决策树、朴素贝叶斯和朴素贝叶斯树的比较①

王守选[1], 叶柏龙[2], 李伟健[3], 谭一云[3]

[1](湖南创博龙智信息科技股份有限公司, 长沙 410205)
[2](中南大学, 长沙 410083)
[3](湖南大学 信息科学与工程学院, 长沙 410082)

摘 要: 利用数据挖掘工具 Weka, 在常态数据集 adult 上进行实验, 从时间、正确率、误差率三个指标对比分析 J48(C4.5 决策树), 朴素贝叶斯分类器, 朴素贝叶斯树(NBTree)三种分类算法. 结论为:在内存充足, 时间要求不高的情况下, 使用朴素贝叶斯树(NBTree)能获得更高的正确率和错误率; J48 则是一种折中的方案; 朴素贝叶斯分类器完成时间最短, 但正确率和错误率为三种算法中最差.

关键词: C4.5 决策树; 朴素贝叶斯分类器; 朴素贝叶斯树

Comparison of Decision Tree, Native Bayesion and Natvie Bayesion Tree

WANG Shou-Xuan[1], YE Bai-Long[2], LI Wei-Jian[3], TAN Yi-Yun[3]

[1](Hunan Belong Soft Information Technology Company Limited, Changsha 410205, China)
[2](Central South University, Changsha 410083, China)
[3](College of Information Science and Engineering, Hunan University, Changsha 410082, China)

Abstract: In this paper, we used the normal dataset ‘adult’ to compare the runing time, correct rate and error rate of c4.5 decition tree , native bayesion and native bayesion tree, with the data mining tool called weka. The result shows that: we can choose the NB Tree when the memory is big enough and the running time is undemanding; and choose natvie bayesion when the job should be done quickly; and the c4.5(j48) decision tree is an compromising approach.

Key words: C4.5 Decition Tree; Native Bayesian; NB Tree

1 引言

C4.5 决策树算法是机器学习中广为人知的算法, 是 ID3 算法的改进, 并继承了 ID3 的全部优点. 在归纳学习中, 它代表着基于决策树(Decision Tree)的方法, 通过对一组训练数据的学习, 构造出决策树形式的知识表示, 在决策树的内部结点进行属性值的比较并根据不同的属性值判断从该结点向下的分支, 在决策树叶结点得到结论. 所以从根到叶结点的一条路径就对应着一条规则, 整棵决策树就对应着一组析取表达式规则. 基于决策树学习算法的一个最大的优点就是它在学习过程中不需要使用者了解很多背景知识[1]. 这样只要训练事例能够用属性——结论式的方式表达出来, 就能使用该算法来进行学习[2]. 虽然 C4.5 算法存在一些不足, 但由于它思想简单, 实现高效, 结果可靠, 使其在归纳学习中的地位依旧十分显著.

和决策树模型相比, 朴素贝叶斯模型发源于古典数学理论, 有着坚实的数学基础, 以及稳定的分类效率. 同时, NBC 模型所需估计的参数很少, 对缺失数据不太敏感, 算法也比较简单. 理论上, NBC 模型与其他分类方法相比具有最小的误差率. 但是实际上并非总是如此, 这是因为 NBC 模型假设属性之间相互独立, 这个假设在实际应用中往往是不成立的, 这给 NBC 模型的正确分类带来了一定影响[3]. 在属性个数比较多或者属性之间相关性较大时, NBC 模型的分类效率比不上决策树模型. 而在属性相关性较小时, NBC 模型的性能最为良好.

① 基金项目:国家发改委信息安全产品专项基金(发改办高技[20091886 号]);国家创新基金重点项目(11C26214305383)
收稿时间:2012-05-09;收到修改稿时间:2012-06-12

朴素贝叶斯树(NBTree)融合了 C4.5 和贝叶斯的思想, 在分类 C4.5 的叶节点时, 使用朴素贝叶斯的方法, 其性能比两者都有提高.

2 实验步骤

Adult Data Set 中含有训练数据集 32561 条, 测试数据集 16281 条. 分类的目标属性为 Salary, 根据年龄(age)、教育(education)、职业(occupation)等情况来判断其工资(Salary)是否大于 50K,实验步骤如下:

(1)从命令行进入 weka 根目录, 键入命令 java –Xmx1408m –jar weka.jar 运行 weka.

(2) 进入 Explorer 面板, 选择 Preprocess, 选择训练数据集 adult_train.arff, 在分类属性中选择 Salary.

(3) 进入 Classify 面板, 从 classifier 中选择分类算法, 从 Test Option 中选择“ using training set ”, 选择分类属性“ Salary ”, 点击“ Start ”开始训练.

(4) 在训练时, 还可以选择较交叉验证, 再用交叉验证的方式学习出来模型预测测试集, 其精度可能会提升, 但是需要的时间增长.

(5) 分类算法选择

选择 J48 算法: weka --> classifiers --> trees --> J48 (C4.5 决策树算法), 设置参数为“ -C 0.25 -M 1 ”;

选择 Naïve Bayes 算法: weka --> classifiers --> bayes --> Naïve Bayes(朴素贝叶斯);

选择 NBtree 算法: weka --> classifiers --> trees --> NBTree(朴素贝叶斯树),Na?ve Bayes 和 NBTree 算法不需要参数; 采用训练集上学习道德分类模型, 从 Test Option 中选择“ Supplied test set ”, 装载测试集数据 adult_test.arff, 使用不同分类模型分别对测试集数据进行分类预测.

3 分类结果

在该数据集的分类上, 使用了 J48(C4.5 决策树), 朴素贝叶斯分类器, 朴素贝叶斯树(NBTree)三种分类算法, 并分别对比分析其分类结果和性能. 三种不同的分类算法的分类结果如表 1 所示:

表 1 不同算法针对 Adult 数据集的分类结果

分类算法	时间(s)	正确率	误差率
J48	7.55	85.9652%	14.0348%
Naive Bayes	0.25	83.1276 %	16.8724 %
Naive Bayes Tree	46.34	86.2293 %	13.7707 %

从实验数据可以看出, NBTree(Naive Bayes Tree)分类算法的效果比 J48(决策树)和 Naive Bayes(朴素贝叶斯)算法都要好一些, NBTree 融合了 J48 和朴素贝叶斯算法的优点.

4 结果分析

4.1 J48(C4.5 决策树)

算法具体分类结果如表 2 所示:

表 2 J48(C4.5 决策树)算法在 Adult 数据集上的分类结果

		Predicted Class	
Actual Class		Class = a	Class = b
	Class = a	U = 11579	V = 856
	Class = b	X = 1429	Y= 2417

注释: a : Salary <=50K, b : Salary >50K; U 表示实际工资大于 50K, 并且分类结果也划分成大于 50K 的实例个数; V 表示实际工资大于 50K, 并且分类结果却被划分成小于等于 50K 的实例个数; X 表示实际工资小于等于 50K, 并且分类结果却划分成大于 50K 的实例个数; V 表示实际工资小于等于 50K, 并且分类结果也划分成小于等于 50K 的实例个数.

分类正确的实例数目: Right=U+Y=11556+2421= 13996, 分类错误的实例数目 Wrong=X+V=1425+879 = 2285. 分类准确率为 13996 / 16281 = 85.9652 %, 分类错误率为 2285 / 16281 = 14.0348 %. C4.5 决策树算法分类的精度(查准率), 检索率(查全率)和 F-Measure 指标如表 3 所示:

表 3 J48(C4.5 决策树)算法在 Adult 数据集上的分类效果

Class	Precision	Recall	F-Measure
Salary <= 50K	0.89	0.931	0.91
Salary >50K	0.738	0.628	0.679

注释: Recall(r),检索率(亦即查全率)表示正确地分类成工资大于(或者小于等于)50K 的实例个数占总的工资大于(或者小于等于)50K 实例个数的比例; Precision(p), 精度(查准率), 表示分类成工资大于(或者小于等于)50K 的实例个数中, 实际上工资真实的大于(或者小于等于)50K 的实例个数占的比例; F-Measure 把查全率和查准率结合到一起, 衡量两者的平衡性能.

例如:

对于 a: Salary <= 50K

检索率 Recall(r) = U /(U + V)= 11575 /(11575 + 860)= 93.1%

精确率 Precision (P) = U /(U + X)= 11575 /(11575 + 1887)= 89%

F-Measure = 2*r*p /(r + p)= 2*U /(2*U + V + X)= 91%

对于 b: Salary >50K

检索率 Recall(r) = Y /(X + Y)= 2417 /(1429 + 2417)= 62.8%

精确率 Precision (P) = Y /(Y + V)= 1959 /(1959 + 860)= 73.8%

F-Measure = 2*r*p /(r + p)= 2*Y /(2*Y + V + X)= 67.9%

J48(即 C4.5 算法)是 ID3 算法的扩展, 其实质是一种决策树算法. 在该数据集的分类预测中, C4.5 的准确率达到了 86%左右, 分类的速度大概在 7.5 秒, 该算法克服了 ID3 算法关于值缺失和噪音数据的缺点(即增加了树枝的后修剪), 应用在该数据集上还是可以接受的.

Weka 中的 J48 算法有两个参数可以调节. -C(Confidence Factor, 置信因子), 置信因子增加, 将减少对树的修剪, 以获得一颗更加特殊的树, 置信因子减少将获得一棵更一般的树. -M, 即新分类的叶子节点上需要含有的实例个数, 越多代表该树越普通否则将更特殊. 调节参数时, 分类精度变化如表 4 所示:

表 4 C4.5 参数调节对分类精度的影响

-C	-M	准确率	错误率
0.25	1	85.9652%	14.0348%
0.25	2	85.8485 %	14.1515 %
0.5	1	85.1176%	14.8824%
0.5	2	85.3572%	14.6428

4.2 **Naïve Bayes**(朴素贝叶斯)

朴素贝叶斯分类器的分类结果如表 5 所示:

表 5 朴素贝叶斯算法在 Adult 数据集上的分类结果

		Predicted Class	
Actual Class		Class = a	Class = b
	Class = a	U = 11575	V = 860
	Class = b	X = 1887	Y = 1959

分类正确的实例数目: Right = U + Y = 11575 + 1959 = 13534, 分类错误的实例数目 Wrong = X + V = 1887 + 860 = 2747. 分类准确率为 13534 / 16281 = 83.1276 %, 分类错误率为 2747 / 16281 = 16.8724 %. 朴素贝叶斯分类器分类的精度(查准率), 检索率(查全率)和 F-Measure 指标如表 6 所示:

表 6 朴素贝叶斯算法在 Adult 数据集上的分类效果

Class	Precision	Recall	F-Measure
Salary <= 50K	0.8598	0.9308	0.894
Salary >50K	0.6949	0.5094	0.588

朴素贝叶斯分类器基于给出样本的统计学习, 它假设样本的各属性之间彼此独立, 使用统计的方法从样本中求出先验概率. 所以该分类方法需要较大的样本空间, 并且样本属性之间的关联尽量小.

4.3 **NB Tree**(朴素贝叶斯树)

朴素贝叶斯树分类结果如表 7 所示:

表 7 朴素贝叶斯树在 Adult 数据集上的分类结果

		Predicted Class	
Actual Class		Class = a	Class = b
	Class = a	U = 11569	V = 866
	Class = b	X = 1376	Y = 2470

分类正确的实例数目: Right = U + Y = 11569 + 2470 = 14039, 分类错误的实例数目 Wrong = X + V = 1376 + 866 = 2242. 分类准确率为 14039 / 16281 = 86.2293 %, 分类错误率为 2242 / 16281 = 13.7707%. 朴素贝叶斯树分类的精度(查准率), 检索率(查全率)和 F-Measure 指标如表 8 所示:

表 8 朴素贝叶斯树在 Adult 数据集上的分类

Class	Precision	Recall	F-Measure
Salary <= 50K	0.894	0.93	0.912
Salary >50K	0.74	0.642	0.688

朴素贝叶斯树分类结合了 C4.5 算法和朴素贝叶斯分类算法的优点, 在 C4.5 决策树的每一个节点使用朴素贝叶斯方法进行验证, 给出了比两者都要稍微高一点的分类精度.

5 结论

数据挖掘中算法的性能比较一般从分类速度、准确率、可伸缩性、强壮性以及可理解性等几个方面进行比较, 下表对所采用的算法进行了一个简单的比较, 如表 9 所示:

表 9 Adult 数据集上的不同算法对比分析

算法	时间(s)	正确率	误差率	可伸缩性	强壮性	可理解性
J48	7.55	85.9652%	14.0348%	在内存构建树	强	较好
Naive Bayes	0.25	83.1276 %	16.8724 %	计算速度较快	可以容忍	较好
Naive Bayes Tree	46.34	86.2293 %	13.7707 %	在内存构建树	可以容忍	较好

强壮性一般指容忍噪音数据和缺失数据的能力; 可伸缩性一般随着数据规模的扩大, 算法处理的效率; 可理解性是指学习出来的模型是否容易理解.

J48(C4.5)算法是 ID3 算法的扩展, 可以容忍缺失数据, 进行了后剪枝(防过度拟合). C4.5 算法的主要思想是, 根据信息熵的增益, 从样本的属性中提取最有利于区分实例类别的属性, 一步一步由根节点向叶节点构造决策树, 可以从生成的决策树中提取规则. 该算法基于内存构造整棵树, 当数据的规模增大时, 算法的可扩展性是一个需要考虑的问题.

C4.5 算法产生的分类规则易于理解, 准确率较高, 对数据分布无任何要求, 应用于银行和金融的效果比较好.

朴素贝叶斯是一种基于统计的学习方法, 它首先从大量的样本中统计出类别先验概率, 然后利用贝叶斯公式(须假设属性之间相互独立), 来计算每一个未见实例的最有可能类别. 朴素贝叶斯在处理不确定性信息的智能化系统中已经得到广泛的应用, 已成功应用于统计决策、医疗诊断、专家系统等领域.

朴素贝叶斯树(NBTree)融合了 C4.5 和贝叶斯的思想, 在分类 C4.5 的叶节点时, 使用朴素贝叶斯的方法, 其性能比两者都有提高. NBtree 实质上也是基于内存构造树, 数据规模是一个较大的问题. 譬如, 具体到下一个数据集 Covertype, NBTree 算法运行时就出现了内存不足, weka 非正常退出的情况(物理内存为 2G 时, Java 虚存至多设置 1408M, 再往上系统不允许, 但此时 NBTree 所需内存仍然不够). 朴素贝叶斯树分类器放松了朴素贝叶斯的属性独立性假设,是对朴素贝叶斯分类器的有效改进.

参考文献

1 林忠惠.基于归纳学习的数据挖掘技术在高校教学研究中的应用[硕士学位论文].哈尔滨:哈尔滨工程大学, 2008.
2 于士涛.基于问答网络论坛知识体系的自动问答系统研究[硕士学位论文].天津:南开大学,2009.
3 沈静.基于 UCL 的网页信息自动分类及标引技术研究[硕士学位论文].绵阳:西南科技大学,2008.
4 姜欣,徐六通,张雷.C4.5 决策树展示算法的设计.计算机工程与应用,2003,(4):93–97.
5 陆远蓉.使用数据挖掘工具.电脑知识与技术,2008,(1).
6 周丹,沈利迪.C4.5 决策树构造算法应用研究.中国高新技术企业,2008.113–114.
7 高岩.朴素贝叶斯分类器的改进研究[硕士学位论文].广州:华南理工大学,2011.
8 王国才.朴素贝叶斯分类器的研究与应用[硕士学位论文].重庆:重庆交通大学,2010.
9 李静梅,孙丽华,等.一种文本处理中的朴素贝叶斯分类器.哈尔滨工程大学学报,2003,(1):71–74.