第36卷 第19期 计 算 机 工 程 2010年10月
Vol.36 No.19 Computer Engineering October 2010
·人工智能及识别技术· 文章编号：1000—3428(2010)19—0174—03 文献标识码：A 中图分类号：TP181

基于相似性分析的SVM快速分类算法

朱 方[a]，顾军华[b]，杨欣伟[b]，杨瑞霞[a]
(河北工业大学 a. 信息工程学院；b. 计算机科学与软件学院，天津 300401)

摘 要：针对支持向量机(SVM)分类速度取决于支持向量数目的应用瓶颈，提出一种SVM快速分类算法。通过引入支持向量在特征空间的相似性度量，构建特征空间中的最小支撑树，在此基础上将支持向量按相似性最大进行分组，依次在每组中找到决定因子和调整因子，用两者的线性组合拟合一组支持向量在特征空间的加权和，以减少支持向量的数量，提高支持向量机的分类速度。实验结果证明，该方法能以很小的分类精度损失换取较大的分类时间缩减，满足SVM实时分类的要求。

关键词：支持向量；相似性系数；最小支撑树；决定因子；调整因子

Fast SVM Classification Algorithm Based on Similarity Analysis

ZHU Fang[a], GU Jun-hua[b], YANG Xin-wei[b], YANG Rui-xia[a]
(a. Institute of Information Engineering; b. Institute of Computer Science and Software, Hebei University of Technology, Tianjin 300401, China)

【Abstract】 Aiming at the bottleneck of SVM that the speed of classification depends on the number of support vectors, this paper proposes a fast classification algorithm for SVM. In feature space it constructs the minimum spanning by introducing the similarity measure and divides the support vectors into groups according to the maximum similarity. The determinant factor and the adjusting factor are found in each group by some rules. In order to simplify the support vectors, it takes the linear combination of determinant factor and adjusting factor to fit the weighted sums of support vectors in feature space, so that the speed of classification is improved. Experimental results show that the algorithm can get higher reduction rate of classification time by minor loss of classification accuracy and it can satisfy the requirements of real-time classification.

【Key words】 support vector; similarity coefficient; minimum spanning tree; determinant factor; adjusting factor

1 概述

支持向量机(SVM)是在统计学习理论基础上发展起来的一种新的机器学习方法，作为一种具有高分类精度的分类器，已经在很多领域得到了广泛的应用，然而，SVM有一个明显的缺点，即分类速度取决于支持向量的数目，如果支持向量数目很大，SVM的分类速度就会很慢，不能满足实时分类的要求，这在很大程度上限制了SVM的应用。目前，针对此问题出现了一系列的快速分类算法[1-6]：

文献[1-2]提出的快速逼近核空间的分类函数方法将寻找简约向量的过程转化为一个多参数的无约束优化问题。

文献[3]提出的简化算法通过在特征空间消减那些能被其他支持向量线性表示的冗余支持向量来实现支持向量的简化。

文献[4-5]提出的FCSVM方法通过优化的方式适当地变换分类函数的形式，以减少分类函数中支持向量的数量。

文献[6]提出的基于核聚类的快速分类算法先寻找特征空间中的聚类中心在输入空间中的原像，然后形成简约向量集来代替支持向量，以减少支持向量的数量。

在分析现有SVM快速分类算法的基础上，本文提出一种基于特征空间相似性分析的快速分类方法，能够满足实时分类的要求，且避免了以往SVM快速分类算法存在的因局部极值而导致的分类精度过度损失、约简率不高、对核函数限制等问题。

2 基于特征空间相似性分析的SVM快速分类算法

对于2类分类问题，SVM的分类判别函数的形式如下：

$$f(\boldsymbol{x})=\mathrm{sgn}\left(\sum_{i=1}^{n}\alpha_i^* y_i K(\boldsymbol{x}_i,\boldsymbol{x})+b^*\right) \tag{1}$$

其中，$\boldsymbol{x}_i$是支持向量；$\alpha_i^*\neq 0$为其对应的拉格朗日系数；$\boldsymbol{x}$为待分类的向量；n为支持向量的数量；b^*为偏置。从式(1)可以看出，判定一个未知类别的样本所需要的时间与支持向量的数目呈正比。

SVM快速分类算法的基本原则是通过寻找一个包含较少支持向量的约简向量集代替原有的支持向量集来提高分类速度，即：

$$f(\boldsymbol{x})=\mathrm{sgn}(\sum_{i=1}^{n}\alpha_i^* y_i K(\boldsymbol{x}_i,\boldsymbol{x})+b^*)=\mathrm{sgn}(\sum_{i=1}^{m}\alpha_i^* y_i K(\boldsymbol{x}_i,\boldsymbol{x})+b^*) \tag{2}$$

其中要求$m<<n$。

基于上述原则，本文提出了基于特征空间相似性分析的SVM快速分类算法，其基本思想是在特征空间中对样本的相似程度进行分析，并通过建立支持向量在特征空间相似性度量的最小支撑树对其按相似性最大进行分组，用每组中与其他支持向量平均相似性最大的向量作为“决定因子”，与其他支持向量平均相似性最小的向量作为“调整因子”，通过2个因子的线性组合对整组支持向量在高维空间中的加权和

基金项目：河北省科技计划基金资助项目“基于人体运动学的客流采集系统的研究”(09213507D)

作者简介：朱 方(1981－)，男，博士研究生，主研方向：模式识别，智能系统；顾军华，教授、博士生导师；杨欣伟，硕士研究生；杨瑞霞，教授、博士生导师

收稿日期：2010-03-20 **E-mail：**sky050607@sina.com

进行拟合，从而将问题转化成求关于“决定因子”系数和“调整因子”系数的二元二次函数的最小值问题，最终实现对支持向量的简化，提高分类速度，如式(3)所示：

$$\sum_{i=1}^{n}\alpha_i^* y_i\varphi(\boldsymbol{x}_i)=\sum_{i=1}^{m}\sum_{j=1}^{v_m}\alpha_{ij}^* y_{ij}\varphi(\boldsymbol{x}_{ij})\approx\sum_{i=1}^{m}(\beta_{i1}\varphi(\boldsymbol{x}_{i\max})+\beta_{i2}\varphi(\boldsymbol{x}_{i\min})) \tag{3}$$

其中规定支持向量数量为 n，被分成 m 组，第 i 组中的支持向量数量为 v_i，$\boldsymbol{x}_{i\max}$ 为第 i 组的决定因子，$\boldsymbol{x}_{i\min}$ 为第 i 组的调整因子。

将式(3)代入式(2)中，可以得到如下支持向量机分类决策函数：

$$f(\boldsymbol{x})=\operatorname{sgn}(\sum_{i=1}^{m}(\beta_{i1}K(\boldsymbol{x}_{i\max},\boldsymbol{x})+\beta_{i2}K(\boldsymbol{x}_{i\min},\boldsymbol{x}))+b^*) \tag{4}$$

2.1 相似性系数的选择

由于向量的夹角余弦值具有旋转和缩放的不变性，更能体现特征空间中支持向量的几何分布特征，因此本文选择支持向量在高维空间中像的夹角余弦值作为相似性度量标准：

$$C_{ij}=\cos(\varphi(\boldsymbol{x}_i),\varphi(\boldsymbol{x}_j))=\frac{\varphi(\boldsymbol{x}_i)\cdot\varphi(\boldsymbol{x}_j)}{\sqrt{(\varphi(\boldsymbol{x}_i)\cdot\varphi(\boldsymbol{x}_i))(\varphi(\boldsymbol{x}_j)\cdot\varphi(\boldsymbol{x}_j))}}=\frac{K(\boldsymbol{x}_i,\boldsymbol{x}_j)}{\sqrt{K(\boldsymbol{x}_i,\boldsymbol{x}_i)K(\boldsymbol{x}_j,\boldsymbol{x}_j)}} \tag{5}$$

当 C_{ij}=1 时，夹角为 0°，说明 2 个向量极其相似；C_{ij}=0 时，夹角为 90°，说明两向量正交，不具有相关性。

通过式(5)可以明显地看到，用原空间的核函数代替特征空间的内积运算可以把运算完全转换到原空间内，避免了对未知映射函数 φ 的限制和猜测。

由相似性系数构成的相似性矩阵 $\boldsymbol{C}$ 是本文算法的数据基础：

$$\boldsymbol{C}=\begin{bmatrix}C_{11}^2 & C_{12}^2 & \cdots & C_{1n}^2\\ C_{21}^2 & C_{22}^2 & \cdots & C_{2n}^2\\ \vdots & \vdots & & \vdots\\ C_{ij}^2 & C_{n2}^2 & \cdots & C_{nn}^2\end{bmatrix}C_{ij}^2$$

$\boldsymbol{C}$ 是一个对称矩阵。

2.2 基于最小支撑树的支持向量分组

以各支持向量在特征空间的像为顶点，1-C_{ij}^2 为权值，构造一个带权值的无环连通图 $G(V, E)$，其中：

V={ $\varphi(\boldsymbol{x}_i),\varphi(\boldsymbol{x}_j)\mid i\neq j$ }

E={1-$C_{ij}^2\mid i\neq j$ }

为了达到按特征空间相似性最大进行分组简化的目的，由该连通图生成一个最小支撑树 Tr，以确保每个顶点都找到与自己相似性最大的点来构成最小支撑树的一条边。

在最小支撑树 Tr 已知的情况下，通过定义最小支撑树的“长边”对顶点和边进行分组。

长边[7]定义如下：设 XY 是支撑树 Tr 的一条边，如果 XY 的边长明显大于以 X(或 Y)为起点的、k(k=1,2,⋯,n-1)步以内的边长平均数(不包括通过 XY 的边)，则称 XY 为最小支撑树 Tr 的一条长边。

根据长边的定义，在得到最小支撑树 Tr 中所有“长边”的基础上，分组规则就是将“长边”两端的子树分到不同的组。这样实现了对支持向量按相似性关系的自动分组，保证了子树规模较小，且子树内有边相连的节点间的相似相关性最大，为在子树中有效地找到能够代替各节点加权值之和的特定因子奠定了理论基础。

2.3 特定因子的选择及相关系数的确定

选择每组中的特定因子是有效实现本文算法的关键，因此，对特定因子做如下规定：(1)决定因子确保与该组内所有特征空间内的支持向量相似相关性近似最大。(2)调整因子确保具有更大的调节范围。

基于上述规定，选取与该组内所有特征空间内的支持向量的平均相似相关性近似最大的支持向量作为决定因子；与该组内所有特征空间内的支持向量的平均相似相关性近似最小的支持向量作为调整因子。

对于求解第 i 组数据的加权系数 β_{i1}、β_{i2}，可以通过如下的规划问题得到：

$$\begin{aligned}&\min f(\beta_{i1},\beta_{i2})=\\ &\min\left\|\sum_{j=1}^{vi}\alpha_{ij}^* y_{ij}\varphi(x_{ij})-(\beta_{i1}\varphi(\boldsymbol{x}_{i\max}+\beta_{i2}\varphi(\boldsymbol{x}_{i\min}))\right\|^2=\\ &\min(\beta_{i1}^2K(\boldsymbol{x}_{i\max},\boldsymbol{x}_{i\max})+\beta_{i2}^2K(\boldsymbol{x}_{i\min},\boldsymbol{x}_{i\min})+\\ &2\beta_{i1}\beta_{i2}K(\boldsymbol{x}_{i\max},\boldsymbol{x}_{i\min})-2\beta_{i1}\sum_{j=1}^{vi}K(\boldsymbol{x}_j,\boldsymbol{x}_{i\max})-\\ &2\beta_{i2}\sum_{j=1}^{vi}K(\boldsymbol{x}_j,\boldsymbol{x}_{i\min})+\sum_{j=1}^{vi}\sum_{k=1}^{vi}K(\boldsymbol{x}_j,\boldsymbol{x}_k))\end{aligned} \tag{6}$$

2.4 算法的具体实现

算法的实现是建立在支持向量的 Hessian 矩阵已知的前提下，具体步骤如下：

Step1 根据所有支持向量的 Hessian 矩阵，计算相似系数矩阵 $\boldsymbol{C}$。

Step2 以各支持向量为顶点、1-C_{ij}^2 为权值，构造一个带权值的无环连通图 G，并通过 G 生成一棵最小支撑树 Tr。

Step3 根据定义的“长边”规则，将 Tr 按 1 步长分成 m 棵子树，记录每棵子树各边的端点，同一子树中的所有端点构成一个支持向量的子集。

Step4 根据定义，寻找每个支持向量子集中的决定因子和调整因子。

Step5 通过计算二元二次函数 $f(\beta_{i1},\beta_{i2})$ 的最小值问题，得到每组中 2 个特定因子的系数。

Step6 用每组中特定因子的线性组合来代替整组支持向量的加权和，构造新的分类决策函数，如式(4)所示。

Step7 用新的近似分类决策函数对新测试样本进行分类识别，并得出分类结果。

2.5 时间复杂度分析

由式(3)可知，SVM 判断一个未知类别的样本所需要的时间与支持向量的数目呈正比。假设简化前后 SVM 的支持向量数分别为 n、$2m$，以核函数的计算次数来度量，则原始 SVM 预测一个样本的时间复杂度为 $O(n)$，简化后对应的时间复杂度为 $O(2m)$，由于 $2m<<n$，因此简化后的 SVM 具有更快的分类速度。

3 实验结果分析

本文用 VC++6.0 和 Matlab7.1 实现提出的快速分类算法，Libsvm 软件包被选作标准的 SVM 训练算法，并通过 Libsvm 网站提供的 2 类分类数据进行实验，以验证算法的有效性及可行性。实验在样本训练过程中采用 5 次交叉验证，实验结果均是重复 10 次实验取平均值。

由于现有 SVM 快速分类算法均具有应用局限性，如文献[3]的算法总体约简率不高，分类速度没有得到明显的提高；文献[6]的算法只适用于等方性核函数，不能得到广泛应用，因此本文只与具有普遍应用性的 Scholkopf 算法和

FCSVM 算法进行对比(最终结果是多次采用不同初始值重复简化过程所获得的结果中的最优值)，证明本文算法能够在保证分类精度的同时较大地提高 SVM 的分类速度。

本文算法与 FCSVM 算法的对比结果如表 1 所示，其中，N_{tr} 代表训练样本集规模；dim 代表样本维数；N_t 代表测试样本集规模；T 为分类时间；N 代表经训练得到的支持向量个数；P 代表对测试样本集进行分类识别的准确率。本文算法相对于原始支持向量机算法的各项指标缩减率如表 2 所示，其中，R 代表本文算法的支持向量缩减率：

$$R = \frac{N_{sv1} - N_{sv2}}{N_{sv1}}$$

R_t 代表快速分类算法的分类时间缩减率：

$$R_t = \frac{T_{tr1} - T_{tr2}}{T_{tr1}}$$

$Erro$ 代表本文算法的分类精度损失率；$ScErro$ 代表 Scholkopf 算法在与本文算法相同约简率情况下的分类精度损失。

表 1　本文算法与 FCSVM 算法对比实验结果

数据集	N_{tr}	dim	N_t	T/s			N			P/(%)		
				未经缩减	FCSVM	本文算法	未经缩减	FCSVM	本文算法	未经缩减	FCSVM	本文算法
svmguid1	5 089	4	2 000	0.218	0.184	0.094	608	453	292	91.680	91.000	90.900
svmguid3	83	21	447	0.063	0.054	0.016	301	291	100	73.378	68.900	69.821
Splicse	2 175	60	1 000	0.953	0.498	0.063	785	392	80	90.500	79.900	88.500
a1a	1 605	123	7 796	7.172	4.231	3.983	581	498	342	77.296	75.948	75.936
a3a	3 185	123	3 445	6.329	4.094	0.609	1 165	729	108	76.633	76.035	75.414

表 2　本文算法缩减率及错误率　(%)

数据集	R	R_t	$Erro$	$ScErro$
svmguid1	51.970	56.880	0.750	2.305
svmguid3	66.780	74.600	3.557	3.831
Splicse	89.810	93.390	2.000	5.400
a1a	41.140	44.460	1.360	1.590
a3a	90.730	90.380	1.219	11.500

通过实验结果可以看出，在具有相同约简率的情况下，本算法的分类精度损失率更低，如对于数据集 a3a，在缩减率为 90.73%时，本文算法的分类精度损失只有 1.219%，远远低于 Scholkopf 算法的精度损失 11.5%，并且 Scholkopf 算法在求解过程中需要预先确定约简率，在约简率确定的情况下，寻找每个约简向量仍要采用不同的初始值重复迭代过程多次，以避免陷入局部最小，代价高昂并且不一定能找到最优值。而本算法无需确定约简率，在寻找约简向量过程中只需求解二元二次函数的最小值问题，较 Scholkopf 算法有更高的效率。与 FCSVM 算法相比，本文算法有更高的约简效果，且 FCSVM 算法只有在待识别样本的先验信息较充分时，才有较好的缩减效果，因此，在实际应用中不具备实时分类的意义。可见，本文算法能够保证在分类精度损失较小的情况下具有较为理想的缩减率，实际应用性能较高。

实验结果还证明，本文算法相对于原始支持向量机，支持向量约简率平均能达到 68.086%，分类精度损失只有 1.777 %，且算法的实现简单，避免了低效的迭代过程，大大提高了支持向量机的分类速度，时间缩减率平均能够达到 71.942%，满足了实际应用中实时分类的要求。

4　结束语

本文提出的基于特征空间相似性分析的 SVM 快速分类算法可以更好地均衡支持向量的缩减率和分类精度损失，在满足实际应用要求的前提下，最终实现了支持向量的简化，提高了 SVM 的分类速度。经实验证明，本文算法思想简单，易于实现，能够在保证分类精度的同时对支持向量进行大幅缩减，有更高的分类速度，是有效且可行的。

参考文献

[1] Scholkopf B, Mika S, Burges C, et al. Input Space Versus Feature Space in Kernel-based Methods. IEEE Trans. on Neural Networks, 1999, 10(5): 1000-1017.

[2] Scholkopf B, Knirsch P, Smola A, et al. Fast Approximation of Support Vector Kernel Expansions, and an Interpretation of Clustering as Approximation in Feature Space[M]//Levi P, Schanz M, Ahlers R J. Mustererkennung. London, UK: Springer-Verlag, 1998: 124-132.

[3] Downs T, Gates K E, Masters A. Exact Simplification of Support Vector Solutions[J]. Journal of Machine Learning Research, 2001, 12(2): 293-297.

[4] 刘向东, 陈兆乾. 一种快速支持向量机分类算法研究[J]. 计算机研究与发展, 2004, 41(8): 1327-1332.

[5] 秦玉平, 王秀坤. 一种改进的快速支持向量机分类算法研究[J]. 大连理工大学学报, 2007, 47(2): 291-294.

[6] 曾志强, 高　济. 基于向量集约简的精简支持向量机[J]. 软件学报, 2007, 18(11): 2719-2727.

[7] 余锦华. 多元统计分析与应用[M]. 广州: 中山大学出版社, 2005: 159-170.

编辑　张　帆