Homework 4

Xiaofan Jiao

Question 1. Comparing multi-class classifiers for handwritten digits classification.

1)

	Classifier	Precision	Recall	F1 Score
0	KNN	0.970688	0.9705	0.970452
1	Logistic Regression	0.925423	0.9256	0.925445
2	Linear SVM	0.918006	0.9183	0.918014
3	Kernel SVM	0.979201	0.9792	0.979186
4	Neural Network	0.951872	0.9518	0.951767

2) KNN works well by looking at nearby data points, but it can be slow and less effective with noisy or poorly separated data. Logistic regression does a good job with linearly separable data but struggles with more complex patterns. Linear SVM also works for linearly separable classes but performs worse on the MNIST dataset because it needs more complex decision boundaries. Kernel SVM, using an RBF kernel, handles non-linear relationships well, making it very effective for MNIST. The neural network, with its simple setup, performs well but could do better with a more complex design. Overall, Kernel SVM and KNN perform best due to their ability to handle non-linear patterns, while logistic regression and linear SVM are limited by their linear nature. Neural networks are promising but might improve with more depth.

Question 2. SVM.

1) Setting the margin c= simplifies the equations for Support Vector Machines (SVMs). The margin is the distance between the decision boundary (the line or plane that separates the classes) and the closest data points (support vectors). When c=1 , we make the math easier without losing generality. This means we can still separate the classes effectively. By scaling the weights and the bias term accordingly, we can adjust any margin to 1.

2)

2) Using the Lagrangian dual formulation

the primal problem is: $\min_{w,b} \frac{1}{2}\|W\|^2$

subject to: $y_i(W \cdot X_i + b) \geq 1$

We introduce Lagrange multipliers $\alpha_i \geq 0$ for each constraint

$$L(w,b,\alpha) = \frac{1}{2}\|W\|^2 - \sum_{i=1}^{n} \alpha_i[y_i(w \cdot x_i + b) - 1]$$

We take partial derivative of L with respect to w and b and set them to zero:

$$\frac{\partial L}{\partial w} = w - \sum_{i=1}^{n} \alpha_i y_i x_i = 0$$

$$\underline{w = \sum_{i=1}^{n} \alpha_i y_i x_i}$$

This implies that w is a linear combination of the training data points x_i, weighted by the Lagrange multipliers α_i and the class labels y_i. Only the data points with non-zero α_i (support vectors) contribute to w.

3)

3) According to the (KKT) conditions, for each data point i:

$\alpha_i(y_i(w \cdot x_i + b) - 1) = 0$

- If $\alpha_i > 0$ then $y_i(w \cdot x_i + b) = 1$ where the points lies on the margin
- If $\alpha_i = 0$, the point lies either correctly classified or outside the margin, not effecting w.

Therefore, only the support vectors (with $\alpha_i > 0$) determine the decision boundary.

4) a)

4)(a) Problem: we need to find a line (decision boundry) that seperates the positive & negative samples.

- For $h \leq 1$: the negative sample $(h, 1)$ is closer to the positive sample (0, 0
  We can draw a line seperating the positive points (0, 0) and (2, 2) from the negative points $(h, 1)$ & (0, 3)
- For $h > 1$: the negative sample $(h, 1)$ moves further right on the x-axi
  There is still a posibility to draw a line seperating the positive & negative points because the negative point (0, 3) is higher up.

∴ The training points are linearly seperable for $0 < h < 2$.

b)

4)(b) For $0 < h < 2$:

- As h increase, the negative point $(h, 1)$ moves rightward
- The decision boundary will adjust to maintain the maximum margin between the positive & negative samples
- When h is small, the boundary will be closer to vertical
- As h approaches 2, the boundary tilts more towards the horizontal to accommodate the separation

∴ The orientation of the decision boundary changes as h changes within the seperable range. Initially more vertical. it becomes more horizontal as h increases

Question 3. Neural networks and backpropagation.

a)

a) $\frac{\partial l(w,\alpha,\beta)}{\partial w} = -\sum_{i=1}^{m} 2(y^i - \sigma(u^i))\sigma(u^i)(1-\sigma(u^i))z^i$

where $u^i = w^T z^i$

① Cost Function:

$$l(w,\alpha,\beta) = \sum_{i=1}^{m} (y^i - \sigma(w^T z^i))^2$$

② Differentiate the cost Function:

$$\frac{\partial l(w,\alpha,\beta)}{\partial w} = \sum_{i=1}^{m} \frac{\partial}{\partial w}(y^i - \sigma(w^T z^i))^2$$

$$= 2(y^i - \sigma(u^i))\frac{\partial}{\partial w}(y^i - \sigma(u^i))$$

$$\left[\begin{array}{l} \frac{\partial}{\partial w}\sigma(u^i) = \sigma'(u^i)\frac{\partial u^i}{\partial w} \\ \text{where } \sigma'(u^i) = \sigma(u^i)(1-\sigma(u^i)) \end{array}\right]$$

$$\left[\frac{\partial u^i}{\partial w} = \frac{\partial}{\partial w}(w^T z^i) = z^i\right]$$

$$= -\sum_{i=1}^{m} 2(y^i - \sigma(u^i))\sigma(u^i)(1-\sigma(u^i))z^i$$

$$= -\sum_{i=1}^{m} 2(y^i - \sigma(u^i))(1-\sigma(u^i))z^i$$

This Proves the given gradient expression

b)

b) Gradient w.r.t α:

$Z_1^i = \sigma(\alpha^T X^i)$

$$\frac{\alpha l(w,\alpha,\beta)}{\alpha \alpha} = \sum_{i=1}^{m} \frac{\alpha}{\alpha a}(y^i - \sigma(W^T z^i))^2$$

Using the chain Rule:

$$\frac{\alpha l(w,a,\beta)}{\alpha a} = \sum_{i=1}^{m} 2(y^i - \sigma(u^i)) \frac{\alpha}{\alpha a}(y^i - \sigma(u^i))$$

$$\frac{\alpha}{\alpha a}\sigma(u^i) = \sigma'(u^i)\frac{\alpha u^i}{\alpha a}$$

Since $u^i = W^T z^i$ and $Z_1^i = \sigma(\alpha^T x^i)$

$$\frac{\alpha u^i}{\alpha a} = W_1 \cdot \sigma'(\alpha^T x^i) \cdot x^i$$

$$\frac{\alpha l(w,\alpha,\beta)}{\alpha a} = -\sum_{i=1}^{m} 2(y^i - \sigma(u^i))(1-\sigma(u^i))W_1\sigma'(\alpha^T x^i))x^i$$

Gradient w.r.t β:

$Z_2^i = \sigma(\beta^T x^i)$

$$\frac{\alpha l}{\alpha \beta} = \sum_{i=1}^{m} \frac{\alpha}{\alpha \beta}(y^i - \sigma(W^T z^i))^2$$

Using the chain Rule:

$$\frac{\alpha l(w,\alpha,\beta)}{\alpha \beta} = \sum_{i=1}^{m} 2(y^i - \sigma(u^i)) \frac{\alpha}{\alpha \beta}(y^i - \sigma(u^i))$$

$$\frac{\alpha}{\alpha \beta}(u^i) = \sigma'(u^i)\frac{\alpha u^i}{\alpha \beta}$$

Since $u^i = W^T z^i$ and $z_2^i = \sigma(\beta^T x^i)$

$$\frac{\alpha u^i}{\alpha \beta} = W_2 \cdot \sigma'(\beta^T x^i) \cdot x^i$$

$$\frac{\alpha l(w,\alpha,\beta)}{\alpha \beta} = -\sum_{i=1}^{m} 2(y^i - \sigma(u^i))(1-\sigma(u^i))W_2\sigma'(\beta^T x^i)x^i$$

Question 4. Feature selection and change-point detection.

a)

Question 4

1) The mutual information $I(X;Y)$ for two discrete random variables X and Y is: $I(X;Y) = \sum_{x \in X}\sum_{y \in Y} P(x,y)\log\left(\frac{P(x,y)}{P(x)P(y)}\right)$

where $P(x,y)$ is the joint Probability of X and Y, and $P(x)$ and $P(y)$ are the marginal Probabilities

'Prize'

① Calculate totals:

$150 + 10 + 1000 + 15000 = 16160$

② Marginal Probability:

$P(spam=1) = \frac{160}{16160}$ $\quad$ $P(spam=0)\ \frac{16000}{16160}$

$P(Prize=1) = \frac{1150}{16160}$ $\quad$ $P(prize=0) = \frac{15010}{16160}$

③ Joint Probabilities

$P(spam=1, prize=1) = \frac{150}{16160}$ $\quad$ $P(spam=1, prize=0) = \frac{10}{16160}$

$P(spam=0, Prize=1) = \frac{1000}{16160}$ $\quad$ $P(spam=0, prize=0) = \frac{15000}{16160}$

Put it all together

$$I(spam; prize) = \sum p(spam, prize) \log \frac{P(spam, Prize)}{P(spam)\,P(price)}$$

'hello'

① Calculate Total:

$145 + 15 + 11000 + 5000 = 16160$

② Marginal Probabilities:

$P(spam=1) = \frac{160}{16160}$ $\quad$ $p(spam=0) = \frac{16000}{16160}$

$P(hello=1) = \frac{11145}{16160}$ $\quad$ $P(hello=0) = \frac{5015}{16160}$

③ Joint

$P(spam=1, hello=1) = \frac{145}{16160}$ $\quad$ $P(spam=1, hello=0) = \frac{15}{16160}$

$P(spam=0, hello=1) = \frac{11000}{16160}$ $\quad$ $P(spam=0, hello=0) = \frac{5000}{16160}$

Put it all together

$$I(spam; hello) = \sum p(spam, hello) \log \frac{P(spam, hello)}{P(spam)\,P(hello)}$$

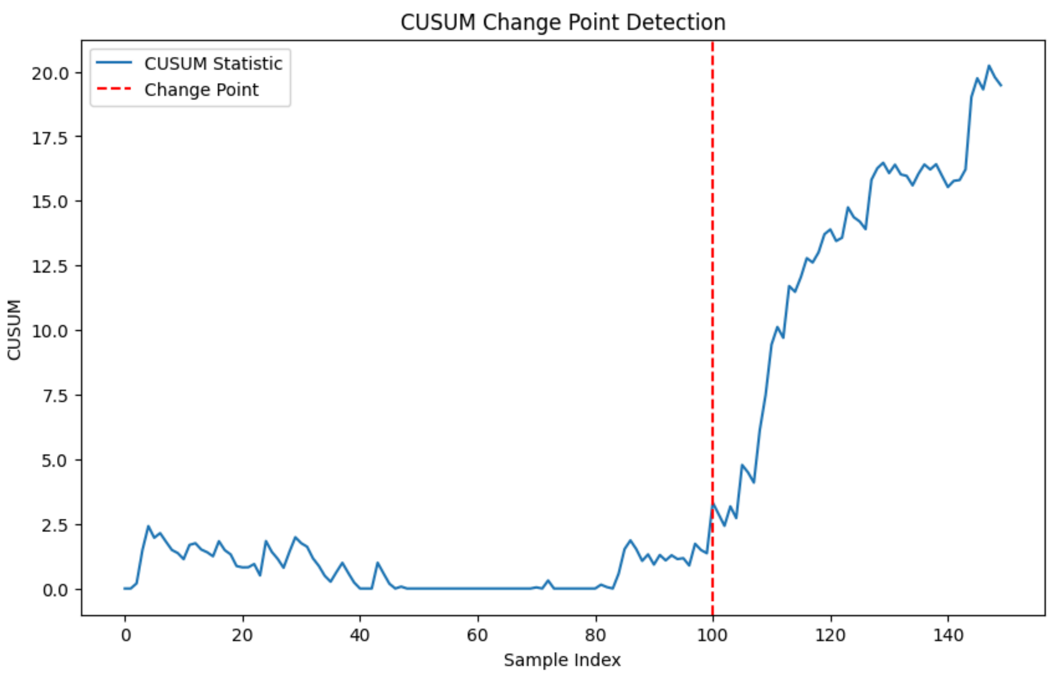
From calculation we know that

"Prize" $\approx 0.0228 > 0.0014$

"hello" ≈ 0.0014

$\therefore$ 'Prize' is more informative to decide if it's spam.

b) In the second part of the analysis, we applied the CUSUM (Cumulative Sum) detection statistic to identify a change point in a sequence of samples. The samples were generated from two different normal distributions: $f0 = \mathrm{N}(0, 1)$ for the first 100 samples and $f1 = \mathrm{N}(0.5, 1.5)$ for the subsequent 50 samples. The CUSUM algorithm involves calculating the log-likelihood ratio (LLR) for each sample and using these values to compute the CUSUM statistic recursively. The plot of the CUSUM statistic clearly shows a significant increase starting at the 100th sample, indicating the change point where the distribution shifts from $f0$ to $f1$. The plot confirms the effectiveness of the CUSUM method in identifying changes in distribution, which is crucial for applications requiring quick detection of shifts in process behavior.



Question 5. Medical imaging reconstruction

Both methods have their strengths and weaknesses. LASSO regression is advantageous for sparse signal recovery and can be particularly useful when the true image is expected to be sparse. On the other hand, Ridge regression provides a smoother and less noisy reconstruction, which may be more suitable for images where smoothness is a key feature. In this case, while both methods provided reasonable reconstructions, the Ridge regression approach produced a clearer and more coherent image.