Problem 0.
Consider the problem of binary classification, with response variable $y \in \{-1, 1\}$, and exponential loss $\ell(y, f(x)) = \exp(-yf(x))$.

(i) The AdaBoost algorithm minimises the empirical (exponential) loss in a stage-wise manner. Put $f^{(m-1)}(x) = \sum_{k=1}^{m-1} \beta_k f_k(x)$, where f_k is the k-th weak learner, and β_k a scaling factor. At iteration m, we solve

$$(\beta_m\, f_m) = \arg\min_{(\beta, f)} \sum_{i=1}^{n} \exp\left\{-y_i(f^{(m-1)}(x_i) + \beta f(x_i))\right\} .$$

Show that f_m can be taken as a tree minimising a weighed error rate.

(ii) Derive the optimal solution β_m.

(iii) Deduce from *(i)* and *(ii)* the AdaBoost algorithm.

Problem 1. *Boosted tree model*
The boosted tree model

$$f^{(M)}(x) = \sum_{m=1}^{M} T(x; \Theta_m)$$

is a sum of trees, where Θ_m parametrizes the split variables, split points and predictions. To estimate the Θ_m, a forward stagewise procedure is used, and at each iteration one must solve

$$\widehat{\Theta}_m = \arg\min_{\Theta_m} \sum_{i=1}^{n} \ell(y_i,\, f^{(m-1)}(x_i) + T(x_i; \Theta_m)) .$$

Given the regions R_{jm}, finding the optimal constants γ_{jm} in each region is typically straightforward:

$$\hat{\gamma}_{jm} = \arg\min_{\gamma_{jm}} \sum_{x_i \in R_{jm}} \ell(y_i,\, f^{(m-1)}(x_i) + \gamma_{jm}) .$$

Show that for an exponential loss, the solution to the above problem is the weighted log-odds in each corresponding region

$$\hat{\gamma}_{jm} = \frac{1}{2} \log \left(\frac{\sum_{x_i \in R_{jm}} w_i^{(m)} \mathbf{1}(y_i = 1)}{\sum_{x_i \in R_{jm}} w_i^{(m)} \mathbf{1}(y_i = -1)} \right) .$$

Remark: This differs from the classifiers built in the AdaBoost procedure, where the $\gamma_{jm} \in \{-1, 1\}$.

Problem 2. *Performance bound*
We prove that if the weak classifiers perform substantially better than a random guess, then the AdaBoost algorithm can return a proportion of the training data misclassified arbitrary small. We assume here that at each iteration of the algorithm, the weights are renormalised by a constant Z_m to remain a probability distribution,

$$w_i^{(m+1)} = \frac{w_i^{(m)} e^{-t_i \beta_m G_m(x_i)}}{Z_m} \quad \text{for } m = 1, \ldots, M, \quad \text{so that} \quad \sum_{i=1}^{n} w_i^{(m+1)} = 1\,.$$

The algorithm is initialised with $w_i^{(1)} = 1/n$ for all $i = 1, \ldots, n$. The error term derived during the lectures thus becomes $err^{(m)} = \sum_{i=1}^{n} w_i^{(m)} \mathbf{1}(y_i \neq f_m(x_i))$. The individual contribution of each classifier is $\beta_m = \frac{1}{2}\log((1 - err^{(m)})/err^{(m)})$.

(i) Show that

$$w_i^{(M+1)} = \frac{e^{-y_i f^{(M)}(x_i)}}{n \prod_{m=1}^{M} Z_m}\,,$$

where $f^{(M)}(x_i) = \sum_{m=1}^{M} \beta_m f_m(x_i)$.

(ii) Derive the upper bound

$$\frac{1}{n}\sum_{i=1}^{n} \mathbf{1}(y_i \neq f^{(M)}(x_i)) \leq \prod_{m=1}^{M} Z_m = \prod_{m=1}^{M} 2\sqrt{err^{(m)}(1 - err^{(m)})}\,.$$

(iii) Conclude that provided $err^{(m)} = 1/2 - \gamma_m$, where $\gamma_m \geq \gamma$ for all m, then

$$\frac{1}{n}\sum_{i=1}^{n} \mathbf{1}(y_i \neq f^{(M)}(x_i)) \leq \exp\left(-2\gamma^2 M\right)\,.$$

Discuss this result.

Problem 3. *Boosting for K-class classification.*
We explore two ways to generalise the *AdaBoost* algorithm to K-class classification problems. In Part I, we consider a generalisation of the exponential loss function, allowing us to derive a similar looking algorithm. In Part II, we consider the deviance as a loss function, and derive a gradient tree boosting algorithm.

Suppose that the K classes are $\mathcal{C}_1, \ldots, \mathcal{C}_K$, and let $\mathcal{C}$ denote a generic class. Our data consists of n observations (x_i, y_i), for $i = 1, \ldots, n$, where each input vector x_i belongs to one of the K classes. The coding used for target variable y_i to denote the class appartenance is specific to the loss function considered.

Part I
Consider the following coding for the i-th target vector $y_i = (y_{i1}, \ldots, y_{iK})^t$,

$$y_{ik} = \begin{cases} 1 & \text{if } x_i \in \mathcal{C}_k \\ -\frac{1}{K-1} & \text{otherwise}\,. \end{cases}$$

Let $f(x) = (f_1(x), \ldots, f_K(x))^t$ with $\sum_{k=1}^K f_k(x) = 0$, and define

$$\ell(y,\, f(x)) = \exp\left(-\frac{1}{K} y^t f(x)\right)\,,$$

where y is a realisation of a random vector $Y = (Y_1, \ldots, Y_K)^t$.

(i) Explain in a few words why it is a good idea to consider the population minimiser $f^*(x)$ of $\mathbf{E}(\,\ell(Y,\, f(X))|X = x)$ as a classifier.

(ii) Using Lagrange multipliers, show that $f^*(x)$ subject to the zero-sum constraint is given by

$$f_k^*(x) = (K-1)\left\{\log \mathbf{P}(\mathcal{C} = \mathcal{C}_k|x) - \frac{1}{K}\sum_{j=1}^K \log \mathbf{P}(\mathcal{C} = \mathcal{C}_j|x)\right\}\,,$$

where $f^*(x) = (f_1^*(x), \ldots, f_K^*(x))$.

(iii) Deduce from (ii) an expression for the class probabilities $\mathbf{P}(\mathcal{C} = \mathcal{C}_k|x)$ in terms of the $f_k^*(x)$.

(iv) Show that a multiclass boosting using this loss function leads to a reweighting algorithm similar to *AdaBoost*. Present your final answer in the form of an algorithm, and explain precisely each step in your derivation.

Part II
Consider now the K-class classification problem where the targets y_{ik} are coded as 1 if x_i is in class k, and zero otherwise. Put $p_k(x) = \mathbf{P}(\mathcal{C} = \mathcal{C}_k|x)$. We make use of the representation

$$p_k(x) = \frac{e^{f_k(x)}}{\sum_{l=1}^K e^{f_l(x)}}\,, \quad k = 1, \ldots, K\,, \tag{1}$$

under the constraint $\sum_{j=1}^{K} f_j(x) = 0$.

Suppose that we have a current model for the $f_k(x)$, $k = 1, \ldots, K$. We wish to update the model for observations in a region R in predictor space, by adding constants $f_k(x) + \gamma_k$, with $\gamma_K = 0$.

(v) Write down the multinomial log-likelihood for (x_i, y_i), $i = 1, \ldots, n$, in terms of $f_k(x)$, $k = 1, \ldots, K$.

(vi) We wish to find the constants γ_k which maximise the log-likelihood in region R. An analytical expression is not available. Using only the diagonal of the Hessian matrix of the log-likelihood, and starting from $\gamma_k = 0$ for all k, show that a one-step approximate Newton-Raphson update for γ_k is

$$\gamma_k^+ = \frac{\sum_{x_i \in R}(y_{ik} - p_k(x_i))}{\sum_{x_i \in R} p_k(x_i)(1 - p_k(x_i))}, \quad \text{for} \quad k = 1, \ldots, K-1\,.$$

(vii) We prefer our update to sum to zero, as the current model does. Using symmetry arguments, show that

$$\hat{\gamma}_k = \frac{K-1}{K}\left\{\gamma_k^+ - \frac{1}{K}\sum_{l=1}^{K}\gamma_l^+\right\},$$

is an appropriate update, where γ_k^+ is defined as in in question (v) for all $k = 1, \ldots, K$.

(viii) Adapting the ideas of gradient boosting for regression, we arrive at `Algorithm 1` presented page 5.

(a) Explain in a few words what `Algorithm 1` is doing.

(b) Show that step (2)(b)(i) in Algorithm 1 reduces to

$$\text{Compute} \quad r_{ikm} = y_{ik} - p_{km}(x_i)\,.$$

(c) Using similar arguments to (v) and (vi), show that the solution to step (2)(b)(iii) in `Algorithm 1` can be approximated as

$$\gamma_{jkm} = \frac{K-1}{K}\frac{\sum_{x_i \in R_{jkm}} r_{ikm}}{\sum_{x_i \in R_{jkm}} |r_{ikm}|(1 - |r_{ikm}|)},$$

for $j = 1, \ldots, J_m$ and $k = 1, \ldots, K$.

`Algorithm 1`

(1) Initialise $f_k^{(1)}(x) = 0$ for $k = 1, \ldots, K$.

(2) For $m = 1, \ldots, M$

(a) Set

$$p_{km}(x) = \frac{e^{f_k^{(m)}(x)}}{\sum_{l=1}^{K} e^{f_l^{(m)}(x)}}, \quad k = 1, \ldots, K\,.$$

(b) For $k = 1, \ldots, K$,

(i) Compute

$$r_{ikm} = -\frac{\partial}{\partial f_k^{(m)}(x_i)} \left\{ \ell(y_{ik}, f_k^{(m)}(x_i)) \right\}$$

(ii) Fit a regression tree to the targets r_{ikm}, $i = 1, \ldots, n$, giving terminal regions R_{jkm}, $j = 1, \ldots, J_m$.

(iii) Compute for $j = 1, \ldots, J_m$ and $k = 1, \ldots, K$,

$$\gamma_{jkm} = \arg\min_{\gamma} \sum_{x_i \in R_{jkm}} \ell\left(y_i, f_k^{(m)}(x_i) + \gamma\right)$$

(iv) Update

$$f_k^{(m+1)}(x) = f_k^{(m)}(x) + \sum_{j=1}^{J_m} \gamma_{jkm} \mathbf{1}(x \in R_{jkm})$$

(3) Output $\hat{f}_k(x) = f_k^{(M+1)}(x)$, for $k = 1, \ldots, K$.

In `Algorithm 1`, $\ell(y, f(x))$ denotes the multinomial deviance loss function derived from the multinomial log-likelihood obtained in question (v), Part II, Problem 4, where the relationship between $f(x) = (f_1(x), \ldots, f_K(x))$ and $p(x) = (p_1(x), \ldots, p_K(x))$ is given in (1).