基于商空间粒度理论的大规模 SVM 分类算法*

文贵华[a], 向　君[b], 丁月华[b]
(华南理工大学　a. 计算机科学与工程学院; b. 计算机应用工程研究所, 广州 510641)

摘　要: 利用商空间粒度理论对已有的 SVM 分类算法进行改进, 给出了一种新的 SVM 分类算法——SVM-G。该算法将 SVM 分类问题划分成两个或多个子问题, 从而降低了 SVM 分类复杂度。实验表明, 改进的算法适用于处理大数据量的样本, 能在保持分类精度的情况下有效地提高支持向量机的学习和分类速度。

关键词: 粒度; 商空间; 支持向量机; 分类; 机器学习
中图分类号: TP301.6　　**文献标志码:** A　　**文章编号:** 1001-3695(2008)08-2299-03

Large-scale SVM classification algorithm based on granularity of quotient space theory

WEN Gui-hua[a], XIANG Jun[b], DING Yue-hua[b]
(a. School of Computer Science & Engineering, b. Research Institute of Computer Application Engineering, South China University of Technology, Guangzhou 510641, China)

Abstract: This paper improved the existing SVM algorithm with the granularity of the quotient space theory, proposed a new SVM algorithm(SVM-G). The improved algorithm divided SVM classification problem into two or more sub-issues, thereby reducing the computation complexity of SVM classification. Experimental results indicate that the improved algorithm is suitable for processing large number of observations and can effectively accelerate the speeds of SVM learning and classifying while keeping the classification precision.

Key words: granularity; quotient space; SVM(support vector machine); classification; machine learning

粒度计算是信息处理的一种新的概念和计算范式[1]。目前, 研究得较多且较成熟的一种粒度计算理论是商空间理论[2]。文献[2]提出了基于商空间的问题求解理论, 商空间将问题表述成不同的粒度世界, 进而进行相应的粒度计算。粒度计算的优点在于使对问题的解决摆脱了一些繁琐且非关键的过程, 抓住问题的关键或本质, 从适当的层次(粒度)来研究问题, 最终可快速获得问题的精确解或近似解[3~5]。

支持向量机(SVM)是由 Vapnik[6] 于 20 世纪 90 年代初提出的一种新的机器学习方法。SVM 算法利用凸二次规划(quadratic programming, QP)、Mercer 核、稀疏解和松弛向量等多项技术, 具有结构简单、训练误差低和泛化能力好等优点, 因而被广泛地应用于分类学习问题中。但是当样本规模较大时, SVM 算法在二次寻优过程中需要存储核矩阵和进行大量矩阵运算, 使得运算时间过长。目前, 改进算法主要有 Vapnik 等人[7] 提出的选块算法、Osuna 等人[8] 提出的分解算法、Platt[9] 提出的序列最小最优化法(sequential minimal optimization, SMO)等。

以上几种改进算法, 虽然有效地改进了 SVM 分类速度, 但仍是对原始样本集进行处理, 也即一直是在较精的粒度上进行运算。基于问题求解的商空间理论用可用空间结构来表示样本的结构。本文依据这个理论, 将商空间粒度计算应用于 SVM 分类, 采用距离度量空间的手段来解决大规模样本集的 SVM 分类问题。据此, 提出一种基于商空间理论的 SVM 分类算法(SVM-G), 该算法用空间距离表示信息粒度, 采用分层递阶的方法, 由粗到精地处理样本点。实验表明, 该算法对大样本集的分类能在保持训练精度的同时进一步提高训练和分类的速度, 具有降低计算复杂度和空间复杂度、适于处理样本数较多的 SVM 分类问题等优点。

1　理论准备

1.1　商空间理论

在实际问题求解过程中, 随着求解问题的不同, 需要不同粒度的世界描述。有时解决某一问题时, 同时需要若干不同粒度的世界。商空间理论用三元组(X, f, T)描述一个问题。其中: x 表示问题的论域; $f(\cdot)$ 是论域属性; T 是论域的结构, 指论域 X 中各元素的相互关系。分析或求解问题(X, f, T)是指对论域 X 及其有关的结构、属性进行分析、研究。可以对 X 的大小粒度进行简化, 产生一个新的较大粒度的论域$[X]$, 那么把原问题(X, f, T)变成新的层次上的问题($[X]$, $[f]$, $[T]$), 即从较粗粒度$[X]$去考察问题。

1.1.1　不同粒度世界的描述

定义 1　设 X、Y 是两个集合, $X\times Y$ 是 X 与 Y 的积集, $R\subset X\times Y$。设 $\forall(x, y)\in X\times Y$, 有$(x, y)\in R$, 则称 X 与 Y 有关系 R, 记为 xRy。称 R 为 $X\times Y$ 上的一个关系。

定义 2　$\forall x\in X$, 令$[x]=\{y\mid x\sim y\}$, 称为 x 的等价类。

定义 3　令$[X]=\{[x]\mid x\in X\}$, 称$[X]$是关于 R 的商集。

故商集是将等价类$[X]$看成新元素而构成的新空间, 是原空间粗粒度的论域。

商空间理论将不同粒度的世界与数学上的商集概念统一

收稿日期: 2007-08-02; 修回日期: 2007-11-10　　基金项目: 广东省科技攻关资助项目(2007B030803006)

作者简介: 文贵华(1968-), 男, 湖北利川人, 副教授, 硕导, 主要研究方向为人工智能、认知科学、数据挖掘与知识发现; 向君(1983-), 女, 湖南常德人, 硕士研究生, 主要研究方向为人工智能、Web 数据挖掘、智能信息处理(poiuy8303@163.com); 丁月华(1950-), 男, 广东惠州人, 广州中间件研究中心主任, 教授, 主要研究方向为计算系统应用.

起来, 或者说以商集作为不同粒度世界的数学模型。

1.1.2 (X, T) 拓扑空间

对于一个具体对象(X, f, T), 结构也很重要。当一个论域被划分, 其结构也发生了变化。一般讲结构 T 很广泛、很复杂, 商空间理论有详细讨论。这里只引用其中(X, T) 对拓扑空间的部分结论。

定义 4 设 X 是一集合, T 是 X 的一个子集族。若 T 满足: a) $\varnothing, X \in T$; b) 若 $u_1, u_2 \in T$, 则$(u_1 \cap u_2) \in T$; c) 若 $u_{\alpha, \alpha \in I} \in T$, 则 $\bigcup_{\alpha \in I} u_\alpha \in T$。其中: I 是指标集, I 的势可以是任意的。称 T 是 X 上的一个拓扑, 称(X, T) 为一拓扑空间; $u \subset X$ 称 u 是开集$\Leftrightarrow u \in T$。

在一个集合 X 上给定一个拓扑, 就给出 X 中各元素之间的一个互相关系的结构。

设 R 是 X 上的一个等价关系。对 R 可以得到对应的商集, 记为$[X]$。在$[X]$上定义由 T 诱导出的拓扑, 记为$[T]$, 其表示为$[T] = \{u \mid p^{-1}(u) \in T, u \subset [X]\}$。称$[T]$ 为商拓扑, $([X], [T])$ 为商拓扑空间。

构造了拓扑空间, 有以下拓扑学的命题:

命题 1 $p: (X, T) \to ([X], [T])$ 是自然投影, 故 p 是连续的。若 $A \subset X$ 且 A 是 X 中的连通集, 则 $p(A)$ 是$[X]$ 的连通集。

这表明, 若一个问题在原论域中有解(是连通的), 在适当的粗粒度论域上$[X]$ 有解。也就是说, 对于 SVM 分类问题而言, 在其所对应的商空间 $X(\lambda)$ $(\lambda = 1, 2, \cdots, p)$ 上的 SVM 分类同样有效。

1.2 SVM 分类原理

SVM 来源于线性可分问题的最优分类超平面, 寻找一个满足分类要求的最优分割超平面, 以最大间隔将数据分开。对于线性不可分问题, 通过选择某种非线性映射, 将输入空间映射到一个维数更高的特征空间, 在这个空间构造最优分类超平面。可以证明, 如果选用适当的映射函数, 大多数输入空间线性不可分问题在属性空间可以转换为线性可分问题。

给定训练集(x_i, y_i), $i = 1, 2, \cdots, l$。其中: $x_i \in R^n$ 且 $y_i \in \{-1, 1\}$, 最优分割超平面的计算问题可以描述为一个条件极值问题。Vapnik 通过求解下列 Lagrange 函数的鞍点来获得其最优解:

$$L_p = \| w \|^2 / 2 + C \sum_i \xi - \sum_i \alpha_i \{ y_i (\Phi(x_i) w + b) - 1 + \xi_i \} - \sum_i \mu_i \xi \tag{1}$$

其中: w 和 b 分别为属性空间最优超平面的法向量和阈值; ξ 和 μ_i 分别为非负的 Lagrange 乘数; C 为非负的误差控制参数; $\Phi(x_i)$ 为核函数。

由 Karush-Kuhn-Tucker 定理可知, 最优解满足以下条件:

$$w = \sum_i \alpha_i y_i \Phi(x_i) \tag{2}$$

$$0 \le \alpha \le C, \sum_i \alpha_i y_i = 0 \tag{3}$$

可得到分类器函数为

$$f(x) = \mathrm{sgn}\left(\sum_{i \in SV} y_i \alpha_i \Phi(x_i) - b \right) \tag{4}$$

一般情况下, 在式(2) 的展开式中, 大多数系数 α 为零值, 并不影响分类的结果。对 w 的确定有贡献的仅仅是非零值的 α 所对应的 x_i, 这就是 SV(support vector, 支持向量)。因此, SV 集充分描述了整个训练数据集数据的特征, 对 SV 集的划分等价于对整个数据集的分割。

在大多数情况下, 训练集中 SV 的数量只占训练样本集的很少一部分, 即 $N_{SV} << N_{train}$。因此, 首先可以在粗粒度上排除掉很多非 SV 的样本点, 然后使用包含 SV 的空间集取代训练样本集进行分类学习。当支持向量的个数在训练数据集中的比重很小且分布较集中时, 可以在不影响分类精度的同时有效减少训练时间。

2 改进算法 SVM-G

根据商空间粒度计算理论, 信息粒度最先来自整个论域 X, 然后根据其相似性、不可分性或一致性等原则对论域进行不同的划分, 形成不同粒度。论域的划分原则上是一个复杂的问题, 通常要根据提出的目标而定, 且与具体论域有关。本文使用欧氏空间中的间隔分析形成相应粒度(其具体构造见 2.1 节), 提出一种改进的 SVM 算法——SVM-G。从距离间隔角度来看商空间 $X(\lambda)$, $X(\lambda)$ 中的元素通过将属于同一间隔区间的元素看成一个元素, 从这个角度理解, $X(\lambda)$ 具有直观的物理意义。

改进后的算法将在不同的粒度(层次) 上进行 SVM 分类训练, 即在粗粒度上进行 SV 的粗选, 在精粒度上进行 SV 的精确判定。

2.1 粒度计算

在欧氏空间 R^n 中, 粒度的空间间隔向量$[b_k]$ 可定义为 n 个间隔的笛卡尔乘积, 表示为

$$[b_k] = [b_{k,1}] \times [b_{k,2}] \times \cdots \times [b_{k,n}]; k = 1, 2, \cdots, N \tag{5}$$

其中: $b_{k,i}$ 为间隔, $[b_{k,i}] = [b_{k,i}^{\min}, b_{k,i}^{\max}]$, $1 \le i \le n$; n 为样本空间维数。R^2 空间中的间隔向量如图 1 所示。

假定 X 为 n 维欧氏空间 R^n 的紧支撑子集, 且所有训练数据 $x_i \in X$, X 表示为所有间隔向量的并集, 即 $X = [b_1] \cup [b_2] \cup \cdots \cup [b_{l_1}]$, 且$[b_k] \cap [b_j] = \varnothing$。其中 $k \ne j$, $1 \le k, j \le l_1$, l_1 表示第一层中数据粒度数。可定义粒度空间中的样本(a_k, t_k):

$$a_k = \mathrm{mean}\{ x_i \mid x_i \in [b_k] \} \tag{6}$$

$$t_k = \mathrm{sign}\left(\sum_{x_i \in [b_k]} y_i \right) \tag{7}$$

本算法中所有的间隔宽度 d 相等。其中: $d = b_{k,i}^{\max} - b_{k,i}^{\min}$, 因此 d 为调节空间粒度的参数。令所有 $x_i \in [d_{\min}, d_{\max}]^n$ 且 $d_{\max} - d_{\min} = r \times d$, 则所有间隔向量$[b_k]$ 表示为

$$[b_1] = [d_{\min}, d_{\min} + d]^n$$

$$[b_2] = [d_{\min} + d, d_{\min} + 2d] \times [d_{\min}, d_{\min} + d]^{n-1}$$

$$\cdots$$

$$[b_k] = [d_{\min} + s_1 d, d_{\min} + (s_1 + 1) d] \times \cdots \times [d_{\min} + s_n d, d_{\min} + (s_n + 1) d]$$

$$\cdots$$

$$[b_l] = [d_{\max} - d, d_{\max}]^n \tag{8}$$

其中:

$$s_n = (k - 1) \bmod (r^{n-1}) + 1$$

$$s_{n-1} = [(k - 1) - (s_n - 1) r^{n-1}] \bmod r^{n-2} + 1$$

$$\cdots$$

$$s_m = \left[(k - 1) - \sum_{i=m+1}^{n} (s_i - 1) r^{i-1} \right] \bmod r^{m-1} + 1$$

$$\cdots$$

$$s_1 = \left[(k - 1) - \sum_{i=2}^{n} (s_i - 1) r^{i-1} \right] + 1; \ 1 \le m \le n \tag{9}$$

2.2 算法流程

SVM-G 分类算法至少包含两个阶段:

a) 通过对粒度空间的选择而粗略地去除对构造最优分类超平面无用的样本点。具体做法是利用粒度空间中新构建的样本点, 然后利用新样本点进行 SVM 训练, 找出支持向量, 这些支持向量也就代表了原样本集中支持向量所在空间。参数 d 可用来控制粒度大小, d 越大表示粒度越大, 则新构建的样本点比原样本点越少。在该层次上的 SVM 训练比原先整个样本上的 SVM 训练复杂度降低。

b) 将原始样本点从第一阶段所得的粒度空间中释放, 这些样本点用来进行最终的 SVM 训练。由于在第一阶段中很多样本点已去除, 该层进行 SVM 训练的样本点也相对较少, 计算

复杂度降低。

如果在第二阶段中样本点仍然很多, 则可将所得样本点按照第一阶段中的步骤再处理一次, 进一步滤除非支持向量所在空间集合。这样可保证在每一阶段中的训练集不会过大, 从而降低整个 SVM 分类中的空间复杂度和时间复杂度。

SVM-G 分类算法描述如下:

训练集 $\varphi_0 = \{(x_i, y_i) \mid x_i \in R^n, y_i \in \{-1, 1\}; i = 1, 2, \cdots, N\}$

a) 将输入数据 $x_i (i = 1, 2, \cdots, N)$ 规范化到空间 $[d_{\min}, d_{\max}]^n$。

b) 按照式(8) 和(9) 选择间隔向量 $[b_k]$, 然后按照式(6) 和(7) 形成粒度空间中的代表向量 (a_k, t_k)。由其中非零向量构成新的训练集: $\varphi_1 = \{(a_k, t_k) \mid a_k \in [d_{\min}, d_{\max}]^n, t_k \in \{-1, 1\}; k = 1, 2, \cdots, l\}$。

c) 对数据集 φ_1 进行 SVM 训练, 由支持向量粒度空间构成集合 θ, 将原训练样本从集合 θ 中释放出来, 形成另一个集合 $\varphi_2 = \{(x_i, y_i) \mid x_i \in b_k, a_k \in \theta\}$。

d) 如果集合 φ_2 的大小比 M 大, 则转 b); 否则转 e)。M 为预先设定的常量。

e) 对数据集 φ_2 进行 SVM 训练, 使用测试集进行测试。

3　实验结果分析

在本实验中, 所用数据均来源于 UCI 机器学习库[10]。选用其中的 satimage、letter 和 shuttle 三种基准数据集, 它们的基本信息如表 1 所示。

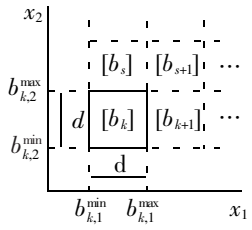

图 1　R^2 空间中的间隔向量

表 1　基准数据集

基本信息	数据集		
	satimage	letter	shuttle
训练样本数	4 435	15 000	43 500
测试样本数	2 000	5 000	14 500
类别数	6	26	7
输入维数	36	16	9

算法通过 MATLAB7 编程实现, 实验在 P4-266 的 PC 机上运行, 内存为 256 MB。实验使用了 LIBSVM 软件包中的 MATLAB 接口[11], 利用该接口中的 SVM 分解算法实现基本 SVM 训练; 同时在 SVM-G 分类的两个训练阶段也是使用该 MATLAB 接口中的分解算法来实现; 并将本文方法与 LIBSVM 软件包中方法的分类效果进行比较。

实验中所有的训练和测试数据都归一化到 $[-1, 1]$ 内, 取 $r = \lceil \sqrt[n]{N} \rceil$ (N 为样本数, n 为样本输入维数)。分类核函数采用径向基函数(radial-basis function, RBF), 交叉验证中参数步长分别为 $C = [2^{12}, 2^{11}, 2^{10}, \cdots, 2^{-2}]$ 和 $\gamma = [2^4, 2^3, 2^2, \cdots, 2^{-10}]$。SVM-G 与原 SVM 算法的准确性如表 2 所示。

表 2　算法准确性比较

数据集	SVM-G		SVM	
	C, γ	正确率/%	C, γ	正确率/%
satimage	$2^3, 2^{-1}/2^3, 2^0$	91.05	$2^4, 2^0$	91.78
letter	$2^6, 2^1/2^5, 2^1$	97.88	$2^4, 2^2$	97.90
shuttle	$2^{11}, 2^0/2^9, 2^1$	99.84	$2^{11}, 2^3$	99.92

实验结果表明, SVM-G 分类法的准确度比 SVM 分类稍有降低。其原因主要是在第一阶段中可能会滤除掉部分支持向量样本点, 造成测试准确性比传统 SVM 有所下降。但从实验结果可以看出, 准确性只是百分比小数位的改变, 所以对分类效果的影响很小。

表 3 给出了 SVM-G 和 SVM 的训练样本数量、支持向量数目和计算时间; 对于 SVM-G 分别给出了第一阶段和第二阶段训练中的样本数量和相应的支持向量数目。表中的运行时间包括训练时间和测试时间。虽然训练时间会受参数设置的影响, 但从实验结果可以清楚看到, SVM-G 算法的运行时间均有所下降, 并且可以注意到对于 shuttle 数据集来说, 改进算法的运行速度是原 SVM 算法的 5 倍多。这也表明当样本集中的支持向量数目少时, 第一阶段中就可移除掉大量训练样本点, 第二阶段中训练的样本点就比原样本集大为减少, 从而缩短整个计算时间。

表 3　测试结果比较

数据集	算法	样本数量	SVs	时间/s
satimage	SVM-G	3 436/1 868	396/864	15.89
	SVM	4 435	1 613	25.19
letter	SVM-G	1 474/14 509	13 24/7 574	341.33
	SVM	15 000	7 742	438.40
shuttle	SVM-G	316/1 258	116/135	9.84
	SVM	43 500	278	48.06

4　结束语

本文针对 SVM 算法在大样本集应用中训练和分类速度慢的问题, 将商空间粒度理论应用于 SVM 分类问题, 并对已有的 SVM 算法进行改进, 提出了一种新的改进算法——SVM-G。该算法将分类样本空间进行由粗到细的划分, 即采用分层递阶的方法, 将 SVM 分类中 QP 求解问题分解成两个或更多个子 QP 问题, 使得问题求解过程中 Q 矩阵的阶由 $N \times N$ 阶转换为 $(l_1 \times l_1 + l_2 \times l_2 + \cdots + l_p \times l_p)$ 阶(其中: l_1、l_2 和 l_p 分别为第 1、第 2 和第 p 阶段训练样本)。改进的算法能在保持分类精度上减少分类运算时间。实验表明, SVM-G 算法实现了比原 SVM 分解算法更快的运行速度; 尤其对于支持向量数目相对较少的大规模 SVM 分类问题来说, 该方法非常有效。

参考文献:

[1] AO Y Y, ZHONG Ning. Potential applications of granular computing in knowledge discovery and data mining[C]//Proc of World Multi-Conference on Systemics, Cybernetics and Informatics. [S. l.]: Computer Science and Engineering, 1999: 573-580.

[2] 张钹, 张铃. 问题求解理论及应用[M]. 北京: 清华大学出版社, 1990.

[3] 张燕平, 张铃, 吴涛. 不同粒度世界的描述法——商空间法[J]. 计算机学报, 2004, 27(3): 328-333.

[4] 高平安, 蒙祖强, 蔡自兴. 基于粒度计算的数据分类建模研究[J]. 计算机应用研究, 2007, 24(3): 37-40.

[5] 卜东波, 白硕, 李国杰. 聚类/分类中的粒度原理[J]. 计算机学报, 2002, 25(8): 810-816.

[6] VAPNIK V N. The nature of statistical learning theory[M]. New York: Springer-Verlag, 1995.

[7] VAPNIK V, GOLOWICH S, SMOLA A. Support vector method for function approximate, regression estimation, and signal processing[M]. Cambridge: MIT Press, 1997: 281-287.

[8] OSUNA E, FREUND R, GIRROSI F. Improved training algorithm for support vector machines[C]//PRINCIPLE J, GILES L, MONGAN N, *et al*. Proc of IEEE Workshop on Neural Networks and Signal Processing. Amelia Island: IEEE Press, 1997: 276-285.

[9] PLATT J C. Fast training of support vector machines using sequential minimal optimization[M]// SCHOLKOPF B, BURGES C J C, SMOLA A J. Advances in kernel methods: support vector learning. Cambridge: MIT Press, 1999: 185-208.

[10] ASUNCION A, NWEMAN D J. UCI machine learning repository[D]. California: Department of Information and Computer Science, University of California, 2007.

[11] CHANG C C, LIN C J. LIBSVM: a library for support vector machines[EB/OL]. (2006-11-17) [2007-01-07]. http://www.csie.ntu.edu.tw/cjlin/libsvm.