机器学习Homework 4——随机森林和Bagging方法实验报告

第1节　随机森林的算法框架

算法 1 随机森林算法

输入： 训练样本集$D = \{(x_i, y_i)\}_{i=1}^{N}, x_i \in \mathcal{X}, y_i \in \mathcal{Y}, i = 1, 2, \cdots, N$; 决策树算法$\mathcal{L}$(作为基分类算法); 决策树个数个数$T$；随机选择特征数$k$

输出： 集成分类器f(x)

1. 对$t = 1, 2, \cdots, T$
   (a) 从D利用自助采样法随机抽取N个样本得到D_t。
   (b) 从D_t中根据决策树算法$\mathcal{L}$学习得到基分类器$f_t(x)$，其中在决策树的学习算法中引入随机属性选择，即决策树算法在选择划分特征时，从当前结点对应的所有特征中随机选择k个特征作为候选划分特征，再从这k个特征中选择最优划分特征。
2. 返回集成分类器

$$f(x) = arg\max_{y \in \mathcal{L}} \sum_{t=1}^{T} I(f_t(x) = y)$$

第2节　随机森林和Bagging方法的算法原理对比

随机森林模型是在Bagging方法中以决策树算法作为基学习器，并在决策树的学习算法中引入随机属性选择，与Baggng方法相比，随机属性选择使得随机森林模型多了一层关于划分特征的随机扰动。这有助于增加基学习器的多样性差异，进而提升最终集成学习器的泛化能力。另一方面，随机森林的基学习算法在每次划分结点的时候只

随机考察一个规模为k的特征子集，这使得每个基分类器的训练效率高，但性能相对比无属性扰动的情形有所降低，但随着基学习器数目的增加，随机森林通常会收敛到比Bagging更低的泛化误差。

第3节　随机森林和Bagging方法的数据实验对比

这次实验，我使用python语言，不借助软件包，自己代码实现了Bagging和随机森林两种集成学习算法。在数据集的选择上，我依旧从UCI上选择了Car Evaluation数据集[1] [1]进行实验比较。

第3.1小节　数据集简介

Car Evaluation数据集一共包括1728条数据，其中包含了对于汽车不同特征的离散描述以及对于的汽车评价标签，其中包括6项特征：

1. **buying**: vhigh, high, med, low.
2. **maint**: vhigh, high, med, low.
3. **doors**: 2, 3, 4, 5more.
4. **persons**: 2, 4, more.
5. **lug_boot**: small, med, big.
6. **safety**: low, med, high.

对应的标签为：

- **Values**: unacc, acc, good, vgood.

第3.2小节　随机森林和Bagging方法实验对比

3.2.1　使用不同算法作为基学习器的Bagging方法实验分析

首先我从Bagging方法入手，研究了不同的基学习器使用Bagging方法进行增强的结果Car Evaluation数据集，我将使用决策树作为基学习器的Bagging算法放在随机森林的实验结果中一同讨论，在此处对比的基学习器包括：K近邻算法(k=3)、线性核函数的SVM方法、Radial Basis Function作为核函数的SVM方法、高斯过程、神经网络、朴素贝叶斯方法，讨论在不同的数据量和训练基学习器时不同的采样数据量的情况下，这些模型作为基学习器的Bagging方法在测试集上的准确率表现如下表1所示：

[1]http://archive.ics.uci.edu/ml/datasets/Car+Evaluation

表 1: 不同基学习器的Bagging方法在Car Evaluation上的准确率

训练集占比	采样占比	基学习器					
		KNN	L-SVM	RBF SVM	GP	Neural Net	Naive Bayes
10%	0.5	0.789	0.699	0.699	0.810	0.833	**0.767**
	0.632	**0.803**	0.699	0.699	0.828	0.843	0.763
	0.8	0.801	0.699	0.699	0.828	**0.848**	0.731
	-	0.786	**0.715**	0.699	**0.844**	0.847	0.737
30%	0.5	0.884	0.755	0.713	0.888	0.904	0.729
	0.632	0.888	0.772	0.715	0.888	0.927	**0.739**
	0.8	**0.889**	0.783	0.724	0.892	0.855	0.727
	-	0.870	**0.787**	**0.749**	**0.922**	**0.950**	0.728
50%	0.5	0.899	0.782	0.719	0.896	0.891	**0.735**
	0.632	0.904	0.794	0.726	0.902	0.936	0.729
	0.8	**0.906**	0.795	0.773	0.904	0.927	0.729
	-	0.904	**0.799**	**0.817**	**0.904**	**0.936**	0.728
70%	0.5	0.900	0.779	0.719	0.917	0.767	0.711
	0.632	**0.908**	0.780	0.759	0.888	0.944	**0.717**
	0.8	0.894	0.784	0.805	0.927	**0.950**	0.707
	-	0.892	**0.788**	**0.836**	**0.933**	0.852	0.707
90%	0.5	**0.931**	0.786	0.763	0.901	0.942	0.694
	0.632	0.902	0.786	0.821	0.965	0.942	**0.711**
	0.8	0.919	0.786	0.884	0.971	0.948	0.699
	-	0.884	**0.803**	**0.913**	**0.978**	**0.954**	0.699

[1] "-" 表示不使用Bagging方法，直接用基学习器进行训练和预测

[2] "L-SVM" 表示线性核函数的SVM方法

[3] "GP" 表示高斯过程(Gaussian Process)

[4] 不同训练集占比下不同基学习算法上准确率表现最好的都用粗体表示。

首先考虑相同训练集占比的条件下，不同采样数据占比的影响，KNN和朴素贝叶斯算法通过调节采样的数据占比，相较于不使用Bagging方法，准确率均可以实现较大程度的提升，而且在使用Bagging方法获得准确率提升的情况当中，大多数情况的采样数据占比为0.632，由此验证了课上讲述的自助采样法的有效性。而对于线性核函数的SVM方法、Radial Basis Function作为核函数的SVM方法、高斯过程、神经网络这四种方法而言，除了极少数情况之外，无论这么调整采样占比或是训练集占比，使用Bagging的方法均不能有明显的提升，我认为这其中的原因是这四种基学习算法相

较于KNN和朴素贝叶斯，已经具备了非常强的学习拟合能力。另外，从表中也可以看出，随着训练数据的增多，除了朴素贝叶斯之外，其他五种基学习算法的表现均能够得到较大程度的提升，这很好理解，而朴素贝叶斯算法随着训练数据的增多，在测试集上的准确率反而降低的反常现象，我觉得是因为朴素贝叶斯作为生成式模型优点在于对小规模的数据表现很好，然而由于朴素贝叶斯算法使用了样本属性独立性的假设，如果样本属性有关联时其效果不好，而随着训练样本的增多，样本属性的关联性在数据上体现的愈发明显，因而导致了朴素贝叶斯算法的准确率不升反降。

总体而言，Bagging方法可以在基学习器本身的拟合能力较弱，或者因为数据量不足而导致的基学习器泛化较弱的情况下，通过集成学习的机制，提升模型的表现，而对于那些原本就具有很强的学习拟合及泛化能力的模型而言，Bagging方法所能带来的效果提升有限。

第3.3小节　随机森林和使用决策树作为基学习器的Bagging方法实验对比

在此对比实验当中，随机森林和Bagging方法使用到的决策树基学习器为第二次作业代码实现的在随机森林和使用决策树(CART树)作为基学习器的Bagging方法的实验对比中，我设定采样数据占比为0.632，表2展示了不同训练数据占比的情况下，这两种集成方法和不使用集成方法的决策树算法在测试集上的表现

表 2: 随机森林和使用决策树作为基学习器的Bagging方法实验对比

训练集占比	10%	30%	50%	70%	90%
随机森林	0.817	0.936	0.948	0.952	**0.971**
决策树Bagging	**0.823**	**0.943**	0.946	0.950	0.960
决策树	0.819	0.941	**0.951**	**0.954**	0.965

首先在数据量较小的情况下，结合表1与表2的数据可以看出，当数据量足够的时候，决策树算法在Car Evaluation数据集上有较强的学习拟合及泛化能力。而且，从表2可以发现，当训练集占比较小时(10%、30%)，以决策树作为基学习器的Bagging方法在测试集上的有最高的准确率，这印证了上一小节的结论，即Bagging方法可以在因为数据量不足而导致的基学习器泛化较弱的情况下提升模型的表现。而当训练数据充足时(90%)，随机森林所得的准确率要明显高于基于决策树的Bagging方法和不使用集成学习的决策树算法，这得以与随机森林引入的随机属性选择，增加了基学习器的多样性差异，进一步提高了模型的泛化能力。

总结来说，随机森林在Bagging以决策树作为基学习器的基础上，进一步引入随机属性选择，有助于基学习器的多样性差异，从而进一步提升集成学习的泛化能力。

第4节　总结

在本作业中，我首先给出了随机森林的算法框架，并从算法原理和数据实验两方面对比了随机森林和Bagging方法。在数据实验方面，我不借助软件包，自己代码实现了Bagging和随机森林两种集成学习算法，对于Bagging方法，我对比了多种基学习器使用Bagging方法集成之后在Car Evaluation上的模型表现，并对基于决策树的Bagging和随机森林两种集成学习算法进行了实验比较。

参考文献

[1] M. Bohanec and V. Rajkovic. Knowledge acquisition and explanation for multi-attribute decision making. In 8th Intl Workshop on Expert Systems and their Applications, pages 59–78, 1988.