Samuele Serri 7069839
Ege Mert Balcik 7071632

1 Cross-Validation (4 Points)

Cross-validation is a widely used data resampling method to estimate the test error of models and to tune model parameters. State true or false for the following statements (answer with proper reasoning):

(a) The purpose of cross-validation is to reduce the size of the dataset for faster computation.

(b) For any k, repeated application of k-fold cross-validation will always produce the same estimation of error.

(c) Repeated application of leave-one-out cross-validation will always produce the same estimation of error.

Note: Leave-one-out cross-validation is a configuration of k-fold cross-validation where k is set to the number of examples (n) in the dataset.

(d) Leave-one-out cross-validation is used when the dataset is extremely large as this reduces the computational cost.

(a) False → Purpose of cross-validation is estimating prediction error.

(b) False → For different k's, different sets will be used for training and hence the estimation of error will presumably be different.

(c) True → $K = N$ will always be the same (except left out data point), which means same datasets will be used for training.

(d) False → Leave-one-out cross-validation is computationally expensive since it requires N applications of the learning method. Hence, it is best for small datasets.

2 KNN Classifier (5 Points)

Training instance	Height	Width	class
I_1	167	75	0
I_2	183	62	1
I_3	175	64	1
I_4	170	85	2

We consider KNN classifier for the following questions:

(a) Consider the data points shown in Table 2, where each point is categorized as class 0, 1, or 2. We need to classify a new instance I_{new} with the values ($Height = 168$, $Width = 80$). To do this we will use the L2 (Euclidean) distance between two points in 2-dimensional space, defined as:

$$d([x_1, x_2], [y_1, y_2]) = \sqrt{(x_1 - y_1)^2 + (x_2 - y_2)^2}$$

How would a 1-NN and a 3-NN classifier classify the new point? Show all steps for working. (2 points)

(b) is it generally better to use an odd or even value for k? Explain your reasoning (1 point).

(c) While KNN works well for low-dimensional data, it becomes cumbersome for high-dimensional data with L2 distance. Explain the problem with high-dimensional data in 1-2 sentences. (1 point)

(d) What strategy can be used to choose the best value for k in the k-Nearest Neighbors (KNN) algorithm? (1 point)

a)

Compute the distances between $I_{new} = (168, 80)$ and all the other points in the dataset:

$$d((168, 80), (167, 75)) = \sqrt{(168 - 167)^2 + (80 - 75)^2} = \sqrt{26} \approx 5.0990$$

$$d((168, 80), (183, 62)) = \sqrt{(168 - 183)^2 + (80 - 62)^2} = \sqrt{549} \approx 23.431$$

$$d((168, 80), (175, 64)) = \sqrt{(168 - 175)^2 + (80 - 64)^2} = \sqrt{305} \approx 17.464$$

$$d((168, 80), (170, 85)) = \sqrt{(168 - 170)^2 + (80 - 85)^2} = \sqrt{29} \approx 5.3852$$

The closest point using euclidean distance is $(167, 75)$ so class 0 will be assigned to I_{new} with 1-NN.
With 3-NN classifier we have to consider the three closest neighbor: $(160, 175), (170, 85), (175, 64)$ respectively of class 0,2 and 1. The point I_{new} will take the class of the majority vote but since we have one representative for each class it depends on the weight assigned to each point.

b)

If we have classification problems we only 2 classes then choosing an odd number as k would be better because it avoids ties.

c)

In high-dimensional spaces almost all vectors have the same euclidean distance with I_{new}.

d)

The choice of k is an hyper-parameter and is strongly data-dependent. We can divide data into training, validation and testing sets, choose the value of k with validation set and test it with the test set. Alternatively, if the dataset is not too large, we can use cross-validation which consists in splitting data into folds, using each fold as validation and then averaging the results.

3 Regularization (5 Points)

Given a linear classifier f with parameters W, the loss L with the regularization term is defined as:

$$L(W) = \frac{1}{N} \sum_{i=1}^{N} L_i(f(x_i, W), y_i) + \lambda R(W)$$

(a) What is the purpose of regularization?

(b) List two regularization techniques other than L1 and L2 regularization.

(c) What is the range of values for the regularization strength λ?

(d) What does increasing the regularization strength λ do?

(e) Which regularization technique(s) are used in the Elastic Net?

a)

The purpose of regularization is to make the model work better on test data. We add a penalization to avoid over-fitting or under-fitting during the training process.

b)

Elastic net and dropout are examples of regularization techniques.

c)

$\lambda \in [0, +\infty)$

d)

If $\lambda = 0$ then we express no preferences on the complexity of the model and we over-fit.
If $\lambda >> 0$ then the weights are forced to be small and we under-fit.

e)

Elastic net combines L1 and L2 regularization minimizing the loss:

$$L(w) = \|y - Xw\|_2^2 + \lambda_1 \|w\|_1 + \lambda_2 \|w\|_2^2$$

4 Gradient Descent (6 Points)

Gradient Descent is a first-order iterative optimization algorithm used to find the values of model parameters that minimize a loss function in many machine learning algorithms. In this exercise, we study an example of Gradient Descent in two-dimensional space. We want to minimize a loss function $f : \mathbb{R} \to \mathbb{R}$ which is defined as $f(x) = x^2 - 6x + 5$.

(a) Assuming the starting point is $x^{(0)} = 2$, please perform Gradient Descent with step size (also known as learning rate) 0.2 to minimize the loss function f. The iteration should stop if the L2-norm of (gradient $\times$ learning rate) at the current point is less than 0.2. Please show your intermediate steps. (2 point)

Hint: The iteration should finish within 2 steps.

(b) If we continue the process from question (a) to minimize the loss function, will it converge at some point? If yes, which point will it converge to? If no, what is the reason it does not converge? (1 point)

Hint: You may check the property of the loss surface using WolframeAlpha.

(c) Starting from initial point $x^{(0)} = 0$, now we perform Gradient Descent to minimize the same loss using a step size of 1. What would happen this time? How would you modify your method to reach the same solution? (2 points)

(d) This time starting from initial point $x^{(0)} = 1$, for what value of learning rate ϵ can you reach the convergence point in a single step? (1 point)

Submission Details:

Upload your submission to our CMS in groups of two to three students until *November 24, 2024 at 17:59*. Late submissions will not be graded! The submission should be uploaded by exactly **one** team member. Make sure that your submission contains the name and matriculation number of each team member. Submit your solution as a *pdf* file with your answers.

4)

(a) $x^{(t+1)} = x^{(t)} - \eta \cdot \nabla f(x^{(t)})$

↳ learning rate

$\nabla f(x) = 2x - 6 \quad x^{(0)} = 2 \quad f'(2) = -2$

Step 1: $2 - 0.2 \cdot (-2) = 2.4 = x^{(1)}$ $\quad |-2 \cdot 0.2| = 0.4 < 0.2$ ✗

Step 2: $2.4 - 0.2 \cdot \underbrace{f'(2.4)}_{-1.2} = 2.64 = x^{(2)}$ $\quad |-1.2 \cdot 0.2| = 0.24$, $0.24 < 0.2$ ✗

Step 3: $2.64 - 0.2 \cdot \underbrace{f'(2.64)}_{-0.72} = 2.784 = x^{(3)}$ $\quad |-0.72 \cdot 0.2| = 0.144$, $0.144 < 0.2$ ✓

(b) $\nabla f(x) = 2x - 6 \rightarrow$  → If it converges, it will converge to 3 since the minimum occurs at $x^{(n)} = 3$ where $\nabla f(x) = 0$.

Also, note that $f(x) = x^2 - 6x + 5$ is a convex function. Moreover, it is strongly convex function since $f''(x) = 2 > 0$; therefore, it has a global minimum.

So, in this case, gradient descent will converge to 3 since we chose proper learning rate. We understood properness of learning rate from question (a) because it did not overshoot in our steps.

(c) $x^{(0)} = 0$, $\alpha = 1$ $\quad \nabla f(x) = 2x - 6$

Step 1: $f'(0) = -6$ $\quad x^{(1)} = 0 - 1 \cdot (-6) = 6$

Step 2: $f'(6) = 6$ $\quad x^{(2)} = 6 - 1 \cdot 6 = 0$

Step 3: $f'(0) = -6$ $\quad x^{(3)} = 0 - 1 \cdot (-6) = 6$

} Points oscillate between 6 and 0. Hence, it diverges.

To address divergence and ensure convergence, we can use smaller learning rate. Thus, we can reach the same result.

(d) $x^{(t+1)} = x^{(t)} - \epsilon \cdot \nabla f(x^{(t)})$

$\nabla f(x) = 2x - 6 \rightarrow$  → minimum point occurs at $x = 3$ that is, $\nabla f(3) = 0$

To reach convergence in one step, we set $x^{(1)} = 3$. Also, remember $x^{(0)} = 1$.

↳ $\nabla f(1) = -4$

$3 = 1 - \epsilon \cdot (-4)$

$2 = 4\epsilon$

$\epsilon = 0.5$