LINGI2262: Assignment 5
A Machine Learning Competition

Group 8: Maxime Dimidschstein, Doriane Olewicki

May 10, 2018

1 Data cleaning

The first step towards our goal is to clean the data set. Indeed, there are too many features to handle, and a lot of them can be considered useless.

The first strategy we used consists in computing the variance of every feature and only keeping the 1000 highest ones. We then decided to refine this method and improve it by not taking the outliers into account. To do that, for each column of the data-frame, we first get the mean, μ, and the initial standard-deviation, σ_i. Then, we re-compute the standard-deviation, σ_r, without taking into account values that are outside of the interval $\mu \pm \sigma_i$.

This method could not be applied to some of the last columns of the set, as they are not numeric. We decided to keep them by default, except for two of them which seemed uninteresting because their values had little to no variation.

2 Prediction and Performance

For the prediction step we considered different strategies and compared the results before deciding which one seemed to be the most suitable. The general approach we used is similar for all those strategies.
We split the training set, Tr, in two parts by random sampling: a training part, Tr_A, and a test part, Tr_B. The model is created on Tr_A, and the balanced classification rate is computed on the prediction of Tr_B.
We do this for each iteration, while also predicting on the test set, Te. Finally, the prediction on the test set is decided by taking the feature with the highest value for each entry. Then, we perform a weighted sum on those test set predictions, using the associated BCR values as weights.
The final BCR is computed by predicting Tr_B the same way.

First, we used `rpart` to perform recursive partitioning.
We added some pre- and post-pruning. The parameters used are `minsplit`$= 10$ and `cp`$= 0.01$.
The results we obtained varied in the range [50% ; 60%] and are not very stable.

Then, we implemented a bagging strategy.
The approach used here is a bit different. We also split Tr in Tr_A and Tr_B, but instead of using the whole Tr_A at once and resampling Tr several times, we perform a few iterations with the same splitting. For each of those iterations, we draw a bootstrap sample from Tr_A and perform the training on it.
We used a `maxdepth` of 5.
The results were quite similar to `rpart`.

We attempted to improve the results of the bagging by building them iteratively.
The same approach is used, but performed several times, with different splits of Tr. Then, we do a weighted sum of those results as usual, and compute its BCR. We believe this approach is better than simply performing the bagging several times and keeping the prediction that has the highest BCR. Indeed, having a high BCR for one bagging depends on our "luck", and on how favorable the training sample Tr_A is for the test sample Tr_B. Meanwhile, performing multiple iterations and using them all is a good way to ensure a good result.
The results stay between [55% ; 60%] and are quite stable.

We also tried using a support vector machine. We performed several tests to tune the parameters to get the best results. Regarding the parameters, we opted for a polynomial kernel with degree 2, a `gamma` ten times higher than the default one, and `coef0` equal to 10.
The results are in the range [58% ; 63%].

In the end, we decided to use the SVM because it gives the best and most stable results.