

Ensemble Methods

TP 2 - Bagging and random forests

Master MALIA

Guillaume Metzler

Institut de Communication (ICOM)

Université de Lyon, Université Lumière Lyon 2

Laboratoire ERIC UR 3083, Lyon, France

guillaume.metzler@univ-lyon2.fr

Abstract

At the end of TP 1, a majority vote of trees learned on bootstrap samples was much better than a single tree. In this session we understand why. We first measure the instability of decision trees and decompose their error into bias and variance; we then implement bagging, use its out-of-bag error, and check that bagging reduces the variance but not the bias, so that it helps unstable models only. Finally, random forests decorrelate the trees, and we observe the trade-off between the accuracy of the voters and their diversity.

About this session - TP 2: Bagging and random forests

Goal. By the end of the session you should be able to explain why trees have a high variance, implement bagging and compute its out-of-bag error, say which models benefit from bagging, implement a random forest and explain the role of the number of features drawn at each node.

To reread before the session. Section 8.7 (decision trees), Section 10.1 (error decomposition) and Section 10.2 (bagging and random forests) of the lecture notes.

Tools. `numpy`, `matplotlib`; the trees of `scikit-learn` (`DecisionTreeClassifier`, `DecisionTreeRegressor`) as base learners. Bagging and random forests are implemented by hand, and compared with `BaggingClassifier` and `RandomForestClassifier`.

Data. `wdbc`, `pima`, `australian` and `spambase` (4597 e-mails, 57 features), loaded with `load(name)` of `common/tp2tools.py`, labels in $\{-1, +1\}$, with 70%/30% stratified training/test splits (`random_state=seed`); and the regression problem $Y = \sin(2\pi X) + \varepsilon$, X uniform on $[0, 1]$, $\varepsilon \sim \mathcal{N}(0, 0.3^2)$. The protocol of TP 1 applies: the test set is used only to report errors. The file `tp2_starter.py` contains the skeletons.

Good practice Store the predictions of all the base models in a $B \times n_{\text{test}}$ matrix `V` of ± 1 values: the vote of the first B models is `np.sign(np.cumsum(V, 0))` (decide what to do with ties), the mean error of a model is `np.mean(V != y)`, and many quantities of this session are one line of `numpy` away.

1 Trees are unstable

1. On `pima` (split `seed=0`), learn two trees on two bootstrap samples of the training set (m indices drawn with replacement) and compute the proportion of test points on which they disagree.
Repeat 20 times, for maximal depths 1, 2, 5 and unlimited.
Compare with the test error of a tree.
2. **Bias and variance.** For the regression problem, draw $R = 200$ training samples of size $m = 100$, learn a regression tree of maximal depth D on each, and estimate on a regular grid of $[0, 1]$ the squared bias $\overline{(\bar{f}_D(x) - f(x))^2}$ and the variance $\text{Var}_r \hat{f}_D^{(r)}(x)$, where $\bar{f}_D$ is the average of the R trees and the bar denotes the average over the grid.
Plot them for $D \in \{1, \dots, 12\}$.
Recall why the expected test MSE is $\text{bias}^2 + \text{variance} + \sigma^2$.

2 Bagging from scratch

1. Show that a bootstrap sample of size m contains on average a proportion $1 - (1 - \frac{1}{m})^m \approx 1 - e^{-1} \approx 63.2\%$ of distinct examples.
Check it numerically for $m = 537$.
2. Write `bagging(Xtr, ytr, Xte, B, rng)` which learns B fully grown trees, each on a bootstrap sample, and returns the $B \times n_{\text{test}}$ matrix of their predictions.
On `wdbc`, `pima` and `spambase` ($B = 200$, 3 splits), compute the test error of one full tree learned on the whole training set, the mean test error of a bagged tree, and the test error of the vote (ties: majority class).
Plot the error of the vote of the first B trees as a function of B .
3. **Out-of-bag error.** Each training example is *out of bag* for about 37% of the trees.
Predict each training example by the vote of these trees only, and compute the out-of-bag (OOB) error.
Compare it with the test error.
Why is it a valid estimate of the risk of the vote, without any validation set?
4. Compare `BaggingClassifier(DecisionTreeClassifier(), n_estimators=200)` with `DecisionTreeClassifier()`.
5. Does the vote overfit when B increases?
How many trees are needed?

3 What does bagging reduce?

1. For the regression problem ($R = 100$ training samples of size 100), compare the squared bias and the variance of a single regression tree and of a bagged average of 50 regression trees, for depths 2, 4 and unlimited.
2. On `pima` and `spambase` (3 splits), compare the test error of a model and of the vote of 50 bagged copies of it, for: a fully grown tree, a tree of depth 2, 15-NN and a logistic regression (both with standardization).
Which models benefit from bagging?
Explain.

4 Random forests: decorrelating the trees

A random forest is bagging in which, at each node of each tree, the best split is searched among m_{try} features drawn at random instead of all the d features (`DecisionTreeClassifier(max_features=mtry)`); $m_{\text{try}} = d$ is bagging.

Notation of Part IV. The vote of B trees with equal weights is the majority vote B_Q of the uniform distribution Q on the trees (Section 12.2). For an example (x, y) , $W_Q(x, y)$ is the proportion of trees that err on it; the vote errs when $W_Q \geq \frac{1}{2}$. The mean error of the trees is the *Gibbs risk* $\hat{R}_S(G_Q) = \hat{\mathbb{E}}[W_Q]$ (Definition 12.1). For two trees h, h' drawn independently from Q (possibly the same one), the *joint error* $\hat{e}(Q)$ is the probability that both err and the *disagreement* $\hat{d}(Q)$ the probability that they predict differently (Definition 15.4).

1. For $m_{\text{try}} \in \{1, 2, 3, 5, 8\}$ on `pima` and $\{1, 3, 7, 15, 30, 57\}$ on `spambase`, learn forests of 100 trees and compute (a) the mean test error of one tree, (b) the mean *disagreement* between two distinct trees (the proportion of test points on which they predict differently, averaged over the pairs), (c) the test error of the vote.
Plot the three quantities against m_{try} and comment.
2. Show that, with $v(x) = \frac{1}{B} \sum_b h_b(x)$, the mean disagreement over the $B(B-1)$ ordered pairs of distinct trees is $\frac{B}{B-1} \hat{\mathbb{E}}_x \left[\frac{1-v(x)^2}{2} \right]$.
Show that the disagreement $\hat{d}(Q)$ of Part IV, which averages over all the B^2 ordered pairs (including $h = h'$), is $\hat{\mathbb{E}}_x \left[\frac{1-v(x)^2}{2} \right] = \hat{\mathbb{E}}[2W_Q(1-W_Q)]$, that $\hat{e}(Q) = \hat{\mathbb{E}}[W_Q^2]$, and deduce the identity $\hat{R}_S(G_Q) = \hat{e}(Q) + \frac{1}{2}\hat{d}(Q)$ (Proposition 15.5).
3. On the four datasets, compare a single tree whose depth is tuned by 5-fold cross-validation (TP 1) with `RandomForestClassifier(500, max_features="sqrt", oob_score=True)`.
Compare the OOB error of the forest with its test error.
4. **The puzzling experiment of Section 12.1.** On the `digits` dataset of `scikit-learn` (task: is the digit smaller than 5?), split the data with `train_test_split(X, y, test_size=0.3, random_state=0)` (1257 training and 540 test digits) and grow 100 trees, each on a bootstrap sample of *half* the size of the training set, with `max_features="sqrt"`.
Plot, as a function of n , the test error of the vote of the first n trees, their Gibbs risk and their disagreement; plot the histogram of the test errors of the 100 trees.

Compute $\hat{e}(Q)$ and $\hat{d}(Q)$ for the 100 trees on the test set and check the identity of question 2.

Why is the vote so much better than its trees?

5 Going further (★)

1. **Variance of an average.** Let $Z_1, \dots, Z_B$ be identically distributed with variance σ^2 and pairwise correlation $\rho \geq 0$.

Show that $\text{Var}(\frac{1}{B} \sum_b Z_b) = \rho\sigma^2 + \frac{1-\rho}{B}\sigma^2$.

What does this formula say about bagging when $B \rightarrow \infty$, and about random forests?

2. Check it numerically: on `pima`, at 40 fixed test points, compute the average of the predictions (± 1) of B bagged full trees, for 60 training sets (random halves of the training set), and estimate its variance for $B \in \{1, 5, 25, 100\}$.

Estimate ρ from $B = 1$ and $B = 100$ and compare the formula with the other values.

3. If B trees erred *independently* with probability $r < \frac{1}{2}$ on every point, what would happen to the error of the vote when $B \rightarrow \infty$ (Condorcet's jury theorem, Proposition 15.3)?

Why is this not what we observe?