

CHAP 6 Notebook N1 - Classification Tree

1.1.a : The confusion matrix estimated from the test set is more relevant : using the same data for learning and evaluating the performances leads to over optimistic performances (remind that the learner is by construction best adapted to the training (or learning) set).

1.1.b : The accuracy is the percentage of correct classification

1.1.c/1.1.d : Estimating the accuracy on a single experiment is not reliable. As for any estimator, it is a function of a random chosen set, and is consequently a random variable. In order to lower the estimation variance, cross-validation is necessary.

1.2.a : From the curve above, and keeping in mind that the accuracy estimates are random variables, depths between 3 and 6 seem to be correct choices.

1.2.b : A value of depth=10 clearly leads to a classifier that is highly sensitive to specific instances of the data. We observe here the effect of over fitting. Another instance of the data set would lead to a very different contour. Such a result seems accurate on the learning data, but will find its performances highly degraded for another set of observations from the same underlying distribution.

1.2.c : The tree stops as the gini index is zero. All leaves represents « pure » subsets.

CHAP 6 Notebook N2-a - Regression Tree

2a.1.a : At each depth, the number of leaves is multiplied by 2, if all leaves are split. Thus the max number of prediction values will be 2^N for N depths.

2a.1.b : In the average, N depth tree will split the observation space into 2^N subsets, making a partition. If M is the number of samples in the training set, each subset will count $M/(2^N)$ samples.

2a.2.a : The optimal depth here is 6. (Be careful, the X-axis represents the indices in the « max-depth » array instead of representing the max-depth values)

2a.2.b / 2a.2.c : The higher the noise, the shorter the tree. In the limit where the noise is highly dominating, a prediction corresponding to the mean of the observations would be MSE optimal.

CHAP 6 Notebook N2-b - Regression Tree CCP

2b.1.a : depth = 13; Nb_Leaves = 125

2b.1.b : The generalization error (e.g. estimated by cross validation) would be bad for such an estimator. This is clearly a situation where overfitting occurs.

2b.2.a : The total impurity is the sum is the weighted sum of the impurity found in each leaf. The weights are the proportion of points in the leaves.

2b.2.b : This means that no regularization is applied. We are back to the previous situation where no max depth is prescribed.

2b.2.c : Very large values of alpha will lead to a high penalization of splitting that no split will be made. The tree will have a single node containing the whole data set.

2b.2.c : Scores are estimated on a single data realization here. Cross validation would decrease the variance of the estimated score.

2b.3.a : From the given code, 39 leaves are obtained for the optimal alpha. This corresponds to the same number of quantization levels

2b.3.b : The major advantage lies in the fact that the tree depth is locally adapted to the data complexity in the case of pruned trees, whereas the max depth limitation imposes that one stops growing the tree even in the case where MSE improvement would be still useful. The key is that pruned trees are a way to adapt the complexity of the data model to the area in the observation space where this is useful, while maintaining low complexity elsewhere.

CHAP 6 Notebook N3-a - Random Forest Regression

3a.1.a. / 3a.1.b. : Changing the max-depth parameter shows that
 - for max-depth low (3 or 4) : the plateau observed in reg tree is still existing for value where y varies only slowly as a function of x. However, values obtained when y exhibits high variations in function of x are smoother (less step like) for random forest. This comes from the fact that the estimated values correspond to average results obtained in the tree forest.
 Too high values of max_depth lead clearly to overfitting the data.

3a.2.a. : Observe that all obtained curves are quite smooth. For low values of max_depth (2, 3), each tree contains very few leaves. Each leaf gives a very rough approximation value of the output, close to the local mean of the observations which are highly varying within the subset (having random contours in this 'extremely randomized trees') defined by a leaf. By averaging over the trees in the forest, we still obtain an important approximation error. Notice also that higher values of n_estimators lead to smoother prediction curves.

3a.2.b. (difficult) For max_depth = 1, each tree in the forest corresponds to a partition in two subsets, $x > \eta$ or $x < \eta$. Extremely randomized trees use random values for η . Consider a small value of x : all trees having η close to (but greater than) x will lead to a 'precise' prediction of y : y will be positive. Similarly for large values of x (around 6 on our example), for all η slightly lower than x, y will be negative. As n_estimator is large, considering that η is drawn at random, the probability density of η is uniform over the dynamical range of x.
 for x around 3 on our example, the probability of getting positive and negative values are identical. In the average, we obtain a predicted value 0.

3a.2.c : Cross validation, as usual...

3a.2.d. Extremely randomized trees (ERT) exhibit good performances if n_estimator is large enough to allow the thresholds used in the splitting operations to cover the full range of possible values. The price to pay for ERT to exhibit nice performances is value of n_estimator that has to be large....although this implies also to have a large training set to avoid constructing correlated trees in the forest.

CHAP 6 Notebook N3-b - Random forest - Classification

3b.1 : 100 samples in the training set. $100 / (2^{(\text{max_depth})}) \geq 2$ gives $n < \log_2(100) - 1$. (~5.6)

3b.2. : reproduce the codes found in previous notebooks.

3b.3.a. / 3b.3.b : Setting `n_estimators` to 40, the results depends on the choice of the purity index chosen, and on the depth chosen. Best results occurs for `mdepth=3`.

Setting both `max-depth` and `n_estimators` can be tackled by evaluating the cross validated scores as function of these two parameters.

3b.4.a : Reporduce the same code as the one for `RandomForestClassifier`. Again, features 2 and 3 appear to be clearly the most important in the classification task. Feature 1 may be discarded without significantly decreasing the performances.