维普资讯 http://www.cqvip.com

Boosting 家族 Boost-by-majority 系列代表算法

涂承胜[1] 刁力力[2] 鲁明羽[2,3] 陆玉昌[2]
(重庆三峡学院计算机科学系 重庆万州404000)[1]
(清华大学计算机科学技术系 智能技术与系统国家重点实验室 北京100084)[2]
(烟台大学计算机学院 烟台264005)[3]

The Typical Algorithm of Boost-by-Majority Series in Boosting Family

TU Cheng-Sheng[1] DIAO Li-Li[2] LU Ming-Yu[2,3] LU Yu-Chang[2]
(Dept. of computer science ChongQing three gorges college ChongQing WanZhou 404000)[1]
(Computer Science and Technology Dept., TsingHua University The State Key Larboratory of Intelligent Technology and System BeiJing China, 100084)[2]
(Computer College of Yan Tai University Yantai 264005)[3]

Abstract Boosting is one of the most representational ensemble prediction methods. It can be divided into two series: Boost-by-majority and Adaboost. This paper briefly introduces the research status of Boosting and one of its serials-Boost-by-majority, analyzes the typical algorithms of Boost-by-majority.

Keywords Data mining, Machine learning, Combining prediction, Algorithms

1 引言

Boosting 由 Freund 和 Schapire 于1990年提出[4]，是提高预测学习系统预测能力的有效工具，也是组合学习中最具代表性的方法，其代表算法可分为 Boost-by-majority 和 AdaBoost 两个系列。Boosting 操纵训练例子以产生多个假设，从而建立通过投票结合的预测器集合。Boosting 在训练例子上维护一套概率分布。Boost-by-majority 通过在每一回迭代中重取样(resampling)生成不同的训练集，AdaBoost 在每个例子上调整这种概率分布。具体的学习算法用于产生成员分类器，成员分类器在训练例子上的错误率被计算出来并以此在训练例子上调整概率分布。权重改变的作用是在被误分的例子上放置更多的权重，在分类正确的例子上减少其权重。通过单个分类器的加权投票建立最终分类器，每个分类器按其在训练集上的精度而加权[6]。Boosting 方法一般用于提高不稳定的学习器的性能。

Kearns 和 Valiant 首先提出：在 Valiant 的 PAC 模型[1]中，一个"弱"学习算法是否能被"提升"为一个具有任意精度的"强"学习算法[2,3]？1989年 Schapire 提出了第一个可证明的多项式时间 Boosting 算法[4]，对上述问题作了肯定回答。之后，Freund 设计了一个更高效的通过重采样或过滤运作的 Boost-by-majority 算法[5]，在一定程度上该算法是优化的，实践上却有一些缺陷。野点(Outliers)是训练样本中被标错的样本。AdaBoost 对野点的识别能力强，因为这些样本通过学习通常会得到较高的权值。但野点数目过多时，过分强调"困难"实例将有损 AdaBoost 的性能。为此，1998年 Friedmand 等提出了被称之为"Gentle AdaBoost"的 AdaBoost 变种算法[10]，它较少地强调野点。1999年 Freund 介绍了 Boost-by-majority 算法的一个自适应扩展版本 BrownBoost，该算法采用了不强调那些太"困难"以至于无法正确分类的野点[11]的方法。

本文集中介绍了 Boost-by-majority 系列的典型算法。限于篇幅，AdaBoost 系列典型算法另文介绍[13]。Boost-by-majority 的着眼点在于从大量数据集中取出少量数据作为学习的训练样本，以解决分类学习问题。

涂承胜 讲师，清华大学访问学者。刁力力 博士研究生。鲁明羽 博士研究生。陆玉昌 教授。

缺点，从而为软件需求分析人员改进和提高软件需求规格说明质量提供指导。

这种模糊度量方法可以推广至整个软件项目管理质量度量，将软件需求质量、进度、资源和费用、软件质量等作为软件项目管理质量属性，进而采用多级评价对软件项目管理质量进行综合量化评价。

下一步工作是以此模糊综合评价方法为基础，结合软件需求规格说明的自动生成、自动验证，开发一套软件需求分析综合管理系统，实现软件需求规格说明文档的自动生成、自动验证以及质量模糊评价等功能，进一步提高软件需求质量评价的自动化程度以及评价的准确性和可信度，从而提高需求分析的质量以及需求分析的自动化。

参考文献

1 The Standish Group. The Scope of Software Development Project Failures. Dennis, MA: The Standish Group. 1995
2 Boehm B W. Software Engineering Economics. Englewood Cliffs, NJ: Prentice-Hall. 1981
3 [美]Wiegers K E 著，陆丽娜等译. 软件需求. 北京：机械工业出版社，2000
4 李凡. 模糊信息处理系统. 北京：北京大学出版社，1998
5 IEEE Std 830-1998. IEEE Recommended Practice for Software Requirements Specifications. Los Alamitos, CA: IEEE Computer Society Press, 1998
6 Leffingwell D, Widrig D. Managing Software Requirements: A Unified Approach. Addison Wesley Longman, Inc.
7 Davis A M. Software Requirements: Objects, Functions, and States. Englewood Cliffs, NJ: Prentice-Hall, 1993

维普资讯 http://www.cqvip.com

2 Boost-by-majority 系列算法

该学习方法是基于与分布无关的概念学习的最小框架提出的。"概念"是在某域 X 上的二值映射，用字母 c 表示。$c(x)$ 表示实例 x 的标签。概念类 C 是概念的集合。学习器的任务是学习 c 的近似。学习器预先知道概念在某已知类 C 中，但并不知道具体是其中哪一个。学习器假设调用 EX 得到样本源。每调用一次过程 EX，按照某个固定而未知的分布 D，一个实例将随机地从 X 中被独立取出，具体返回的是选择的实例 x 及其相应标签 $c(x)$。学习算法调用 EX 多次并输出一假设 h。该假设是一个接受实例 $\chi\in X$ 产生二值输出的决策规则。学习算法 A 有一致样本复杂度 $m(E,\delta)$，如果它满足条件：对所有的 $0<E,\delta<1$，所有 D 和 $c\in C$，在算法 A 接收到参数 E 和 δ（作为输入）后，最多调用 EX 过程 $m(E,\delta)$ 次，且以 $1-\delta$ 的概率输出 D 下精度为 E 的 c 的近似。此处，用 WeakLearn 表示要提升其性能的学习算法。WeakLearn 产生的有保证精度的假设称之为弱假设。假设存在实值 $0\leqslant E_0<1/2$ 和 $0\leqslant\delta_0<1$，使有 m_0 个带标签的样本的 WeakLearn 在训练例子的分布以至少 $1-\delta_0$ 的概率产生错误率最多为 E_0 的弱假设。下述 Boosting 算法可以以任意高的可靠性 $1-\delta$ 产生有任意精度 E 的假设。定义 $\gamma=1/2-E_0$，$\lambda=1-\delta_0$。这两个参数用于测量该学习算法与完全无用算法（即准确度只有50%的算法）之间的差距。

2.1 B_{samp} 算法

输入：EX，WeakLearn，γ，m。

输出：在大小为 m 的某随机样本上一致的假设。

1. 调用 EX 过程 m 次以产生某样本 $S=\{(x_1,l_1),\cdots,(x_m,l_m)\}$。S 中的每个样本 (x_j,l_j) 都对应于一个权重 w_j 和一个计数器 r_j。开始时所有权重都是 1/m，所有计数器都是 0。

2. 寻找一个（小的）k，它满足：$\sum_{i=\lceil k/2\rceil}^{k}\binom{k}{i}(\frac{1}{2}-\gamma)^i(\frac{1}{2}+\gamma)^{k-i}<\frac{1}{m}$。

   对 $i=1,\cdots,k$ 重复下述步骤：1）对 $L=1,\cdots,(1/\lambda)ln(2k/\delta)$ 重复下述步骤或直到找到一弱假设：①调用 WeakLearn（其中须调用过程 FiltEX 作为其样本源）并保存返回的假设 h_i。②在 $h_i(\boxtimes_j)\neq l_j$ 的样本上权重求和。如果该和小于 $1/2-\gamma$，则认为 h_i 是一个弱假设并退出循环。2）在 $h_i(\boxtimes_j)=l_j$ 的样本上 R_j 增 1。3）按 $w_j=\alpha_j^{r_j}$ 修改样本权重，其中 $\alpha_r^i=\binom{k-i-1}{\lfloor k/2\rfloor-r}(\frac{1}{2}+\gamma)^{\lfloor\frac{k}{2}\rfloor-r}(\frac{1}{2}-\gamma)^{\lceil\frac{k}{2}\rceil-i-1+r}$。4）通过用 $\sum_{j=1}^{m}w_j$ 除每个权重来得到归一化的权重。

3. 返回在 $h_1,\cdots,h_k$ 上的多数投票结果 h_M 作为最终假设。

子过程 FiltEX：①在 $0\leqslant x<1$ 范围内一致地随机选择一实数 x。②对 $\sum_{i=1}^{j-1}w_i\leqslant x<\sum_{i=1}^{j}w_i$ 按索引号 j 执行二值搜索（$\sum_{i=1}^{0}w_i$ 定义为 0）。③返回例子 (x_j,l_j)。

如果样本是大小为 m 的有限集，要使假设在所有样本上都正确，就需所有假设在样本上的分布均有小于 1/m 的错误率。为此，该算法在训练样本上产生不同的分布，每次调用 WeakLearn 都产生一个按给定分布其错误率小于 $1-\gamma$ 的弱假设。每个不同的分布迫使 WeakLearn 产生其错误基于不同样本点的弱假设。Boosting 算法的目标是以这样一种方式控制这些错误的位置：少量弱假设产生之后，在所有弱假设上的多数投票将在每个点上给出正确标签。即对 S 中的每个点，半数以上的弱假设给这个点以正确的标签。

2.2 B_{Filt} 算法

输入：EX，WeakLearn，γ，λ，E，δ。

输出：至少 $1-\delta$ 的概率其错误小于 E 的假设 h_M。

1. 找一个（小的）k，它满足：$\sum_{i=\lceil k/2\rceil}^{k}\binom{k}{i}(\frac{1}{2}-\frac{\gamma}{2})^i(\frac{1}{2}-\frac{\gamma}{2})^{k-i}<\epsilon^2$。对 $i=0,\cdots,k-1$ 重复下述步骤（每次迭代开始前，设置两个计数器 #accept 和 #reject 为 0）：①调 B_{rnd} 过程。它通过 FiltEX 在线选择样本并提供给 WeakLearn 学习。如果 B_{Rnd} 未失败，则产生一假设 h_{i+1} 其错误率至少以 $1-\delta/2k$ 的概率小于 $1/2-\gamma/2$。②如果 B_{Rnd} 失败，则定义 h_{i+1} 是一个产生随机预测的假设。

2. 返回在 $h_1,\cdots,h_k$ 上的多数投票结果 h_M 作为最终假设。

[子过程 FiltEX]：①调 EX，得到一带标签的例子 (χ,L)。②若 $i=0$ 则接受该例子并返回，否则继续第③步。③设 r 为使 $h_j(x)=l$ 的索引号 $1\leqslant j\leqslant i$，并计算：

$\alpha_r^i=\binom{k-i-1}{\lfloor k/2\rfloor-r}(1/2+\gamma/2)^{\lfloor k/2\rfloor-r}(1/2-\gamma/2)^{\lceil k/2\rceil-i-1+r}$；

$\alpha_{\max}^i=\max_{0\leqslant r\leqslant i}\alpha_r^i$。

④在 $0\leqslant x<1$ 范围内一致地随机选择一实数 x。⑤若 $x<\alpha_r^i/\alpha_{\max}^i$ 则接受该例，并作为结果返回。否则，拒绝之并跳回第①步。每种情况下都相应地修改 #accept 和 #reject。

B_{Rnd} 的失败条件：$\#accept+\#reject>\frac{k\gamma\alpha_{\max}^i}{\epsilon(1-\epsilon)}max(\#accept,4ln\frac{8k^2\gamma\alpha_{max}^i}{\delta\epsilon(1-\epsilon)})$。

B_{Samp} 的空间复杂度来源于训练例子的存储。而在内存中存储所有例子代价很大。实际上，为找到错误率小于 ϵ 的假设，$O(\frac{1}{\epsilon}(log(\frac{1}{\epsilon}))^2)$ 个训练样本中仅有 $O(\log(1/\epsilon))$ 个样本被 WeakLearn 使用。B_{Filt} 提供了 B_{Samp} 的一个在线风格的版本，它将空间复杂度减少到 $O(\log(1/\epsilon))$，从而改善了空间复杂度。它对 EX 得到的每个例子计算权重并作出随机决策以接受或抛弃之，即对其进行"过滤"。

2.3 BrownBoost 算法

输入：N 个带标签的例子组成的集合 $S=(x_1,y_1),\cdots,(x_N,y_N)$。其中 $x_i\in R^d$ 且 $y_i\in\{-1,+1\}$，一个弱学习算法 WeakLearn，一个正实值参数 c，一个用于避免退化情况的小常数 $v(>0)$。

预测值：每个例子都与一个实值 margin 相联系。在第 i 次迭代中例子 (x,y) 的 margin 用 $r_i(x,y)$ 表示。所有样本的初始预测值是 0，即：$r_1(x,y)=0$。

初始化："剩余时间"$s1=c$。

Do for i=1,2,…

①对每个例子赋正权重 $W_i(x,y)=e^{-(r_i(x,y)+s_i)^2/c}$。②用通过归一化 $W_i(x,y)$ 定义的分布调用 WeakLearn，并从中获取假设 $h_i(x)$：$\sum_{(x,y)}W_i(x,y)h_i(x)y=\gamma_i>0$。③设 γ，α 和 t 为遵循下述微分方程的实值标量变量：

$$\frac{dt}{d\alpha}=\gamma=\frac{\sum_{(x,y)\in T}\exp(-\frac{1}{c}(r_i(x,y)+\alpha h_i(x)y+s_i-t)^2)h_i(x)y}{\sum_{(x,y)\in T}\exp(-\frac{1}{c}(r_i(x,y)+\alpha h_i(x)y+s_i-t)^2)}$$

其中

维普资讯 http://www.cqvip.com

$r_t(x,y)$,$h_t(x)y$ 和 s_t 此处都是常数。给定边界条件 $t=0$,$\alpha=0$,求解该方程组,得 $t_t=t^*>0$和 $\alpha_t=\alpha^*$,使得要么 $\gamma^*\leqslant v$,要么 $t^*=s_t$。④修改每个例子的预测值为:$r_{t+1}(x,y)=r_t(x,y)+\alpha_t h_t(x)y$。⑤修改"剩余时间"$s_{t+1}=s_t-t_t$。

Until $s_{t+1}\leqslant 0$

输出:最终假设:

$$\begin{cases} \text{若 } p(x)\in[-1,+1],\text{则 } p(x)=erf\left(\dfrac{\sum_{t=1}^{N}\alpha_t h_t(x)}{\sqrt{c}}\right) \\ \text{若 } p(x)\in\{-1,+1\},\text{则 } p(x)=sign\left(\sum_{t=1}^{N}\alpha_t h_t(x)\right) \end{cases}$$

其中 erf 是"错误函数":$erf(a)=\frac{2}{\pi}\int_0^a e^{-x^2}dx$。

虽然 Boost-by-majority(简称 BBM)算法有优化特性,但它不是自适应的,因此很少得到实际应用。BrownBoost 算法是 BBM 算法的一个自适应扩展版本,对实际的学习问题是有用的。

AdaBoost 的成功是无可争辩的,但该算法很容易受到噪声的影响,因为它倾向于给噪声加较其它样本高的权重,导致后面的迭代步中产生的假设对噪声的过分匹配。AdaBoost 中,对一给定例子(x,y)如果大多数假设的预测是不正确的,那么将会有大负 margin 值,同时例子上的权重会无限制地快速增长。为避免这种情况,一些学者建议使用增长比 e^x 慢得多的 margin 的函数作为加权方案,如"Gentle-Boost"算法。然而,所有建议均无在 PAC 框架中定义的正规 Boosting 属性。上述与 AdaBoost 有关的实验性问题迫使我们重新审视 BBM。该算法赋给各例子的权重是 margin 的函数(BBM 产生其中所有假设都有相同权值的多数规则时,margin 是正确弱假设数目和迭代数目的线性组合)。然而,反映 magin 和权重关系的函数形式与其在 AdaBoost 中的形式有很大的差异。其原因是:BBM 是一个被优化的用于在预先设定的 Boosting 迭代步数内最小化训练错误的算法。该算法在接近预定的终点时,有大负 margin 值的例子将变得越来越不可能最终被正确标记。因此,放弃这些例子集中精力于具有小负 margin 值的例子上的 BrownBoost 算法效果应当更好。BBM 需要预先指定弱学习器上错误的上边界 $1/2-\gamma$ 和一"目标"错误 $\varepsilon>0$。BrownBoost 可以省去参数 γ。设置 ε 为0将使 BrownBoost 变为 AdaBoost。因此,可以说 AdaBoost 是 BrownBoost 的一种特殊情况。这就和 AdaBoost 在大噪声数据集性能很差这一事实在直观上相吻合。

参 考 文 献

1 Valiant L G. A theory of the learnable. Communications of the ACM, 1984, 27(11): 1134～1142
2 Kearns M J, Vazirani L G. Learning Boolean formulae or finite automata is as hard as factoring: [Technical Report TR-14-88]. Harvard University Aiken Computation Laboratory, Aug. 1988
3 Kearns M J, Vazirani L G. Cryptographic limitations on learning Boolean formulae and finite automata. Journal of the Association for Computing Machinery, 1994, 41(1): 67～95
4 Schapire R E. The strength of weak learnability. Machine Learning, 1990, 5(2): 197～227
5 Freund Y. Boosting a weak learning algorithm by majority. Information and computation, 1995, 121(2): 256～285
6 Freund Y, Schapire R E. A decision-theoretic generalization of online learning and an application to boosting. Journal of Computer and System Science, 1997, 55(1): 119～139
7 Dietterich T G, Bakiri G. Solving multiclass learning problems via error-correcting output codes. Journal of Artificial Intelligence Research, 1995, 2: 263～286
8 Schapire R E, Singer Y. Using output codes to boost multiclass learning problems. In: Machine Learning: Proc. of the Fourteenth Intl. Conf. 1997. 313～321
9 Schapire R E, Singer Y. Improved boosting algorithms using confidence-related predictions. In: Proc. of the eleventh Annual Conf. on Computational Learning Theory, 1998. 80～91
10 Friedman J, Hastie T, Tibshirani R. Additive logistic regression: a statistical view of boosting: [Technical Report]. 1998
11 Freund Y. An adaptive version of the boost by majority algorithm. In: Proc. of the Twelfth Annual Conf. on Computational Learning Theory, 1999
12 刁力力,等. 数据挖掘与组合学习. 计算机科学[J], 2001, 28(7)
13 涂承胜,刁力力,陆玉昌. BoostingAdaBoost 系列代表算法. 计算机科学, 2003, 30(3)

(上接第120页)

基于列表的线性访问接口;通过标准的访问接口,任何符合该接口的高层应用模块可从中检索需要的信息。

程序模型的内聚度高,可扩充性强。程序模型是以程序中的对象为基本的单元,以域的层次来划分对象,以对象的型为主要的组成部分来构建,它还特别包含了模板的信息。模型中使用指针相互关联,使得各个部分之间既相对独立又相互依赖,形成一个紧密的整体。同时,指针的使用可以使得增加或减少单元的操作非常方便,对整个模型的影响比较小。当针对不同的语言构建模型时,可以根据不同语言的语法特点相应修改模型中的成员。

结论 基于对象、域和型的层次式程序表示模型是一种能够详尽描述程序结构的一种中间表示形式,它把程序中所有对象的作用域、类型等信息紧密地关联起来。该模型采用了层次式的结构,有利于通过计算机自动地分析和理解程序的结构;并且它提供了标准的访问接口,任何符合该接口的应用模块均可从中抽取所需要的相关信息。这样,在该模型的基础上,我们可以构筑很多有意义的应用。

参 考 文 献

1 Eckel B. C++编程思想. 北京:机械工业出版社, 2000
2 Pressman R S. 软件工程--实践者的研究方法(第四版). 北京:机械工业出版社, 1999
3 刘超,等. 可视化面向对象建模技术. 北京:北京航空航天大学出版社, 1999
4 张幸儿. 计算机编译原理. 北京:科学出版社, 1999
5 Binder R V. 面向对象系统的测试. 北京:人民邮电出版社, 2001
6 Hughes C. 掌握标准C++类. 北京:人民邮电出版社, 2000
7 Pratt T W. Programming Languages: Design and Inplementation Prentice-Hall International, Inc 1996
8 Norman R J. Object-Oriented Systems Analysis and Design Prentice-Hall International, Inc 1996
9 Martin R C. Pattern Languages of Program Design Addison Wesley Longman, Inc. 1998
10 Peter B J. Introduction to Compiling Techniques A First Course using ANSI C, LEX and YACC. McGRAW—HILL Book Company, 1990
11 Aho A V, Sethi R, Ullman J D. Compilers: Priciples, Techniques, and Tools. Addison-Wesley, 1986
12 赵洋,蔡志旻,潘金贵. 基于EPOM的程序可视化表示系统的设计与实现. 上海:计算机工程(已录用于2002年第7期)