COMP4434 Big Data Analytics

Assignment 2 PolyU, Hong Kong

9 March, 11:55 PM.

**Problem 1 (2 points)**

Please answer the following questions briefly.

1. In the classification task, why do we prefer logistic loss over mean-squared error?

Logistic loss function is defined as

Mean Square Error (MSE) is defined as

We prefer the logistic loss over mean-square error because logistic loss outputs probabilities, where MSE does not, which is preferred for classification tasks. Logistic loss is also convex, so there is only 1 minimum, whereas MSE is non-convex, and would have multiple local minimums, so the gradient descent algorithm may converge and be stuck in a local minimum.

1. How does a standard Support Vector Machine (SVM) differ from an SVM with a soft margin, and in what way do slack variables (facilitate the handling of non-linearly separable data?

SVM is a supervised algorithm used for classification tasks. The algorithm aims to find a hyperplane that can separate linearly separate data while maximizing the margin . The formula of SVM is as follows:

However, sometimes the classes of data that we use would not be linearly separable – i.e. cannot be perfectly separated by a single line. This means that it cannot satisfy the constraint of , causing it to misclassify data that are non-linearly separable.

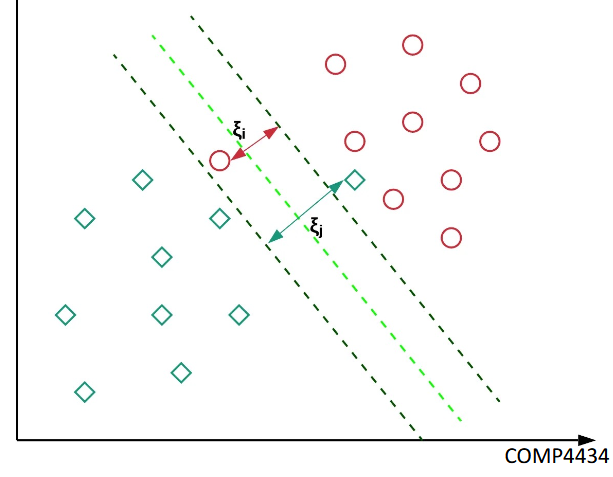
In contrast, SVM with a soft margin adds a slack variable to the formula to handle cases where there are data that cannot be linearly separable. The slack variable helps relax the constraint of SVM, allowing some data points to violate the margin or be misclassified. Thus, the formula becomes

Subject to:

Where the magnitude of determines the degree of misclassification:

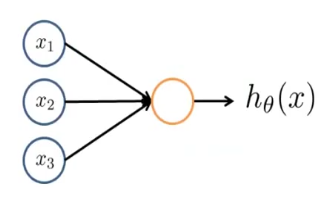
* The point is correctly classified outside the margin.
* The point lies inside the margin but on the correct side.
* : The point is misclassified.

The regularization Parameter controls the trade-off between margin maximization and slack penalty. A large C prioritizes minimizing misclassifications (narrower margin, risk of overfitting), while a small C allows more slack (wider margin, better generalization).



1. If your multi-layer perceptron model has an overfitting issue, what are the strategies that you could use to handle the issue?

The perceptron is a binary linear classifier where is the input node and is the output



The formula is as follows:

In a multi-layer perceptron layer network, overfitting, where the model matches the training data too closely, can be handled by using different techniques such as feature reduction, where you reduce the number of features in the data using domain specific knowledge. The second strategy that you can use is L1(Ridge) or L2(LASSO) regularisation, which reduces their influence by giving smaller values to the weights, which is useful when there are many features to account for in the dataset. The regularisation techniques are added into the loss function to constraint the weight magnitudes to reduce the complexity of the models.

L1, or LASSO regularisation, has the formula

Which can lead to sparsity, where some weights become 0.

L2, or ridge regularisation, has the formula

Which distributes the errors among the features, making them smaller with non-zero features. L1 is more useful when there are more features and where only some features are important, and L2 is more useful when most features are useful. Adding dropout to the model is also potentially useful by randomly deactivating neurons during training to improve generalisation.

Another method that we can use is cross fold validation techniques such as k-fold technique to ensure the model would generalise across different data splits.

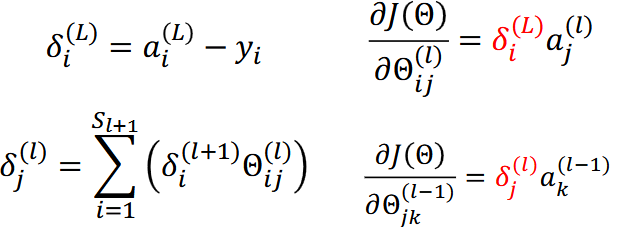
Some other techniques such as data augmentation and batch normalisation could potentially be useful too.

1. How does backpropagation use the gradient descent algorithm to update the weights in a neural network?

Backpropagation works by employing gradient descent to update the weights of the neural network by computing the gradient of the loss function with respect to each weight and adjusting the weight to reduce error.

In a neural network, the input data is passed through the network, computing the activations using the weights and functions. The loss or error is then calculated through MSE or logistic loss, then we find the difference between the predicted output and the actual target. This is called the forward pass.

Then, backpropagation is done through calculating the gradient of the loss with respect to the output layer’s pre-activations or weight, then for the hidden layers as well, going back each layer for each step. This is done using the chain rule.



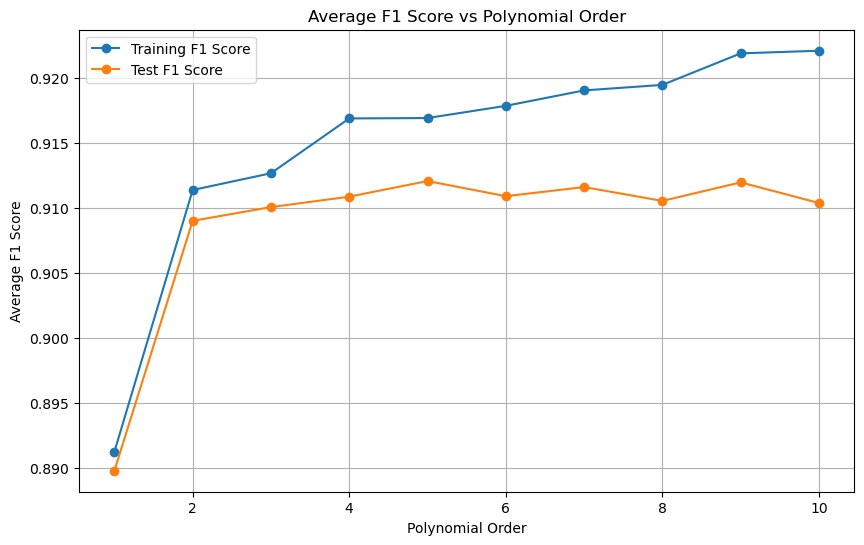
After finding the gradient, we update the weight by subtracting the weight with the learning rate multiplied by the gradient.

**Problem 2 (5 points)**

Load dataset in problem2data.txt by using “numpy.loadtxt()” in Python, which contains two synthetic classes separated by a non-linear function, with additive noise. The last column contains “0” or “1”, indicating the corresponding classes. You are asked to investigate the extent to which polynomial functions can be used to build a logistic regression classifier.

(a) Build a logistic regression classifier by using a polynomial function of order 1 (e.g., θ0 + θ1x1 + θ2x2 +θ3x3). Apply 5-fold-cross-validation to generate training and test sets. Use the training set to train the model. Compute its five F1 scores on test set (one F1 score for each fold). Compute its five F1 scores on training set. You will need “f1 score” from “sklearn.metrics”.

(b) Repeat part (a) for polynomials of order 2 to 10. You will need “PolynomialFeatures” from “sklearn.preprocessing”.

(c) Repeat parts (a-b) 20 times, and estimate the average F1 score for each polynomial order (average across its 5 × 20 runs, both for training and test sets). Generate a plot that shows the F1 scores versus the polynomial order. 

[0.89126033 0.91139814 0.91268994 0.91690569 0.91693499 0.91787389

0.91906119 0.91948549 0.92191177 0.92211545]

(d)Discuss how the F1 scores of the model changes as a function of the polynomial order.

Generally, it could be said that the higher the polynomial order, the higher the F1 score. The dataset shows a vast increase initially from polynomial orders 1-4 for the F1 score, suggesting that the model’s performance in capturing the structure of the dataset has improved. However, at degrees 5-10, the increase of the F1 score diminishes until improvement is minimal. The small increase in F1 scores at higher polynomial orders suggest that the model may overfit in those degrees, which is proven as the Test F1 score is comparatively lower at higher orders than Training F1 scores. As such, the polynomial order of 4-5 may be best for the model where a balance of performance and complexity can be achieved.

Use Jupyter Notebook to perform implementation. You are required to submit your “.ipynb” file. You could use “LogisticRegression” from “sklearn.linear model”. Do not add any regularizations.

**Problem 3 (2 points)**

Consider a set of data points represented by the coordinates (x, y). The objective is to apply the k-means algorithm to identify two distinct clusters within the data. Use (0.5, 1) as the initial centroid for the first cluster and (3, 4) as the initial centroid for the second cluster. Perform a single iteration of the k-means algorithm with k = 2, utilizing cosine distance as the distance metric. Recall that cosine distance is defined as 1 - cosine similarity. The data points are listed as follows.

|  |  |  |
| --- | --- | --- |
| Data# | x | Y |
| 1 | 1 | 0.5 |
| 2 | 0.5 | 2 |
| 3 | 3 | 1 |
| 4 | 2 | 1.5 |

1. Determine the cluster assignments for each data point after one iteration.

Cluster 1 centroid

Cluster 2 centroid

Data point 1: (1, 0.5)

Distance to c1:

Distance to c2:

Data point 2: (0.5,2)

Distance to c1:

Distance to c2:

Data point 3: (3,1):

Distance to c1:

Distance to c2:

Data point 4: (2,1.5):

Distance to c1:

Distance to c2:

Cluster Assignments (closest distance):

1. Compute the new centroids for each cluster in Euclidean space after completing the first iteration.

Cluster 1:

Cluster 2: