Course: Machine learning

By: Pavlo Mozharovskyi

Tutorial to Lecture 2: Classification tree, bagging, and random forest

Task 1: Grow and visualize a classification tree.

For distributional setting 1 from Table 1 in Appendix (training sample size = 100) grow the classification tree, interpret its console output, picture it in a user-friendly way (use functions `rpart`, `plot`, `text` from `R`-package `rpart`).

Task 2: Explore influence of the minimum size of a splittable node in the classification tree.

For distributional setting 1 from Table 1 in Appendix, by means of a simulation study, explore performance of the classification tree depending on the minimum size of a splittable node; restrict to the equally spaced grid of 10 values between 0 and size of a class. Simulate training (training sample size = 100, 200, 500) and test (test sample size = 1000) data 100 times, grow trees on the training data and calculate their error on the test data. Represent the results in form of a composite boxplot (use function `boxplot`) chart (each for one sample size) over the 100 error rates. Include the optimal (linear discriminant analysis, LDA, use function `lda` from `R`-package `MASS`) classifier and the tree completely separating the training sample into your analysis. Interpret the results. Repeat for another distributional setting of your choice.

Task 3: Explore performance of the bagged LDA and classification tree.

Make a single draw from distributional setting 2 from Table 1 in Appendix (training sample size = 200, test sample size = 10000). Plot both training and test data.

1. Explore performance of the bagged LDA.

   Program the bagged LDA. Check its performance by training it on the training set and calculating its error on the test set once for a different number of base classifiers, *e.g.* for an equally spaced grid of twenty values between 0 and 500. Plot the error depending on the number of base classifiers. Interpret the graph.

2. Explore performance of the bagged classification tree.

   Repeat the experiment from above with the bagged tree (use function `bagging` from `R`-package `ipred`) and 50 equally spaced numbers of base classifiers fixing minimum size of a splittable node to 1 and to 10. Interpret the results.

Task 4: Explore the random forest classifier on `spam` data (use `R`-package `kernlab`, should contain 4601 57-dimensional observations).

Divide `spam` data set in a data independent way into a training set (2300 observations) and test set (2301 observations).

1. Explore influence of the number of randomly chosen variables at each node on the performance of the random forest.

   Using these training and test sets, for a random forest (use function `randomForest` from R-package `randomForest`) consisting of 1,2,...,50 trees, plot its classification error dependent on the number of trees when choosing randomly 1, 7 (default), 35 variables at each node, on the same plot. Interpret the graphs.

2. Compare error estimates.

   For the random forest choosing 1 (then 7 and 35) variable at each node, plot classification error for a random forest consisting of 1,2,...,50 trees dependent on the number of trees measured:

   - on the test set,
   - as the out-of-bag error,
   - sequence of the out-of-bag errors of the random forest containing 50 trees.

Task 5: Compare performance of classifiers on `pima` data (use R-package `MASS`). Consider the following classification algorithms:

- Linear discriminant analysis (use R-function `lda`).
- Quadratic discriminant analysis (use R-function `qda`).
- Robustified equal-prior quadratic discriminant analysis defined as:

  $$g(\boldsymbol{x}) = \underset{i\in\{0,1\}}{\arg\min}(\boldsymbol{x} - \bar{\boldsymbol{x}}_i)^T \boldsymbol{S}_i^{-1}(\boldsymbol{x} - \bar{\boldsymbol{x}}_i)$$

  with $\bar{\boldsymbol{x}}_i$ and $\boldsymbol{S}_i$ being robust mean and covariance estimates for class “i” (use R-function `covMcd` from R-package `robustbase`).

- k-nearest neighbors classifier (choose the number k of nearest neighbors by leave-one-out cross-validation, restrict k for better speed; use R-functions `knn` and `knn.cv` from R-package `class`).
- Random forest classifier (use `randomForest` from R-package `randomForest`, let the number of variables chosen at each node be default = 2).

Create the complete `pima` data set by merging data sets `Pima.tr` and `Pima.te` (should contain 532 complete 7-dimensional observations). Check performance of classifiers by randomly dividing data into train (250 observations) and test (282 observations) sets 100 times. Plot the composite boxplot chart (use function `boxplot`). Interpret the results, conclude about the data set.

Appendix

10 bivariate distributional settings for testing a binary supervised classifier, taken from the article

- Li, J., Cuesta-Albertos, J. A., and Liu, R. Y. (2012). DD-classifier: Nonparametric classification procedure based on DD-plot. *Journal of the American Statistical Association*, **107**(498), 737–753.

Table 1: Distributional settings for the simulation study.

No.	Alternative	class “0”	class “1”
1	Normal location	$\mathrm{N}(\left[\begin{smallmatrix}0\\0\end{smallmatrix}\right], \left[\begin{smallmatrix}1&1\\1&4\end{smallmatrix}\right])$	$\mathrm{N}(\left[\begin{smallmatrix}1\\1\end{smallmatrix}\right], \left[\begin{smallmatrix}1&1\\1&4\end{smallmatrix}\right])$
2	Normal location-scale	$\mathrm{N}(\left[\begin{smallmatrix}0\\0\end{smallmatrix}\right], \left[\begin{smallmatrix}1&1\\1&4\end{smallmatrix}\right])$	$\mathrm{N}(\left[\begin{smallmatrix}1\\1\end{smallmatrix}\right], \left[\begin{smallmatrix}4&4\\4&16\end{smallmatrix}\right])$
3	Cauchy location	$\mathrm{Cauchy}(\left[\begin{smallmatrix}0\\0\end{smallmatrix}\right], \left[\begin{smallmatrix}1&1\\1&4\end{smallmatrix}\right])$	$\mathrm{Cauchy}(\left[\begin{smallmatrix}1\\1\end{smallmatrix}\right], \left[\begin{smallmatrix}1&1\\1&4\end{smallmatrix}\right])$
4	Cauchy location-scale	$\mathrm{Cauchy}(\left[\begin{smallmatrix}0\\0\end{smallmatrix}\right], \left[\begin{smallmatrix}1&1\\1&4\end{smallmatrix}\right])$	$\mathrm{Cauchy}(\left[\begin{smallmatrix}1\\1\end{smallmatrix}\right], \left[\begin{smallmatrix}4&4\\4&16\end{smallmatrix}\right])$
5	Normal contaminated location	Learning sample: 90% as No. 1, 10% from $\mathrm{N}(\left[\begin{smallmatrix}10\\10\end{smallmatrix}\right], \left[\begin{smallmatrix}1&1\\1&4\end{smallmatrix}\right])$. Testing sample: as No. 1	as No. 1
6	Normal contaminated location-scale	Learning sample: 90% as No. 2, 10% from $\mathrm{N}(\left[\begin{smallmatrix}10\\10\end{smallmatrix}\right], \left[\begin{smallmatrix}1&1\\1&4\end{smallmatrix}\right])$. Testing sample: as No. 2	as No. 2
7	Exponential location	$(\mathrm{Exp}(1), \mathrm{Exp}(1))$	$(\mathrm{Exp}(1)+1, \mathrm{Exp}(1)+1)$
8	Exponential location-scale	$(\mathrm{Exp}(1), \mathrm{Exp}(1/2))$	$(\mathrm{Exp}(1/2)+1, \mathrm{Exp}(1)+1)$
9	Asymmetric location	$(\mathrm{MixN}(0;1,2), \mathrm{MixN}(0;1,4))$	$(\mathrm{MixN}(1;1,2), \mathrm{MixN}(1;1,4))$
10	Normal-exponential	$\mathrm{N}(\left[\begin{smallmatrix}0\\0\end{smallmatrix}\right], \left[\begin{smallmatrix}1&0\\0&1\end{smallmatrix}\right])$	$(\mathrm{Exp}(1), \mathrm{Exp}(1))$

$$\text{with } \mathrm{MixN}(\mu, \sigma_1, \sigma_2) = \begin{cases} -\sigma_1 * |\mathrm{N}(0,1)| + \mu & \text{with probability } 1/2, \\ \sigma_2 * |\mathrm{N}(0,1)| + \mu & \text{with probability } 1/2. \end{cases}$$