1. Linearization to help classification

As was discussed in lecture, the naive straightforward way of picking the decision boundary (by looking at the mean example of each category and drawing the perpendicular bisector) is not always the best.

Consider trying to classify a set of measurements $\vec{x}_i$ with given labels ℓ_i. For the binary case of interest here, we will think of the labels as being "+" and "-" and fold our threshold implicitly into the weights by augmenting the constant "1" in the first position of each $\vec{x}_i$. Now, the classification rule becomes simple. **We want to learn a vector of weights $\vec{w}$ so that we can deem any point with $\vec{x}_i^\top \vec{w} > 0$ as being a member of the "+" category and anything with $\vec{x}_i^\top \vec{w} < 0$ as being a member of the "-" category.**

The way that we will do this, is to do a minimization in the spirit of least squares. Except, instead of necessarily using some sort of squared loss function, we will just consider a generic cost function that can depend on the label and the prediction for the point. For the i-th data point in our training data, we will incur a cost $c(\vec{x}_i^\top \vec{w}, \ell_i)$ for a total cost that we want to minimize as:

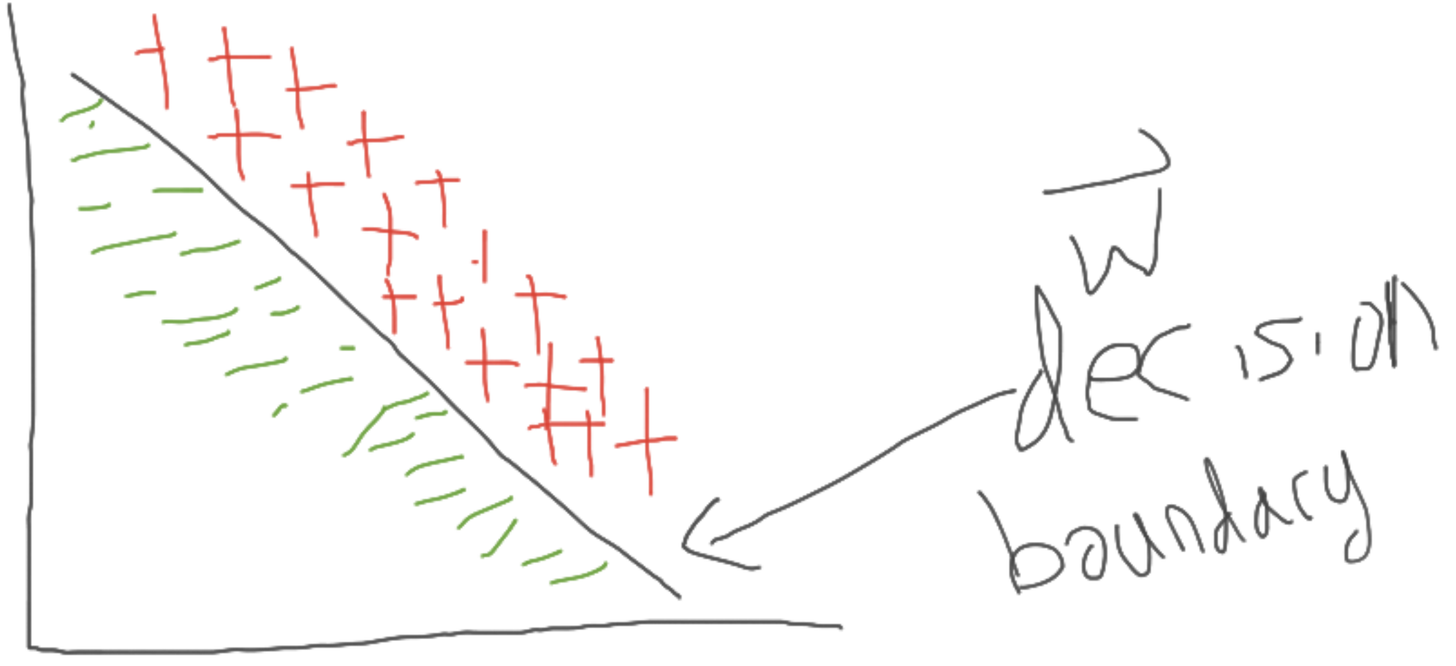
$$\arg\min_{\vec{w}} \quad c_{total}(\vec{w}) = \sum_{i=1}^{m} c(\vec{x}_i^\top \vec{w}, \ell_i) \tag{1}$$

Goal: Learn $\vec{w}$ s.t.

$x_i \in X^+ \Rightarrow \vec{x}_i^T \vec{w} > 0$

$x_i \in X^- \Rightarrow \vec{x}_i^T \vec{w} < 0$

How?



Because this can be a nonlinear function, our goal is to solve this iteratively as a sequence of least-squares problems that we know how to solve.

Consider the following algorithm:

1: $\vec{w} = \vec{0}$ ▷ Initialize the weights to $\vec{0}$
2: **while** Not done **do** ▷ Iterate towards solution
3: Compute $\vec{w}^\top \vec{x}_i$ ▷ Generate current estimated labels
4: Compute $\frac{d}{d\vec{w}} c(\vec{w}^\top \vec{x}_i, \ell_i)$ ▷ Generate derivatives with respect to $\vec{w}$ of the cost for update step
5: Compute $\frac{d^2}{d\vec{w}^2} c(\vec{w}^\top \vec{x}_i, \ell_i)$ ▷ Generate second derivatives of the cost for update step
6: $\delta\vec{w} = LeastSquares(\cdot, \cdot)$ ▷ We will derive what to call least squares on
7: $\vec{w} = \vec{w} + \delta\vec{w}$ ▷ Update parameters
8: **end while**
9: Return $\vec{w}$

The key step above is figuring out with what arguments to call LeastSquares while only having the labels ℓ_i and the points $\vec{x}_i$.

(a) Now, suppose we wanted to approximate the cost for each data point

$$c_i(w) = c(x_i^\top w, \ell_i) \tag{3}$$

where

$$\vec{w} = \begin{bmatrix} w[1] \\ \vdots \\ w[n] \end{bmatrix}$$

in the neighborhood of a weight vector $\vec{w}_*$. Our goal is to write out the first-order expression for approximating the cost function $c_i(w_* + \delta w)$. This should be something in vector/matrix form like you have seen for the approximation of nonlinear systems by linear systems. We don't want to take any second derivatives just yet — only first derivatives. We have outlined a skeleton for the derivation with some parts missing. Follow the guidelines in each sub-section.

i) Comparing to eq. (2), we know that $c_i(w_* + \delta\vec{w}) \approx c_i(w_*) + D_{\vec{w}}c_i(w_*)\delta\vec{w}$. **Write out the vector form of $D_{\vec{w}}c_i(w_*)$ in terms of the partial derivatives.**.

ii) **Write out the partial derivatives of $c_i(\vec{w})$ with respect to $w[g]$, the g^{th} component of $\vec{w}$.** *(HINT: Use the linearity of derivatives and sums to compute the partial derivatives with respect to each of the $w[g]$ terms. Don't forget the chain rule and the fact that $x_i^\top w = \sum_{j=1}^n x_i[j]w[j] = x_i[g]w[g] + \sum_{j\neq g} x_i[j]w[j]$.)*

iii) With what you had above, **can you fill in the missing part to express the row vector $D_{\vec{w}}c_i(\vec{w})$?**

$$D_{\vec{w}}c_i(\vec{w}) = c'(\vec{x}_i^\top \vec{w}, \ell_i)$$

i) $\dfrac{\partial c_i}{\partial \vec{w}}\Big|_{\vec{w}_*} = \left[\dfrac{\partial c_i}{\partial w_1}\Big|_{w_*} \cdots \dfrac{\partial c_i}{\partial w_n}\Big|_{w_*}\right]$

iii) $c'(\vec{x}_i^\top \vec{w}, \ell_i)\begin{bmatrix} x_i[1] & \cdots & x_i[n] \end{bmatrix}$

$= \boxed{c'(\vec{x}_i^\top \vec{w}, \ell_i)\vec{x}_i^\top}$

ii) $\dfrac{\partial c_i(\vec{w})}{\partial w_g} = \dfrac{\partial c(\vec{x}_i^\top \vec{w}, \ell_i)}{\partial w_g} = \dfrac{\partial c(\vec{x}_i^\top \vec{w}, \ell_i)}{\partial \vec{x}_i^\top \vec{w}} \dfrac{\partial \vec{x}_i^\top \vec{w}}{\partial w_g}$

$= c'(\vec{x}_i^\top \vec{w}, \ell_i) x_i[g]$

(b) Now, we want a better approximation that includes second derivatives. For a general function, we would look for

$$f(\vec{x}_0 + \vec{\delta x}) \approx f(\vec{x}_0) + f'(\vec{x}_0)\vec{\delta x} + \frac{1}{2}\vec{\delta x}^\top f''(\vec{x}_0)\vec{\delta x} \tag{4}$$

where $f'(\vec{x}_0)$ is an appropriate row vector and, as you've seen in the note, $f''(\vec{x}_0)$ is called Hessian that represents the second derivatives.

i) Comparing to eq. (4), we know that

$$c_i(\vec{w}_* + \vec{\delta w}) \approx c_i(\vec{w}_*) + D_{\vec{w}}c_i(\vec{w}_*)\vec{\delta w} + \frac{1}{2}\vec{\delta w}^\top H_{\vec{w}}c_i(\vec{w}_*)\vec{\delta w}$$

Write out the matrix form of $H_{\vec{w}}c_i(\vec{w}_*)$.

$$\frac{\partial^2 c_i}{\partial \vec{w}^2} = \begin{bmatrix} \frac{\partial^2 c_i}{\partial w_1^2} & \cdots & \frac{\partial^2 c_i}{\partial w_1 \partial w_n} \\ \vdots & & \\ \frac{\partial^2 c_i}{\partial w_n \partial w_1} & \cdots & \frac{\partial^2 c_i}{\partial w_n^2} \end{bmatrix}$$

$n \times n$ — $|c_n(\vec{w})$

ii) **Take the second derivatives of the cost $c_i(\vec{w})$, i.e. solve for $\frac{\partial^2 c_i(\vec{w})}{\partial w[g]\partial w[h]}$.**

(HINT: You should use the answer to part (a) and just take another derivative. Once again, use the linearity of derivatives and sums to compute the partial derivatives with respect to each of the $w[h]$ terms. This will give you $\frac{\partial^2}{\partial w[g]\partial w[h]}$. Don't forget the chain rule and again use the fact that $\vec{x}_i^\top \vec{w} = \sum_{j=1}^n x_i[j]w[j] = x_i[h]w[h] + \sum_{j\neq h} x_i[j]w[j]$.)

$$\frac{\partial^2 C_i(\vec{w})}{\partial w_h \partial w_g} = \frac{\partial}{\partial w_h}\left(\frac{\partial}{\partial w_g} C_i(\vec{w})\right) = \frac{\partial}{\partial w_h} C'(\vec{x}_i^\top \vec{w}, l_i)\, x_i[g]$$

$$\frac{\partial C'(\vec{x}_i^\top \vec{w}, l_i)\, x_i[g]}{\partial \vec{x}_i^\top \vec{w}} \frac{\partial \vec{x}_i^\top \vec{w}}{\partial w_h} = C''(\vec{x}_i^\top \vec{w}, l_i) x_i[g] x_i[h]$$

$C'(f(\vec{w}), l_i),\ f(\vec{w}) = \vec{x}_i^\top \vec{w}$

iii) The expression in part (ii) is for the $[g, h]$-th component of the second derivative. $\frac{1}{2}$ times this times $\delta\vec{w}[g]$ times $\delta\vec{w}[h]$ would give us that component's contribution to the second-derivative term in the approximation, and we have to sum this up over all g and h to get the total contribution of the second-derivative term in the approximation. Now, we want to group terms to restructure this into matrix-vector form by utilizing the outer-product form of matrix multiplication. **What should the space in the following expression be filled with?**

$$H_{\vec{w}} c_i(\vec{w}) = c''(\vec{x}_i^\top \vec{w}, \ell_i) \underline{\qquad} \vec{x}_1 \vec{x}_1^\top$$

$$c''(\vec{x}_1^\top \vec{w}, \ell_1) \begin{bmatrix} x_1[1]x_1[1] & \cdots & x_1[1]x_1[n] \\ \vdots & & \vdots \\ x_1[1]x_1[n] & \cdots & x_1[n]x_1[n] \end{bmatrix} = c''(\vec{x}_1^\top \vec{w}, \ell_1) \vec{x}_1 \vec{x}_1^\top$$

(c) Now we have successfully expressed the second order approximation of $c_i(\vec{w}_* + \delta\vec{w})$. Since we eventually want to minimize the total cost $c_{total}(\vec{w}) = \sum_{i=1}^{m} c_i(\vec{w})$, **can you write out the second order approximation of $c_{total}(\vec{w}_* + \delta\vec{w})$ using results from (a) and (b)?**

$$c_i(\vec{w}_t + \delta\vec{w}) \simeq c_i(\vec{w}_t) + \frac{\partial c_i(\vec{w})}{\partial \vec{w}}(\vec{w}_t)\,\delta\vec{w} + \frac{1}{2}\delta\vec{w}^T \frac{\partial^2 c_i(\vec{w})}{\partial \vec{w}}(\vec{w}_t)\,\delta\vec{w}$$

$$= c_i(\vec{w}_t) + c'(\vec{x}_i^T\vec{w}, l_i)\,x_i^T\delta\vec{w} + \frac{1}{2}\delta\vec{w}^T c''(\vec{x}_i^T\vec{w}, l_i)\,\vec{x}_i\vec{x}_i^T\delta\vec{w}$$

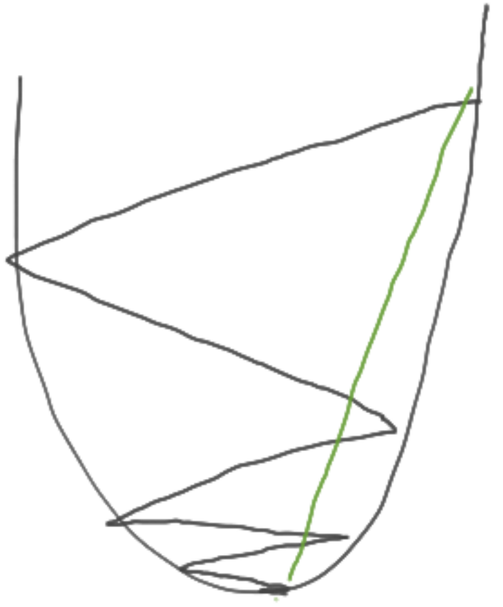
$$c_{total}(\vec{w}+\delta\vec{w}) = \sum_{i=1}^{m} c_i(\vec{w}_t + \delta\vec{w})$$

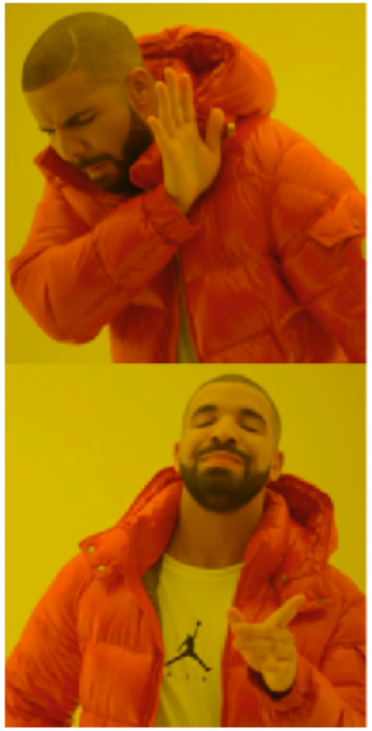
(d) In this part we explore "Newton's method", as an alternative to the iterative least squares formulation above. Recall that for a differentiable function $f(\vec{w})$, we can set $D_{\vec{w}} f(\vec{w}^\star) = \vec{0}^\top$ to find a minimum. Consider a linearization of the derivative of our cost function, $D_{\vec{w}} c^\top$ at the point $\vec{w}$. **Using the linearlization for $D_{\vec{w}} c(\vec{w})^\top$ and the fact that $D_{\vec{w}} c\Big|_{\vec{w}^\star} = \vec{0}^\top$ to derive an update.**

$$\frac{\partial C_1(\vec{w}_T)}{\partial \vec{w}} = \frac{\partial C_1(\vec{w}_0)}{\partial \vec{w}} + \frac{\partial^2 C_1(\vec{w}_0)}{\partial \vec{w}^2}(\vec{w}_T - \vec{w}_0)$$

$$0 = \quad \| \quad$$

$$\vec{w}_T = \vec{w}_0 - \left(\frac{\partial^2 L_1(\vec{w}_0)}{\partial \vec{w}^2}\right)^{-1} \frac{\partial C_1(w_0)}{\partial \vec{w}}$$





$\vec{x}_i^T \vec{w} < 0$

$\vec{x}_i^T \vec{w} > 0$

Feedback: https://tinyurl.com/manav16b