Machine Learning for biology

V. Monbet

UFR de Mathématiques
Université de Rennes 1

Outline

Introduction

Dimension Reduction

Unsupervised learning

Supervised learning

Linear model

Data driven supervised learning

Ensemble methods (I)

Ensemble methods (II)

Outline

Introduction

Dimension Reduction

Unsupervised learning

Supervised learning

Linear model

Data driven supervised learning

Ensemble methods (I)

Ensemble methods (II)

Outline

Introduction

Dimension Reduction

Unsupervised learning

Supervised learning

Linear model

Data driven supervised learning

Ensemble methods (I)

Ensemble methods (II)

Outline

Introduction

Dimension Reduction

Unsupervised learning

Supervised learning

Linear model

Data driven supervised learning

Ensemble methods (I)

Ensemble methods (II)

Outline

Introduction

Dimension Reduction

Unsupervised learning

Supervised learning

Linear model

Data driven supervised learning

Ensemble methods (I)

Ensemble methods (II)

Outline

Introduction

Dimension Reduction

Unsupervised learning

Supervised learning

Linear model

Data driven supervised learning

Ensemble methods (I)

Ensemble methods (II)

Outline

Introduction

Dimension Reduction

Unsupervised learning

Supervised learning

Linear model

Data driven supervised learning

Ensemble methods (I)
- Introduction
- Bumping, bootstrap
- Bagging (bootstrap aggregation)
- Random Forests

Ensemble methods (II)

Outline

Ensemble methods (I)

Introduction

Bumping, bootstrap

Bagging (bootstrap aggregation)

Random Forests

Introduction

- The motivation for **model aggregation** is to develop procedures that combine the outputs of many "weak" classifiers to produce a powerful "committee."

- **Bootstrap**: choose the best model among B models.
- **Bagging** (bootstrap aggregating): combine linearly several models to reduce the estimation variance and the risk of overfitting

$$\hat{f}(x) = \frac{1}{B}\sum_{b=1}^{B}\hat{f}_b(x).$$

  Example: **random forest**
- **Boosting**: combine a sequence of (elementary) models fit to weighted observations; at each step the weights of badly predicted observations are increased.
  Example : **Adaboost**

Outline

Ensemble methods (I)

- Introduction
- Bumping, bootstrap
- Bagging (bootstrap aggregation)
- Random Forests

Bumping

- Bootstrap is usually used to find the best model among M models.

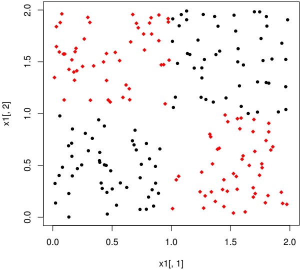
$$m^* = \arg\min_m E_{\mathcal{P}} \left[Y - \hat{f}_m \right]^2$$

- **Algorithm**
  Let $z = \{(\mathbf{x}_1, y_1), \cdots, (\mathbf{x}_n, y_n)\}$ be a training sample.
  For $b = 1, \cdots, B$
  - Randomly sample with replacement $z^{(b)}$ of size n_{app} in z.
  - Fit model $f^{(b)}$.
  - Estimate the *out-of-bag* error $\mathcal{E}_{oob}^{(b)}$.

  Choose model $f^{(b^*)}$ such that $b^* = \arg\min_{b \in \{1, \cdots, B\}} \mathcal{E}_{oob}^{(b)}$

Example

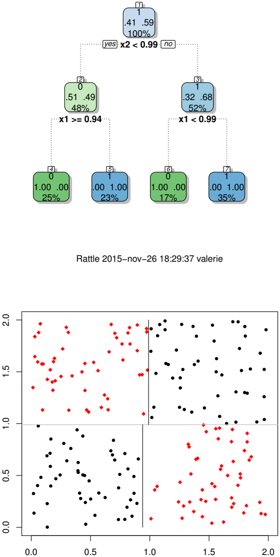
- One looks for a classifier for the following data (n=200).



- 500 trees are fit to bootstrap samples of size 150; the sampling unit is $z_i = (x_i, y_i)$. At each step, the samples which are not in the training set are gathered in the *out-of-bag* sample.
- The classification error is estimated on the *out-of-bag* sample.

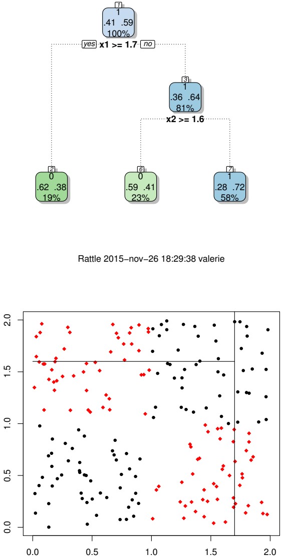
Bumping

Best model : err.min = 0



Rattle 2015−nov−26 18:29:37 valerie

Worst model : err.min = 0.5



Rattle 2015−nov−26 18:29:38 valerie

Bumping

- Since bumping compares different models to the training data, one must ensure that the models have roughly the same complexity.

  Training data: although at each step the out-of-bag data are used to estimate the model performances, at the end the final model has been chosen on the basis of all the train database.

  Complexity: if one of the models has a larger complexity it might be favorised.

  In the case of trees, this would mean growing trees with the same number of terminal nodes on each bootstrap sample.

Outline

Ensemble methods (I)

- Introduction
- Bumping, bootstrap
- Bagging (bootstrap aggregation)
- Random Forests

Bagging

- Bagging is inspired by bootstrap.
- Suppose we fit a model $\hat{f}$ to our training data $\mathcal{S} = \{(x_1, y_1), \cdots, (x_n, y_n)\}$ obtaining prediction $\hat{f}(x)$ at input x.
- Bootstrap aggregation or bagging average the prediction of a collection of B boostrap samples.
- The bagging estimate is defined by

$$\hat{f}_{bag} = \frac{1}{B} \sum_{b=1}^{B} \hat{f}_b(x)$$

- The bagged estimate will differ from the original estimate $\hat{f}(x)$ only when the latter is a nonlinear or adaptive function of the data.
- Bagging helps to reduce the variance but does not change the bias
  Bias: $E(\hat{f}_{bag}) = E(\hat{f}_1)$
  Variance: $Var(\hat{f}_{bag}) = \frac{1}{B} Var(\hat{f}_1)$

Impact of bagging on bias and variance

Bagging helps to reduce the variance but does not change the bias.

- If the predictions $\hat{f}_1(x), \cdots, \hat{f}_B(x)$ are i.i.d,
  Bias: $E(\hat{f}_{bag}(x)) = E(\hat{f}_1(x))$
  Variance: $Var(\hat{f}_{bag}(x)) = \frac{1}{B} Var(\hat{f}_1(x))$
- Here, the bootstrap samples are correlated.
  Lets denote $\rho(x) = corr(\hat{f}_b(x), \hat{f}_{b'}(x))$
  $Var(\hat{f}_{bag}) \simeq \rho(x) Var(\hat{f}_1)$ if B is large.

Trees are unstable: they may change a lot when the training sample is perturbated. They are good candidates for bagging!

Variance reduction

Let's consider a theoretically ideal case.

- $(\mathbf{x}_i, y_i)$, $i = 1, \cdots, n$ are observations sampled according to distribution $\mathcal{P}$ and $f_{\text{bag}}(\mathbf{x}) = E_{\mathcal{P}}(\hat{f}^*(\mathbf{x}))$.
  Here, $\mathbf{x}$ is fixed and the bootstrap samples $(\mathbf{x}_i^*, y_i^*)$, $i = 1, \cdots, n$ are sampled in the population $\mathcal{P}$ (and are independent from the observations). Then [1],

$$
\begin{aligned}
E_{\mathcal{P}}\left[Y - \hat{f}^*(\mathbf{x})\right]^2 &= E_{\mathcal{P}}\left[Y - f_{\text{bag}}(\mathbf{x}) + f_{\text{bag}}(\mathbf{x}) - \hat{f}^*(\mathbf{x})\right]^2 \\
&= E_{\mathcal{P}}\left[Y - f_{\text{bag}}(\mathbf{x})\right]^2 + 2E_{\mathcal{P}}\left[Y - E_{\mathcal{P}}\hat{f}^*(\mathbf{x})\right] E_{\mathcal{P}}\left[E_{\mathcal{P}}\hat{f}^*(\mathbf{x}) - \hat{f}^*(\mathbf{x})\right] \\
&\quad + E_{\mathcal{P}}\left[f_{\text{bag}}(\mathbf{x}) - \hat{f}^*(\mathbf{x})\right]^2 \\
&= E_{\mathcal{P}}\left[Y - f_{\text{bag}}(\mathbf{x})\right]^2 + E_{\mathcal{P}}\left[f_{\text{bag}}(\mathbf{x}) - \hat{f}^*(\mathbf{x})\right]^2 \\
&\geq E_{\mathcal{P}}\left[Y - f_{\text{bag}}(\mathbf{x})\right]^2
\end{aligned}
$$

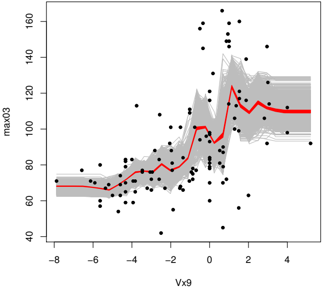
- The second term in the error comes from the variance of $\hat{f}^*(\mathbf{x})$. So that aggregation in the true population never increases the RMSE. This suggest that bagging (aggregation in the data) will often allow RMSE to decrease.

[1] $E_{\mathcal{P}}\left[\hat{f}^*(\mathbf{x}) - E_{\mathcal{P}}\hat{f}^*(\mathbf{x})\right] = E_{\mathcal{P}}\hat{f}^*(\mathbf{x}) - E_{\mathcal{P}}\hat{f}^*(\mathbf{x}) = 0$

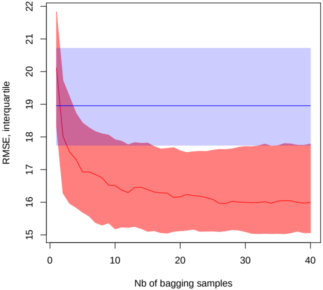
Bagging with kNN and trees

MSE = Bias^2+variance

kNN, interquartile of 50 predictions
kNN, 7 neigh., bagging



Trees
(RMSE vs number of bagging iterations)
Tree, bagging, tree depth=2



In both examples, the variance is drastically reduced.

Bagging for classification

For a K class classification problem,

$$\hat{f}_{\text{bag}}(x) = \arg\max_{k\in\{1,\cdots,K\}} \sum_{b=1}^{B} \delta_{f^{(b)}(x)=k}$$

In classification the reduction of variance does not hold anymore. Bagging a good classifier can make it better, but bagging a bad classifier can make it worse.

- Example of a bad classifier.
  Suppose $Y = 1$ for all x, and the classifier $\hat{G}(x)$ predicts

  $$\begin{gathered} Y = 1 \text{ (for all } x\text{) with probability 0.4} \\ Y = 0 \text{ (for all } x\text{) with probability 0.6.} \end{gathered}$$

  Then the prediction of the bagged classifier will always be 0 (which has a higher probability to occur).
  So the error of $G(x)$ is 0.6 but that of the bagged classifier is 1.0.

Bagging for classification

For a K class classification problem,

$$\hat{f}_{\text{bag}}(x) = \arg \max_{k \in \{1,\cdots,K\}} \sum_{b=1}^{B} \delta_{f^{(b)}(x)=k}$$

In classification the reduction of variance does not hold anymore. Bagging a good classifier can make it better, but bagging a bad classifier can make it worse.

- Example of a good classifier.
  Let the Bayes optimal decision at x be $G(x) = 1$ in a two-class example.
  Suppose each of the weak learners G_b have an error-rate $e_b = e < 0.5$, and let

  $$S_1(x) = \frac{1}{B} \sum_{b=1}^{B} \mathbb{I}_{G_b^*(x)=1}$$

  be the consensus vote for class 1.
  Since the weak learners are assumed to be independent[2],

  $$S_1(x) \sim \mathcal{B}(1, 1-e)$$

  and $P(S_1(x) > 1/2) \to 1$ when B gets large.

[2] mean of B independent $\{0, 1\}$ samples

Advantages and disadvantages

- Advantages:
  - Many weak learners aggregated typically outperform a single learner over the entire set, and has less overfit
  - Removes variance in high-variance low-bias weak learner
  - Can be performed in parallel, as each separate bootstrap can be processed on its own before combination

- Disadvantages:
  - For weak learner with high bias, bagging will also carry high bias into its aggregate
  - Loss of interpretability of a model.
  - Can be computationally expensive depending on the data set

Outline

Ensemble methods (I)

- Introduction
- Bumping, bootstrap
- Bagging (bootstrap aggregation)
- Random Forests

Random forest

- A **random forest** is defined from a set of trees.
- Let $T_b(x)$, $b = 1, \cdots, B$ be B predictors associated to trees $(T_b)_b$.
  The random forest predictor is obtain by aggregating the tree collection:

$$\hat{T}_B(x) = \frac{1}{B} \sum_{b=1}^{B} T_b(x)$$

- The most famous random forests are due to L. Breiman (2000).

Random forest

- The trees of the forest are fit to bootstrap samples.
- In Breiman's random forest, at each node of a given tree, the split is based on the "best" variable of a set composed of *m* variables randomly chosen.
- This trick allows to reduce the correlation between the trees of the forest.
- Note that there are two sources of randomness:
  - the bootstrap sampling,
  - the sampling of the variables.
- "*m* variables randomly chosen"
  It is a similar idea as the *dropout* of the deep learning algortihms.

Random forest, algorithm

Random Forest

- $\mathcal{S} = \{(\mathbf{x}_1, y_1), \cdots, (\mathbf{x}_n, y_n)\}$ a learning set.
- For $b = 1, \cdots, B$

```
(a) Draw a bootstrap sample $\mathcal{S}^{(b)}$ of size $n$ among $\mathcal{S}$.
(b) Grow a tree $\mathcal{S}^{(b)}$ by recursively repeating the following steps
until the minimum node size $n_{min}$ is reached.
for each terminal node of the current tree,
    i. Select $m$ variables at random from the $p$ variables.
    ii. Pick the best variable/split-point among the $m$.
    iii. Split the node into two daughter nodes.
(d) Estimate the generalization error using out-of-bag samples .
```

- For a new observation, $\mathbf{x}$, y is predicted by the mean of the responses of the B regression trees.

Ref : Leo Breiman (2001), "Random Forests", Machine learning, vol. 45, p. 5-32.

Properties of random forest

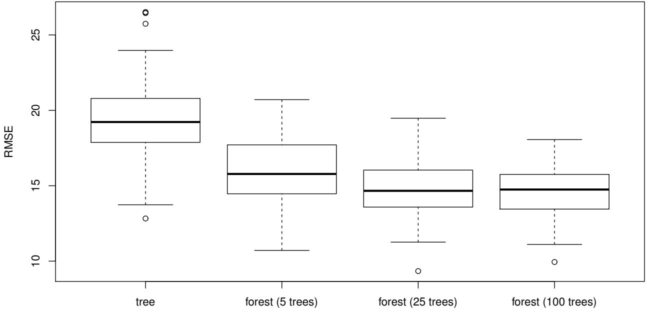
- Random forest are easy to use.
- They lead to good estimates (or predictors) even if the data are complex (high dimension, missing values).
- The estimate is robust in the sense that it is not sensitive to the choice of the algorithm parameters (*B*, *m*, ...)
- However, *B* should be large in order to reduce the variance

$$Var(\hat{T}_B) = \rho(x) Var(T_k(x)) + \frac{1-\rho(x)}{B} Var(T_k(x))$$

- Bias is not reduced by using random forest.

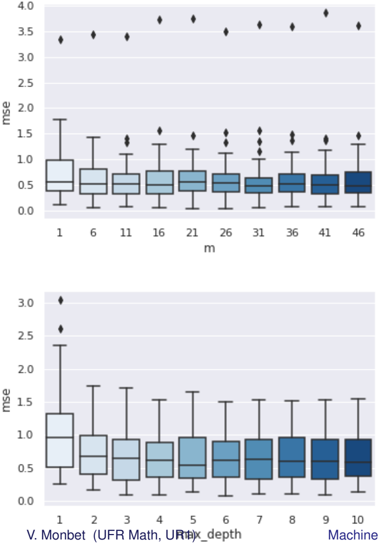
Random forest in practice

- Usually, m has to be chosen close to $p/3$. But, Breiman shows with examples that the results may be very good even if $m = 1$. Example below, $m = 2$.
- The larger m, the higher the correlation between the trees.
- When the number of variables is large, but the fraction of relevant variables small, random forests are likely to perform poorly with small m.
- Random Forest stabilizes quite fast by increasing the number of trees.



Regression example: Fat concetration prediction (cookies).

Variables: NIR spectra
42 observations, about 700 highly correlated variables
Validation criteria: MSE.



Each experiment is repeated 30 times.

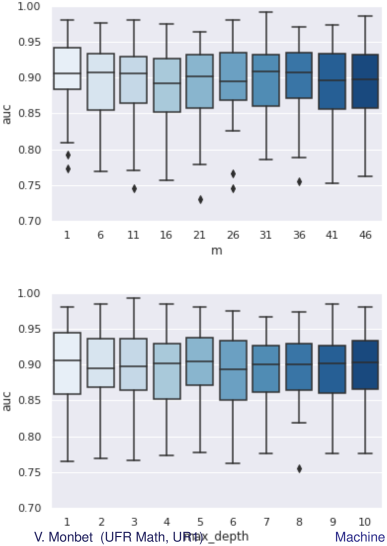
Top panel shows the changes in AUC with respect to *m* for a tree depth equal to 5.

Bottom panel shows the changes in AUC with respect to the tree depth for a frozen *m* = 27.

The impact of the algorithm's parameters is low.

Classification example : Relapse after a treatment again cancer.

Variables : gene expression
97 observations, 754 variables
Validation cirteria : AUC.



Each experiment is repeated 30 times.

Top panel shows the changes in AUC with respect to *m* for a tree depth equal to 5.

Bottom panel shows the changes in AUC with respect to the tree depth for a frozen *m* = 27.

The impact of the algorithm's parameters is low.

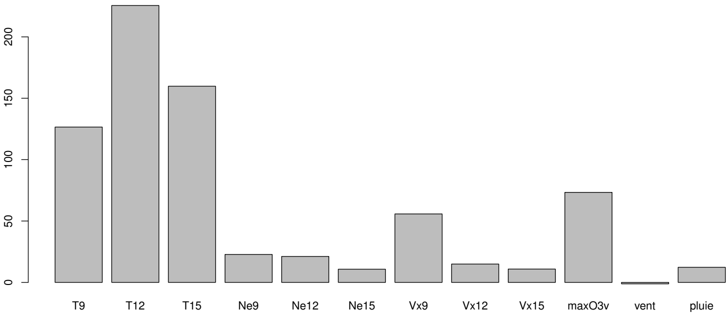
Random Forest interpretability

- Variable importance plots can be constructed for random forests.
- The most advanced variable importance measure available in random forests is the *permutation accuracy importance* measure.

- **Principle** : permutation test
  - For a given predictor variable X_j, the observations are randomly permuted, and the out-of-bag samples are predicted.
  - The prediction accuracy decreases substantially compared to the original ones, if X_j is associated with the response.
- Thus, a reasonable measure for variable importance is the difference in prediction accuracy before and after permuting X_j.

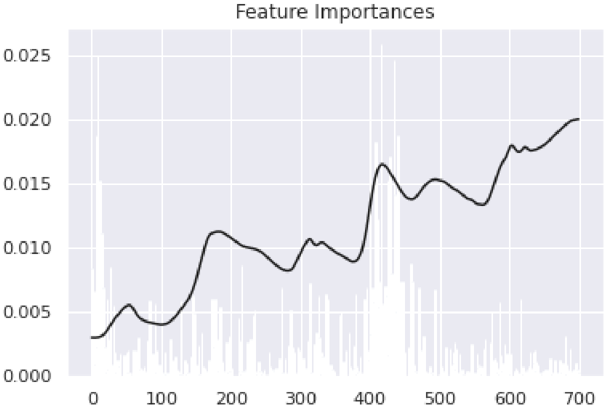
- **Comments**
  - By randomly permuting the predictor variable X_j, its original association with the response Y is broken.
  - For variable selection purposes the advantage of the random forest permutation accuracy importance measure as compared to univariate screening methods is that it covers the impact of each predictor variable individually as well as in multivariate interactions with other predictor variables.

Variable importance, Ozone data (regression)



Regression example: Fat concetration prediction (cookies).

White bars illustrate the importance measure.
For interpration, a spectrum has been had on the plot.
The area between indices 400 and 480 is important. It corresponds to a location where the lipid mollecules vibrate.



Concluding remarks

- In bagging, the elementary models (or weak learners) are fit on subsamples and combined by a majority vote (or a mean for regression problems).
- The subsamples should be small (for instance 1/3 of the learning database).
- Bagging is an ensemble method which helps to decrease the learners's variance.
- Random forest is the most famous bagging algorithm.
- The vote (or mean) computation can be adapted to take into account the performance of the weak models using weights.

Outline

Introduction

Dimension Reduction

Unsupervised learning

Supervised learning

Linear model

Data driven supervised learning

Ensemble methods (I)

Ensemble methods (II)