Homework - Ensemble methods

Introduction
The goal of this homework is to get acquainted with some ensemble methods like bagging, boosting and random forests. For testing we use the dataset about handwritten digits. In order to make the implementation easier we have stripped it down to only 2 digits: 2 and 3. The datasets are in file Ensemble.RData and can be read into R using command
print(load("Ensemble.RData")).
There are 2 datasets test and train, both with the same structure, first column indicates the class and all other columns are the features.
In all the models we use classification trees that in R are implemented in package rpart. For building one tree on training data, predicting values on test data and calculating misclassification rate, the code looks roughly like that:

```r
m = rpart(y ~ ., data = train, method = "class")
p = predict(m, newdata = test, type = "vector") * 2 - 3
```

```r
t = table(test$y, p); t
1 - sum(diag(t))/sum(t)
```

Bagging
Bagging is a method where an ensemble of classifiers is trained on bootstrap samples of training data. To generate a bootstrap sample of a dataset with \( n \) rows one has to sample with replacement \( n \) rows from training data. To predict a response, a prediction is made with all members of the ensemble and the result with most "votes" is reported.

**Exercise 1 (3 points)**
Train a bagging classifier on training data using 50 bootstrap samples. Measure its misclassification rate.

**Exercise 2 (1 point)**
Draw a figure that shows change in misclassification rate of bagging classifier if we add samples. (x-axis: number of datasets 1-50; y-axis: misclassification rate)

Random Forest
Bagging estimate can be improved, if we decrease the correlation between the trees in ensemble. One way to do it, is to use random subsets of data when constructing trees.

**Exercise 3 (1 point)**
Train a simple random forest type classifier, where each tree is constructed using only a subset of \( m = 50 \) variables. Again use 50 bootstrap samples. Measure its misclassification rate.
Excercise 4 (2 points)
Try values 10, 50 and 300 for \( m \) and draw similar figure as in Excercise 2, but now with all the random forest and bagging curves. What can be said about the results?

Boosting

Boosting works a bit differently that bagging. Instead of taking samples of the data, the training dataset is re-weighted every step, to concentrate efforts on the points that were misclassified. Adaboost algorithm is given by:

**Algorithm 10.1** AdaBoost.M1.

1. Initialize the observation weights \( w_i = 1/N, \ i = 1, 2, \ldots, N. \)
2. For \( m = 1 \) to \( M \):
   
   (a) Fit a classifier \( G_m(x) \) to the training data using weights \( w_i \).
   
   (b) Compute
   
   \[
   \text{err}_m = \frac{\sum_{i=1}^{N} w_i I(y_i \neq G_m(x_i))}{\sum_{i=1}^{N} w_i}.
   \]
   
   (c) Compute \( \alpha_m = \log((1 - \text{err}_m)/\text{err}_m) \).
   
   (d) Set \( w_i \leftarrow w_i \cdot \exp[\alpha_m \cdot I(y_i \neq G_m(x_i))] \), \( i = 1, 2, \ldots, N \).
3. Output \( G(x) = \text{sign} \left[ \sum_{m=1}^{M} \alpha_m G_m(x) \right] \).

As a classifier one can still use trees. To do weighted fitting of the tree one can use the *weight* parameter in *rpart*.

Exercise 5 (3 points)
Train Adaboost classifier on our example doing 50 steps. Measure its misclassification rate.

Exercise 6 (1 point)
The default tree parameters might provide too rich model for boosting and therefore induce overfitting. To avoid this, limit the tree depth to 2 (can be set as *control* = list(maxdepth = 2) in *rpart*) and train again the classifier for 50 steps. Measure its misclassification rate.

Exercise 7 (1 point)
Combine the curves for all classifiers to one figure as in Exercises 2 and 4. Add also the single tree error to the picture.
Exercise 8 (1 point)
Interpret the results from previous figure. What methods perform the best. How does changing the model complexity affect the ensemble performance in bagging and in boosting?