The rminer package for regression

Gabriele Venturato

Abstract

The aim of this work is to have an insight into the *rminer* package for regression analysis. Starting from a brief theoretical introduction, towards the description of the main functions of the package, and concluding with a simple case study to show how the package can be used.

Introduction

Regression

Regression is the problem of learning a *functional relationship* between variables using a dataset where the specific functional form learned depends on the choice of the model (it can be linear or not). The parameters of the function are learned using the *explanatory variables (features)* into the training set, and then performance are evaluates testing the model on the test set. The aim of a regression model — as opposed to a classification model — is to perform a *numeric prediction* based on the features in input.

Linear Regression

Random Forest

The rminer package

The goal of this package is to facilitate the use of data mining algorithms for classification and regression. It offers a short and coherent set of functions in order to easily develop a project, letting the user to follow in particular three CRISP-DM stages: *data preparation*, *modeling* and *evaluation*.

The package can be installed and loaded with:

```
install.packages("rminer")
```

And loaded with:

```
library(rminer)
```

As usual, a complete list of all functions available can be found in the documentation of the package:

```
help(package=rminer)
```

For the purpose of this work instead of reporting what can be found easily — and with more details — inside the help, I preferred to report a brief list of the function organized by their purpose, in order to quickly move toward the practical example that is more useful to show the package capabilites.

First of all, for the data preparation phase, after having loaded the dataset, the first function that can be used are mainly:

- `delevels(x, levels, label = NULL)` – reduce or replace factor x with *levels*, with an optional new label;
- `imputation(imethod = "value", D, Attribute = NULL, Missing = NA, Value = 1)` – perform imputation to remove missing values from dataset D and from a specific attribute, with the value specified.

- `CaseSeries` – create a data.frame from a time series (vector) using a sliding window. This function is not used in this work and its behavior can be further analized in official documentation.

Case Study

The dataset

The model

Evaluation

Conclusion

References

Cortez, P. 2010. “Data Mining with Neural Networks and Support Vector Machines using the R/rminer Tool.” In *Advances in Data Mining – Applications and Theoretical Aspects, 10th Industrial Conference on Data Mining*, edited by P. Perner, 572–83. Berlin, Germany: LNAI 6171, Springer.

Trevor, Hastie, Tibshirani Robert, and Friedman JH. 2009. “The Elements of Statistical Learning: Data Mining, Inference, and Prediction.” New York, NY: Springer.

Witten, Ian H, Eibe Frank, Mark A Hall, and Christopher J Pal. 2016. *Data Mining: Practical Machine Learning Tools and Techniques*. Morgan Kaufmann.