Aprenentatge Automàtic 1

GCED

Lluís A. Belanche

belanche@cs.upc.edu

Soft Computing Research Group
Dept. de Ciències de la Computació (Computer Science)
Universitat Politècnica de Catalunya

2019-2020

LECTURE 10: Random Forests and ensemble methods

Random Forests and ensemble methods

What is ensemble learning?

Methods that learn by training a number of better-than-chance individual learners and then combine their predictions

Why ensemble learning?

1. More accurate models can be obtained by combining the output of multiple local "experts"

2. A complex problem can be decomposed into multiple sub-problems that are easier to understand and solve (divide-and-conquer)

3. We often train many models and feel we can get more by combining a number of them (and the computational effort is not wasted)

Random Forests and ensemble methods

When to use ensemble learning?

When we can build individual learners that are more accurate than chance and, more importantly, that are independent from each other

How to combine the outputs?

Classification : plurality voting (most votes); majority voting for 2 classes

Regression : arithmetic average

Random Forests and ensemble methods

REGRESSION

Call $\mathcal{E}$ the ensemble, $\mathcal{L}_i$ the i-th individual learner, $1 \leq i \leq L$; fix an input vector $\boldsymbol{x}$ and let $\epsilon_i := \mathbb{E}[|f(\boldsymbol{x}) - \mathcal{L}_i(\boldsymbol{x})|]$, where f is the regression function

1. The **expected MSE** for a randomly chosen individual learner is:

$$\frac{1}{L}\sum_{i=1}^{L}\epsilon_i^2$$

2. The **expected MSE** for the ensemble is:

$$\left(\frac{1}{L}\sum_{i=1}^{L}\epsilon_i\right)^2$$

Random Forests and ensemble methods

REGRESSION

Cauchy's inequality:

$$\left(\frac{1}{L}\sum_{i=1}^{L}\epsilon_i\right)^2 \leq \frac{1}{L}\sum_{i=1}^{L}\epsilon_i^2$$

- In particular, if the individual learners are independent, the reduction in expected error would be $1/L$

- In practice this is not the case, because the individual learners are normally trained out of the same data: they will tend to correlate the errors and the improvement will be smaller

Random Forests and ensemble methods

CLASSIFICATION

Assume that:

1. The number of classifiers, L, is odd; we have $K \geq 2$ class labels
2. Each classifier gives the correct class with probability p, for any $\boldsymbol{x}$
3. The classifier outputs are independent

The majority vote will give an accurate class label if at least $\lfloor L/2 \rfloor + 1$ classifiers give correct answers

---

The majority (50 % + 1) of votes is necessary and sufficient for a correct decision in the case $K = 2$; and is sufficient but not necessary for $K > 2$. Thus the real accuracy of an ensemble using plurality when $K > 2$ could be greater than the majority vote accuracy

Random Forests and ensemble methods

CLASSIFICATION

Consider $K = 2$. The accuracy of the ensemble is:

$$a_{\mathcal{E}} := \sum_{i=\lfloor L/2 \rfloor + 1}^{L} \binom{L}{i} p^i (1-p)^{L-i}$$

Condorcet Jury Theorem

1. If $p > 1/2$, then $a_{\mathcal{E}}$ is monotonically increasing and $a_{\mathcal{E}} \to 1$ as $L \to \infty$

2. If $p < 1/2$, then $a_{\mathcal{E}}$ is monotonically decreasing and $a_{\mathcal{E}} \to 0$ as $L \to \infty$

3. If $p = 1/2$, then $a_{\mathcal{E}} = 1/2$ for any L

Random Forests and ensemble methods

The **CART** (Classification and Regression Tree) algorithm, uses the **Gini Gain**: how often a randomly chosen example would be incorrectly labeled if it were labeled according to the empirical probability distribution of the classes:

$$\text{Gini}(S) := \sum_{k=1}^{K} p_k(S)[1 - p_k(S)] = 1 - \sum_{k=1}^{K} p_k^2(S)$$

being $p_k(S)$ the fraction of examples in S labeled with value k

1. If a variable completely characterizes the class, then $\text{Gini}(S) = 1 - 1^2 = 0$, which is a "pure" scenario

2. If the probability distribution is uniform, $(1/K, \ldots, 1/K)$, then the Gini index achieves the maximum value $= 1 - 1/K$

Random Forests and ensemble methods

CARTs using Gini impurity

The **Gini gain** is an impurity-based criterion that measures the divergence between probability distributions of the class labels:

- The Gini gain of a variable is:

$$\text{GiniGain}(S) := \text{Gini}(S) - \sum_{v=1}^{V} \frac{|S_v|}{|S|} \text{Gini}(S_v)$$

  then we should choose the variable that leads a maximum GiniGain

- For continuous variables, we sort and bin-split; if all features are continuous, we will obtain a binary tree

Random Forests and ensemble methods

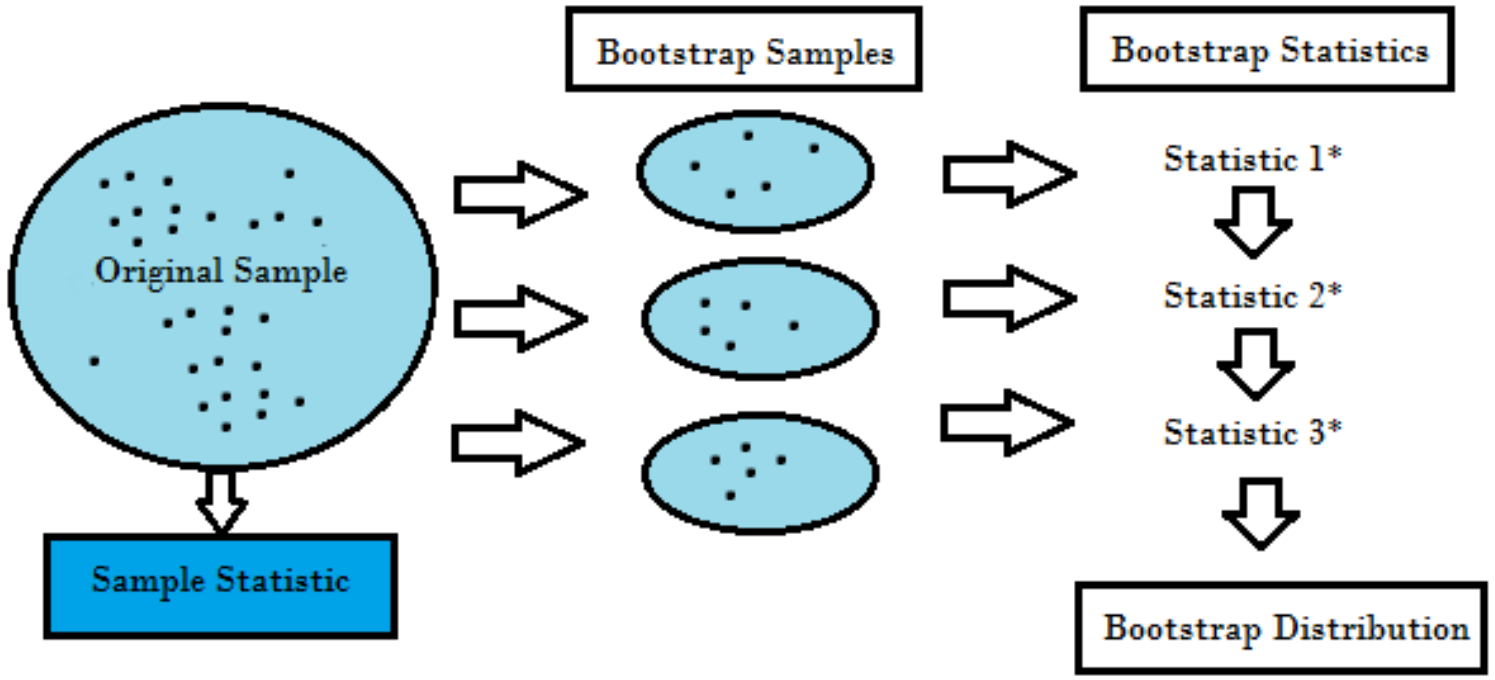
Methods for constructing ensembles

- Subsample the training examples
- Manipulate the features
- Manipulate the targets
- Change the learning parameters
- ...

A combination of the above

Random Forests and ensemble methods

Bootstrap resampling



Random Forests and ensemble methods

Bootstrap resampling

- Assuming we have a data set $D = \{(\boldsymbol{x}_1, t_1), \ldots, (\boldsymbol{x}_N, t_N)\}$, we draw **bootstrap resamples** $D_1^*, \ldots, D_B^*$ of size N by sampling D with replacement and then fit a model to each of the D_b^*

- A statistic $\widehat{\theta} = \theta(D)$ can be estimated in the usual way, e.g.:

$$\overline{\theta}^* = \frac{1}{B} \sum_{b=1}^{B} \theta_b^*; \qquad \mathrm{Var}(\theta^*) = \frac{1}{B-1} \sum_{b=1}^{B} (\theta_b^* - \overline{\theta}^*)^2$$

  are the **mean** and **variance** of the bootstrap distribution of $\widehat{\theta}$, and $\theta_b^* := \theta(D_b^*)$. An estimation for the **bias** of $\widehat{\theta}$ is $\overline{\theta}^* - \widehat{\theta}$

Random Forests and ensemble methods

Subsampling the training set: Bagging

Bagging [Breiman, 1996] (**B**ootstrapping **agg**regat**ing**) creates an ensemble by training individual classifiers on bootstrap resamples of the training set:

- Generate a bootstrap sample from a sample of size N (N independent draws with replacement)
- Train a predictor y_b^* on every bootstrap resample D_b^*
- Repeat the process until you have B resamples and predictors

---

In a dataset with N examples, each has a probability $p(N) := 1 - (1 - 1/N)^N$ of being selected at least once; now $p(N) \to 1 - e^{-1} \approx 0{,}632$ as $N \to \infty$

Random Forests and ensemble methods

Subsampling the training set: Bagging

- The perturbation in the training set due to bootstrap resampling causes different models to be built, particularly if the classifier is unstable ...

- A modelling method is called **unstable** if a small change in the training data (e.g., order of presentation, addition/deletion of data) can lead to a radically different hypothesis (typical of **overfit** models: low bias/high variance)

  $\longrightarrow$ examples include decision trees and neural networks

- Bagging can dramatically **reduce the variance** of unstable methods of the same type, leading to improved prediction

Random Forests and ensemble methods

Bootstrap resampling

Out-of-bag error (OOB): data not in the bootstrap resamples is used as predictive data:

$$\text{Err}^* := \frac{1}{N} \sum_{n=1}^{N} \frac{1}{|D^{-n}|} \sum_{b \in D^{-n}} \mathbb{I}[t_n \neq y_b^*(\boldsymbol{x}_n)]$$

where

- $\mathbb{I}(z)$ is 1 when z is true and 0 otherwise
- D^{-n} is the set of indexes of the resamples that do not contain observation $\boldsymbol{x}_n$
- y_b^* is the model fitted to D_b^*

Random Forests and ensemble methods

Bootstrap resampling

- Similarly, the **resubtitution** (aka training) error is estimated as:

$$\overline{e}^* := \frac{1}{N} \sum_{n=1}^{N} \frac{1}{|D^n|} \sum_{b \in D^n} \mathbb{I}[t_n \neq y_b^*(\boldsymbol{x}_n)]$$

  where $D^n = \{1, \ldots, B\} \setminus D^{-n}$

- The **0.632-bootstrap estimate** is defined by

$$\text{Err}^{(0,632)} := 0{,}368\, \overline{e}^* + 0{,}632\, \text{Err}^*$$

  Intuitively, this pulls the OOB bootstrap estimate down toward the training error, thereby reducing its likely upward bias

Random Forests and ensemble methods

Random Forests [Breiman, 2001] is bagging where the individual predictors are decision trees, with additional randomization:

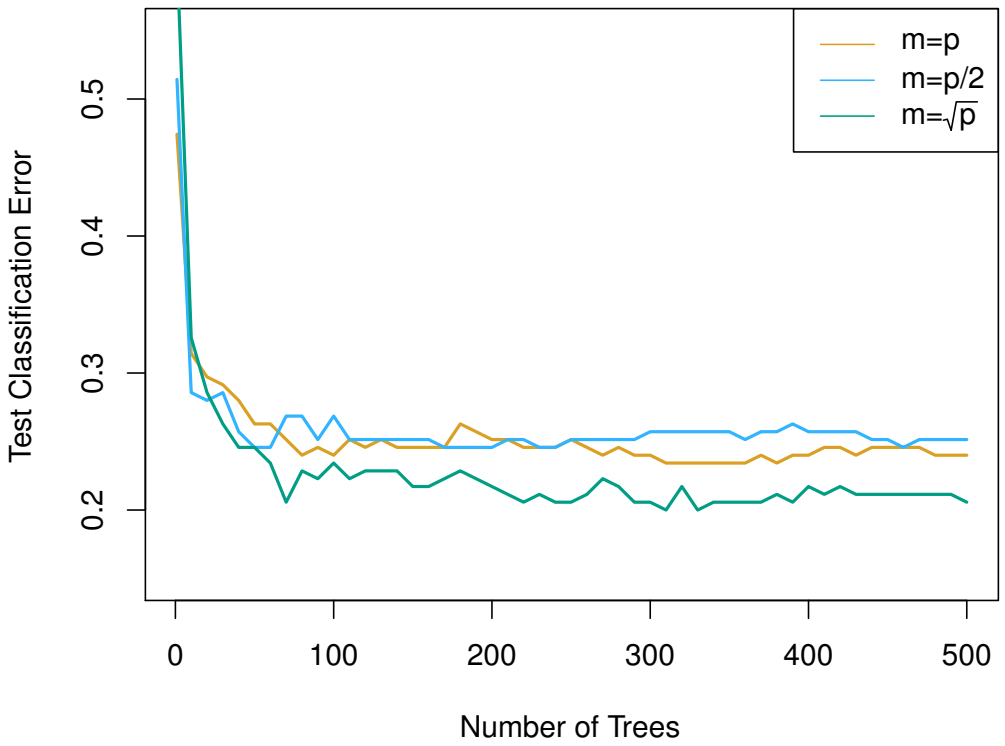
- Generate bootstrap resamples and build one tree for each one
- Increase diversity (decorrelate the trees) by randomly picking a subset of features of size $m << p$ to split at each node
  (default values are $\sqrt{p}$ for classification and $p/3$ for regression)
- Two basic parameters: B (number of trees) and m (number of local features); both can be optimized via the OOB error
- Construction is fast, since just a few features are explored per tree

---

Breiman, L. (2001). Random Forests. *Machine Learning*, 45(1):5–32.

Random Forests and ensemble methods

Random Forest in action



RFs usually (but not always) outperform both the individual predictors and direct bagging; the error rate of a single tree is 0.457, and the null rate is 0.754 –Fig. 8.10 from *An Introduction to Statistical Learning*.

Random Forests and ensemble methods

Further ideas on Random Forests

RFs can also used to estimate **variable relevance**:

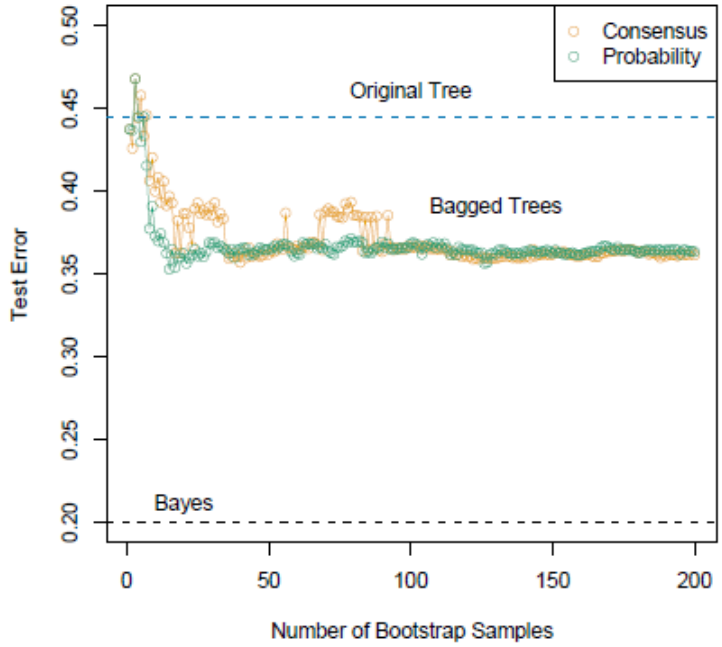
- For each tree, the out-of-bag prediction error is computed
- Then again after permuting each predictor variable
- The difference between the two is averaged over all trees, and normalized by its standard deviation

1. $\text{raw}_b(j) := \text{OOB}(b, j) - \text{OOB}(b),\ 1 \leq b \leq B$
2. $\text{imp}_j := \frac{1}{B} \sum_{b=1}^{B} \text{raw}_b(j)$

Random Forests and ensemble methods

Further ideas on Random Forests

Instead of bagging the plurality (consensus) vote, we can also average the probabilities



The error rate of a single tree is 0.447, and the Bayes error is 0.2 %. –from *The Elements of Statistical Learning*.