Jackknife，Bootstraping, bagging, boosting, AdaBoosting, Rand forest 和 gradient boosting的区别

原创 2012年07月03日 20:09:59 15426

nife，Bootstraping, bagging, boosting, AdaBoosting, Rand forest 和 gradient boosting

卡语，我经常搞混淆，现在把它们放在一起，以示区别。(部分文字来自网络，由于是之前记的笔记，忘记来源了，特此向作者抱歉)

Bootstraping: 名字来自成语“pull up by your own bootstraps”，意思是依靠你自己的资源，称为自 它是一种有放回的抽样方法，它是非参数统计中一种重要的估计统计量方差进而进行区间估计的统计方法。其核心思想和基本步骤如下：

（1） 采用重抽样技术从原始样本中抽取一定数量（自己给定）的样本，此过程允许重复抽样。
（2） 根据抽出的样本计算给定的统计量T。
（3） 重复上述N次（一般大于1000），得到N个统计量T。
（4） 计算上述N个统计量T的样本方差，得到统计量的方差。

应该说Bootstrap是现代统计学较为流行的一种统计方法，在小样本时效果很好。通过方差的估计可以构造置信区间等，其运用范围得到进一步延伸。

Jackknife： 和上面要介绍的Bootstrap功能类似，只是有一点细节不一样，即每次从样本中抽样时候只是去除几个样本（而不是抽样），就像小刀一样割去一部分。

(pku， sewm，shinningmonster.)

==========

下列方法都是上述Bootstraping思想的一种应用。

bagging：bootstrap aggregating的缩写。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个训练样本组成，某个初始训练样本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，…
…h_n ，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。

[训练R个分类器f_i，分类器之间其他相同就是参数不同。其中f_i是通过从训练集合中(N篇文档)随机取(取后放回)N次文档构成的训练集合训练得到的。对于新文档d，用这R个分类器去分类，得到的最多的那个类别作为d的最终类别。]

boosting: 其中主要的是AdaBoost（Adaptive Boosting）。初始化时对每一个训练例赋相等的权重1 / 然后用该学算法对训练集训练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,…, h_m , 其中h_i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。
（类似Bagging方法，但是训练是串行进行的，第k个分类器训练时关注对前k-1分类器中错分的文档，即不是随机取，而是加大取这些文档的概率。）
（pku， sewm，shinningmonster.）

Bagging与Boosting的区别：二者的主要区别是取样方式不同。Bagging采用均匀取样，而Boosting根据错误率来取样，因此Boosting的分类精度要优于Bagging。Bagging的训练集的选择是随机的，各轮训练集之间相互独立，而Boostlng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，而Boosting是有权重的；Bagging的各个预测函数可以并行生成，而Boosting的各个预测函数只能顺序生成。对于象神经网络这样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。
bagging和boosting都可以有效地提高分类的准确性。在大多数数据集中，boosting的准确性比bagging高。在有些数据集中，boosting会引起退化--- Overfit。
Boosting思想的一种改进型**AdaBoost**方法在邮件过滤、文本分类方面都有很好的性能。

gradient boosting（又叫Mart, Treenet）：Boosting是一种思想，Gradient Boosting是一种实现Boosti

xianlingmao

原创 27 粉丝 980 喜欢 18

等级： 博客 5 访问量： 108
积分： 4764 排名： 7254

失函数持续的下降，则说明我们的模型在不停的改进，而最好的方式就是让损失函数在其梯度（Gradient）的方向上下降。

Rand forest： 随机森林，顾名思义，是用随机的方式建立一个森林，森林里面有很多的决策树组成，随机森林的每一棵决策树之间是没有关联的。在得到森林之后，当有一个新的输入样本进入的时候，就让森林中的每一棵决策树分别进行一下判断，看看这个样本应该属于哪一类（对于分类算法），然后看看哪一类被选择最多，就预测这个样本为那一类。 在建立每一棵决策树的过程中，有两点需要注意 - 采样与完全分裂。首先是两个随机采样的过程，random forest对输入的数据要进行行、列的采样。对于行采样，采用有放回的方式，也就是在采样得到的样本集合中，可能有重复的样本。假设输入样本为N个，那么采样的样本也为N个。这样使得在训练的时候，每一棵树的输入样本都不是全部的样本，使得相对不容易出现over-fitting。然后进行列采样，从M个feature中，选择m个(m << M)。之后就是对采样之后的数据使用完全分裂的方式建立出决策树，这样决策树的某一个叶子节点要么是无法继续分裂的，要么里面的所有样本的都是指向的同一个分类。一般很多的决策树算法都一个重要的步骤 - 剪枝，但是这里不这样干，由于之前的两个随机采样的过程保证了随机性，所以就算不剪枝，也不会出现over-fitting。 按这种算法得到的随机森林中的每一棵都是很弱的，但是大家组合起来就很厉害了。可以这样比喻随机森林算法：每一棵决策树就是一个精通于某一个窄领域的专家（因为我们从M个feature中选择m让每一棵决策树进行学习），这样在随机森林中就有了很多个精通不同领域的专家，对一个新的问题（新的输入数据），可以用不同的角度去看待它，最终由各个专家，投票得到结果。

Rand forest与bagging的区别：1）. Rand forest是选与输入样本的数目相同多的次数（可能一个样本会被选取多次，同时也会造成一些样本不会被选取到），而bagging一般选取比输入样本的数目少的样本；2）. bagging是用全部特征来得到分类器，而rand forest是需要从全部特征中选取其中的一部分来训练得到分类器； 一般Rand forest效果比bagging效果好！

pku， sewm，shinningmonster.