# Tutorial 3

CS 337 Artificial Intelligence & Machine Learning, Autumn 2021

September, 2021

**Problem 1.** Given a set of data points $\{x_n\}$ (for $n = 1...N$) in some $d$ dimensional space ($\Re^d$), we can define the convex hull to be the set of all points $x$ given by

$$x = \sum_n \alpha_n x_n$$

where $\alpha_n \geq 0$ and $\sum_n \alpha_n = 1$. Consider a set of points $\{x_m\}$ together with its convex hull and a second set of points $\{y_m\}$ together with its corresponding convex hull. By definition, the two sets of points will be linearly separable if there exists a vector $\hat{w}$ and a scalar $w_0$ such that $\mathbf{w}^T x_n + w_0 > 0$ for all $x_n$ and $\mathbf{w}^T y_m + w_0 < 0$ for all $y_m$. Show that if the convex hulls of $\{x_n\}$ and $\{y_n\}$ intersect, the two sets of points cannot be linearly separable.

**Optional and advanced**: Also prove the converse statement that if they are not linearly separable, their convex hulls must intersect.

**Solution:** If the convex hulls intersect, there must be at least one point in common between $\{x_n\}$ and $\{y_m\}$. Let's call that point $xy$. Since $xy$ belongs to both convex hulls, there must be a set of $\alpha_n$ and $\beta_m$ that give rise to $xy$ which can be expressed equivalently as two convex expressions of $\{x_n\}$ and $\{y_m\}$ as $xy = \sum_n \alpha_n x_n = \sum_m \beta_m y_m$. Thus, the linear discriminant for $xy$ can now also be written in two separate but equivalent ways.

Now, the linear classification function $f(\mathbf{x})$ can be therefore written in two equivalent ways as

$$f(xy) = \sum_n \alpha_n \left(\mathbf{w}^T x^n + w_0\right) \tag{1}$$

$$f(xy) = \sum_m \beta_m \left(\mathbf{w}^T y^m + w_0\right) \tag{2}$$

If we had linear separability, we should have had

$$\mathbf{w}^T x^n + w_0 > 0 \tag{3}$$

$$\mathbf{w}^T y^m + w_0 < 0 \tag{4}$$

Based on equations 1 and 3, $f(xy) > 0$. Whereas based on equations 2 and 4, $f(xy) < 0$. These are totally contradictory. Hence it is impossible that when the convex hulls intersect, the points are linearly separable. **That is, if convex hulls intersect, the points cannot be linearly separable. Which is equivalent to the contrapositive that if we have linear separability, the convex hulls cannot intersect.**

To prove the converse, we prove the inverse (that is, the contrapositive of the converse) that - if their convex hulls do not intersect, the points must be linearly separable. This is a more involved proof which can be seen in the form of the **Separating Hyperplane theorem** based on slides 35 and 39 of these slides from my on convex optimization: `https://www.cse.iitb.ac.in/~cs709/notes/enotes/5-31-07-2018-primal-dual-descripti` `pdf` **Hence, if we have linear separability, the convex hulls cannot intersect.**

**Problem 2.** Let $X$ have a uniform distribution over integers in an interval $[0, \theta)$:

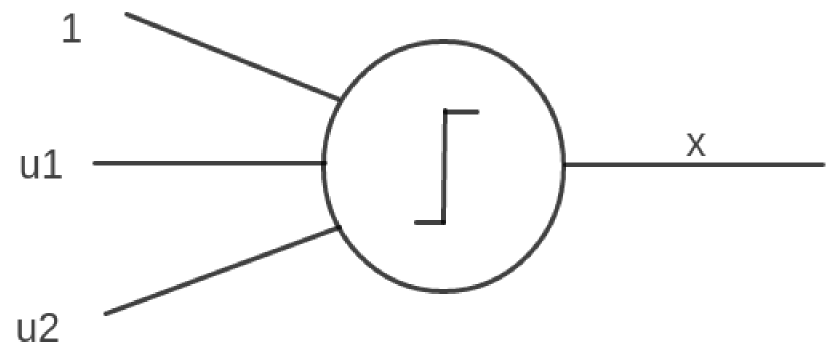
$$p(X = x; \theta) = \begin{cases} \frac{1}{\theta} & \text{if } 0 \leq x < \theta \\ 0 & \text{otherwise} \end{cases}$$

Suppose $n$ samples $x_1, \ldots, x_n$ are drawn i.i.d based on $p(x; \theta)$. What is the MLE estimate of $\theta$?

**Problem 3. Computing power of perceptrons.** Perceptrons can only separate Linearly separable data as discussed in class. Given n variables we can have $2^{2^n}$ boolean functions, but not all of these can be represented by a perceptron. For example when n=2 the XOR and XNOR cannot be represented by a perceptron. Given n boolean variables how many of $2^{2^n}$ boolean functions can be represented by a perceptron?

**Problem 4.** Consider a perceptron for which $u \in R^2$ and

$$f(a) = \begin{cases} 1 & a > 0 \\ 0 & a = 0 \\ -1 & a < 0 \end{cases}$$




Let the desired output be 1 when elements of class A = {(1,2),(2,4),(3,3),(4,4)} is applied as input and let it be -1 for the class B = {(0,0),(2,3),(3,0),(4,2)}. Let the initial connection weights $w_0(0) = +1, w_1(0) = -2, w_2(0) = +1$ and learning rate be h = 0.5.

This perceptron is to be trained by perceptron convergence procedure, for which the weight update formula is $w(t+1) = w(t) + \eta(y^k - x^k(t))u^k$

1. (a) Mark the elements belonging to class A with x and those belonging to class B with o on input space.
   (b) Draw the line represented by the perceptron considering the initial connection weights w(0).
   (c) Find out the regions for which the perceptron output is +1 and –1
   (d) Which elements of A and B are correctly classified, which elements are misclassified and which are unclassified?
2. If u=(4,4) is applied at input, what will be w(1) ?
3. Repeat a) considering w(1).
4. If u=(4.2) is then applied at input, what will be w(2)?

5. Repeat 1) considering w(2).

6. Do you expect the perceptron convergence procedure to terminate? Why?

**Solution:** Would like students to present their solutions.

**Problem 5.** In the class, we discussed the probabilistic binary (class) logistic regression classifier. How will you extend logistic regression probabilistic model to multiple (say $K$) classes? Are their different ways of extending? What is the intuition behind each? Discuss and contrast advantages/disadvantages in each.

**Solution:** One might suggest handling multi-class ($K$) classification via $K$ one-vs-rest probabilistic classifiers. But there is no obvious probabilistic semantics associated with such a classifier (question asked for a probabilistic MODEL for multiple classes).

Basic idea is that each class $c$ can have a different weight vector $[w_{c,1}, w_{c,2}, \ldots, w_{c,k}, \ldots, w_{c,K}]$ .

**Extension to multi-class logistic**

1. Each class $c = 1, 2, \ldots, K-1$ can have a different weight vector $[\mathbf{w}_{c,1}, \mathbf{w}_{c,2}, \ldots, \mathbf{w}_{c,k}, \ldots, \mathbf{w}_{c,K-1}]$ and

$$p(Y = c|\phi(\mathbf{x})) = \frac{e^{-(\mathbf{w}_c)^T\phi(\mathbf{x})}}{1 + \sum_{k=1}^{K-1} e^{-(\mathbf{w}_k)^T\phi(\mathbf{x})}}$$

for $c = 1, \ldots, K-1$ so that

$$p(Y = K|\phi(\mathbf{x})) = \frac{1}{1 + \sum_{k=1}^{K-1} e^{-(\mathbf{w}_k)^T\phi(\mathbf{x})}}$$

**Alternative (equivalent) extension to multi-class logistic**

1. Each class $c = 1, 2, \ldots, K$ can have a different weight vector $[\mathbf{w}_{c,1}, \mathbf{w}_{c,2} \ldots \mathbf{w}_{c,p}]$ and

$$p(Y = c|\phi(\mathbf{x})) = \frac{e^{-(\mathbf{w}_c)^T\phi(\mathbf{x})}}{\sum_{k=1}^{K} e^{-(\mathbf{w}_k)^T\phi(\mathbf{x})}}$$

for $c = 1, \ldots, K$.

This function is also the called the **softmax**[1] function.

[1] https://en.wikipedia.org/wiki/Softmax_function