随机森林：从理论到应用

易凯

西安交通大学人工智能与机器人研究所

西安交通大学社会心理学研究所

西安交通大学软件学院

学号: 2151601053

班级：软件 53 班

邮箱：williamyi96@gmail.com

Abstract

该实验报告首先对随机森林的基本概念（包括决策树生成学习算法，优劣分析以及超参数等）进行了合理的分析，接着在 IRIS 数据集上在与决策树等四种不同算法的比较中说明了随机森林算法的有效性，接着对代码中的实验结果以及不同参数属性进行了说明。最后对全文进行了总结，针对于试验中存在的问题明确了下一步的发展方向。

1 随机森林基本概念

该部分将首先对随机森林的相关概念进行总体介绍，然后说明基于决策树生成随机森林的经典学习算法，接着从宏观上对随机森林的优势以及劣势进行分析，同时额外地对于随机森林使用到的超参数进行合理的评估。

1.1 随机森林基本介绍

随机森林 [1] 是机器学习中使用较为广泛的包含多个决策树的分类器，并且其输出的类别由所有树输出的类别的众数而定，此在一定程度上有效地避免了决策树中易于产生的过拟合问题。

1.2 树生成学习算法

其构造每棵树使用到的学习算法为：

1. 输入特征数目 m，用于确定决策树上一个节点的决策结果 (m « M，M 为特征的数目)；

2. 从 N 个训练样本中以有放回抽样的方式，取样 N 次，形成一个训练集（即 bootstrap 取样），并用未抽到的样本做预测，评估其误差；

3. 对于每一个节点，随机选择 m 个特征，决策树上每个节点的决定都是基于这些特征的。根据这 m 个特征，计算其最佳的分裂方式。

通过上述学习算法，同时保证每棵树都会完整成长而不会剪枝的方式，来纵深生成决策树。

1.3 优劣分析

1. 输入变量具有鲁棒性，分类精度较决策树好 [2]。可以进行大量输入变量的处理，同时对不同的类别提供了高精度的分类；

2. 变量的加权性 [3]。随机森林可以在决定类别时，对变量的重要性进行评估；

3. 缺失数据容忍。随机森林允许在很大一部分数据损失的情况下，不损失分类的精度；

4. 不平衡数据的自调整。对于不平衡的分类数据集，可以通过森林生成过程中的控制来平衡误差。

5. 学习速度快 [4]。随机森林的学习速度相对于深度网络等方法具有很大优势。

1.4 劣势分析

1. 噪声相关性。随机森林在一些噪声较大的分类或者回归任务上会产生过拟合的现象。

2. 属性加权不可置信性。对于有不同取值属性的数据，取值划分较多的属性会对随机森林产生较大的影响，自平衡能力有限。

3. 针对非凸问题性能较差。对于非凸分类或者回归问题，随机森林所表现出的性能较差，需要考虑使用深度神经网络 [5, 6, 7, 8, 9]。

1.5 超参数

随机森林具有三个主要的超参数需要进行调整 [2, 10]，分别为：

1. 节点规模：随机森林不像决策树，每一棵树叶所包含的观测样本数量可能十分少。该超参数的目标是生成树的时候尽可能保持小的偏差；

2. 树的数量：在实践中生成多少棵树是考量的一个重要超参数，一般选择数百棵树进行生成；

3. 预测器采样的数量：一般来说，如果我们一共有 D 个预测期器，那么我们可以在回归任务中使用 D/3 个预测器作为采样数，在分类任务总使用 $\sqrt{D}$ 个预测器作为抽样。

2 数据集测试

2.1 IRIS 数据集

2.1.1 数据集基本描述

该数据集是对鸢尾花进行特征分类的数据集，同时还可以出于简单的需要，选取其中的子集，对三种不同特征的花进行分类。总体上来看，该数据集包含有 150 个数据，其可以分为三类（setosa, versicolor, virginica），每类均分有 50 个数据，每个数据都有 4 个不同的测量花朵类别度量的属性变量，在实验设计的过程中，使用到了四种不同的分类器来进行分类能力的综合评估。

2.1.2 实验说明

2.1.3 实验代码

```
import numpy as np
import os
import cv2
import matplotlib.pyplot as plt

from matplotlib.colors import ListedColormap

# load IRIS data set
from sklearn.datasets import load_iris
from sklearn.ensemble import (RandomForestClassifier, ExtraTreesClassifier,
                              AdaBoostClassifier)
from sklearn.tree import DecisionTreeClassifier


n_classes = 3    # the number of total classes
n_estimators = 30    # the number of estimators length
cmap = plt.cm.RdYlBu
plot_step = 0.02
plot_step_coarser = 0.5
```

```
RANDOM_SEED = 13

iris = load_iris()  # load IRIS dataset

plot_idx = 1

# load different models
models = [DecisionTreeClassifier(max_depth=None),
          RandomForestClassifier(n_estimators=n_estimators),
          ExtraTreesClassifier(n_estimators=n_estimators),
          AdaBoostClassifier(DecisionTreeClassifier(max_depth=3),
                             n_estimators=n_estimators)]

for pair in ([0, 1], [0, 2], [2, 3]):
    for model in models:
        X = iris.data[:, pair]
        y = iris.target

        idx = np.arange(X.shape[0])
        np.random.seed(RANDOM_SEED)
        np.random.shuffle(idx)
        X = X[idx]
        y = y[idx]

        mean = X.mean(axis=0)
        std = X.std(axis=0)
        X = (X - mean) / std

        model.fit(X, y)

        scores = model.score(X, y)
        model_title = str(type(model)).split(
            ".")[-1][:-2][:-len("Classifier")]

        model_details = model_title
        if hasattr(model, "estimators_"):
```

```
        model_details += " with {} estimators".format(
            len(model.estimators_))
    print(model_details + " with features", pair,
          "has a score of", scores)

    plt.subplot(3, 4, plot_idx)
    if plot_idx <= len(models):
        plt.title(model_title, fontsize=9)

    x_min, x_max = X[:, 0].min() - 1, X[:, 0].max() + 1
    y_min, y_max = X[:, 1].min() - 1, X[:, 1].max() + 1
    xx, yy = np.meshgrid(np.arange(x_min, x_max, plot_step),
                         np.arange(y_min, y_max, plot_step))

    if isinstance(model, DecisionTreeClassifier):
        Z = model.predict(np.c_[xx.ravel(), yy.ravel()])
        Z = Z.reshape(xx.shape)
        cs = plt.contourf(xx, yy, Z, cmap=cmap)
    else:
        estimator_alpha = 1.0 / len(model.estimators_)
        for tree in model.estimators_:
            Z = tree.predict(np.c_[xx.ravel(), yy.ravel()])
            Z = Z.reshape(xx.shape)
            cs = plt.contourf(xx, yy, Z, alpha=estimator_alpha, cmap=cmap)

    xx_coarser, yy_coarser = np.meshgrid(
        np.arange(x_min, x_max, plot_step_coarser),
        np.arange(y_min, y_max, plot_step_coarser))
    Z_points_coarser = model.predict(np.c_[xx_coarser.ravel(),
                                           yy_coarser.ravel()]
                                           ).reshape(xx_coarser.shape)
    cs_points = plt.scatter(xx_coarser, yy_coarser, s=15,
                            c=Z_points_coarser, cmap=cmap,
                            edgecolors="none")

    plt.scatter(X[:, 0], X[:, 1], c=y,
```

```
                    cmap=ListedColormap(['r', 'y', 'b']),
                    edgecolor='k', s=20)
        plot_idx += 1

# print the final result
plt.suptitle("Classifiers␣on␣feature␣subsets␣of␣the␣Iris␣dataset", fontsize=12)
plt.axis("tight")
plt.tight_layout(h_pad=0.2, w_pad=0.2, pad=2.5)
plt.show()
```

代码参数分析：

参数类别	赋值	说明
n_estimators	100	决策树个数的度量
bootstrap	True	是否为有放回采样
oob_score	False	oob (out of band) 带外数据，也即某次决策树训练中没有被 bootstrap 选中的数据, 应用于交叉验证
warm_start	False	热启动，决定是否在之前对应类基础之上增值
class_weight	None	不同 label 的权值

2.2 实验结果分析

上述代码完成的是使用随机森林等四种经典且常用的分类算法进行对比实验，实验结果的分析得到，第一行生成的图片使用到了 sepal length 以及 sepal width 特征，第二行生成的图片使用到了 petal length 以及 sepal length 特征，第三行生成的图片使用到了 petal length 以及 petal length 特征。具体的结构如下图 1 所示：

实验命令行输出结果为图 2：

3 总结与未来发展

通过随机森林的相关实验，结合赵老师课上所讲的相关内容，对于随机森林的相关基础概念，以及其存在的优势以及劣势有了较为细致与深入的分析，为下一步进一步地学习奠定了坚实的基础。同时，基于 IRIS 数据集随机森林算法的实现，对于随机森林有了更为深入的理解，同时基于实现层面对其相关方法明确了下一步发展的基本方向。通过对实验结果的相关分析，针对随机森林存在的问题制定了下一步的提升可能方向。

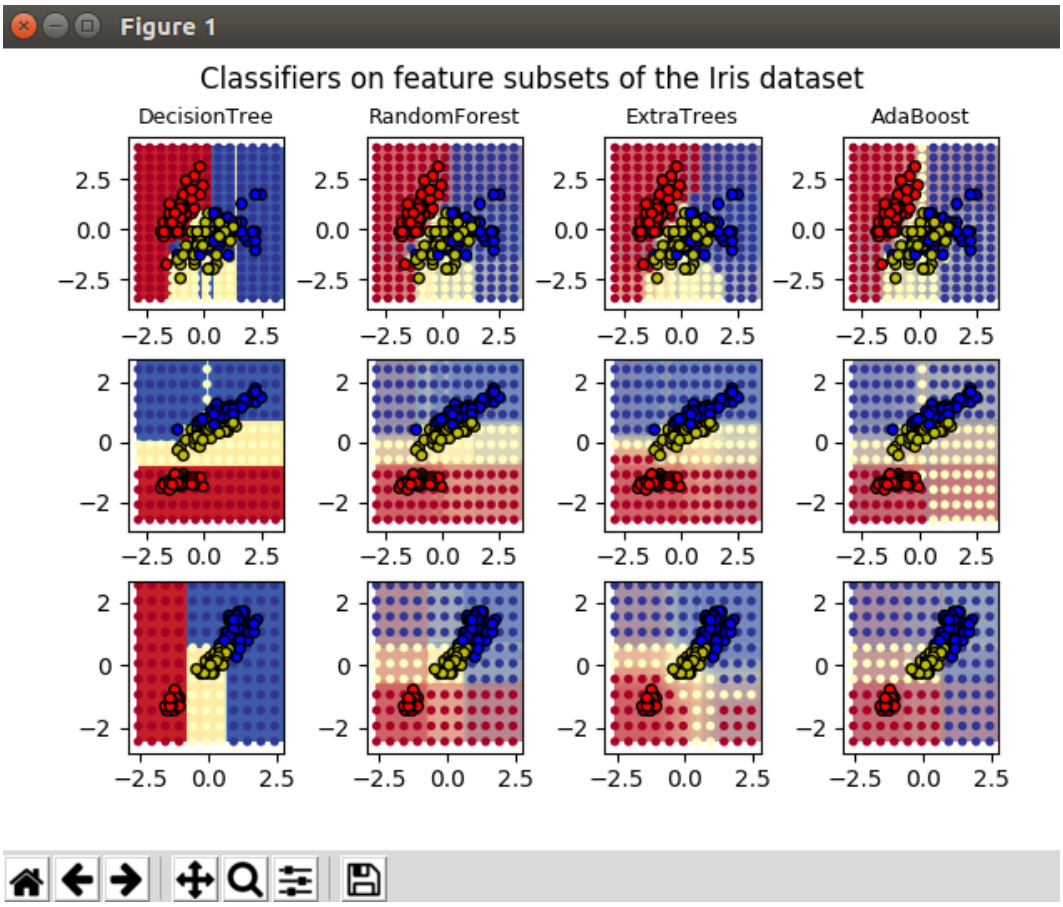

Figure 1: 四中经典分类方法在 IRIS 数据集上的对比试验

```
(tensorflow3) kyi@cver:~/Desktop/Research18/论文/机器学习作业--随机森林/code/random forest$ python IRIS-RF.py
DecisionTree with features [0, 1] has a score of 0.926666666667
RandomForest with 30 estimators with features [0, 1] has a score of 0.926666666667
ExtraTrees with 30 estimators with features [0, 1] has a score of 0.926666666667
AdaBoost with 30 estimators with features [0, 1] has a score of 0.84
DecisionTree with features [0, 2] has a score of 0.993333333333
RandomForest with 30 estimators with features [0, 2] has a score of 0.993333333333
ExtraTrees with 30 estimators with features [0, 2] has a score of 0.993333333333
AdaBoost with 30 estimators with features [0, 2] has a score of 0.993333333333
DecisionTree with features [2, 3] has a score of 0.993333333333
RandomForest with 30 estimators with features [2, 3] has a score of 0.993333333333
ExtraTrees with 30 estimators with features [2, 3] has a score of 0.993333333333
AdaBoost with 30 estimators with features [2, 3] has a score of 0.993333333333
```

Figure 2: IRIS 数据集上不同分类方法所产生的命令行输出

总的来说，随机森林算法作为一类极为经典的机器学习方法，在多个不同的领域有着较为广泛的应用，尽管随着深度网络的这一波高潮的到来，其地位有着一定程度的降低，但是实际上其很多的思想仍然是很具有参考作用的。从研究的角度，我想要引用不确定度来对随机森林的稳定可解性进行度量，主要使用到的方法是贝叶斯变分推理 [11]（Bayesian Variational Inference）

我们在此假设框架下目标为构建一个复杂的函数（函数形式可以未知），对后验概率进行近似估计。具体思想是：将所有观测变量的集合记为 X (也就是所有不同离散时间点的压力指标)，将所有的 hidden variables 记为集合 Z，使用一定的概率模型（如高斯模型等）确定联合概率分布 $p(X, Z)$，我们的目标是找到对后验概率分布 $p(Z|X)$ 以及模型 evidence $p(X)$ 的近似。通过对数边缘分解，得到：

$$\ln p(X) = \zeta(q) + KL(q||p) \tag{1}$$

其中，

$$\zeta(q) = \int q(Z) \ln \left\{ \frac{p(X, Z)}{q(Z)} \right\} dz \tag{2}$$

$$KL(q||p) = -\int q(Z) \ln \left\{ \frac{p(Z|X)}{q(Z)} \right\} dz \tag{3}$$

通过关于概率分布 q(Z) 的最优化来使下界 $\zeta(q)$ 达到最大值，这等于最小化 KL 散度。

使用分解概率分布，我们可以得到:

$$\begin{aligned} \zeta(q) &= \int \prod_i q_i \left\{ \ln p(X, Z) - \sum_i \ln q_i \right\} dZ \\ &= \int q_i \left\{ \int \ln p(X, Z) \prod_{i \neq j} q_i \, dZ_i \right\} dZ_j - \int q_j \ln q_j dZ_j + C \\ &= \int q_j \ln \tilde{p}(X, Z) dZ_j - \int q_j \ln q_j dZ_j + C \end{aligned} \tag{4}$$

其中：

$$\ln \tilde{p}(X, Z_j) = \mathbb{E}_{i \neq j}[\ln p(X, Z)] + C \tag{5}$$

则我们得到最优解 $q_j^*(Z_j)$ 的一般表达式形式为：

$$\ln q_j^* = \mathbb{E}_{i\neq j}[\ln p(X,Z)] + C, \quad \mathbb{E}_{i\neq j}[\ln p(X,Z)] = \int \ln p(X,Z) \prod_{i\neq j} q_i dZ_i \tag{6}$$

因此，我们可以根据迭代学习的方式对 $q_j(Z_j)$ 进行优化直到最优解，通过最优解对之后的对应输入时间进行预测。

如假设使用一元高斯分布，则有似然函数为：

$$p(\mathcal{D}|\mu,\tau) = (\frac{\tau}{2\pi})^{\frac{N}{2}} \exp\left\{-\frac{\tau}{2}\sum_{n=1}^{N}(x_n-\mu)^2\right\} \tag{7}$$

其中，引入 μ 和 τ 的共轭先验分布，形式为：

$$p(\mu|\tau) = \mathcal{N}(\mu|\mu_0,(\lambda_0\tau)^{-1}), \quad p(\tau) = Gam(\tau|a_0,b_0) \tag{8}$$

通过化归得到：

$$\begin{aligned} \ln q_\mu^*(\mu) &= \mathbb{E}_\tau[\ln p(\mathcal{D}|\mu,\tau) + \ln p(\mu|\tau)] + C \\ &= -\frac{\mathbb{E}[\tau]}{2}\left\{\lambda_0(\mu-\mu_0)^2 + \sum_{n=1}^{N}(x_n-\mu)^2\right\} + C \end{aligned} \tag{9}$$

此外，还可以考虑高斯变分混合以及其他等多种更为复杂的方式进行模型构建。

以此为依据来构造基于不确定度度量的更加鲁棒的随机森林决策树生成。

致谢

感谢赵老师的指导以及点拨，感谢帮助过我的师兄师姐，感谢开源社区提供无私的答疑解惑的平台。

References

[1] L. Breiman. Random forest. *Machine Learning*, 45:5–32, 2001.

[2] M. Pal. Random forest classifier for remote sensing classification. *International Journal of Remote Sensing*, 26(1):217–222, 2005.

[3] Kellie J. Archer and Ryan V. Kimes. Empirical characterization of random forest variable importance measures. *Computational Statistics & Data Analysis*, 52(4):2249–2260, 2008.

[4] J. Ham, Yangchi Chen, M. M. Crawford, and J. Ghosh. Investigation of the random forest framework for classification of hyperspectral data. *IEEE Transactions on Geoscience & Remote Sensing*, 43(3):492–501, 2005.

[5] Alex Krizhevsky, Ilya Sutskever, and Geoffrey E Hinton. Imagenet classification with deep convolutional neural networks. In *Advances in neural information processing systems*, pages 1097–1105, 2012.

[6] Karen Simonyan and Andrew Zisserman. Very deep convolutional networks for large-scale image recognition. *arXiv preprint arXiv:1409.1556*, 2014.

[7] Christian Szegedy, Wei Liu, Yangqing Jia, Pierre Sermanet, Scott Reed, Dragomir Anguelov, Dumitru Erhan, Vincent Vanhoucke, Andrew Rabinovich, et al. Going deeper with convolutions.

[8] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 770–778, 2016.

[9] Gao Huang, Zhuang Liu, Laurens Van De Maaten, and Kilian Q. Weinberger. Densely connected convolutional networks. In *CVPR*, 2017.

[10] Svetnik Vladimir, Liaw Andy, Tong Christopher, Culberson J. Christopher, P. Sheridan Robert, and Bradley P Feuston. Random forest: a classification and regression tool for compound classification and qsar modeling. *Journal of Chemical Information & Computer Sciences*, 43(6):1947, 2003.

[11] David M. Blei, Alp Kucukelbir, and Jon D. Mcauliffe. Variational inference: A review for statisticians. *Journal of the American Statistical Association*, 2016.