Exercises:
Classifier Combining, Bagging & Boosting

Jesse Krijthe

March 2018

1 Combining, Bagging & Random Forests

Learning Objectives

After this week's lecture, reading and exercises, you will be able to

- explain Condorcet's jury theorem
- explain the different between trained and untrained combiners
- list several ways to combine the results of models and apply them.
- explain the difference between bootstrapping and random subspaces
- implement a random forest classifier when given a decision tree implementation
- give one way of determining variable importance for a random forest
- effectively apply a random forest classifier to a dataset and interpret its parameters and results
- give an explanation why a random forest might perform better than a single decision tree

Relevant Literature

- Hastie, T., Tibshirani, R., & Friedman, J. H. (2009). The Elements of Statistical Learning (2nd ed.). Springer. Chapter 15
- Kuncheva, L. I. (2004). Combining Pattern Classifers. Methods and Algorithms. Wiley, Chichester. Chapter 4 & 5.

- Ho, T. K. (1998). The random subspace method for constructing decision forests. IEEE transactions on pattern analysis and machine intelligence, 20(8), 832-844.
- Breiman, L. (2001). Random forests. Machine learning, 45(1), 5-32.

Exercises

1. Suppose you are given an x-ray image of a patient, and you are given the task of choosing between the following three ways of making a diagnosis (yes vs. no bronchitis):

   (a) An expert radiologist who has a probability of making a correct diagnosis with probability ($p = 0.85$

   (b) A group of 3 doctors, each of which has $p = 0.8$

   (c) A group of 21 medical students, each of which has $p = 0.6$

   Please answer the following questions:

   (a) In the second case, what is the probability that all three doctors give the correct answer? What is the probability that at least 2 doctors make the right call? Combining these results, what is the probability that this group makes the right decision based on majority voting?

   (b) Can you come up with a general formula to calculate the probability that c doctors with competence p make the correct decision by majority voting? Use it to calculate the probability of a correct decision for the group of medical students.

   (c) Code a simulation to check your answer to the previous question, or, if you did not find a formula, use this simulation to answer the previous question.

   (d) Make a graph of the probability of a correct decision for various sizes of the jury and different competence levels (p) of the individual doctors.

   (e) Who has the highest chance to make the correct decision: the radiologist, the group of doctors or the group of students? How big does the group of medical students need to be to make the probability of a correct decision (almost) equal to the prediction of the group of doctors?

2. Table 1 shows the posterior probability estimates, $p_c(\omega|x)$, of three different classifiers, for two different classes $\omega \in \{\text{A}, \text{B}\}$.

   (a) Complete the table by filling in the values for the calculating the values produced by the different combiners and indicate which decision each combiner would make based on these values.

$p_1(\omega\|x)$		$p_2(\omega\|x)$		$p_3(\omega\|x)$		Mean		Max		Min		Prod	
A	B	A	B	A	B	A	B	A	B	A	B	A	B
0.9	0.1	0.9	0.1	0.0	1.0								
0.9	0.1	0.9	0.1	0.3	0.7								
0.9	0.1	0.2	0.8	0.1	0.9								
0.0	1.0	0.0	1.0	0.0	1.0								

Table 1: Posterior predictions of three classifiers for a classification problem with two classes (A, B), for 4 objects. Complete the table by calculating the values assigned by the different combiners and determine what decision each combiner makes for each object.

3. When taking a bootstrap sample, typically, some observations from the dataset will be present twice, while others will not be present. In bagging, these left-out samples (called out-of-bag examples) are useful, because they will not be used in the construction of the classifier/decision tree, so they can be used to evaluate its performance. What percentage of the observations from the original sample of size N will not be present in the bootstrap sample? Derive a formula, or, alternately, code a simulation and plot its result for various sample sizes N. How does this percentage change for different sample sizes N? What percentage does it converge to?

4. In your own words, explain the difference between bootstrapping and random subspaces.

5. To explain the performance of a random forest model, people often turn to measures of variable importance to determine which variables are most important in obtaining the given result. Find out, in the literature, or, for instance, in the documentation of a random forest implementation, how this importance are typically calculated for random forests. In a few sentences, explain and critique this approach.

6. Find out what widely used random forest implementations are available in your favourite programming language and apply the method to a prediction problem you find interesting (see, for instance the UCI Machine Learning repository for interesting datasets). Write a short description (min. 100 words) of your findings, including what dataset and implementation you used, how you set up your experiment, what the effect of different parameter settings was, what the performance was, which variables were important, etc.).

2 Boosting

Learning Objectives

After this week's lecture, reading and exercises, you will be able to

- give an informal definition of a weak learner
- list and explain the steps in the AdaBoost algorithm
- list and explain the steps in the GradientBoost algorithm
- explain how boosting is related to empirical risk minimization
- describe the difference between bagging and boosting
- give one way of determining variable importance for boosting models
- effectively apply gradient boosting to a dataset and explain the meaning and effect of its parameters

Relevant Literature

- Hastie, T., Tibshirani, R., & Friedman, J. H. (2009). The Elements of Statistical Learning (2nd ed.). Springer. Chapter 10
- Mohri, M., Rostamizadeh, A., & Talwalkar, A. (2012). Foundations of Machine Learning. The MIT Press. Chapter 6
- Chen, T., & Guestrin, C. (2016, August). Xgboost: A scalable tree boosting system. In Proceedings of the 22nd acm sigkdd international conference on knowledge discovery and data mining (pp. 785-794). ACM.
- Friedman, J. H. (2001). Greedy function approximation: a gradient boosting machine. Annals of statistics, 1189-1232.

Exercises

1. In the AdaBoost.M1 algorithm, in each step a base-learner is added to the ensemble with a weight that depends on its (weighted) error. Derive this weight of the added tree, Equation 10.12 on page 344 of The Elements of Statistical Learning (Exercise 1, Chapter 10).
2. Plot the weight given to a base-learner in the AdaBoost algorithm for different values of the error the base-learner makes. Explain what you see. What does it mean for these weights if we assume the base-learners are weak-learners? What happens to the weights if the probability of error of the base-learner is > 0.5 and why?

3. In your own words, explain some of the main differences between bagging and boosting.

4. How does AdaBoost relate to gradient boosting?

5. Similar to what you did for the random forest, find out what widely used gradient boosting *or* AdaBoost implementations are available in your favourite programming language and apply the method to a prediction problem you find interesting (see, for instance the UCI Machine Learning repository for interesting datasets). Write a short description (min. 100 words) of your findings, including what dataset and implementation you used, how you set up your experiment, what the effect of different parameter settings was, what the performance was, which variables where important, etc.).