基于SMO的多层次文本分类法研究

何建兵[1] 何 清[2] 史忠植[2]
[1](中国科学院研究生院软件学院,北京 100049)
[2](中科院计算技术研究所智能信息处理重点实验室,北京 100080)
E-mail:hjb_rlh@sohu.com

摘 要 在以往的自动文本分类研究中,大多比较流行的分类技术都是在一个层次上将文本分成几个类别。但随着信息检索的量越来越大,文本的种类将越来越多,仅仅通过一层对海量信息进行组织分类越来越不适合海量信息的检索工作,这种平坦式的分类组织难以进一步提高信息检索的速度。论文将SMO分类算法结合到文本分类研究中,通过构建多层支持向量机文本分类树,实现了基于SMO的多层次文本分类系统。

关键词 文本分类 多层次文本分类 支持向量机 SMO算法 多层支持向量机

文章编号 1002-8331-(2006)13-0152-03 **文献标识码** A **中图分类号** TP311

Research of Multi-layer Text Categorization Method Based on SMO

He Jianbing[1] He Qing[2] Shi Zhongzhi[2]
[1](College of Software Engineering, Graduate School of The Chinese Academy of Sciences, Beijing 100049)
[2](The Key Laboratory of Intelligent Information Processing, Institute of Computing Technology, Chinese Academy of Sciences, Beijing 100080)

Abstract: In former automatic text categorization research, most of the prevalent classification technologies divided text into several classes in one lever. However, with the increase of quantity of information retrieval, this flat kind organization classification is more and more unsuitable to the information retrieval task with vast information, and it hampers the improvement of the information retrieval speed. This paper tries to adopt SMO algorithm to establish a multi-layer SVM text categorization system.

Keywords: text categorization, multi-layer text categorization, SVM, SMO algorithm, multi-layer SVM

1 引言

在信息处理的研究中,主要是从信息检索和文本自动分类这两个方面来进行的。信息检索最基本的任务是根据用户的需求检索出相关的文件,而文本自动分类则解决信息的有效组织问题。一般来说,信息检索是建立在对海量信息文件进行分门别类规整的基础之上,依照信息文本之间的相互关系,将信息文本建成多层次结构化的分类体系更符合信息检索原理,可以进一步提高文件检索的速度。例如,DMOZ和YAHOO等网页搜索网站就是按照层次化结构来组织网页搜索类别的,通过层次化结构选择来指定检索类别,缩小检索范围,简化信息查找和分类任务,从而提高信息检索的速度。

另一方面,多层次结构化文本分类有利于提高文本分类的准确率。在一些分类不是很明显的类别之间,总存在一些文本处在类与类的交界处,这些文本很容易被错误地分到别的类别中去。训练一个单层次结构的分类器时,一开始就试图区分那些差别不大的类别,选择训练样本的范围太大,能够代表类别之间细小差别的训练样本可能会被其他样本所"淹没",分类器难免错分那些处于类与类交界处的文本。多层次结构可以先将差别不大的小类组成一个大类,而大类之间区分得很开,准确率非常高。将大类分开之后,再在一个大类中区分差别不大的小类,其准确率比在整个范围内区分它们要高。原因在于,在一个大类的范围内选择训练样本进行训练时,能够选择到用来精确区分小类的更具有代表性的训练样本,训练样本集中没有其他大类的训练样本,使分类器的训练抛开了属于其他大类的训练样本的信息干扰。

本文第二节介绍了SMO算法,第三节描述了构建多层次文本分类树方法,第四节给出了单层支持向量机和多层支持向量机的对比实验结果,实验结果证实了上述观点。

2 序列最小优化(SMO)分类算法

支持向量机(SVM)分类算法思想是从训练样本中寻找能够确定一个最优超平面 $w\cdot x+b=0$ 的支持向量,该超平面试图将空间中的点最大间隔地分成两类 $y_i(w\cdot x_i+b)\geq 1$,尽量提高其泛化能力。它最早由Vapnik等学者提出[1],通过Vapnik结构风险最小化原则[2]和核函数方法,很好地解决了学习机的泛化能力和复杂性问题,在机器学习领域引起了极大的关注。近年来,在支持向量机研究方面,相继开发出了很多SVM快速训练算法,例如Joachims的 SVM^{Light}[3],John Platt的SMO[4]等算法,使得SVM在文本分类领域取得了很大的成功。Joachims曾使用

基金项目:国家自然科学基金资助项目(编号:60435010);国家863高技术研究发展计划资助项目(编号:2003AA115220);中澳科技合作特别基金项目;北京市自然科学基金资助项目(编号:4052025)

SVM^{light}进行文本分类实验，结果表现出比标准方法如简单贝叶斯、Rocchio、决策树算法 C4.5 和 k 近邻算法更好的性能。

SVM 的分类规则函数表示为：

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

其中：x_i 是训练样本，y_i 是样本类别，α_i 是拉格朗日乘子。α_i 通过优化如下的目标函数来求解：

$$\begin{aligned}&\underset{\alpha}{Maximise}\ L_D=\sum_{i=1}^{\ell}\alpha_i-\frac{1}{2}\sum_{i,j=1}^{\ell}\alpha_i y_i\alpha_j y_j K(x_i,x_j)\\&Subjectto\sum_{i=1}^{\ell}\alpha_i y_i=0,0\leqslant\alpha_i\leqslant C(i=1,\cdots,\ell)\end{aligned} \tag{2}$$

目标函数的最优解必须满足 Karush-Kuhn-Tucher[5]互补条件，根据该条件，仅仅最靠近超平面的点对应的 α_i 非 0。而在超平面的权重 w 表达式中，只有 α_i 非 0 的点才包括在内，因此这些点被称为支持向量。通过优化目标函数，寻找支持向量，从而训练得到一个 SVM 的分类规则。

SMO 优化(2)时使用了块与分解技术，并将分解算法思想推向极致，每次迭代仅优化两个点的最小子集，其威力在于两个数据点的优化问题可以获得解析解，从而不需要将二次规划优化算法作为算法一部分。尽管需要更多的迭代才收敛，但每个迭代需要很少的操作，因此算法在整体速度上有数量级的提高。不失一般性，假设优化 α_1,α_2，其他 α_i 固定，由线性约束条件 $\sum_{i=1}^{\ell}\alpha_i y_i=0$ 可知：

$$a_1^{old}+sa_2^{old}=a_1^{new}+sa_2^{new}=r \tag{3}$$

其中 $s=y_1y_2$，r 为常数，a_1^{old}，a_2^{old} 为 α_1，α_2 变化前的值，a_1^{new}，a_2^{new} 为变化后的值。将(3)代入(2)并优化目标函数可以得到：

$$a_2^{new}=a_2^{old}+\frac{y_2(E_2-E_1)}{k} \tag{4}$$

其中 $k=2K(x_1,x_2)-K(x_1,x_1)-K(x_2,x_2)$，$E_1=f(x_1)-y_1$，$E_2=f(x_2)-y_2$。为使 a_2^{new} 满足 $0\leqslant\alpha_i\leqslant C,i=1,\cdots,\ell$ 和条件(3)，必须对 a_2^{new} 进行修剪，将 a_2^{new} 的解限制在更为严格的取值范围内。

当 $s=-1$ 时，a_2^{new} 的取值范围为：

$$\max(0,a_2^{old}-a_1^{old})\leqslant a_2^{new}\leqslant\min(C,C-a_1^{old}+a_1^{old}) \tag{5}$$

当 $s=1$ 时，a_2^{new} 的取值范围为：

$$\max(0,a_1^{old}+a_2^{old}-C)\leqslant a_2^{new}\leqslant\min(C,a_1^{old}+a_2^{old}) \tag{6}$$

设 a_2^{new} 取值范围的最小值为 L、最大值为 H，通过(5)或(6)可以计算 L 和 H 的值。因此，a_2^{new} 修剪后的值为：

$$a_2^{new}=\begin{cases}L & a_2^{new}<L\\ a_1^{new}+\dfrac{y_2(E_2-E_1)}{k} & L\leqslant a_2^{new}\leqslant H\\ H & a_2^{new}>H\end{cases} \tag{7}$$

求得 a_2^{new} 的值后，由(3)求解 a_1^{new}。

SMO 每次迭代时，从训练集中启发式地选择最可能违反 KKT 条件的两点进行优化，并通过监视满足 KKT 条件的允许偏差来判断算法是否可以终止，如果偏差小于某个值(如 0.01)则可以停止计算。

3 多层次文本分类

多层次文本分类要比单层次文本分类复杂得多，它的难点不是说多训练出几个分类器就算解决问题了，而在于训练出的分类器是有上下层次关系和左右分支关系的分类器体系结构，这种体系结构反映出了文本类别之间的内在联系。要构造一个基于 SMO 的多层次文本分类系统，关键在于构造一个多层的支持向量机体系结构。

3.1 多层支持向量机结构

多层 SVM 是一个树形结构，每个节点是一个基于 SVM 的多分类器。对于一个有 m 类的多分类器，它由 m 个 SVM 构成，每个 SVM 负责区分本类和非本类数据。图 1 展示了一个多层 SVM 结构图。

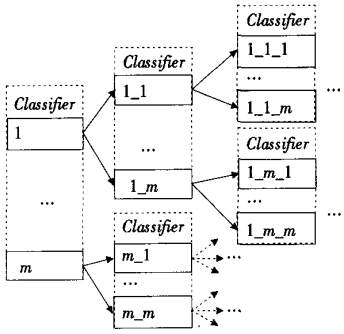

图 1 多层支持向量机结构图

多层 SVM 的层次关系通过各分类器所包含的 SVM 的层次结构逻辑编号来表示。如图 1 所示，第一层只有一个分类器，有 m 种分类，由 m 个 SVM 构成，其逻辑编号分别为 1,2,…，m；相应的，第二层有 m 个分类器，每个分类器分别对应上一层的类别，继续对上一层的分类结果进行分类。例如，第二层第一个分类器，对应上一层第一个类别，分类器又有 m 种分类，由 m 个 SVM 构成，其逻辑编号分别为 1_1,1_2,……,1_m。依次类推，多层 SVM 的结构关系通过这样的逻辑编号表示出来了。

3.2 组织多层支持向量机训练文件集

多层 SVM 机分类树中的叶节点数由文本的最终分类数确定，而初始的训练样本的类别是通过人工方式按照文本分类的最终分类来划分。假设文本最终分成 m 类，则初始的训练样本集分别为 $S_1,S_2,\cdots,S_m$，按照 1 至 m 的自然数顺序编码。其中，S_i 的正样本属于第 i 类，S_i 的负样本属于其他类（或者没有负样本）。因此，初始的训练文件集的个数等于分类树中的叶子节点个数。在为分类树中的每个 SVM 生成训练样本集时，应按照分类树的层次结构信息对初始训练文件进行重新组合，生成分类树中的所有 SVM 对应的训练文件。按照预先定义的分类树结构，通过交互方式可以构建图 2 所示的训练文件信息树。

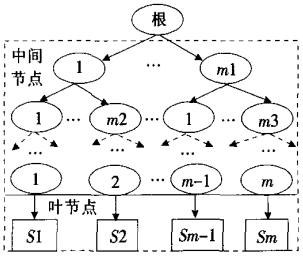

图 2 训练文件信息树示意图

为了能够为分类树中的SVM正确生成训练样本集，在构建训练文件信息树时，系统对人工输入作了如下限制：(1)叶节点必须是存在的初始训练样本集文件；(2) 根节点下的子节点必须是中间节点，不能直接是叶节点，否则叶节点没有分层信息；(3)初始训练样本不能是二义的，不能有交叉分类，只属于一个最终的类别。所以叶节点不能有其他兄弟；(4)中间节点必须有其他兄弟，不能是父节点的唯一的孩子。

训练文件信息树中的中间节点对应分类树中的SVM，采用深度优先搜索算法遍历整个训练文件信息树，确定每个叶节点(初始训练文件)的搜索路径($n_1\_n_2\_n_3\cdots\_S_i$)。解析所有叶节点(S_i)的搜索路径($n_1\_n_2\_n_3\cdots\_S_i$)，为搜索路径上的每个中间节点(n_1,$n_1\_n_2$,$n_1\_n_2\_n_3$,$\cdots$)添加叶节点(S_i)的正样本，作为中间节点(n_1,$n_1\_n_2$,$n_1\_n_2\_n_3$,$\cdots$)的正样本。然后为中间节点添加负样本，其负样本来自与中间节点有相同父节点的其他兄弟节点的正样本。从而组合出分类树中每个SVM的训练样本，组成分类树中所有SVM的训练文件集。系统按如下规则命名训练文件：文件名为相应的中间节点在信息树中的搜索路径名(n_1,$n_1\_n_2$,$n_1\_n_2\_n_3$,$\cdots$)；扩展名为mlt。

3.3 多层支持向量机测试

在测试过程中，需要测试分类树中的每个分类器的分类正确率。如3.2所述，多层SVM分类树中除根节点外的节点(SVM)和训练文件信息树的中间节点是一一对应的。在完成多层SVM分类树的训练后，生成了各SVM的模型文件和训练文件信息树叶节点搜索路径信息。在文本分类阶段，根据叶节点搜索路径信息可以构造一颗分类树，分类树节点的逻辑编号和模型文件的逻辑编号是一致的。分类树中的每个分类器由多个SVM组成，测试一个分类器需要遍历一个分支所有节点。因此，多层SVM测试算法采用了递归算法和树广度优先搜索算法。其算法步骤如下：

步骤1 初始化测试活动集，包含所有测试向量。

步骤2 从分类树的根节点开始，获得第一个分支=>Branch，递归调用树广度优先搜索算法，转步骤3。搜索函数返回，算法结束。

步骤3 遍历分支Branch中的所有节点，根据节点的逻辑编号获得对应的模型文件。计算测试活动集中每个向量到超平面的距离，计算公式为：

$$dist=\sum_{i\in SV}\alpha_i y_i K(\boldsymbol{x}_i,\boldsymbol{x}_j)+b \tag{8}$$

其中，$\boldsymbol{x}_i$是支持向量，$\boldsymbol{SV}$是支持向量集，α_i是支持向量的拉格朗日乘子，y_i是支持向量类标(±1)，b是偏置，K是核函数。SV，α_i，y_i，b以及核函数类型都是从模型文件中读取的，也就是说，模型文件确定了这个超平面。记录最大距离的节点逻辑编号。

步骤4 对测试活动集中的测试向量进行分类，将测试向量分到距离超平面最远的那一类。

步骤5 (1)统计分类器分类正确率，等于(正确分类数/测试活动集中向量数)*100%。假设测试向量的分类编号为：$n_1\_n_2\_n_3$，测试向量的已知类别的搜索路径信息为$m_1_m_2_m_3\cdots_S_i$)，其中i就是测试向量的已知类标。如果$n_1_n_2_n_3$和$m_1_m_2_m_3$匹配上，表明测试向量的分类是正确的，否则分类是错误的。(2)取Branch的第一个节点=>Node。

步骤6 (1)如果节点Node有孩子，获得下一层分支=>Branch；保存当前测试活动集测试向量，测试活动集变成类别属于Node的测试向量集，包括被错分到Node的测试向量(多层分类的错误是向下层传递的)；递归调用树广度优先搜索算法，转步骤3。搜索函数返回，将测试活动集恢复到递归调用前状态。(2)继续取Node的下一个兄弟节点，如果Node没有下一个兄弟节点，返回到上一层递归调用返回点。否则取Node的下一个兄弟节点=>Node，转步骤6。

4 实验结果

4.1 训练结果

在进行本次SMO算法训练过程中，我们使用了9 000个训练样本，分别进行了单层和多层文本分类器训练的对比实验。

这些训练样本通过人工方式，按照政治、军事、经济、法律、农业、体育、卫生、工业、科技、交通、生活、宗教、天气分成13类。首先生成$S_1,S_2,\cdots,S_{13}$训练文件集，构成单层SVM的训练文件集。然后将政治、军事、经济、法律组成一个大类，工业、农业、交通、卫生组成一个大类，体育、科技、生活、宗教、天气组成一个大类，建立图3所示的训练文件信息树。

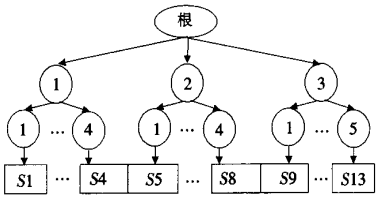

图3 训练文件信息树实例图

按照图3所示的训练文件信息树的结构，对初始训练文件进行组合，生成16个训练文件：1.mlt，$\cdots$，3.mlt；1_1.mlt，$\cdots$，1_4.mlt；2_1.mlt，$\cdots$，2_4.mlt；3_1.mlt，$\cdots$，3_5.mlt。这16个训练文件分别对应的分类树的16个SVM，训练生成16个模型文件：1.model，$\cdots$，3.model；1_1.model，$\cdots$，1_4.model；2_1.model，$\cdots$，2_4.model；3_1.model，$\cdots$，3_5.model。这16个模型文件构成了事实上的多层SVM分类树。

多层SVM比单层SVM要多训练3个训练文件。因此，训练一个多层SVM的时间要比训练一个单层SVM的时间一般要长。实验的结果表明，在相同的条件下，多层SVM的训练时间比单层SVM的训练时间总体上增加20%。

4.2 测试结果

本次测试从4 265个测试样本中按每类以一定比例抽取了984个测试数据集，分别对单层SVM和多层SVM进行了测试，测试得到的分类准确率结果如表1。

表1 分类准确率

分类器	测试样本数	错分数	准确度/%
单层SVM分类器	984	21	97.87
多层SVM分类器	984	17	98.27
多层SVM第一层分类器	984	10	98.98
多层SVM第二层第一个分类器	401	9	97.75
多层SVM第二层第二个分类器	261	1	99.61
多层SVM第二层第三个分类器	320	5	98.44

表中的数据表明多层分类的准确率比单层分类的准确率稍微高一些。另外，虽然多层SVM的训练时间比单层SVM的训练时间要长，但测试时间要比单层SVM的测试时间要短。我们在实验中记录了单层SVM测试时间为50s，多层SVM的测试时间为40s，测试时间减少了20%。从分类的过程来分析，在

(下转167页)

能。Servlet 控制器接收、解析、指派用户请求和生成输出页面的过程如下：

第一步：Servlet 接收到用户请求建立标识该用户的用户会话，并创建请求解析类。

第二步：请求解析类解析出用户所请求的元数据目录服务模块对应的 URL 及各项请求参数并负责设置输入输出编码的格式。

第三步：Servlet 将请求解析类、用户会话以及参数传入主指派类进行服务和输出指派。

第四步：主指派类先构造服务上下文，设置好其中的 URL、用户会话、服务器路径以及配置参数，然后将服务上下文及请求解析类交于服务指派类进行服务指派。

第五步：服务指派类根据请求的 URL 调用相应的元数据目录服务模块并将请求解析类传递给元数据目录服务模块进行具体处理。

第六步：元数据目录服务模块为用户请求服务后将 XML 格式的响应交于主指派类，主指派类再将响应传递给输出指派类进行输出指派。

第七步：输出指派类根据请求的 URL 取出该元数据目录服务模块所对应的所有页面输出类(即视图)并与响应进行匹配，得到正确的页面输出类。

第八步：页面输出类将响应、生成 Html 页面所需要调用的类和方法、字符串、图像、Html 链接以及元数据目录服务模块链接格式化为完整的 XML 文档并调用对应的 XSL 文件生成 Html 页面返回给用户。

Servlet 控制器为用户请求的服务方式分为两种：一种是并行处理方式，另一种是串行处理方式。并行处理方式是指 Servlet 控制器为用户请求服务时，可将具体操作分为两部分，两个部分相互间无因果关系，可同时并行处理，最后将结果汇总成 XML 格式的响应再调用相应的视图返回给用户；串行处理方式是指若一个元数据目录服务模块无法单独完成用户请求，Servlet 控制器将用户请求及该服务模块处理的结果传递至另一个元数据目录服务模块继续为用户请求服务。Servlet 控制器通过串行处理和并行处理相结合的方式为用户请求服务并指派给对应的元数据目录服务模块，完成了元数据的创建、导入、检索、显示、编辑和删除、资源上传和下载等目录服务的各种操作，实现了地理空间信息的共享和管理。

图 4 给出了系统管理员输入关键字为“中国”时的元数据检索结果，并显示了该用户所具有的权限以及可以对该元数据可以执行的操作。

图 4 元数据检索结果

4 结论

本文从地理空间信息共享的角度，分析了空间元数据目录服务和 MVC 设计模式的特点，并根据 ISO19115 国际元数据标准，实现了基于 MVC 模式的空间元数据目录服务。但该系统还有一些地方有待改善，比如系统对元数据标准可扩展性的支持及如何让用户定义自己所需要的元数据格式，还需要进一步的研究与探讨。(收稿日期：2005 年 8 月)

参考文献

1.OpenGIS Catalogue Services Specification.http://portal.opengis.org/files/?artifact_id=5929&version=1,2004-05-11

2.Subrahmanyam Allamaraju.Professional Java Server Programming J2EE Edition[M].China Machine Press,2001-09

3.顾蔚，陈天滋.基于 XML 的元数据系统在地理信息共享中的研究与设计[J].计算机应用研究，2003;20(1):104~107

(上接 154 页)

单层 SVM 分类过程中，一个测试样本被最终分类需要计算该样本到十三个超平面的距离；而在多层 SVM 分类过程中，一个测试样本被最终分类需要计算距离的超平面数为：第一层 3 个，第二层 4 或者 5 个，比单层分类少计算 5 到 6 个。所以，多层 SVM 分类速度比单层 SVM 的分类速度快。

5 结论

本文尝试建立了一个分层的 SVM 文本分类体系，这种层次结构的文本分类体系是一颗分类树，在树的根节点，将文本分成几个大类，然后往下继续细分，将属于某一大类的文本分到小类中，最后在叶节点得到文本的最终分类结果。

通过实验，本文比较了单层次结构文本分类和多层次结构文本分类的分类效果。结果表明，多层次结构文本分类系统的分类速度和准确率都有一定的提高。(收稿日期：2005 年 7 月)

参考文献

1.B E Boser,I M Guyon,V N Vapnik.A training algorithm for optimal margin classifiers[C].In:D Haussler ed.Proceeding of the 5th Annual ACM Workshop on Computational Learning Theory,ACM Press,1992:144~152

2.Vapnik V.The Nature of Statistical Learning Theory[M].Spinger Verlag,1995

3.Thorsten Joachims.Making large-scale SVM learning practical[C].In:B Schölkopf,C J C Burges,A J Smola eds.Advances in Kernel Methods-Support Vector Learning,MIT Press,1999:169~184

4.J Platt.Fast training of support vector machines using sequential minimal optimization[C].In:B Scholkopf,C Burges,A Smola eds.Advances in Kernel Methods-Support Vector Learning,MIT Press,1998

5.Nello Cristianini,John Shawe-Taylor.An Introduction to Support Vector Machines and Other Kernel-based Learning Methods[M].Cambridge University Press,2000

6.Ronan Collobert,Samy Bengio,Johnny Mariethoz.TORCH:A MODULAR MACHINE LEARNING SOFTWARE[R].IDIAP Research Report 02-46

7.K Aas,L Eikvil.Text categorization:A survey[R].Technical report,Norwegian Computing Center,1999-06

8.史忠植.知识发现[M].清华大学出版社，2002