Machine Learning

Bagging and Boosting

S. Nõmm

[1]Department of Software Science, Tallinn University of Technology

24.04.2018

Lectures and practices in May

- Practice on 26.04 will take place in a form of consultation also many students has to defend their practices.
- There is no lecture on 01.05!!!
- Practice on 03.05 as usual.
- Lecture on 08.05 as usual.
- Practice on 10.05 defence of home assignment II.
- Closed book test II on 15.05 usual lecture time!
- Practice on 17.05 consultation + defence of home assignment II.
- Session starts on 21.05.

Bootstrap

- Remind what is the main goal of cross validation.
- Let $Z = (z_1, \ldots, z_n)$ is the training set.
- Draw randomly data sets with replacement (the samples are independent) from Z. This will result in B *bootstrap* data sets.
- Fit the model for each of B data sets. Examine behaviour over B replacements.
- This approach allows to estimate any aspect of distribution $S(Z)$.

Bagging

- Induced from the bootstrap technique (which is used to assess accuracy of estimate).
- Draw B samples with replacements and train the model on each sample.
- The bagging estimate then is defined by:

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

Random Forests

The idea is to build large collection of de-correlated trees, and then average them.

- For $b = 1$ to B:
    - Draw a bootstrap sample Z^* of size N from the available training data.
    - Grow tree T_b. Repeat recursively for each terminal node until minimum node size is reached.
        - Select m variables from p.
        - Pick the best variable among m.
        - Split the node.
- Output the ensemble of trees $\{T_b\}_1^B$.
- Prediction:
    - Regression: $\hat{f}_{\mathrm{rf}}^B(x) = \frac{1}{B}\sum_{b=1}^B T_b(x)$.
    - Classification: $\hat{C}_{\mathrm{rf}}^B(x) = \mathrm{mode}\{\hat{C}_b(x)\}_1^B$.

Committee learning

- Some times referred as ensemble learning.
- The idea is to combine a number of weak (accuracy is slightly larger than of random guessing) classifiers into a powerful committee.
- Motivation is to improve estimate by reducing variance and sometimes bias.

Boosting

- The final prediction is given by:

$$G(x) = \operatorname{sign}\Big(\sum_{m=1}^{M} \alpha_m G_m(x)\Big).$$

  which is weighted majority vote of classifiers $G_m(x)$. Here α_m are weights describing contribution of each classifier.

- While on the first view result is very similar to the bagging, there are some major differences.
- Two class problem where output variable coded as $Y \in \{-1, 1\}$.
- For the classifier $G(X)$ error rate is given by:

$$\overline{\mathrm{err}} = \frac{1}{N}\sum_{i=1}^{N} I(y_i \neq G(x_i)),$$

  where N is the power of training data set.

Ada Boost

AdaBoost.M1. by Freund and Shcapire (1997).

- Initialize observation weights $w_i = 1/N$, $i = 1, \ldots, N$.
- For $m = 1$ to M:
  - Fit weak classifier G_m that minimizes the weighted sum error for misclassified points.

$$\epsilon_m = \frac{\sum_{i=1}^{N} w_i I(G_m(x_i) \neq y_i)}{\sum_{i=1}^{N} w_i}$$

  - Compute $\alpha_m = \log((1 - \epsilon_m)/\epsilon_m)$.
  - Update weights w_i as

$$w_i = w_i * \exp(\alpha_m * I(y_i \neq G_m(x_i))), \quad i = 1, \ldots, N.$$

- Output classifier:

$$G(x) = \operatorname{sign}\Big(\sum_{m=1}^{M} \alpha_m G_m(x)\Big).$$