Solutions to Selected Problems in *Machine Learning: An Algorithmic Perspective*

Alex Kerr
email: ajkerr0@gmail.com

Chapter 2

Problem 2.1

Let's say S is the event that someone at the party went to the same school, R is the event that someone at the party is vaguely recognizable. By Bayes' rule:

$$P(S|R) = \frac{P(R|S)P(S)}{P(R)} \tag{1}$$

We are given:

$$\begin{aligned} P(R|S) &= 1/2 = 0.5 \\ P(S) &= 1/10 = 0.1 \\ P(R) &= 1/5 = 0.2 \end{aligned}$$

The result is

$$P(S|R) = \frac{0.5 \times 0.1}{0.2} = 0.25 \tag{2}$$

Chapter 3

Problem 3.1

$$\boldsymbol{x} \cdot \boldsymbol{w}^T - \text{bias} = 0 \tag{3}$$

$$x_1 + x_2 = -1.5 \tag{4}$$

This function would not correspond to a logic gate, but a constant voltage source.

Problem 3.7

Lagrange Multipliers

We are trying to minimize $f(\boldsymbol{x}) = ||\boldsymbol{x} - \boldsymbol{x}'||^2$ while constraining $\boldsymbol{x}$ to the hyperplane $y = \sum_i w_i x_i + b$. First look at partial derivatives:

$$\nabla f = 2 \sum_i \left(x_i - x_i'\right) \hat{\boldsymbol{x}}_i \tag{5}$$

$$\nabla g = \sum_i w_i \hat{\boldsymbol{x}}_i \tag{6}$$

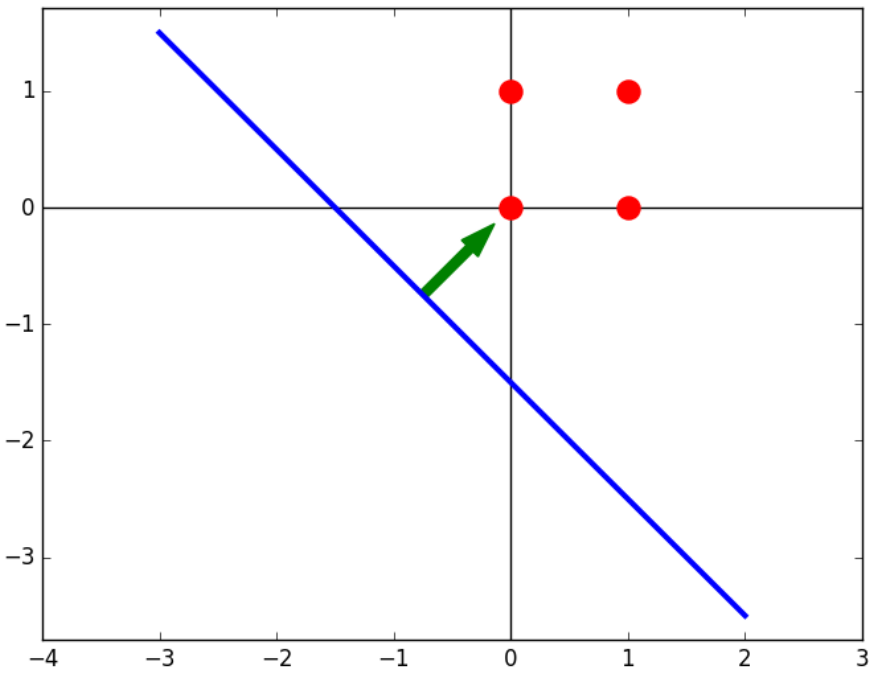


Figure 3.1: The *blue* line is the decision boundary ($x_2 = -x_1 - 1.5$), the *green* arrow is the weight vector $\boldsymbol{w} = (1, 1)$, and the *red* points are the standard logic gate inputs.

In accordance to $\nabla f - \lambda \nabla g = 0$ we have

$$x_i = x_i' + \frac{\lambda}{2} w_i \tag{7}$$

Sub this back into the constraint g:

$$\sum_i w_i x_i' + \frac{\lambda}{2} \sum_i w_i^2 + b = 0 \tag{8}$$

$$\lambda = -2 \frac{\sum_i w_i x_i' + b}{\sum_i w_i^2} \tag{9}$$

Sub Equation 7 back into f and use our value for λ:

$$f_{\min} = \frac{\lambda^2}{4} \sum_i w_i^2 = \frac{\left(\sum_i w_i x_i' + b\right)^2}{\sum_i w_i^2} = \frac{y(\boldsymbol{x}')^2}{||\boldsymbol{w}||^2} \tag{10}$$

Finally the distance:

$$||\boldsymbol{x} - \boldsymbol{x}'||_{\min} = \sqrt{f_{\min}} = \frac{|y(\boldsymbol{x}')|}{||\boldsymbol{w}||} \tag{11}$$

Projection

The shortest vector between $\boldsymbol{x}'$ and the hyperplane is parallel to the vector orthogonal to the hyperplane. This is the vector that points 'most directly' to $\boldsymbol{x}$. In the case of the Perceptron boundary, this vector is $\boldsymbol{w}^T$. The boundary is 'flat' so if we ensure we project $\boldsymbol{x}' - \boldsymbol{x}$ to $\boldsymbol{w}^T$ we are good to go. For the projection, I like to use the Gram-Schmidt process as a reference.

$$\text{proj}_{\boldsymbol{w}}(\boldsymbol{x}' - \boldsymbol{x}) = \frac{(\boldsymbol{x}' - \boldsymbol{x}) \cdot \boldsymbol{w}}{\boldsymbol{w} \cdot \boldsymbol{w}} \boldsymbol{w} = \frac{\boldsymbol{x}' \cdot \boldsymbol{w} - \boldsymbol{x} \cdot \boldsymbol{w}}{w^2} \boldsymbol{w} \tag{12}$$

Substitute in the discriminant function:

$$\boldsymbol{w} \cdot \boldsymbol{x} = -b \tag{13}$$

$$\frac{\boldsymbol{x}' \cdot \boldsymbol{w} + b}{w^2} \boldsymbol{w} = \frac{y(\boldsymbol{x}')}{w^2} \boldsymbol{w} = \frac{y(\boldsymbol{x}')}{w} \hat{\boldsymbol{w}} \tag{14}$$

The magnitude is

$$\frac{|y(\boldsymbol{x}')|}{||\boldsymbol{w}||} \tag{15}$$

Chapter 4

Problem 4.5

Refer to the `train_seq` method of the multi-level Perceptron class `MLP` on my **github repository**.

Problem 4.12

$$\begin{aligned} \tanh(h) &= \frac{e^h - e^{-h}}{e^h + e^{-h}} = \frac{1 - e^{-2h}}{1 + e^{-2h}} \\ &= \frac{2 - 1 - e^{-2h}}{1 + e^{-2h}} \\ &= \frac{2}{1 + e^{-2h}} - \frac{1 + e^{-2h}}{1 + e^{-2h}} \\ &= 2g(2h) - 1 \end{aligned} \tag{16}$$

Chapter 6

Problem 6.1

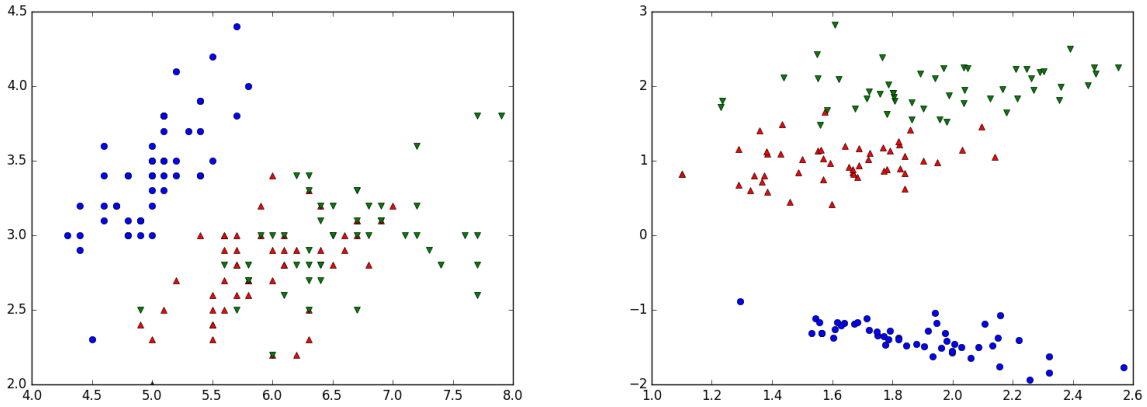
Refer to my `lda` function on **github**.
The author may have made mistakes in the text relating the scatter matrices. At the very least he wasn't clear. The covariance matrix is estimated by the total scatter matrix in the `numpy` routine:

$$\text{cov}(\boldsymbol{x}_a, \boldsymbol{x}_b) = \text{E}\left[(\boldsymbol{X} - \text{E}[\boldsymbol{X}])(\boldsymbol{X} - \text{E}[\boldsymbol{X}])^T\right] \simeq \frac{1}{n} \sum_{i=1}^{n} (\boldsymbol{x}_i - \boldsymbol{\mu})(\boldsymbol{x}_i - \boldsymbol{\mu})^T \tag{17}$$

where the mean data point is understood to be the sample mean:

$$\boldsymbol{\mu} = \frac{1}{n} \sum_{i=1}^{n} \boldsymbol{x}_i \tag{18}$$

We can expand the total scatter matrix as:



(a) (b)

Figure 6.1 *Left*: The original iris data. *Right*: The reduced iris data after LDA.

$$\begin{aligned}\frac{1}{n}\sum_{i=0}^{n}(\boldsymbol{x}_i-\boldsymbol{\mu})(\boldsymbol{x}_i-\boldsymbol{\mu})^T &= \frac{1}{n}\sum_c\sum_{j\in c}(\boldsymbol{x}_j-\boldsymbol{\mu})(\boldsymbol{x}_j-\boldsymbol{\mu})^T \\ &= \frac{1}{n}\sum_c\sum_{j=1}^{m_c}(\boldsymbol{x}_j-\boldsymbol{\mu}_c)(\boldsymbol{x}_j-\boldsymbol{\mu}_c)^T+(\boldsymbol{\mu}_c-\boldsymbol{\mu})(\boldsymbol{\mu}_c-\boldsymbol{\mu})^T \\ &+(\boldsymbol{x}_j-\boldsymbol{\mu}_c)(\boldsymbol{\mu}_c-\boldsymbol{\mu})^T+(\boldsymbol{\mu}_c-\boldsymbol{\mu})(\boldsymbol{x}_j-\boldsymbol{\mu}_c)^T \qquad (19)\end{aligned}$$

Let's look at Equation 19 piece by piece. The third and fourth terms goes to zero:

$$\sum_{j=1}^{m}(\boldsymbol{x}_j-\boldsymbol{\mu}_c)=\sum_{j=1}^{m}\boldsymbol{x}_j-m\boldsymbol{\mu}_c=\sum_{j=1}^{m}\boldsymbol{x}_j-\frac{m}{m}\sum_{j=1}^{m}\boldsymbol{x}_j=0 \qquad (20)$$

The within-scatter matrix as written in the text is invalid. It comes from the our first term in Equation 19:

$$\begin{aligned}\frac{1}{n}\sum_c\sum_{j=1}^{m}(\boldsymbol{x}_j-\boldsymbol{\mu}_c)(\boldsymbol{x}_j-\boldsymbol{\mu}_c)^T &= \sum_c\frac{m_c}{n}\frac{1}{m_c}\sum_{j=0}^{m}(\boldsymbol{x}_j-\boldsymbol{\mu}_c)(\boldsymbol{x}_j-\boldsymbol{\mu}_c)^T \\ &= \sum_c p_c\,\mathrm{cov}_c(\boldsymbol{x}_a,\boldsymbol{x}_b) \qquad (21)\end{aligned}$$

where the covariance is calculated as the total scatter matrix of class c ala Equation 17. This is where the $p_c = m_c/n$ term comes from in the code, but the expression is not exactly right in the text. I think there is a similar error in book equation 6.2. It comes from the second term of Equation 19 and it should read:

$$S_B=\frac{1}{n}\sum_c\sum_{j=1}^{m}(\boldsymbol{\mu}_c-\boldsymbol{\mu})(\boldsymbol{\mu}_c-\boldsymbol{\mu})^T=\frac{1}{n}\sum_c m_c(\boldsymbol{\mu}_c-\boldsymbol{\mu})(\boldsymbol{\mu}_c-\boldsymbol{\mu})^T=\sum_c p_c(\boldsymbol{\mu}_c-\boldsymbol{\mu})(\boldsymbol{\mu}_c-\boldsymbol{\mu})^T \qquad (22)$$