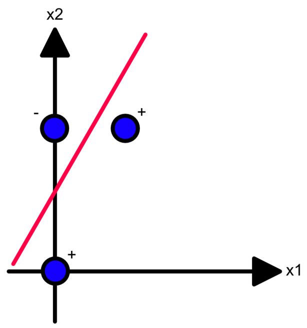
50.021 Artificial Intelligence
Theory Homework 1

Due: every Monday, 4PM before class starts

[Q1]. Find *any* separating hyperplane equation for these three sample points: $\boldsymbol{x}_1 = (0,0), y_1 = +1$, $\boldsymbol{x}_2 = (1,3), y_2 = +1$, and $\boldsymbol{x}_3 = (0,3), y_3 = -1$. Draw (by hand) or plot (using Python, see matplotlib) the result.

Solution:

Any line that separates the points into two classes is ok. Below is one example.



[Q2]. Find by hand the value of optimum weights $\hat{\boldsymbol{w}}$ and bias $\hat{b}$ using linear regression for the four following sample points: $\boldsymbol{x}_1 = (1,1), y_1 = +1$, $\boldsymbol{x}_2 = (2,2), y_2 = +1$, $\boldsymbol{x}_3 = (1,3), y_3 = -1$ $\boldsymbol{x}_4 = (2,3), y_4 = -1$. Show your working.

Solution:
We can begin by making the matrix X,

$$X = \begin{bmatrix} 1 & 1 & 1 \\ 2 & 2 & 1 \\ 1 & 3 & 1 \\ 2 & 3 & 1 \end{bmatrix}$$

and the vector $\boldsymbol{y} = [1\ 1\ -1\ -1]^T$. We know that the optimum weight is $\hat{\boldsymbol{w}} = [w_1\ w_2\ \hat{b}]^T = (X^T \cdot X)^{-1} X^T \cdot Y$. Substituting the values above,

$$\hat{\boldsymbol{w}} = \left(\begin{bmatrix} 1 & 2 & 1 & 2 \\ 1 & 2 & 3 & 3 \\ 1 & 1 & 1 & 1 \end{bmatrix} \cdot \begin{bmatrix} 1 & 1 & 1 \\ 2 & 2 & 1 \\ 1 & 3 & 1 \\ 2 & 3 & 1 \end{bmatrix} \right)^{-1} \begin{bmatrix} 1 & 2 & 1 & 2 \\ 1 & 2 & 3 & 3 \\ 1 & 1 & 1 & 1 \end{bmatrix} \cdot \begin{bmatrix} 1 \\ 1 \\ -1 \\ -1 \end{bmatrix}$$

$$X^T \cdot X = \begin{bmatrix} 10 & 14 & 6 \\ 14 & 23 & 9 \\ 6 & 9 & 4 \end{bmatrix}, \ (X^T \cdot X)^{-1} = \begin{bmatrix} 1.1 & -0.2 & -1.2 \\ -0.2 & 0.4 & -0.6 \\ -1.2 & -0.6 & 3.4 \end{bmatrix}$$

Hence,

$$\hat{\boldsymbol{w}} = \begin{bmatrix} 1.1 & -0.2 & -1.2 \\ -0.2 & 0.4 & -0.6 \\ -1.2 & -0.6 & 3.4 \end{bmatrix} \cdot \begin{bmatrix} 1 & 2 & 1 & 2 \\ 1 & 2 & 3 & 3 \\ 1 & 1 & 1 & 1 \end{bmatrix} \cdot \begin{bmatrix} 1 \\ 1 \\ -1 \\ -1 \end{bmatrix} = \begin{bmatrix} 0.6 \\ -1.2 \\ 1.8 \end{bmatrix}$$

[Q3]. In the lecture notes, we solve the objective function:

$$\hat{\boldsymbol{w}} = \text{argmin}_{\boldsymbol{w}} \sum_{i=1}^{n} (y_i - \boldsymbol{w} \cdot \boldsymbol{x}_i)^2 \tag{1}$$

by hand, and we have the analytical solution for optimum weights $\boldsymbol{w}*$ in linear regression,

$$\hat{\boldsymbol{w}} = (X^T \cdot X)^{-1} X^T \cdot Y$$

$$\boldsymbol{x}, \hat{\boldsymbol{w}} \in \mathbb{R}^d$$

Now suppose instead of using $\boldsymbol{x}_i$ directly, we want to use some basis function $\phi(\boldsymbol{x}_i)$ on each dataset i, and suppose that we use a slightly different squared error loss function than the lecture notes,

$$L(y, f(\boldsymbol{x})) = \frac{1}{2N}(y - f(\boldsymbol{x}))^2,$$

$$f(\boldsymbol{x}) = \phi(\boldsymbol{x}) \cdot \boldsymbol{w}$$

1. Rewrite the objective function in equation (1) using the new loss function $L(y, f(x))$ above.
2. Show that, even when the loss function is changed from whats shown in the lecture notes to the above, and although we apply basis function $\phi(\boldsymbol{x}_i)$, the solution for optimum weight $\hat{v}$ still takes the similar form,

$$\hat{\boldsymbol{w}} = (\Phi^T \cdot \Phi)^{-1} \Phi^T \cdot Y,$$

   where,

$$\Phi = \begin{bmatrix} \phi_1(\boldsymbol{x}_1) & \dots & \phi_d(\boldsymbol{x}_1) \\ \vdots & \ddots & \vdots \\ \phi_1(\boldsymbol{x}_N) & \dots & \phi_d(\boldsymbol{x}_N) \end{bmatrix},$$

   and d is number of dimensions of each sample $\boldsymbol{x}$, and N is the number of samples.

3. Show that if we define a function,

$$\mathcal{K}(\boldsymbol{x}_i, \boldsymbol{x}_r) = \phi(\boldsymbol{x}_i) \cdot \phi(\boldsymbol{x}_t),$$

   then $f(\boldsymbol{x})$ can be written only in terms of the function $\mathcal{K}$ above without the need to specify ϕ explicitly.
   Hint: Let $\boldsymbol{w} = \Phi^T \boldsymbol{v}$, where v is some new parameter vector that you now maximise.

4. Write down the analytical form of the optimum parameter for $L(y, f(\boldsymbol{x}))$ using your new expression of $f(\boldsymbol{x})$ (that now contains $\mathcal{K}$) in part (3). This can be done by reusing some of your answer in part (2).

Solution:

1. The function we are going to minimise is the sum of the loss functions over the entire training samples,

$$\begin{aligned} \hat{\boldsymbol{w}} &= \text{argmin}_{\boldsymbol{w}} \sum_{i=1}^{N} \frac{1}{2N}(y_i - f(\boldsymbol{x}_i))^2 \\ &= \text{argmin}_{\boldsymbol{w}} \sum_{i=1}^{N} \frac{1}{2N}(y_i - \phi(\boldsymbol{x}_i) \cdot \boldsymbol{w})^2 \end{aligned}$$

In vectorised form, it is,

$$\hat{\boldsymbol{w}} = \text{argmin}_{\boldsymbol{w}} \frac{1}{2N} \|\boldsymbol{y} - \Phi \boldsymbol{w}\|^2,$$

where $\boldsymbol{y} = (y_1, ..., y_N)^T$, and $\boldsymbol{w} = (w_1, ..., w_d)^T$.

2. We can differentiate the equation in part 1 with respect to $\boldsymbol{w}$ and set it to zero. In short, we can use the vector derivative,

$$\nabla_{\boldsymbol{w}} \left[\frac{1}{2N} \|\boldsymbol{y} - \Phi \boldsymbol{w}\|^2 \right] = 0$$

Hence,

$$\begin{aligned}
0 &= \frac{1}{2N} \nabla_{\boldsymbol{w}} \left[(\boldsymbol{y} - \Phi \boldsymbol{w})^T (\boldsymbol{y} - \Phi \boldsymbol{w}) \right] \\
&= \frac{1}{2N} \nabla_{\boldsymbol{w}} \left[\boldsymbol{y}^T \boldsymbol{y} - 2\boldsymbol{y}^T \Phi \boldsymbol{w} + \boldsymbol{w}^T \Phi^T \Phi \boldsymbol{w} \right] \\
&= \frac{1}{2N} (-2\Phi^T \boldsymbol{y} + 2\Phi^T \Phi \boldsymbol{w}) \\
&\text{(Google "Matrix Calculus" if you're lost on how to get the above expression)} \\
&= -\Phi^T \boldsymbol{y} + \Phi^T \Phi \boldsymbol{w} \\
\therefore \hat{w} &= (\Phi^T \Phi)^{-1} \Phi^T \boldsymbol{y}
\end{aligned}$$

3. The key is to first realise that the possible values of $\boldsymbol{w}$ lies in the span of data points $\boldsymbol{x}_i \; \forall \; i = 1, ..., N$. In other words, the optimal $\hat{w}$ depends on *all* of the training data,

$$\boldsymbol{w} = \Phi^T \boldsymbol{v},$$

for some vector $\boldsymbol{v} = (v_1, ..., v_N), \;\; \boldsymbol{v} \in \mathbb{R}^N$.
We know that $f(\boldsymbol{x}) = \phi(\boldsymbol{x}) \cdot \boldsymbol{w}$. Plugging in the new expression for $\boldsymbol{w}$ above,

$$\begin{aligned}
f(\boldsymbol{x}) &= \phi(\boldsymbol{x}) \cdot (\Phi^T \boldsymbol{v}) \\
&= \phi(\boldsymbol{x}) \cdot \left(\sum_{i=1}^{N} \phi(\boldsymbol{x}_i) \; v_i \right) \\
&= \sum_{i=1}^{N} (\phi(\boldsymbol{x}) \cdot \phi(\boldsymbol{x}_i)) \; v_i \\
&= \sum_{i=1}^{N} \mathcal{K}(\boldsymbol{x}, \boldsymbol{x}_i) \; v_i
\end{aligned}$$

This is called the *dual* form of linear regression. It is more efficient to compute $\mathcal{K}$ than to compute ϕ, because ϕ. In the literature, $\mathcal{K}$ is known as a *kernel*.

4. Using results from part (2),

$$\begin{aligned}
\Phi^T \boldsymbol{y} &= \Phi^T \Phi \; \hat{\boldsymbol{w}}, \\
\Phi^T \boldsymbol{y} &= \Phi^T \Phi \; (\Phi^T \hat{v}), \\
\therefore \boldsymbol{y} &= \Phi \Phi^T \hat{v}, \\
\hat{v} &= (\Phi \Phi^T)^{-1} \boldsymbol{y}, \\
\hat{v} &= (\mathbf{K})^{-1} \boldsymbol{y},
\end{aligned}$$

where,

$$\mathbf{K} = \begin{bmatrix} \mathcal{K}(\boldsymbol{x}_1, \boldsymbol{x}_1) & ... & \mathcal{K}(\boldsymbol{x}_1, \boldsymbol{x}_N) \\ \vdots & \ddots & \vdots \\ \mathcal{K}(\boldsymbol{x}_N, \boldsymbol{x}_1) & ... & \mathcal{K}(\boldsymbol{x}_N, \boldsymbol{x}_N) \end{bmatrix}$$