数据科学基础第四次作业

王晨曦

2018 年 5 月 15 日

1 问题重述

研究模型的性能及改善模型的方法。

- 选择一个现成的数据集或自己创建一个数据集（如利用 make_classification）；
- 选择一个模型（如 kNN, 回归, 决策树...）；
- 探索模型单独工作的性能；
- 探索模型组合工作的性能（bagging 或 boosting）。

2 实验设置

本章首先介绍数据集的选取方法，接下来列出了实验中用到的模型，最后介绍了评价模型性能用到的指标。所介绍的参数是大部分实验默认的参数，对于部分特殊实验，会在稍后的章节中指明参数变化。实验代码保存在了 *Jupyter Notebook* 中。

2.1 数据集选取

本次实验采用 *sklearn* 库中的 *make_classification()* 函数自建二分类数据集，生成 1000 个 30 维的样本，其中 500 个样本用于训练，另外 500 个样本用于验证。为了加大学习的难度，数据集的噪声率设定为 0.2，后面部分实验会对数据集噪声率作出改动。

2.2 模型选择

在大部分实验中，我们均采用决策树作为实验模型，模型深度为 5，Boosting 和 Bagging 算法的模型个数设为 10。在最后的拓展实验中，我们分别以决策树、KNN、SVM、Logistic 回归和深度森林作为基础模型，结合 Boosting 和 Bagging 进行对比。

2.3 评价指标

本次实验用两种指标来衡量模型的性能：正确率和 F1 分数，详细介绍如下。

2.3.1 正确率

模型的正确率是指在所有样本中分类正确的样本所占的比例，对于一个分类模型来说，正确率是其最基础的性能度量，但它有很多不足之处，无法全面反映一个模型的性能。因此很多时候还需要有其他指标去评价模型的优劣性。

2.3.2 F1 分数

用 TP、FP 和 FN 分别代表真阳性、假阳性和假阴性样本的个数，则模型的精确率 P 和召回率 R 可分别表示为

$$P = \frac{TP}{TP + FP}$$
$$R = \frac{TP}{TP + FN}$$

对于复杂问题来说精确率和召回率有着不可调和的矛盾，精确率越高，往往召回率就越低，反之亦如此。为了能综合反映模型的性能，定义 F1 分数如下：

$$F1 = 2 \times \frac{PR}{P + R}$$

可以看出，F1 分数是精确率和召回率的调和平均，因此只有当二者均很高时 F1 分数才会较高；若其中有一方明显低于另一方，F1 分数就会大打折扣。

3 问题解答

这一章分为五个实验，首先分析了模型单独工作和组合工作时的性能，并进行对比，接着探索了组合模型在分类器数量和数据集噪声率变化时的表现，最后进行了拓展实验，在五种不同的模型上进行测试，对比其预测能力。

3.1 模型单独工作的性能

以决策树为基础模型在训练集上进行训练，并且分别在训练集和验证集上进行测试，最终结果见表1。注意，这里的决策树模型深度为 5，为了验证深度为 5 时模型的合理性。我们可以对深度进行遍历，依次在同样的训练集上训练，对比模型的正确率，结果见图1。

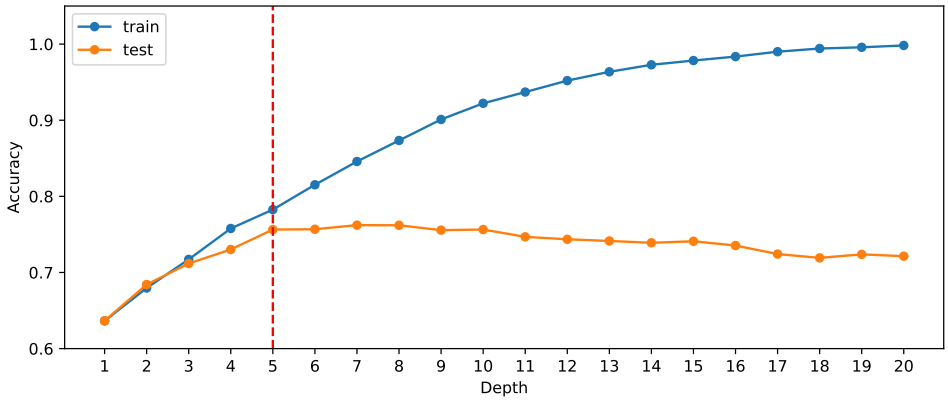

图 1: 决策树深度对性能的影响

从图1中可以看出，随着决策树深度增加，训练集的正确率持续升高，但测试集的效果先增后减，这说明太深的模型会出现过拟合现象。当深度为 5 时，验证集和训练集的正确率相近，且验证集的效果接近峰值，故采用深度为 5 的模型进行后续的验证和对比是合理有效的。

3.2 模型组合工作的性能

对于同样的模型参数，分别使用 Boosting 和 Bagging 改进模型的性能，最终模型在训练集和验证集上的表现见表1。加粗的结果表示在同条件下为最优结果。

表 1: 单独模型与组合模型对比

	训练集				验证集			
	正确率	精确率	召回率	F1 分数	正确率	精确率	召回率	F1 分数
单独模型	78.26%	0.78	0.78	0.78	75.64%	0.76	0.76	0.76
Boosting	**87.82%**	**0.88**	**0.88**	**0.88**	80.52%	**0.81**	0.80	0.80
Bagging	83.64%	0.84	0.84	0.84	**80.76%**	**0.81**	**0.81**	**0.81**

从表1中可以发现，不论是 Boosting 还是 Bagging，对模型均有明显改进，在验证集上达到了相近且显著优于单独模型的效果。但是还应该看到，单独模型和 Bagging 不存在明显过拟合的现象，而 Boosting 在训练和验证集上的效果已经有了很大差距。可以想象，如果数据集的噪声率更低的话，Boosting 的过拟合就会更加严重。

3.3 分类器数量对组合模型性能的影响

上一个实验中，组合模型均采用了 10 个分类器的设置，现在我们来看一看分类器数量对组合模型的影响，图2展示了采用不同数量分类器的组合模型经同样的训练后在验证集上的正确率。

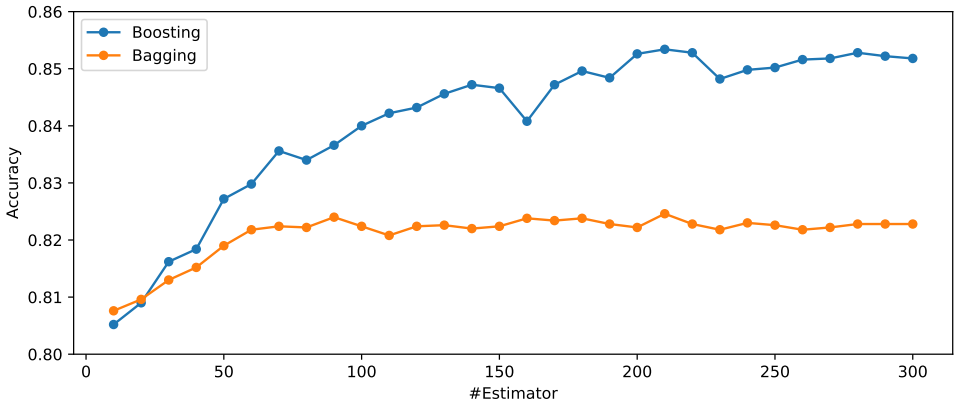

图 2: 分类器数量对性能的影响

从图2中可以看到，Bagging 算法在 60 个分类器以后效果就基本不变，而 Boosting 算法在 200 个分类器以后开始趋于平缓。总体来说，Boosting 算法对模型的提升效果比 Bagging 要好，随着分类器数量的增加，可以看出其提升性能的潜力也很大。但正如前面分析的一样，Boosting 对数据的过拟合很明显，如果测试集的数据分布比例与训练集不同，效果可能就大大降低了。

3.4 数据集噪声率对组合模型性能的影响

数据集中噪声的比例也会影响模型的训练，在实验中，我们可以调整数据集中噪声的比例，进而改变训练的难度，这可以从另一个方面体现模型的泛化能力。对数据集中的噪声比例从 0 到 0.5 进行遍历，测试模型在验证集上的分类正确率，得到的结果如图3所示。

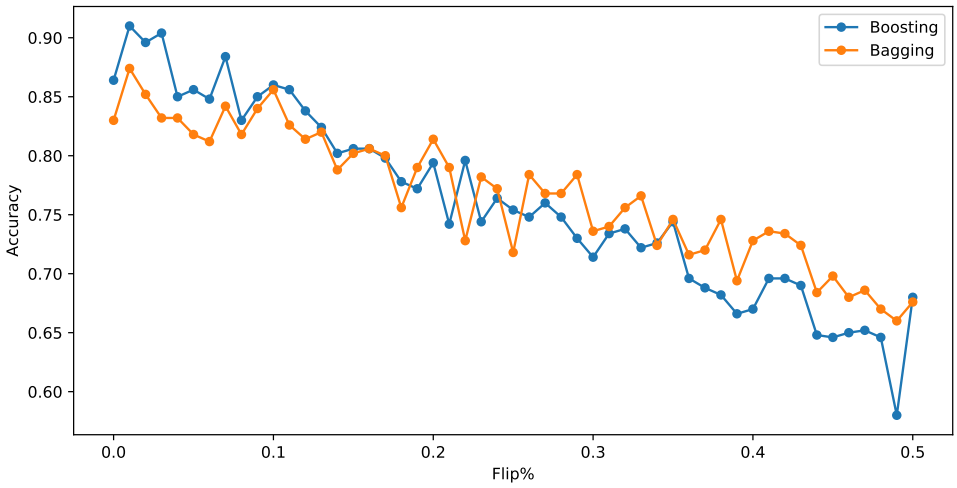

图 3: 数据集噪声率对性能的影响

从图3中可以看到，无论是哪种集成模型，在数据集噪声率逐渐增高的情况下对模型的预测效果都会逐渐下降，我们应该关注的是哪种算法对于不同难度的数据集表现更为稳定。

表2展示了单独模型和两种集成算法的准确率平均值及标准差，其中单独模型的结果作为基准来比较。可以看到，两种集成算法虽然都比单独模型效果要好，但都不如原来稳定，其中 Boosting 的标准差最大，这一点从图3中也能得到验证：噪声率较低时 Boosting 正确率比 Bagging 高，而噪声率较高时却比 Bagging 低，其变化幅度比 Bagging 要剧烈，说明 Bagging 比 Boosting 要更不稳定。

表 2: 模型在不同难度数据集上的效果和稳定程度对比

	正确率平均值	正确率标准差
单独模型	0.672	**0.050**
Boosting	**0.767**	0.079
Bagging	0.760	0.055

3.5 拓展实验：模型性能综合对比

前面的实验均采用决策树作为基础模型，这一节会用不同种类的模型搭配集成算法进行对比，实验结果见图4。图中横坐标为正确率，纵坐标为 F1 分数，从比较的角度来说，模型的位置越靠近右上角，说明其正确率和 F1 分数越高，也即预测效果越好。在图4中，由于 SVM (Boosting) 效果与其他模型太过悬殊，为了方便展示，没有将其绘制在图中。

从实验结果可以看到，KNN 对于生成的数据集预测效果最好，这是由于我们使用的函数生成的数据以欧氏距离和高斯分布为基准，所以用 KNN 直接预测就可以达到很好的效

果。可以发现，除了 SVM (Boosting) 以外，其余模型在加入了集成算法之后均得到了明显的改进，改进效果最明显的是决策树（三个蓝色标记点所在的位置）。值得一提的是，随机森林（紫色标记点所在的位置）对于数据的拟合能力相当之强，比绝大部分集成算法的效果还要好。

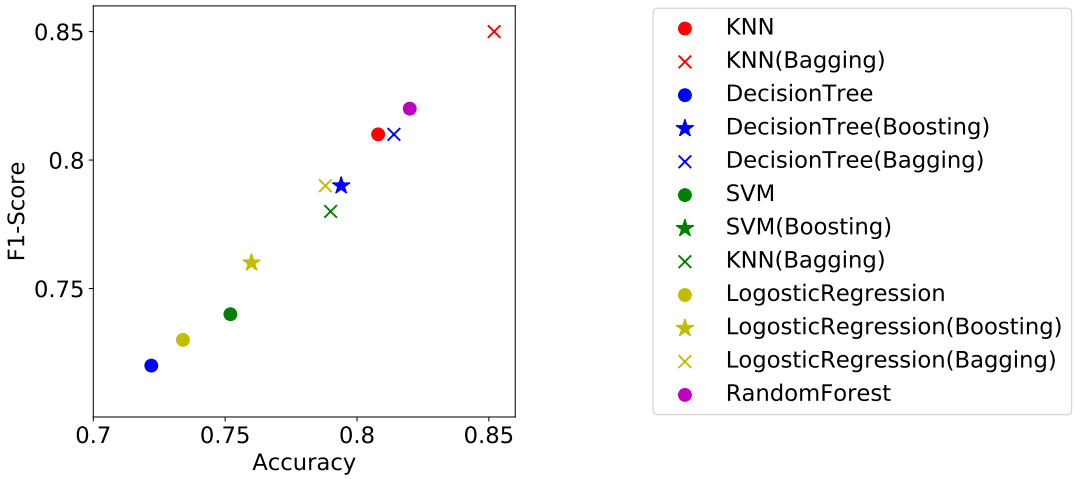

图 4: 模型性能综合对比

我们还能发现所有模型的正确率和 F1 分数都很接近，并且在之前所有的实验中精确率和召回率几乎相等，看似并没有冲突。这跟理论分析的效果有些偏差，其主要原因包括以下两点：

- 自建数据集太过于规范，遵循了欧式距离下的高斯分布，且为简单的二分类。在实际问题中，数据分布往往更为复杂，标签也不只是二类，如图像分割、物体识别等。
- 实验中设置的模型的复杂程度较低，使得其还没有完全学习本来能学习到的数据特征，过拟合也较少。在深度学习中模型的复杂度要远高于机器学习的基础模型，因而数据在模型中得到了更好的表征，模型的性能才能清晰的显露出来。