3F8: Inference
Short Lab Report

Author's Name

February 18, 2025

Abstract

This is the abstract.
Try for 1-2 sentences on each of: motive (what it's about), method (what was done), key results and conclusions (the main outcomes).

- Don't exceed 3 sentences on any one.
- Write this last, when you know what it should say!

1 Introduction

1. What is the problem and why is it interesting?
2. What novel follow-up will the rest of your report present?

2 Exercise a)

In this exercise we have to consider the logistic classication model (aka logistic regression) and derive the gradients of the log-likelihood given a vector of binary labels $\mathbf{y}$ and a matrix of input features $\mathbf{X}$. The gradient of the log-likelihood can be writen as

$$\begin{aligned}\frac{\partial \mathcal{L}(\beta)}{\partial \beta} &= \frac{\partial}{\partial \beta}\prod_{n=1}^{N} log\{\sigma(\beta^T\tilde{x}^{(n)})^{y^{(n)}}(1-\sigma(\beta^T\tilde{x}^{(n)}))^{1-y^{(n)}}\} \\ &= \sum_{n=1}^{N}\{y^{(n)}\frac{\partial}{\partial\beta}log\{\sigma(\beta^T\tilde{x}^{(n)})\} + (1-y^{(n)})\frac{\partial}{\partial\beta}log\{1-\sigma(\beta^T\tilde{x}^{(n)})\}\}\end{aligned}$$

Using derivative results (substituting $z = \beta^T\tilde{x}^{(n)}$):

$$\begin{aligned}\frac{\partial}{\partial z}\log\sigma(z) &= \frac{1}{\sigma(z)}\sigma(z)(1-\sigma(z)) = 1-\sigma(z) \\ \frac{\partial}{\partial z}\log(1-\sigma(z)) &= \frac{1}{1-\sigma(z)}(-\sigma(z)(1-\sigma(z))) = -\sigma(z)\end{aligned}$$

We can then write:

$$\begin{aligned}\frac{\partial \mathcal{L}(\beta)}{\partial \beta} &= \sum_{n=1}^{N}\{y^{(n)}(1-\sigma(\beta^T\tilde{x}^{(n)}))\tilde{x}^{(n)} - (1-y^{(n)})\sigma(\beta^T\tilde{x}^{(n)})\tilde{x}^{(n)}\} \\ &= \sum_{n=1}^{N}\{y^{(n)} - \sigma(\beta^T\tilde{x}^{(n)})\}\tilde{x}^{(n)}\end{aligned}$$

Figure 1: Visualisation of the data.

Figure 2: Learning curves showing the average log-likelihood on the training (left) and test (right) datasets.

3 Exercise b)

In this exercise we are asked to write pseudocode to estimate the parameters β using gradient ascent of the log-likelihood. Our code should be vectorised. The pseudocode to estimate the parameters β is shown below:

```
Function estimate_parameters:

   Input:  feature matrix X, labels y
   Output: vector of coefficients b

    max_iter = 1000 # maximum number of iterations set to avoid infinite training
    eta = 0.01 # learning rate (choice explained below)
    b = zeros(D) # initialise coefficients vector with dimention D
    for iter in range(max_iter):
        grad_old = X.T @ (y - sigmoid(X @ b)) # calculate old gradient
        b_new = b + eta * grad_old # gradient ascent
        if norm(b_new - b) == 0: # check whether has reached local maximum
            break
        b = b_new
   return b
```

The learning rate parameter η is chosen to avoid both divergence due to too large η and slow convergence due to too small η. Therefore, it is here set to the widely accepeted default value of 0.01.

4 Exercise c)

In this exercise we visualise the dataset in the two-dimensional input space displaying each datapoint's class label. The dataset is visualised in Figure 1. By analising Figure 1 we conclude that a linear classifier...

5 Exercise d)

In this exercise we split the data randomly into training and test sets with 800 and 200 data points, respectively. The pseudocode from exercise a) is transformed into python code as follows:

```
#
# Python code to be included
#
```

We then train the classifier using this code. We fixed the learning rate parameter to be $\eta = \ldots$. The average log-likelihood on the training and test sets as the optimisation proceeds are shown in Figure 2. By looking at these plots we conclude that ...

Figure 2 displays the visualisation of the contours of the class predictive probabilities on top of the data. This figure shows that...

Figure 3: Visualisation of the contours of the class predictive probabilities.

Avg. Train ll	Avg. Test ll
-	-

Table 1: Average training and test log-likelihoods.

		$\hat{y}$	
		0	1
y	0	-	-
	1	-	-

Table 2: Confusion matrix on the test set.

Figure 4: Visualisation of the contours of the class predictive probabilities for $l = 0.01$ (left), $l = 0.1$ (middle), $l = 1$ (right).

6 Exercise e)

The final average training and test log-likelihoods are shown in Table 1. These results indicate that... The 2x2 confusion matrices on the and test set is shown in Table 2. By analising this table, we conclude that...

7 Exercise f)

We now expand the inputs through a set of Gaussian radial basis functions centred on the training datapoints. We consider widths $l = \{0.01, 0.1, 1\}$ for the basis functions. We fix the learning rate parameter to be $\eta = \{..., ..., ...\}$ for each $l = \{0.01, 0.1, 1\}$, respectively. Figure 4 displays the visualisation of the contours of the resulting class predictive probabilities on top of the data for each choice of $l = \{0.01, 0.1, 1\}$.

8 Exercise g)

The final final training and test log-likelihoods per datapoint obtained for each setting of $l = \{0.01, 0.1, 1\}$ are shown in tables 3, 4 and 5. These results indicate that... The 2×2 confusion matrices for the three models trained with $l = \{0.01, 0.1, 1\}$ are show in tables 6, 7 and 8. After analysing these matrices, we can say that... When we compare these results to those obtained using the original inputs we conclude that...

9 Conclusions

1. Draw together the most important results and their consequences.
2. List any reservations or limitations.

Avg. Train ll	Avg. Test ll
-	-

Table 3: Results for $l = 0.01$

Avg. Train ll	Avg. Test ll
-	-

Table 4: Results for $l = 0.1$

Avg. Train ll	Avg. Test ll
-	-

Table 5: Results for $l = 1$

		$\hat{y}$	
		0	1
y	0	-	-
	1	-	-

Table 6: Conf. matrix $l = 0.01$.

		$\hat{y}$	
		0	1
y	0	-	-
	1	-	-

Table 7: Conf. matrix $l = 0.1$.

		$\hat{y}$	
		0	1
y	0	-	-
	1	-	-

Table 8: Conf. matrix $l = 1$.