Foundations of Machine Learning Assignment 4

Daniel Atighetchi - 3664061- B.Sc.Informatik

Omar Aboulgadayel - 3610497 - M.Sc.Informatik

19. May 2025

Question 1:

Task 1:

Step 1: Expand the Loss

$$L(w) = \|Xw - y\|_2^2 + \lambda \|w\|_2^2$$

$$= (Xw - y)^T (Xw - y) + \lambda w^T w$$

Step 2: Take the derivative with respect to w and set it to 0

$$\nabla_w L = 2X^T Xw - 2X^T y + 2\lambda w$$

$$2X^T Xw - 2X^T y + 2\lambda w = 0$$

$$X^T Xw - X^T y + \lambda w = 0$$

$$(X^\top X + \lambda I)w = X^\top y$$

Step 3: Solve for w

$$w^* = (X^T X + \lambda I)^{-1} X^T y$$

Final Closed-Form Solution

$$w^* = (X^T X + \lambda I)^{-1} X^T y$$

Task 2:

$$w^* = (X^\top X)^{-1} X^\top y$$

- Compute $X^\top X$: $O(nd^2)$.
- Invert $X^\top X$ (using Cholesky decomposition): $O(d^3)$.
- Compute $X^\top y$: $O(nd)$.
- Compute $(X^\top X)^{-1}(X^\top y)$: $O(d^2)$.

Total Complexity:

$$O(nd^2 + d^3)$$

Task 3:

Woodbury Identity:

$$(A + UCV)^{-1} = A^{-1} - A^{-1}U(C^{-1} + VA^{-1}U)^{-1}VA^{-1}$$

Ridge Regression:

$$w^* = (X^\top X + \lambda I)^{-1} X^\top y$$

Assign:

$$A = \lambda I, \quad U = X^\top, \quad C = I, \quad V = X.$$

Plugging in:

$$(X^\top X + \lambda I)^{-1} = \lambda^{-1} I - \lambda^{-1} X^\top \left(I + X\lambda^{-1} X^\top\right)^{-1} X\lambda^{-1}$$

$$= \frac{1}{\lambda} I - \frac{1}{\lambda^2} X^\top \left(I + \frac{1}{\lambda} X X^\top\right)^{-1} X$$

Final form:

$$w^* = \frac{1}{\lambda} X^\top \left(I + \frac{1}{\lambda} X X^\top\right)^{-1} y$$

Task 4:

Computation Parts:

- Compute $XX^\top$: $O(n^2 d)$.
- Invert $I + \frac{1}{\lambda} XX^\top$: $O(n^3)$.
- Multiply $X^\top$ with the inverse: $O(n^2 d)$.
- Multiply by y: $O(n^2)$.

Total Complexity:

$$O(n^2 d + n^3)$$

Task 5:

$$(XX^\top)_{ij} = x_i^\top x_j$$

Represents the inner product between the i-th and j-th data points in X.

Used in kernel methods such as PCA!

Question 3:

Task 1:

The maximum F1-score is 1. This is achieved when both the precision and the recall is 1.

$$\frac{2 \cdot 1 \cdot 1}{1 + 1} = 1$$

Example: A perfect binary classifier that always predicts the correct class.

The minimum F1 is 0. This is achieved when either the precision or recall is 0.

$$\frac{2 \cdot 0 \cdot x}{0 + x} = 0$$

Example: Classifier always predicts the wrong class for every positive example (e.g., predicts all negatives, missing all positives).

Task 2:

Since $\frac{0}{0}$ is undefined, the F1-score is undefined!

Task 3:

- Total samples: n
- Positives: m
- Negatives: $n - m$
- True Positives: Half of the actual positives will be predicted correctly:

$$TP = \frac{m}{2}$$

- False Positives: Half of the negatives predicted as positives:

$$FP = \frac{n-m}{2}$$

- False Negatives: Half of the positives predicted as negatives:

$$FN = \frac{m}{2}$$

Now we can plug these values in:

- **Precision**:

$$\text{Precision} = \frac{TP}{TP+FP} = \frac{\frac{m}{2}}{\frac{m}{2}+\frac{n-m}{2}} = \frac{m}{n}$$

- **Recall**:

$$\text{Recall} = \frac{TP}{TP+FN} = \frac{\frac{m}{2}}{m} = \frac{1}{2}$$

F1-score:

$$F1 = \frac{2 \cdot \text{Precision} \cdot \text{Recall}}{\text{Precision} + \text{Recall}} = \frac{2 \cdot \frac{m}{n} \cdot \frac{1}{2}}{\frac{m}{n}+\frac{1}{2}} = \frac{\frac{m}{n}}{\frac{m}{n}+\frac{1}{2}}$$

Special case: $n - m = m$

Then:

$$n = 2m \Rightarrow \frac{m}{n} = \frac{1}{2}$$

Plug into the F1 formula:

$$F1 = \frac{2 \cdot 0.5 \cdot 0.5}{0.5 + 0.5} = \frac{0.5}{1} = 0.5$$

So the expected F1-score is 0.5 when the classes are balanced and the classifier guesses randomly.

Task 4:

1. **Cats and Dogs**
   - **C1:** Perfect classification (no errors).
   - **C2:** Random guessing (50% accuracy).
   - **Winner:** C1 because it has no misclassifications.
2. **Trees**
   - **C1:** Horrible classification (all errors).
   - **C2:** Random guessing (50% accuracy).
   - **Winner:** C2 because random guessing is better than getting all wrong.
3. **Fire detection**
   - **C1:** More False Positives
   - **C2:** More False Negatives
   - **Winner:** C1 since false positives are better in this case than false negatives.
4. **Mushroom Soup**
   - **C1:** More False Positives
   - **C2:** More False Negatives
   - **Winner:** C2 since false negatives are better in this case than false positives.

Question 4:

Task 1:

Actual \ Predicted	Apple	Grapes	Orange
Apple	5	1	2
Grapes	1	7	1
Orange	1	1	6

Task 2:

- **Precision:**

$$\text{Precision} = \frac{TruePositives}{TruePositives + FalsePositives}$$

- **Recall :**

$$\text{Recall} = \frac{TruePositives}{TruePositives + FalseNegatives}$$

- **F1-score:**

$$\text{F1-score} = 2 \cdot \frac{\text{Precision} \cdot \text{Recall}}{\text{Precision} + \text{Recall}}$$

Metrics	Precision	Recall	F1
Apple	$\frac{5}{7} \approx 0.714$	$\frac{5}{8} = 0.625$	$\frac{2}{3} \approx 0.666$
Grapes	$\frac{7}{9} \approx 0.778$	$\frac{7}{9} \approx 0.778$	$\frac{7}{9} \approx 0.778$
Orange	$\frac{2}{3} \approx 0.666$	$\frac{3}{4} = 0.75$	$\frac{12}{17} \approx 0.706$

Task 3:

Micro-averaging calculates metrics globally by counting total TP, FP, FN across all classes.

- **Micro Precision:**

$$\frac{TruePositives}{TruePositives + FalsePositives} = \frac{18}{25} = 0.72$$

- **Micro Recall:**

$$\frac{TruePositives}{TruePositives + FalseNegatives} = \frac{18}{25} = 0.72$$

- **Micro F1:**

$$\frac{2 \cdot (0.72 \cdot 0.72)}{0.72 + 0.72} = \frac{2 \cdot 0.5184}{1.44} = \frac{1.0368}{1.44} = 0.72$$

Macro-averaging computes the metric for each class independently and then takes the average.

$$\text{Macro-F1} = \frac{0.666 + 0.778 + 0.706}{3} = \frac{2.15}{3} \approx 0.7167$$