1 AdaBoost (10 points)

Recall: In AbaBoost we learn a model of the form

$$\hat{y}_*(x) = \sum_{c=1}^{C} \alpha_c \hat{y}_c(x)$$

where all the component models belong to the same class $\hat{y}_c(x) = \hat{y}(x, \theta_c)$ (i.e. all are SVMs or Decision Trees, but not mixed). The component model parameters θ_c are learned by minimizing a weighted miss-classification rate

$$\theta_c = \operatorname*{argmin}_{\theta} \sum_n w_n \delta(y_n \neq \hat{y}(x_n, \theta))$$

and the component model weights are learned by minimizing the exponential loss

$$\alpha_c = \operatorname*{argmin}_{\alpha} \sum_{n=1}^{N} w_n e^{-\alpha y_n \hat{y}_c(x_n)} = \ldots = \log\left(\frac{1-\mathrm{err}_c}{\mathrm{err}_c}\right)$$

and the weights w_n are updated after learning a component model. The prediction of the joint model is given by $\mathrm{sign}(\hat{y}_*(x))$. In pseudo code:

Algorithm 1: AdaBoost

input : Dataset $\mathcal{D} = (X, y)$, weak learner $\hat{y}$, number of component models C
output : Learned component model parameters θ_c and component weights α_c
init : $w_n = \frac{1}{N}$
for $c = 1 \ldots C$ **do**
- $\theta_c = \operatorname*{argmin}_{\theta} \sum_n w_n \delta(y_n \neq \hat{y}(x_n, \theta))$ // optimal component parameters
- $\mathrm{err}_c = \sum_n w_n \delta(y_n \neq \hat{y}(x_n, \theta_c))$
- $\alpha_c = \log\left(\frac{1-\mathrm{err}_c}{\mathrm{err}_c}\right)$ // compute component weights
- $w_n = w_n e^{\alpha_c \delta(y_n \neq \hat{y}(x_n, \theta_c))}, \quad n = 1 \ldots N$ // update the weights
- $w = w / (\sum_n w_n)$ // normalize the weights

end
return : (α, θ)

Consider the following dataset consisting of 4 training samples followed by 3 test samples:

Train data

x_1	x_2	x_3	y
1	-1	-1	1
-1	-1	1	-1
-1	1	-1	1
-1	1	-1	-1

Test Data

x_1	x_2	x_3
1	-1	-1
-1	-1	-1
1	-1	1

A. [?p] Perform three rounds of AdaBoost learning on this data in order to predict the test labels. Use Decision-Stumps (one-level decision trees) as the underlying *weak* predictive model, assuming that each stump minimizes error as much as possible on the training set.

B. [?p] After running the three iterations, provide your final predictions and comment on the boosted model as compared to one of the decision trees.

2 Gradient Boosting with XGBoost (10 points)

Learn a Gradient Boosted Decision Tree model for two stumps with $\lambda = \gamma = 0.5$.
You can have a look at the slides here:
www.ismll.uni-hildesheim.de/lehre/ba-18w/script/4_predictive-analytics-xgboost.pdf