Assignment 3

STAT 541, Winter 2025

Instructor: Andrew McCormack, mccorma2@ualberta.ca
Due date: March 7th, 11:59 PM.

To receive full marks you must show your work for all derivations.

Problem 1: LDA, QDA, and Naive Bayes

(a) The naive Bayes model assumes that for each class j, $\mathbf{x}|y = j \sim \mathcal{N}_p(\boldsymbol{\mu}_j, \boldsymbol{\Sigma}_j)$ where $\boldsymbol{\Sigma}_j$ is a *diagonal* covariance matrix, and that $y \sim \text{Multinomial}_k(\boldsymbol{\pi})$ where $\boldsymbol{\pi} \in \mathbb{R}^k$ is a vector of probabilities. Find the maximum likelihood estimates of $\boldsymbol{\mu}_j, \boldsymbol{\Sigma}_j$. The likelihood function you have to maximize has the form

$$L(\boldsymbol{\mu}_1, \ldots, \boldsymbol{\mu}_k, \boldsymbol{\Sigma}_1, \ldots, \boldsymbol{\Sigma}_k, \boldsymbol{\pi}) = \prod_{i=1}^{n} p(x^{(i)}|y^{(i)})p(y^{(i)})$$
$$= \prod_{i=1}^{n}\prod_{j=1}^{k} \left(\frac{\pi_j}{(2\pi)^{p/2}\det(\boldsymbol{\Sigma}_j)^{1/2}} \exp\left(-\frac{1}{2}(\mathbf{x}^{(i)} - \boldsymbol{\mu}_j)^\top \boldsymbol{\Sigma}_j^{-1}(\mathbf{x}^{(i)} - \boldsymbol{\mu}_j) \right) \right)^{I(y^{(i)}=j)}.$$

(**Hint**: It is easier to maximize the logarithm of the likelihood function. Remember that the matrices $\boldsymbol{\Sigma}_j$ are diagonal, meaning that the inverse and determinant can be easily computed and you only have to find the diagonal entries.)

(b) Find the maximum likelihood estimator of $\boldsymbol{\pi}$ using the naive Bayes likelihood from part a). (**Hint**: As $\boldsymbol{\pi}$ is a probability vector it must satisfy the constraint $\sum_{i=1}^{k} \pi_i = 1$. You may want to introduce the Lagrange multiplier $\lambda(1 - \sum_{i=1}^{k} \pi_i)$ to maximize this, just like we did in ridge regression or the LASSO.)

(c) We do not need to use only Gaussian distributions in our generative modeling setup. Suppose instead that we have $k = 2$ classes and the conditional distribution of the feature in each class is $x|y = j \sim \text{Poisson}(\lambda_j)$. As before, we have $y \sim \text{Multinomial}_k(\boldsymbol{\pi})$. Here the $x \in \mathbb{R}$ is a univariate feature of count data. Compute $p(y = j|x)$ under this model. Find a simple expression for the decision boundary in terms of λ_1, λ_2, and $\boldsymbol{\pi}$. That is, find the set of all x where $p(y = 2|x) = p(y = 1|x)$. (You do not have to find the MLEs of λ_1, λ_2, and $\boldsymbol{\pi}$ here. You may assume that these are known parameters.)

Problem 2: Splines

(a) Cubic splines have a nice smooth appearance as they have continuous derivatives up to order two. The spline basis functions include functions of the form $f(x) = (x - \xi)_+^3 := \max(x - \xi, 0)^3$. Show that $f(x)$ is a differentiable function in x by computing the limit

$$f'(x) := \lim_{\epsilon \to 0} \frac{f(x + \epsilon) - f(x)}{\epsilon}$$

for every $x \in \mathbb{R}$. (**Hint**: At $x = \xi$ it is easiest to do this by computing the left and and right hand limits as $\epsilon \to 0^-$ and $\epsilon \to 0^+$ and showing that these two limits are the same.)

(b) Suppose we have the following vector of univariate features $x^{(i)} \in \mathbb{R}$:

$$\begin{bmatrix} x^{(1)} \\ x^{(2)} \\ x^{(3)} \\ x^{(4)} \\ x^{(5)} \\ x^{(6)} \end{bmatrix} = \begin{bmatrix} -3 \\ 5 \\ 2 \\ -4 \\ 0 \\ 1 \end{bmatrix}.$$

We use a cubic spline to fit this data set. Find the design matrix $\mathbf{X} \in \mathbb{R}^{6\times 5}$ representing the feature transformation associated with a cubic spline that has a single knot at $\xi = 1$. Indicate what feature transformation each column of this design matrix is representing.

(c) Using the `smooth.spline` function, fit smoothing splines with degrees of freedom of $5, 10, 20, 30$ to the `PetrolPrice` column in the `Seatbelts` dataset in R. Plot this data set along with the spline fits by applying `lines` to the fitted model. Make predictions for the years 1985 to 1995 and plot these predictions.

Problem 3: Nearest Neighbours and Kernel Smoothers

(a) KNN for classification classifies a new observation with feature x_* according to the class that appears most often within the K nearest neighbours of x_*. Suppose that the training data is given by

$$\begin{aligned} x^{(1)} &= (1, 0), \;\; y^{(1)} = 1 \\ x^{(2)} &= (-1, 0), \;\; y^{(2)} = 1 \\ x^{(3)} &= (2, 0), \;\; y^{(3)} = 2 \\ x^{(4)} &= (0, 1), \;\; y^{(4)} = 3 \\ x^{(5)} &= (0, -1), \;\; y^{(5)} = 2, \end{aligned}$$

where there are three different classes in this dataset. Draw or plot the regions where a 1-NN classifier would classify a new point according to classes $1, 2$, and 3. These regions are called *Voronoi cells*.

(b) KNN regression or classification can be used even if we have more exotic, non-Euclidean predictors. Suppose we have the following data that consists of a DNA snippet for patients and labels that indicated whether a patient has cancer:

$$\begin{aligned} x^{(1)} &= CCTGTGA, \;\; y^{(1)} = No\ Cancer \\ x^{(2)} &= CCTGGGT, \;\; y^{(2)} = No\ Cancer \\ x^{(3)} &= ACTATGT, \;\; y^{(3)} = Cancer \\ x^{(4)} &= ACTGTGT, \;\; y^{(4)} = No\ Cancer \\ x^{(5)} &= ACATGTG, \;\; y^{(5)} = Cancer \\ x^{(6)} &= ACTTGTG, \;\; y^{(6)} = Cancer \end{aligned}$$

We can define a *distance* between the xs as the number of locations we the DNA snippets disagree e.g. the distance between $x^{(1)}$ and $x^{(2)}$ is 2. Suppose we receive a new DNA snippet $x_* = ACTGGGT$. If we use a 3-NN rule (classifying the new observation according the class that appears most often in the three closest observations) what is our prediction for the cancer status of the new patient?

Note that you could also do something similar when the feature is not a DNA snippet but instead is text document.

(c) Let $K(z) = \exp\big(-z^2\big)$ be a kernel function. The prediction made by a kernel smoother at a new point $x_* \in \mathbb{R}$ is given by $\hat{f}(x_*) = \hat{\beta}_0(x_*)$ where $\hat{\beta}_0(x)$ is defined by

$$\hat{\beta}_0(x_*) := \underset{\beta_0}{\operatorname{argmin}} \sum_{i=1}^{n} K\left(\frac{|x_* - x^{(i)}|}{\lambda}\right)(y^{(i)} - \beta_0)^2, \quad \text{for some} \;\; \lambda > 0.$$

Find $\hat{\beta}_0(x_*)$. When the bandwidth parameter $\lambda \to \infty$ what does the prediction $\hat{f}(x_*)$ converge to? When $\lambda \to 0^+$ what does the prediction converge to? You may assume that the values of $|x_* - x^{(i)}|$ are distinct for this. Do larger values of λ result in predictions that have a high bias or a high variance?

Problem 4: Classification in R

For parts b)-e) of the question we will use the Mnist image classification dataset that was shown in class.

(a) Consider a polynomial, logistic regression basis transformation that takes $x \mapsto (1, x, x^2, x^3)$. Suppose we fit a logistic regression model and arrive at the estimated regression function $(1, x, x^2, x^3)\hat{\beta} = \frac{1}{10}(3x^3 - 8x^2 - 9x + 6)$. Plot $p(y = 1|x, \hat{\beta}) = \sigma((1, x, x^2, x^3)\hat{\beta})$ as a function of x for $x \in [-6, 6]$. Notice that this function will cross the threshold of $p(y = 1|x, \hat{\beta}) = 0.5$ more than once.

(b) Use the "Logistic and Multinomial Regression" code from the course website to fit the multinomial regression model with transformed features to the Mnist data. Time how long the `predict` function takes to make predictions for the test data by using the function `system.time`. Compute the misclassification error on the test data and find the confusion matrix using the `table` function.

(c) Fit a 1-NN classifier (using the original features) and make predictions on the first 200 observations in the test data set. You may either code this classifier from scratch or use a KNN function from a package of your choice. Time how long it takes to make these predictions – this should not take more than five minutes. Compute the misclassification rate on these 200 observations and the confusion matrix.

(d) A simpler variant of LDA is one where we make the extra assumptions that $\pi_0 = \pi_2 \cdots = \pi_9 = \frac{1}{10}$ and $\mathbf{\Sigma} = \mathbf{I}_{784}$. Determine the prediction function associated with this model. How is this similar to a NN prediction rule?

(e) Use the prediction rule in d) (using the original features) to make predictions on the test data set. Compute how long it takes to make these predictions, the misclassification rate, and the confusion matrix.

(f) One of the three above methods will provide very accurate predictions. Knowing that we are working with images of digits, provide an explanation for why this is the case.

The misclassification rate is defined as

$$\text{Misclassification Rate} = \frac{\#\text{Incorrect Predictions}}{\text{Total Number of Predictions Made}}$$

The confusion matrix C for a classification problem with k classes is the matrix where the entry C_{ij} counts the number of total predictions in class i where the true, underlying class turned out to be class j. Ideally, the confusion matrix will have the largest counts along its diagonal. Confusion matrices allow us to spot which pairs of classes are easily "confused" with each other. For instance, 4 and 9 are easily confused as they have similar shapes.