基于支持向量机的渐进直推式分类学习算法*

陈毅松+, 汪国平, 董士海

(北京大学 计算机科学技术系 人机交互与多媒体实验室,北京 100871)

A Progressive Transductive Inference Algorithm Based on Support Vector Machine

CHEN Yi-Song+, WANG Guo-Ping, DONG Shi-Hai

(HCI & Multimedia Laboratory, Department of Computer Science and Technology, Peking University, Beijing 100871, China)

+Corresponding author: Phn: 86-10-62765819, E-mail: cys@graphics.pku.edu.cn

http://www.pku.edu.cn

Received 2001-09-25; Accepted 2002-02-26

Chen YS, Wang GP, Dong SH. A progressive transductive inference algorithm based on support vector machine. *Journal of Software*, 2003,14(3):451~460.

Abstract: Support vector machine is a new learning method developed in recent years based on the foundations of statistical learning theory. It is gaining popularity due to many attractive features and promising empirical performance in the fields of nonlinear and high dimensional pattern recognition. TSVM (transductive support vector machine) takes into account a particular test set and tries to minimize misclassifications of just those particular examples. Compared with traditional inductive support vector machines, TSVM is often more practical and can give results with better performance. In this paper, a progressive transductive support vector machine is devised and the experimental results show that the algorithm is very promising on a mixed training set of a small number of unlabeled examples and a large number of labeled examples.

Key words: statistical learning; support vector machine; transductive inference

摘 要: 支持向量机(support vector machine)是近年来在统计学习理论的基础上发展起来的一种新的模式识别方法,在解决小样本、非线性及高维模式识别问题中表现出许多特有的优势.直推式学习(transductive inference)试图根据已知样本对特定的未知样本建立一套进行识别的方法和准则.较之传统的归纳式学习方法而言,直推式学习往往更具普遍性和实际意义.提出了一种基于支持向量机的渐进直推式分类学习算法,在少量有标签样本和大量无标签样本所构成的混合样本训练集上取得了良好的学习效果.

关键词: 统计学习;支持向量机;直推式学习

中图法分类号: TP181 **文献标识码**: A

基于结构化风险最小化方法的统计学习理论是一种专门的小样本统计理论.它为研究有限样本情况下的统计模式识别,并为更广泛的机器学习问题建立了一个较好的理论框架,同时也发展了一种新的模式识别方法——支持向量机(support vector machine,简称 SVM)[1~3].该方法已经得到了日益广泛的重视.

* Supported by the National Natural Science Foundation of China under Grant No.60033020 (国家自然科学基金)

第一作者简介: 陈毅松(1973－),男,四川资阳人,博士,讲师,主要研究领域为数字视频技术.

© 中国科学院软件研究所 http://www.jos.org.cn

虽然统计学习理论有比较坚实的理论基础和严格的理论分析,但是其中从理论到应用都还有很多尚未得到充分研究和解决的问题.例如,目前该领域的相关研究大多是试图设计某种分类器,使其对未来所有可能样本的预期性能最优,而在很多实际问题中,没有可能也没有必要用这样一个分类器对所有可能的样本进行识别,而往往只需要对一些特定的样本进行识别,于是可以考虑设计这样一种更为经济的分类器,用它来建立一种直接从已知样本出发对特定的未知样本进行识别和分类的方法和原则.相对于传统的归纳和演绎推理,这种推理方式在文献[4]中被称为直推式学习(transductive inference).统计学习领域的直推式学习是一个较新的研究领域,目前已经有了一些初步的研究成果[5,6].

本文是对直推式学习的进一步研究,试图寻找一个比已有的方法更为普遍和通用的直推式学习算法.本文在详细论述直推式学习思想的基础上,基于支持向量机分类的固有特点,设计了一个支持渐进直推式学习算法的支持向量机分类器.该分类器使用的渐进判别法充分利用了支持向量机的最优超平面分割特性,能够在训练过程中有效地对无标签样本循序渐进地作出判别分类,并具有一定的差错修复能力.同时,通过直推式学习,有效地优化了原始分类器的分类性能,得到了比直接进行归纳式学习好得多的测试结果.

本文第 1 节简单介绍了支持向量机分类算法的原理和实现.第 2 节介绍了直推式学习的概念、用途和研究现状,并重点描述了 T.Joachims 的直推式支持向量机分类算法.第 3 节结合支持向量机分类器的特点提出了渐进直推式支持向量机的学习算法 PTSVM,给出了具体的实现步骤和算法有效性的证明.第 4 节给出了算法的实验结果并作了详细的分析.第 5 节总结全文,并指出进一步研究的方向和思路.

1 支持向量机理论简述

考虑一个用某特征空间的超平面对给定训练数据集作二值分类的问题.对于给定样本点:

$$(x_1,y_1),\ldots,(x_l,y_l),x_i\in R^n,y_i\in\{-1,+1\},\tag{1}$$

其中向量 x_i 可能是从对象样本集中抽取某些特征直接构造的向量,也可能是原始向量通过某个核函数映射到核空间中的映射向量.

在特征空间中构造分割平面:

$$(w\cdot x)+b=0,\tag{2}$$

使得

$$\begin{cases}(w\cdot x_i)+b\ge 1 & y_i=1\\ (w\cdot x_i)+b\le -1 & yi=-1\end{cases}\Leftrightarrow y_i[(w\cdot x_i)+b]\ge 1,\ i=1,2,\ldots,l.\tag{3}$$

可以计算出,训练数据集到一给定的分割平面的最小距离为

$$p(w,b)=\min_{\{x_i|y_i=1\}}\frac{w\cdot x_i+b}{|w|}-\max_{\{x_i|y_i=-1\}}\frac{w\cdot x_i+b}{|w|}=\frac{2}{|w|}.\tag{4}$$

从 SVM 对优化分割平面的定义可以看出,对该平面的求解问题可以简化为:在满足条件式(3)的情况下,计算能最大化 $p(w,b)$ 的分割平面的法向量 w 和偏移量 b.Vapnik 等人证明了分割超平面的法向量 w_0 是所有训练集向量的线性组合.即 w_0 可以描述为

$$w_0=\sum_{i=1}^{l}(a_i^0y_i)x_i\quad(a_i^0\ge 0),i=1,\ldots,l.\tag{5}$$

定义判别函数

$$f(x)=w_0\cdot x+b_0,\tag{6}$$

则测试集的分类函数可以描述为

$$label(x)=sgn(f(x))=sgn(w_0\cdot x+b_0).\tag{7}$$

由式(3)可知,在线性可分的情形下,对所有的训练样本都应该满足 $|f(x)|\ge 1$.在下文中,我们把满足 $|f(x)|<1$ 的区域称为分割超平面所对应的边界区域.

在多数情况下,式(5) w_0 的展开式中,系数 a_i^0 为零值,而非零值的 a_i^0 所对应的 x_i 就称为支持向量 SV.这些向量充分描述了整个训练数据集数据的特征,使得对 SV 集的线性划分等价于对整个数据集的分割.

© 中国科学院软件研究所 http://www.jos.org.cn

由式(4)可见,最优分割平面的求解等价于在式(3)约束下最大化下面的式(8):

$$\Phi(w)=\frac{1}{2}\|w\|^2. \tag{8}$$

引入拉格朗日乘子 $\alpha_i, i=1,2,...,l$,并定义

$$w(\alpha)=\sum_{i=1}^{l}\alpha_i y_i x_i. \tag{9}$$

使用 Wolfe 对偶定理把上述问题转化为其对偶问题:

$$\left.\begin{array}{l}\mathrm{Max}W(\alpha)=\sum_i \alpha_i-\frac{1}{2}w(\alpha)\cdot w(\alpha)\\ \text{subject to } \alpha_i\ge 0, \sum_i a_i y_i=0\end{array}\right\}. \tag{10}$$

对于线性不可分的训练集,可以引入松弛变量 ξ_i,把式(8)改写为下面的求解问题[3]:

$$\left.\begin{array}{l}\mathrm{Min}(\frac{1}{2}\|w\|^2+C\sum_i \xi_i)\\ \text{Subject to } y_i(w\cdot x_i+b)\ge 1-\xi_i, \xi_i\ge 0\end{array}\right\}. \tag{11}$$

类似地可以得到相应的对偶问题:

$$\left.\begin{array}{l}\mathrm{Max}W(\alpha)=\sum_i \alpha_i-\frac{1}{2}w(\alpha)\cdot w(\alpha)\\ \text{subject to } 0\le\alpha_i\le C, \sum_i a_i y_i=0\end{array}\right\}. \tag{12}$$

形如式(10)和式(12)的求解是一个典型的有约束的二次型优化问题,已经有了很多成熟的求解算法,近年来,V.Vapnik,C.Burges,E.Osuna,T.Joachims,J.Platt 等人的一系列工作使得对大规模训练集的支持向量机算法的实现成为可能[7~11].

2 直推式学习

在直推式学习[12,13]中,学习机在训练过程中使用较少的有标签样本和较多的无标签样本进行学习.直推式学习的一个重要特点就在于,在这样一种混合样本的学习过程中,测试集的样本分布信息从无标签样本转移到了最终的分类器中.由于无标签样本的数量较多,所以与有标签样本相比能够更好地刻画整个样本空间上的数据特性,从而使训练出的分类器具有更好的推广性能.直推式学习在模式识别的各个领域中都已有了不同程度的研究和应用[14,15].

有关基于支持向量机的直推式学习算法的研究尚处于起步阶段,目前最主要的研究成果是 T.Joachims 的直推式支持向量机(TSVM).下面我们简单介绍一下 TSVM 的算法原理和实现.具体的描述和证明参见文献[6].

给定一组独立同分布的有标签训练样本点:

$$(x_1,y_1),\ldots,(x_n,y_n), x_i\in R^m, y_i\in\{-1,+1\} \tag{13}$$

和另一组来自同一分布的无标签样本点:

$$x_1^*, x_2^*, x_3^*, ..., x_k^*. \tag{14}$$

在一般的线性不可分条件下,T.Joachims 的直推式向量机的训练过程可以描述为以下的优化问题:

$$\left.\begin{array}{l}\text{Minimize over } (y_1^*,...,y_k^*,w,b,\xi_1,...,\xi_n,\xi_1^*,...,\xi_k^*)\\ \frac{1}{2}\|w\|^2+C\sum_{i=1}^{n}\xi_i+C^*\sum_{j=1}^{k}\xi_j^*\\ \text{subject to}: \forall_{i=1}^{n}: y_i\left[w\cdot x_i+b\right]\ge 1-\xi_i\\ \qquad\qquad \forall_{j=1}^{k}: y_j\left[w\cdot x_j^*+b\right]\ge 1-\xi_j^*\\ \qquad\qquad \forall_{i=1}^{n}: \xi_i\ge 0\\ \qquad\qquad \forall_{j=1}^{k}: \xi_j^*\ge 0\end{array}\right\}. \tag{15}$$

其中参数 C 和 C^* 为用户指定和调节的参数.与式(11)中的参数 C 作用类似.我们把参数 C^* 称作是无标签样本在

© 中国科学院软件研究所 http://www.jos.org.cn

训练过程中的影响因子,$C^*\xi_j^*$ 称为无标签样本 j 在目标函数中的影响项.

TSVM 的训练过程也就是求解上述优化问题的过程.训练算法大致可以分为以下几个步骤:

步骤 1. 指定参数 C 和 C^*,使用归纳式学习对有标签样本进行一次初始学习,得到一个初始分类器,并按照某个规则指定一个无标签样本中的正标签样本数 N.

步骤 2. 用初始分类器对无标签样本进行分类,根据对每一个无标签样本的判别函数输出,对输出值最大的 N 个无标签样本暂时赋正标签值,其余的赋负标签值,并指定一个临时影响因子 C^*_{tmp}.

步骤 3. 对所有样本重新训练,对新得到的分类器,按一定的规则交换一对标签值不同的测试样本的标签符号,使得优化问题式(15)中的目标函数值获得最大下降.这一步骤反复执行,直到找不出符合交换条件的样本对为止.

步骤 4. 均匀地增加临时影响因子 C^*_{tmp} 的值并返回到步骤 3,当 $C^*_{tmp} \geq C^*$ 时,算法结束,并输出结果.

步骤 3 中的标签符号交换保证了交换后的解优于交换前的解,而步骤 4 中的临时影响因子由小到大逐步递增,则试图通过逐渐增加无标签样本对算法的影响的办法来追求尽可能小的对无标签样本的分类误差.由于步骤 1 中指定的 C^* 是有限的,由步骤 4 中的结束准则可知,算法能够在有限次执行后终止,并输出结果.

3 一种新的渐进直推式支持向量机学习算法

由于成功地把无标签样本中隐含的分布信息引入了支持向量机的学习过程中,TSVM 算法比单纯使用有标签样本训练得到的分类器在性能上有了显著提高.但是,TSVM 算法本身仍然存在着一些不足和值得进一步改进的方面.TSVM 算法的一个主要缺陷在于,在算法执行之前必须人为地指定待训练的无标签样本中的正标签样本数 N,而在一般情况下,N 值是很难作出比较准确的估计的,在 TSVM 算法中采用了一种简单的方法,即根据有标签样本中的正标签样本所占比例来相应地估计无标签样本中的正标签样本比例,进而估计出 N 值.不难看出,这一方法在有标签样本数较少的情况下很容易导致较大的估计误差,一旦事先设定的 N 值和实际上的正标签样本数相差较大,就会导致学习机性能的迅速下降.

例如,一种常见的情况是,在有标签样本中正负标签各占一半,TSVM 算法即假定无标签样本中也是正负标签各占一半并据此设定 N 值.但是,实际应用中的样本在两个类别中的分布可能是完全不均衡的,有很大的可能是某个类别中的样本数要远远多于甚至数倍于另一个类别中的样本数.这种分布的不均衡性虽然往往在大数量的无标签样本中有所体现,但是对于训练算法而言却是未知的.由于 TSVM 算法错误地估计了 N 值,将导致训练算法产生一个不能正确描述样本分布特征的学习器.这一缺陷在很大程度上限制了 TSVM 算法的实用价值.

本节试图使用一种新的直推式学习算法来解决上述问题,在这种新的直推式算法中,没有必要事先设定无标签样本中的正标签样本数,而是在训练过程中根据一定的原则动态地对无标签样本逐一赋予可能的标签,并对新得到的有标签样本集重新训练.在这一过程中,动态地调整最优分割平面并用类似于 TSVM 中的方法来渐进地求解式(15)所描述的优化问题.由于这种算法没有对无标签样本中的正标签样本数作出盲目规定,而是在训练过程中渐进地对无标签样本赋予标签并动态地予以调整,所以可以预期,该算法所产生的分类器可以更好地描述样本的分布特征,从而具有更好的推广性能.在下文中,我们把这种算法称为渐进直推式支持向量机(progressive transductive support vector machine,简称 PTSVM).

PTSVM 算法的基本思想可以描述如下:考虑到有标签样本往往不是随机获得的,而是人工处理后的有一定代表性的样本,所以没有理由把它的标签分布作为估计整个样本中正负标签大致比例的根据.而且,即使有标签样本是随机获得的,由于直推式学习中的有标签样本数量往往很少,用它来估计整个样本中正负标签大致比例往往是相当不准确的.所以在 PTSVM 算法中,在训练开始之前,不对无标签样本的分布特征作任何估计.而是在训练过程中,一次选择 1~2 个有可能对后续训练过程产生较大影响的无标签样本,赋予其当前状态下最可能的标签值,然后,把它们加入到有标签样本中,并进行新一轮的训练.一般地,新加入样本将会影响新一轮训练的过程,并导致当前分割平面的少量偏移.在这一过程中,可能会发现部分先前的标注是不合适的,一旦发现这种情况,就取消这个不合适的标注,使其恢复为无标签的样本.如果精心设计这种渐进赋值和动态调整的规则,就可

© 中国科学院软件研究所 http://www.jos.org.cn

以预期分割平面将在训练过程中逐步逼近最优分割平面,并在训练结束时获得式(15)的一个局部最优解.

在以上算法中,对于无标签样本的标注不是一次完成的,而是渐进地进行的,我们希望每一次标注都能够尽可能地准确,只有这样,才能保证后续过程的正确进行,而不会导致误差的传递.所以,我们希望较先被标注的样本有足够的能力来向好的方向调整当前的分割平面,使得一些在原先的分割准则下可能被误分的无标签样本逐渐向正确的方向移动,并最终能够在某个分割平面下得到正确的分类.基于这一考虑,我们放弃了 TSVM 算法中由小到大逐渐增加 C^* 的作法,而在一开始就给 C^* 赋予一个较大的值,较大的 C^* 值意味着新标注的样本对后续的训练过程有较大的影响能力.事实上,在 TSVM 算法中,采用由小到大逐渐增加 C^* 的作法来逐步增大无标签样本在训练中的影响程度,是为了在同时对所有无标签样本进行标注的情况下对无标签样本的标注能够更加慎重地进行,而在本算法中,由于取消了同时对所有无标签样本进行标注的作法,而用渐进标注的方法取而代之,因此由小到大逐渐增加 C^* 的方法已不再适用.

在 PTSVM 算法中,需要精心设计对无标签样本渐进赋值和动态调整的规则,这是整个算法的核心内容.一个好的设计规则有助于学习机从已有的数据更好地刻画出整个数据的分布特征.一个简单的想法是,在训练的每一步,用当前的有标签样本作归纳式学习得到当前分割平面和形如式(6)的判别函数,并计算出当前所有无标签样本的判别函数值.按照式(6)选出当前分割平面的边界区域中判别函数绝对值最大的一个无标签样本.

$$\mathrm{Max}\left\|f(x_i^*)\right\|,\mathrm{s.t.}\left\|f(x_i^*)\right\|<1. \tag{16}$$

对按照式(16)选出的无标签样本,按照下面的式(17)将其判别函数的符号值作为其标签值赋予该样本.也就是说,选择一个对其标签最有把握的处于边界区域的无标签样本作为赋予标签的对象.

$$Label(x_i^*)=sgn(w\cdot x_i^*+b). \tag{17}$$

以上作法有一个潜在的缺点,在某些样本分布下,第 1 个无标签样本被标注之后,由于受该样本的影响,新的分割平面倾向于朝该样本点被标注类的相反类的方向移动,这会进一步导致下一个被标注的无标签样本的标注和前一个相同,这一过程积累下去,分割平面向某个类的方向不断移动,终将导致该类的部分样本点的误分.

为克服这一缺点,在 PTSVM 算法中采用了一次标注两个样本点的作法.即在每一次新的训练完成之后,按照下面的式(18)标注一个新的正标签,同时按照式(19)标注一个新的负标签.

$$\mathrm{Max}(f(x_i^*)),\mathrm{s.t.}0<f(x_i^*)<1, \tag{18}$$

$$\mathrm{Min}(f(x_i^*)),\mathrm{s.t.}-1<f(x_i^*)<0. \tag{19}$$

如果在某次训练后不存在满足式(18)的无标签样本,则在当前循环体中不标注新的正标签,反之,如果不存在满足式(19)的无标签样本,则在当前循环体中不标注新的负标签.这一过程持续下去,直到某次训练后所有的无标签样本都不出现在当前最优分割平面的边界区域中.此时,我们认为得到了问题的最优解,对所有余下的无标签样本用当前的分割平面和判别准则进行分类并加标签,然后结束算法并输出结果.这个方法不但可以有效地避免以上缺点,而且可以加快标注的速度,减少到达收敛所需要的迭代次数.我们把这种方法称为成对标注法.

在成对标注法的进行过程中,有可能在某一次训练后发现一个或多个已标注的无标签样本值和用当前分割平面对其分类所得到的标签值不一致,这一现象指示了一个可能在迭代过程中早期出现的误标.在这种情况下,就把发生了不一致现象的样本重新置为无标签状态,并继续执行迭代,这个样本有可能在未来的某次训练后得到更为可靠的新的标注.我们把这种方法称为标签重置法.标签重置法使得 PTSVM 具有了一定的差错修复能力.

成对标注法和标签重置法是 PTSVM 的核心思想.综合以上分析,可以写出渐进直推式支持向量机训练算法的主要步骤:

步骤 1. 指定参数 C 和 C^*,使用归纳式学习对有标签样本进行一次初始学习,得到一个初始分类器.

步骤 2. 用初始分类器对无标签样本进行学习,计算每一个无标签样本的判别函数输出,用成对标注的法则在当前边界区域内的无标签样本标注一个新的正标签和一个新的负标签.

步骤 3. 对所有样本重新训练,计算每一个无标签样本的判别函数输出.如果发现某一个早期标注的无标签

© 中国科学院软件研究所 http://www.jos.org.cn

样本的标签值和当前判别函数的输出值不一致,则按照标签重置的法则取消对该样本的标注.

步骤 4. 用成对标注法寻找当前边界区域内符合新加标注条件的未标注的无标签样本.如果存在这样的无标签样本,则对其加以标注并返回步骤 3;如果不存在这样的无标签样本,则用当前的分割平面对剩下的全部无标签样本做分类并加标签.算法结束,并输出结果.

下面两个定理证明了成对标注法和标签重置法的合理性.

定理 1. 在 PTSVM 算法的一个迭代过程中,对于满足式(18)和式(19)的两个无标签样本 x_1^* 和 x_2^* 的所有标注方法,按照成对标注法标注所得到的形如式(15)中的目标函数是最小的.

证明:设 x_1^* 是满足式(18)的样本点,显然,x_1^* 处在当前分割平面的边界区域内,所以无论给 x_1^* 赋予正标签还是负标签,必然有相应的松弛变量 $\xi^* > 0$.设给 x_1^* 赋予正标签时对应的松弛变量为 ξ_+^*,给 x_1^* 赋予负标签时对应的松弛变量为 ξ_-^*.也即

$$\left.\begin{aligned}\left[w\cdot x_1^*+b\right]&=1-\xi_+^*\\ -\left[w\cdot x_1^*+b\right]&=1-\xi_-^*\end{aligned}\right\}. \tag{20}$$

由式(20)立即可得

$$\xi_+^* + \xi_-^* = 2. \tag{21}$$

由式(21)可知,当 $\xi_+^* < 1$ 即 $f(x_1^*) > 0$ 时,有 $\xi_+^* < \xi_-^*$,从而有

$$C^* \cdot \xi_+^* < C^* \cdot \xi_-^*. \tag{22}$$

也就是说,给 x_1^* 赋予正标签时目标函数中的 x_1^* 的影响项的值小于给 x_1^* 赋予负标签时目标函数中的 x_1^* 的影响项的值.所以,给 x_1^* 赋予正标签时将得到更小的目标函数值.同理可以推出,给 x_2^* 赋予负标签时将得到更小的目标函数值.所以,对于 x_1^* 和 x_2^* 的所有标注方法而言,按照成对标注法标注所得到的目标函数是最小的. □

定理 2. 对于 cancelimproperlabel()过程找到的不一致样本点 x_i^*,依据当前判别函数的符号对其赋予标签可以得到更小的目标函数值.

证明:不失一般性,设 x_i^* 的早期赋予标签值为负,而当前判别值为正,设在当前的判别条件下,给 x_i^* 赋予正标签时对应的松弛变量为 ξ_+^*,给 x_i^* 赋予负标签时对应的松弛变量为 ξ_-^*.参考式(20)可知,必定有

$$0 \le \xi_+^* < 1, \xi_-^* > 1. \tag{23}$$

从而必定满足 $\xi_+^* < \xi_-^*$,因此可以得到和式(22)一致的结论:

$$C^* \cdot \xi_+^* < C^* \cdot \xi_-^*. \tag{24}$$

所以,给 x_i^* 赋予正标签时将得到更小的目标函数值.

对于 x_i^* 的早期赋予标签值为正,而当前判别值为负的情况,同理可得

$$C^* \cdot \xi_-^* < C^* \cdot \xi_+^*. \tag{25}$$

所以,给 x_i^* 赋予负标签时将得到更小的目标函数值. □

定理 2 的证明指出,依据当前判别函数的符号对其赋予标签可以得到更小的目标函数值.这说明,对于当前的分割平面和判别准则而言,早期赋予的标签值是不合适的.但是这并不表示早期赋予的标签值一定是错误的,而仅仅意味着在目前状况下所掌握的信息还不适合对该样本点仓促地赋予标签,而是需要通过进一步的训练来试图获得关于它的更准确的判决信息.所以,在 PTSVM 算法中,对于这样的样本点,不是立刻改变对其赋予的标签值,而是简单地把其重置为未标注的无标签样本,把对它的判决推迟到积累了更多的关于样本分布的知识之后再进行.

4 实验结果和分析

我们基于 T.Joachims 的 SVMLight 软件[16]实现了上述的渐进直推式支持向量机学习算法 PTSVM.下面将给出 PTSVM 算法在两个不同数据集上的实验结果和分析.

© 中国科学院软件研究所 http://www.jos.org.cn

4.1 Tutorial数据集上的实验

第 1 个实验数据集 Tutorial 是一个简单的线性可分的二维点集.设计这个实验的目的是为了能够通过该实验更为直观地阐述 PTSVM 算法的特点和优势.初始训练集中的部分数据分布如图 1 所示,图中的 3 个“+”表示有正标签的样本,图中的 3 个“○”表示有负标签的样本,其余的黑点表示未经标注的无标签样本.从图中可以清楚地看出,该数据集在二维空间中具有良好的线性可分性质.这个数据集具有以下特点:

(1) 无标签样本的数据量要远远大于有标签样本的数据量.

(2) 无标签样本的数据量较多,能够较好地反映整个数据集的数据分布特征,从中可以看出,在整个样本集上,具有实际正标签的样本数要多于具有实际负标签的样本数.

(3) 因为有标签样本的数据量较少,所以有标签样本的分布特征不能反映整个数据集的分布特征.

由于具有这 3 个特征,因此该数据集是一个用于检验直推式学习的简单明了的例子.

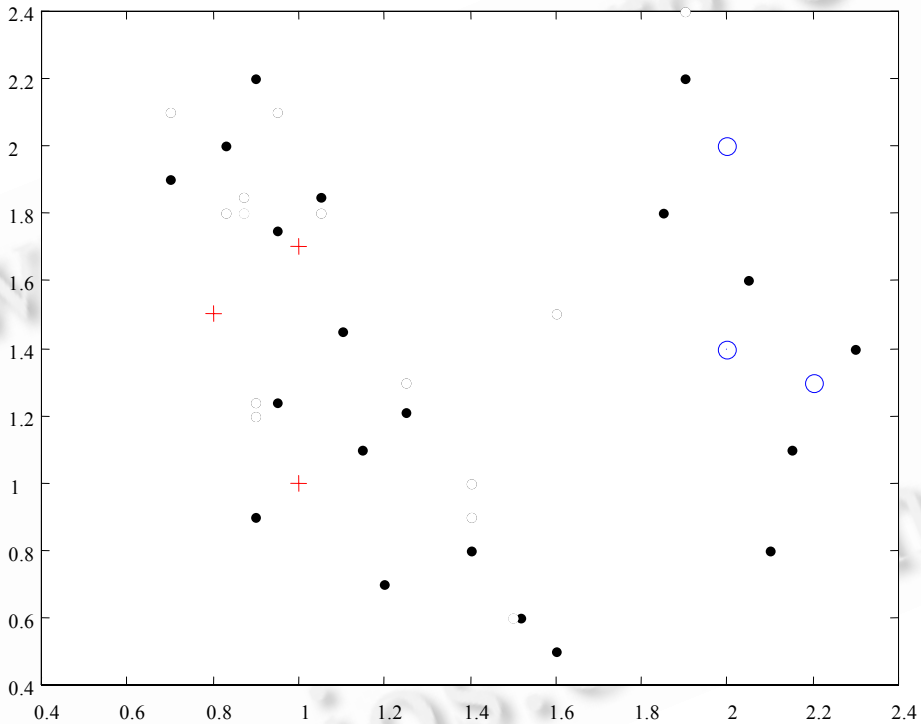

Fig.1 Training data of a Tutorial dataset

图 1 一个 Tutorial 训练样本集

从图 1 中可以直观地看出,对于 Tutorial 数据集来说,采用有标签样本中正负标签所占的比例来估计无标签样本中的正负标签的方法是不合适的,因为对这个数据集中的总体分布而言,具有正标签的样本数占有压倒性的多数.TSVM 由于在一开始就没有能对无标签样本的正负标签比例作出比较准确的估计,很难获得理想的学习结果.相比之下,由于 PTSVM 算法不需要估计无标签样本中的正负标签的比例,从而有效地避免了这一问题的产生.

表 1 给出了 TSVM 和 PTSVM 在该数据集上的训练和测试结果比较.

Table 1 Comparison of the training and test result of TSVM and PTSVM on Tutorial dataset

表 1 TSVM 和 PTSVM 在 Tutorial 数据集上的训练和测试结果比较

	Training time	Training errors	Total points in test set	Test errors	Test accuracy (%)
TSVM	0.67	4	100	15	85
PTSVM	0.17	0	100	0	100

© 中国科学院软件研究所 http://www.jos.org.cn

从表 1 可以看出,PTSVM 可以获得比 TSVM 高得多的测试准确率,这是因为在学习机的训练中,PTSVM 比 TSVM 更合理地利用了无标签样本中所隐含携带的关于数据分布特征的信息.下面我们仔细地分析一下这两种算法的训练过程.

训练集中共有 3 个正标签样本,3 个负标签样本和 20 个无标签样本.TSVM 算法在初始化的过程中就假设无标签样本中有 10 个应该赋予正标签,由图 2 可以看出,这个假设已经在很大程度上偏离了实际数目,因为事实上在无标签样本中有 14 个应该赋予正标签.由于 TSVM 在训练过程中始终试图为 10 个无标签样本赋正标签,这就决定了在它的训练结果中必然有一些实际上具有负标签的无标签样本被错误地赋予了正标签.由于受这些被错误标注的样本的影响,TSVM 算法最终得到的分割平面也是有误差的,这种误差在训练结果中体现为部分无标签样本被错误标注(表 1 中的 4 个训练误差),在测试结果中则体现为正标签的招回率或负标签的精度不高(表 1 中的 15 个测试误差),导致整体分类准确率的下降.

PTSVM 算法则不同,在 PTSVM 算法中,无标签样本的正标注率不是事先估计而是在训练过程中逐步变化的,直到训练结束的时候才能最终确定下来,显然,这种作法可以适应更一般的数据分布规律,从而具有更好的推广性.跟踪 PTSVM 算法的训练过程可以看出,每一次新的成对标注操作都促使分类平面朝着正确的方向偏移,这种偏移使得部分原先可能被误分的样本点也能在一定次数的迭代后被正确标注.在 PTSVM 的训练结果中,所有的无标签样本均得到了正确标注.注意,图 1 中最下面一个坐标大致为(1.6,0.5)的无标签样本点,只使用有标签样本进行的初始训练获得的分类器将会把该样本点错误分类,但是由于新加入的样本点影响了后面的训练过程中生成的分类器,使其判别平面朝正确的方向不断偏移,该样本点最终能够被正确标注.这个样本点从错误标注到正确标注的过程直观地体现了 PTSVM 算法的优势.

从表中还可以看出,PTSVM 算法的训练时间比 TSVM 算法少得多.这主要是因为 TSVM 算法需要逐渐增加临时影响因子 C^*_{tmp} 的值,在每一个值下都要重复一次求解过程,因而耗费的时间较多.而在 PTSVM 算法中,C^* 的值是直接指定的,只需要在一个值下进行求解,因而耗费的时间较少.需要指出的是,这个结果不具有普遍性,而是仅仅在小训练集的情况下成立.在训练集样本数较多的情况下,PTSVM 渐进标注方法的复杂度就会相应增加.这从下文的第 2 个实验的结果中可以看出来.

通过以上实验可以直观地看出,对低维空间中线性可分的数据集而言,如果数据的分布相对集中,则 PTSVM 算法和 TSVM 算法相比有相当明显的优势.在高维空间或者线性不可分的情况下不太直观.但是,我们预期 PTSVM 的优越性仍然能够充分或部分地得到体现,这可以通过下面的第 2 个实验来加以证实.

4.2 Reuters数据集上的实验

第 2 个实验的数据集来自著名的 Reuters-21578 数据集.该数据集选自路透社 1987 年的新闻专线,是一个典型的文本分类用实验数据集[16].我们的实验数据是对 T.Joachims 的共享预处理数据集进行改编而得到的.每一个文本都用高维空间中的特征向量来表示,每一个特征对应于一个单词词根.实验目的是训练一个分类器来找出有关“corporate acquisition”的文本.

在所有实验的训练集中均采用同一组有标签样本,即事先手工分类好的 5 个正标签样本和 5 个负标签样本.训练集中的无标签样本则按照不同的数量和不同的比例来选取,以测试算法在各种可能的无标签样本数据分布下的性能.我们预期,无标签样本的数量越多,则其分布越能够反映整个数据集的真实分布情况.所有的实验共用一个含有 300 个正标签样本和 300 个负标签样本的测试集.

表 2 中给出了 8 组不同的初始训练样本集下 TSVM 和 PTSVM 学习算法的结果.其中第 1 组训练集中不包含无标签样本,也就是说,在这一组样本集上,TSVM 算法和 PTSVM 算法进行的实际上都是传统的归纳式学习,因而学习的过程和结果都是完全一样的.后面的 7 组训练集中包含了数量不等、正负比例不同的无标签样本,用于检验 TSVM 算法和 PTSVM 算法在各种不同的无标签训练样本分布下的性能.

从表 2 的数据中可以得出以下结论:

由于直推式学习适当考虑了无标签样本中所包含的信息,TSVM 算法和 PTSVM 算法都能够获得比原始的归纳式学习算法更好的性能.而且,训练中所使用的无标签样本越多,训练产生的分类器性能就越好.

比较同一个训练集下 TSVM 算法和 PTSVM 算法的性能可以看出,当训练集中无标签样本的实际正负比例

© 中国科学院软件研究所 http://www.jos.org.cn

大致相等时,TSVM 算法由于在训练的初始化阶段正确设置了正标签估值参数而在测试集上获得了比 PTSVM 算法更好的性能.在更一般的情况下,训练集中无标签样本的实际正负比例是不相等的,而且可能有较大的差别,此时 PTSVM 算法能够在训练过程中不断积累无标签样本的分布知识,并自动地调节正负标签赋予的比例,从而具有更好的适应性和灵活性,获得了比 TSVM 算法更好的结果.

Table 2 Results comparison of TSVM and PTSVM on Reuters dataset

表 2 TSVM 和 PTSVM 在 Reuters 数据集上的学习效果比较

Unlabeled examples	Train algorithm	Training time (s)	SV num	Test errors	Precision (POS) (%)	Recall (POS) (%)	Test accuracy (%)
Pos=0	TSVM	0.01	10	165	64.64	99.33	72.50
Neg=0	PTSVM	0.01	10	165	64.64	99.33	72.50
Pos=10	TSVM	0.77	30	125	71.19	98.00	79.17
Neg=10	PTSVM	0.30	30	91	77.43	98.33	84.83
Pos=20	TSVM	0.82	39	91	90.98	77.33	84.83
Neg=10	PTSVM	0.35	39	79	89.35	83.00	86.83
Pos=100	TSVM	5.17	181	43	89.78	96.67	92.83
Neg=100	PTSVM	3.62	184	46	87.57	98.67	92.33
Pos=50	TSVM	4.19	146	109	73.46	99.67	81.83
Neg=100	PTSVM	2.40	140	93	76.61	99.33	84.50
Pos=1000	TSVM	58.87	768	54	97.67	84.00	91.00
Neg=500	PTSVM	167.60	805	41	97.44	88.67	93.17
Pos=500	TSVM	110.72	697	87	77.66	99.67	85.50
Neg=1000	PTSVM	175.48	770	80	80.22	97.33	86.67
Pos=1000	TSVM	69.96	826	19	95.47	98.33	96.83
Neg=1000	PTSVM	303.05	894	36	90.74	98.00	94.00

实验表明,在 PTSVM 算法的训练过程结束以后,对于训练集中无标签样本所赋标签的正负比例总是比 TSVM 算法更接近实际正负标签的比例,在无标签样本较少的情况下,实验比例和实际比例往往相当接近,随着实验中无标签样本的增多,实验比例和实际比例的差距渐渐增大,但仍然比 TSVM 算法更接近于实际比例.这种对无标签样本正负比例的更精确的估计使得 PTSVM 算法比 TSVM 算法有更好的性能.

最后,我们来分析一下表 2 中 TSVM 算法和 PTSVM 算法的训练时间.正如上文所提到的,在无标签样本数较少的情况下,PTSVM 算法有更快的执行速度.但是在无标签样本较多的时候,频繁的成对标注和标签重置后的再训练过程使得 PTSVM 算法的复杂度迅速增加,并且超过了 TSVM 算法,这是 PTSVM 算法的一个不足之处.一种可能的改进是在训练过程中采用增量和减量学习的方法来加快训练的速度[17].

4.3 进一步的改进措施

下面我们结合以上两个实验的结果,进一步分析一下 PTSVM 算法的特性和可能的改进方法.在基于 Reuters 数据集的实验中,虽然总体而言,PTSVM 算法的性能要比 TSVM 算法优越,但是测试结果表明,在大多数情况下,这种优势不像在 Turtorial 数据集上那么明显.这种差别主要是由于 Reuters 和 Tutorial 两个数据集的特性不同造成的.Turtorial 的数据样本的维数为 2,而 Reuters 的数据样本的维数接近 10 000,高维空间的情况比低维空间要复杂得多,我们对 Reuters 数据样本在其相应的特征向量空间的分布情况远没有对 Tutorial 数据集那么明了,可以想象,两者的分布特征可能存在着很大的差异.首先,Reuters 数据样本集在特征空间中可能不是线性可分的;其次,Reuters 数据样本集在特征空间中的分布不一定像 Tutorial 数据集那样相对集中,而可能相当松散.在这种不利的数据分布条件下,PTSVM 算法的训练过程就可能产生较多被错误标注的无标签样本,因为 PTSVM 算法采用固定的较大的影响因子 C^*,所以这种错误可能会影响后继的训练过程,并导致算法性能的下降.虽然标签重置法具有一定的差错修复的功能,但是标签重置法的能力有限,不足以改善较大规模误分的情况.我们认为,不合适的数据分布是不利于 PTSVM 算法的直推式学习方式并导致算法性能下降的主要原因.也就是说,PTSVM 算法更适用于可分性较好或者同类数据相对集中的样本集.

一种可能的改进是在 PTSVM 算法中结合使用 TSVM 算法中的影响因子递增法.从上文可以看出,较大的影响因子意味着对后继训练有较大的影响能力,但同时也增大了错误标注的风险.因此,可以考虑用较小的影响因子来抑制训练过程中尚不确定的已标注的无标签样本对后继训练的影响能力,从而控制训练误差的传递和累积.但是,注意到较小的影响因子和 PTSVM 算法渐进标注的思想具有一定的矛盾,所以,如果采用这种方法,就

© 中国科学院软件研究所 http://www.jos.org.cn

应谨慎地处理这种矛盾,试图寻找一个较好的折衷,根据已标注样本的标注可信度对其赋予不同的影响因子.

另一种可能的改进是核函数的合理使用.通过引入适当的核函数,能够把原本线性不可分的数据映射成相应的核空间中的线性可分的数据.可以很自然地想到,对于某些分布特征不太好或者分布特征未知的数据集,如果能够通过核函数映射来改善它的数据分布状况,在映射空间中应用 PTSVM 算法来进行直推式学习,则可以有效地改进学习算法的性能.在这样一种考虑下,学习算法中核函数的选择就是非常重要的了.核函数的设计和选择将会是这一改进算法的核心内容.

5 结束语

本文在简介了支持向量机理论和直推式学习的基础上,提出了一种渐进直推式支持向量机学习算法 PTSVM.该算法可以较好地适应各种不同的训练样本分布,实现了较一般意义上的直推式学习.实验结果表明,PTSVM 算法在各种样本分布情况下都取得了较好的分类效果.

直推式学习是一个较新的研究领域,还有很多有意义的课题值得进一步挖掘和研究.例如,本文的算法找到的解和全局最优解尚有多少差距?全局数据分布特征对于不同的直推式学习方法有何影响,如何根据不同的数据分布特性、结合不同的算法思想对 PTSVM 算法作进一步的改进等等.

References:

[1] Vapnik V. The Nature of Statistical Learning Theory. New York: Springer-Verlag, 1995.

[2] Stitson MO, Weston JAE, Gammerman A, Vovk V, Vapnik V. Theory of support vector machines. Technical Report, CSD-TR-96-17, Computational Intelligence Group, Royal Holloway: University of London, 1996.

[3] Cortes C, Vapnik V. Support vector networks. Machine Learning, 1995,20:273~297.

[4] Vapnik V. Statistical Learning Theory. John Wiley and Sons, 1998.

[5] Gammerman A, Vapnik V, Vowk V. Learning by transduction. In: Proceedings of the 14th Conference on Uncertainty in Artificial Intelligence. Wisconsin, 1998. 148~156.

[6] Joachims T. Transductive inference for text classification using support vector machines. In: Proceedings of the 16th International Conference on Machine Learning (ICML). San Francisco: Morgan Kaufmann Publishers, 1999. 200~209.

[7] Boser BE, Guyon IM, Vapnik VN. A training algorithm for optimal margin classifiers. In: Haussler D, ed. Proceedings of the 5th Annual ACM Workshop on Computational Learning Theory. Pittsburgh, PA: ACM Press, 1992. 144~152.

[8] Burges CJC. Simplified support vector decision rules. In: Saitta L, ed. Proceedings of the 13th International Conference on Machine Learning. San Mateo, CA: Morgan Kaufmann Publishers, 1996. 71~77.

[9] Osuna E, Freund R, Girosi F. An improved training algorithm for support vector machines. In: Proceedings of the IEEE NNSP'97. Amelia Island, FL, 1997. 276~285.

[10] Joachims T. Making large-scale SVM learning practical. In: Schölkopf, Burges C, Smola A, eds. Advances in Kernel Methods——Support Vector Learning B. MIT Press, 1999.

[11] Platt J. Sequential minimal optimization: a fast algorithm for training support vector machines. Technical Report, MSR-TR-98-14, Microsoft Research, 1998.

[12] Bennett K. Combining support vector and mathematical programming methods for classification. In: Scholkopf B, Burges C, Smola A, eds. Advances in Kernel Methods——Support Vector Learning. MIT Press, 1998.

[13] Branson K. A naive Bayes classifier using transductive inference for text classification. 2001. http://www-cse.ucsd.edu/.

[14] Blum A, Mitchell T. Combining labeled and unlabeled data with co-training. In: Annual Conference on Computational Learning Theory (COLT'98). 1998. 92~100.

[15] Nigam K, McCallum A, Mitchell T. Learning to classify text from labeled and unlabeled documents. In: Proceedings of the AAAI'98. 1998.

[16] Joachims T. SVMlight. 2001. http://www-ai.cs.uni-dortmund.de/SOFTWARE/SVM_LIGHT/svm_light.eng.html.

[17] Cauwenberghs G, Poggio T. Incremental and decremental support vector machine learning. In: Advances Neural Information Processing Systems (NIPS 2000). Cambridge, MA: MIT Press, 2001.

© 中国科学院软件研究所 http://www.jos.org.cn