

Lab 7: Data Mining.

Ensemble methods

1. Ensemble methods

Ensemble methods construct *meta-models* consisting of several base models (e.g. classifiers). The main motivation of combining several models to construct of ensemble is to reduce the prediction error by addressing one or both of its components: *bias* and *variance* (see Lecture 12 for the mathematics behind the split of error in the two components). The bias component is mainly caused by the limits of the model (e.g. a linear classifier is used for a nonlinearly separable problem) while the variance component is caused by the limited dataset used for training.

There are different ways of constructing ensemble methods:

- By generating different models based on one training dataset (e.g. *bucket of models*)
- By generating different models through a sampling process of the training dataset (e.g. *bagging* and *boosting*)
- By using different models and in the same time splitting the dataset (e.g. *stacking*)

The most popular ensemble methods are Random Tress (it relies on bagging = bootstrap aggregation) and AdaBoost (which relies on adaptive boosting). An important issue, which impacts on the model performance, is related to the *independence* between the models which are part of the ensemble.

Related R packages:

- For RandomForests: `randomForest` (2002), `Willows` (2009), `party` (2006), `randomForestSRC` (2015), `ranger` (2015) = Random Forests Generator

Exercise 1/Rattle, R. Use [Rattle](#) and [ranger](#) package to construct a Random Trees classifier to solve classification problems as those included in Lab 3: (a) iris; (b) breast cancer (Wisconsin); (c) Titanic.

Starting point for us of [ranger](#): [ExampleRanger.r](#)

Exercise 2/R. Analyze the functions included in the R package `SuperLearner`. Example: [ExampleSuperLearner.r](#)

Exercise 3/Weka. By using Weka Experimenter compare the performance of the following ensemble models (metamodels): [Vote](#), [Bagging](#), [Random Forest](#), [AdaBoost](#) and [Stacking](#) for the following datasets: [iris.arff](#), [glass.arff](#)

- a) Use the default values of the parameters
- b) Try to improve the behavior of [Vote](#), [Bagging](#) and [AdaBoost](#) by replacing the default individual Classifier with other models.

2. Text Mining

Text mining refers to extracting information from documents (interpreted as sequence of words). The main text mining tasks are classification and clustering of documents based on their content. The simplest approach for classification/clustering documents is based on the following steps:

- Pre-process the text by:
 - Removing the *stop words* (words which do not provide specific information being rather syntactic components used to link various parts of speech). Lists of stopwords corresponding to different languages can be found at <http://www.ranks.nl/stopwords>
 - Transform the words by *stemming* (i.e. reduces the inflected variants of words to their root form). The most popular stemming algorithm is that proposed by Porter (see <http://tartarus.org/martin/PorterStemmer/>). A web service for stemming in various languages is available at <http://text-processing.com/demo/stem/>
- Construct for each document a *frequency vector* containing quantitative measures of the presence of words belonging to a dictionary in each of the documents. If the dictionary contains N words then to each document in the collection of documents to be processed one have to associate a vector of N elements specifying the number of occurrences of the corresponding word in the document. Since words which are specific to only some documents have a higher discriminative power, instead of using frequencies of terms it is used the so-called *TF-IDF* (term frequency – inverse document frequency) encoding characterized by the fact that the frequency of a term in a given document is divided by the number of documents in the collection which contain that term. Once these numerical vectors are constructed then one can apply any classification/clustering technique.

Exercise 4/R. Analyze the functions in `tm` R package for text mining. Data set: `spamSMS.csv`. Starting point: [ExampleTextMining.r](#)

Exercise 5/Weka.

- a) Open the file `movieReviews.arff` (it contains reviews on movies grouped in two categories: positive and negative)
- b) Construct the dataset with the occurrence of terms in the collection of reviews by using `Filters->Unsupervised->Attribute->StringToWordVector`
- c) Apply a classifier (e.g. `SMO` – Support Vector Machine) to the dataset. Remark: it requires to set first the attribute called `@@class@@` as class attribute (using `Edit`, right click on `@@class@@` and selecting `Attribute as class`)
- d) Analyze the impact of using a stemming step on the quality of the classification.