Cost Functions for Half-space Models

Chris Cornwell

November 3, 2025

Outline

Outline

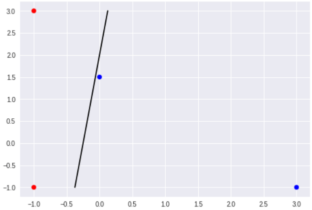
Outline

Returning to the “Simple Example” for Perceptron algorithm

Example in $\mathbb{R}^2$, with $P = 4$ points.

$$\text{x: } \begin{bmatrix} -1 & 3 \\ -1 & -1 \\ 3 & -1 \\ 0 & 1.5 \end{bmatrix} \qquad \text{y: } \begin{bmatrix} -1 \\ -1 \\ 1 \\ 1 \end{bmatrix}$$

Final $\tilde{\mathbf{w}} = \begin{bmatrix} 1 \\ 4 \\ -0.5 \end{bmatrix}$; hyperplane in $\mathbb{R}^2$ (a line) shown below.



What if this data were part of a sample, which has noise? What about the point near the hyperplane?

Getting a Buffer – a "margin"

Say that $S = \{\mathbf{x}_i, y_i\}_{i=1}^{P}$ (with $y_i = \pm 1$) is linearly separable. Instead of simply trying to find a hyperplane that successfully separates the data, try to find one that has some distance from points on each side.

Getting a Buffer – a "margin"

Say that $S = \{\mathbf{x}_i, y_i\}_{i=1}^{P}$ (with $y_i = \pm 1$) is linearly separable. Instead of simply trying to find a hyperplane that successfully separates the data, try to find one that has some distance from points on each side.

Can express this as wanting no points between two parallel hyperplanes – between the set of $\mathbf{x}$ where $\tilde{\mathbf{x}}^\top \tilde{\mathbf{w}} = 1$ and where $\tilde{\mathbf{x}}^\top \tilde{\mathbf{w}} = -1$.

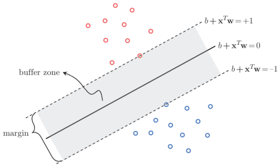


Figure: Taken from Figure 4.4 in textbook, p. 79. (Note: they used blue for the -1 label.)

This is equivalent to wanting, for all $i = 1, \ldots, P$, $\tilde{\mathbf{x}}_i^\top \tilde{\mathbf{w}} \geq 1$ if $y_i = 1$ and $\tilde{\mathbf{x}}_i^\top \tilde{\mathbf{w}} \leq 1$ if $y_i = -1$. That is, we want $y_i\left(\tilde{\mathbf{x}}_i^\top \tilde{\mathbf{w}}\right) \geq 1$.

Recall, S being linearly separable implies that there must be a $\tilde{\mathbf{w}}$ that achieves this.

Cost function for margin halfspace model

As we are looking for $\tilde{\mathbf{w}}$ so that $y_i(\tilde{\mathbf{x}}_i^\top \tilde{\mathbf{w}}) \geq 1$ for all $i = 1, \ldots, P$, it makes sense to have $(\mathbf{x}_i, y_i)$ contribute 0 to our cost function if this inequality works for that i. So, we might set

$$g_3(\tilde{\mathbf{w}}) = \sum_{i=1}^{P} \max\left(0, 1 - y_i(\tilde{\mathbf{x}}_i^\top \tilde{\mathbf{w}})\right).$$

(Note that, when not using a margin and using the max cost function, rather than softmax, there was a trivial minimizer of $\tilde{\mathbf{w}} = \vec{\mathbf{0}}$, which we do not want to use.)

Softmax approximation for margin halfspace: As before, if we want a smooth function we can replace the max in the cost function with softmax:

$$(\textit{Alternate}) \quad g_3(\tilde{\mathbf{w}}) = \sum_{i=1}^{P} \text{softmax}\left(0, 1 - y_i(\tilde{\mathbf{x}}_i^\top \tilde{\mathbf{w}})\right).$$

Squared max for margin halfspace: Another option for a

$$g_4(\tilde{\mathbf{w}}) = \sum_{i=1}^{P} \left(\max\left(0, 1 - y_i(\tilde{\mathbf{x}}_i^\top \tilde{\mathbf{w}})\right)\right)^2.$$

Outline

The Counting Cost Function

Suppose that we have found our stationary point (or approximation to it), $\tilde{\mathbf{w}}^{\star} = \left(b^{\star}, \mathbf{w}^{\star}\right)$. Given a "new" data point, $\mathbf{x}_{new}$, the halfspace model that we would use to predict labels on data is: make predicted $y_{new} = 1$ if the dot product $\tilde{\mathbf{x}}_{new} \cdot \tilde{\mathbf{w}}^{\star} = b^{\star} + \mathbf{x}_{new} \cdot \mathbf{w}^{\star}$ is positive; alternatively, when that dot product is negative, predict $y_{new} = -1$.
Using the "sign" function:

$$f\left(\mathbf{x}_{new}\right) = \text{sign}\left(\tilde{\mathbf{x}}_{new} \cdot \tilde{\mathbf{w}}^{\star}\right).$$

To measure the accuracy of this model on our given data $S = \left\{\left(\mathbf{x}_i, y_i\right)\right\}$, compute how many this prediction function gets to agree with y_i; that is,

$$\sum_{i=1}^{P} \max\left(0, -y_i f\left(\mathbf{x}_i\right)\right) = \sum_{i=1}^{P} \max\left(0, -y_i\left(\tilde{\mathbf{x}}_{new} \cdot \tilde{\mathbf{w}}^{\star}\right)\right).$$

Outline