基于 SVM-Adaboost 的中文组块分析

别 致[1],周俊生[2],陈家骏[1]
BIE Zhi[1],ZHOU Jun-sheng[2],CHEN Jia-jun[1]

1.南京大学 计算机软件新技术国家重点实验室,南京 210093
2.南京师范大学 计算机科学系,南京 210097
1.State Key Laboratory for Novel Software Technology,Nanjing University,Nanjing 210093,China
2.Department of Computer,Nanjing Normal University,Nanjing 210097,China
E-mail:biez@nlp.nju.edu.cn

BIE Zhi,ZHOU Jun-sheng,CHEN Jia-jun.SVM-Adaboost based Chinese text chunking.Computer Engineering and Applications,2008,44(21):171-173.

Abstract: Text chunking is a very important approach to preprocessing parsing.It divides text into syntactically related non-overlapping groups of chunks in order to reduce the complexity of the full parsing.In this paper,a SVM-Adaboost algorithm is applied for Chinese text chunking which combines Adaboost with linear-kernel SVM.This algorithm uses SVM as weak learners for AdaBoost and adjusts the kernel parameter of SVM in the learning process.The experimental results show that it is an effective approach.
Key words: Chinese text chunking;Adaboost;support vector machine

摘 要:组块分析是一种非常重要的句法分析预处理手段,通过将文本划分成一组互不重叠的片断,来达到降低句法分析的难度。提出一种基于 SVM-Adaboost 的中文组块分析方法,将基于线性核函数的支持向量机与 Adaboost 算法相结合,以基于线性核函数的 SVM 作为 Adaboost 的分量分类器,在学习过程中改变分量分类器的核参数。实验结果表明了该算法的有效性。
关键词:中文组块分析;Adaboost;支持向量机
DOI:10.3778/j.issn.1002-8331.2008.21.047 **文章编号:**1002-8331(2008)21-0171-03 **文献标识码:**A **中图分类号:**TP18

1 引言

近年来,基于机器学习方法的语言学习和获取成了自然语言处理的一个新热点。组块分析,也叫浅层句法分析或者部分句法分析,是一种比较新的语言句法分析处理策略。作为一种预处理手段,组块分析可以大大降低进行短语划分和短语分析处理的复杂性,为进一步对句子的深层次分析提供了基础,使得句法分析任务在某种程度上得到简化,同时对机器翻译、信息提取、信息检索、专有名词识别等都具有非常重要的意义。目前,应用于组块分析的机器学习方法有,基于转换的学习,基于记忆的学习,隐马尔科夫模型,最大熵等。

在中文组块分析方面,国内已经有人做过了一些有益的工作。例如周强[7]等提出了一种汉语短语的自动划分和标注方法,通过引入词界块和成分组概念,将成分识别问题从完整的句法分析任务中分离出来。赵军[8]等从语言学的系统角度定义了汉语基本名词短语,提出了将汉语基本名词短语的结构模板和其上下文环境特征结合的汉语基本名词短语识别模型。近些年来,多种机器学习方法已经被用于解决中文组块分析问题。李素建等利用最大熵模型进行汉语组块分析[11],然而基于最大熵的组块分析方法,有着计算开销较大,数据稀疏问题比较严重等缺点。徐中一,胡谦等利用 CRF 模型进行了汉语组块分析[13]。李衍,朱靖波等人提出了基于 SVM 的汉语组块识别算法[9]和基于 Stacking 算法[10]的多分类器组合方法,取得了较好的结果。

本文利用 SVM 分类器具有良好的泛化(generalization)性,而 Adaboost 算法可以自适应提高弱分类器的学习精度,提出一种基于 SVM-Adaboost 的中文组块分析方法,将线性核函数的 SVM 与 Adaboost 算法相结合,在学习过程中改变分量分类器的核参数。SVM-Adaboost 算法与已有的 Adaboost 算法相比,具有更好的泛化性能。将该算法应用于中文组块分析中,实验结果表明了该算法的有效性。

2 中文组块分析任务

Abney[4]首先提出了组块(chunk)的概念,并提出了一个完整的组块描述体系,对组块有着权威性的定义。他把组块定义为从句内的一个非递归的核心成分,这种成分包含核心成分的前置修饰成分,而不包含后置附属结构。组块不一定覆盖整个句子,例如:常有一些介词,连词等不是任何一个组块的部分。

基金项目:国家自然科学基金(the National Natural Science Foundation of China under Grant No.60673043);国家社科基金资助项目(No.07BYY051);江苏省自然科学基金(the Natural Science Foundation of Jiangsu Province of China under Grant No.BK2006117);江苏省高校自然科学基金资助项目(No.07KJB520057)。
作者简介:别致(1985-),女,硕士,主研方向:自然语言处理;周俊生,讲师;陈家骏,教授、博导。
收稿日期:2008-04-30 **修回日期:**2008-05-29

在宾州中文树库 CTB4 的基础上,一共定义了 9 种汉语组块类型:ADJP,ADVP,DP,DNP,LCP,NP,PP,QP,VP。各种组块类型的具体含义见表 1。

表 1 汉语组块类型定义

类型	定义
ADJP	Adjective Phrase
ADVP	Adverbial Phrase
DNP	DEG Phrase
DP	Determiner Phrase
LCP	Localizer Phrase
NP	Noun Phrase
PP	Prepositional Phrase
QP	Quantifier Phrase
VP	Verb Phrase

以下是一个中文文本组块的例子:

[VP 展出] [NP 各种 冰雕 作品] [MP 千余 件]

中文组块的标注形式有 IOE1,IOE2,IOB1,IOB2 和 IOBES 等多种。本文采用 IOB2 的标注集合,该标注集合包含三种类型的标记:B-X 表示 Chunk 类型为 X,并且是该 Chunk 的起始词;I-X 表示 Chunk 类型为 X,并且处于该 Chunk 的内部,即非起始词;O 表示不在任何 Chunk 内的词,即不属于任何组块。于是,上述例子也可以表示如下:

展出/B-VP 各种/B-NP 冰雕/I-NP 作品/I-NP 千余/B-QP 件/I-QP

于是,中文组块分析的过程就转换为对文本进行组块标注的过程,即分别对句子中的各个词按组块标记进行分类的过程。

3 基于 SVM 的 Adaboost 算法

在对句子中的词语分别进行组块标记时,使用支持向量机(SVM)作为分类器是一个比较好的选择。SVM 的基本思想是:利用核函数将特征空间从低维映射到高维,从而在高维空间中使得原数据集可以用超平面进行分割,并寻找一个满足分类要求的最优超平面,使其在保证分类精度的同时最大化超平面两侧的空白区域。常用的映射函数有线性核函数、多项式核函数以及高斯核函数。通常多项式核函数的 SVM 和高斯核函数 SVM 应用较多。但是,当特征空间维数很大时,这两类 SVM 算法学习速度比较慢。线性核函数学习效果相对较差,但是其学习速度很快。

Boosting[2]是在 PAC 学习问题框架模型下提出的一种提高任意给定弱分类器分类精度的方法,在解决实际问题时有一个重大的缺陷,即要求事先知道弱学习器学习正确率的下限,但这在实际问题中难以做到[3]。AdaBoost[6]即自适应提升(AdaBoost,Adaptive Boosting)算法,解决了早期 Boosting 算法很多实践上的困难,不需要预先知道弱学习器学习正确率的下限[5],可以很容易应用到实际问题中。

基于以上分析,提出一种将 Adaboost 与线性核函数 SVM 相结合的学习算法来提高汉语组块分析的效果。其基本思想是:依次训练一组分量 SVM 分类器,其中每个分量分类器的训练集都是选择由其它分量分类器给出的"最富信息"(most informative)的样本组成,最后用线性加权集成这些分量分类器,从而得出最终判决结果。其中,"最富信息"样本的选取方法:每个训练样本都被赋予一个权重,表明它被某个分量分类器选入训练集的概率。如果某个样本被当前弱分类器准确分类,那么它的权重就会被降低,则在构造下一个分量分类器的训练集时,它被选中的概率就被降低;相反,如果某个样本没有被正确分类,则它的权重就相应被提高,它入选下一个分量分类器的训练集的概率被提升。通过这种方式,Adaboost 能够"聚焦于"那些比较困难(容易出现错分)的样本。

在具体实现上,令每个训练样本的初始权重都相等,对于第 t 次迭代操作,需要根据第 $t-1$ 次训练得到的样本权重来选取新的训练样本集,进而训练分类器 Ct。然后,用分类器 Ct 对整个样本集进行测试,提高被它错分样本的权重,同时降低可以被正确分类样本的权重。之后,权重更新过的样本集被用来训练下一个分类器 $Ct+1$,整个训练过程如此迭代进行,直到满足结束条件为止。

如果弱分类器是个多分类学习算法,并且即便在 Adaboost 上产生的困难分布上也能获得较高的精度,那么该算法就能获得较好的学习效果。但是如果该算法在迭代过程中,某次迭代的正确率低于 50%,算法终止。对于多分类问题,这往往会导致算法提前终止,导致算法学习效果不佳。对此,Eibl 等人提出了改进算法[1],指出可以改变算法的终止条件,而不影响 Adaboost 算法收敛性。另外,为使 Adaboost 算法获得较好的学习效果,通常要求迭代时的分类器之间具有一定的差异性,王晓丹[12]等人的算法就是在每次迭代时,按照一定的规则设定对应 SVM 算法的学习参数,从而增加迭代时分类器之间的差异性,使得算法获得性能上的提升。SVM 有两个目标:分类间隔尽可能大,错划程度尽可能小。为把这两个目标综合为一个目标,引进一个惩罚参数 C 作为综合这两个目标的权重。线性 SVM 算法的学习效果受参数 C 的影响,所以可以改变 C 以产生具有差异性的分类器。因此,可以选择一组不同的惩罚参数 C,作为迭代时线性核 SVM 算法的学习参数,从而在迭代过程中生成一组具有差异性的弱分类器。

综上所述,可以得到基于线性核函数 SVM 的 Adaboost 算法,该算法与王晓丹等人提出的算法相比,使用了不同的 SVM 核函数,并在迭代中采用了更优的分类器权值设置,使得算法的性能得到了提高。算法描述如下:

第 1 步 输入:一组具有标记的训练样本集 $D=\{(x_1,y_1),\cdots,(x_n,y_n)\}$,$x_i\in X$,$y_i\in Y=\{l_1,\cdots,l_m\}$,其中 X 代表数据的特征集,Y 代表标签集,学习参数 $C_{List}=\{c_1,\cdots,c_{last}\}$;

第 2 步 初始化:初始化各样本对应的权值,令各样本的权值 $w_{11}(i)=1/n$,$i=1,2,\cdots,n$ 初始化学习参数 c_1,每个参数所对应的循环次数 T;

第 3 步

For $c_j=c_1,\cdots,c_{last}$

For $t=1,2,\cdots,T$

按照 $w_{jt}(i)$ 在 D 中采样,得到训练学习器的训练样本集 d_{jt}

以 d_{jt} 为训练样本集,c_j 为参数,训练弱分类器,得到 $h_{jt}(x)$

计算 $h_{jt}(x)$ 的训练误差 $\varepsilon_{jt}=\sum_{i=1}^{n}w_{jt}(i)\parallel y_i\neq h_{jt}(x_i)\parallel$,即 ε_{jt} 相当于错分样本的权值和

If $\varepsilon_{jt}>\frac{m-1}{m}$,continue

Else

设置当前分类器权值 $\alpha_{jt}=\ln\frac{(m-1)(1-\varepsilon)}{\varepsilon}$

更新各样本的权值,$w_{j(t+1)}(i)=$

$\frac{w_{jt}(i)\exp(-\alpha_{jt}(1-\|h_{jt}(x_i)=y_i\|))}{Z_{jt}}$，$Z_{jt}$为归一化算子使得，$\sum_{i=1}^{n} w_{j(t+1)}(i)=1$。

End

End

第 4 步 输出：总体分类器的判决函数值：$H(x)=\text{argmax}_{y\in Y}\sum \alpha_t \| h_t(x)=y \|$。

4　实验结果及分析

采用来自于 LDC 的中文树库 CTB4 作为训练语料和测试语料。取前 9 878 个句子作为训练集，后 5 290 个句子作为测试集。实验中的性能指标定义如下：

$$\text{组块准确率}(Precision)=\frac{\text{正确标注的组块的个数}}{\text{标注的组块的总个数}}\times 100\%$$

$$\text{组块召回率}(Recall)=\frac{\text{正确标注的组块的个数}}{\text{正确的组块的总个数}}\times 100\%$$

$$F_\beta=\frac{2\times Precision\times Recall}{Precision+Recall}\times 100\%$$

在特征的选取上，用来描述样本点的特征越多，可获得的关于样本的信息就越多，从而对样本点的识别也就更有利。充分利用当前标记位置的上下文信息，采用了其组合信息作为特征，将每一个样本 X 用以下特征来表示：

X=(pos0pos1，pos-1pos0，pos0pos2，pos-1pos0pos1，pos0pos1pos2，w@0pos@2，w@1pos@0，w@-1pos@0，pos@0pos@-1w@1，pos@0pos@-1w@0，t-1，t-2)

pos-1pos0pos1 的意思是前一个词的词性，当前词本身的词性再加上后一个词的词性，这三者共同构成一个特征。pos@0pos@-1w@1 是指当前词本身的词性，前一个词的词性与后一个词的组合。t-1 为前一个词的组块类型标记，t-2 为前数第二个词的组块类型标记。

实验中的分量分类器采用了 SVM_Multiclass(http://www.cs.cornell.edu/People/tj/svm_light/svm_multiclass.html)。当迭代次数 T=1 500 时，各种组块的识别结果见表 2。实验结果显示，比起仅仅基于 SVM 的中文组块分析，基于 SVM-Adaboost 算法的组块分析取得了良好的效果。在特征的选取方面，由于结合了丰富的上下文信息并将其进行了组合，使得组块分析的正确率和召回率都略有提高。从表 2 中可以看出，各种类型组块识别的结果还是有一定差异的。尤其是仅仅基于 SVM 时结果不佳的 ADVP 和 ADJP 组块，在 SVM-Adaboost 算法中得到了比较明显的改善。另外，由于 CRF 是目前比较主流的机器学习方法之一并在自然语言处理的多个领域都得到了广泛应用，还在表 2 中将实验结果与徐中一等人提出的基于 CRF 的中文组块分析做了比较，可以看到，本文的算法将 F 值提高了近 0.9%。

表 2　基于 SVM-Adaboost 算法与基于 SVM，CRF 的中文组块分析结果比较

Chunk type	SVM-Adaboost			SVM			CRF		
	P/%	R/%	F/%	P/%	R/%	F/%	P/%	R/%	F/%
All	90.13	89.40	89.76	86.84	87.72	87.28	87.94	89.76	88.84
NP	90.27	84.35	87.21	86.46	83.80	85.10	83.56	88.94	86.17
VP	89.10	88.73	88.91	83.01	87.66	85.27	87.12	83.68	85.37
PP	91.17	90.36	90.76	89.80	85.49	87.59	92.05	86.44	89.15
DP	88.34	86.19	87.25	81.96	84.77	83.34	80.78	85.17	82.91
QP	87.93	89.62	88.77	85.21	89.59	87.35	83.36	87.05	85.17
LCP	90.46	87.30	88.85	88.94	90.13	89.53	87.93	83.16	85.48
ADVP	92.55	90.16	91.33	86.56	82.94	84.71	83.81	88.47	86.08
DNP	93.70	92.49	93.09	90.97	91.06	91.01	90.27	84.34	87.22
ADJP	87.22	83.67	85.41	81.32	78.14	79.70	82.13	79.84	80.97

为了验证算法中采用线性核 SVM 作为弱分类器的效果，还将基于最大熵模型的 Adaboost 算法在数据集上做了实验，与 SVM-Adaboost 算法进行组块分析的比较(见图 1)。以测试结果中的 F 值作为迭代次数的函数。图 1 中 x 轴表示迭代次数，y 轴表示测试结果的 F 值。从图 1 中可以看出，SVM-Adaboost 算法在中文组块分析中取得的 F 值要高于基于最大熵的 Adaboost 算法，也证明了该算法比其他 Adaboost 算法具有更好的泛化性。

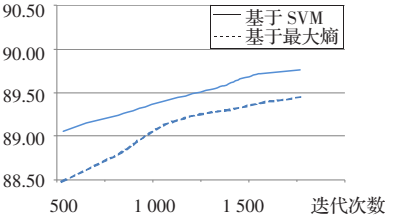

图 1　基于 SVM 的 Adaboost 与基于最大熵的 Adaboost 算法比较

5　结论及展望

中文组块分析是处于语句的分词标注和完整句法分析之间的一个步骤，它对于一些未登录词和常用语的识别有很好的效果，可以降低分词标注中的错误，同时也降低了完整语法分析的复杂度。本文将 Adaboost 算法与线性核的 SVM 算法相结合，可以构建出比较高效的学习算法。该算法学习速度较快，同时，又能够保证较高的正确率。实验结果表明，该算法取得了较好的分类效果。进一步的工作是改进该算法，将其运用于其他自然语言处理的任务，并争取能够在汉语浅层分析中发挥更大的作用。

参考文献：

[1] Eibl G，Pfeiffer K.How to make AdaBoost.M1 work for weak base classifiers by changing only one line of the code[C]// 13th European Conference on Machine Learning，2002：72-83.

[2] Schapire R E.The boosting approach to machine learning：An overview[C]//MSRI Workshop on Nonlinear Estimation and Classification，2002.

[3] Freund Y，Schapire R E.A decision-theoretic generalization of on-line learning and an application to boosting[J].Journal of Computer and System Sciences，1997，55(1)：119-139.

[4] Abney S P.Parsing by chunks[M]//Berwick R C，Abney S P，Tenny C.Principle-Based Parsing：Computation and Psycholinguistics.Dordrecht：Kluwer Academic Publishers，1991：257-278.

[5] Schapire R E，Singer Y，Bartlett P，et al.Boosting the margin：a new explanation for the effectiveness of voting methods[J].The Annals of Statistics，1998，26(5)：1651-1686.

[6] 涂承胜，刁力力，鲁明羽.Boosting 家族 Adaboost 系列代表算法[J].计算机科学，2003，30(3)：30-34.

[7] 张昱琪，周强.汉语基本短语的自动识别[J].中文信息学报，2002，16(6)：1-81.

(下转 211 页)