第22卷第5期 统计与信息论坛 2007年9月
Vol.22 No.5 Statistics & Information Forum Sept.,2007

【国际统计动态】

机器学习及其相关算法综述

陈 凯[1],朱 钰[1,2]

(1. 中国人民大学 统计学院,北京 100872; 2. 西安财经学院 统计学院,陕西 西安 710061)

摘要:自从计算机被发明以来,人们就想知道它能不能学习。机器学习从本质上是一个多学科的领域。它吸取了人工智能、概率统计、计算复杂性理论、控制论、信息论、哲学、生理学、神经生物学等学科的成果。文章主要从统计学习基础的角度对机器学习的发展历程以及一些相关的常用算法进行了简要的回顾和介绍。

关键词:机器学习;有指导学习;无指导学习;半指导学习

中图分类号:TP181 **文献标识码**:A **文章编号**:1007-3116(2007)05-0105-08

一、引 言

机器学习的研究主旨是使用计算机模拟人类的学习活动,它是研究计算机识别现有知识、获取新知识、不断改善性能和实现自身完善的方法。这里的学习意味着从数据中学习,它包括有指导学习(Supervised Learning)、无指导学习(Unsupervised Learning)和半指导学习(Semi-Supervised Learning)三种类别。

有指导学习,之所以称它为"有指导的",是指有结果度量(Outcome Measurement)的指导学习过程。我们希望根据一组特征(Features)对结果度量进行预测,例如根据某病人的饮食习惯和血糖、血脂值来预测糖尿病是否会发作。我们通过学习已知数据集的特征和结果度量建立起预测模型来预测并度量未知数据的特征和结果。这里的结果度量一般有定量的(Quantitative)(例如身高、体重)和定性的(Qualitative)(例如性别)两种,分别对应于统计学中的回归(Regression)和分类(Classification)问题。常见的有指导学习包括:决策树、Boosting与Bagging算法、人工神经网络和支持向量机等。

在无指导学习中,只能观察特征,没有结果度量。此时只能利用从总体中给出的样本信息对总体作出某些推断以及描述数据是如何组织或聚类的。它并不需要某个目标变量和训练数据集,例如,聚类分析或关联规则分析等。

半指导学习是近年来机器学习中一个备受瞩目的内容:已得的观察量中一部分是经由指导者鉴认并加上了标识的数据,称之为已标识数据;另一部分观察量由于种种原因未能标识,被称为未标识数据。需要解决的是如何利用这些观察量(包括已标识数据和未标识数据)及相关的知识对未标识的观察量的标识做出适当合理的推断。解决这类问题常用方法是采用归纳-演绎式的两步骤路径,即先利用已标识数据去分析并指出适当的一般性的规律,再利用此规律去推断得出有关未标识数据的标识。这里,前一步是从特殊得到一般结论的归纳步,后一步则是将一般规律用于特殊情况的演绎步。[1]这里的关键是如何选择出合适的无标识样本并进行标记。值得注意的是,现有的半指导学习方法的性能通常不太稳定,而半指导学习技术在什么样的条件下能够有效地改善学习性能,仍然是一个未决问题。比较有代表的做法有:利用Naive Bayes这样的生成式模型(Generative Model),通过EM算法来进行标记估计和参数估计;[2]通过转导推断(Transductive Inference)来优化特定测试集上的性能;[3]利用独立冗

收稿日期:2007-07-03

基金项目:国家自然科学基金重点项目(10431010);教育部重点基地重大项目(05JJD910001);中国人民大学应用统计中心项目

作者简介:陈 凯(1978-),男,安徽巢湖人,博士生,研究方向:统计模型;

朱 钰(1964-),男,山西运城人,副教授,博士生,研究方向:应用数理统计学。

余的属性集来进行协同训练等。[4]

二、机器学习发展历程

机器学习是人工智能研究较为年轻的分支，它的发展过程大体上分为四个时期。[5]

第一阶段是20世纪50年代中叶到60年代中叶，属于热烈时期。在这个时期，所研究的是“没有知识”的学习，即“无知”学习。其研究目标是各类自组织系统和自适应系统，其主要研究方法是不断修改系统的控制参数和改进系统的执行能力，不涉及与具体任务有关的知识。本阶段的代表性工作是：塞缪尔(Samuel)的下棋程序。但这种学习的结果远不能满足人们对机器学习系统的期望。

第二阶段是在60年代中叶到70年代中叶，被称为机器学习的冷静时期。本阶段的研究目标是模拟人类的概念学习过程，并采用逻辑结构或图结构作为机器内部描述。本阶段的代表性工作有温斯顿(Winston)的结构学习系统和海斯罗思(Hayes－Roth)等的基本逻辑的归纳学习系统。

第三阶段从20世纪70年代中叶到80年代中叶，称为复兴时期。在此期间，人们从学习单个概念扩展到学习多个概念，探索不同的学习策略和方法，且在本阶段已开始把学习系统与各种应用结合起来，并取得很大的成功，促进机器学习的发展。1980年，在美国的卡内基—梅隆(CMU)召开了第一届机器学习国际研讨会，标志着机器学习研究已在全世界兴起。

当前机器学习围绕三个主要研究方向进行：

1. 面向任务：在预定的一些任务中，分析和开发学习系统，以便改善完成任务的水平，这是专家系统研究中提出的研究问题；

2. 认识模拟：主要研究人类学习过程及其计算机的行为模拟，这是从心理学角度研究的问题；

3. 理论分析研究：从理论上探讨各种可能学习方法的空间和独立于应用领域之外的各种算法。

这三个研究方向各有自己的研究目标，每一个方向的进展都会促进另一个方向的研究。这三个方面的研究都将促进各方面问题和学习基本概念的交叉结合，推动了整个机器学习的研究。

三、八种常用算法简介

(一)决策树算法

决策树可看作一个树状预测模型，它通过把实例从根节点排列到某个叶子节点来分类实例，叶子节点即为实例所属的分类。决策树的核心问题是选择分裂属性和决策树的剪枝。决策树的算法有很多，有ID3、C4.5、CART等等。这些算法均采用自顶向下的贪婪算法，每个节点选择分类效果最好的属性将节点分裂为2个或多个子结点，继续这一过程直到这棵树能准确地分类训练集，或所有属性都已被使用过。下面简单介绍最常用的决策树算法—分类回归树(CART)[6]。

分类回归树(CART)是机器学习中的一种分类和回归算法。设训练样本集 $L=\{x_1, x_2, \cdots, x_n, Y\}$。其中，$x_i(i=1,2,\cdots,n)$称为属性向量；$Y$ 称为标签向量或类别向量。当 Y 是有序的数量值时，称为回归树；当 Y 是离散值时，称为分类树。

在树的根节点 t_1 处，搜索问题集(数据集合空间)，找到使得下一代子节点中数据集的非纯度下降最大的最优分裂变量和相应的分裂阈值。在这里非纯度指标用Gini指数来衡量，它定义为：

$$i(t)=\sum_{i\neq j}p(i/t)p(j/t)=1-\sum_{j}[p(j/t)]^2 \quad (1)$$

其中，$i(t)$ 是节点 t 的Gini指数，$p(i/t)$ 表示在节点 t 中属于 i 类的样本所占的比例，$p(j/t)$ 是节点 t 中属于 j 类的样本所占的比例。用该分裂变量和分裂阈值把根节点 t_1 分裂成 t_2 和 t_3，如果在某个节点 t_i 处，不可能再有进一步非纯度的显著降低，则该节点 t_i 成为叶结点，否则继续寻找它的最优分裂变量和分裂阈值进行分裂。

对于分类问题，当叶节点中只有一个类，那么这个类就作为叶节点所属的类，若节点中有多个类中的样本存在，根据叶节点中样本最多的那个类来确定节点所属的类别；对于回归问题，则取其数量值的平均值。

很明显，一棵很大的树可能过分拟合数据，但较小的树又可能无法捕获重要的结构。树的最佳大小是控制模型复杂性的调整参数，它应该由数据自适应的选择。一种可取的策略是增长一棵较大的树 T_0，仅当达到最小节点大小(比如5)时才停止分裂过程。然后利用剪枝策略和5折或10折交叉验证相结合的方法来修剪这棵树，从而将一些噪声和干扰数据排除，获得最优树。

(二)随机森林算法

从上面决策树的介绍我们可以看到，一般用选择分裂属性和剪枝来控制树的生成，但是当数据中噪声或分裂属性过多时，它们也解决不了树不平衡

和对训练集过度拟合的问题。

最新的研究表明,构造多分类器或回归器的集成可以提高分类或预测的精度。而随机森林就是一个由多个决策树$\{h(X,\Theta_k)\}$组成的多分类器或多回归器,其中$\{\Theta_k\}$是相互独立且同分布的随机向量。每一棵决策树都会对输入向量X进行投票,给出一个得分,对于分类问题来说,最终投票最多的那一类就是输入向量X的最终类标签;对于回归问题,可以对得分取平均数。

随机森林算法(RFA)是Leo Breiman提出的一种利用多个树分类器进行分类和预测的方法[7]。随机森林算法可以用于处理回归、分类、聚类以及生存分析等问题,当用于分类或回归问题时,它的主要思想是通过自助法重采样,生成很多个树回归器或分类器。其步骤如下:

假设现有N个训练样本$(x_i,y_i)_{i=1}^N$,其中x_i是第i个样本,它包含M个解释变量,y_i是x_i的对应的响应变量。

通过自助法重抽样,从原始训练数据中生成k个自助样本集。每个自助样本集形成一棵分类或回归树。根据生成的多棵树对新的数据进行预测,分类结果按投票最多的作为最终类标签;回归结果按每棵树得出的结果进行简单平均或按照训练集得出的每棵树预测效果的好坏(比如按$1/MSE_i$)进行加权平均而定。

我们通常将全体样本中不在每次抽样生成的自助样本中的剩余样本称为袋外数据(out - of - bag, OOB)。据经验得知,每次抽样后大约剩余1/3的袋外数据,将每次的预测结果进行汇总可以得到袋外数据的估计误差,我们常常将它和测试样本的估计误差相结合用于评估组合树学习器的拟合和预测精度。

一般来说,随机森林的广义误差(Generalization Error)上界可以根据两个参数推导出来:森林中每棵决策树的预测精度和这些树之间的相互依赖程度$\bar{\rho}$。当随机森林中每棵树的相关程度$\bar{\rho}$增大时,随机森林的广义误差上界就增大;当每棵决策树的预测精度提高时,随机森林的广义误差上界就下降。

(三)人工神经网络(Artificial Neural Networks——ANN)算法

人工神经网络提供了一种普遍而且实用的方法,来从样例中学习值为实数、离散或向量的函数。ANN学习对于训练数据中的拟合效果很好,且已经成功地涉及到医学、生理学、哲学、信息学、计算机科学等众多学科领域,这些领域互相结合、相互渗透并相互推动。不同领域的科学家从各自学科的特点出发,提出问题并进行了研究。ANN的研究始于1943年,心理学家W. Mcculloch和数理逻辑学家W. Pitts首先提出了神经元的数学模型。此模型直接影响着这一领域研究的进展。1948年,冯·诺依曼在研究中提出了以简单神经元构成的再生自动机网络结构;20世纪50年代末,F. Rosenblatt设计制作了"感知机",它是一种多层的神经网络,这项工作首次把人工神经网络的研究从理论探讨付诸工程实践;60年代初期,Widrow提出了自适应线性元件网络,这是一种连续取值的线性加权求和阈值网络,在此基础上发展了非线性多层自适应网络。这些实际上就是一种ANN模型;80年代初期,美国物理学家Hopfield发表了两篇关于ANN研究的论文,引起了巨大的反响。人们重新认识到神经网络的威力以及付诸应用的现实性。随即,研究人员围绕着Hopfield提出的方法展开了进一步的研究工作,形成了80年代中期以来ANN的研究热潮[8]。

人工神经网络的研究在一定程度上受到了生物学的启发,因为生物的学习系统是由相互连接的神经元(Neuron)组成的异常复杂的网络。而人工神经网络与此大体相似,它是由一系列简单单元相互密集连接构成,其中每一个单元有一定数量的实值输入(可能是其他单元的输出),并产生单一的实数值输出(可能成为其他很多单元的输入)。

在ANN的研究中提出了很多模型,它们之间的差异主要表现在研究途径、网络结构、运行方式、学习算法及其应用上。常见的ANN模型有:多层前向神经网络MLFN、自组织神经网络—SOM和ART、Hopfield神经网络、模糊神经网络FNN等。

人工神经网络算法的重点是构造阈值逻辑单元,一个值逻辑单元是一个对象,它可以输入一组加权系数的量,对它们进行求和,如果这个和达到或者超过了某个阈值,输出一个量。如有输入值X_1,$X_2,\cdots,X_n$和它们的权系数:$W_1,W_2,\cdots,W_n$,求和计算出的$X_i\times W_i$,产生了激发层$a=(X_1\times W_1)+(X_2\times W_2)+\cdots+(X_i\times W_i)+\cdots+(X_n\times W_n)$,其中$X_i$是各条记录出现频率或其他参数,$W_i$是实时特征评估模型中得到的权系数。神经网络是基于经验风险最小化原则的学习算法,有一些固有的缺陷,比如层数和神经元个数难以确定,容易陷入局部极小,还有过学习现象,这些本身的缺陷在SVM算法中可以得到很好的解决。

(四)SVM 算法

SVM 法即支持向量机(Support Vector Machine)法,由 Vapnik 等人于 1995 年提出,具有相对优良的性能指标[9]。该方法是建立在统计学习理论基础上的机器学习方法。通过学习算法,SVM 可以自动寻找出那些对分类有较好区分能力的支持向量,由此构造出的分类器可以最大化类与类的间隔,因而有较好的适应能力和较高的区分率。该方法只需要由各类域的边界样本的类别来决定最后的分类结果。

支持向量机算法的目的在于寻找一个超平面 H(d),该超平面可以将训练集中的数据分开,且与类域边界的沿垂直于该超平面方向的距离最大,故 SVM 法亦被称为最大边缘(Maximum Margin)算法。所谓最优超平面就是要求超平面不但能将两类正确分开,而且使分类间隔最大;使分类间隔最大实际上就是对模型推广能力的控制,这正是 SVM 的核心思想所在。

首先我们来看看在线性可分情况下两分类问题的最优超平面是如何构建的。在线性可分情况下,最优超平面的构建可转化为下面的最优化问题:

$$\min\phi(\boldsymbol{w}) = \| \boldsymbol{w} \|^2 \text{ subject to } y_i(\boldsymbol{w} \cdot x_i + b) \geqslant 1, i = 1,2,\cdots,l \quad (2)$$

其中:$\boldsymbol{w}$ 为全向量,$\phi(\boldsymbol{w})$ 为向量函数。x_i 为第 i 个训练样本,$y_i = \pm 1$,b 为常数。

利用 Lagrange 优化方法可以把上述最优超平面问题转化为其对偶问题:

$$\max W(\alpha) = \sum_{i=1}^{l} \alpha_i - \frac{1}{2} \sum_{i,j} \alpha_i \alpha_j y_i y_j (x_i \cdot x_j) \text{ subject to } \sum_{i=1}^{l} \alpha_i y_i = 0; \alpha_i \geqslant 0, i = 1,2,\cdots,l \quad (3)$$

其中 α_i 为 Lagrange 乘子。式(3)是一个不等式约束下二次函数寻优的问题,存在唯一解。解中只有一部分(通常是少部分)α_i 不为 0,对应的样本就是支持向量。求解上述问题,得到最优分类函数 $f(x) = \text{sgn}\{(w \cdot x) + b\} = \text{sgn}\{\sum_{I} \alpha_i y_i (x_i \cdot x) + b\}$,其中 $\boldsymbol{I}$ 表示支持向量。

在线性不可分的情况下,可以在式(2)条件中增加一个非负的松驰项 ξ_i,式(2)变为:

$$\min\phi(w) = \frac{1}{2} \| w \|^2 + C[\sum_{i=1}^{l} \xi_i] \text{ subject to } y_i(w \cdot x_i + b) + \xi_i \geqslant 1, i = 1,2,\cdots,l \quad (4)$$

就得到广义最优分类超平面。广义最优分类面的对偶问题与线性可分情况下几乎完全相同,只是条件变为:$C \geqslant \alpha_i \geqslant 0, i = 1,2,\cdots,l$。于是,构建最优超平面的问题就转化为二次规划问题。

非线性问题可以通过非线性变换转化为某个高维空间中的线性问题,在变换空间求最优分类面。上面的对偶问题只涉及训练样本之间的内积运算,在高维空间只需进行内积运算,而这种内积运算是可以用原空间中函数运算来实现。因此,只要一种核函数 $K(x_i, x_j)$ 满足 Mercer 条件,它就对应某一变换空间中的内积。因此 SVM 的决策函数就变成 $f(x) = \text{sgn}\{\sum_{I} \alpha_i y_i K(x_i \cdot x_j) + b\}$,其中,$K(x_i \cdot x_j) = \phi(x_i)\phi(x_j)$。选择不同的核函数就可以生成不同的支持向量机。常用的核包括:多项式核、高斯(径向基函数)核、二层神经网络核等。

根据上面的分析可知,SVM 学习算法最终归结为求解二次规划的问题。常用的 SVM 学习算法包括 SVM - light、SMO、Chunking 等。

一般而言,支持向量机有如下三个主要特点:

(1)于结构风险最小化原则,给出实际风险的上界,保证学习机器具有良好的推广能力。(2)算法最终转化为一个线性约束的凸优化问题,保证了算法的全局最优性和解的唯一性。(3)应用核技术,将输入空间中的线性不可分问题转化为特征空间的线性可分问题。

(五)Boosting 与 Bagging 算法[10]

Boosting 是一种用来提高学习算法准确度的方法,这种方法通过构造一个预测函数系列,然后以一定的方式将它们组合成一个预测函数,达到把一弱学习算法提升为强学习算法的目的。1989 年 Schapire 提出了第一个可证明的多项式时间 Boosting 算法,对这个问题作出了肯定的回答。一年后,Freund 设计了一个高效得多的通过重取样或过滤运作的 Boosting - by - Majority 算法。这个算法尽管在某种意义上是优化的,但却有一些实践上的缺陷。1995 年 Freund 和 Schapire 介绍了通过调整权重而运作的 AdaBoost 算法—AdaBoost、AdaBoost1M1、AdaBoost1M2、AdaBoost1R,解决了早期 Boosting 算法很多实践上的困难。

AdaBoost 是 Boosting 家族中的基础算法。Boosting 家族中的大部分扩展(算法)都由它得来,对 AdaBoost 的分析结论也适用于其它的 Boosting 方法。下面简要地介绍一下它的思想。

AdaBoost 算法的主要思想是给定一弱学习算

法和训练集$(x_1,y_1),\cdots,(x_n,y_n)$。这里$x_i$为一向量,$y_i$对于分类问题为一类别标志,对于回归问题为一数值。初始化时对每一个训练例赋相等的权重$1/n$,然后用该学习算法对训练集训练t轮,每次训练后,对训练失败的训练例赋以较大的权重,也就是让学习算法在后续的学习中集中对比较难的训练例进行学习,从而得到一个预测函数序列$h_1,\cdots,h_t$,其中h_j也有一定的权重,预测效果好的预测函数权重较大,反之较小。最终的预测函数H对分类问题采用有权重的投票方式,对回归问题采用加权平均的方法对新示例进行判别。

Boosting算法是一种基于其他机器学习算法之上的用来提高算法精度和性能的方法。当用于回归分析时,不需要构造一个拟合精度高、预测能力好的回归算法,只要一个效果只比随机猜测略好的粗糙算法即可,称之为基础算法。通过不断地调用这个基础算法就可以获得一个拟合和预测误差都相当好的组合回归模型。Boosting算法可以应用于任何的基础回归算法,无论是线性回归、神经网络、还是SVM方法,都可以有效地提高精度。因此,Boosting可以被视为一种通用的增强基础算法性能的回归分析算法。

Bagging(Bootstrap Aggregating)又被称为自举聚合,是Breiman提出的与Boosting相似的技术。[11]Bagging技术的主要思想是给定一弱学习算法和一训练集$(x_1,y_1),\cdots(x_n,y_n)$。让该学习算法训练多轮,每轮的训练集由从初始的训练集中随机取出的n个训练例组成,初始训练例在某轮训练集中可以出现多次或根本不出现。训练之后可得到一个预测函数序列:$h_1,\cdots,h_t$,最终的预测函数H对分类问题采用投票方式,对回归问题采用简单平均方法对新示例进行判别。

Bagging与Boosting的区别在于Bagging的训练集的选择是随机的,各轮训练集之间相互独立,而Boosting的训练集的选择不是独立的,各轮训练集的选择与前面各轮的学习结果有关;Bagging的各个预测函数没有权重,可以并行生成,而Boosting是有权重的,只能依次顺序生成;Boosting往往从一些弱的学习器开始,组合形成一个集成学习器,从而给出一个好的学习结果,而Bagging学习效果的好坏往往取决于集成学习器中每个学习器的相关性和各个学习器的学习效果。对于神经网络这类极为耗时的学习方法,Bagging可通过并行训练节省大量时间开销。

(六)关联规则算法

关联规则挖掘是由R. Agrawal, T. Imielinski和A. Swami于1993年最先提出[12],它是在大的事务数据集上挖掘项集之间的关联性的一类问题。关联规则分析是机器学习中一大类任务。它起源于对二分变量的分析,用规则的方式来表达两个二分变量之间的关系,以及多个二分变量之间的关系。当然,后来的发展,也使得关联规则不仅仅局限于二分变量,也可以对多分类变量和连续变量进行分析。所以,关联规则可以看作是分析变量之间关系,并且把这种关系表达成非常容易解释的规则的方法[13]。

关联规则分析方法对数据分布不作任何要求,所得的结果是完全基于数据的,没有任何主观假定,客观地反映了数据的本质,有很强的说服力。关联规则对数据分析得到的结果可以看作是对数据中变量间所有规律的总结。因此关联规则在提出之后,在各行各业得到了大量的应用。

关联规则的算法就是由输入向输出的一个求解过程。设$I=\{i_1,i_2,\cdots,i_m\}$是m个不同项目的集合,其中的元素称为项(Item)。记D为交易T(Transaction)的集合,这里交易T是项的集合,并且$T\subseteq I$。对应每一个交易有唯一的标识,如交易号,记作TID。一个关联规则是形如X⇒Y的蕴涵式,这里,$X\subset I$,$Y\subset I$,并且$X\cap Y=\varnothing$。X称为规则的前提,Y是结果。

规则X⇒Y在交易集D中的支持度(Support)是指包含X和Y的交易数与所有交易数之比,记为support(X⇒Y),即support(X⇒Y)=|{T:X∪Y⊆T,T∈D}|/|D|。

规则X⇒Y在交易集D中的可信度(Confidence)是指包含X和Y的交易数与包含X的交易数之比,记为confidence(X⇒Y),即confidence(X⇒Y)=|{T:X∪Y⊆T,T∈D}|/|{T:X⊆T,T∈D}|。给定一个交易集D,挖掘关联规则问题就是产生支持度和可信度分别大于用户给定的最小支持度(Minsupp)和最小可信度(Minconf)的关联规则,称为强规则。

关联规则挖掘的任务就是要挖掘出数据集D中所有的强规则。强规则X⇒Y对应的项目集(X∪Y)必定是频集,频集(X∪Y)导出的关联规则X⇒Y的置信度可以以频集X和(X∪Y)的支持度计算。因此,可以把关联规则挖掘划分为以下两个子问题:

(1)根据最小支持度找出数据集D中的所有频

集;

(2)根据频繁项目集和最小置信度产生关联规则。

第一个子问题的任务是迅速高效地找出D中全部频集,它是关联规则挖掘的中心问题,是衡量关联规则挖掘算法的标准;第二个子问题由最小置信度的定义求解比较容易。目前,所有的关联规则挖掘算法都是针对第一个子问题而提出的,也是决定关联规则挖掘算法性能的问题。

现有的各种关联规则挖掘算法大致可分为搜索算法、层次算法、数据集划分算法、抽样算法等。

(七)贝叶斯学习算法

Bayes法是一种在已知先验概率与类条件概率的情况下的模式分类方法,待分样本的分类结果取决于各类域中样本的全体。

设训练样本集分为M类,记为$C=\{c_1,\cdots,c_i,\cdots,c_M\}$,每类的先验概率为$P(c_i)$,$i=1,2,\cdots,M$。当样本集非常大时,可以认为$P(c_i)=c_i$类样本数/总样本数。对于一个待分样本$X$,其归于$c_i$类的类条件概率是$P(X/c_i)$,则根据Bayes定理,可得到$c_i$类的后验概率$P(c_i/X)$:

$$P(c_i/X)=P(X/c_i)\times P(c_i)/P(X) \quad (5)$$

若$P(c_i/X)=Max_jP(c_j/X)$,$i=1,2,\cdots,M$;$j=1,2,\cdots,M$,则有$X\in c_i$。 (6)

式(6)是最大后验概率判决准则,将式(5)代入式(6),则有:

若$P(X/c_i)P(c_i)=Max_jP(c_j/X)$,$i=1,2,\cdots,M$;$j=1,2,\cdots,M$,则$X\in c_i$。这就是最大后验概率判决准则,这就是常用到的Bayes分类判决准则。经过长期的研究,Bayes分类方法在理论上论证得比较充分,在应用上也是非常广泛的。

Bayes方法的薄弱环节在于实际情况下,类别总体的概率分布和各类样本的概率分布函数(或密度函数)常常是不知道的。为了获得它们,就要求样本足够大。此外,当用于文本分类时,Bayes法要求表达文本的主题词相互独立,这样的条件在实际文本中一般很难满足,因此该方法往往在效果上难以达到理论上的最大值。

(八)EM算法

在人工智能、数据挖掘、模式识别和机器学习中有许多的应用都要进行模型的参数估计,也就是要进行极大似然估计或极大后验似然估计。当模型中的变量均为可以直接观察的变量时,极大似然或极大后验似然是显然的。但是当某些变量隐藏时,进行极大似然估计就比较复杂。在存在潜在变量的情况下,对模型参数进行估计的方法有很多种,一种非常流行的极大似然估计方法是Expectation-Maximization算法,通常简称为EM算法。它不是直接对复杂的后验分布进行极大化或模拟,而是在观察数据的基础上添加一些"潜在数据",从而简化计算并完成一系列简单的极大化或模拟。它之所以被称为EM算法是因为算法的每一次迭代由一个期望步(E-step)和极大步(M-step)构成。这个名字首先是由Dempster、Laird和Rubin(以下简称DLR)给出的。[14]

EM算法的特点是简单和稳定,特别是每一次迭代能保证观察数据对数后验似然是单调不减的。也就是说,假定$L(\theta)$是观察数据对数后验似然,θ^i和θ^{i+1}分别是第i次和$i+1$次迭代得到的参数估计值那么就有$L(\theta^{i+1})>L(\theta^i)$。另外,DLR还定义了广义的EM算法(Generalized EM algorithm)(简称GEM算法),把EM算法视为GEM算法的特例。GEM算法在计算上更有效,且保持了对数似然单调不减的特性。

EM算法是一种从"不完全数据"中求解模型参数的极大似然估计方法。所谓"不完全数据"一般分为两种情况:一种是由于观察过程本身的限制或者错误,造成观察数据成为错漏的不完全数据;一种是参数的似然函数直接优化十分困难,而引入额外的参数(隐含的或丢失的)后就比较容易优化,于是定义原始观察数据加上额外数据组成"完全数据",原始观察数据自然就成为"不完全数据"。

EM算法的基本原理可以表述如下:可以观察到的数据是y,完全数据$x=(y,z)$,z是隐变量,表示缺失数据,θ是模型参数。θ关于y的后验分布$p(\theta\mid y)$很复杂,难以进行各种不同统计计算。假如z已知,则可能得到一个关于θ的简单的添加后验分布$p(\theta\mid y,z)$,利用$p(\theta\mid y,z)$的简单性可以进行各种统计计算。然后,又可以对z的假定作检查和改进,从而将一个复杂的极大化或抽样问题简化。

EM算法是一种迭代方法,主要用于求后验分布的众数。

EM算法的具体实现步骤如下:

假设y是服从某一分布的非完全观测数据集,且存在一个完全数据集$x=(y,z)$,则x的密度函数为:

$$p(x\mid\theta)=p(y,z\mid\theta)=p(z\mid y,\theta)p(y\mid\theta) \quad (7)$$

从式(7)可以看出,密度函数$p(x\mid\theta)$是由边

际密度函数 $p(y \mid \theta)$、隐变量 z 的假设、参数 θ 初始估计值以及隐变量 z 与观测变量 y 之间的关系决定。

下面讨论密度函数 $p(x \mid \theta)$ 的具体形式。

由式(7)给出的密度函数可以定义一个新的似然函数

$$L(\theta \mid x) = L(\theta \mid y,z) \triangleq p(y,z \mid \theta) \quad (8)$$

称此函数为完全数据似然函数。由于隐变量 z 未知,因此似然函数 $L(\theta \mid x)$ 是随机的,且由隐变量 z 所决定。

EM 算法的第一步 E - step:即给定观测 y 和当前参数估计值 θ^{i-1},计算完全数据对数似然函数 $\log_p(y,z \mid \theta)$ 关于未知数据 z 的期望。为此,定义对数似然函数的期望

$$Q(\theta,\theta^{i-1}) = E[\log_p(y,z \mid \theta) \mid y,\theta^{i-1}] \quad (9)$$

在式(9)中,y 和 θ^{i-1} 为常数,θ 为待优化的参数。z 为一随机变量,并假设它服从某一分布 $f(\cdot)$。

$$z \sim f(z \mid y,\theta^{i-1}) \quad (10)$$

因此,式(9)可写为:

$$Q(\theta,\theta^{i-1}) = E[\log_p(y,z \mid \theta) \mid y,\theta^{i-1}] = \int_{z \in D} \log_p(y,z \mid \theta) \cdot f(z \mid y,\theta^{i-1}) dz \quad (11)$$

其中 $f(z \mid y,\theta^{i-1})$ 是不可观测数据 z 的边际分布密度函数,并且依赖于观测数据 y 和当前参数 θ^{i-1},D 为 z 的取值空间。在一些特殊情况下,边际分布 $f(z \mid y,\theta^{i-1})$ 是 y 和 θ^{i-1} 的简单解析函数,但通常这个函数很难得到。由乘法公式,得

$$f(z,y \mid \theta^{i-1}) = f(z \mid y,\theta^{i-1}) \cdot f(y \mid \theta^{i-1}) \quad (12)$$

由于因子 $f(y \mid \theta^{i-1})$ 与 θ 无关,所以在实际问题处理中,用 $f(z,y \mid \theta^{i-1})$ 代替 $f(z \mid y,\theta^{i-1})$ 不影响式(11)中似然函数的最优化。

定义二元函数 $h(\theta,z) \triangleq \log L(\theta \mid y,z)$,其中 z 服从某一分布,那么,有:

$$E_z[h(\theta,z)] = \int_z h(\theta,z) \cdot f_z(z) dz \triangleq q(\theta) \quad (13)$$

从式(13)可知 $E_z[h(\theta,z)]$ 是关于 θ 的函数,以通过简单的最优化方法得到参数 θ 的估计值 $\hat{\theta}$。期望值 $E_z[h(\theta,z)]$ 的计算也就是 EM 算法的 E - step。

EM 算法的第二步 M - step:最大化期望值 $Q(\theta,\theta^{i-1})$,即找到一个 $\theta^{(i)}$,在参数空间 Θ 下满足 $\theta^{(i)} = \arg_\Theta \max Q(\theta,\theta^{i-1})$。

如此形成一次迭代 $\theta^{i-1} \to \theta^i$,将上述 E 步和 M 步反复迭代直至 $\| \theta^i - \theta^{i-1} \|$ 或者 $\| Q(\theta^i,\theta^{i-1}) - Q(\theta^{i-1},\theta^{i-1}) \|$ 充分小时才停止。

在一般的情况下,*EM* 算法的结果只能保证收敛到后验分布密度函数的稳定点,并不能保证收敛到极大值点,事实上,任何一种算法都很难保证其结果为极大值点。

EM 算法得到广泛运用的一个重要原因是在 M 步中求极大化的方法与完全数据下求极大化的方法完全一样。在许多场合,这样的极大化有显式的表达式,然而并不总是这样,有时要找一个使 $Q(\theta,\theta^{i-1})$ 达到最大的 θ 是很困难的,一个较为简单的方法是找一个 θ^i,使得 $Q(\theta^i,\theta^{i-1}) > Q(\theta^{i-1},\theta^{i-1})$,这就是 GEM 算法。由于每一个迭代算法都隐含着一个映射,用 $M(\theta)$ 来表示 GEM 算法所隐含的映射,即对任意的 θ,有 $Q(M(\theta),\theta) > Q(\theta,\theta)$。

四、小　结

机器学习并不是为了替代传统的统计分析技术。相反,它是统计方法学的延伸和拓展。大多数的统计分析技术都基于完善的数学理论和严格的假定条件之下,而随着计算机能力的不断增强,我们有可能只利用计算机强大的计算能力只通过相对简单和固定的方法达到传统统计方法无法达到的效果和目的。

近年来,国内外有关机器学习的研究发展较快,由于集成学习(Ensemble Learning)可以有效地提高模型的推广能力,因此从 20 世纪 90 年代开始,对集成学习理论和算法的研究成为了机器学习的一个热点。早在 1997 年,国际机器学习界的权威 T. G. Dietterich 就将集成学习列为机器学习四大研究方向之首。四个大方向指通过集成学习方法提高学习精度、扩大学习规模、强化学习和学习复杂的随机模型。而在今天,集成学习仍然是机器学习中最热门的研究领域之一,研究人员众多、成果层出不穷。现在已经有很多集成学习算法,比如:Bagging 算法、Boosting 算法、Arcing 算法、Random Forest 算法等等。

需要注意的是,Bagging 算法和其他大多数的集成学习算法都是为有指导学习而设计的,对聚类这样的无指导学习来说,由于训练样本缺乏类别标记,聚类结果之间没有直接的对应关系,这将使得对个体学习器的结合难以直接进行。因此,用于无指导学习的集成学习算法比一般的用于有指导学习的集成学习算法设计起来更加困难。实际上,利用集成学习技术来提高聚类分析的性能已经引起了一些

研究者的关注。例如,在 Strehl 和 Ghosh 的工作中,他们利用互信息(Mutual Information)把聚类集成问题定义为一个基于互信息的优化问题,但同时又指出由于该优化问题的计算开销过于庞大,因此难以应用于实际领域。

参考文献:

[1] 龙卫江.基于相近原则的半指导直推学习及其增量算法[J]. 应用数学学报,2006,29(4): 619-632.

[2] Nigam K, McCallum A K, Thrun S, Mitchell T. Text classification from labeled and unlabeled documents using EM[J]. Machine Learning, 2000, 39(2-3): 103-134.

[3] Joachims T. Transductive inference for text classification using support vector machines[G]. In: Proc 16th Int'l Conf Machine Learning, Bled, Slovenia, 1999, 200-209.

[4] Blum A, Mitchell T. Combining labeled and unlabeled data with co-training[G]. In: Proc 16th Annual Conf Computational Learning Theory, Madison, WI, 1998, 92-100.

[5] 刘琴. 机器学习[J]. 武钢职工大学学报,2001(6):41-44.

[6] Breiman L., Friedman, J., Olshen, R., and Stone, C. Classification and Regression Trees[M]. Wadsworth, 1984.

[7] Breiman L. Random forests[J]. Machine Learning, 2001, 45(1): 5-32.

[8] 漆书青,戴海琦,丁树良. 现代教育与心理测量学原理[M]. 南昌:江西教育出版社,1998.

[9] Cortes, C., Vapnik, V.M.,. Support Vector Networks[J]. Machine Learning, 1995, 20: 273-297.

[10] Dietterich, T. G. An Experimental Comparison of Three Methods for Constructing Ensembles of Decision Trees: Bagging, Boosting, and Randomization[J]. Machine Learning, 2000, 40: 139-157.

[11] Breiman, L., Bagging Predictors[J]. Machine Learning, 1996, 24(2): 123-140.

[12] Agrawal R, Imielinski T, Swami A. Mining association rules between sets of items in large databases[A][G]. Proc of ACM SIGMOD Conf on Management of Data[C]. Washington, 1993. 207-216.

[13] 刘星沙,谭利球,等. 关联规则挖掘算法及其应用[J]. 计算机工程与科学,2007,29(1): 83-86.

[14] P. Dempster, N. M. Laird, AandD. B. Rubin. Maximum Likelihood From Incomplete Data Via the EM Algorithm[J]. RoyalStat. Soc. 1997, 39(1): 1-38.

(责任编辑:张治国)

A Summary of Machine Learning and Related Algorithms

CHEN Kai[1], ZHU Yu[1,2]

(1. School of Statistics, Renmin University of China, Beijing 100872, China;
2. Xi'an University of Finance & Economic, Xi'an 710061, China)

Abstract: Since the computer was invented, people have been wanted to know that whether it can learn. Machine learning is essentially a multidisciplinary field. It absorbed some results of artificial intelligence, probability and statistics, computational complexity theory, control theory, information theory, philosophy, physiology, neurobiological. This paper mainly based on statistical learning wanted to give a brief review and presentation to the perspective of machine learning and the development of related algorithms.

Key words: machine learning; supervised learning; unsupervised learning; semi-supervised learning