要条件[6]，所以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。

3. SVM 算法中目前的研究状况

由于 SVM 方法较好的理论基础和它在一些领域的应用中表现出来的优秀的推广性能，近年来，许多关于 SVM 方法的研究，包括算法本身的改进和算法的实际应用，都陆续提了出来。尽管 SVM 算法的性能在许多实际问题的应用中得到了验证，但是该算法在计算上存在着一些问题，包括训练算法速度慢、算法复杂而难以实现以及检测阶段运算量大等等。

传统的利用标准二次型优化技术解决对偶问题的方法可能是训练算法慢的主要原因：首先，SVM 方法需要计算和存储核函数矩阵，当样本点数目较大时，需要很大的内存，例如，当样本点数目超过 4000 时，存储核函数矩阵需要多达 128 兆内存；其次，SVM 在二次型寻优过程中要进行大量的矩阵运算，多数情况下，寻优算法是占用算法时间的主要部分。

SVM 方法的训练运算速度是限制它的应用的主要方面，近年来人们针对方法本身的特点提出了许多算法来解决对偶寻优问题。大多数算法的一个共同的思想就是循环迭代：将原问题分解成为若干子问题，按照某种迭代策略，通过反复求解子问题，最终使结果收敛到原问题的最优解。根据子问题的划分和迭代策略的不同，又可以大致分为两类。

第一类是所谓的“块算法”(chunking algorithm)，“块算法”基于的是这样一个事实，即去掉 Lagrange 乘子等于零的训练样本不会影响原问题的解。对于给定的训练样本集，如果其中的支持向量是已知的，寻优算法就可以排除非支持向量，只需对支持向量计算权值（即 Lagrange 乘子）即可。实际上支持向量是未知的，因此“块算法”的目标就是通过某种迭代方式逐步排除非支持向量。具体的作法是，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（一般是指违反 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。

当支持向量的数目远远小于训练样本数目时，“块算法”显然能够大大提高运算速度。然而，如果支持向量的数目本身就比较，随着算法迭代次数的增多，工作样本集也会越来越

大，算法依旧会变得十分复杂。因此第二类方法把问题分解成为固定样本数的子问题：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”与工作样本集中的样本进行等量交换，即使支持向量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。

固定工作样本集的方法和块算法的主要区别在于：块算法的目标函数中仅包含当前工作样本集中的样本，而固定工作样本集方法虽然优化变量仅包含工作样本，其目标函数却包含整个训练样本集，即工作样本集之外的样本的 Lagrange 乘子固定为前一次迭代的结果，而不是像块算法中那样设为 0。而且固定工作样本集方法还涉及到一个确定换出样本的问题（因为换出的样本可能是支持向量）。这样，这一类算法的关键就在于找到一种合适的迭代策略使得算法最终能收敛并且较快地收敛到最优结果。

固定工作样本集的方法最早大概是由 Osuna et al.^[6] 提出的。在[4]中，Edgar Osunal 等人介绍了一种具体的算法并对人脸识别问题进行了实验。将样本集分为两个集合 B 和 N ，集合 B 作为子问题工作样本集进行 SVM 训练，集合 N 中所有样本的 Lagrange 乘子均置为零。显然，如果把集合 B 中对应 Lagrange 乘子为零的样本 i （即 $a_i = 0, i \in B$ ）与集合 N 中的样本 j （即 $a_j = 0, j \in N$ ）交换，不会改变子问题与原问题的可行性（即仍旧满足约束条件）；而且，当且仅当样本满足条件 $(F_i - b_i) y_i \geq 0$ (14a) 时，替换后的子问题的最优解不变。于是可以按照以下步骤迭代求解：1. 选择集合 B ，构造子问题；2. 求子问题最优解 $a_i, i \in B$ 及 b ，并置 $a_j = 0, j \in N$ ；3. 计算 $F_j, j \in N$ 找出其中不满足条件 $(F_i - b_i) y_i \geq 0$ (14a) 的样本 j ，与 B 中满足 $a_i = 0$ 的样本 i 交换，构成新的子问题。[4]证明了这种迭代算法的收敛性，并给出了两阶多项式分类器在人脸识别问题中的应用结果。

需要说明的是，文中没有说明集合 B 的大小是否改变。作者期望的是支持向量的数目非常少，当然可以固定 B 的大小，作者的意图正是如此。不过为此需要选择一个较大的 B 集合，这样看来，其效率可能还不如块算法。而且如果集合 B 不足以包括所有的支持向量，该算法没有提出改变 B 的大小的策略，有可能得不到结果。

前面提到，固定工作样本集方法的关键在于选择一种合适的换入换出策略。Joachims 指出如果采用某种启发式的迭代策略将会提高算法的收敛速度。最近 John C. Platt 在[5]中提出 SMO (Sequential Minimal Optimization 或 SMO) 算法。将工作样本集的规模减到最小——两个样本。之所以需要两个样本是因为等式线性约束的存在使得同时至少有两个 Lagrange 乘子发生变化。由于只有两个变量，而且应用等式约束可以将其中一个用另一个表示出来，所以迭代过程中每一步的子问题的最优解可以直接用解析的方法求出来。这样，算法避开了复杂的数值求解优化问题的过程；此外，Platt 还设计了一个两层嵌套循环分别选择进入工作样本集的样本，这种启发式策略大大加快了算法的收敛速度。标准样本集的实验结果证明，SMO 表现出在速度方面的良好性能。

子问题的规模和迭代的次数是一对矛盾，SMO 将工作样本集的规模减少到 2，一个直接的后果就是迭代次数的增加。所以 SMO 实际上是将求解子问题的耗费转嫁到迭代上，然后在迭代上寻求快速算法。但是，SMO 迭代策略的思想是可以用到其他迭代算法中的，可见，SMO 还有改进的余地。

SMO 在实际应用中取得了较好的效果,但它也存在着一些问题。SMO 算法每次迭代都要更新 b 值,但是该值有可能是无法确定的(例如不存在 $0 < a_i < C$ 的样本,尽管这种情况很少出现),这时 SMO 采用的方法是确定出 b 的上下界然后取平均值;另外,每一次迭代过程中的 b 值仅取决于上次迭代结果的两个变量的最优值,用这个 b 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,从而影响了该算法的效率^[6]。

解决算法速度问题的另一个途径是采用序列优化的思想。这种方法主要目的是研究当出现新的单个样本时,它与原有样本集或其子集,或是原有样本集训练结果的关系,例如,它的加入对原有样本集的支持向量集有什么样的影响,怎样迅速地确定它对新的分类器函数的贡献等等。[10]中提出了一种用卡尔曼滤波器求解的方法。

4. 研究方向

应该说,块算法和固定工作样本集算法是各有优缺点的。毫无疑问,固定工作样本集的算法解决了占用内存的问题,而且限制了子问题规模的无限增大;但是,从这个意义上来说,固定工作样本集的算法把解标准二次型的寻优问题的时间转嫁到循环迭代上了,它的迭代次数一般会比“块算法”多。尤其是 SMO,如果没有一个好的启发式迭代策略,该算法就是一种盲目爬山法。

基于此,我们提出一种算法思想,希望能够综合两类算法的特点。我们仍旧从最终目标中抽取子问题,借用某种迭代策略使算法收敛,关键的,我们希望一方面子问题规模不会太小,以免迭代次数太多,另一方面能借鉴 SMO 的思想,利用二次问题的特点,找到子问题的解析解法,或者是近似解,从而不必对每一个子问题都调用寻优算法。

此外,由于 SVM 方法的性能与实现上的巨大差异,我们在求解子问题时不一定要得到精确解(解的精确度可以由迭代来保证),甚至还可以考虑对最终目标求取近似解。这样,尽管结果的性能会受到影响,但是如果能够大幅度提高运算速度,它仍不失为一种好方法。

一种在二维数据实验中取得一定效果的方法是近邻 SVM。由于 SVM 的目标是在高维特征空间中最大化分类间隔,亦即最小化 $\|w\|$,而目标函数的度量就是欧氏距离,所以两类样本点之间的欧氏距离应该与目标函数有着密切的关系,可以认为,两类样本中距离最近的点最有可能成为支持向量,相反地,与异类样本距离较远则意味着它与分类面关系不大。事实上,传统的近邻法就是以距离作为判定样本点类别的依据,只不过在这里由于 SVM 的特点,距离可以直接用欧氏距离定义。

在高维特征空间中,两个样本 x_i, x_j 之间的欧氏距离的平方为:

$$\begin{aligned}\|\Phi(x_i) - \Phi(x_j)\|^2 &= (\Phi(x_i) - \Phi(x_j)) \cdot (\Phi(x_i) - \Phi(x_j)) \\ &= \Phi(x_i) \cdot \Phi(x_i) + \Phi(x_j) \cdot \Phi(x_j) - 2\Phi(x_i) \cdot \Phi(x_j) \\ &= K(x_i, x_i) + K(x_j, x_j) - 2K(x_i, x_j)\end{aligned}$$

我们仍旧可以利用核函数,而不需要知道具体的变换形式。有了距离的定义后,我们可以对每一样本找出与它距离最近的几个异类样本。遍历所有样本后,我们可以得到这些最近邻的并集,通常,如果近邻个数选择适当,这个并集能够包含且只包含大多数处于最优分类面附近的样本,用这个并集作为训练样本集,可以大大提高算法的速度,得到的结果也比较令

人满意。当然，近邻 SVM 还存在着许多问题，例如对某些特殊分布的样本集可能效果很差，再比如计算近邻样本也是一个很耗时的工作，这些都需要进一步的研究改进。

采用序列优化的思想也可以解决算法速度问题。如果能够简单有效地确定单个样本加入工作样本集后对训练结果的影响，一方面，出现新的样本时，可以利用原来的训练结果而不必重新开始；另一方面，让训练样本逐个进入工作样本集也可以简化寻优过程，提高算法速度。这实际上是将工作样本集中的样本数减少到一个，[11]中提出的 SOR 方法就是这样一种思路。

核函数是 SVM 方法中少数几个能够调整的参数之一，目前的方法一般都是使用多项式、径向基函数等等。尽管一些实验结果表明核函数的具体形式对分类效果的影响不大，但是核函数的形式以及其参数的确定决定了分类器类型和复杂程度，它显然应该作为控制分类器的性能的手段。有关核函数选择的理论依据仍旧很少，[12]中提到一种在 SVM 算法过程中自适应地选择模型参数的方法。我们的想法是，找出样本集分布特点与最优分类器之间的可能的对应关系，然后根据待训练样本的一些先验知识选择分类器的类型和参数，或者直接构造新的类型，可以预先确定，也可以在训练过程中逐步优化。

另外，SVM 方法在分类方面的应用比较多，其实它在其它方面也有其优势，例如数据挖掘、特征选择和提取等，它的核函数的思想也已经应用到。尤其是特征选择，SVM 方法用少数支持向量代表整个样本集的思想与特征选择极为类似。SVM 方法的最优分类面是以分类间隔来衡量的，如果用不同的样本作支持向量，应该得到不同的分类间隔。因此，如果把样本看作特征，建立某种准则函数，它以特征为变量，同时与分类间隔相对应，那么训练过程就完成了特征选择。在[13]中对此提出了一些方法，我们可以做进一步的研究。

References:

- [1] V.Vapnik. *Nature of Statistical Learning Theory*. John Wiley and Sons, Inc., New York, in preparation.
- [2] J.C.Burges. *A Tutorial on Support Vector Machines for Pattern Recognition*. Bell Laboratories, Lucent Technologies. 1997.
- [3] Filip Mulier. *Vapnik-Chervonenkis (VC) Learning Theory and Its Applications*. IEEE Trans. on Neural Networks. Vol.10, No.5, Sep 1999.
- [4] Edgar Osuna et al. *Training Support Vector Machines: an Application to Face Detection*.
- [5] John C. Platt. *Using Analytic QP and Sparseness to Speed Training of Support Vector Machines*.
- [6] S.S. Keerthi et al. *Improvements to Platt's SMO Algorithm for SVM Classifier Design*.
- [7] Chris J.C. Burges. *Simplified Support Decision Rules*.
- [8] Chris J.C. Burges, Beruhard Scholkopf. *Improving the Accuracy and Speed of Support Vector Machines*. Neural Information Processing Systems, Vol.9, M. Mozer, M. Jordan, & T. Petsche, eds. MIT Press, Cambridge, MA, 1997.
- [9] Corinna Cortes, V.Vapnik. *Support-Vector Network*. Machine Learning, 20.273 - 297 (1995)

- [10]Nando de Freitas et al. *Sequential Support Vector Machines*. Neural Networks for Signal Processing IX. 31-40(1999)
- [11]Olvi L. Mangasarian, David R. Musicant. *Successive Overrelaxation for Support Vector Machines*. IEEE Trans. on Neural Networks. Vol.10, No.5, Sep 1999.
- [12]Nello Cristianini, Bristol et al. *Dynamically Adapting Kernels in Support Vector Machines*.
- [13]P.S. Bradley, O.L. Mangasarian. *Feature Selection via Concave Minimization and Support Vector Machines*.
- [14]J.Platt. *Sequential minimal optimization: A fast algorithm for training support vector machines*. Advances in Kernel Methods-Support Vector learning. Cambridge, MA: MIT Press,1999, pp.185-208.
- [15]S.S.Keerthi et al. *A Fast iterative Nearest Point Algorithm for Support Vector Machine Classifier Design*.
- [16]边肇祺等。模式识别。清华大学出版社。1988