

SUPERVISED AND EXPERIENTIAL LEARNING

(Master in Artificial Intelligence, UPC-URV-UB)

Spring semester, Course 2019/2020

April 16th, 2020

Practical Work 2 (PW2, Individual): combining multiple classifiers

The objective of this exercise is to implement, compare and validate two combinations of multiple classifiers: *a random forest* and *a decision forest*.

The implemented *Random Forest* [Breiman, 2001] and *Decision Forest* [Ho, 1998] will be compared and evaluated in several domains. The main steps that students must undertake are listed below.

Procedure

1. *Implement a Random Forest and a Decision Forest technique* in your selected programming language (Java, C++, R, Python, etc.). You can choose the *base-learner* for inducing trees among ID3, C4.5 or CART methods.
 - a. Both the *Random Forest (RF)* and the *Decision Forest (DF)* classifier must be able to read a dataset in csv file format
 - b. Then, they should *induce the model (the random forest or decision forest)* from the training data set, and at the same time *produce an ordered list of the features* used in the forest, according to its importance. The importance can be estimated as the frequency of its appearance in the random forest/random decision constructed.
 - c. The models must have, at least, the hyper-parameter *F* (number of random features used in the splitting of the nodes in RF or in each tree in DF) and the number of trees (*NT*) desired.
2. *Implement a forest interpreter* that given a *random forest* or a *decision forest* would be able to classify a validation/test dataset, obtaining the corresponding classification accuracy values (or generalization error) for different combination of values of *F* and *NT*. For instance, try (when make sense), being *M* the total number of features:

Random Forest

- Each training set for each tree is a bootstrapped sampling of the original training set
- $NT = 1, 10, 25, 50, 75, 100$
- $F = 1, 3, \text{int}(\log_2 M + 1), \sqrt{M}$

Decision Forest

- Each training set for each tree is the same original training set
- $NT = 1, 10, 25, 50, 75, 100$
- $F = \text{int}(M/4), \text{int}(M/2), \text{int}(3 * M/4), \text{Runif}(1, M)$ for each tree

Where $Runif(1,M)$ is a function generating a pseudorandom integer value, ru , such that $1 \leq ru \leq M$ with a uniform distribution probability.

3. *Evaluate both classifier models obtained in at least 3 databases. databases* (one small, one medium and one large). You can use databases from UCI ML repository or other sources. Small \approx (# instances ≤ 500), Medium \approx ($500 < \#$ instances ≤ 2000), and Large \approx (# instances > 2000). Obtain a summary table with the classification results (accuracy/error) and an ordered list of features for the 3 databases and the different combination of hyper-parameters.

Deliverable

A **ZIP file** labelled as “**PW2-SEL-1920-NameSurname**”, delivered through “**Racó de la FIB**” (in the “**Practical**” tab) with the following content:

1. A folder named “**Documentation**” with a report (maximum 20 pages on 10 pt. letter size) containing:
 - a. Pseudo-code of *your implemented algorithms* of the *random forest* and the *decision forest* technique
 - b. Evaluation of results for both algorithms and for all the tested databases:
 - i. Table with the accuracy/error results for the different combination of hyper-parameters
 - ii. Ordered list of features (relevance) resulting from the different combination of hyper-parameters
 - c. Instructions on how to execute the code
 - d. Other comments
2. A folder named “**Data**” with the files with the original dataset/s or database/s used both for training and for testing.
3. A folder named “**Source**” containing the source code of the implementation
4. An **executable object file** (*.jar, etc.) if available
5. A **README.txt** file specifying the structure and contents of the ZIP file

Students must deliver the ZIP file on **7/5/2020**,

Qualification

The qualification of this work will take into account the quality/functionality of the software delivered (correctness, efficiency and scalability), the robustness of the code, and the written documentation delivered.

References

- [Ho, 1998] Tin Kam Ho. The Random Subspace Method for Constructing Decision Forests. *IEEE Transactions on Pattern Analysis and Machine Intelligence* 20(8):832-844, 1998.
- [Breiman, 2001] Leo Breiman. Random Forests. *Machine Learning* 45:5-32, 2001

PW2 is due on May 7th, 2020