文章编号:1674-2869(2010)07-0089-05

# 基于SVM的多类分类算法改进

王 忠[1],王春丽[1],刘 莉[2]

(1.武汉工程大学计算机科学与技术学院,湖北 武汉 430074;
2.湖北水利水电职业技术学院,湖北 武汉 430070)

摘 要:在各种基于支持向量机的多类分类算法中,基于二叉树的多类支持向量机分类算法训练和分类速度相对较快,且解决了不可分问题,是一种很好的方法.本文系统研究和分析了基于二叉树的多类支持向量机分类算法,并在此基础上对其作出了改进,即当测试文本集规模较大时,对其先聚类再分类.改进的目的是,使测试文本不必总是从二叉树的根结点开始进行判断,而是有指导的代入分类函数中计算.在测试文本集规模较大,分类函数个数较多时,可以很大程度上增加分类效率,并加大了文本正确分类的概率.

关键词:支持向量机;分类算法;统计学习;二叉树

中图分类号:TP312 文献标识码:A doi:10.3969/j.issn.1674-2869.2010.07.023

## 0 引 言

支持向量机(Support Vector Machine,简称SVM)[1-3]算法的研究起源于对数据分类问题的处理,是统计学习理论的一种实现方法,它建立在样本数量有限的基础之上,能在现有训练文本包含的信息下得到最佳分类效果.同时,SVM源于统计学习理论中的VC维理论和结构风险最小化(Structure Risk Minimization,简称SRM)原理[4],有效地解决了其它机器学习算法中的过学习问题,即SVM在数量有限的训练样本上,由训练误差最小化可以确保测试误差最小化.然而,标准的SVM算法只能解决两类分类问题,虽然分类精度高,但通常不能满足现实中多类分类问题的需要.将SVM算法推广到多类分类问题中,具有重要的意义,引起了人们的广泛关注.

多类分类问题可用数学语言描述为:给定训练库

$$TR=\{(x_i,y_i)\mid x_i\in R^n,y_i\in Y,i=1,2,\cdots,n\} \tag{1}$$

和测试文本集合$X\in R^n$.需要求解分类函数$f(x)$,使得

$$f(x):\ X\rightarrow Y \tag{2}$$

$x_i$为文本向量,$y_i$为类别标记,$Y$类别集合,$|Y|>2$.在目前的研究中,常用的SVM多类分类方法有一对多、一对一、决策有向无环图、二叉树方法和一次性求解函数参数的方法[5-7]等.其中,基于二叉树的多类SVM分类算法在训练时不需要总在整个训练库上进行,其训练子库规模逐步减小,而在测试时也不必要总是遍历整个二叉树[8],所以其训练和测试速度都比较快,并且解决了不可分区域问题.目前为止,已有很多对它的改进和应用,但主要集中在对训练库的处理和二叉树的结构的改进上,取得了不少成果.本文在深入研究基于二叉树的多类SVM分类算法的基础上,从分类阶段出发,对它提出了改进.

## 1 基于二叉树的多类SVM分类算法分析

首先给出标准的基于二叉树的多类SVM分类算法描述如下[9]:

(1)计算训练库$TR$中类别数$k$.

(2)若$k>2$转至(3),若$k\leqslant 2$转至(6).

(3)将训练库随机分成两个子集$A$和$B$,以$A$(或$B$)为正类,$B$(或$A$)为负类构造分类函数$f(x)$.

(4)以$f(x)$为根结点构造二叉树.

(5)对子集$A$和$B$重复步骤(1)、(2)、(3),并将以$A$(或$B$)为训练库生成的分类函数为左子树,以$B$(或$A$)为训练库生成的分类函数为右子树,构造分类函数.

(6)若$k=2$转至(7),若$k=1$转至(8).

(7)以其中一类为正样本,另一类为负样本构造分类函数$f(x)$,并以$f(x)$为父结点,正(或负)

收稿日期:2010-04-01

作者简介:王忠(1968-),男,湖北江陵人,教授,硕士.研究方向:数字水印.

样本编号为左子树，负（或正）样本编号为右子树构造子二叉树，将此子二叉树加入到相应的内部结点中作为孩子结点.

（8）以该样本编号为叶子结点，加入到相应的内部结点中作为其孩子结点.

（9）将测试数据 $x_e$ 输入到已建好的二叉树根结点中.

（10）若 $f(x_e)\geq 0$ 转至（11），若 $f(x_e)<0$ 转至（12）.

（11）进入左子树，若该结点为叶子结点输出 $x_e$ 类别，否则转至（10）.

（12）进入右子树，若该结点为叶子结点输出 $x_e$ 类别，否则转至（10）.

算法1　标准的基于二叉树的多类SVM分类算法

对于有 $k$ 个类别的问题而言，基于二叉树的多类SVM分类算法需要构造 $k-1$ 个分类函数，其中第 $i$ 个分类函数以第 $i$ 类为正训练样本，以 $i+1$ 到 $k$ 类为负训练样本，$1\leq i\leq k$. 然后将 $k-1$ 个两类分类函数作为内部结点组合成二叉树形式，并以 $k$ 个类别标记为叶子结点. 测试时，从根结点开始计算分类函数，根据值的正负决定下一步的走向，如此下去，直到到达某一叶结点为止，此叶结点所代表的类别就是测试样本的类别. 在此过程中，可能使用到的分类函数数目介于1和二叉树的深度之间.

可以看出，基于二叉树的多类SVM分类算法具有层次结构，每个层次的分类函数的级别和重要性不同，在构造二叉树的过程中可以考虑各个类别的先验知识. 由于在训练时它不需要总在整个训练库上进行，其训练子库规模逐步减小，而在测试时也不必要总是遍历整个二叉树，所以其训练和测试速度都比较快，并且解决了不可分区域问题[5]. 相对于其它基于SVM的多类分类算法而言，基于二叉树的多类SVM分类算法主要有如下特点：

（1）针对 $k$ 类分类问题，只需构造 $k-1$ 分类函数，数目相对较少，训练速度较快.

（2）训练分类函数时，不需要总是在整个训练库上进行，二次规划问题的规模随着训练过程的进行逐渐减小.

（3）测试时不需要总是遍历整个二叉树，而是沿着从根结点到文本类别的方向前进，可能使用到的SVM分类函数数目介于1到二叉树的深度之间，分类速度相对较快.

（4）解决了其它方法中的不可分问题.

虽然基于二叉树的多类SVM分类算法具有良好的分类性能，然而，在测试过程中，大量的测试文本都必须从二叉树的根结点开始，逐步判断，带有一定的盲目性. 例如，如果将二叉树内部结点按照前序遍历的顺序，依次编号为 $i$ 到 $k-1$，这样这 $k-1$ 个分类函数对应的类别也被从1标记到 $k-1$. 当测试文本 $x$ 属于二叉树中第 $i$ 类时，在某些情况下测试文本必须依次计算 $f_0(x)$，$f_1(x)$，……，直到 $f_{i-1}(x)$ 为止，$0\leq i\leq k-2$. 在此过程中，前面的 $i$ 次计算都不能确定 $x$ 的类别，从某种意义上说是无用的计算. 并且在实际情况中，属于二叉树中第一个类别的测试文本总是有限的，它们在测试文本集合中仅占一定的比重，甚至仅仅占很小的比重，那么其它不属于该类别的文本都从第一个类别开始进行计算，是没有必要的. 对于二叉树中其它类别，也是如此. 可以发现，如果测试文本数量越大，训练库中类别数越多，则由于盲目计算所浪费的资源越多. 将上述问题总结如下：

（1）所有的测试文本都必须从二叉树的根结点开始，逐步判断，带有一定的盲目性. 这种盲目性会造成很多无用的计算，浪费大量资源.

（2）测试文本数量越多时，无用的计算越多.

（3）训练库中类别数越多时，分类函数越多，无用的计算越多.

因此，有必要寻找一种能减少这种盲目性的方法，提高基于二叉树的多类SVM分类算法的分类效率. 本文提出，当测试文本数量庞大，训练库类别较多时，可以先对测试文本进行聚类，然后分类，使得分类带有一定的指导性. 具体步骤将在下文中给出.

## 2　基于二叉树的多类SVM分类算法的改进

在对基于二叉树的多类SVM分类算法的改进方法中，本文使用到了文本聚类算法[9]，其基本思想是指把一组对象集合根据其特征归成若干类别，其目的是将一个大的集合分为若干小类别，使得属于同一类别的对象之间相似程度最大，而不同类别之间的相似程度最小. 在常用的聚类方法中，平面划分方法简单易行，且具有良好的性能. 它的基本思想是：先从数据集中任意选取若干数据作为聚簇中心，然后依据一定规则将剩余数据归入到与各聚簇中心距离最近的聚簇中去. 在平面划分方法中，结果依赖于聚簇中心的选择，一般是先对样本进行归类，求出各个类的均值向量（中心点），再将各个样本归到与其最近的均值向量的类别中，如此反复. $k$-means是一种比较流行的启

发式平面划分聚类方法，它的每个聚簇中心用该聚簇中对象的平均值来表示。$k$-means 算法的基本步骤是：在数据集合中任意选择 $k$ 个数据，分别代表 $k$ 个聚簇的平均值，将剩余的数据根据它们与各个聚簇中心的距离归入到最近的聚簇中，然后重新计算每个聚簇的平均值。重复该过程，直到准则函数收敛为止。

本文在对测试文本进行聚类时，使用训练库中的文本作为初始聚簇中心。这样做的目的是出于以下三点原因：

(1)本文对测试文本进行聚类，是为了根据各个聚簇中心信息，将聚簇中文本输入到与之最相似的训练文本类别所对应的分类函数中进行计算。这样，测试文本能被最先代入到它最可能属于的类别所对应的分类函数中，能减少测试过程中文本类别判断的盲目性。

(2)当训练库中类别数为 $k$ 时，以训练库中的文本作为聚簇中心，能将测试文本聚集成与训练库对应的 k 个类别，有利于测试文本的分类。

(3)以训练库中的文本作为聚簇中心，能有效的减少聚类算法的迭代次数。

根据上文聚类的思想，可以先对测试文本集进行聚类，形成与训练库对应的聚簇，然后在聚簇中选择聚簇中心，代入二叉树分类算法中判断其类别，再将该聚簇中的其它文本直接代入聚簇中心所在的类别的分类函数中，进行类别判断(如图 1 所示)。由于在聚类生成的类别中，同一类别的文本之间相似程度很大，且其聚簇中心特性最能代表类别的特性，所以能用聚簇中心信息来对同聚簇中的数据进行判断。

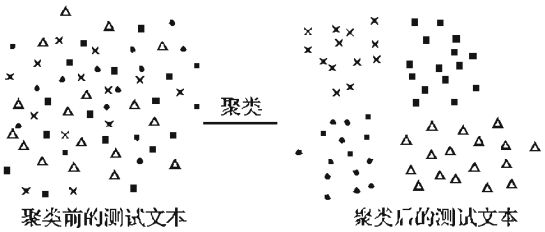


图 1　测试文本聚类示意图

Fig. 1　Schematic diahram of testing text clustering

使用训练库中的文本作为聚类的聚簇中心，则在文本聚类后形成的各个聚簇中，其聚簇中心的类别是已知的。属于该聚簇的文本 $x$ 直接代入聚簇中心所在类别对应的分类函数 $f_t(x)$ 中进行计算，根据聚簇的特性可知，此时函数值为正，即文本 $x$ 属于类别 $t$ 的概率很大。可以认为，经过第一次判断后，大部分的测试文本都可归入到正确的类别中。当然可能存在少数的文本，第一次不能确定其类别。对于这部分文本而言可能存在两种情况：

(1)聚类时没有被聚集到正确的类别中。

(2)它是不属于聚簇中心所属类别的文本，即聚类时发生误差。

本文的处理办法是，让它们从根结点开始重新遍历二叉树，使用在标准算法中的方法来判定其类别。由于基于二叉树的多类 SVM 分类算法的特性，测试文本不存在不可分情形，即分类后所有测试文本都可以被归入到相应的类别中。

本文对基于二叉树的多类 SVM 分类算法的改进，是对分类阶段的改进。在改进后的算法中，本文提出的改进思想将在算法的分类阶段体现出来。改进后的基于二叉树的多类 SVM 分类算法描述如下(设训练库中类别数为 $k$)：

(1)计算训练库 $TR$ 中类别数 $k$；

(2)若 $k>2$ 转至(3)，若 $k\leqslant 2$ 转至(6)；

(3)将训练库随机分成两个子集 $A$ 和 $B$，以 $A$(或 $B$)为正类，$B$(或 $A$)为负类构造分类函数 $f(x)$；

(4)以 $f(x)$ 为根结点构造二叉树；

(5)对子集 $A$ 和 $B$ 重复步骤(1)、(2)、(3)，并将以 $A$(或 $B$)为训练库生成的分类函数为左子树，以 $B$(或 $A$)为训练库生成的分类函数为右子树，构造分类函数。

(6)若 $k=2$ 转至(7)，若 $k=1$ 转至(8)。

(7)以其中一类为正样本，另一类为负样本构造分类函数 $f(x)$，并以 $f(x)$ 父结点，正(或负)样本编号为左子树，负(或正)样本编号为右子树构造子二叉树，将此子二叉树加入到相应的内部结点中作为孩子结点。

(8)以该样本编号为叶子结点，加入到相应的内部结点中作为其子结点。

(9)重复上述过程，直到训练库为空，生成二叉树 SVM $T$。

(10)在已知的 $k$ 个类中分别选取一个文本 $x_i$，$1\leqslant i\leqslant k$，以 $x_i$ 为聚类中心对测试文本聚类，在测试文本中得到 $k$ 个聚簇 $c_i$，$1\leqslant i\leqslant k$。

(11)若 $c_t$ 中包含训练文本 $x_i$，则将 $c_t$ 中所有文本代入 $x_i$ 所在类别对应的分类函数 $f_i(x)$ 中计算。

(12)若 $f_i(c_{ij})\geqslant 0$，$j=|c_i|$ 则文本 $c_{ij}$ 属于类 $i$。

(13)否则，让文本 $c_{ij}$ 从根节开始遍历二叉树，直到确定其类别为止。

算法 2　改进后的基于二叉树的多类 SVM 分类算法描述

在此算法中，针对 $k$ 类分类问题，需要构造 $k-1$ 个分类函数. 与标准算法一样，改进后的算法也分为训练(1 到 9)和测试(10 到 13)两个阶段，其中分类器的训练与标准算法一致，本文对算法的改进体现在测试阶段. 在测试阶段，改进后的算法先对测试文本集进行聚类，然后根据聚簇的聚簇中心的类别信息来对该聚簇中其它文本进行类别判断.

## 3 改进后算法性能分析

下面本文将使用具体数据对改进前后的算法效率进行分析. 为此，收集 10 个类别的测试文本各 1 000 篇. 并在由分类函数构成的单边二叉树和完全二叉树上进行分析.

$k$-means 聚类算法的平均准确率可以达到 75% 以上[10-13]. 在此，不失一般性，假设聚类后每个类别的测试文本有 75% 被准确聚类. 在单边二叉树的情形下，原始算法完成全部测试文本的分类需要计算 $(1+2+\cdots+9)\times 1\,000$，即 45 000 次分类函数. 而使用改进后的算法时，至少有 75% 的测试文本能够在第一次计算后确定其类别，而只有剩下的 25% 需要重新计算. 则总的计算次数为：$10\,000+1\,000\times 25\%\times(1+2+\cdots+9)=21\,250$ 次.

在完全二叉树的情形下，原始算法完成全部测试文本的分类需要计算 $4\,000\times 4+6\,000\times 3$ 次，即 34 000 次分类函数. 而使用改进后的算法时，总次数为 $10\,000+1\,000\times+1\,500\times=18\,500$ 次.

现将两种情况下的分类函数计算次数统计如表 1 所示.

表 1 分类函数计算次数统计

Tabel 1 Function calculates the number of statisticoul dassificatiion

| 分类函数计算次数 | 二叉树形态 | |
|---|---|---|
| | 单边二叉树 | 完全二叉树 |
| 原始算法 | 45 000 | 34 000 |
| 改进后的算法 | 21 250 | 18 500 |

由表 1 可知，改进后的基于二叉树的多类 SVM 分类算法，在两种情况下都可以使分类函数计算次数几乎减少一半，从很大程度上提高了算法的分类效率. 测试时分类函数的计算次数与二叉树的深度有关，由数据结构中相关知识可知，在结点数一定的情况下，单边二叉数具有最大的深度，而完全二叉树具有最小的深度，因此单边二叉树和完全二叉树具有代表性.

在上面的分析中，仅选取了 10 个类别，每个类别也只有 1 000 篇文本. 当多类分类问题的类别数更多，每个类别测试文本数量更大时，改进后的算法比原算法分类效率更高. 同时，测试文本聚类后，同一聚簇的测试文本之间具有很强的相似性，能在一定程度上指导二叉树分类器的分类，提高分类精度. 现在将改进后算法的特点总结如下：

(1)本文关于基于二叉树的多类 SVM 算法的改进是在测试阶段，因此改进后的算法在分类函数的训练阶段保持了原算法的特性，具有较高的训练效率.

(2)改进后的算法在测试阶段对测试文本先聚类再分类，然后使用聚簇中心信息来指导文本分类，使得测试文本的第一次类别判断是有目的性的，增大了快速将测试文本归类的概率，有效的减少了计算分类函数的次数.

(3)多类分类问题类别数越多，减少的分类函数计算次数越多.

(4)测试文本数量越多，减少的分类函数计算次数越多.

(5)聚类算法的准确性越大，减少的分类函数计算次数越多. 因为高准确性的聚类算法将属于同一类别的测试文本都聚集在一个聚簇中，使得测试文本被快速分类的概率增大.

(6)改进后的算法能在一定程度上指导二叉树分类器的分类，提高分类精度.

## 4 结 语

以上在系统研究了基于 SVM 的多类分类算法的基础上，深入地描述了基于二叉树的多类 SVM 分类算法. 并针对其不足之处提出了改进，即当测试文本集规模较大，类别数较多时，对其先聚类，再分类，增大分类效率，提高分类精度. 作为改进的准备知识，本章对聚类算法作了简要分析. 最后给出了改进后的算法描述，并将其与标准的算法相比较，分析了改进后算法的性能.

参考文献：


[1] Vapnik V. The Nature of Statistical Learning Theory [M]. New York: Springer-Verlag, 2000.

[2] 方辉，王倩. 支持向量机的算法研究[J]. 长春师范学院学报：自然科学版，2007，26(3)：90－91.

[3] 付香英，王春丽. 非线性可分文本的 SVM 算法研究及改进[J]. 九江学院学报，2008，(3)：69－61.

[4] Shawe-Taylor J, Bartlet P L, Williamson R C. Structural risk minimization over data dependent hierarchics. IEEE Transactions on Information Theory, 1998, 44(5): 1926－1940.

[5] Hsu Chih-Wei, Lin Chih-Jen. A comparison of methods

for multi-class support vector machines [J]. IEEE Transactions on Neural Networks, 2002, 13(2): 415 - 425.

[6] 黄琼英. 支持向量机多类分类算法的研究及应用[D]. 河北: 河北工业大学, 2005.

[7] Kuncheva L. Combining classifiers by clustering, selection and decision templates[D]. Technical report: University of Wales, 2000.

[8] 杜圣东. 基于多类支持向量机的文本分类研究[D]. 重庆: 重庆大学, 2007.

[9] Jiawei Han, Michcline amber. 数据挖掘概念与技术[M]. 范明, 孟晓峰, 译. 北京: 机械工业出版社, 2001.

[10] 赵毓高. 按聚类算法及其应用研究[D]. 成都: 西华大学, 2007.

[11] 胡学军, 腾达, 胡林文. 基于MATLAB的时滞对擦控制算法仿真分析[J]. 武汉工程大学学报, 2010, 32(3): 92 - 95.

[12] 陈伟亚, 徐佳彬, 李伟波. 基于技术线路图的循环经济发展规划技术研究[J]. 武汉工程大学学报, 2010, 32(5): 53 - 56.

[13] 崔士杰, 汪建华. 基于MATLAB的单相全控整流电路功率因数测试[J]. 武汉工程大学学报, 2010, 32(1): 90 - 92.

# Improvement on bintree multi-class categorization algorithm based on SVM

*WANG Zhong*[1], *WANG Chun-li*[1], *LIU-li*[2]

(1. Wuhan Institute of Technology, Wuhan 430074, China;
2. Hubei Water Resources Technical College, Wuhan 430070, China)

**Abstract**: It's a hotspot to research on support vector machine that extends from two-class issues to multi-class. Among all kinds of methods, bintree multi-class text categorization algorithm based on support vector machine is more effective in training and sorting than others, and it works out the impartibility problem. So it is a good method. The dissertation systematically researches and analyses bintree multi-class text categorization algorithm based on support vector machine, and then has some improvement on it. That is, we assembles firstly, and then sorts them when the size of testing texts is too large. The aim of this improvement is to make the testing text be computed more aimable, but does not begin from the base crunode of bintree at all time . The improvement can enhance the effect of text categorization and make it move accural when the size of testing texts is too large and the quantity of sorted function is too much .

**Key words**: support vector machine; categorization algorithm; statistical learning theory; quadratic programming

本文编辑: 陈小平