Assignment 3

May 2020

Predicting protein location

The prediction of the cellular location of proteins based on protein characteristics is a typical bioinformatics invention from the 1990's. Proteins can be located in several areas or organelles of the cell, like cytoplasmic, mitochondrial, or lysosomal membranes, in the periplasm, or in the cytoplasm. The question was whether it was possible to predict that location from sequence characteristics of the proteins. In the early 1990's two data sets were created, one for the bacterium *E. coli* and one for Yeast (Nakai and Kanehisa, 1991; Nakai and Kanehisa, 1992), in which the location of hundreds of proteins was listed, together with a number of characteristics of these proteins (see the data (data/) folder). A paper published somewhat later (Horton and Nakai, 1997) also gave an improved method for the prediction of protein localization, using a k-nearest-neighbor (KNN) classifier. Although the number of characteristics (predictive variables) listed in the dataset is relatively low, and although this is therefore not a high-dimensional dataset, it is still a nice data set to compare modern methods to a method that was state of the art in the 1990's.

Assignment

Your task is to make a comparison of three methods, by reproducing results from two methods also used by Horton & Nakai (The pdf of the paper is provided in the file literature/Horton1997.pdf (literature/Horton1997.pdf)), namely *k-nearest neighbor* and *Naive Bayes* and a method that was not available at that time, the *Random Forest*. Horton and Nakai (1997) used stratified cross-validation to test the accuracy of the classifiers. This means that they randomly divided the data sets in equal partitions (Tables 3 and 4 in the paper), subject to the constraint that the *proportion of the classes in each partition was approximately equal* (this is what the adjective *stratified* indicates). Subsequently, they used each of the partitions as a test set on a classifier constructed from the training set consisting of the remaining partitions. The accuracy of the predictor was evaluated as the average accuracy from these cross validations.

Please note the following:

- When using these algorithms it is essential that the response variable has the `factor` object class.
- The success rate of the Naive Bayes classifier depends on the object classes of the different predictors, as should be clear from the explanation of this classifier in the syllabus Chapter 23.1 (http://www.few.vu.nl/~molenaar/courses/statR/syllabus/classifiers.html#naivebayes).

Requirements

- label your results with the **same numbers as below**, otherwise we will miss the fact that you have carried out a task, which would decrease your grade.
- Graphs and tables should have proper captions and/or legends.
- Hand in two documents, an Rmarkdown (*.Rmd) and a document (*.pdf or *.html) generated from it with **code displayed** (global chunk option: `echo=TRUE`).
- Use the **same number of stratified cross validations as in the original paper** for each of the data sets (4 for *E. coli* data, and 10 for Yeast data).
- You **must** use the function `partition()` in script/partition_function.R (script/partition_function.R) to create stratified data sets. *Tip: it can be read with a `source()` statement in your Rmarkdown file.*
- If performing any of the following analyses you **must** use the following functions/packages:
    - K-nearest neighbour classification: the `knn()` function from the `class` package (in the standard system library), with the **same values for k** (number of neighbours considered) as in the Horton

and Nakai (1997) paper.
- Naive Bayes classification: the `naiveBayes()` function from the `e1071` package. Use the default settings of this function.
- Random Forest classification: the `randomForest()` function from the `randomForest` package. Use the default settings of this function.

The use of similar functions from other packages is **not allowed**.

E. coli data

1. Read the data into R. Carefully investigate the data. You may want to use the `summary()` (or alternatively `skimr::skim()` which works nicely with Rmarkdown by producing tabular-formatted output) and `duplicated()` functions. Also make a table of the number of times a location is observed (as table 1 in the paper).
2. Remove from the dataset the locations that have been so scarcely observed that they would occur on average less than twice in a (stratified) training set. Why do we want to remove these?
3. Make a pairplot of the data. Usually `pairs()` works, but you could also use the fancy `ggpairs()` function from the `GGally` package. Give a brief account of your conclusions and their consequences (modifications that you possibly want to make) from the examination of the data and the pairplot.
4. Carry out KNN, Naive Bayes and Random Forest classification and cross validation of the data. Summarize the conclusions as in table 3 of the paper. *Since getting mean and standard deviation in the same table as the results from the individual partitions is rather complicated you can show the results in two tables.*
5. Perform suitable statistical tests to find out whether algorithms perform better than the others. Present their results in a table. What do you conclude from the tests?

Yeast data

Now perform the same analyses on the yeast data set:

6. Read the data into R. Carefully investigate the data. Make a table of the number of times a location is observed.
7. Make a pairplot of the data. Give a brief account of your conclusions and their consequences.
8. Carry out KNN, Naive Bayes and Random Forest classification and cross validation of the data. Summarize the conclusions.
9. Perform suitable statistical tests to find out whether algorithms perform better than others on these data. Present their results in a table and give a brief conclusion. Give an explanation if there is a marked difference between your results and the one from the paper.
10. Are the conditions for the naive Bayes classifier, namely conditional independence of all pairs of predictors, satisfied in general in the yeast data? Illustrate your conclusion with a graph.
11. Without actually carrying this out, is it possible to obtain probability distributions over the classes (cellular locations) from each of the algorithms? If so, please explain how.

References

Horton, P., and Nakai, K. (1997) Better prediction of protein cellular localization sites with the k nearest neighbors classifier. *Proc Int Conf Intell Syst Mol Biol* **5**: 147–152.

Nakai, K., and Kanehisa, M. (1991) Expert system for predicting protein localization sites in gram-negative bacteria. *Proteins* **11**: 95–110.

Nakai, K., and Kanehisa, M. (1992) A knowledge base for predicting protein localization sites in eukaryotic cells. *Genomics* **14**: 897–911.