

集成分类对比: Bagging NB & Boosting NB

李晓波

(潍坊学院 计算机与通信工程学院, 山东 潍坊 261061)

摘 要: Bagging 和 Boosting 是两种重要的投票分类算法, 前者并行生成多个分类器, 后者通过调整样本权重, 串行生成多个分类器. 将 Bagging 与 Boosting 算法与朴素贝叶斯算法相集成, 构建了 Bagging NB 和 AdaBoosting NB 算法. 以 UCI 数据集为基础, 进行实验对比, 结果表明, Bagging NB 算法较为稳定, 可以产生优于 NB 算法的分类结果, 而 Boosting 算法受到数据分布中的奇异值影响较大, 部分数据集上与 NB 算法的基础效果较差.

关键词: 分类算法; Bagging; Boosting; 朴素贝叶斯

中图分类号: TP311

文献标识码: A

文章编号: 1000-7180(2010)08-0136-04

Integrated Classification Comparison: Bagging NB & Boosting NB

LI Xiao-bo

(College of Computer and Communication Engineering, Weifang University, Weifang 261061, China)

Abstract: Bagging and Boosting are two important voting classification algorithms. Bagging parallel generates multiple classifiers and by adjusting the sample weights, Boosting generates multiple classifiers serially. This paper integrated Bagging and Boosting algorithms with Naive Bayesian to construct Bagging NB and AdaBoosting NB. Based on UCI data sets, experiment result show that Bagging NB is more stable and can produce more accurate classifier than NB. Boosting algorithms are sensitive with the distribution of the data set and sometimes less effective.

Key words: classification algorithm; Bagging; Boosting; naïve Bayesian

1 引言

如何获取更高的分类精度是研究者不断研究的目的之一, 单一的分类算法往往存在缺陷: NB 算法的独立性假设通常不成立^[1], 最近邻算法的实现过程对大规模数据集处理效率低下, C4.5 算法受到数据集中的奇异数据影响较大^[2]. 因此, 研究者提出集成分类的概念, Bagging 算法^[3]和 Boosting 算法^[4]是最为著名的集成分类算法, 采用投票的方式由多个分类器决定最终的分类结果.

本文重点讨论 Bagging 算法和 Boosting 算法的实现过程, 并将其与朴素贝叶斯算法相集成, 构建 Bagging NB 和 Boosting NB, 通过对比试验, 获取两种投票集成算法的分类正确率表现并讨论其优缺点.

2 Bagging & Boosting

对于分类问题, 测试样本的不完全性、奇异值等特征影响分类器的训练过程. 单独使用一个分类器进行分类, 往往不能取得最优的分类准确性, 此时, 将这种分类器称作弱分类器. 经过迭代学习, 弱分类器可以转化为强分类器^[3], 具有更加精确地分类能力.

Bagging 是一种并行训练过程, 通过分类测试样本的有放回抽样, 获取多个分类测试子样本, 通过分类子样本训练多个分类器, 最终取多分类器的投票结果作为分类判别依据.

Bagging 的过程如图 1 所示: 模式训练阶段, 对于有限样本组成的训练集合 S , 通过 bootstrap^[3] 过程生成测试样本 S_i , 以 S_i 为训练集, 构建分类器

C_i ; 模式匹配时, 依据 $\{C_1, C_2, \dots, C_T\}$ 的综合投票结果, 确定测试样本 x 的类别。

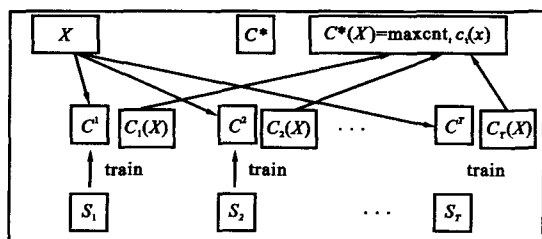


图1 Bagging 过程

Bagging 算法可以描述如下:

算法 1: Bagging 算法

输入: 训练集合 S , 测试数据 x , 分类方法 C ;

输出: x 的类别判别 R ;

begin

for $i = 1, 2, \dots, T$ do

$S_i = \text{bootstrap}(S)$;

$C_i = C(S_i)$;

$R_i = C_i(x)$;

endfor.

$R = \text{CntMax}(R_i)$;

end.

其中, CntMax 表示从 T 个分类器的判定结果中, 寻求最多的支持类别。可见, Bagging 算法是一个典型的并行分类器集成投票算法, 各分类器的学习过程彼此独立。

Boosting 算法是一种类似 Bagging 的集成投票方法, AdaBoosting 算法作为 Boosting 的典型代表, 受到的关注最多^[5]。

AdaBoosting 算法流程如图 2 所示, 同时参考图 1, AdaBoosting 与 Bagging 的主要区别在于三处: 首先, Bagging 的测试集 $\{S_1, S_2, \dots, S_T\}$ 来源于随机的有放回抽样, 而 AdaBoosting 的测试集在 T 轮迭代过程中均采用 S ; 其次, AdaBoosting 方法对测试集辅以权值 $w\{w_1, w_2, \dots, w_m\}$, 通过调整权值使得容易错分的样本受到更多的关注; 最后一区别在于, 分类判别时, AdaBoosting 算法使用权重算子 $\log(1/\beta_i)$ 来衡量每个分类器的结果。

AdaBoosting 算法可以描述如下:

算法 2: AdaBoosting 算法

输入: 训练集合 S (含 N 个样本), 测试数据 x , 分类方法 C ;

输出: x 的类别判别 R ;

begin

$\{W_1, W_2, \dots, W_s\} = 1/N$;

for $i = 1, 2, \dots, T$ do

$C(S)$;

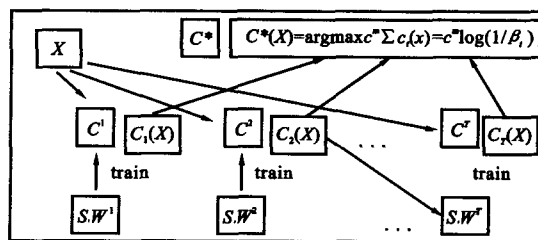


图2 AdaBoosting 过程

$E = \sum_{S \text{ Misclassified}} W_i$;

$\beta_i = E/(1 - E)$;

if (MisClassified(S_i))

$W_i = W_i$;

else

$W_i = \beta_i * W_i$;

Normalization(W);

$R_i = C(x)$;

endfor.

$R = \text{CntMax}(\log(1/\beta_i) R_i)$;

end.

AdaBoosting 算法中, 三点需要注意: 其一, 每个样本记录拥有自己的权重, 初始权重为 $1/N$, 而后随着分类的迭代, 权重逐渐调整, 错分的样本的权重逐渐增加, 此时分类器朝着更有利于易错样本的方向演化。每次迭代结束, 均需要对权重进行归一化处理, 保证所有样本权重之和为 1; 每次迭代过程中, 需要计算测试样本集合的误分率 E , 所以, AdaBoosting 算法较 Bagging 算法的训练过程的时间复杂度高; 对于类别的判断, 每个弱分类器的贡献与 β 有关, 弱分类器的贡献权值为 $\log(1/\beta)$ 。

3 BaggingNB 与 BoostingNB

朴素贝叶斯分类器是应用广泛的分类器, 本部分讨论投票方法与 NB 的集成问题, 给出具体的实现方法。

NB 算法基于属性独立性假设, 不考虑属性间的关联关系。其核心步骤可以分为机器学习阶段与模式匹配阶段, 机器学习阶段通过训练集样本获取其分布特征, 一般采用正态分布对样本分布进行逼近, 获取其均值与方差; 模式匹配依据正态分布概率密度函数, 计算其隶属于特定类别的可能性。

3.1 Bagging NB

Bagging 过程是一个独立于具体分类器的过程, 集成 Bagging 与 NB 的算法称之为 Bagging NB。

给定训练集样本 $S\{S_1, S_2, \dots, S_N\}$, 测试集样本 $X\{x_1, x_2, \dots, x_M\}$, 所有的样本数据均包含 k 个属性 $A\{a_1, a_2, \dots, a_k\}$, 共包含 $C\{c_1, c_2, \dots, c_c\}$ 个类别. Bagging NB 可以描述如下:

算法 3: Bagging NB

输入: 训练集 S , 测试集 X , Bagging 轮数 T ;

输出: X 的分类正确率 Rat ;

begin

for $t = 1, 2, \dots, T$ do

$S_i = \text{bootstrap}(S)$;

for $j = 1, 2, \dots, k$ do

$$\mu_j = \sum_{p=1}^N S_i(a_{jp}) / N;$$

$$\sigma_j = \sum_{p=1}^N (S_i(a_{jp}) - \mu_j)^2 / N;$$

endfor.

for $p = 1, 2, \dots, M$ do

for $q = 1, 2, \dots, C$ do

$$CTR_{pq} = \sum_{h=1}^k -\log(\sigma_j) - (\mu_j - x_p(a_k))^2 / (2\sigma^2);$$

endfor

$R_p = \max(CTR_p)$

endfor

$R_t = \text{CntMax}(R_p)$;

endfor.

$Rat = (1 - \text{MisClass}(R_t)) / N$;

end.

通过算法可以看出, Bagging 过程是一个并行过程, 多个 NB 之间彼此独立, 互不影响. 通过多算法独立运行的判定结果, 可以获取最终的群体推荐.

3.2 AdaBoosting NB

与 Bagging N 算法不同, AdaBoosting 与 NB 的集成是一串行过程, 每一次迭代的执行结果, 均会影响次轮的迭代. 同时, 对于 N 条测试数据集样本, 每个样本独立的拥有一个权重值, 记为 $W\{w_1, w_2, \dots, w_n\}$, AdaBoosting NB 算法可以描述如下:

算法 4: AdaBoosting NB

输入: 训练集 S , 测试集 X , Boosting 轮数 T ;

输出: X 的分类正确率 Rat ;

begin

$w_1 = w_2 = \dots = w_n = 1/N$;

for $t = 1, 2, \dots, T$ do

for $j = 1, 2, \dots, k$ do

$$\mu_j = \sum_{p=1}^N S(a_{jp}) \times w_p;$$

$$\sigma_j = \sum_{p=1}^N (S_i(a_{jp}) - \mu_j)^2 \times w_p;$$

endfor.

$$E = \sum_{S \text{ Misclassified}} W_j;$$

$$\beta_t = E / (1 - E);$$

if $(\text{MisClassified}(S_i)) w_i = w_i$;

else $w_i = \beta_t * w_i$;

Normalization(W);

for $p = 1, 2, \dots, M$ do

for $q = 1, 2, \dots, C$ do

$$CTR_{pq} = \sum_{h=1}^k -\log(\sigma_j) - (\mu_j - x_p(a_k))^2 / (2\sigma^2);$$

endfor

$R_p = \max(CTR_p)$

endfor

$R_t = \text{CntMax}(\log(1/\beta_t) * R_p)$;

endfor.

$Rat = (1 - \text{MisClass}(R_t)) / N$;

end.

算法 4 过程中, 有四点需要说明: 其一, 所有样本的权重在开始时均为 $1/N$, 随着迭代的进行, 错误样本的权重保持不变, 正确分类样本的权重降低, 从而突出了错分样本在总体中的比重, 使得算法朝着有利于判别错误样本的方向转变; 其二, NB 算法本身是适应样本加权的, 只需要将权重因子作为计算样本属性均值与方差的参数, 即可在不改变 NB 原理的前提下, 集成 AdaBoosting 算法; 其三, T 轮迭代所产生的弱分类器, 其分类判别结果均作为最终投票的一部分, 权值为 $\log(1/\beta_t)$, β_t 源于样本的错分率, β_t 越大, 则该轮迭代的结果对最终分类的影响越低; 最后, 算法过程中, 必须保证所有测试集样本的权重之和为 1, 故需要在每轮迭代中进行权值的归一化操作, 对应 Normalization 方法, 其计算方法包含两步: (1) $\text{Sum} = \sum_{i=1}^N w_i$; (2) $w_i = w_i / \text{Sum}$.

4 实验与讨论

可以使用分类正确率或错误率验证分类器的优劣, 本文采用分类正确率指标. 为了在统一数据基础上进行分类效果比较, 研究者已经提供了大量的分类测试数据集, 其中以 UCI 的数据集合^[6]最为著名. 本文的对比实验设计如下:

(1) 采用 4 个 UCI 标准测试数据集, 详细情况见表 1. 数据集合中的数据属性分为连续值和枚举值两种, 对于枚举值, 为了使其参与计算, 以数字 0, 1, 2... 等加以顺序替换;

(2) 多数分类数据集并未区分测试数据与学习数据, 不同的集合划分方式会直接影响分类器的分类正确率. 本文采用 10 轮 10 折交叉纠错进行分

表1 测试数据详细信息

数据集	样本数	连续值	枚举值	类别数
breastc	286	9	0	2
diabetes	768	0	5	2
hepatitis	155	13	6	2
iris	150	4	0	3

类验证,基本做法是:将数据集随机分为大小相同(或近似相同)的10组,称为10折,取一组作为测试数据,其余9组作为学习数据,依次将每组数据作为测试数据一次,称作一轮,迭代10次,即可得到尽可能消除随机性的分类测试结果;

(3) 在4个测试数据集上分别进行NB、Bagging NB和AdaBoosting NB分类实验,统计分类正确率;

(4) Bagging的轮数和Boosting的轮数 T 均取值50.

实验结果如图3所示,可以看出,Bagging NB的表现要好于NB,对于四个数据集,breastc和hepatitis集合上Bagging NB要优于NB,在diabetes和iris上两者几乎相当.Bagging的过程是一个并行过程,并未改变数据集本身的样本分布,可以说明,diabetes和iris的样本分布均匀,多次重复抽样未获得更好结果;AdaBoosting NB的表现不同于Bagging NB,对于diabetes和hepatitis,AdaBoosting NB的表现要优于NB,而对于另外两个数据集,特别是breastc,其表现远逊于NB算法.AdaBoosting是一种突出错分样本的串行投票算法,可以说明,breastc中存在部分奇异值样本,对其的突出不但未提高分类器的效能,反而使得NB的均值和方差与实际情况背离较大,造成较高的误分率.

5 结束语

本文首先介绍了Bagging和AdaBoosting算法的实现过程,而后结合朴素贝叶斯算法,给出了Bagging NB和AdaBoosting NB的集成实现.通过实验,可以得到结论:Bagging算法提供了重复抽样的多个弱分类器,其分类准确率不会明显低于单一的

朴素贝叶斯方法;AdaBoosting的原理更加偏向于较

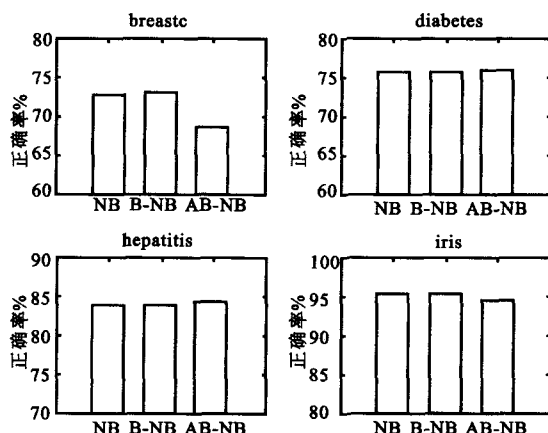


图3 NB, Bagging NB (B-NB) 与 AdaBoosting NB (AB-NB) 的分类正确率对比

为奇异的数据,可能会对分类器算法造成负面影响.下一步的工作将注重决策树、神经网络等分类算法与Bagging和AdaBoosting的集成研究.

参考文献:

- [1] Frank E, Trigg L, Holmes G, et al. Technical note: naive bayes for regression[J]. Machine Learning, 2000, 41(1): 5-26.
- [2] Quinlan J. C4.5: Programs for machine learning[M]. San Francisco, Morgan Kaufmann, 1993.
- [3] Breiman L, Bagging predictors[J]. Machine Learning, 1996(24): 123-140.
- [4] Bauer E, Kohavi R. An empirical comparison of voting classification algorithms: bagging, boosting, and variants[J]. Machine Learning, 1999(36): 105-139.
- [5] 王刚. 基于AdaBoost和PCA的动态人脸考勤技术研究[J]. 微电子学与计算机, 2008, 25(4): 188-191.
- [6] Blake C L, Merz C J. UCI repository of machine learning databases[EB/OL]. [2009-12-10]. <http://www.ics.uci.edu/mllearn/MLRepository.html>.

作者简介:

李晓波 女, (1971-), 硕士, 讲师. 研究方向为智能算法、软件工程.