一种可减少训练时间的分层并行支持向量机方法

文益民, 廖洪元, 周立华

(湖南工业职业技术学院 信息工程系, 湖南 长沙 410007)

[摘要] 基于支持向量的本质和并行计算方法, 提出了一种新的分层并行的机器学习方法以加速支持向量机的训练过程. 该方法首先按照分而治之的思想将原分类问题分成若干子问题, 然后将支持向量机的训练过程分解成级联的两个层次, 在每层采用并行的方法训练各个子支持向量机. 各层训练集中的非支持向量被逐步筛选掉, 交叉合并的规则保证问题的一致性. 仿真结果表明该方法在保证分类器推广能力的同时, 缩短了训练支持向量机的时间.

[关键词] 分层筛选, 支持向量机, 交叉合并

[中图分类号] TP181; [文献标识码] A; [文章编号] 1672-1292-(2005)01-0008-04

A Hierarchical and Parallel Support Vector Machines Algorithm for Reducing the Training Time

WEN Yimin, LIAO Hongyuan, ZHOU Lihua

(Department of Information Engineering, Hunan Industry Polytechnic, Hunan Changsha 410007, China)

Abstract Based on the essence of support vectors and parallel algorithm, the paper proposes a novel strategy of filtering the training samples in a hierarchical and parallel way to speed up the training of support vector machines (SVMs). During the training process, the entire classification problem is divided into several small sub-problems that can be handled in a parallel way. Having hierarchically filtered out the non-support-vector data, we can obtain the final training data set, which is used to train a SVM that will be used as the final pattern classifier. In order to keep the consistency, the cross-combining principle is introduced. The simulation results illustrate that our method speeds up training while maintaining the generalization accuracy of SVMs.

Key words hierarchical filtering; support vector machines; cross-combining

0 引言

支持向量机(SVM)[1,2]是一种重要的模式分类方法, 已被广泛地应用于文本分类、人脸检测与识别和语音识别等领域. 支持向量机的训练本质上是求解一个二次优化问题. 因此, 支持向量机训练过程的空间复杂度为 $O(N^2)$, 时间复杂度为 $O(N^3)$. 由于大规模模式分类问题中训练样本数量巨大, 训练样本不可能整体装入内存, 必需采用分而治之的方法将问题分解. 使用支持向量机解决大规模模式分类问题, 通常采用的学习方法有两种: 串行学习方法[3,4]和并行学习方法[5,6]. 串行学习方法将一个问题分成若干子问题, 然后对各子问题进行串行处理, 学习过程随着子问题的不断加入而不断延续. 但在处理大规模模式分类问题时, 使用串行学习方法会导致迭代次数过多和训练时间过长等问题. 并行学习方法按照分而治之的原则将原问题分解成若干子问题, 将各个子问题并行处理后再进行集成. 并行学习方法的优点是能缩短训练时间, 具有良好可扩充性, 但会增加支持向量的数目.

根据 Syed的结论[7], 支持向量集包含了训练样本集的全部分类信息, 也就是在训练样本中如果只将支持向量留下来训练支持向量机将得到与使用全部训练样本训练的支持向量机相同的结果. 为解决大规模模式分类问题, 根据支持向量的这个特点, 本文在串行和并行学习方法的基础上, 提出了

收稿日期: 2004-11-16
基金项目: 湖南省青年骨干教师资助项目(湘教通[2001]204号).
作者简介: 文益民(1969-), 上海交通大学博士研究生, 副教授, 主要从事统计理论、生物信息学等方面的教学与研究.
E-mail: yimin_wen@163.com

一种新的分层并行筛选训练样本的支持向量机方法.该方法在保证分类器推广能力的同时减少了支持向量机的训练时间,并减少了支持向量的数目.

1 支持向量机

假设有训练集 $S = \{(X_1, Y_1), \ldots, (X_N, Y_N)\}$, $X_i \in \mathbf{R}^n$ 表示一个训练样本, $Y_i \in \{-1, 1\}$, 其中 $Y_i = 1$表示X_i属于类别C_1, $Y_i = -1$表示X_i属于类别 C_2. 定义在 S 上的模式分类问题为:根据训练集 S训练一个分类函数 f, 对于未知类别的样本 X, 分类函数 f要能够对 X 所属的类别给出判断. 利用支持向量机方法解决此问题,需要寻找一个分类超平面: $\boldsymbol{W}^{\mathrm{T}}X + b = 0$, 其中 $\boldsymbol{W}$为权值向量, b为偏值, 以 $f(X) = \mathrm{sgn}(\boldsymbol{W}^{\mathrm{T}}X + b)$为分类函数. 根据统计学习理论[1,2], 为了使分类函数有尽可能好的推广能力,当训练集线性可分时,需对 $\boldsymbol{W}$和 b施加以下约束:对于 C_1 中的各样本X_i要求 $\boldsymbol{W}^{\mathrm{T}}X_i + b \geqslant 1$; 对于 C_2 中的各样本X_i 要求 $\boldsymbol{W}^{\mathrm{T}}X_i + b \leqslant -1$; 同时还要求超平面 $\boldsymbol{W}^{\mathrm{T}}X + b = 1$和 $\boldsymbol{W}^{\mathrm{T}}X + b = -1$之间的距离尽可能远. 根据数学规划理论,要找出满足以上条件的分类超平面,需要求解以下二次规划问题:

$$\text{Minimize } \Phi(\boldsymbol{W}) = \frac{1}{2}\boldsymbol{W}^{\mathrm{T}}\boldsymbol{W}$$

$$\text{Subject to } Y_i(\boldsymbol{W}^{\mathrm{T}}X_i + b) \geqslant 1 \quad i = 1, 2, \ldots, N \tag{1}$$

满足 $Y_i(\boldsymbol{W}^{\mathrm{T}}X_i + b) = 1$的训练样本 X_i称为支持向量.

当训练集 S线性不可分时,采用非线性变换 $X \mapsto \phi(X)$ 可将训练样本映射到一个更高维的特征空间,这里$X \in \mathbf{R}^n$, $\phi(X) = (\phi_1(X), \ldots, \phi_m(X))^{\mathrm{T}}$ $m > n$为正整数或无穷大. 于是,在原空间中的模式分类问题被转化成一个在更高维空间中的模式分类问题. 根据 Cover理论,在此高维空间中模式线性可分的可能性将增大. 此时需要在新的特征空间中寻求分类超平面: $\boldsymbol{W}^{\mathrm{T}}\phi(X) + b = 0$. 要使分类函数$f(X) = \mathrm{sgn}(\boldsymbol{W}^{\mathrm{T}}\phi(X) + b)$有尽可能好的推广能力, $\boldsymbol{W}$和 b仍然必须满足前面的约束条件. 考虑到在新的特征空间中训练集 S还是有可能线性不可分,对应每个样本X_i引入非负的松弛变量 ξ_i, 以衡量$\boldsymbol{W}$和 b违背约束条件的程度. 根据数学规划理论,为了寻找此分类超平面,需要求解下面的二次规划问题:

$$\text{Minimize } \Phi(\boldsymbol{W}, \xi) = \frac{1}{2}\boldsymbol{W}^{\mathrm{T}}\boldsymbol{W} + C\sum_{i=1}^{N}\xi_i$$

$$\text{Subject to } Y_i(\boldsymbol{W}^{\mathrm{T}}\phi(X_i) + b) \geqslant 1 - \xi_i$$

$$i = 1, 2, \ldots, N$$

$$\xi_i \geqslant 0, \quad i = 1, 2, \ldots, N \tag{2}$$

式中,正常数 C 是对违背约束条件的一种惩罚,为求解以上优化问题,需要求解其对偶问题:

$$\text{Minimize } \Pi(\alpha) = \sum_{i=1}^{N}\alpha_i - \frac{1}{2}\sum_{i=1}^{N}\sum_{j=1}^{N}\alpha_i\alpha_j Y_i Y_j(\phi(X_i)^{\mathrm{T}}\phi(X_j)) \tag{3}$$

$$\text{Subject to } \sum_{i=1}^{N}\alpha_i Y_i = 0$$

$$0 \leqslant \alpha_i \leqslant C, \quad i = 1, 2, \ldots, N$$

其中 α_i为问题(2)的拉格朗日乘子. 由于在求解上述对偶问题时,并不需要知道 $\phi(X_i)$ 的确切值. 于是,可记$K(X_i, X) = \phi^{\mathrm{T}}(X_i)\phi(X)$, 并将 K 称为核函数. 解对偶问题(3),求得如下的分类函数:

$$f(X) = \mathrm{sgn}\left(\sum_{i \in sv}\alpha_i Y_i K(X_i, X) + b\right) \tag{4}$$

上式中的 sv表示支持向量标号的集合,对应 $\alpha_i > 0$ 的训练样本 X_i 被称为支持向量. 因此, $f(X) \geqslant 0$时,可将未知样本 X 判为 C_1 类; $f(X) < 0$时,可将未知样本 X 判为 C_2 类.

2 分层并行筛选训练样本的方法

2.1 分层并行筛选训练样本

假设两类分类问题中属于类 C_1 的样本集为: $P = \{X_i\}_{i=1}^{L_m}$, 属于类 C_2的样本集为: $N = \{X_i\}_{i=1}^{L_n}$. 其中: X_i表示一个训练样本, L_m和L_n分别表示两类样本的数目,整个训练样本集可表示为 $T = P \cup N$. 分层筛选训练样本的过程可分为两步,这个过程如图 1所示:

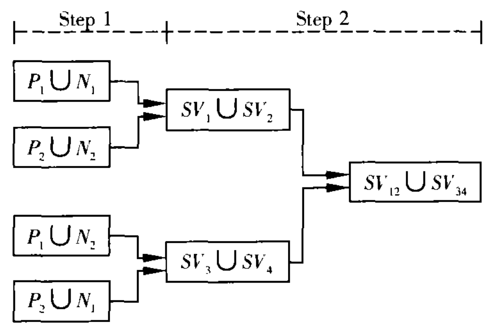

图 1 分层筛选训练样本示意图

第一步:根据事先确定的比率 $r(0 < r \leqslant 0.5)$, r为 0.5,将原训练集 P和 N分别分解为两个子集:

$$P_1 = \{X_i\}_{i=1}^{L_{p_1}}, \quad P_2 = \{X_i\}_{i=L_{p_1}+1}^{L_m},$$

$$N_1 = \{X_i\}_{i=1}^{L_{n_1}}, \quad N_2 = \{X_i\}_{i=L_{n_1}+1}^{L_n} \tag{5}$$

其中 $L_{p_1} = \lceil r * L_m \rceil$, $L_{n_1} = \lceil r * L_n \rceil$. 于是,原两类分类问题 T 可以分解成下列 4个规模较小的两类

分类问题:

$$T_1 = P_1 \cup N_1,\ T_2 = P_2 \cup N_2$$
$$T_3 = P_1 \cup N_2,\ T_4 = P_2 \cup N_1 \tag{6}$$

将得到的这4个子问题 T_1、T_2、T_3 和 T_4 按照通常训练SVM的方法(比如 SVM^{light})[4],并行地处理而得到4个支持向量机,它们的支持向量集合分别表示为: SV_1, SV_2, SV_3 和 SV_4.

第二步:将 SV_1 与 SV_2 及 SV_3 与 SV_4 分别合并成两个训练集:

$$T_{12} = SV_1 \cup SV_2,\qquad T_{34} = SV_3 \cup SV_4 \tag{7}$$

以上这种将 T_1 和 T_2、T_3 和 T_4 中的支持向量集合并的方法被定义为"交叉合并". 交叉合并能保证原问题分解以后,用于分类的信息几乎不会损失,这可以从棋盘数据分类问题中得到直观的说明. 如果将 T_1 和 T_3、T_2 和 T_4 的支持向量集合并,子集 P_1 将同时出现在 T_1 和 T_3 中,子集 P_2 也将同时出现在 T_2 和 T_4 中,这将在以后的训练集中人为带来两类训练样本数量的不平衡,从而导致分类信息的损失,因为支持向量机在处理不平衡问题时,分类面会靠近训练样本较少的一类.

采用通常训练SVM的方法对 T_{12} 和 T_{34} 进行并行处理,可以得到2个SVM,它们的支持向量集合分别表示为: SV_{12} 和 SV_{34}. 将这两个集合合并得到:

$$T_{final} = SV_{12} \cup SV_{34} \tag{8}$$

T_{final} 将被作为最终支持向量机的训练集. 由于经过了前两步的阶层筛选,非支持向量被逐步过滤. T_{final} 将含有与原整个训练集 T 相比较少的样本. 由于在分层筛选的过程中采用了交叉合并的方法, T_{final} 中几乎包含了原训练集 T 中全部的分类信息.

2.2 算法描述

前提:已知训练样本集 $T = P \cup N$, r. 算法:

(1) 根据 r 将 P 和 N 分解组合成4个规模较小的分类问题 T_1、T_2、T_3 和 T_4;

(2) 用支持向量机方法将 T_1、T_2、T_3 和 T_4 并行处理,得到与之对应的4个支持向量集合: SV_1, SV_2, SV_3 和 SV_4

(3) 按照交叉合并的原则将它们组合成两个分类问题 T_{12} 和 T_{34}, 采用支持向量机方法将它们并行处理得到两个支持向量集合 SV_{12} 和 SV_{34};

(4) 令 $T_{final} = SV_{12} \cup SV_{34}$;

(5) 以 T_{final} 作为训练集获得最终支持向量机,将其作为最终的模式分类器;

3 试验结果和分析

为了验证本文所提方法的有效性,本文进行了3个数值试验. 第一个试验采用人工数据,其余两个均采用实际问题的数据. 各实验中采用的核函数均为径向基函数. 在以下试验中,用"标准方法"表示使用通常的训练方法[4]对全部样本训练SVM. 用"分层并行方法"表示本文提出的训练方法. 实验平台为: 2.4 GHz 512MB RAM Pentium 4 PC.

3.1 棋盘(checkerboard)数据实验

一个[0, 200]×[0, 200]的棋盘被分成4块,所有训练样本均匀分布在这4块上. 位于[0, 100]×[0, 100]和[100, 200]×[100, 200]中的样本为正例样本,位于余下空间中的样本为反例样本. 为了检验本文所提出的方法的健壮性,随机生成了4个不同的训练集和一个共同的测试集. 每个训练集包含5 000个正例样本和5 000个反例样本,测试集中包含10 000个正例样本和10 000个反例样本. 将这4个两类分类问题分别记为: A_1、A_2、A_3 和 A_4. 分解比例 $r = 0.5$. 从表1可以看出本文提出的方法确实有健壮性,即在4次试验中均能保持推广能力,同时减少了训练时间和支持向量. 由表1得知:采用分层并行方法训练的SVM与使用标准方法训练的SVM有几乎一致的推广能力. 但是分层并行的方法减少了大约59%的训练时间,同时减少了大约12%的支持向量.

表1 当 $\sigma = 31.62$, $c = 1000$ 和 $r = 0.5$ 时,棋盘数据的4次试验情况,"*av*"表示4次实验的平均结果

	训练方法	准确率/%		训练时间/s	支持向量数目
		训练集	测试集		
A_1	标准方法	99.84	99.81	46.39	93
	分层并行方法	99.78	99.72	13.08	81
A_2	标准方法	99.89	99.72	38.00	96
	分层并行方法	99.85	99.70	15.34	83
A_3	标准方法	99.93	99.84	32.44	88
	分层并行方法	99.86	99.75	13.45	79
A_4	标准方法	99.89	99.81	35.50	94
	分层并行方法	99.92	99.83	19.87	84
av	标准方法	99.89	99.80	38.08	93
	分层并行方法	99.85	99.75	15.44	82

3.2 Forest CoverType实验

Forest CoverType数据来自UCI[8]. 这个数据库用来预测原始地区森林的覆盖情况. 该数据集包括7类数据共计581 012条. 本文抽取该数据集第六类和第七类的数据构成分类问题 A_5, 抽取第四类和第五类数据构成分类问题 A_6. 在每个问题中,随机取一半数据作为训练数据,另一半作为测试数据.

在这个实验中,选择参数 $\sigma = 100$, $c = 1\,000$. 分解比例 r 分别取值0.1和0.5, 以检验 r 对分层并行筛选训练样本方法的影响. 从表3和表4中可以

看出 r对最终支持向量机的推广能力没有影响, 但它会影响分层并行训练的总时间. 显然 $r = 0.5$是最好的分割比例. 与串行学习方法不同, 由于被分层筛选掉的样本不会再进入训练过程, 采用分层并行筛选训练样本的方法能加速支持向量机的训练过程. 如果原问题中支持向量所占总训练样本的比例较小, 则本文提出的训练方法将非常有效.

表 3　问题 A_5 中两种训练方法在 r取不同值时的比较

	标准方法	分层并行方法	
		$r = 0.1$	$r = 0.5$
训练时间 /s	461.92	508.05	268.20
支持向量数目	3 943	3 827	3 778
训练准确率 /%	100	100	100
测试准确率 /%	99.82	99.82	99.82

表 4　问题 A_6 中两种训练方法在 r取不同值时的比较

	标准方法	分层并行方法	
		$r = 0.1$	$r = 0.5$
训练时间 /s	27.02	36.66	24.69
支持向量数目	1 661	1 605	1 571
训练准确率 /%	100	100	100
测试准确率 /%	96.93	96.93	96.93

3.3　文本分类实验

文本分类试验的数据采用日本读卖新闻提供的文本分类数据库. 经过特征提取后, 特征空间的维数为 5 000. 本文从该数据库中提取了如表 5所示的 3类数据. 任选其中的两类构成一个两类分类问题, 于是得到 3个两类问题: A_7, A_8和 A_9. 这里 $r = 0.5$.

从表 6中可以看出分层并行方法在保证分类器推广能力的同时, 大大减少了分类器的训练时间, 并且减少了支持向量. 由于在文本分类问题中支持向量在训练样本集中所占的比例通常较小, 因此, 本文提出的方法对于文本分类问题非常有效.

表 5　文本分类数据的分布情况

类别	数据	
	训练集	测试集
事故	34 044	8 483
健康	35 932	7004
业余生活	33 590	7 702

表 6　当 $\sigma = 2$, $c = 64$和 $r = 0.5$时, 文本分类的数值仿真结果

	训练方法	A_7	A_8	A_9
训练集准确率 /%	标准方法	97.74	97.93	96.67
	分层并行方法	97.73	97.75	96.67
测试集准确率 /%	标准方法	95.81	96.01	93.62
	分层并行方法	95.83	96.02	93.62
训练时间 /s	标准方法	12 664	7 458	18 566
	分层并行方法	9 519	4 491	15 060
支持向量数目	标准方法	10 933	9 445	12 750
	分层并行方法	10 553	9 222	12 387

4　结束语

基于支持向量能够代表整个训练集的分类信息这一特点, 为了加速支持向量机的训练, 本文提出了一种分层并行筛选训练样本的支持向量机方法. 该方法降低了问题的规模. 全部数值仿真试验表明: 1)本文所提方法与串行方法相比, 在保证分类器推广能力的前提下, 能提高支持向量机的训练速度. 2)与并行方法相比, 本文提出的方法没有使支持向量增加, 反而减少了支持向. 这一优点有利于提高支持向量机进行识别时的响应速度, 有利于降低支持向量机在软件和硬件实现时的成本. 本文的实验表明支持向量减少不超过 10%时, 分类器的推广能力不变, 这从另一个方面验证了 Syed[7]中有关支持向量减少数目的结论. 综上所述: 本文提出的方法在解决大规模模式分类问题中有着很强的实用价值.

[参考文献]

[1] Vladimir N. Vapnik. Statistical Learning Theory[M]. NewYork: Springer-Verlag, 1998.

[2] 许建华, 张学工, 李衍达. 支持向量机的新发展[J]. 控制与决策, 2004, 19: 481-484.

[3] Edgar Osuna, Robert Freund, Federico Girosi. An improved training algorithm for support vector machines[A]. Proceedings of IEEE[C]. NNSP, 1997. 276-285.

[4] Thorsten Joachims. Making large-scale SVM learning pratical[A]. Advances in KernelMethods-Support Vector Learning. Cambridge[C]. MIT Press, 2000. 169-184.

[5] Lu Baoliang, WangKaian, UtiyamaM, et al. A part-versus-part method for massively parallel training of support vector machines[A]. Proceedings of IEEE/INNS Int Joint Conf on Neural Networks (IJCNN2004)[C]. Hungary: Budapest, 2004. 735-740.

[6] Anton Schwaighofer, Vloker Tresp. The bayesian committee support vector machine[J]. Lecture Notes in Computer Science, 2001, 2130: 411-417.

[7] Nadeem Ahmed Syed, Huan Liu, Kah Kay Sung. Incremental learning with support vector machines[A]. Proceedings of the Workshop on Support VectorMachines at the International Joint Conference on Artificial Intelligence[C]. Sweden: Stockholm, 1999.

[8] Blake C L, Merz C J. UCI(ftp://ftp.ics.uci.edu/pub/machines-learning-database).

[责任编辑: 刘健]