Applied Machine Learning

Random Forests

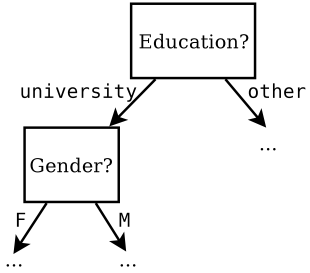
UNIVERSITY OF GOTHENBURG

CHALMERS

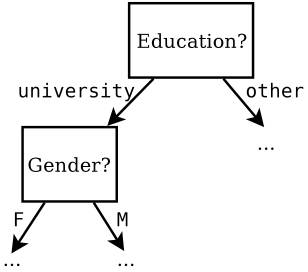
Richard Johansson

`richard.johansson@cse.gu.se`

decision tree recap

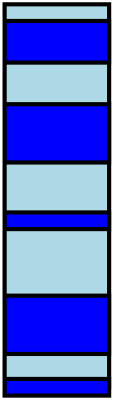


decision tree recap

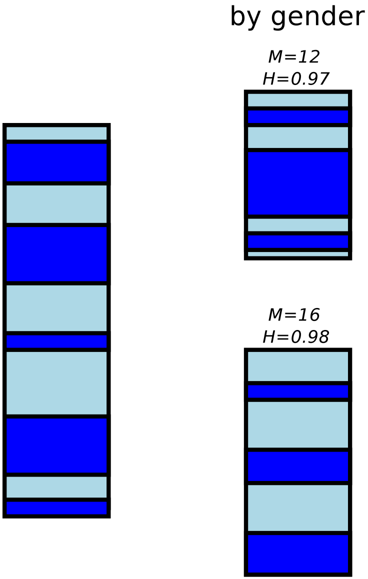


- learning algorithm (simplified):
  1. select the feature F that gives the "best" split
  2. make a tree with F at the top
  3. split into subsets based on F
  4. make a tree for each subset

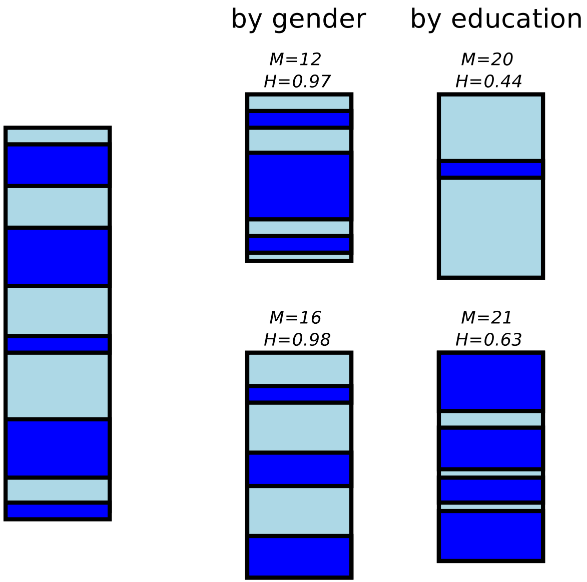
selecting the feature for the top node



selecting the feature for the top node

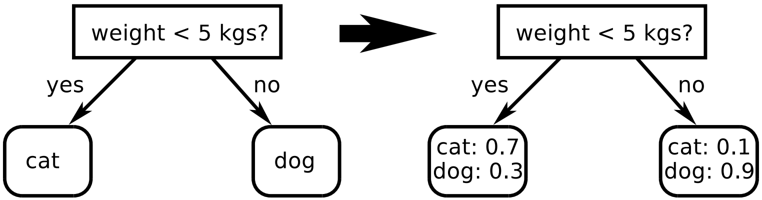


selecting the feature for the top node



probabilities in decision tree classifiers

- optionally, a decision tree classifier may output **probabilities** in a leaf node



- they are estimated using simple MLE

decision trees aren't fantastic

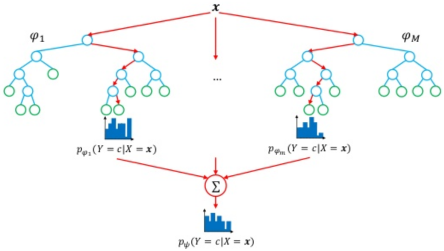
"Decision trees are a popular method for various machine learning tasks. Tree learning come[s] closest to meeting the requirements for serving as an off-the-shelf procedure for data mining, because it is invariant under scaling and various other transformations of feature values, is robust to inclusion of irrelevant features, and produces inspectable models. However, they are seldom accurate."

[Hastie et al., *The Elements of Statistical Learning*]

- decision trees on their own are usually not that successful
- they tend to overfit

random forests

- decision trees are more useful when part of an ensemble
- the most famous type of tree ensemble is called a **random forest** (Breiman, 2001)

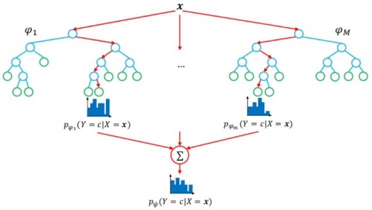


[source]

- often a good default choice: robust in practice
  - tree-based ensembles (RF, boosting) often dominate in competitions for various prediction tasks, such as **Kaggle**
  - see also Fernández-Delgado et al. (2014) *Do we Need Hundreds of Classifiers to Solve Real World Classification Problems?*

random forests: implementation

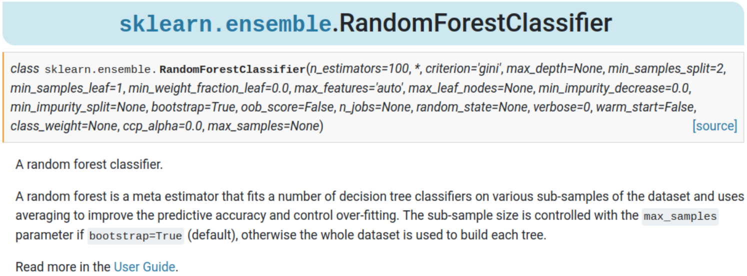
- each tree in the ensemble is trained on its own sampled training set (**bagging**)
- the recursive tree learning algorithm is modified: each time we build a tree node, only a **subset** of the features is considered
  - typically, $\sqrt{|F|}$ features if $|F|$ is the total number
- at prediction time, the outputs from the trees are aggregated
  - regression: outputs are averaged
  - classification: voting or averaging of probabilities



[source]

hyperparameters of random forests

- how many trees in the ensemble?
- how many features to consider when building a node?
- plus all hyperparameters for regular decision trees
  - maximal tree depth
  - feature selection criterion
  - ...



sklearn.ensemble.RandomForestClassifier

class `sklearn.ensemble.`**`RandomForestClassifier`**(*n_estimators=100, *, criterion='gini', max_depth=None, min_samples_split=2, min_samples_leaf=1, min_weight_fraction_leaf=0.0, max_features='auto', max_leaf_nodes=None, min_impurity_decrease=0.0, min_impurity_split=None, bootstrap=True, oob_score=False, n_jobs=None, random_state=None, verbose=0, warm_start=False, class_weight=None, ccp_alpha=0.0, max_samples=None*) [source]

A random forest classifier.

A random forest is a meta estimator that fits a number of decision tree classifiers on various sub-samples of the dataset and uses averaging to improve the predictive accuracy and control over-fitting. The sub-sample size is controlled with the `max_samples` parameter if `bootstrap=True` (default), otherwise the whole dataset is used to build each tree.

Read more in the User Guide.

pros and cons of random forests

- **pros**:
  - more robust than a single decision tree
  - easy to tune: often work well “out of the box”
  - easy to mix different types of features, good for tabular data
  - little need for feature normalization
- **cons**:
  - less interpretable than a single decision tree
  - computationally heavier than a single tree
  - need “nice” features (e.g. not images, signals, text)

references

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

M. Fernández-Delgado, E. Cernadas, S. Barro, and D. Amorim. 2014. Do we need hundreds of classifiers to solve real world classification problems? *JMLR* 15:2133–3181.