支持向量机处理大规模问题算法综述

文益民[1,2] 王耀南[1] 吕宝粮[3] 陈义明[4]

(湖南大学电气与信息工程学院 长沙 410082)[1] (湖南工业职业技术学院 长沙 410208)[2]

(上海交通大学计算机科学与工程系 上海 200030)[3] (湖南农业大学信息科学技术学院 长沙 410073)[4]

摘 要 支持向量机在处理大规模问题时存在训练时间过长和内存空间需求过大的问题。分析了支持向量机在处理大规模问题时存在的局限性;对利用支持向量机处理大规模问题的各种算法进行了分类,并对每种算法的研究状况进行了较全面而深入的综述;对该领域内值得进一步研究的问题进行了讨论。

关键词 支持向量机,大规模问题,机器学习

中图法分类号 TP391 **文献标识码** A

Survey of Applying Support Vector Machines to Handle Large-scale Problems

WEN Yi-min[1,2] WANG Yao-nan[1] LU Bao-liang[3] CHEN Yi-ming[4]

(College of Electrical and Information Engineering, Hunan University, Changsha 410082, China)[1]

(Hunan Industry Polytechnic, Changsha 410208, China)[2]

(Department of Computer Science and Engineering, Shanghai Jiaotong University, Shanghai 200030, China)[3]

(School of Information Science and Technology, Hunan Agricultural University, Changsha 410073, China)[4]

Abstract Being applied to handling large-scale problems, support vector machines(SVMs) needs longer training time and larger memory. The paper analyzed the limitation of SVMs, classified the algorithms of applying SVMs to handle large-scale problems into seven types, and made profound and comprehensive analysis of each kind of algorithm. Moreover, some issues valuable for future exploration in this area were indicated and discussed.

Keywords Support vector machines, Large-scale problem, Machine learning

1 引言

随着人类社会的发展以及科学技术在一些关键领域如生物技术等领域的突破,大量的数据产生了且更新速度很快。比如:仅登录在美国 GenBank 数据库中的 DNA 序列总量已超过 70 亿碱基对,未来 DNA 序列数据的增长将更为惊人。根据中国互联网络中心多年的年度调查报告:2002 年中国国内的网页量为 1.57 亿个网页,2006 年 12 月 31 日止全国网页数达到 44.7 亿个。日本的读卖新闻在 1987－2001 年间就收集了 2,190,512 条记录[1],并以很高的速度继续增加。在空间信息挖掘领域,空间数据的数量也在快速增长。通过高分辨率、高动态的新型卫星传感器获取的数据容量均在千兆量级以上[2,3]。

尽管数据是信息和知识的源泉,但数据并不等于信息和知识,关键在于人类如何从中挖掘它们,理解它们。在生命科学技术领域,理解大量生物学数据所包含的生物学意义已成为后基因组时代极其重要的课题,生物学数据的海量积累和分析处理将导致重大生物学规律的发现。然而,与正在以指数方式增长的生物学数据相比,人类相关知识的增长却十分缓慢。一边是海量数据;另一边是对生命技术、农业和环保等方面新知识的渴求,这些新知识将帮助人们改善生存环境和提高生活质量,这就构成了一个极大的矛盾。能否有效处理这些海量数据成为了科学技术发展和人类生活质量进一步提高的瓶颈。

2 支持向量机处理大规模问题的局限性

自 V. N. Vapnik 于 1979 年开始研究支持向量机[4,5]以来,支持向量机逐渐得到了机器学习领域专家的认同。由于其在机器学习中显示的优点,人们希望使用支持向量机方法处理大规模问题。但是,在处理大规模问题时支持向量机还存在以下局限性:1)由于支持向量机的训练过程实质是求解一个二次规划问题,其求解时间复杂度为 $O(N^3)$。由于要存储核矩阵,空间复杂度为 $O(N^2)$。当训练集规模巨大时,支持向量机的训练时间会太长,同时核矩阵的规模太大将导致

到稿日期:2008-12-05 返修日期:2009-03-20 本文受国家 863 项目(2007AA04Z244),国家自然科学基金重点项目(60835004),湖南省博士后科研资助专项计划项目(2008RS4005)资助。

文益民(1969－),男,副教授,博士后,CCF 高级会员,主要研究方向为机器学习等,E-mail:ymwen2004@yahoo.com.cn;王耀南(1957－),男,博士,教授,主要研究方向为智能控制理论与应用、智能信息处理与融合、模式识别与图像处理、智能机器人系统、电气控制工程、复杂工业综合自动化系统;吕宝粮(1960－),男,博士,教授,主要研究方向为仿脑计算理论与模型、神经网络、并列机器学习等;陈义明(1969－),男,博士生,主要研究方向为数据挖掘。

内存空间不足;2)支持向量机的训练结果是用支持向量表示的,当支持向量数目太大将导致超出内存限制,使得分类器不能全部装入内存,影响分类器的使用;3)由于计算机系统的不可靠性,集中表示的分类器将面临失效的严重风险;4)二次规划问题的求解过程本质是面向批量数据,已经训练好的支持向量机无法将新增加的训练样本纳入。因此研制使用支持向量机处理大规模问题的方法势在必行[6]。

3 支持向量机处理大规模问题算法综述

使用支持向量机处理大规模问题的方法可分为7种:工作集方法、并行方法、避免求解二次规划问题、几何方法、减少训练样本、训练集分解法、增量学习法。

3.1 工作集方法

工作集方法就是每次只针对一部分拉格朗日乘子进行优化而将其他拉格朗日乘子视为常量。被优化的拉格朗日乘子的集合叫工作集,工作集的规模通常较小,主要由可用的内存多少决定。于是每一步只需要求解一个规模大大小于原问题规模的二次规划问题。在下一次迭代中采用一些启发式规则再挑选下一个工作集,这样经过多次迭代最终求得原问题的最优解。E. Osuna 的工作[7]奠定了工作集方法的理论基础。工作集方法的特点是,要在算法的快速收敛与每个迭代步的二次规划子问题求解的计算代价之间取得平衡。工作集方法的实现主要有 SMO(Sequential Minimal Optimization)[8],SVMlight[9]和 libSVM[10]等,其中 SVMlight 和 libSVM 已经成为了目前训练支持向量机的最常用的方法。这些快速有效的训练方法客观上促进了支持向量机方法的应用,这些算法的时间复杂度一般为 $O(N^2)$。

J. Platt 提出了 SMO 算法[8]。该算法将工作集的规模减小到最小——2,也就是每次只优化两个拉格朗日乘子,同时固定其他拉格朗日乘子。由于两个变量的最优化问题可以解析求解,在算法中不需要使用数值计算方法求解二次规划问题,因此内循环只需很少的计算。而且,该算法不需要存储核矩阵,没有矩阵运算。因此该算法通常表现出整体的快速收敛性质。仿真实验表明:该算法的时间复杂度为 $O(N^2)$,成功地降低了支持向量机训练的时间复杂度。SMO 算法提出后,针对工作集中拉格朗日乘子的选择、工作集大小、缓存使用等多种因素,很多学者做了很多工作。比如:L. L. Dai 等提出[11]在内循环中每次优化 3 个拉格朗日乘子,因为 3 个变量的优化问题同样可以解析求解。仿真实验表明:该算法比 SMO 的训练时间更短,但同时能确保支持向量机的一般化能力。业宁等则提出了在内循环中每次优化 3 个拉格朗日乘子的 MLSVM4 算法[12]。实验结果表明该算法的训练速度超过 SMO 算法 3 到 42 倍。李建民等[13]通过对 SMO 可行方向的解释,提出了一种收益代价平衡的工作集选择方法,综合考虑与工作集相关的目标函数的下降量和计算代价,以提高缓存的效率。实验结果表明,该算法可以提高 SMO 算法的性能,缩短支持向量机的训练时间。

T. Joachims 提出了 SVMlight 算法[9]。该算法所取工作集规模不小于 2,采取 Zoutendijk 方法,通过利用一阶信息选择可行方向,通过求解一个简单的优化问题得到工作集。同时该算法还采用收缩(Shrinking)方法,也就是通过估计拉格朗日乘子中的非支持向量和界上支持向量,以减小核矩阵的规模,达到加速训练过程的目的。仿真实验表明 SVMlight 算法比 SMO 算法更快。C. C. Chang 和 C. J. Lin 提出的 libSVM 算法取工作集大小为 2,通过采用二阶信息来选择可行方向,实现支持向量机的训练过程提速。

3.2 并行化方法

工作集方法不容易在算法的快速收敛与每个迭代步的二次规划子问题求解的计算代价之间取得平衡。比如:libSVM 取工作集大小为 2,使工作集方法中的二次规划子问题的求解代价达到最小,但由于算法每次只对两个变量求解,算法的收敛速度无疑降低;SVMlight 取规模不小于 2 的工作集,利用数值方法求解二次规划子问题,增大了计算代价,但由于每次对多个变量求解,提高了算法的收敛速度。鉴于工作集方法的快速收敛与每个迭代步的二次规划子问题求解的计算代价之间难以平衡,2003 年 G. Zaghirati 和 L. Zanni 提出将 SVMlight 算法中的二次规划子问题的求解过程使用并行计算手段并行化[14,15]。该算法提高了 SVMlight 中的二次规划子问题的求解速度,使得可处理的工作集规模从 $O(10)$ 增加到 $O(10^2)$ 或 $O(10^3)$,从而实现算法加速。经过进一步改进,他们又提出了并行的梯度投影分解技术(parallel gradient projection-based decomposition technique, PGPDT)[16],使得可处理的工作集规模达到 $O(10^6)$。仿真实验表明:PGPDT 取得了比 libSVM 和 SVMlight 更快的训练速度。

3.3 避免求解二次规划问题

通过修改目标函数,避免求解二次规划问题。O. L. Mangasarian 提出了拉格朗日支持向量机(Lagrangian support vector machines, LSVM)[17]。他通过改变二次规划问题的目标函数的形式,使得二次规划问题变成一个无约束二次规划问题,从而实现迭代方法求解。在迭代方法求解过程中只需要计算一个 $(n+1)\times(n+1)$(其中 n 为训练样本的维数)矩阵的逆,因此 LSVM 算法既避免了求解二次规划问题,又降低了矩阵运算的时间复杂度。但是,LSVM 只适应核函数是线性、样本数量巨大但维数较低的情形。仿真实验表明:该算法在确保一般化能力的前提下,加速了训练过程。O. L. Mangasarian 等还提出了 LSVM 的两个变种——牛顿拉格朗日支持向量机(Newton Lagrangian SVM, NSVM)[18]和主动支持向量机(Active support vector machine, ASVM)[19]。这些算法在数据维数很高时(等于样本规模)也能加速训练过程。C. Yang 等将 LSVM 扩展到非线性核情形。该算法通高斯变换使得支持向量机的求解时间复杂度降至 $O(N)$。杨绪兵等通过对基于广义特征值的最接近支持向量机[20]的改进,提出了基于原型超平面的多类最接近支持向量机,提高了最接近支持向量机的训练速度,确保了一般化能力[21]。

3.4 几何方法

利用支持向量机的几何本质,使用几何方法求解分类超平面或将核方法转化成几何问题。第一种方法根据支持向量内在的几何意义使用几何方法求分类超平面。S. S. Keerthi 等提出将求线性可分的两类分类问题的支持向量机求解问题转化成为求解两个点集的凸壳的最近两点的问题[22],并提出了求解两个点集的凸壳的最近两点的问题(Nearest Point Problem, NPP)的快速迭代方法。第二种将核方法直接转化成几何问题。基于支持向量机求解过程的近似性,I. W. Tsing 等提出将核方法转换成最小覆盖球问题(minimum enclosing

ball, MEB) 从而提出了核向量机(core vector machine, CVM)[23]。作者证明了两类 L2-SVM[22] 与 MEB 问题等价。核向量机算法首先使用某种启发式规则选择一个样本作为核向量,同时计算初始 MEB 的半径和中心,进而找到离 MEB 中心最远的样本,将其加入到核向量集合中,然后求新的 MEB 的中心和半径。依此循环直到没有样本可加入,从而求得整个训练集的 MEB。最后根据支持向量机与 MEB 问题的等价关系,求得分类超平面。由于求解 MEB 的算法的时间复杂度与样本数量呈线性关系而其空间复杂度与样本数量独立,核向量机的时间复杂度与样本数量呈线性关系而其空间复杂度与样本数量无关。仿真实验表明,核向量机与标准支持向量机具有相同的一般化能力,但测试时间更短。

3.5 减少训练样本方法

采取某种策略,通过挑选最可能为支持向量的训练样本或筛减最不可能为支持向量的训练样本或以上两种方法同时采用对训练集实施预处理,以实现训练集规模的减小,实现训练过程的加速。H. Shin 等提出了一种基于快速样本选择的支持向量机方法[24],通过对训练样本的邻域信息的分析,挑选位于分类超平面附近的训练样本作为最终的训练集,从而实现训练过程的加速,同时确保分类器的一般化能力。

文献[25]提出了一种 RSVM(reduced support vector machines),利用文献[26]中提出的一般支持向量机 GSVM(generalized support vector machine)方法,随机地从全部训练样本中选择 1%的样本构造长方形核 $K(A,\bar{A})$,其中 A 代表全部训练样本构成的矩阵,$\bar{A}$ 表示从训练样本中随机选择的 1%的样本构成的矩阵。然后按照 GSVM 方法求解优化问题得到分类面。这种方法降低了训练时间复杂度,且分类面只与 $\bar{A}$ 相关,相比于 GSVM 提高了测试速度。实验结果表明 RSVM 的一般化能力要比标准支持向量机要高。K. M. Lin 和 C. J. Lin 进一步研究了 RSVM[27],他们发现:如果支持向量在训练集中的比例较大,则 RSVM 的一般化能力将比标准支持向量机略低。

使用聚类方法对大量数据进行预处理,抽取聚类子集的有效信息,从而实现训练样本的筛减。在文献[28]中,首先将训练数据按照类别分别实施聚类,靠近相反类的样本聚成包含样本比较少的子集,因为这些样本体现更多的分类信息。而远离相反类的样本聚成比较大的子集。通过这种方法,将各个子集的中心的集合作为训练集,并且将各个子集中的样本数目作为该中心的权值,然后求解一个松弛变量带权值的二次凸优化方程。仿真实验表明:利用该方法求得的支持向量机和用整个训练集训练的支持向量机有几乎一样的一般化能力。通过聚类大大减少了训练样本,从而大大提高了训练速度。

D. Boley 等提出了筛减训练样本的自适应聚类方法 ClusterSVM[29]。他首先将训练集按照类别分别实施聚类,用每个子集的中心作为该子集的"代表",然后用这些"代表"的集合训练初始支持向量机,从而初步发现训练集中可能的非支持向量。将只包含非支持向量的子集用该子集的中心代表,从而起到筛减非支持向量的目的。最后将剩余样本合并训练得到最终的分类器。仿真实验表明:ClusterSVM 的时间复杂度是支持向量数量的平方。该算法能加速支持向量机的训练过程,同时确保支持向量机的一般化能力。

H. Yu 等学者提出了扩展支持向量机的分层聚类算法(Clustering-Based SVM, CB-SVM)[30]。该算法采用分层微聚类算法 BIRCH[31]来扩展支持向量机。该算法首先将正反样本分别根据 BIRCH 算法构造两个 CF 树(clustering feature tree)。然后使用根节点的条目数据训练一个支持向量机,将邻近分类超平面的条目进行分解,将新增加的子条目数据加入原训练集后再训练一个支持向量机,依此循环直到没有任何子条目数据可加入。该算法的时间复杂度为支持向量数目的平方量级。实验表明该算法能得到更好的一般化能力。

李红莲等提出了训练样本的修剪策略[32]。首先从大规模样本集中抽取一个小规模样本集 S,然后在 S 上训练初始支持向量机。然后将大规模样本集中离初始支持向量机分类超平面较近的训练样本保留,而将其他样本修剪,用剩余样本训练得到最终的支持向量机。该算法提高了支持向量机的训练速度,同时确保了支持向量机的一般化能力不会降低。类似地,罗瑜等提出了在训练前先求出类别质心,去除非支持向全对应的样本,从而达到缩小样本集的方法[33]。郑志洵等提出了寻找两类别交界处的样本的方法[34]。

基于训练集样本修剪技术,G. H. Bakir 等打破支持向量数目增长与训练集规模增长之间的线性关系,提出了交叉训练样本修剪算法[35],从而得到可分且基本保持原有分类面的训练集。该算法不但能够加速训练过程,同时能够加速测试过程,但该算法得到的支持向量机的一般化能力略有降低。

3.6 训练集分解方法

训练集分解方法就是基于某种策略将训练集分解成若干子集,在每个子集上训练支持向量机,最后采用某种策略将各支持向量机组合。该方法的实现有串行和并行两种方式。并行的训练集分解方法有:混合支持向量机专家[36]、贝叶斯支持向量机(BC-SVM)[37]、最小最大模块化支持向量机 M3-SVM[38]、并行混合支持向量机专家[39]、快速模块化支持向量机(Fast modular network implementation for support vector machines)[40]、分布式支持向量机(Distributed support vector machines)[41]和作者提出来的基于可信多数投票的并行模块化支持向量机等。

V. Tresp 提出将支持向量机嵌入贝叶斯集成(BCM)学习算法,以实现对大规模数据的处理。贝叶斯集成(BCM)学习的概念最早出现在 2000 年的文献[42]中。在文献[43]中作者又提出了一般化贝叶斯集成学习的概念(GBCM)。贝叶斯集成学习方法是根据贝叶斯规则将整个训练集划分成若干子集,在各个子集上进行贝叶斯学习,然后再将各个子系统的结果按照式(1)进行加权组合:

$$\hat{E}(f^q|D)=C^{-1}\sum_{i=1}^{K}\mathrm{cov}(f^q|D^i)^{-1}E(f^q|D^i) \tag{1}$$

其中:D^i 表示一个数据子集,$E(f^q|D^i)$ 是第 i 个子系统的输出结果。$\hat{E}(f^q|D)$ 是整个集成网络的输出,$\mathrm{cov}(f^q|D^i)$ 是第 i 个子系统输出范围,表示第 i 个系统对它输出的一种确信程度。贝叶斯集成学习中对自身输出不太确信的分类器,将自动减弱它对最终组合输出结果的影响。在文献[44]中作者将贝叶斯集成学习方法扩展成贝叶斯集成支持向量机(BC-SVM)。BC-SVM 能将训练的时间复杂度降至 $O(N)$。但实验结果表明 BC-SVM 的一般化能力不比单个支持向量机要好,可算作相当。在文献[45,46] 中,V. Tresp 又将贝叶斯集

成学习扩展到其他核方法。

在文献[36]中，J. T. Kwok 提出了一种能将单个支持向量机和支持向量机专家网络统一的框架。将支持向量机专家网络的训练过程转化为一个与求解单个支持向量机的二次优化问题非常相近的问题，通过解此二次凸优化问题，求得支持向量机专家网的解。这种方法从数学上有严格的推导，但文献中没有仿真实例，且并没有从理论上证明这么做就能提高一般化能力，并且该方法中的二次凸优化问题是建立在全部训练集上的，因此该方法并不能降低训练时间复杂度。

在文献[39]中，R. Collobert 等在文献[36]的基础上，遵照相同的指导思想对算法进行了重新设计。该方法采用专家网络的体系结构，首先将整个训练集随机划分成若干子集，在每个子集上单独训练支持向量机。各个子网络输出的组合权值由门网生成，门网的训练使用最小均方差法，通过门网输出的权值将尽量减少专家网的经验误差。由于第一次对整个训练集的随机划分不见得最好，在训练过程中动态地调整各个子集的组成，然后重新训练各个专家，再训练门网。这样最后得到最优解(局部的)。由于各个专家相互独立，该方法与文献[36]相比能降低训练时间复杂度。实验结果表明该方法不但提高了一般化能力，而且大大缩短了训练时间(如果是并行训练，加速比会更大)，但由于门网训练的限制，该方法的时间复杂度在 $O(N)$ 与 $O(N^2)$ 之间。

在文献[47]中，A. Choudhury 等处理了大规模高斯过程模型。该方法根据局部学习的思想(local learning)，首先采用 GeoClust 方法对训练集进行聚类分割，将整个训练集聚合成样本数量大致相等的若干子集。高斯过程的方差函数根据训练集的划分而尽量实现各训练样本之间的解相关。实验结果表明这种方法比 BC-SVM[37] 要好。

G. B. Huang 等提出了一种快速模块化支持向量机算法[40]。该算法首先采用随机的平行超平面簇将训练集分裂分割成若干规模相近的子集，然后在每个子集上训练支持向量机。测试样本落在哪个子集中，就用该子集上的支持向量机对其分类。该算法不管是串行实现还是并行实现都极大地加速了训练过程。仿真实验表明：该算法得到的快速模块化支持向量机的一般化能力在高维情形下略低于不对训练集进行分解的标准支持向量机，而在低维时一般化能力要好一些。该算法在处理 Banana 数据时就得到了超过标准支持向量机的一般化能力。但是，该方法对训练集划分方法的依赖性较大。

A. Vazquez-Navia 等提出了分布式支持向量机算法[41]。该算法首先对训练集实施分割，然后在每个子集上训练支持向量机，各子集之间按照某种方式共享有关信息后重新训练分类器，依此循环直到训练过程收敛，则其中任何一个子集上得到支持向量机均可作为最后的学习结果。各子集间共享信息的方式有两种：一是在各个子集上训练支持向量机，然后将支持向量在各子集间共享。此时的算法叫分布式块算法(distributed chucking)；二是在各个子集上训练增长支持向量机[48]，然后将增长支持向量机的中心在各子集间共享，此时的算法叫分布式半参数化支持向量机。与很多基于训练集分解的支持向量机扩展算法不同，该算法能够得到与基于全部训练样本的标准支持向量机相同的一般化能力，所付出的代价就是通信代价。分布式半参数化支持向量机能保持分类器的大小，从而能确保通信量的不变。与分布式块算法相比，分布式半参数化支持向量机具有与之相当的一般化能力、较小的分类器规模、更好的数据私密性和更少的训练时间。

吕宝粮等提出了最小最大模块化支持向量机(Min-Max-Modular support vector machines，M3-SVM)[38]。该算法基于涌现原理，采用最小最大模块化方法(Min-Max-Modular，M3)[49]扩展支持向量机。在大型文本分类数据上的仿真实验表明：该方法不但能够加速训练过程，还能够得到比标准支持向量机更好的一般化能力。

文益民等提出的基于可信多数投票的并行模块化支持向量机算法首先将训练集随机划分成若干子集，并行在每个子集上训练支持向量机模块。测试时采取带权多数投票框架，使用模块分类器的分类置信度作为该模块分类器的对应权值，然后将各模块支持向量机的分类结果进行组合。分类置信度的估计采取了两种方法，一种是 Woods 提出来的局部分类准确率[50]，另一种采用了后验概率[51]。仿真实验表明该算法在提高支持向量机训练速度的同时，基本确保了泛化能力不下降，而且在训练集划分度增加的情形下，组合支持向量机的泛化能力也不会下降。

H. C. Kim 提出使用 Bagging 算法来扩展支持向量机[52]。该算法按照小 bag 训练集分解策略[53]将训练集分解成若干子集，然后在每个子集上训练基支持向量机，采用多数投票、带权投票和亚学习策略[54] 3 种策略组合各个基支持向量机。仿真实验表明：使用这些组合策略的 bagging 方法得到的分类器都比基于整个训练集的标准支持向量机的一般化能力要好。

基于支持向量的特点——将训练集中排除非支持向量后得到的训练集重新训练得到的支持向量机与原支持向量机相同。多名学者提出了利用某些策略通过训练集分解并行筛选非支持向量后将剩余训练样本训练得到最终分类器的算法。J. X. Dong 等提出[55]将训练集进行分裂分割后得到各个子集上的支持向量，最后将属于各类的支持向量合并得到最后的训练集从而得到最终的支持向量机。他提出的算法的时间复杂度仅与样本类别数量和样本数量呈线性关系。H. P. Graf 等提出了一种循环分层并行算法[56]。该算法首先将训练集分裂分割成 2^l 个子集(l 为整数)，并行地在第一层的各个子集上训练支持向量机以寻找支持向量，在下一层计算中将相邻的两个子集的支持向量合并成一个训练集，然后并行筛选掉其中的非支持向量，再将相邻两个子集的支持向量合并成一个训练集。如此直到训练集只剩一个，通过筛选掉其中的非支持向量后，再将支持向量与第一层的训练集中剩余的非支持向量合并开始下一轮循环。在该算法实现中，各节点之间的通讯量很小，且该算法能收敛于全局最优解。仿真实验表明：该算法在使用 16 个处理器的情形下能加速 5～10 倍。文益民等提出将训练集按照类别进行分解后，采用最小最大模块化(Min-Max-Modular，M3)[49]方法中训练集的组合方法，然后分层筛选掉其中非支持向量，最后得到了一个与不对训练集进行分解训练的支持向量机几乎一致的支持向量机[57,58]，该算法与 H. P. Graf[54] 的思想非常接近，不过该算法没有一个循环过程。

基于训练集分解串行方法主要是使用 Boosting 算法[59,60]来扩展支持向量机。D. Pavlov 等提出了 Boost-

SMO[61]。该算法与 Adaboosting 方法[59,60]基本相同，都是用上次训练过程产生的分类器在整个训练集上的测试准确率决定下次样本的抽取，这将使被前面产生分类器错分的样本更可能出现在下次的训练集中。所不同的是：第一，基分类器为支持向量机；第二，每次抽取的样本规模仅为整个训练集的2%～4%。因此 Boost-SMO 能加速训练过程至 3～10 倍。实验结果表明 Boost-SMO 与基于整个训练集的标准支持向量机相比提高了一般化能力。

3.7 增量学习方法

支持向量机的增量学习研究始于 N. A. Syed 等[62]的研究工作。C. Domeniconi 等在 N. A. Syed 提出算法的基础上提出了另外 4 种不同的增量学习技术：固定分割、错误驱动、过间隔、错误驱动＋过间隔[63]。G. Cauwenberghs 等[64]为支持向量机的在线增量学习作了开拓性的研究工作：当采集到一个新样本时，修改原有支持向量的系数，以保持 Kuhn-Tucker 条件成立。C. P. Diehl[65]将其扩展到一次增量学习多个样本。G. Cauwenberghs 提出的方法能够得到与重新训练一致的支持向量机，该算法还能够进行减量学习，但该算法仍需要多次扫描以前学习过的所有样本。L. Ralaivola 等在局部学习理论[66]的启发下提出了支持向量机的局部增量学习算法[67,68]。G. Fung 等[69]在近似支持向量机[70]的基础上提出了一种可以实现支持向量机的在线增量(减量)学习和块增量(减量)学习的算法，该算法只需一次扫描原来学习过的样本，其在特征空间的维数较低(＜100)时非常有效。A. Tveit 等[71]在 G. Fung 的基础上提出了一种能遗忘原来学习的知识的算法。Y. G. Liu[72]把增量学习的过程看作是一个极限过程，而把增量学习过程定义成为一个二次优化问题。文益民等基于分类器组合原理提出了支持向量机的块增量学习算法，该算法在学习新知识的同时较好地保留了原来学习到的知识[73]。基于支持向量集包含了训练集中的分类信息的原理来实现增量学习，国内学者做了很多研究工作，这些工作的共同特点是通过反复迭代尽量找出包含全部支持向量的最小训练集[74-79]。

4 进一步的问题

从本质上说工作集方法、并行化方法、避免求解二次规划问题方法和几何方法都是严格的支持向量机训练过程，将得到基本相同的解。减少训练样本法、训练集分解法和增量学习法则是支持向量机训练过程的近似。工作集方法、并行化方法、避免求解二次规划问题方法和几何方法都可以有效地嵌入减少训练样本法、训练集分解法和增量学习法。

总的来讲，要使用支持向量机实现对大规模数据的处理，未来的方法应该是建立在训练集分解方法的基础上。这是因为计算机网络的发展方向是网格计算，未来支持向量机算法的物理平台也必然是网格计算。训练集分解法和增量学习方法还存在以下问题：第一，利用它们训练的支持向量机的一般化能力有时会低于标准支持向量机。第二，基于训练集的分解总会导致分类信息的损失。另外，对于基于训练集分解的并行学习方法还缺少理论上的结论。如何有效地将集成学习理论、PAC 理论、分类器偏置方差理论、信息融合理论、粒计算理论应用到基于训练集分解的并行学习方法将是未来研究需要解决的问题。

参 考 文 献

[1] Utiyama M, Isahara H. Large-scale text categorization[C]// The 9th Annual Meeting of the Association for Natural Language Processing. Tokyo Japan, 2003: 385-388

[2] 李德仁，关泽群. 空间信息系统的集成与实现[M]. 武汉：武汉大学出版社，2000

[3] 李德仁，王树良，李德毅. 空间数据挖掘理论与应用[M]. 北京：科学出版社，2006

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

[5] Vapnik V N. The nature of statistical learning theory[M]. Berlin: Springer, 1995

[6] Cristianini N, Campbell C, Burges C. Editorial: Kernel Methods: Current research and future directions[J]. Machine Learning, 2002, 46: 5-9

[7] Osuna E, Freund R, Girosi F. An improved training algorithm for support vector machines [C]// Proceedings of IEEE Workshop on Neural Networks and Signal Processing. Amelia Island, 1997: 276-285

[8] Platt J. Sequential minimal optimization: A fast algorithm for training support vector machines. Microsoft Research [R]. MSR-TR-98-14. 1998

[9] Joachims T. Making large - scale SVM learning practical. Advances in Kernel Methods-Support Vector Learning. MIT Press, 2000

[10] Chang C C, Lin C J. LIBSVM: A Library for support vector machines[OL]. http://www.csie.ntu.edu.tw/-cjlin/libsvm, 2001

[11] Dai L L, Huang H Y, Chen Z X. Ternary sequential analytic optimization algorithm for SVM classifier design[J]. Asian Journal of Information Technology, 2005, 4(3): 9-15

[12] 业宁，孔瑞祥，董逸生. MLSVM4----一种多乘子协同优化的SVM快速学习算法[J]. 计算机研究与发展，2005，42(9)：1467-1471

[13] 李建民，张钹，林福宗. 序贯最小优化的改进算法[J]. 软件学报，2003，14(5)：918-924

[14] Zanghirati G, Zanni L. A parallel solver for large quadratic programs in training support vector machines[J]. Parallel Computing, 2003, 29(4): 535-551

[15] Zanni L, Serafini T, Zanghirati G. Parallel software for training large scale support vector machines on multiprocessor systems [J]. Journal of Machine Learning, 2006, 7: 1467-1493

[16] PGPDT[OL]. http://www.dm.unife/gpdt

[17] Mangasarian O L, Musicant D R. Largrangian support vector machines[J]. Journal of Machine Learning Research, 2001, 1: 121-177

[18] Fung G, Mangasarian O L. Finite newton method for Lagrangian support vector machine classification [J]. Neurocomputing, 2003, 55(1/2): 39-55

[19] Mangasarian O L, Musicant D R. Active set support vector machine classification[G]// Advances in Neural Information Processing Systems. Cambridge: MIT Press, 2001: 577-583

[20] Fung G, Mangasarian O L. Proximal support vector machine classifiers[C]// Proc. of Knowledge Discovery and Data Mining. New York: ACM Press, 2001: 77-86

[21] 杨绪兵，陈松灿. 基于原型超平面的多类最接近支持向量机[J].

计算机研究与发展,2006,43(10):1700-1705
[22] Keerthi S S, Shevade S K, Bhattacharyya C, et al. A fast iterative nearest point algorithm for support vector machine classifier design[J]. IEEE Trans. Neural Networks, 2000, 11(1): 124-136
[23] Tsing I W, Kwok J T, Cheung P M. Core vector machines: fast svm training on very large data sets[J]. Journal of Machine Learning Research, 2005, 6(4): 363-392
[24] Shin H, Cho S. Fast pattern selection for support vector classifiers[C]// Proc. of the 7th Pacific-Asia Confe-rence on Knowledge Discovery and Data Mining. Korea, Seoul, 2003: 376-387
[25] Lee Y J, Mangasarian O L. RSVM: Reduced Support Vector Machines[C]// Proc. of the First SIAM International Conference on Data Mining. USA, Chicago, 2001: 1: 7
[26] Mangasarian O L. Generalized Support Vector Machines[G]// Advances in Large Margin Classifiers. Cambridge: MIT Press, 2000: 135-146
[27] Lin K M, Lin C J. A study on reduced support vector machines [J]. IEEE Transactions on Neural Networks, 2003, 14(6): 1449-1459
[28] Evgeniou T, Pontil M. Support Vector Machines with Clustering for Training with Very Large Datasets[C]// Proceedings of the Second Hellenic Conference on AI: Methods and Applications of Artificial intelligence. Greece, Thessaloniki, 2002: 346-354
[29] Boley D, Cao D W. Training support vector machine using adaptive clustering[C]// Proceedings of International Conference on Data Mining. USA, Florida, 2004: 235-242
[30] Yu H, Yang J, Han J W. Making SVMs scalable to large data sets using hierarchical cluster indexing[J]. Data Mining and Knowledge Discovery, 2005, 11(3): 295-321
[31] Zhang T, Ramakrishnan R, Livny M. BIRCH: An efficient data clustering method for very large databases[C]// Proceedings of the ACM SIGMOD International Conference on Management of Data. Canada, Montreal, Quebec, 1996: 103-114
[32] 李红莲,王春花,袁保宗,等. 针对大规模训练集的支持向量机的学习策略[J]. 计算机学报,2004,27(6):715-719
[33] 罗瑜,易文德,王丹璨. 大规模数据集下支持向量机训练样本的缩减策略[J]. 计算机科学,2007,34(10):211-213
[34] 郑志洵,杨建刚. 大规模训练数据的支持向量机学习新方法[J]. 计算机工程与设计,2006,27(13):2425-2426
[35] Bakir G H, Bottou L, Weston J. Breaking SVM complexity with cross-training[G]// Advances in Neural Information Processing Systems. Cambridge: MIT Press, 2005: 81-88
[36] Kwok J T. Support Vector Mixture for Classification and Regression Problems[C]// Proc. of International Conference on Pattern Recognition. Austria, Brisbane, 1998: 255-258
[37] Schwaighofer A, Tresp V. The Bayesian committee support vector machine[C]// Proceedings of the International Conference on Artificial Neural Networks. Austria, Vienna, 2001: 411-420
[38] Lu B L, Wang K A, Utiyama M. A part-versus-part method for massively parallel training of support vector machines[C]// Proc. of International Joint Conference on Neural Networks. Hungary, Budapest, 2004: 735-740
[39] Collobert R, Bengio S, Bengio Y. A Parallel Mixture of SVMs for Very Large Scale Problems[J]. Neural computation, 2002, 14(7): 1105-1114
[40] Huang G B, Mao K Z, Siew C K, et al. Fast modular network implementation for support vector machines[J]. IEEE Transactions on Neural Networks, 2005, 16(6): 1651-1663
[41] Vazquez-Navia A, Gutierrez-Gonzalez D, Parrado-Hernandez E, et al. Distributed support vector machines[J]. IEEE Transactions on Neural Networks, 2006, 17(4): 1091-1097
[42] Tresp V. A Bayesian Committee Machine[J]. Neural Computation, 2000, 12(11): 2718-2741
[43] Tresp V. The Generalization Bayesian Committee Machine[C]// Proc. of the Sixth ACM SGKDD International Conference on Knowledge Discovery and Data Mining. USA, Boston, 2000: 355-359
[44] Schwaighofer A, Tresp V. The Bayesian Committee support vector machine[C]// Proc. of International Conference of Artificial Neural Networks. Austria, Vienna, 2001: 411-417
[45] Tresp V, Schwaighofer A. Scalable Kernel Systems[C]// Proc. of International Conference of Artificial Neural Networks. Austria, Vienna, 2001: 285-291
[46] Tresp V. Scaling Kernel-Based Systems to Large Data Sets[J]. Data Mining and Knowledge Discovery, 2001, 5(3): 197-211
[47] Choudhury A, Prasanth B N, Andy J K. A Data Parallel Approach for Large-Scale Gaussian Process Modeling[C]// Proc. of the Second SIAM International Conference on Data Mining. VA, Arlington, 2002: 95-111
[48] Parrado-Hernandez E, Arenas-Garcia J, Mora-Jimenez I, et al. Growing support vector classifiers with controlled complexity [J]. Pattern Recognition, 2003, 36(8): 1479-1488
[49] Lu B L, Ichikawa M. Task decomposition and module combination based on class relations: a modular neural network for pattern classification[J]. IEEE Trans. Neural Networks, 1999, 10 (5): 1244-1256
[50] Wen Y M, Lu B L. A confident majority voting strategy for parallel and modular support vector machines[C]// International Symposium on Neural Networks. Nanjing, China, 2007: 525-534
[51] Jin X M, Wen Y M. On combining distributed SVMs by simple bayesian formalism rules[C]// Proceedings of the 6th International Conference on Machine Learning and Cybernetics. Hongkong, China, 2007: 3630-3635
[52] Kim H C, Pang S N, Je H M, et al. Support vector machine ensemble with bagging[C]// Proc. of First International Workshop SVM. 2002: 397-408
[53] Chawla N V, Moore T E, Hall L O, et al. Distributed Learning with Bagging like Performance[J]. Pattern Recognition Letters, 2003, 24(1): 455-471
[54] Chan P K, Stolfo S J. Experiments on multistrategy learning by meta-learning[C]// Proc. of the Second International Conference on Information and Knowledge Management. 1993: 314-323
[55] Dong J X, Krzyzak A, Suen C Y. Fast SVM training algorithm with decomposition on very large data sets[J]. IEEE Transaction on Pattern Analysis and Machine Intelligence, 2005, 27(4): 603-618
[56] Graf H P, Cosatto E, Bottou L, et al. Parallel support vector machines: the cascade svm[C]// Proceedings of NIPS. 2005: 521-528
[57] Wen Y M, Lu B L. A Cascade Method for Reducing Training Time and the Number of Support Vectors[C]// Proc. of International Symposium on Neural Networks. Dalian, China, 2004: 480-486

(下转第 31 页)

[41] Ferreira C. Gene Expression Programming: Mathematical Modeling by an Artificial Intelligenc. Angra do Heroismo Portugal, 2002

[42] Miller J F . An Empirical Study of the Efficiency of Learning Boolean Functions Using a Cartesian Genetic Programming Approach[A]// Banzhaf W, et al. , eds. Proceedings of the Genetic and Evolutionary Computation Conference (GECCO'99) [C]. San Francisco, CA, Morgan Kaufmann, 1999: 1135-1142

[43] Eusuff M M , Lansey K E . Shuffled Frog Leaping Algorithm : Amemetic Meta-heuristic for Combinatorial Optimation[M]. J. Heuristics, 2000

[44] Brameier M, Banzhaf W. A Comparison of Linear Genetic Programming and Neural Networks in Medical Data Mining[J]. IEEE Transactions on Evolutionary Computation, IEEE Press, NY, USA, 2001(5): 17-26

[45] Mihai O , Dumitrescu D . Multi Expression Programming [R]. Babes-Bolyai University, 2002

[46] Vedral V, Plenio M B. Basics of Quantum Computation[J]. Progress in Quantum Electronics, 1998, 22(1): 1239

[47] Davidor Y . An Ecological Model for Evolutionary Computing [J]. System/Control/Information, 1993, 31(8): 468-474

[48] Potter M A. The Design and Analysis of a Computational Model of Cooperative Covelution[D]. Fairfax, VA: George Mason University, 1997

[49] 李敏强, 寇纪淞. 遗传算法的基本理论与应用[M]. 北京: 科学出版社, 2002

[50] Menon A. Frontiers of Evolutionary Computation[M]. Kluwer Academic Publishers, 2004

[51] 陈毓屏, 康立山. 演化计算与 Data Mining 自动化[J]. 计算机工程与科学, 1999, 21(1): 1-8

[52] Schwefel H P, Wegener I, Weinert K. Advances in Computational Intelligence[M]. Springer, 2002

[53] Eiben A E, Smith J E. Introduction to Evolutionary Computing [M]. Springer Press, 2003

[54] Coello C A, Lamont C B. Application of Multi-objective Evolutionary Algorithms[M]. World Scientific Publishing Co. , 2004

[55] Yao X. Following the Path of Evolvable Hardware[J]. Communications of ACM, 1999, 42(4): 46-79

(上接第 25 页)

[58] Wen Y M , Lu B L . A Hierarchical and Parallel Method for Training Support Vector Machines[C]// Proc. of International Symposium on Neural Networks. Chengdu, China, 2005: 881-886

[59] Freund Y, Schapire R E. Experiments with a new boosting algorithm[C]// Proc. of the 13th International Conference on Machine Learning. Bari, Italy, 1996: 148-156

[60] Schapire R E. A brief introduction boosting[C]// Proc. of the sixteenth International Conference on Artificial Intelligence. 1999

[61] Pavlov D, Mao J C, Dom B. Scaling-up support vector machines using boosting algorithm [J]. Pattern Recognition, 2000, 2 (2000): 219-222

[62] Syed N A, Liu H, Suang K K. Incremental learning with support vector machines[C]// Proc. of the Workshop on Support Vector Machines at the International Joint Conference on Artificial Intelligence. San Diego, USA, 1999: 317-321

[63] Domeniconi C , Gunoppulos D . Incremental support vector machine construction[C]// Proc. of the IEEE International Conference on Data Mining. San Jose, USA, 2001: 589-592

[64] Cauwenberghs G, Poggio T. Incremental and decremental support vector machine learning[G]// Advances in Neural Information Processing Systems. Cambridge: MIT Press, 2001, 13: 409-415

[65] Diehl C P, Cauwenberghs G. SVM incremental learning, adaptation and optimization[C]// Proc. of IJCNN-03. Portland, Oregon, 2003: 2685-2690

[66] Bottou L, Vapnik V. Local learning algorithms[J]. Neural Computation, 1992, 4(6): 888-900

[67] Ralaivola L, d'Alche-Buc F. Incremental support vector machine learning: a local approach[J]. Lecture Notes in Computer Science, 2001, 2130: 322-329

[68] d'Alche-Buc F, Ralaivola L. Incremental learning algorithms for classification and regression: local strategies[C]// American Institute of Physics Conference Proceedings. 2001: 320-329

[69] Fung G, Mangasarian O L. Incremental support vector machine classification[C]// Proc. of the second SIAM International Conference on Data Mining. Virginia, Arlington, 2002: 247-260

[70] Fung G , Mangasarian O L . Proximal support vector machine classifiers[C]// Proc. of the 7th ACM Conference on Knowledge Discovery and Data Mining. San Francisco, USA, 2001: 77-86

[71] Tveit A, Hetland M L, Engum H. Incremental and decremental proximal support vector classification using decay coefficients [C]// Proc. the 5th International Conference on Data Warehousing and Knowledge Discovery. Czech Republic, Prague, 2003: 371-376

[72] Liu Y G, He Q M, Chen Q. Incremental batch learning with support vector machines[C]// Proc. of the 5th World Congress on Intelligence Control and Automation. Hangzhou, China, 2004: 1857-1861

[73] Wen Y M, Lu B L. Incremental Learning of Support Vector Machines by Classifier Combining [C]// Proc. of the 11th Pacific-Asia Conference on Knowledge Discovery and Data Mining. Nanjing, China, 2007: 904-911

[74] 曾文华, 马健. 一种新的支持向量机增量学习算法[J]. 厦门大学学报: 自然科学版, 2002, 41(6): 687-690

[75] 李忠伟, 张健沛, 杨静. 基于支持向量机的增量学习算法研究[J]. 哈尔滨工程大学学报, 2005, 26(5): 643-646

[76] 萧嵘, 王继成, 孙正兴, 等. 一种 SVM 增量学习算法-(-ISVM [J]. 软件学报, 2001, 12(12): 1818-1824

[77] 吴飞, 庄越挺, 潘云鹤. 基于增量学习支持向量机的音频例子识别与检索[J]. 计算机研究与发展, 2003, 40(7): 950-955

[78] 安金龙, 王正欧. 一种适合于增量学习的支持向量机的快速循环算法[J]. 计算机应用, 2003, 23(10): 12-17

[79] 李凯, 黄厚宽. 支持向量机增量学习算法研究[J]. 北方交通大学学报, 2003, 27(5): 34-37