Basics of Machine Learning

SD 210 - P3

Lecture 5 - Ensemble methods: bagging and random forests

Florence d'Alché-Buc

Contact: florence.dalche@telecom-paristech.fr,
2A Filière SD, Télécom ParisTech,Université of Paris-Saclay, France

Table of contents

Outline

Ensemble methods for classification and regression

1. Remark:
   - Machine Learning not so "automatic": too many hyperparameters to tune
2. **meta-learning**: a procedure that learns to learn
3. **committee learning** or **wisdom of the crowd**: better results are obtained by combining the predictions of a set of **diverse** classifiers/regressors

   have a try in challenge
4. **ensemble learning**: Improve upon a single base predictive model by building an ensemble of predictive model (with no hyperparameter)

Ensemble methods for regression

Let $f_t, t = 1, \ldots, T$ be T different regressors.
Notations:

$$\begin{aligned}
\epsilon_t(x) &= y - f_t(x) \\
MSE(f_t) &= \mathbb{E}[\epsilon_t(x)^2] \\
f_{ens}(x) &= \frac{1}{T}\sum_t f_t(x) \\
&= y - \frac{1}{T}\sum_t \epsilon_t(x).
\end{aligned}$$

Encourage the diversity of base models

$$MSE(f_{ens}) = \mathbb{E}[(y - f_{ens}(x))^2]$$

If ϵ_t are mutually independent with zero mean, then we have:

$$MSE(f_{ens}) = \frac{1}{T^2}\mathbb{E}[\sum_t \epsilon_t(x)^2]$$

The more diverse are the models, the more we reduce the mean square error !

Ensemble methods for supervised classification

Binary classification

$$h_{ens}(x) = \text{sign}(\sum_t h_t(x))$$

Multiclass classification

$$h_{ens}(x) = \arg\max_c \text{vote}(c, h_1, \ldots, h_T)$$

with : $\text{vote}(c, h_1, \ldots, h_T) = \sum_t 1_{h_t(x)=c}(h_t(x))$

Ensemble methods

- **Encourage the diversity of base models by:**
  - using bootstrap samples (Bagging and Random forests)
  - using randomized models (ex: Random forests)
  - using weighted version of the current sample (Boosting) with weights dependent on the previous model (adaptive sampling) we'll see that in next module SD207

Ensemble methods at a glance

- 1995: Boosting, Freund and Schapire
- 1996: Bagging, Breiman
- 2001: Random forests, Breiman
- 2006: Extra-trees, Geurts, Ernst, Wehenkel

Outline

Reminder: Decomposition bias/variance in regression

Given x,

$$\mathbb{E}_S \mathbb{E}_{Y|x}(Y - f_S(x))^2 = \mathit{noise}(x) + \mathit{bias}^2(x) + \text{variance}(x) \tag{1}$$

noise(x): $E_{Y|x}[(Y - E_{Y|x}(Y))^2]$:
quantifies the error made by the Bayes model ($E_{y|x}(y)$)
$\mathit{bias}^2(x) = (E_{Y|x}(Y) - E_S[f_S(x)])^2$
measures the difference between minimal error (Bayes error) and the average model
$\mathit{variance}(x) = E_S[(f_S(x) - E_S[f_S(x)])^2]$
measures how much $h_S(x)$ varies from one training set to another

Introduction to bagging (regression) - 1

Assume we can generate several training independent samples $\mathcal{S}_1, \ldots, \mathcal{S}_T$ from P(x,y).

A first algorithm:

- draw T training independent samples $\{\mathcal{S}_1, \ldots, \mathcal{S}_T\}$
- learn a model $f_t \in \mathcal{F}$ from each training sample $\mathcal{S}_t$; $t = 1, \ldots, T$
- compute the average model : $f_{ens}(x) = \frac{1}{T}\sum_{t=1}^{T} f_t(x)$

Introduction to bagging - 2

The bias $(E_{\mathcal{S}_1,\dots,\mathcal{S}_T}[f_{ens}(x)] - f_{target}(x))$ remains the same because :
$E_{\mathcal{S}_1,\dots,\mathcal{S}_T}[f_{ens}(x)] = \frac{1}{T}\sum_t E_{\mathcal{S}_t}[f_t(x)] = E_{\mathcal{S}}[f_{\mathcal{S}}(x)]$
But the variance is divided by T:
$E_{\mathcal{S}_1,\dots,\mathcal{S}_T}[(f_{ens}(x) - E_{\mathcal{S}_1,\dots,\mathcal{S}_T}[f_{ens}(x)])^2] = \frac{1}{T}E_{\mathcal{S}}[(f_{\mathcal{S}}(x) - E_{\mathcal{S}}[f_{\mathcal{S}}(x)])^2]$
When is it useful? When the learning algorithm is unstable, producing high variance estimators such as trees !

Bagging (Breiman 1996)

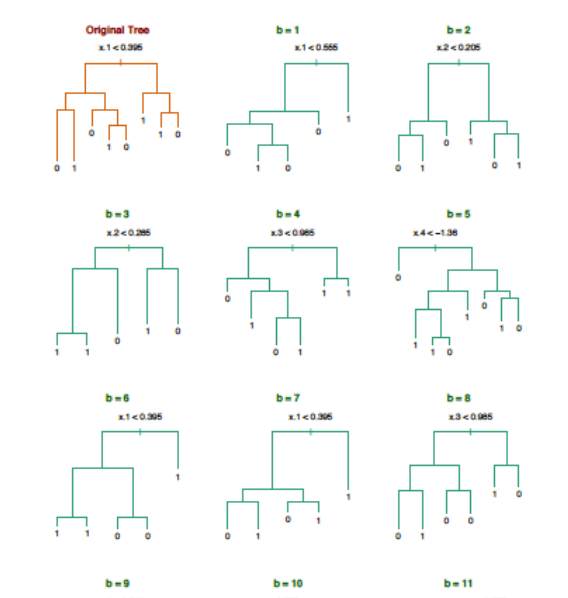
In practice, we do not know P(X,Y) and we have only **one training sample** $\mathcal{S}$: we are going to use Bootstrap samples !

Bagging = Bootstrap Aggregating

- draw T bootstrap samples $\{_1, \ldots, _T\}$ from $\mathcal{S}$ (bootstrap: uniform sampling with replacement)
- Learn a model f_t for each $_t$
- Build the average model: $f_{bag}(x) = \frac{1}{T} \sum_t f_t(x)$

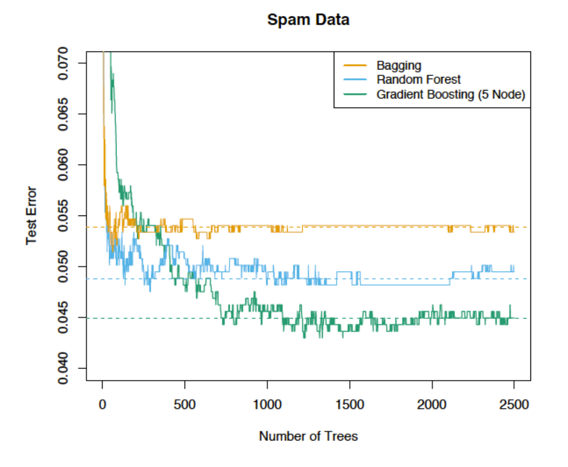
Example of bagged trees

[Book: The elements of statistical learning, Hastie, Tibshirani, Friedman,



Example of bagged trees

[Book: The elements of statistical learning, Hastie, Tibshirani, Friedman, 2001]



Bagging in practise

- Variance is reduced but the bias can increase a bit (the effective size of a bootstrap sample is 30% smaller than the original training set $\mathcal{S}$
- The obtained model is however more complex than a single model
- Bagging works for unstable predictors (neural nets, trees)
- In supervised classification, bagging a good classifier usually makes it better but bagging a bad classifier can make it worse

Outline

Random forests

Produce more diversity by building "more" de-correlated trees

- Perturbe and combine algorithms
  - Perturbe the base predictive model by bagging and variable randomization
  - Combine the perturbed predictive model

REFS: Random forests: Breiman 2001

Geurts, Ernst, Wehenkel, Extra-trees, 2006

Random forests: Breiman 2001

Random forests algorithm

- INPUT: F= p candidate feature splits, $\mathcal{S}_{train}$
- for t=1 to T
  - $\mathcal{S}_{train}^{(t)}$ m instance randomly drawn with replacement from $\mathcal{S}_{train}$
  - $h_{tree}^{(t)} \leftarrow$ randomized decision tree learned from $\mathcal{S}_{train}^{(t)}$
- OUTPUT: $H^T = \frac{1}{T}\sum_t h_{tree}^{(t)}$

Learning a single randomized tree

- To select a split at a node:
  - $R_f(F) \leftarrow$ randomly select (without replacement) f feature splits from F with $f << p$
  - Choose the best split in $R_f(F)$ (consider the different cut-points)
- Do not prune this tree
  修剪

Extra-trees: Geurts et al. 2006

Extra-trees

- INPUT: candidate feature splits $F = \{1, \ldots, p\}$,S_{train}
- for t=1 to T
  - Always use $\mathcal{S}_{train}$
  - $h_{tree}^{(t)} \rightarrow$: randomized decision tree learned from $\mathcal{S}_{train}$
- OUTPUT: $H^T = \frac{1}{T} h_{tree}^{(t)}$

Learning a single randomized tree in extra-trees

- To select a split at a node:
    - randomly select (without replacement) K feature splits from F with $K << |F|$
    - Draw K splits using the procedure Pick-a-random-split($\mathcal{S}$,i):
        - let a^i_{max} and a^i_{min} denote the maximal and minimal value of x_i in $\mathcal{S}$
        - Draw uniformly a cut-point a_c in $[a^i_{max}, a^i_{min}]$
    - Choose the best split among the K previous splits

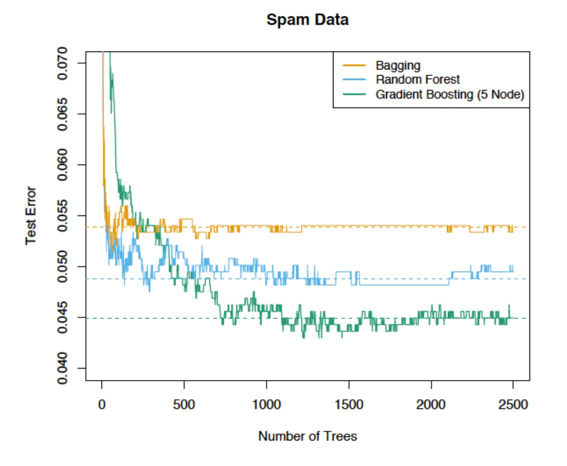
Do not prune this tree

Random Forests and extra-trees

- Extra-trees faster (do not need to build bootstrap samples + shorter split selection procedure)
- Recent consistency results: for random forests (Scortnet et al. 2016)

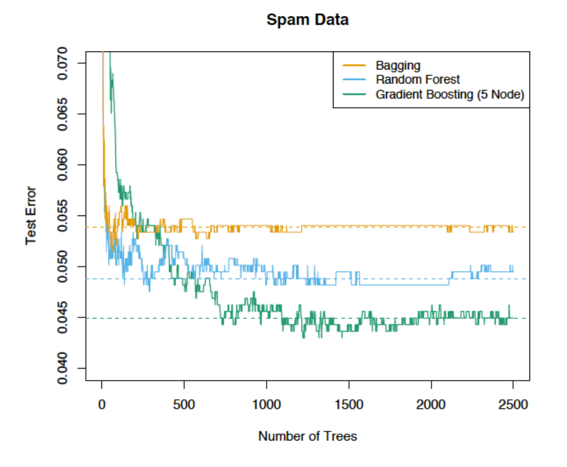
Random forest

Example of decision frontier:



Comparison (just an example)

[Book: The elements of statistical learning, Hastie, Tibshirani, Friedman, 2001]



Random forest

Pros

- Fast, parallelizable and appropriate for a large number of features
- Relatively easy to tune
- Frequently the winner in challenges

Cons

- Overfitting if the size of the trees is too large
- Interpretability is lost (however importance of feature can be measured)

Variable importance

Definition

A variable X^j is important to predict Y if breaking the link between X^j and Y increase the prediction error

$\{\bar{\mathcal{S}}_n^t = \mathcal{S}_n - \mathcal{S}_n^t, t = 1, \ldots, n_{tree}\}$ **out-of-bag samples**: contains the samples not selected by bootstrap

Variable importance

一种测试importance的方法

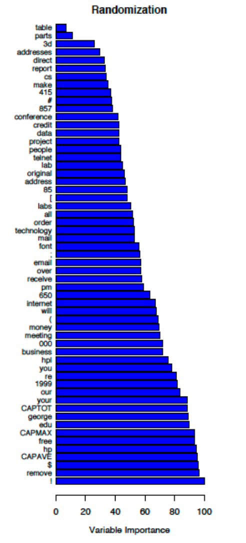
Let $\{\bar{\mathcal{S}}_n^t = \mathcal{S}_n - \mathcal{S}_n^t, t = 1, \dots, n_{tree}\}$ **out-of-bag samples**
Let $\{\bar{\mathcal{S}}_n^{t,j}, t = 1, \dots, n_{tree}\}$: permuted out-of-bag-samples (the values of the jth variable have been randomly permuted).

$$\hat{I}(X^j) = \frac{1}{n_{tree}} \sum_{t=1}^{n_{tree}} R_n(f_t, \bar{\mathcal{S}}_n^{t,j}) - R_n(f_t, \bar{\mathcal{S}}_n^t)$$

with $R_n(f, \mathcal{S})$: empirical loss of h measured on $\mathcal{S}$

Variable importance: spam data

Spam dataset :



Outline

References

- Perrone, Cooper, When classifiers disagree, 1992
- Tumer and Gosh, 1996
- Breiman, Bagging predictors, 1996
- Further reading: Buhlman and Yu, Analyzing bagging, Annals of stats., 2002
- Breiman, Random Forests, Machine Learning, 2001.
- Geurts, Ernst, Wehenkel, Extra-trees, JMLR, 2006