COMP9417 - Machine Learning
Tutorial: Classification

Question 1. Distance between parallel hyperplanes

When constructing linear classifiers, a common calculation that comes up is to compute the distance between two parallel hyperplanes. Consider two parallel hyperplanes: $H_1 = \{x \in \mathbb{R}^n : w^T x = a\}$ and $H_2 = \{x \in \mathbb{R}^n : w^T x = b\}$. Show that the distance between H_1 and H_2 is given by $\frac{|b-a|}{\|w\|_2}$.

Hint: draw a picture.

Question 2 (Perceptron Training & Capacity)

(a) Consider the following training data:

x_1	x_2	y
-2	-1	-1
2	-1	1
1	1	1
-1	-1	-1
3	2	1

Apply the Perceptron Learning Algorithm with starting values $w_0 = 5$, $w_1 = 1$ and $w_2 = 1$, and a learning rate $\eta = 0.4$. Be sure to cycle through the training data in the same order that they are presented in the table. Present your results in table form:

Iteration	$\langle w, x\rangle$	$y\langle w, x\rangle$	w

(b) Consider the following three logical functions:

1. $A \wedge \neg B$
2. $\neg A \vee B$
3. $(A \vee B) \wedge (\neg A \vee \neg B)$

Which of these functions can a perceptron learn? Explain.

Question 3. Binary Logistic Regression, two perspectives

Recall from previous weeks that we can view least squares regression as a purely optimisation based problem (minimising MSE), or as a statistical problem (using MLE). We now discuss two perspectives of the Binary Logistic Regression problem. In this problem, we are given a dataset $D = \{(x_i, y_i)\}_{i=1}^n$ where the x_i's represent the feature vectors, just as in linear regression, but the y_i's are now binary. The goal is to model our output as a probability that a particular data point belongs to one of two classes. We will denote this predicted probability by

$$P(y = 1|x) = p(x)$$

and we model it as

$$\hat{p}(x) = \sigma(\hat{w}^T x), \qquad \sigma(z) = \frac{1}{1+e^{-z}},$$

where $\hat{w}$ is our estimated weight vector. We can then construct a classifier by assigning the class that has the largest probability, i.e.:

$$\hat{y} = \arg\max_{k=0,1} P(\hat{y} = k|x) = \begin{cases} 1 & \text{if} \quad \sigma(\hat{w}^T x) \geq 0.5 \\ 0 & \text{otherwise} \end{cases}$$

note: do not confuse the function $\sigma(z)$ with the parameter σ which typically denotes the standard deviation.

(a) What is the role of the logistic sigmoid function $\sigma()$ in the logistic regression formulation? Why are we not able to simply use linear regression here? (a plot of $\sigma(z)$ may be helpful here).

(b) We first consider the statistical view of logistic regression. Recall in the statistical view of linear regression, we assumed that $y|x \sim N(x^T\beta^*, \sigma^2)$. Here, we are working with binary valued random variables and so we assume that

$$y|x \sim \text{Bernoulli}(p^*), \qquad p^* = \sigma(x^T w^*)$$

where $p^* = \sigma(x^T w^*)$ is the true unknown probability of a response belonging to class 1, and we assume this is controlled by some true weight vector w^*. Write down the log-likelihood of the data D (as a function of w), and further, write down the MLE objective (but do not try to solve it).

(c) An alternative approach to the logistic regression problem is to view it purely from the optimisation perspective. This requires us to pick a loss function and solve for the corresponding minimizer. Write down the MSE objective for logistic regression and discuss whether you think this loss is appropriate.

(d) Consider the following problem: you are given two discrete probability distributions, P and Q, and you are asked to quantify how far Q is from P. This is a very common task in statistics and information theory. The most common way to measure the discrepancy between the two is to compute the Kullback-Liebler (KL) divergence, also known as the relative entropy, which is defined by:

$$D_{\text{KL}}(P\|Q) = \sum_{x\in\mathcal{X}} P(x) \ln \frac{P(x)}{Q(x)},$$

where we are summing over all of the possible values of the underlying random variable. A good way to think of this is that we have a true distribution P, an estimate Q, and we are trying to figure out how bad our estimate is. Write down the KL divergence between two bernoulli distributions $P = \text{Bernoulli}(p)$ and $Q = \text{Bernoulli}(q)$.

(e) Continuing with the optimisation based view: In our set-up, one way to quantify the discrepancy between our prediction $\hat{p}_i$ and the true label y_i is to look at the KL divergence between the two bernoulli distributions $P_i = \text{Bernoulli}(y_i)$ and $Q_i = \text{Bernoulli}(\hat{p}_i)$. Use this to write down an appropriate minimization for the logistic regression problem.

(f) In logistic regression (and other binary classification problems), we commonly use the cross-entropy loss, defined by

$$\mathcal{L}_{XE}(a,b) = -a \ln \frac{a}{b} - (1-a)\ln \frac{1-a}{1-b}.$$

Using your result from the previous part, discuss why the XE loss is a good choice, and draw a connection between the statistical and optimisation views of logistic regression.

Question 4. Numerically solving the logistic regression problem

In the previous problem, we show that in order to solve the logistic regression problem, we must solve the following optimisation:

$$\begin{aligned} \hat{w} &= \arg\min_w \mathcal{L}(w) \\ &= \arg\min_w -\left[\sum_{i=1}^n y_i \ln p_i + (1-y_i)\ln(1-p_i)\right], \end{aligned}$$

where

$$p_i = \sigma(w^T x_i) = \frac{1}{1+e^{-w^T x_i}}.$$

Unfortunately in this case, we cannot solve for $\hat{w}$ in closed form. In other words, we cannot simply take derivatives, equate to zero and solve to get a nice solution as in the linear regression case. Instead, we must rely on numerical techniques such as gradient descent. In this question, we will work through and derive the gradient descent updates for the logistic regression problem.

(a) We will need to take derivatives in order to do any form of gradient descent. Show that

$$\sigma'(z) = \sigma(z)(1-\sigma(z)).$$

Then use this result to show that

$$\frac{dp_i}{dw} = p_i(1-p_i)x_i,$$

where $p_i = \sigma(w^T x_i)$.

(b) Use the previous result to show that

$$\frac{d\ln p_i}{dw} = (1-p_i)x_i.$$

(c) Using the results of the previous parts, compute

$$\frac{d\mathcal{L}(w)}{dw}$$

and write down the gradient descent update for w with step size η.

(d) A convex function does not have any local minima, and so we are guaranteed to converge to a global minimum when doing gradient descent on a convex function, regardless of our initialisation $w^{(0)}$. Prove that $\mathcal{L}(w)$ is convex.