Exercise 1. (Train vs. test datasets)

You want to predict whether or not a product fails based on historical data on the amount of the different materials used to make the product. You have a set of 1,000 data points, where each data point contains the amount of the 5 different materials ($x^i \in \mathbb{R}^d$) and the information on failure or non-failure of the product, $y^i \in \{0, 1\}$. Here 0 denotes no failure and 1 denotes failure.

1. Determine what type of learning problem you are dealing with: i.e. supervised or unsupervised learning? If supervised learning, is it a regression or classification problem.

   Solution: It is supervised leaning since we have data with labels of "failure, no failure" . Furthermore, since the labels are finite, it is a classification problem.

2. You randomly split the data such that 800 data points are used as the training set and 200 are used as the test set. Why shouldn't we use all the 1,000 available data points to train the model?

   Solution: The test dataset should be separate from the training dataset because we want to make sure the classifier performs well on the data it has never seen before.

3. After training your model, you observe it performs well on the training data, but poorly on the test data. What is one possible explanation? What could you try to improve the performance on the test data?

   Solution: The classifier could be overfitting. Conceptually, it has "memorized" the training dataset instead of learning to generalize to the unseen test dataset. You could use regularization to fit a simpler model and thus avoid overfitting to improve the performance on the test data.

Exercise 2. (Polynomial embedding)

You are given a data set $\{(x^i, y^i)\}_{i=1}^N$ with $x^i \in \mathbb{R}^3, y^i \in \mathbb{R}$. We aim to map the independent variables x^i using an appropriate feature vector $\Phi(x) = \{\Phi_i(x)\}_{i=1}^p$, $\Phi_i : \mathbb{R}^3 \to \mathbb{R}$. Then, we will use linear regression, $y = w^T\Phi(x)$. Your job is to determine an appropriate feature map Φ based on some expert knowledge as follows.

An expert on the data and associated application believes that a polynomial Φ will give a good model. Specifically, she believes that a good prediction model can be found as a degree two polynomial, with degree in each component x_l, $l = 1, 2, 3$, no more than one.

We describe the "degree". A polynomial of a vector $x \in \mathbb{R}^3$ is a linear combination of terms $x_1^p x_2^q x_3^r$, which are called monomials, where p, q, r are nonnegative integers, called the degree of the monomial in x_1, x_2, x_3, respectively. The degree of the monomial $x_1^p x_2^q x_3^r$ is $p+q+r$. The degree of a polynomial $\Phi(x)$ is the maximum of the degrees of all of its monomials, and the degree of $\Phi(x)$ in each x_l is the maximum of the degrees of its monomials in x_l. For example, the polynomial $p(x) = x_1x_2 + x_1^3 + x_1x_2^3 + x_3^2$ has degree four and its degree in x_2 is three.

Suggest an appropriate embedding $\Phi(x)$ based on the expert's advice. What is the dimension

p? In other words, how many parameters do you need to be identifying?

Solution: We enumerate all monomials that the expert has suggested are needed. There is one monomial of degree 0, namely $1= x_1^0 x_2^0 x_3^0$. There are three monomials of degree 1, namely:x_1, x_2, x_3. There are three monomials of degree 2 with degree in each component x_l, $l = 1, 2, 3$ no more than one, namely, x_1x_2, x_2x_3, x_1x_3. Thus, all together there are 7 monomials of x that have degree no more than 2 with degree in each component no more than 1. These monomials will be the components of Φ. Hence, $w^T\Phi(x) = w_0 1 + w_1x_1 + w_2x_2 + w_3x_3 + w_4x_1x_2 + w_5x_1x_3 + w_6x_2x_3$, where $\{w_i\}_{i=0}^6$ are the parameters that need to be learned from data. Thus, $p = 7$. Note that a second degree polynomial is also referred to as a quadratic polynomial. However, a general quadratic polynomial will also have three additional terms, corresponding to monomials of degree 2, namely, x_1^2, x_2^2, x_3^2.

Exercise 3. (Constant predictors)

In this problem we will investigate a "constant" predictor. Given $\{(x^i, y^i)\}_{i=1}^N$, the constant predictor gives a constant value, regardless of the independent variable x^i. We will see how to formulate this as a linear regression and what kind of predictor we get if we use the mean-square-error (MSE) loss function.

1. Consider the feature vector 1. Formulate the loss function for the problem.

   Solution: Our prediction model is $\hat{y} = w_0 \cdot 1$. Hence, the loss function is defined as follows:

   $$L(w_0) = \frac{1}{N}\sum_{i=1}^{N}(w_0 - y_i)^2 .$$

2. Show that the optimal solution is given by $w_0^* = \frac{1}{N}\sum_{i=1}^N y^i$. In other words, we will get the mean of the labels as the constant predictor.

   Solution: We take derivative of $L(w_0)$ and set it to zero to find optimal w_0.

   $$\frac{\partial L(w_0)}{\partial w_0} = \frac{2}{N}\sum_{i=1}^{N}(w_0 - y_i) = 2w_0 - \frac{2}{N}\sum_{i=1}^{N} y_i.$$

   Since the loss function is strongly convex in w_0, setting the above derivative to zero, gives us the minimum loss. Hence, $w_0^* = \frac{1}{N}\sum_{i=1}^N y_i$ achieves the minimum of the loss.

Reflection: The above predictor gives the mean of the dependent variables, y^i's. This predictor could be useful in some applications. The mean of the data is an example of the statistical information of the data. If we use other types of loss function, we can get other statistical information of the data. For example, using the mean absolute error, $L(w_0) = \frac{1}{N}\sum_{i=1}^N |w_0 - y^i|$ instead of the MSE error, we get the median of the data. Optional: for more about constant predictors, you can see this topic explored in the Stanford course on machine learning.

Exercise 4. (Logistic regression)

Consider a binary classification problem with data $\{x^i, y^i\}_{i=1}^N$, $x^i \in \mathbb{R}^d$, $y^i \in \{0, 1\}$. Let our predictor be 1 if $z^i > 0$, where $z = w^T x + b$ and $z^i \in \mathbb{R}$ corresponds to using x^i above, and 0 otherwise. Suppose you are using classification for a fault diagnosis scenario, and $x^i \in \mathbb{R}^d$ is some attributes of the process, whereas the two classes $y^i = 0$ and $y^i = 1$ correspond to *no fault* and *fault*, respectively. Suppose we are more disturbed by a *false negative* prediction than a false positive because we want to ensure we do not miss any faults. As such we slightly modify the loss function for training by introducing a constant $c \in \mathbb{R}$:

$$L(w, b) = \frac{1}{N} \sum_{i=1}^{N} c \times y^i \log(1 + e^{-z_i}) + (1 - y^i) \log(1 + e^{z_i}), \tag{0.1}$$

1. Should we choose $c > 1$ or $c < 1$? Justify your answer.

   Solution: $c > 1$. False negative happens when $y^i = 1$ and $\hat{y}^i = 0$. When $y^i = 1$, the term $y^i \log(1+e^{-z^i}) = \log(1+e^{-z^i})$ while $(1-y^i)\log(1+e^{z^i})$ equals zero. Similarly, $y^i \log(1+e^{-z^i})$ equals zero when false positive happens. Since we care about false negative more than false positive, the term $y^i \log(1+e^{-z^i})$ that corresponds to false negative should be penalized more than the term $(1 - y^i) \log(1 + e^{z^i})$.

2. Derive the gradient of the loss and write the gradient descent procedure to find the parameters (b, w) of the logistic regression.

   Solution: Define $x_0^i = 1, \forall i \in \{1, \cdots, N\}$ and $w_0 = b$. We have $\forall j \in \{0, 1, \cdots, d\}$,

$$\frac{\partial L(w)}{\partial w_j} = \sum_{i=1}^{N} \frac{\partial L(w)}{\partial z^i} \frac{\partial z^i}{\partial w_j} \tag{0.2}$$

$$= \frac{1}{N} \sum_{i=1}^{N} \left(c(-y^i) \frac{1}{1 + e^{-z^i}} e^{-z^i} + (1 - y^i) \frac{1}{1 + e^{z^i}} e^{z^i} \right) \frac{\partial z^i}{\partial w_j} \tag{0.3}$$

$$= \frac{1}{N} \sum_{i=1}^{N} \left(c(-y^i) \frac{1}{1 + e^{-z^i}} e^{-z^i} + (1 - y^i) \frac{1}{1 + e^{z^i}} e^{z^i} \right) x_j^i \tag{0.4}$$

$$= \frac{1}{N} \sum_{i=1}^{N} \frac{(1 - y^i)e^{z^i} - cy^i}{1 + e^{z^i}} x_j^i, \tag{0.5}$$

   where (0.2) follows by the chain rule. Concatenating $\frac{\partial L(w)}{\partial w_j}, j \in \{0, 1, \cdots, d\}$ gives $\nabla L(w)$.

   Next, we write the gradient descent procedure to find the parameters w of the logistic regression as follows.

   (a) We set K as number of training iterations, α as step size and initialize w randomly.

   (b) We run steps (c) and (d) for K times.

   (c) Calculate the gradient $\nabla L(w)$.

   (d) $w \leftarrow w - \alpha \nabla L(w)$.

3. Explain an approach to verify the convexity of $L(w, b)$.

   Solution: There are two common approaches to checking convexity. In class we only talked about the first approach below but here we provide the second one for completeness.

   Second-order method: Compute the Hessian of $L(w)$ and check that it is positive semi-definite for every w. An easier method is provided below.

   Convexity of composite functions: (optional) The composition of a convex function with an affine function is convex. We can verify that $g(z) := \log(1 + e^{-z})$ is convex in $z \in \mathbb{R}$ by taking the second derivative of this function $\frac{d^2g}{dz^2}$. Furthermore, $z = w_0 + w_1x_1 + \cdots + w_dx_d$ is affine in w. Hence, the composition function $\log(1 + e^{-(w_0+w_1x_1+\cdots+w_dx_d)})$ is convex in w.

Exercise 5 (Logistic function)

In this exercise, we will analyze the behaviour of the logistic function. To this end, we consider a binary classification problem with one-dimensional input data $x \in \mathbb{R}$ and class prediction $P(Y = 1 \mid x) = \sigma(wx + b)$, where $\sigma(z) = \frac{1}{1+e^{-z}}$ is the logistic or sigmoid function.

1. (Plotting) When given a plot for $y = f(x)$, one can obtain the plot for $y = f(wx + b)$ by shifting the origin to $x = \frac{-b}{w}$ and compressing the axis by a factor of $|w|$. In case $w < 0$, we flip the plot around $x = -\frac{b}{w}$. Verify this by plotting $\sigma(wx + b)$ for $w = -2$ and $b = 4$.

   Solution: Consider $f(x) = \sigma(x)$ and suppose we want to plot $\sigma(-2x + 1)$. We have $w = -2$ and $b = 4$. As illustrated in Figure 1 below, we can plot $\sigma(wx + b)$ as follows:

   - $f(x - (-\frac{b}{w})) = \sigma(x - 2)$: Shift of origin to 2.
   - $f(|w|(x + \frac{b}{w})) = \sigma(2(x - 2))$: Compression by a factor of 2.
   - $f(wx + b) = \sigma(-2(x - 2))$: Flip around 2.

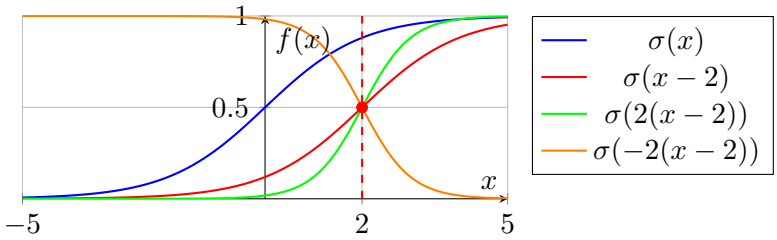

Figure 1: Plot of $\sigma(wx + b)$.

2. For the above choice of w and b, what happens with $P(Y = 1 \mid x)$ as x increases? Where do we have $P(Y = 1 \mid x) = 0.5$?

   Solution: As illustrated in Figure 1, $P(Y = 1 \mid x) = \sigma(wx + b)$ is strictly decreasing in x, which is due to w being negative. We have $P(Y = 1 \mid x) = 0.5$ for $x = -\frac{b}{w} = 2$.

3. Fix $w = 1$ and consider a data point $x = 1$. How does changing the value of b from $b = 0$ to $b = -2$ affect our prediction of $P(Y = 1 \mid x)$?

Solution: For $b = 0$, we have $P(Y = 1 \mid x) = 1/(1 + e^{-1}) \approx 0.73$ and for $b = -2$ we have $P(Y = 1 \mid x) = 1/(1 + e^{1}) \approx 0.27$. Hence, decreasing the value of b, decreases the probability of predicting $Y = 1$.

4. Now fix $b = 0$. Plot (qualitatively) $P(Y = 1 \mid x) = \sigma(wx)$ for $w \in \{0.5, 1, 2\}$. What happens with $P(Y = 1 \mid x)$ as $w \to 0$ and $w \to \infty$?

   Solution: As illustrated in Figure 2, the transition from predicting $Y = 0$ to predicting $Y = 1$ becomes sharper with increasing magnitude of w. In the limit $w \to 0$, we get $P(Y = 1 \mid x) = 0.5$, and for $w \to \infty$, the sigmoid becomes a step-function

$$\lim_{w\to\infty} \sigma(wx) = \begin{cases} 0, & x < 0, \\ 0.5, & = 0, \\ 1, & x > 0. \end{cases}$$

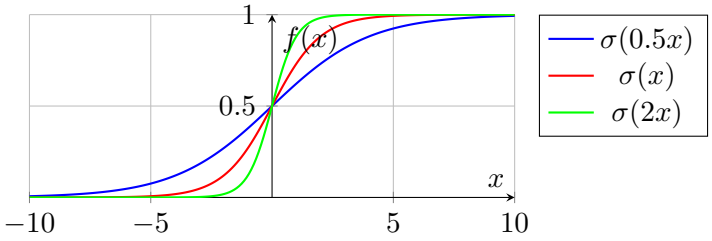

Figure 2: Sigmoid function $\sigma(wx)$ for different values of w.

5. **Bonus:** Consider the training data set

$$\mathcal{D} = \{x_i, y_i\}_{i=1}^{8} = \{(20, 1), (5, 0), (25, 1), (10, 0), (7.5, 1), (22.5, 0), (17.5, 1), (12.5, 0)\}.$$

   Plot the training data on the real-line using circles, '$\circ$', for $y = 0$, and crosses, '$\times$', for $y = 1$. Are there w and b such that for all i we have $y_i = 1$ if and only if $wx_i + b > 0$? What can you say about the benefits of using a probabilistic prediction model for the above classification problem?

   Solution: The training data is shown in Figure 3. If there existed values of w and b such that $y_i = 1$ if and only if $wx_i + b > 0$ for all i, then there would be a threshold x_0 such that either $y_i = 0$ or $y_i = 1$ for all $x_i < x_0$. In this case, the data would be referred to be *linearly separable*. However, this is not the case in our example. Since the data is not linearly separable, a probabilistic prediction model may be more appropriate to account for the uncertainty in the data.

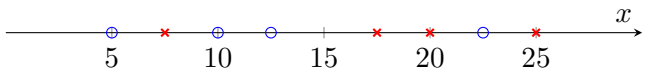

Figure 3: Plot of the training data.

Bonus Exercise (Logistic loss using Maximum Likelihood Estimation)
In this exercise, we will derive the logistic loss function using *Maximum Likelihood Estimation*. Maximum Likelihood Estimation is a statistical method used to estimate the unknown parameters of a probability distribution based on observed data. This exercise serves as a complement to help understand the logistic loss from a different perspective.

1. Consider a sequence of N independent coin tosses, with the outcome for the i^{th} toss denoted by Y_i, where $Y_i = 1$ corresponds to the outcome *heads* and $Y_i = 0$ to *tails*. Suppose the probability of *heads* is given by some parameter $p \in [0, 1]$, i.e. $P(Y^i = 1) = p$.

   (a) Verify that the probability for outcome $Y_i = y_i \in \{0, 1\}$ can be written as

   $$P(Y_i = y_i) = p^{y_i}(1-p)^{1-y_i}, \quad y_i \in \{0, 1\}.$$

   Hint: Evaluate for $P(Y_i = 1)$ and $P(Y_i = 0)$.
   Solution:

   $$P(Y_i = 1) = p^1(1-p)^0 = p$$
   $$P(Y_i = 0) = p^0(1-p)^1 = (1-p)$$

   (b) Verify that the joint probability distribution of $Y := (Y_1, \ldots, Y_N)$ can be written as

   $$P(\boldsymbol{Y} = \boldsymbol{y}) = \prod_{i=1}^{N} P(Y_i = y_i) = \prod_{i=1}^{N} p^{y_i}(1-p)^{1-y_i}.$$

   The above expression is called the Likelihood function for the sequence of outcomes. It is often denoted as $\mathcal{L}(p \mid \boldsymbol{y}) = P(\boldsymbol{Y} = \boldsymbol{y})$.
   Solution: Since we have N independent coin tosses, one can write the joint probability as a product

   $$P(\boldsymbol{Y} = \boldsymbol{y}) = P(\cap_{i=1}^{N} (Y_i = y_i)) = \prod_{i=1}^{N} P(Y_i = y_i) = \prod_{i=1}^{N} p^{y_i}(1-p)^{1-y_i}.$$

   (c) Suppose that the true parameter p is unknown. You only have access to the observations y_i. The idea of maximum likelihood estimation is to choose the value of $\hat{p}$ which maximizes the probability of the observations $\boldsymbol{Y} = \boldsymbol{y}$.

      i. Argue that the value of $\hat{p}$ which maximizes the likelihood $\mathcal{L}(p \mid \boldsymbol{y}) = P(\boldsymbol{Y} = \boldsymbol{y})$ is also minimizing the negative log-likelihood, i.e. show that

      $$\text{argmax}_p \mathcal{L}(p \mid \boldsymbol{y}) = \text{argmin}_p - \log \mathcal{L}(p \mid \boldsymbol{y}).$$

      Hint: Use the monotonicity of the $\log(\cdot)$ function.
      Solution: $\mathcal{L}(p \mid \boldsymbol{y}) \in (0, \infty)$. On this domain, $\log(\dot{})$ is a strictly monotonically increasing function. Hence,

      $$\text{argmax}_p \mathcal{L}(p \mid \boldsymbol{y}) = \text{argmax}_p \log \mathcal{L}(p \mid \boldsymbol{y}) = \text{argmin}_p - \log \mathcal{L}(p \mid \boldsymbol{y}).$$

ii. Show that the negative log-likelihood can be expressed as

$$-\log \mathcal{L}(p \mid \boldsymbol{y}) = \sum_{i=1}^{N} -(y_i \log(p) + (1-y_i)\log(1-p)).$$

Solution:

$$\begin{aligned}-\log \mathcal{L}(p \mid \boldsymbol{y}) &= -\log\left(\prod_{i=1}^{N} P(Y_i = y_i)\right) = -\sum_{i=1}^{N} \log\left(P(Y_i = y_i)\right) \\ &= -\sum_{i=1}^{N} \log\left(p^{y_i}(1-p)^{1-y_i}\right) = \sum_{i=1}^{N} -(y_i \log(p) + (1-y_i)\log(1-p)).\end{aligned}$$

2. Now we move towards formulating the logistic regression problem in terms of minimizing log-likelihood of the dataset. We will do so using the following example. Suppose that we want to estimate the relationship between the amount of study hours x a student spends for a course and his/her passing outcome for the course, denoted by $Y \in \{0, 1\}$. We propose that the probability of passing $P(Y = 1 \mid x)$ is modelled as $P(Y = 1 \mid x) = \sigma(wx + b)$, where $\sigma(z) = \frac{1}{1+e^{-z}}$ and $w \in \mathbb{R}$.

(a) Define $\hat{p}(z) := (\frac{1}{1+e^{-z}}, \frac{e^{-z}}{1+e^{-z}})$ for any $z \in \mathbb{R}$. Show that $\hat{p}$ is a probability distribution on $\{1, 0\}$ for every $z \in \mathbb{R}$.
Solution:
From the properties of e^z, we have

$$0 < \frac{1}{1+e^{-z}} < 1, 0 < \frac{e^{-z}}{1+e^{-z}} < 1 \ \forall z \in \mathbb{R},$$

and also $\hat{p}(1) + \hat{p}(0) = 1 \ \forall z$. Hence $\hat{p}(z)$ is a probability distribution $\forall z \in \mathbb{R}$.

(b) Now suppose that we are given a dataset $\{x_i, y_i\}_{i=1}^{N}$ for a class of N students where x_i is the amount of hours student i has studied and $Y_i \in \{0, 1\}$ is his/her outcome for the course. Assume that the relationship between passing probability and the amount of study hours for student i is given by $P(Y_i = 1) = \sigma(wx_i + b)$ for all students, with w, b being constant for all students.

i. Using the concepts you learned in the first part, show that the probability of outcome y_i for a student i in terms of his/her passing probability $p_i = P(Y_i = 1)$ is given by

$$P(Y_i = y_i) = p_i^{y_i}(1-p_i)^{1-y_i}.$$

Solution:

$$\begin{aligned}P(Y_i = 1) &= p_i^1(1-p_i)^0 = p_i \\ P(Y_i = 0) &= p_i^0(1-p_i)^1 = (1-p_i)\end{aligned}$$

ii. Write the above expression for $P(Y_i = 1 \mid x_i) = \sigma(wx_i + b)$.
Solution:

$$P(Y_i = y_i \mid x_i) = (\sigma(wx_i + b))^{y_i} (1 - \sigma(wx_i + b))^{1-y_i}$$

iii. Now, verify that the joint probability for outcomes $Y_1 = y_1, \cdots, Y_N = y_N$ when given the observations $\{x_i\}_{i=1}^N$ can be written as

$$P(\boldsymbol{Y} = \boldsymbol{y} \mid \boldsymbol{x}) = \prod_{i=1}^{N} (\sigma(wx_i + b))^{y_i} (1 - \sigma(wx_i + b))^{1-y_i}.$$

Solution:

$$P(\boldsymbol{Y} = \boldsymbol{y} \mid \boldsymbol{x}) = \prod_{i=1}^{N} P(Y_i = y_i \mid x_i) = \prod_{i=1}^{N} (\sigma(wx_i + b))^{y_i} (1 - \sigma(wx_i + b))^{1-y_i}.$$

iv. Suppose you do not know the parameters w, b and you would like to estimate them from the dataset. You decide to use maximum likelihood estimation to do this. Show that the expression for negative log-likelihood can be written as

$$-\log \mathcal{L}(w, b \mid \boldsymbol{y}, \boldsymbol{x}) = \sum_{i=1}^{N} \left\{ y_i \log \left(\frac{1}{\sigma(wx_i + b)} \right) + (1 - y_i) \log \left(\frac{1}{1 - \sigma(wx_i + b)} \right) \right\}$$

What happens when you minimize the negative log-likelihood over the parameters w, b? How does the above compare to the logistic loss and logistic regression that you have seen in class?

Solution:

$$\begin{aligned} -\log \mathcal{L}(w, b \mid \boldsymbol{y}, \boldsymbol{x}) &= \sum_{i=1}^{N} -y_i \log (p_i) - (1 - y_i) \log (1 - p_i) \\ &= \sum_{i=1}^{N} y_i \log \left(\frac{1}{p_i} \right) + (1 - y_i) \log \left(\frac{1}{1 - p_i} \right) \\ &= \sum_{i=1}^{N} \left\{ y_i \log \left(\frac{1}{\sigma(wx_i + b)} \right) + (1 - y_i) \log \left(\frac{1}{1 - \sigma(wx_i + b)} \right) \right\} \end{aligned}$$

The above is exactly equal to logistic loss that you have seen in class. Hence, minimizing the logistic loss can be interpreted as finding classification parameters w, b which maximize the likelihood of the training data.

The above exercise shows that logistic regression can be interpreted as finding the parameters of the probability distribution from which the data inputs and labels are generated, and serves to explain the intuition behind why we had selected this particular form of logistic loss function. Similarly, one can also formulate linear regression, regularised linear and logistic regression as well as advanced concepts in unsupervised learning like clustering, using a statistical learning perspective.