STATISTICAL MACHINE LEARNING: EXAM ASSIGNMENT EXAMPLES

Assignment 1 (6p). Assume you are going to learn a two-class classifier $h\colon \mathcal{X} \to \{+1, -1\}$ from examples with the goal to minimize the expected classification error. The classifier is selected from a hypothesis space $\mathcal{H}$ based on the minimal training error defined as the number of misclassified examples. Consider the following three cases of hypothesis space:

(1) $\mathcal{H}_1 = \{h(x) = \text{sign}(x - \theta) \mid \theta \in \mathbb{R}\}$.
(2) $\mathcal{H}_2 = \{h(x) = \text{sign}(|x - \mu_1| - |x - \mu_2|) \mid \mu_1 \in \mathbb{R}, \mu_2 \in \mathbb{R}\}$.
(3) $\mathcal{H}_3 = \{h(\boldsymbol{x}) = \text{sign}(\langle \boldsymbol{w}, \boldsymbol{x} \rangle + b) \mid \boldsymbol{w} \in \mathbb{R}^d, b \in \mathbb{R}\}$.

a) What is the Vapnik-Chervonenkis dimension of $\mathcal{H}_1$, $\mathcal{H}_2$ and $\mathcal{H}_3$?

b) Assume that in all three cases your algorithm can find a classifier with the minimal training error. In which cases is the algorithm statistically consistent?

Solution:
a) $VC(\mathcal{H}_1) = 1, VC(\mathcal{H}_2) = 2, VC(\mathcal{H}_3) = d + 1$ (lecture 4, slide 5-6).
b) If the VC dimension of a hypothesis space is finite then ULLN applies, and in turn the ERM is statistically consistent (lecture 4, slide 13). Hence the algorithm is statistically consistent in all three cases.

Assignment 2. (6p) We are given a set $\mathcal{H} = \{h_i\colon \mathcal{X} \to \{1, \ldots, 100\} \mid i = 1, \ldots, 1000\}$ containing 1000 strategies each predicting the human age $y \in \{1, \ldots, 100\}$ from a facial image $x \in \mathcal{X}$. The quality of a single strategy is measured by the expected absolute deviation between the predicted age and the true age

$$R^{\text{MAE}}(h) = \mathbb{E}_{(x,y)\sim p}(|y - h(x)|) \,,$$

where the expectation is computed w.r.t. an unknown distribution $p(x, y)$. The empirical estimate of $R^{\text{MAE}}(h)$ reads

$$R_{\mathcal{T}^m}(h) = \frac{1}{m} \sum_{j=1}^{m} |y^j - h(x^j)|$$

where $\mathcal{T}^m = \{(x^j, y^j) \in (\mathcal{X} \times \mathcal{Y}) \mid j = 1, \ldots, m\}$ is a set of examples drawn from i.i.d. random variables with the distribution $p(x, y)$. What is the minimal number of the training examples m which guarantees that $R^{\text{MAE}}(h)$ is in the interval $(R_{\mathcal{T}^m}(h) - 1, R_{\mathcal{T}^m}(h) + 1)$ for every $h \in \mathcal{H}$ with probability at least 95%?

Solution:
The question can be restated as: what is the minimal number of examples m such that

$$\mathbb{P}\Big(\max_{h\in\mathcal{H}} \big| R_{\mathcal{T}^m}(h) - R^{\text{MAE}}(h) \big| < \varepsilon \Big) \geq \gamma \tag{1}$$

holds, where $\varepsilon = 1$ and $\gamma = 0.95$? Using probability of a complementary event we get

$$\mathbb{P}\Big(\max_{h\in\mathcal{H}} \big|R_{\mathcal{T}^m}(h) - R^{\mathrm{MAE}}(h)\big| < \varepsilon \Big) = 1 - \mathbb{P}\Big(\max_{h\in\mathcal{H}} \big|R_{\mathcal{T}^m}(h) - R^{\mathrm{MAE}}(h)\big| \geq \varepsilon \Big) \,. \quad (2)$$

Hoeffding inequality generalized for finite hypothesis space (lecture 3, slide 9) states that

$$\mathbb{P}\Big(\max_{h\in\mathcal{H}} \big|R_{\mathcal{T}^m}(h) - R(h)\big| \geq \varepsilon \Big) \leq 2\,|\mathcal{H}|\, e^{-\frac{2m\,\varepsilon^2}{(\ell_{max}-\ell_{min})^2}} \,. \quad (3)$$

Combining (2) and (3) yields

$$\mathbb{P}\Big(\max_{h\in\mathcal{H}} \big|R_{\mathcal{T}^m}(h) - R^{\mathrm{MAE}}(h)\big| < \varepsilon \Big) \geq 1 - 2\,|\mathcal{H}|\, e^{-\frac{2m\,\varepsilon^2}{(\ell_{max}-\ell_{min})^2}} = \gamma \,. \quad (4)$$

Solving (4) for m when everything else is fixed yields

$$m \geq \frac{\log(2|\mathcal{H}|) - \log(1-\gamma)}{2\,\varepsilon^2}(\ell_{max} - \ell_{min})^2 \,.$$

In case of $\mathcal{Y} = \{1, \ldots, 100\}$ and the MAE loss $\ell(y, y') = |y - y'|$ we have $\ell_{min} = 0$ and $\ell_{max} = 99$. We further have $\gamma = 0.95$, $\varepsilon = 1$ and $|\mathcal{H}| = 1000$. Hence,

$$m \geq \frac{\log(2 \cdot 1000) - \log(1 - 0.95)}{2 \cdot 1^2}(99 - 0)^2 \approx 51\,928.8 \,.$$

That is, the minimal number of training examples we need is $m = 51\,929$.

Assignment 3 (6p). A non-negative random variable $X \geqslant 0$ has exponential distribution $p(x) = be^{-bx}$, where b is an unknown parameter.

a) Explain how to estimate this parameter from an i.i.d. training set $\mathcal{T}^m = \{x_j \in \mathbb{R}_+ \mid j = 1, \ldots, m\}$ by using the maximum likelihood estimator .

b) The random variable Y is a mixture

$$Y = \lambda X_1 + (1 - \lambda)X_2$$

of two exponentially distributed variables with unknown parameters b_1, b_2 and unknown mixture weight $0 < \lambda < 1$. Explain how to estimate all mixture parameters from an i.i.d. training set $\mathcal{T}^m = \{y_j \in \mathbb{R}_+ \mid j = 1, \ldots, m\}$ by using the EM-algorithm.

Solution:

a) The maximum likelihood estimator is

$$e_{ML}(\mathcal{T}^m) = \arg\max_b \frac{1}{m} \sum_{x\in\mathcal{T}^m} \log p_b(x).$$

Substituting the model, we get the task

$$-\frac{b}{m} \sum_{x\in\mathcal{T}^m} x + \log b \to \max_b \,.$$

Setting the derivative of the objective function w.r.t. b to zero, we get the solution $b = 1/\bar{x}$, where $\bar{x}$ denotes the average of the training examples.

b) The EM algorithm starts with some initialisation $\lambda^{(0)}$, $b_1^{(0)}$, $b_2^{(0)}$ of the model and iterates the following steps.

E-step: For each example $y \in \mathcal{T}^m$, compute the posterior probability of the first component, i.e.

$$\alpha_y = \frac{\lambda p_{b_1}(x)}{\lambda p_{b_1}(x) + (1-\lambda) p_{b_2}(x)},$$

by using the current estimate of the model parameters. The posterior probability of the second component is $1 - \alpha_y$.

M-step: We have to solve the task

$$\frac{1}{m} \sum_{y \in \mathcal{T}^m} \big[\alpha_y \log \lambda p_{b_1}(y) + (1-\alpha_y) \log(1-\lambda) p_{b_2}(y)\big] \to \max_{\lambda, b_1, b_2}$$

This task decomposes into independent optimisation tasks for λ, b_1 and b_2. We obtain

$$\lambda^{(t+1)} = \frac{1}{m} \sum_{y \in \mathcal{T}^m} \alpha_y, \quad b_1^{(t+1)} = \frac{\sum_{y \in \mathcal{T}^m} \alpha_y}{\sum_{y \in \mathcal{T}^m} \alpha_y y}$$

and a similar expression for $b_2^{(t+1)}$, where α_y is replaced by $1 - \alpha_y$.

The algorithm is stopped if the change in the α-s becomes smaller than some predefined threshold.

Assignment 4 (4p). Consider a homogeneous Markov model for sequences $s = (s_1, \ldots, s_n)$ with elements from a finite set K. Its joint distribution is given by

$$p(s) = p(s_1) \prod_{i=2}^{n} p(s_i \,|\, s_{i-1}),$$

where $p(s_1 = k)$ is the marginal distribution for the first element of the sequence and $p(s_i = k \,|\, s_{i-1} = k')$ is the matrix of transition probabilities. Given a state $k^* \in K$, we want to know its expected number of occurrences in a sequence generated by the model. Give an algorithm for computing this expectation.
Hint: Use the fact that the expected value of a sum of random variables is equal to the sum of their expected values.

Solution:
The expected number of occurrences can be written as

$$\mathbb{E}_{s \sim p(s)} n(k^*) = \mathbb{E}_{s \sim p(s)} \sum_{i=1}^{n} [\![s_i = k^*]\!] = \sum_{i=1}^{n} p(s_i = k^*)$$

For this we need to compute the marginal probabilities of the state k^* in all positions $i = 1, \ldots, n$. We can compute all marginal probabilities by dynamic matrix-vector multiplication from left to right with run-time complexity $\mathcal{O}(nK^2)$, then pick the needed ones and sum them.

Assignment 5 (6p). Consider the following simple neural network having n inputs:

$$\hat{y}(\boldsymbol{x}, \boldsymbol{w}) = \sigma\left(\sum_{i=1}^{n} w_i\, x_i\right),$$

where σ is the logistic sigmoid function:

$$\sigma(s) = \frac{1}{1 + e^{-s}}.$$

The network is trained using Stochastic Gradient Descent where the training set can be described as $\mathcal{T}^m = \{(x^i, y^i) \in (\mathbb{R}^n \times \{0, 1\}) \mid i = 1, \ldots, m\}$. The loss function is the binary cross-entropy:

$$\ell(y, \hat{y}) = -y \log \hat{y} - (1 - y) \log(1 - \hat{y}).$$

(1) Use the back-propagation algorithm and derive the gradient for a single sample:

$$\nabla \ell(\boldsymbol{w}) = \left(\frac{\partial \ell}{\partial w_1}, \frac{\partial \ell}{\partial w_2}, \ldots, \frac{\partial \ell}{\partial w_n}\right).$$

(2) Reuse the neuron activity computed during the forward pass and simplify the result.

Solution:

(1) Describe messages for all layers (including loss):
   (a) cross entropy backward message:

$$\frac{\partial \ell}{\partial \hat{y}} = -\frac{y}{\hat{y}} + \frac{1 - y}{1 - \hat{y}},$$

   (b) sigmoid backward message:

$$\frac{\partial \sigma}{\partial s} = \frac{e^{-s}}{(1 + e^{-s})^2},$$

   (c) linear layer parameter message:

$$\frac{\partial s}{\partial w_i} = x_i.$$

   Define δs passing errors in backward direction:

$$\delta_{\text{loss}} = \frac{\partial \ell}{\partial \hat{y}},$$

$$\delta_{\text{sigmoid}} = \delta_{\text{loss}} \frac{\partial \sigma}{\partial s}.$$

   The elements of the gradient of loss w.r.t. the parameters of the linear layer are then defined as:

$$\frac{\partial \ell}{\partial w_i} = \delta_{\text{sigmoid}} \frac{\partial s}{\partial w_i}.$$

(2) The sigmoid backward message can be expressed using the sigmoid itself:

$$\frac{\partial \sigma}{\partial s} = \frac{1+e^{-s}}{(1+e^{-s})^2} - \frac{1}{(1+e^{-s})^2} = \sigma(s) - \sigma^2(s) = \hat{y}(1-\hat{y}).$$

We can now simplify:

$$\delta_{\text{sigmoid}} = \left(-\frac{y}{\hat{y}} + \frac{1-y}{1-\hat{y}}\right) \cdot \hat{y}(1-\hat{y}) = -y(1-\hat{y}) + (1-y)\hat{y} = \hat{y} - y.$$

Finally we get:

$$\frac{\partial \ell}{\partial w_i} = (\hat{y} - y)x_i.$$

Assignment 6 (4p). Consider a regression problem with multiple training datasets $\mathcal{T}^m = \{(x_i, y_i) \mid i = 1, \ldots, m\}$ of size m generated by using

$$y = f(x) + \epsilon, \tag{5}$$

where ϵ is noise with $\mathbb{E}(\epsilon) = 0$ and $\mathrm{Var}(\epsilon) = \sigma^2$. Derive the bias-variance decomposition for the 1-nearest-neighbor regression. The response of the 1-NN regressor is defined as:

$$h_m(x) = y_{n(x)} = f(x_{n(x)}) + \epsilon,$$

where $n(x)$ gives the index of the nearest neighbor of x in $\mathcal{T}^m$. For simplicity assume that all x_i are the same for all training datasets $\mathcal{T}^m$ in consideration, hence, the randomness arises from the noise ϵ, only.

Give the squared bias:

$$\mathbb{E}_x\left[\big(g_m(x) - f(x)\big)^2\right] = \mathbb{E}_x\left[\Big(\mathbb{E}_{\mathcal{T}^m}\big[h_m(x)\big] - f(x)\Big)^2\right]$$

and variance:

$$\mathrm{Var}_{x,\mathcal{T}^m}\big(h_m(x)\big).$$

Solution: The squared bias is:

$$\mathbb{E}_x\left[\Big(\mathbb{E}_{\mathcal{T}^m}\left[h_m(x)\right] - f(x)\Big)^2\right] = \mathbb{E}_x\left[\Big(\mathbb{E}_{\mathcal{T}^m,\epsilon}\big[f(x_{n(x)}) + \epsilon\big] - f(x)\Big)^2\right] =$$
$$\mathbb{E}_x\left[\Big(\mathbb{E}_{\mathcal{T}^m}\left[f(x_{n(x)})\right] + \mathbb{E}_\epsilon\left[\epsilon\right] - f(x)\Big)^2\right] = \mathbb{E}_x\left[\big(f(x_{n(x)}) - f(x)\big)^2\right],$$

where the first expected value vanished due to the fixed x_i assumption.

The variance is similarly:

$$\mathrm{Var}_{x,\mathcal{T}^m}\big(h_m(x)\big) = \mathrm{Var}_{x,\mathcal{T}^m,\epsilon}\big(f(x_{n(x)}) + \epsilon\big) = \mathrm{Var}_\epsilon\big(\epsilon\big) = \sigma^2.$$

Assignment 7 (7p)**.** Consider a linear classifier $h\colon \mathcal{X} \times \mathcal{X} \to \mathcal{Y} \times \mathcal{Y}$ predicting a pair of labels $(y_1, y_2) \in \mathcal{Y} \times \mathcal{Y}$ from a pair of inputs $(x_1, x_2) \in \mathcal{X} \times \mathcal{X}$ based on the rule

$$h(x_1, x_2; \boldsymbol{\theta}) = \operatorname*{arg\,max}_{y_1 \in \mathcal{Y}, y_2 \in \mathcal{Y}} \left(\langle \boldsymbol{\phi}(x_1), \boldsymbol{w}_{y_1} \rangle + \langle \boldsymbol{\phi}(x_2), \boldsymbol{w}_{y_2} \rangle + g(y_1, y_2)\right) \tag{6}$$

where $\boldsymbol{\phi}\colon \mathcal{X} \to \mathbb{R}^n$ is a feature map, $\boldsymbol{w}_y \in \mathbb{R}^n$, $y \in \mathcal{Y}$, are vectors and $g\colon \mathcal{Y} \times \mathcal{Y} \to \mathbb{R}$ is a function. The vector $\boldsymbol{\theta} \in \mathbb{R}^{n|\mathcal{Y}|+|\mathcal{Y}|^2}$ encapsulates all parameters of the classifier,

that is, the vectors $\{\boldsymbol{w}_y \in \mathbb{R}^n \mid y \in \mathcal{Y}\}$ and the function values $\{g(y, y') \in \mathbb{R} \mid y \in \mathcal{Y}, y' \in \mathcal{Y}\}$. Let $\mathcal{T}^m = \{(x_1^j, x_2^j, y_1^j, y_2^j) \in (\mathcal{X} \times \mathcal{X} \times \mathcal{Y} \times \mathcal{Y}) \mid j = 1, \ldots, m\}$ and $\mathcal{S}^l = \{(x_1^j, x_2^j, y_1^j, y_2^j) \in (\mathcal{X} \times \mathcal{X} \times \mathcal{Y} \times \mathcal{Y}) \mid j = 1, \ldots, l\}$ be a set of training and testing examples, respectively, both being drawn from i.i.d. random variables with the distribution $p(x_1, x_2, y_1, y_2)$.

a) Describe a variant of the Perceptron algorithm which finds the parameters $\boldsymbol{\theta}$ such that the classifier (6) predicts all examples from $\mathcal{T}^m$ correctly provided such parameters exists.

b) Assume that $\mathcal{Y} = \{0, \ldots, 9\}$ and that we want to measure the prediction accuracy of h by computing the absolute deviation between the sum of the correct and the predicted labels. To this end, we define a loss function $\ell(y_1, y_2, \hat{y}_1, \hat{y}_2) = |y_1 + y_2 - \hat{y}_1 - \hat{y}_2|$ and its expected value

$$R(h) = \mathbb{E}_{(x_1,x_2,y_1,y_2)\sim p}\left[\ell\big(y_1, y_2, h_1(x_1), h_2(x_2)\big)\right].$$

As the distribution $p(x_1, x_2, y_1, y_2)$ is unknown we estimate $R(h)$ by the test error

$$R_{\mathcal{S}^l}(h) = \frac{1}{l}\sum_{j=1}^{l} \ell(y_1^j, y_2^j, h_1(x_1^j), h_2(x_2^j)) .$$

What is the minimal number of the test examples l which we need to collect in order to guarantee with probability γ that $R_{\mathcal{S}^l}(h)$ deviates from $R(h)$ by ε at most ? Write l as a function of ε and γ.

Solution:
a)

1: $\boldsymbol{w}_y \leftarrow \mathbf{0}$, $y \in \mathcal{Y}$ and $g(y, y') \leftarrow 0$, $(y, y') \in \mathcal{Y}^2$
2: Find $(\bar{x}_1, \bar{x}_1, \bar{y}_1, \bar{y}_2) \in \mathcal{T}^m$ such that

$$(\bar{y}_1, \bar{y}_2) \neq (\hat{y}_1, \hat{y}_2) \in \operatorname*{arg\,max}_{y_1\in\mathcal{Y}, y_2\in\mathcal{Y}} (\langle \boldsymbol{\phi}(\bar{x}_1), \boldsymbol{w}_{y_1}\rangle + \langle \boldsymbol{\phi}(\bar{x}_2), \boldsymbol{w}_{y_2}\rangle + g(y_1, y_2))$$

If such $(\bar{x}_1, \bar{x}_1, \bar{y}_1, \bar{y}_2)$ does not exist then stop and return $\boldsymbol{\theta} = (\boldsymbol{w}_y, y \in \mathcal{Y}$ and $g(y, y'), (y, y') \in \mathcal{Y}^2)$.
3: Update solution

$$\begin{aligned}
\boldsymbol{w}_{\bar{y}_1} &\leftarrow \boldsymbol{w}_{\bar{y}_1} + \boldsymbol{\phi}(\bar{x}_1)\\
\boldsymbol{w}_{\bar{y}_2} &\leftarrow \boldsymbol{w}_{\bar{y}_2} + \boldsymbol{\phi}(\bar{x}_2)\\
\boldsymbol{w}_{\hat{y}_1} &\leftarrow \boldsymbol{w}_{\hat{y}_1} - \boldsymbol{\phi}(\bar{x}_1)\\
\boldsymbol{w}_{\hat{y}_2} &\leftarrow \boldsymbol{w}_{\hat{y}_2} - \boldsymbol{\phi}(\bar{x}_2)\\
g(\bar{y}_1, \bar{y}_2) &\leftarrow g(\bar{y}_1, \bar{y}_2) + 1\\
g(\hat{y}_1, \hat{y}_2) &\leftarrow g(\hat{y}_1, \hat{y}_2) - 1
\end{aligned}$$

4: Go to 2.

---

b) From the Hoeffding inequality we can derive (lecture 2, slide 10) that

$$l = \frac{\log(2) - \log(1-\gamma)}{2\,\varepsilon^2}(\ell_{\max} - \ell_{\min})^2$$

where $\ell_{\min} = \min_{(y_1,y_2,\hat{y}_1,\hat{y}_2)\in\mathcal{Y}^4} |y_1+y_2-\hat{y}_1-\hat{y}_2| = 0$ and $\ell_{\max} = \min_{(y_1,y_2,\hat{y}_1,\hat{y}_2)\in\mathcal{Y}^4} |y_1+y_2-\hat{y}_1-\hat{y}_2| = 18$.

Assignment 8 (4p)**.** A discrete random variable $x \in \mathbb{N}$ is Poisson distributed with

$$p(x=k) = \frac{\lambda^k e^{-\lambda}}{k!},$$

where λ is a positive parameter.

a) Show that Poisson distributions form an exponential family. Express the natural parameter as a function of λ.

b) Deduce the maximum likelihood estimate for the natural parameter of a Poisson distribution, given an i.i.d. training set $\mathcal{T}^m = \{k_i \in \mathbb{N} \mid i = 1\ldots, m\}$.

Solution:
a) We can write

$$p(x=k) = \frac{1}{k!}\exp[k\log\lambda - \lambda] = h(k)\exp[k\eta - A(\eta)]$$

with $\eta = \log\lambda$ and $A(\eta) = e^{\eta}$.

b) Substituting this into the log-likelihood of the training set, we get the task

$$\eta\bar{k} - A(\eta) \to \max_{\eta},$$

where $\bar{k}$ denotes the average over the training set. The solution is $\eta = \log\bar{k}$.

Assignment 9 (3p)**.** Consider a homogeneous Markov chain model with three states $k = 1, 2, 3$ and the transition probability matrix

$$P_{k,k'} = 0.5\begin{pmatrix} 1 & 0 & 0 \\ 1 & 1 & 0 \\ 0 & 1 & 2 \end{pmatrix}$$

It starts in state $k = 1$. Compute the marginal probabilities for its state after three transitions.

Solution: We need to compute $\pi_3 = P^3\pi_0$ with $\pi_0^T = (1, 0, 0)$ and obtain $\pi_3^T = \frac{1}{8}(1, 3, 4)$.

Assignment 10 (3p)**.** A convolutional layer transforms an input volume $W_{\text{in}} \times H_{\text{in}} \times C$ into an output volume $W_{\text{out}} \times H_{\text{out}} \times D$, where W_{in} and H_{in} define spatial dimensions of the input and C is the number of input channels. Similarly W_{out} and H_{out} denote spatial dimensions of the output and D the number of filters. Consider stride S, zero padding P and filters having receptive field of $F \times F$ units.

a) Give the types and the total number of parameters of the layer.

b) Consider padding P preserving the size of the output in the W dimension, i.e., $W_{\text{in}} = W_{\text{out}}$. Give P as a function of F, S and W_{in}.

Solution:

a) There are F^2CD weights and D biases.

b) The padding have to be set to $P = \frac{(W-1)\cdot S+F-W}{2}$ to preserve spatial dimensions of the input.

Assignment 11 (5p)**.** Consider the squared logarithmic loss:

$$\ell\big(y, h(x)\big) = \Big[\log\big(1+y\big) - \log\big(1+h(x)\big)\Big]^2,$$

where y is the target and $h(x)$ the output of the regressor for input x. Give the pseudo code for the corresponding Gradient Boosting Machine using this loss (including the gradient) and discuss differences to the squared loss GBM.

Solution: We need the following derivative:

$$\frac{\partial l}{\partial h(x)} = -2\Big[\log\big(1+y\big) - \log\big(1+h(x)\big)\Big]\frac{1}{1+h(x)}.$$

Then the GBM is defined as follows:

(1) Initialize $f_0(x) = 0$ or $f_0(x) = \arg\min_\gamma \sum_{i=1}^m \ell(y_i, \gamma)$

(2) For $k = 1$ to K:

(a) Compute:

$$\boldsymbol{g}_k = \left[\frac{\partial \ell(y_i, f_{k-1}(x_i))}{\partial f_{k-1}(x_i)}\right]_{i=1}^m = \left[-2\Big[\log\big(1+y\big) - \log\big(1+f_{k-1}(x_i)\big)\Big]\frac{1}{1+f_{k-1}(x_i)}\right]_{i=1}^m$$

(b) Fit a regression model $b(\cdot;\theta)$ to $-\boldsymbol{g}_k$ using squared loss:

$$\theta_k = \arg\min_\theta \sum_{i=1}^m \left[(-\boldsymbol{g}_k)_i - b(x_i;\theta)\right]^2$$

(c) Choose a fixed step size $\beta_k = \beta > 0$ or use line search:

$$\beta_k = \arg\min_{\beta>0} \sum_{i=1}^m \ell\Big(y_i, f_{k-1}(x_i) + \beta b(x_i;\theta_k)\Big)$$

(d) Set $f_k(x) = f_{k-1}(x) + \beta_k b(x;\theta_k)$

(3) Return $h_m(x) = f_K(x)$