Implementation of Random Forests

Alejandro Suárez Hernández

Abstract—**This report presents our implementation of Random Forests for classification and presents some exploratory results. Namely, we are interested in the performance of our classifier under different settings, varying the number of features considered randomly at each decision stump and the size of the ensemble. Another interesting aspect of Random Forests that we will review in this document is that they perform implicit feature selection. We can take advantage of this to rank the input attributes according to their relevance predicting the output class.**

I. Introduction

RANDOM forests is an ensemble method for both regression and classification. It essentially consists in learning several decision trees from the data, introducing some source of randomness in the learning process so they are not equal. When classifying a new instance, the forest aggregates the predictions of all the trees into a single answer. In the classification case, the aggregation method is typically the mode (i.e. the most frequently predicted class). In regression, the average of all the guesses is a common choice. In this document, we focus on the classification problem.

These are the most frequent techniques to introduce stochasticity in the learning process:

- **Bootstrap aggregating:** also known as bagging. It is a general method that consists in training each learner (in our case, each tree) with a random sample of the original train data.
- **Feature bagging:** when learning, select the best splitting feature from a random sample of features. Each selected feature is splitted optimally according to some information criterion (entropy, Gini impurity, error...). The number of random features to consider is a parameter of the algorithm. This technique is explored in this document.
- **Random splits:** take randomization even one step further making the local splits random as well. That is, the information criterion is used exclusivelly to select the best randomly splitted feature. This technique has been adopted by the ExtraTrees (Extremely Randomized Trees) algorithm.

This document revolves around a custom implementation of Random Forests. It incorporates feature bagging and accepts several parameters in order to tweak its performance for each data set. More specifically:

- **Number of selected features** (F)**:** the number of candidate features chosen randomly at each node for evaluation. F is clipped if the number of available features is smaller.
- **Number of learners** (M)**:** the number of trees in the ensemble.
- **Minimum number of instances to split** (N)**:** the learner will continue splitting the current node if the number of instances that have arrived to it is at least N.
- **Split quality criterion:** the implementation accepts three criterions: (1) information gain (or entropy minimization); (2) Gini impurity coefficient; (3) classification error or $1 - p_{max}$ where p_max is the probability of the most frequent class in the split.

TABLE I
Data sets used for testing our RF implementation

Data set	Records	Nom. Attr.	Num. Attr	Classes	Missing(%)
Audiology	226	69	0	24	2
Credit	690	9	6	2	0.6
Hepatitits	155	13	6	2	5.7
Votes	435	16	0	2	5.6
Iris	150	0	4	3	0
Chess	3196	36	0	2	0
Lenses	24	4	0	3	0
Soybean	47	35	0	4	0
Splices	3190	60	0	3	0
Zoo	101	16	1	7	0

Our implementation offers other features like the possibility of storing the learnt ensembles in a JSON file, and visualizing them via Graphviz. We refer the reader to the `README.md` file in this repository in order to know more about the technical details of our application [1].

In the next section we explore the prediction power of our classifier with different parametrizations. In addition, we analyze its ability to rank features based on their frequence of apparition as criterion in the decision stumps.

II. Experiments

The tested data sets are shown in Table I. We aim mostly at studying the accuracy of the ensemble under different selections of M (number of trees) and F (number of random features). For the sake of simplicity, the rest of parameters will be fixed: $N = 4$; and the quality metric is the Giny impurity. We invite the reader to experiment further with our application.

A. Accuracy

We have tested our ensemble model for $M \in \{50, 100\}$ and for $F \in \{1, 3, \lfloor \log_2(N_{attr} + 1) \rfloor, \lceil \sqrt{N_{attr}} \rceil\}$ (discarding repeated values of F).

The results are as follows:

- **Audiology:** shown in Table II. There is a slight improvement for increasing M and F.
- **Credit:** shown in Table III. The accuracy does not seem to be affected.

[1] Github: https://github.com/sprkrd/randomforest

- **Hepatitis:** shown in Table IV. Surprisingly, the accuracy seems to be higher for the smallest value of F.
- **Votes:** shown in Table VI. The best accuracy has been obtained with $F = 4$ and 100 learners.
- **Chess:** shown in Table VII. The difference between the best and the worst configuration is almost a 3% with a very low deviation.
- **Lenses:** shown in Table VIII. Perhaps not a very relevant data set because of the low number of examples. The deviation is too high to draw conclusions.
- **Soybean:** shown in Table IX. Perfect classification with all the configurations.
- **Splice:** shown in Table X. One of the data sets in which the parameters choice have the most notable effect, jumping from a worst case accuracy of 87.81% ($M = 50$ and $F = 1$) to 96.39% ($M = 100$ and $F = 8$). The deviation is also greatly reduced, suggesting that this is indeed a very stable configuration.
- **Zoo:** shown in Table XI. The parameters do not have a very noticeable effect.

TABLE II
ACCURACY RESULTS FOR AUDIOLOGY DATA SET

M\F	**1**	**3**	**7**	**8**
50	73.78 ± 6.19	74.67 ± 7.65	74.22 ± 7.52	74.67 ± 9.06
100	72.44 ± 7.77	75.11 ± 6.50	74.67 ± 8.15	76.00 ± 8.93

TABLE III
ACCURACY RESULTS FOR CREDIT APPROVAL DATA SET

M\F	**1**	**3**	**4**
50	86.52 ± 2.36	85.65 ± 1.86	86.38 ± 1.68
100	86.67 ± 2.27	86.96 ± 1.02	86.23 ± 2.00

TABLE IV
ACCURACY RESULTS FOR HEPATITIS DATA SET

M\F	**1**	**3**	**4**	**5**
50	82.58 ± 3.29	81.29 ± 3.76	82.58 ± 1.58	82.58 ± 4.83
100	85.16 ± 2.58	84.52 ± 1.29	81.94 ± 4.38	80.65 ± 5.40

TABLE V
ACCURACY RESULTS FOR IRIS DATA SET

M\F	**1**	**2**	**3**
50	93.33 ± 4.71	93.33 ± 4.71	94.67 ± 2.67
100	94.00 ± 4.90	94.00 ± 4.90	94.67 ± 2.67

B. Feature ranking

As said before, Random Forests are useful to sort the features of a data set according to their relevance, as long as $F > 1$. Otherwise, at each node the chosen feature is completely random. This is evidenced by the following example.

TABLE VI
ACCURACY RESULTS FOR VOTES DATA SET

M\F	**1**	**3**	**4**	**5**
50	94.71 ± 2.25	96.78 ± 1.69	96.55 ± 1.63	96.09 ± 1.17
100	95.17 ± 1.52	96.78 ± 1.52	97.01 ± 1.56	96.78 ± 1.52

TABLE VII
ACCURACY RESULTS FOR CHESS DATA SET

M\F	**1**	**3**	**6**
50	96.87 ± 0.75	99.03 ± 0.35	99.56 ± 0.15
100	97.18 ± 0.57	99.19 ± 0.44	99.53 ± 0.17

The ranking and scores for the Iris data set with $M = 50$ and $F = 1$ is:

1) sepal-length (0.27821)
2) sepal-width (0.266537)
3) petal-length (0.2393)
4) petal-width (0.215953)

We can see that the scores do not actually differ too much. However, let us see what happens maintaining the same value of M, but with $F = 3$:

1) petal-length (0.445693)
2) petal-width (0.348315)
3) sepal-length (0.198502)
4) sepal-width (0.00749064)

We can see that now petal-length and petal-width have gained much more relevance than the other two features.

The full list of rankings and scores can be found in the `experiment_results` folder. We refer the reader there.

TABLE VIII
ACCURACY RESULTS FOR LENSES DATA SET

M\F	**1**	**2**	**3**
50	75.00 ± 15.81	85.00 ± 12.25	85.00 ± 12.25
100	70.00 ± 18.71	85.00 ± 12.25	85.00 ± 12.25

TABLE IX
ACCURACY RESULTS FOR SOYBEAN DATA SET

M\F	**1**	**3**	**6**
50	100.00 ± 0.00	100.00 ± 0.00	100.00 ± 0.00
100	100.00 ± 0.00	100.00 ± 0.00	100.00 ± 0.00

TABLE X
ACCURACY RESULTS FOR SPLICE DATA SET

M\F	**1**	**3**	**6**	**8**
50	87.81 ± 2.20	95.02 ± 0.89	96.14 ± 0.80	96.39 ± 0.52
100	88.62 ± 3.01	96.05 ± 0.52	96.71 ± 0.38	96.39 ± 0.60

TABLE XI
ACCURACY RESULTS FOR ZOO DATA SET

M\F	**1**	**3**	**4**	**5**
50	95.00 ± 4.47	93.00 ± 4.00	94.00 ± 3.74	95.00 ± 4.47
100	95.00 ± 4.47	95.00 ± 3.16	96.00 ± 4.90	94.00 ± 3.74