The rminer package for regression

Gabriele Venturato

February 7, 2019

1. Regression
2. The rminer package
3. Case Study: Life Expectancy
4. Conclusions

Regression

Linear Regression

- If you're here you know it!

Classification and Regression Trees (CART)

- commonly used in data mining
- "growing the tree" recusively:
  - for each node, select a feature and perform a split that minimize

$$RSS = \sum_{left}(y_i - \bar{y_L})^2 + \sum_{right}(y_i - \bar{y_R})^2$$

  - $\bar{y_L}$ and $\bar{y_R}$ are the means of the left and right node respectively
  - stop when a region have less than k values in it (with $k \geq 1$)
- advantages over traditional statistical methods
  - they don't do any formal distribution assumption
  - they can automatically fit non-linear interactions
  - they handle missing values with surrogate variables

Random Forests

- "bag" of CARTs
- higher accuracy, more stable, less sensitive to overfitting, speed in learning, but slower in prediction
- built with the repetition of two phases:
  - take a bootstrap sample D_i from the data D
  - fit a classification or regression tree on D_i set
    - grow the tree only on m *randomly* chosen features (out of M)
- at the end combined — in case of regression — by averaging

The rminer package

Data Preparation

- `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.

Modeling

- `holdout(y, ratio = 2/3, mode = "stratified", ...)` – it computes indexes for holdout data split into training and test sets
- `fit(x, data = NULL, model = "default", task = "default", ...)` – it fits a supervised data mining model
- `crossvaldata(x, data, theta.fit, theta.predict, ngroup = 10, model, task, ...)` – compute k-fold cross-validation for models

Evaluation

After having fitted the model one can proceed with the evaluation in order to understand the goodness of the model and eventually fix it. Main functions here are:

- `mmetric(y, metric, ...)` – used to get the metrics specified in the parameter *metric* about the model *y*
- `mgraph(y, graph, ...)` – used to print graphs about model accuracy: "RSC" and "REC" are common options for regression
- `mining(x, data = NULL, Runs = 1, method = NULL, model = "default", task = "default", ...)` – it's a powerful function that trains and tests a particular fit model under several *runs* and a given validation *method*

Case Study: Life Expectancy

Case Study: Life Expectancy

(source code)

Conclusions

Conclusions

- rminer is a good tool to perform regression analysis
- small set of functions, but good variety of parameters and models
- maybe limiting for advanced users with specific requirements