Winter Course Statistical Learning

Tree ensembles: Bagging and Random Forests

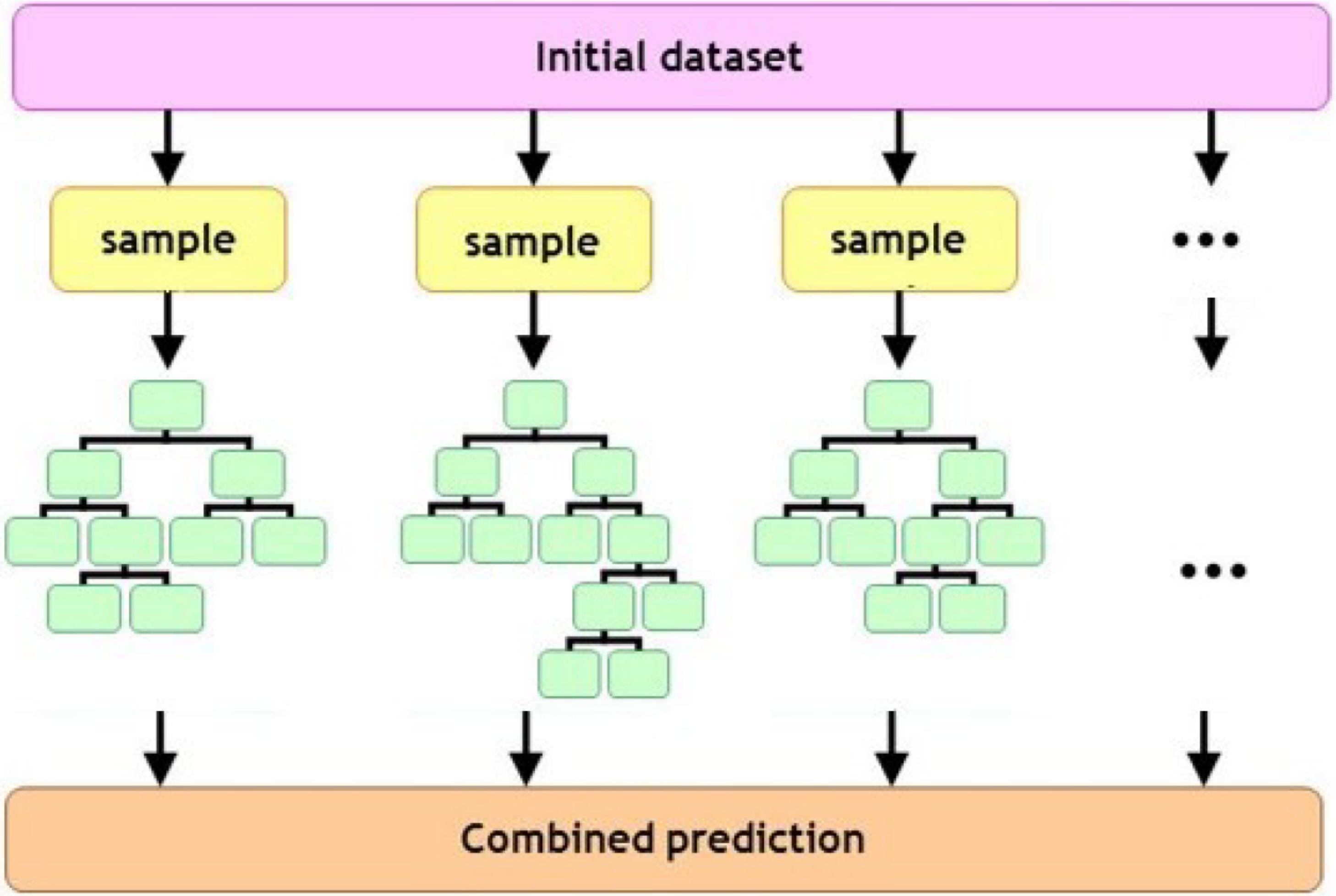
Marjolein Fokkema

Leiden University

Single trees

- Good: Interpretability
- Bad: Not most accurate prediction method
- Ugly: Instability

Ensembling trees



Bagging

- Draw B samples from the (initial) training dataset.
  - Can use bootstrap (bagging) or sub (subagging) sampling.

- Fit a tree $\hat{f}_b(x)$ on each sample.

- Final predictive model averages over individual trees' predictions:

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

  - For regression, averaging is straightforward.
  - For classification, individual trees predictions are generally class labels. The prediction of the ensemble is the proportion of trees predicting the target class.

Random forest

Same as bagging, in addition:

- Select random sample of *mtry* candidate predictors for *every* split.
  - Random sampling of both rows and columns makes trees more dissimilar, so ensembling more effective (take advantage of *instability*!).
  - Particularly effective in presence of multicollinearity, correlated predictors all get a chance to be selected for splitting.

- Final predictive model is again:

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

Parameter tuning

Random forests are said to work well 'out of the box'. But tuning the following parameters (or choosing non-default settings) can improve performance:

- Try a range of `mtry` (the number of candidate predictors for each split).
- Sampling strategy: Bootstrap sampling yields higher inclusion frequencies for noise variables than subsampling (mostly disadvantageous for interpretation, e.g., variable importances, not so much for predictive accuracy).
- Tree depth: Restricting tree size to be lower than default (e.g, increase `nodesize` which defaults to minimum 5 observations per terminal node for regression, 1 for classification; or decrease `maxnodes`, the maximum number of terminal nodes.

Out-of-bag (OOB) error

Gives a running estimate of generalization error (need no separate test set):

- For every training observation i, compute $\hat{y}_i^{OOB}$: The predicted value, based only on trees that were fitted on samples *excluding* observation i.

- Allows computing OOB MSE (or other error measure) for the model using the usual formulas, e.g.:

$$MSE = \frac{1}{N} \sum_{i=1}^{N} (\hat{y}_i - y_i)^2$$

Although computed on training data, OOB error gives more realistic, less optimistic estimate of generalization error than standard training error.

Interpretation: Variable importance

Importance of a predictor variable j can be computed in many (!) ways. E.g.:

- Training error: Sum over the error reductions resulting from each split involving variable j (comparable to sums of squares in GLMs).

- OOB permutation importance:

  1. Compute OOB error, e.g. MSE_{OOB}.
  2. For each variable j ($j = 1, \ldots, p$):
     - Permute values of x_j.
     - Compute OOB predictions using observed values for all variables $\neq j$ and permuted values for variable j. Recompute MSE_{OOB}.
     - Difference between the MSE_{OOB} under 1) and 3) is the importance of variable j.

Importances: Words of warning

- ”Importance” may sound exactly like the quantity we want to know, but given the different and complex computations, they’re a bit of a black box.
  - Importance of the same variable will differ between different fitted models and between different importance measures.
  - No statistical theory for the distribution of importances, so no valid inference.

- Use importances as a rough and approximate ranking.

Interpretation: Partial dependence functions

- The effect of predictor j can be computed using a partial dependence function:

$$\bar{f}(x_j) = \frac{1}{n}\sum_{i=1}^{n} \hat{f}(x_j, x_{i\setminus j})$$

- where x_j indicates the predictor variable(s) of interest, and $x_{\setminus j}$ are all the remaining variables.
- This computes *marginal* effects (cf. GAMs where effects are additive and *conditional* effects can be computed).
- Like variable importances, partial dependence plots should be interpreted with care:
  - Possible interactions of x_j with other variables are averaged over.
  - Different packages may use different computation strategies (e.g., compute using training versus test observations).

Tuning parameters: Bagging and random forests

- `ntree`: Number of samples to draw / trees to generate. Default of 500 often suffices, does not hurt to use more (e.g., $\geq 1,000$).

- `mtry`: Number of predictors to be considered for each split; p for bagging; random forests use $\sqrt{p}$ for classification, $p/3$ for regression. Can use CV to obtain optimal value.

- `replace` and `sampsize`: Sampling strategy and fraction. Bootstrap sampling is sensible default, but may increase inclusion frequency for noise variables. Subsampling with sampling fraction of .632 may be preferred.

- `maxnodes` or `nodesize`: Large trees can lead to unstable results when there are many multicollinear predictors at best weakly related to the response (Segal, 2003). Thus, smaller trees (with optimal tree size e.g., determined through CV) may perform better.