Stat 648: Assignment 4 Solutions

Problem 11: Various solutions

Problem 12: Details for the first 3 iterations are given in Table 1. After 3 iterations all points are correctly classified and thus the misclassification error is 0. If one were to continue the AdaBoost algorithm, the split variables and cutoffs values repeat (i.e. for iteration 4 we split variable 1 at 2, for iteration 5 we split variable 1 at 7, etc.). The final classifier is displayed in Figure 1.

	Iteration		
	$m = 1$	$m = 2$	$m = 3$
w_{1m}	0.1000	0.1000	0.1000
w_{2m}	0.1000	0.2333	0.2333
w_{3m}	0.1000	0.2333	0.2333
w_{4m}	0.1000	0.2333	0.2333
w_{5m}	0.1000	0.1000	0.1000
w_{6m}	0.1000	0.1000	0.1000
w_{7m}	0.1000	0.1000	0.3667
w_{8m}	0.1000	0.1000	0.3667
w_{9m}	0.1000	0.1000	0.1000
w_{10m}	0.1000	0.1000	0.3667
α_m	0.8472979	1.2992830	1.8458267
$\overline{err}_m$	0.3000000	0.2142857	0.1363636
Split variable	1	1	2
Cutoff value	2	7	5.25

Table 1: Values from the first 3 iterations of the AdaBoost algorithm

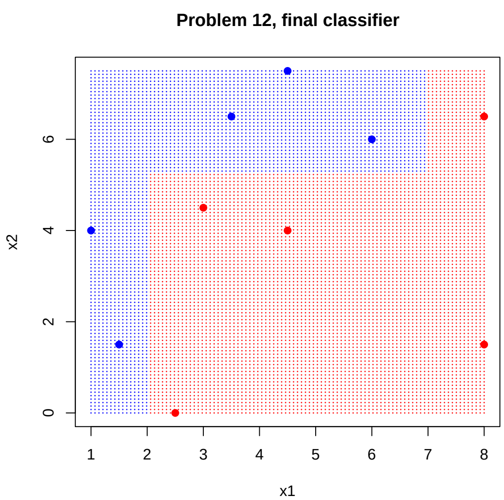

Figure 1: Final classifier using AdaBoost

Problem 13: Let X represent the input variable, Z the first hidden layer variable, W the second hidden layer variable, and Y the output.

For hidden layer 1, $z_{0i} = 1$ and $z_{mi} = \sigma(\alpha_{0m} + \boldsymbol{\alpha}'_m \mathbf{x}_i)$, where $\boldsymbol{\alpha}_m = (\alpha_{1m}, \alpha_{2m}, \alpha_{3m})'$ and $\mathbf{x}_i = (x_{i1}, x_{i2}, x_{i3})'$, $i = 1, \ldots, N$, $m = 1, 2$. Let $\mathbf{z}_i = (z_{1i}, z_{2i})'$ and note that $x_{i0} = 1$.

For hidden layer 2, $w_{0i} = 1$ and $w_{ki} = \sigma(\beta_{0k} + \boldsymbol{\beta}'_k \mathbf{z}_i)$, where $\boldsymbol{\beta}_k = (\beta_{1k}, \beta_{2k})'$, $i = 1, \ldots, N$, $k = 1, 2$. Let $\mathbf{w}_i = (w_{1i}, w_{2i})'$.

Finally, $f(\mathbf{x}_i) = g(\gamma_0 + \boldsymbol{\gamma}' \mathbf{w}_i)$, where $\boldsymbol{\gamma} = (\gamma_1, \gamma_2)'$.

Let $\boldsymbol{\theta}$ stand for the whole set of parameters and consider the SEL fitting criterion

$$R(\boldsymbol{\theta}) = \frac{1}{2} \sum_{i=1}^{N} (y_i - f(\mathbf{x}_i))^2.$$

Denote $R_i = \frac{1}{2}(y_i - f(\mathbf{x}_i))^2$. Partial derivatives with respect to the parameters are (by the chain rule)

$$\begin{aligned} \frac{\partial R_i}{\partial \gamma_k} &= -(y_i - f(\mathbf{x}_i)) g'(\gamma_0 + \boldsymbol{\gamma}' \mathbf{w}_i) w_{ki}, \quad k = 0, 1, 2 \\ &= \delta_i w_{ki} \quad \text{for } \delta_i = -(y_i - f(\mathbf{x}_i)) g'(\gamma_0 + \boldsymbol{\gamma}' \mathbf{w}_i) \end{aligned} \tag{1}$$

$$\begin{aligned} \frac{\partial R_i}{\partial \beta_{mk}} &= -(y_i - f(\mathbf{x}_i)) g'(\gamma_0 + \boldsymbol{\gamma}' \mathbf{w}_i) \gamma_k \sigma'(\beta_{0k} + \boldsymbol{\beta}'_k \mathbf{z}_i) z_{mi}, \quad m = 0, 1, 2, \quad k = 1, 2 \\ &= \delta_i \gamma_k \sigma'(\beta_{0k} + \boldsymbol{\beta}'_k \mathbf{z}_i) z_{mi} \\ &= \eta_{ki} z_{mi} \quad \text{for} \quad \eta_{ki} = \delta_i \gamma_k \sigma'(\beta_{0k} + \boldsymbol{\beta}'_k \mathbf{z}_i) \end{aligned} \tag{2}$$

$$\begin{aligned} \frac{\partial R_i}{\partial \alpha_{lm}} &= -(y_i - f(\mathbf{x}_i)) g'(\gamma_0 + \boldsymbol{\gamma}' \mathbf{w}_i) \frac{\partial \sum_{k=1}^{2} \gamma_k w_{ki}}{\partial \alpha_{lm}} \\ &= -(y_i - f(\mathbf{x}_i)) g'(\gamma_0 + \boldsymbol{\gamma}' \mathbf{w}_i) \sum_{k=1}^{2} [\gamma_k \sigma'(\beta_{0k} + \boldsymbol{\beta}'_k \mathbf{z}_i) \beta_{mk}] \sigma'(\alpha_{0m} + \boldsymbol{\alpha}'_m \mathbf{x}_i) x_{il} \\ &= \left(\sum_{k=1}^{2} \eta_{ki} \beta_{mk} \sigma'(\alpha_{0m} + \boldsymbol{\alpha}'_m \mathbf{x}_i) \right) x_{il} \\ &= \lambda_{mi} x_{il} \quad \text{for} \quad \lambda_{mi} = \sum_{k=1}^{2} \eta_{ki} \beta_{mk} \sigma'(\alpha_{0m} + \boldsymbol{\alpha}'_m \mathbf{x}_i) \end{aligned} \tag{3}$$

An iterative search to make $R(\boldsymbol{\theta})$ small, with learning rate φ_r, proceeds by setting

$$\gamma_k^{(r+1)} = \gamma_k^{(r)} - \varphi_r \sum_{i=1}^{N} \left. \frac{\partial R_i}{\partial \gamma_k} \right|_{\gamma_0^{(r)} \text{ and } \boldsymbol{\gamma}^{(r)}} = \gamma_k^{(r)} - \varphi_r \sum_{i=1}^{N} \delta_i^{(r)} w_{ki}^{(r)} \tag{4}$$

$$\beta_{mk}^{(r+1)} = \beta_{mk}^{(r)} - \varphi_r \sum_{i=1}^{N} \eta_{ki}^{(r)} z_{mi}^{(r)} \tag{5}$$

$$\alpha_{lm}^{(r+1)} = \alpha_{lm}^{(r)} - \varphi_r \sum_{i=1}^{N} \lambda_{mi}^{(r)} x_{il}^{(r)} \tag{6}$$

So operationally

1. Forward pass: use the r^{th} iterates of α's, β's, and γ's to compute $f^{(r)}(\mathbf{x}_i)$
2. Backward pass:
   - use the r^{th} iterates of $f(\mathbf{x}_i)$, γ's and w's to compute $\delta_i^{(r)}$
   - use the r^{th} iterates of δ's, γ's, β's and z's to compute $\eta_{ki}^{(r)}$
   - use the r^{th} iterates of η's, β's, and α's to compute $\lambda_{mi}^{(r)}$
3. The $\delta_i^{(r)}$, $\eta_{ki}^{(r)}$, and $\lambda_{mi}^{(r)}$ then provide partials using (1), (2), and (3)
4. Updates for the parameters α, β, and γ come from (4), (5), and (6)