Predictive Modelling - V

Ensemble Models

Rita P. Ribeiro

Data Mining I - 2023/2024

Summary

- Ensembles: Motivation
- Types Ensembles
- Ensemble Methods
  - Random Forest
  - AdaBoost
  - XGBoost

Ensembles

Predictive Modelling: Where we at?

- Probabilistic Approaches
  - e.g. Naive Bayes, Bayesian Networks
- Mathematical Formulae
  - e.g. multiple linear regression
- Logical Approaches
  - e.g. CART
- Distance-based Approaches
  - e.g. kNN
- Optimization Approaches
  - e.g. SVM, ANN
- Ensemble Approaches
  - e.g. Random Forest, XgBoost

Ensemble Models

- Ensembles: collections of models that are used together to address a certain prediction problem
- Different learning algorithms exploit:
  - different languages for representing generalizations of the examples;
  - different search spaces;
  - different evaluation functions of the hypothesis;
- For complex problems it is hard to find a model that “explains” all observed data
- There is no overall better algorithm $\rightarrow$ *No free lunch theorem*

Ensemble Models

- Averaging over a set of models typically leads to significantly better results, given certain conditions.
- An ensemble of classifiers improves over individual classifiers iif (Dietterich 2002):
  - they perform better than random guess;
  - they have non-correlated errors;
  - they commit errors in different regions of the instance space.

Ensemble Models

How to achieve such diversity?

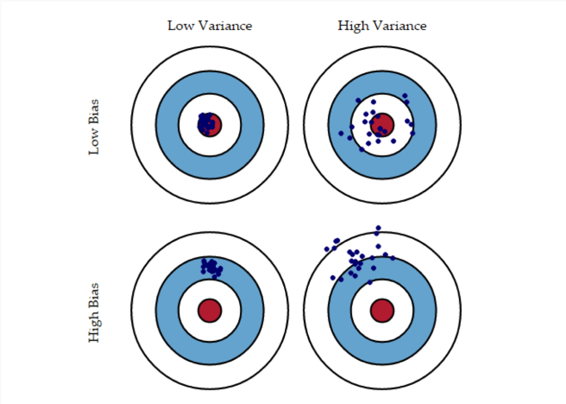
- Combining outputs in different ways
- Perturbing the set of training examples
  - Homogeneous Models (Bagging, Boosting)
  - Heterogeneous Models (Cascading, Stacking)
- Perturbing the set of attributes

Bias-Variance Trade-off

The Bias-Variance Decomposition of Prediction Error

- The prediction error of a model can be split in two main components: the bias and the variance components

- bias: error that is due to the poor ability of the model to fit the seen data
- variance: error related to the sensibility of the model to the given training data



Bias-Variance Trade-off

When learning a prediction model, there is bias-variance trade-off.

- Decreasing the bias by adjusting more to the training sample $\rightarrow$ higher variance - the over-fitting phenomenon
- Decreasing the variance by being less sensitive to the given training data $\rightarrow$ higher bias

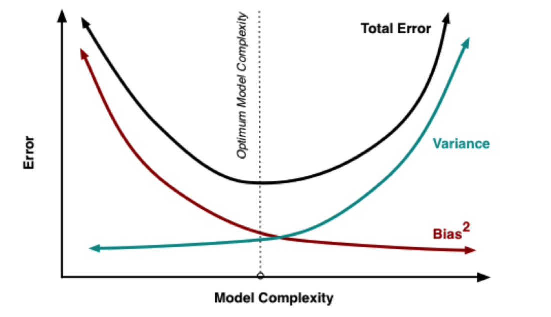


Fig. 6 Bias and variance contributing to total error.

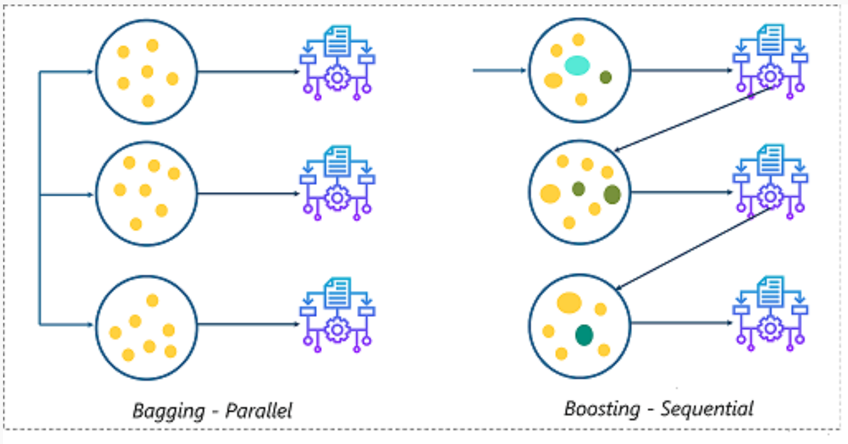
Bias-Variance Trade-off

- Ensembles are able to reduce both components of the error
- Their approach consists on:
  - applying the same algorithm to different samples of the data
  - use the resulting models in a voting/averaging schema to obtain predictions for new cases

Types of Ensembles

Types of Ensembles

- Independent or Parallel Models
- Coordinated or Sequential Models



Types of Ensembles: Independent or Parallel Models

- Construct the models independently in a way that ensures some diversity among them
- How to reach diversity?
  - applying the models on somewhat different training sets
  - applying the models on data sets using different predictors

Types of Ensembles: Coordinated or Sequential Models

- Construct a “larger” model by composing it from smaller and integrated models
- Each individual model has a weighted participation in the ensemble predictions
- What is the right component models and their respective weight to achieve a good predictive performance?

Ensemble Methods

Ensembles using Independent Models: Bagging

Bagging or Bootstrap Aggregating (Breiman 1996)

- Method that obtains a set of k models using different bootstrap samples of the given training data
  - sample with replacement of the same size as the available data
  - for each learner, a small proportion of examples will be different
- If the base learner has a high variance (i.e. very sensitive to variations on the training sample), this will ensure diversity among the k models
- Bagging should be applied to base learners with high variance

Ensembles using Independent Models: Bagging

- Requires unstable algorithms (greedy like)
- Algorithms sensible to small perturbations of the training set;
  - Decision trees, Rule learners, etc.
- Easy to implement with any algorithm.
- Easy to implement in parallel environments.
- The bias-variance argument:
  - Error decreases due to reduction in the variance component.

Ensembles using Independent Models: Random Forests

Varying the Predictors

- Another way of generating a diverse set of models is by using different randomly chosen predictors
- The idea is similar to bagging but instead of generating samples of the cases we generate samples of the variables

Ensembles using Independent Models: Random Forests

Random Forests (Breiman 2001)

- Combine the ideas of bagging together with the idea of random selection of predictors
- Set of tree-based models where each tree is obtained from a bootstrap sample of the original data and uses random selection of variables during tree growth

Ensembles using Independent Models: Random Forests

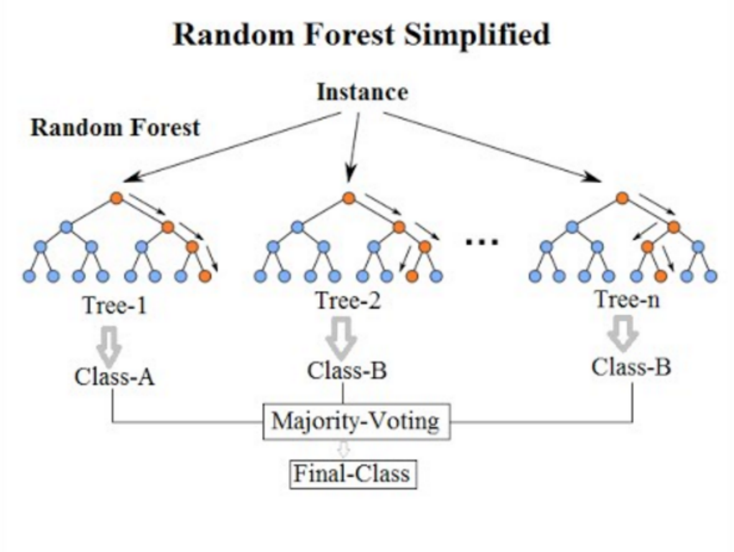
Learning phase (main idea)

- For $t = 1$ to T, T is number of trees
  - draw a random sample with replacement from the training set D_t
  - train a tree model $h_t(\mathbf{x})$ on D_t without pruning
  - at each candidate split in the learning process, uses a random subset of the m features.
- Return $\{h_t(\mathbf{x}) | 1 \leq t \leq T\}$

Ensembles using Independent Models: Random Forests

Prediction phase

- Predict the class obtained by majority vote, or the value by averaging the output of each tree



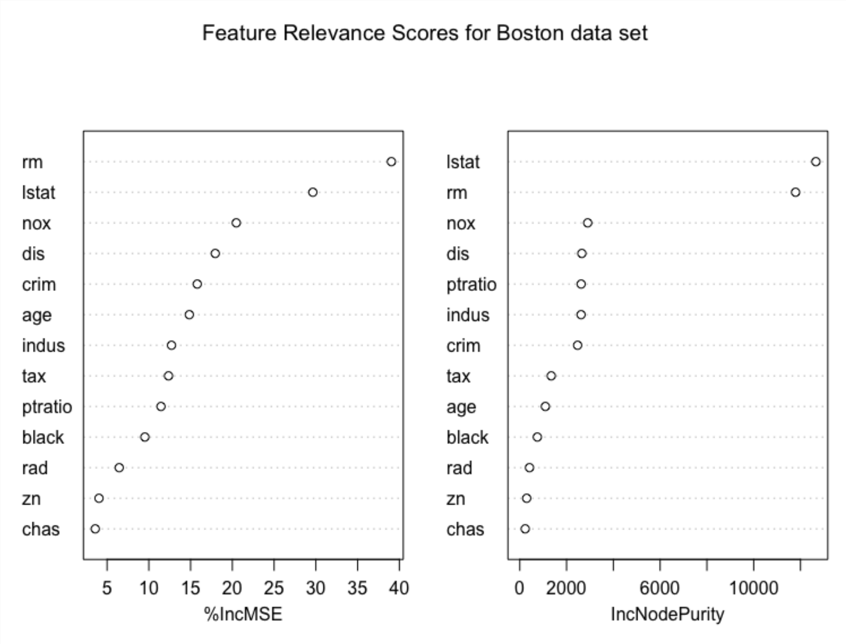
Ensembles using Independent Models: Random Forests

Other Uses of Random Forests: Variable Importance

- Which variables have the most predictive power?
- Two importance measures:
  - how much the accuracy decreases / mean square error increases when the variable is excluded
  - how much the impurity decreases when the variable is chosen to split a node.

Ensembles using Independent Models: Random Forests

Variable Importance: an example



Ensembles using using Independent Models: Random Forests

Hyperparameters

- Number of trees
  - recommended number of trees is 1000.
  - to obtain more reliable statistics for the attribute importance, 5000 trees are recommended.
- Number of attributes to randomly select at each node
  - it must be tuned
  - its optimum value is problem dependent.
  - rule of thumb: $\sqrt{p}$, p is the number of predictive attributes

Ensembles using Independent Models: Random Forests

Pros

- Do not require elaborate tuning of the hyper-parameters. Often these can/should be optimized.
- The most important parameter to tune is the number of trees to grow, typically the larger the best.
- Do not need to worry about creating very complex trees.

Cons

- Do not provide the interpretability level of a Decision Tree

Ensembles using Coordinated Models: Boosting

Boosting (Schapire 1990)

Can a set of weak learners create a single strong learner?

- A “weak” learner is a model that alone is unable to correctly approximate the unknown predictive function
- A “strong” learner has that ability
- Boosting algorithms work by iteratively creating a strong learner by adding at each iteration a new weak learner to make the ensemble
- Weak learners are added with weights that reflect the learner’s predictive power

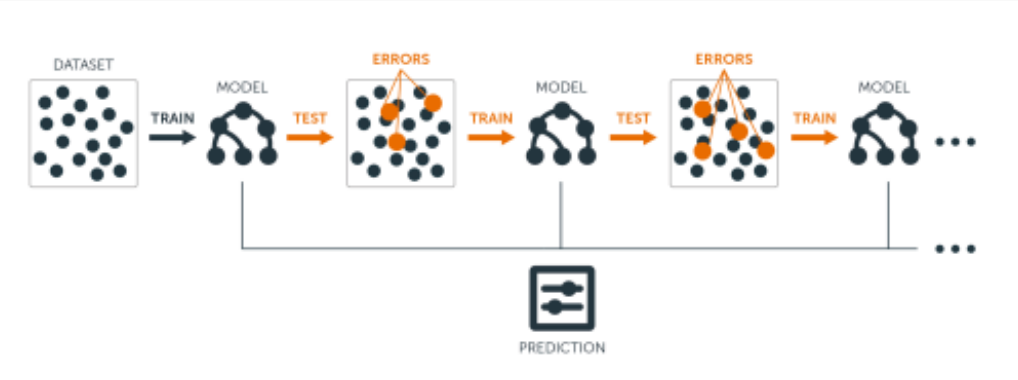
Ensembles using Coordinated Models: Boosting

- After each addition the data is re-weighted such that cases that are still poorly predicted gain more weight
- The weight indicates the probability of the example being select in a uniform sampling;
- This means that each new weak learner will focus on the errors of the previous ones

- It fits many real-world problems, where observed examples tend to have different learning difficulty levels.
  - e.g. examples close to the decision surface are typically more difficult

Ensembles using Coordinated Models: Boosting

- The prediction: weighted voting/average of each learner.



Ensembles using Coordinated Models: Boosting

Three ways through which boosting can be carried out:

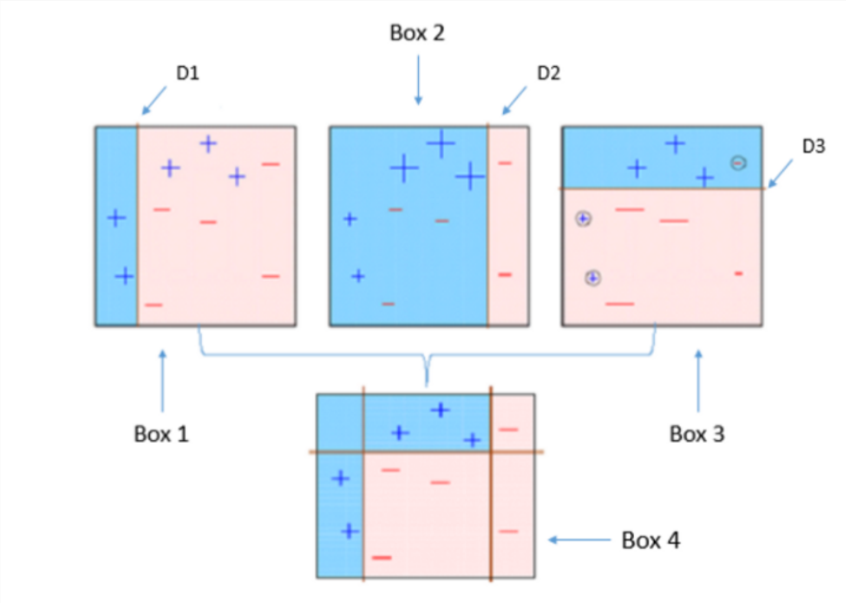
- Adaptive Boosting or AdaBoost (Freund and Schapire 1996)
- Gradient Boosting Machine (Friedman 2000)
- XGBoost (Chen and Guestrin 2016)

Ensembles using Coordinated Models: AdaBoost

AdaBoost or Adaptive Boosting (Freund and Schapire 1996)

- Iterative process: new models are added to form an ensemble
- Adaptive: at each new iteration of the algorithm, the new models are built to try to overcome the errors made in the previous iterations
- At each iteration the weights of the training cases are adjusted
- Cases wrongly predicted get their weight increased to make new models focus on accurately predicting them
- The main hyperparameter is number of iterations
- AdaBoost was created for classification although variants for regression exist

Ensembles using Coordinated Models: AdaBoost



Source: https://medium.com/divyagera2402

Ensembles using Coordinated Models: Boosting

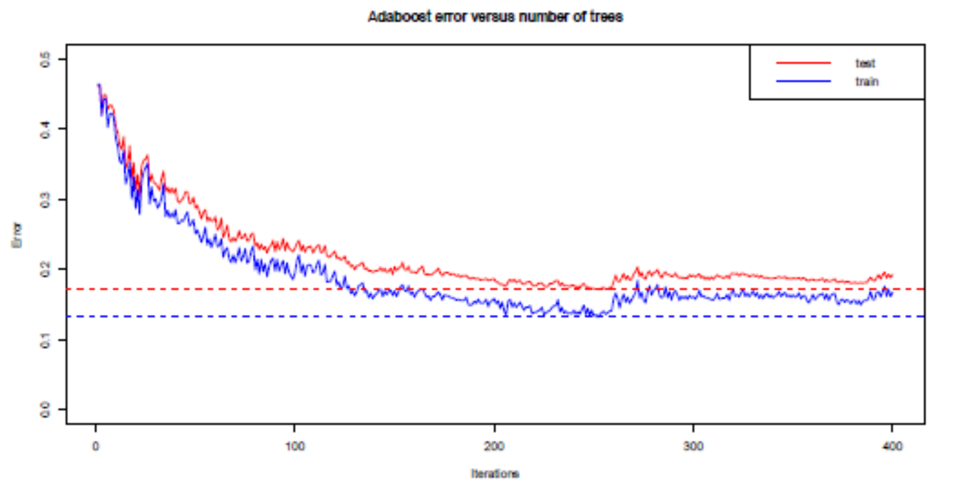
The Algorithm (main idea)

- Start with uniform weights $w_i^{(1)} = 1/|D|$ for all $\mathbf{x}_i \in D$
- For $t = 1$ to T
  - build weak model $h_t(\mathbf{x})$
  - calculate weighted error $e_t = \sum_i w_i^{(t)} I(y_i \neq h_t(\mathbf{x}_i))$
  - the weight of this weak model: $\alpha_t = \frac{1}{2} ln\left(\frac{1-e_t}{e_t}\right)$
  - update case weights $w_i^{(t+1)} = \frac{w_i^{(t)} exp(-\alpha_t I(y_i \neq h_t(\mathbf{x}_i)))}{Z_t}$ where Z_t is chosen to make all $w_i^{(t+1)}$ sum up to 1
- Return a form of additive model composed of t weak models

$$H(\mathbf{x}) = \sum_{t=1}^{T} \alpha_t h_t(\mathbf{x})$$

Ensembles using Coordinated Models: Boosting

Evolution of the error as you increase the number of weak learners.



Ensembles using Coordinated Models: GBM

Gradient Boosting Machine (Friedman 2000)

- Sequential ensemble learning
- Contrary to AdaBoost, it does not adjust the examples weights at every iteration
- It fits the new learner to the residual errors made by the previous learner
- The present learner is always more effective than the previous one
- Goal: at each step, adds a weak learner to increase the performance and build a strong learner.

Ensembles using Coordinated Models: GBM

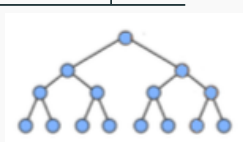
- Re-defines boosting as a numerical optimization problem
- Objective: minimize the loss function of the model by adding weak learners using a gradient-descent procedure.
- Major difference: how it identifies the shortcomings of weak learners (e.g. decision trees).
- It uses gradients in the loss function as a measure indicating how good are model's coefficients are at fitting the underlying data
- Like AdaBoost, it can be used for both classification and regression problems

Ensembles using Coordinated Models: GBM

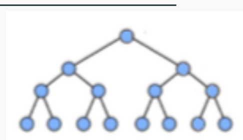
Learning phase (main idea)

- Build an additive tree model by adding new trees to complement the already-built ones

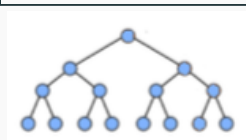
#	$\mathbf{x}$	y
1	$\mathbf{x}_1$	y_1
2	$\mathbf{x}_2$	y_2
3	$\mathbf{x}_3$	y_3
...	...	...

h_1: 

#	$\mathbf{x}$	y
1	$\mathbf{x}_1$	$\ell(y_1, h_1(\mathbf{x}_1))$
2	$\mathbf{x}_2$	$\ell(y_2, h_1(\mathbf{x}_2))$
3	$\mathbf{x}_3$	$\ell(y_3, h_1(\mathbf{x}_3))$
...	...	...

h_2: 

#	$\mathbf{x}$	y
1	$\mathbf{x}_1$	$\ell(y_1, h_1(\mathbf{x}_1) + h_2(\mathbf{x}_1))$
2	$\mathbf{x}_2$	$\ell(y_2, h_1(\mathbf{x}_2) + h_2(\mathbf{x}_2))$
3	$\mathbf{x}_3$	$\ell(y_3, h_1(\mathbf{x}_3) + h_2(\mathbf{x}_3))$
...	...	...

h_3: 

Ensembles using Coordinated Models: GBM

Learning phase (main idea) - cont.

- Objective: minimize $Obj = \sum_{i=1}^{n} \ell(y_i, \hat{y}_i) + \sum_{k=1}^{K} \mathscr{R}(h_k)$

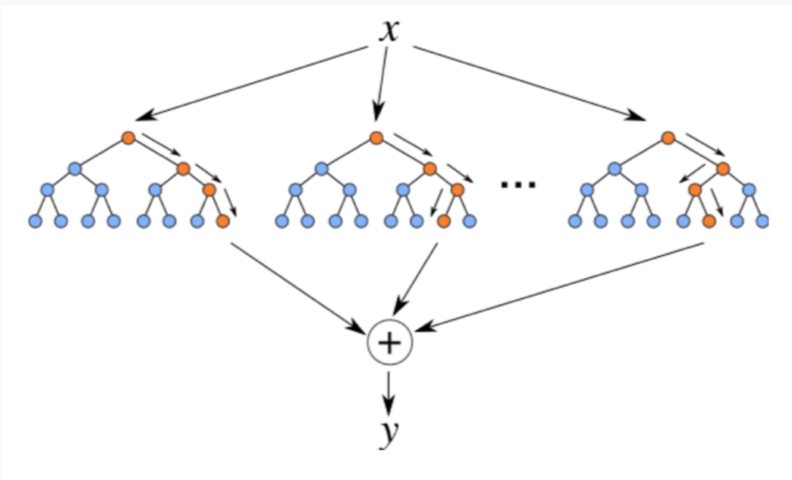
where

- $\sum_{i=1}^{n} \ell(y_i, \hat{y}_i)$ is the training loss
  - measures how well the model fits on training data
- $\sum_{k=1}^{K} \mathscr{R}(h_k)$ is the regularization term
  - measures the complexity of trees (nr of leafs and L_2-norm of leaf scores)

Ensembles using Coordinated Models: GBM

Prediction phase

- Response is the optimal linear combination of all decision trees



Ensembles using Coordinated Models: GBM

Hyperparameters

- Learning rate (α) is a multiplying factor on the errors for the subsequent trees.
  - It controls how fast the model learns: the lower α, the slower the model learns.
  - The advantage of slower learning rate: the model becomes more robust and avoids overfitting.
  - However, learning slowly comes at a cost: it takes more time to train the model
- Number of trees used in the model.
  - If the learning rate is low, we need more trees to train the model.
  - However, it creates a high risk of overfitting to use too many trees.

Ensembles using Coordinated Models: XGBoost

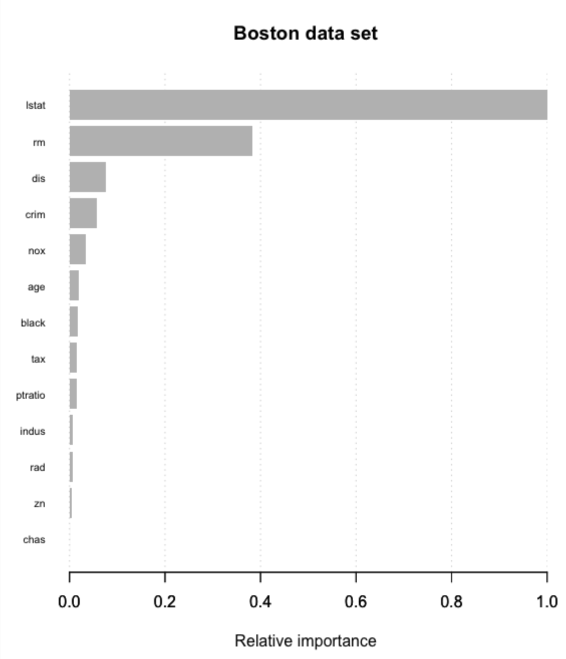
eXtreme Gradient Boosting (XGBoost) (Chen and Guestrin 2016)

- An advanced version of Gradient Boosting Method
- Software and hardware optimization
  - a scalable tree boosting system
- Some features:
  - clever penalisation of trees: weights of the trees that are calculated with less evidence is shrunk more heavily
  - extra randomisation parameter to reduce correlation between trees
  - parallelization, cache optimization, distributed computing, etc.

Ensembles using Coordinated Models: XGBoost

Feature Importance

- How useful each feature was in the construction of the boosted decision trees?



- The more the feature is selected for splitting, the higher its relative importance.
- Importance is calculated for a single decision tree by number of times the feature is selected for splitting, weighted by the improvement to the model as a result of each split.
- The feature importances are then averaged across all of the the decision trees within the model.

Ensemble Methods: Wrap-up

- Well designed ensembles of predictive models allow improvement of performance over their individual elements.
- Necessary conditions:
  - variability between elements;
  - low error correlation;
  - each individual model must be better than a random choice

Ensemble Methods: Wrap-up

Bagging Methods

- Error reduction due to reduction in variance;
- Effective with unstable models;

Boosting Methods

- Error reduction due to reduction in bias and variance;
- Risky in problems with noise (increase of the error);

References

References

Breiman, Leo. 1996. "Bagging Predictors." *Machine Learning* 24 (2): 123–40.

———. 2001. "Random Forests." *Machine Learning* 45 (1): 5–32.

Chen, Tianqi, and Carlos Guestrin. 2016. "XGBoost: A Scalable Tree Boosting System." In *22nd ACM SIGKDD*, 785–94.

Dietterich, Thomas. 2002. "Ensemble Learning." In *The Handbook of Brain Theory and Neural Networks*, 2nd ed., 405–8. MIT Press.

Flach, Peter. 2012. *Machine Learning: The Art and Science of Algorithms That Make Sense of Data*. Cambridge University Press.

Freund, Yoav, and Robert E. Schapire. 1996. "Experiments with a New Boosting Algorithm." In *13th ICML*, 148–56. Morgan Kaufmann.

Friedman, Jerome H. 2000. "Greedy Function Approximation: A Gradient Boosting Machine." *Annals of Statistics* 29: 1189–1232.

Gama, João. 2018. "KDD Course." Slides.

Schapire, Robert E. 1990. "The Strength of Weak Learnability." *Machine Learning* 5 (2): 197–227.

Torgo, Luís. 2017. "Data Mining i Course." Slides.