Solutions to Practical 5

All boosting techniques use the “logit” method in this solution.

1) Find the best suited Weak classifier (WC) for each of the three distribution types in the Standard Dataset Generator: Checkerboard, concentric circles and Swiss roll (keep the number of swirls/circles/grid count to 2).
   a. Checkerboard

Decision stumps cannot classify this kind of distribution because one classifier always classifies 50% of the points well and 50% badly (whatever the horizontal or vertical line). Boosting of decision stumps does not work here.

Random rectangles work quite well although there are several errors (due to the limited amount of generated rectangles), random circles work actually better than rectangles. (2 are enough)

Random Gaussians perform worse than circles since circles fit already well the points while there are more parameters to set for a Gaussian so random Gaussians are more sparsely generated (across their parameter space) and fewer “good” ones are generated close to the solution.

   b. Concentric circles

Obviously, circles are fit for this distribution. Three circles are enough to classify this data correctly. Gaussians work as well.

   Random Gaussians:
c. Swiss Roll

Through empirical testing, we find that seven random Gaussians are the minimum it takes to classify correctly all the points. Using other random figures do not lead to good results without a very high number of WC.
2) Test the different weak classifiers available for boosting and compare their performance on the provided datasets.

Tests with “logit” boost method, 50% train/test ratio, 10 folds crossvalidation:

a) House-votes (first 2 dimensions after PCA)

SVM (with 10 Support vectors) brings the best performance for low number of weak classifiers: one SVM is considered a Weak Classifier, however, its complexity is much higher than the complexity of the other WC (a line or a circle for instance). The comparison is thus unfair. Apart from rectangles that do not function properly, and SVM for the reason cited previously (one SVM already has enough complexity to classify the points, there is no need to increase the number of SVMs), all other methods increase their performance with the number of classifiers.
b) Ionosphere

This classification is done on the original data, without PCA projection.

There is a strong increase in performance for Stumps, GMM and Circles along with the number of weak classifiers.

When comparing with the training curves, you can see that Random Projections suffer heavily from overfitting since in training it almost reaches perfect classification while in testing the performance does not improve with more classifiers. Besides SVM and random rectangles, almost all methods do a perfect training classification. Simple methods like decision stumps or methods that take into account the local structure of the data like GMM are better at generalizing and therefore have better results in testing than the other methods.
3) Compare the performance of boosting of SVMs and boosting of Gaussians with classic SVM and GMM; which perform better and what are the advantages/drawbacks of each?
   a. GMM
      i. Ionosphere

In training, boosting is clearly advantaged by its iterative structure that aims at adding Gaussians to correct for misclassified datapoints. In testing, the two methods perform similarly.

ii. House-votes (first 2 dimensions after PCA)

Both methods classify perfectly the training points with a sufficient number of Gaussians; however, classic GMM overfits the data above 5 Gaussians and thus the testing performance drops.
b. SVM
   i. House-votes (first 2 PCA dimensions)

C-SVM hyperparameters were manually optimized: with only one support vector machine (and one WC), classic C-SVM outperforms Boosting of SVMs both in training and testing. Boosting of 10 Support Vectors SVM and 20 Support Vectors SVM yield approximately the same results and increasing the number of WC does not improve the testing performance. This can be understood because of the overlapping of points in this dataset: once the base structure of the datapoints is fitted by the model, adding complexity only leads to overfitting the training set and does not improve the performance on the testing set.

![Graph: SVM vs. boosting of SVMs on House-votes (training and 50%T/T ratio testing tests)]
4) Sensitivity analysis  
   a) SVM (House votes, PCA dimensions 1 and 2)  

A positive correlation between the kernel width and the Cost parameter C can be found in order to minimize the error with a 66% training/testing ratio. This can be understood because big kernel widths (compared to the data size) may require support vectors with big alphas (so big C’s) so that their influence can be modulated when the separating line needs to vary locally. See second picture: With C=1, a lot of support vectors have their alphas saturated (in white), whereas this is not the case with C=20.

However, the error does not vary much with these parameters (Between 0.05 and 0.11 in the worst case in this range), the algorithm is thus not very sensitive to these hyperparameters.
b) GMM

In GMM, one must choose the number of Gaussians, their covariance matrix type, and the initialization method.

K-means initialization: One can immediately notice that GMM is pretty sensitive to its hyperparameters, and that the effect seems much less smooth than with SVM: there is a big difference between 3 and 4 Gaussians (with full and spherical covariance matrices), then the error increases a little and goes down again for 8 Gaussians (Full cov. matrix). Indeed, the local minimum found by GMM depends heavily on the number of Gaussians available. Sometimes GMMs with more Gaussians find less efficient fits than with fewer Gaussians. This is illustrated in the next picture.