2252654 NLP Lab 6 Math Exercise

Phuong Huynh

March 2025

1 Problem 1

1.1 Question a:

Using OLS, the optimal values of a and b minimize the sum of squared residuals:

$$\sum_{i=1}^{n}(y_i - ax_i - bx_i^2)^2$$

To find b, we differentiate the loss function with respect to b.

$$\frac{d}{db}\sum_{i=1}^{n}(y_i - ax_i - bx_i^2)^2 = 0$$

$$\sum_{i=1}^{n} 2(y_i - ax_i - b_i^2)(-x_i^2) = 0$$

$$\sum_{i=1}^{n}(-x_i^2 y_i + ax_i^3 + bx_i^4) = 0$$

$$b\sum_{i=1}^{n} x_i^4 = \sum_{i=1}^{n}(x_i^2 y_i) - a\sum_{i=1}^{n} x_i^3$$

$$b = \frac{\sum_{i=1}^{n}(x_i^2 y_i) - a\sum_{i=1}^{n} x_i^3}{\sum_{i=1}^{n} x_i^4}$$

1.2 Question b:

Model 1 is Linear Regression using one parameter a while Model 2 is Quadratic Regression using 2 parameters a and b. Model 2 has more flexibility because it includes an additional term bx^2 allowing it to capture non-linear patterns in the data.
Since we are evaluating a training data fit, a Quadratic Regression with more parameters generally fits the data better. So the answer is **(b) Model 2**

1.3 Question c:

Model 1 is simpler and less prone to overfitting. Model 2 is more flexible but might overfit the training data. If the true relationship between y is linear, Model 1 will generalize better. If the relationship is quadratic, Model 2 will be better.
Since we are not provided enough information, the answer is **(d) impossible to tell**

2 Problem 2:

2.1 Question a:

Since the parameter w in model A is squared, it would always be positive. This is a huge restriction for model A because it cannot correctly determine the mapping when the true relation requires a negative parameter. (for example: $y = -3x$)
Therefore, the answer is **(b) There are datasets for which B would perform better than A.**

2.2 Question b:

Model A has two parameters, while Model B has only one. Model A can represent all relationships that model B can. For example, if w of model B is -3, model A can represent that by setting $w_1 = 0$ and let $w_2 = -3$. Since we have unlimited data, Model A will always have the potential to fit the data at least as well as Model B. There is no dataset where Model B performs better than Model A, because Model A has strictly greater expressiveness.
The answer is **(a) There are datasets for which A would perform better than B.**

3 Problem 3:

3.1 Question a:

The least square objective function is:

$$\sum_{i=1}^{n}(y_i - w_1^2 x_{i,1} - w_2^2 x_{i,2})^2$$

To derive w_1, we differentiate with respects to w_1:

$$\frac{d}{dw_1}\sum_{i=1}^{n}(y_i - w_1^2 x_{i,1} - w_2^2 x_{i,2})^2 = 0$$

$$= \sum_{i=1}^{n} 2(y_i - w_i^2 x_{i,1} - w_2^2 x_{i,2})(-2w_1 x_{i,1}) = 0$$

$$-4w_1 \sum_{i=1}^{n}(y_i - w_i^2 x_{i,1} - w_2^2 x_{i,2})(x_{i,1}) = 0$$

$$\sum_{i=1}^{n}(y_i - w_i^2 x_{i,1} - w_2^2 x_{i,2})(x_{i,1}) = 0$$

$$\sum_{i=1}^{n} w_1^2 x_{i,1} = \sum_{i=1}^{n} x_{i,1} y_i - \sum_{i=1}^{n} w_2^2 x_{i,2}$$

$$w_1^2 = \frac{\sum_{i=1}^{n} x_{i,1} y_i - \sum_{i=1}^{n} w_2^2 x_{i,2}}{\sum_{i=1}^{n} x_{i,1}}$$

$$w_1 = \pm\sqrt{\frac{\sum_{i=1}^{n} x_{i,1} y_i - \sum_{i=1}^{n} w_2^2 x_{i,2}}{\sum_{i=1}^{n} x_{i,1}}}$$

4 Problem 4:

4.1 Question a:

- **(4.1):** Ordinary Least Squares (OLS)
- **(4.2):** Ridge Regression (L2 Regularization)
- **(4.3):** Lasso Regression (L1 Regularization)

4.2 Question b:

Ridge regression shrinks the coefficients but does not set them exactly to zero. As λ increases, the model's flexibility decreases because the weight magnitudes are restricted. This leads to **higher bias** (since the model is less capable of capturing complex relationships).

4.3 Question c:

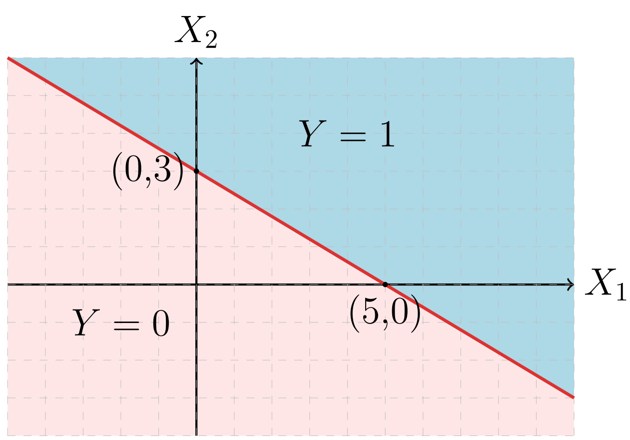
Lasso regression enforces sparsity by setting some coefficients to exactly zero. As λ increases, more weights become zero, leading to a simpler model. This reduces the model's flexibility, which **decreases variance**.

4.4 Question d:

I guess something suppose to happen here

5 Question 5:

5.1 Question a:



5.2 Question b:

$$P(Y = 1 \mid X_1, X_2) = \sigma(3X_1 + 5X_2 - 15) = \frac{1}{1 + \exp(-(3X_1 + 5X_2 - 15))}$$

6 Problem 6:

6.1 Question a:

With the assumption that threshold is 0.5:
With classifier 1: It correctly classifies the 2 labels '0' but misclassifies the label '1'.

With classifier 2: It correctly classifies the 2 labels that are of the left of x-axis. It misclassifies the label '0' at the right of the x-axis.

6.2 Question b:

Calculate joint probability:

$$P(y = 0|x = -1; w) \times P(y = 1|x = 0; w) \times P(y = 0|x = 1; w)$$

Joint probability of Classifier 1: $(1 - 0.35) \times 0.35 \times (1 - 0.35) = 0.1479$
Joint probability of Classifier 2: $(1 - 0) \times 1 \times (1 - 1) = 0$
Since the joint probability of Classifier 1 is higher, the ML solution is **Classifier 1**

6.3 Question c:

Since classifier 1 is a constant, It would not be affected by regularization that penalized w parameter. On the other hand, classifier 2 would be penalize by an unknown amount. Classifier 2 would give an even worse outcome compared to classifier 1 thus the answer is **N (not affect)**.

7 Problem 7:

7.1 Question a:

When $\lambda = 0$, the regularization term disappears.

$$l(x^{(j)}, y^{(j)}, w) = y^{(j)} \sum_{i=1}^{d} w_i x_i^{(j)} - \ln \left(1 + \exp \left(\sum_{i=1}^{d} w_i x_i^{(j)} \right) \right)$$

Taking the derivative with respect to w_i:

$$\frac{\partial l}{\partial w_i} = y^{(j)} x_i^{(j)} - \frac{x_i^{(j)}}{1 + \exp\left(-\sum_{k=1}^{d} w_k x_k^{(j)}\right)}$$

Using learning rate η, the stochastic gradient descent update rule is:

$$w_i \leftarrow w_i + \eta \left(y^{(j)} x_i^{(j)} - \frac{x_i^{(j)}}{1 + \exp\left(-\sum_{k=1}^{d} w_k x_k^{(j)}\right)} \right)$$

7.2 Question b:

A dense data structure stores all **d** features, even if many are zero while sparse data structure only stores **s** nonzero elements
The update rule requires computing the summation $\sum_{k=1}^{d} w_k x_k^{(j)}$ which takes $\boldsymbol{O(d)}$ time complexity when use **dense** data structure and take $\boldsymbol{O(s)}$ time complexity when use **sparse** data structure.

7.3 Question c:

The L2-regularized logistic loss function when $\lambda > 0$:

$$F(w) = l(x^{(j)}, y^{(j)}, w) - \frac{\lambda}{2} \sum_{i=1}^{d} w_i^2$$

where the logistic loss function is:

$$l(x^{(j)}, y^{(j)}, w) = y^{(j)} \sum_{i=1}^{d} w_i x_i^{(j)} - \ln\left(1 + \exp\left(\sum_{i=1}^{d} w_i x_i^{(j)}\right)\right)$$

Taking the derivative with respect to w_i:

$$\frac{\partial l}{\partial w_i} = y^{(j)} x_i^{(j)} - \frac{x_i^{(j)}}{1 + \exp\left(-\sum_{k=1}^{d} w_k x_k^{(j)}\right)} - \lambda w_i$$

Using a step size η, the stochastic gradient descent update rule is:

$$w_i \leftarrow w_i + \eta \left(y^{(j)} x_i^{(j)} - \frac{x_i^{(j)}}{1 + \exp\left(-\sum_{k=1}^{d} w_k x_k^{(j)}\right)} - \lambda w_i\right)$$

7.4 Question d:

The answer is the same with **b)**. Only $\sum_{k=1}^{d} w_k x_k^{(j)}$ require $\boldsymbol{O(d)}$ time while all other operations take $\boldsymbol{O(1)}$ times. Therefore dense structure takes $\boldsymbol{O(d)}$ time complexity.

7.5 Question e:

Since $x_i^{(j)} = 0$ for all sequences. The update rule can be simplified to:

$$w_i \leftarrow w_i + \eta\,(-\lambda w_i)$$

$$w_i \leftarrow w_i \times (1 - \eta\lambda)$$

For each example in sequence:

$$w_i^{(t+1)} \leftarrow w_i^{(t)} \times (1 - \eta\lambda)$$

Apply recursively for $\boldsymbol{k}$ updates, we would get:

$$w_i^{(t+k)} \leftarrow w_i^{(t)} \times (1 - \eta\lambda)^k$$

7.6 Question f:

Instead of applying update at every step for all d weights, only apply it lazily when w_i is updated due to a **nonzero** feature.
Since only s features (on average) are nonzero in each sample, We only update those s weights instead of all d weights.

Algorithm

1. Initialize weights w and timestamp=0.
2. For each training example $(x^{(j)}, y^{(j)})$:
   (a) For each nonzero feature i in $x^{(j)}$:
      i. Apply delayed weight decay:
      $$w_i \leftarrow (1 - \eta\lambda)^{(\text{steps since last update})} w_i$$
      ii. Compute gradient and update weight:
      $$w_i \leftarrow w_i + \eta \left(y^{(j)} - \frac{1}{1 + \exp\left(-\sum_{k \in \text{nonzero}(x^{(j)})} w_k x_k^{(j)}\right)} \right) x_i^{(j)}$$
      iii. Store timestamp of the last update for w_i.

Time complexity for sparse data structure would be $\boldsymbol{O(s)}$.