FAI HW3

B09901142 EE4 呂睿超

May 2024

Handwritten Part

Problem 1

The error function is defined as:

$$E_{\text{in}}(\mathbf{w}) = \frac{1}{N}\sum_{n=1}^{N}\left(\max(1 - y_n\mathbf{w}^T\mathbf{x}_n, 0)\right)^2$$

Considering only the case where $1 - y_n\mathbf{w}^T\mathbf{x}_n > 0$:

$$\nabla E_{\text{in}}(\mathbf{w}) = \nabla\left(\frac{1}{N}\sum_{n=1}^{N}\left(1 - y_n\mathbf{w}^T\mathbf{x}_n\right)^2\right)$$

$$= \frac{1}{N}\sum_{n=1}^{N}\nabla\left(\left(1 - y_n\mathbf{w}^T\mathbf{x}_n\right)^2\right)$$

Using the chain rule:

$$= \frac{1}{N}\sum_{n=1}^{N}2\left(1 - y_n\mathbf{w}^T\mathbf{x}_n\right)\left(-y_n\mathbf{x}_n\right)$$

$$= -\frac{2}{N}\sum_{n=1}^{N}\left(1 - y_n\mathbf{w}^T\mathbf{x}_n\right)y_n\mathbf{x}_n$$

Problem 2

Given the problem:

$$\mathbf{u}^* = \arg\max_{\mathbf{u}\in\mathbb{R}^d} \prod_{n=1}^{N} p_{\mathbf{u}}(\mathbf{x}_n), \quad \mathbf{x}_n \text{ from multivariate Gaussian } \mathcal{N}(\mathbf{u}, \mathbf{I})$$

This can be rewritten as:

$$= \arg\max_{\mathbf{u}\in\mathbb{R}^d} \prod_{n=1}^{N} \left(\frac{1}{(2\pi)^{(\frac{d}{2})}} \exp\left(-\frac{1}{2}(\mathbf{x}_n - \mathbf{u})^T(\mathbf{x}_n - \mathbf{u}) \right) \right)$$

Assume the right-hand side as $\arg\max L(\mathbf{u})$:

$$\log L(\mathbf{u}) = \log \left(\left(\frac{1}{(2\pi)^{(\frac{d}{2})}} \right)^N \times \exp\left(-\frac{1}{2} \sum_{n=1}^{N} (\mathbf{x}_n - \mathbf{u})^T(\mathbf{x}_n - \mathbf{u}) \right) \right)$$

This simplifies to:

$$= N \log \left(\frac{1}{(2\pi)^{(\frac{d}{2})}} \right) - \frac{1}{2} \sum_{n=1}^{N} (\mathbf{x}_n - \mathbf{u})^T(\mathbf{x}_n - \mathbf{u})$$

Now, take the derivative of $\log L(\mathbf{u})$ with respect to $\mathbf{u}$:

$$\frac{\partial \log L(\mathbf{u})}{\partial \mathbf{u}} = -\frac{1}{2} \left(-2 \sum_{n=1}^{N} (\mathbf{x}_n - \mathbf{u}) \right)$$

$$= \sum_{n=1}^{N} (\mathbf{x}_n - \mathbf{u})$$

To find the maximum, set the derivative to zero:

$$\frac{\partial \log L(\mathbf{u})}{\partial \mathbf{u}} = \sum_{n=1}^{N} (\mathbf{x}_n - \mathbf{u}^*) = 0$$

$$\sum_{n=1}^{N} \mathbf{x}_n - N\mathbf{u}^* = 0$$

$$\mathbf{u}^* = \frac{1}{N} \sum_{n=1}^{N} \mathbf{x}_n$$

Problem 3

The second-order feature transform is given by:

$$\Phi_2(\mathbf{x}) = [1, x_1, x_2, x_1x_2, x_1^2, x_2^2]$$

Applying this transformation to the data set, we get:

$$\mathbf{z}_1 = \Phi_2(\mathbf{x}_1) = [1, 1, 1, 1, 1, 1], \quad y_1 = -1$$

$$\mathbf{z}_2 = \Phi_2(\mathbf{x}_2) = [1, -1, 1, -1, 1, 1], \quad y_2 = 1$$

$$\mathbf{z}_3 = \Phi_2(\mathbf{x}_3) = [1, -1, -1, 1, 1, 1], \quad y_3 = -1$$

$$\mathbf{z}_4 = \Phi_2(\mathbf{x}_4) = [1, 1, -1, -1, 1, 1], \quad y_4 = 1$$

Choosing the weight vector:

$$\tilde{\mathbf{w}} = [-1, 0, 0, 4, -1, -1]$$

The classification boundary is:

$$-1 + 4(x_1x_2) - x_1^2 - x_2^2 = 0$$

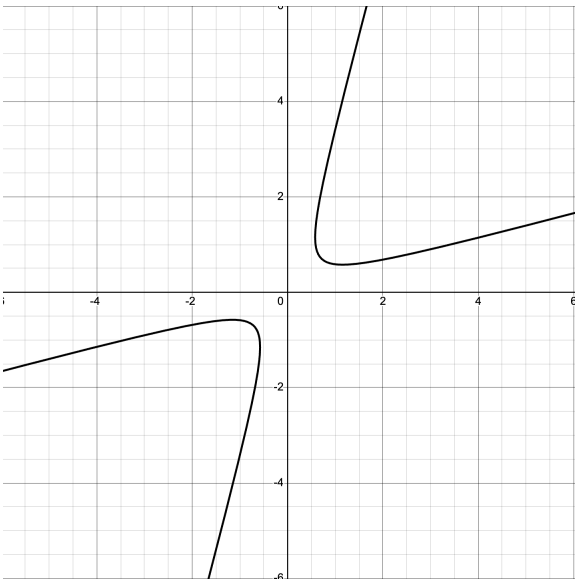

Figure 1: Classification Boundary

Problem 4

Given:

$$\epsilon_t = \frac{\sum_{n=1}^{N} w_n^t \delta(g_t(\mathbf{x}_n), y_n)}{\sum_{n=1}^{N} w_n^t}$$

And:

$$w_n^{t+1} = \begin{cases} w_n^t \cdot d_t & \text{if } g_t(\mathbf{x}_n) \neq y_n \\ w_n^t / d_t & \text{if } g_t(\mathbf{x}_n) = y_n \end{cases}$$

We have:

$$\sum_{n=1}^{N} w_n^{t+1} = \sum_{g_t(\mathbf{x}_n) \neq y_n} w_n^t \cdot d_t + \sum_{g_t(\mathbf{x}_n) = y_n} \frac{w_n^t}{d_t}$$

We know:

$$\epsilon_t = \frac{\sum_{n=1}^{N} w_n^t \delta(g_t(\mathbf{x}_n), y_n)}{\sum_{n=1}^{N} w_n^t} = \frac{\sum_{g_t(\mathbf{x}_n) \neq y_n} w_n^t}{\sum_{n=1}^{N} w_n^t}$$

So:

$$1 - \epsilon_t = \frac{\sum_{g_t(\mathbf{x}_n) = y_n} w_n^t}{\sum_{n=1}^{N} w_n^t}$$

Thus:

$$\sum_{n=1}^{N} w_n^{t+1} = d_t \epsilon_t \sum_{n=1}^{N} w_n^t + \frac{1}{d_t}(1 - \epsilon_t) \sum_{n=1}^{N} w_n^t$$

Putting in $d_t = \sqrt{\frac{1-\epsilon_t}{\epsilon_t}}$:

$$\sum_{n=1}^{N} w_n^{t+1} = \sum_{n=1}^{N} w_n^t \left(\sqrt{\frac{1-\epsilon_t}{\epsilon_t}} \epsilon_t + \sqrt{\frac{\epsilon_t}{1-\epsilon_t}} (1 - \epsilon_t) \right)$$

Simplifying:

$$\sum_{n=1}^{N} w_n^{t+1} = 2 \sum_{n=1}^{N} w_n^t \sqrt{\epsilon_t (1 - \epsilon_t)}$$

New error rate:

$$\epsilon_{t+1} = \frac{\sum_{n=1}^{N} w_n^{t+1} \delta(g_t(\mathbf{x}_n), y_n)}{\sum_{n=1}^{N} w_n^{t+1}}$$

Substitute the expressions:

$$\epsilon_{t+1} = \frac{\sum_{g_t(\mathbf{x}_n) \neq y_n} w_n^t d_t}{2 \sum_{n=1}^{N} w_n^t \sqrt{\epsilon_t (1 - \epsilon_t)}}$$

Using $d_t = \sqrt{\frac{1-\epsilon_t}{\epsilon_t}}$:

$$\epsilon_{t+1} = \frac{\epsilon_t \sum_{n=1}^{N} w_n^t \sqrt{\frac{1-\epsilon_t}{\epsilon_t}}}{2 \sum_{n=1}^{N} w_n^t \sqrt{\epsilon_t (1 - \epsilon_t)}}$$

Simplifying:

$$\epsilon_{t+1} = \frac{\epsilon_t \sqrt{1-\epsilon_t}}{2\sqrt{\epsilon_t(1-\epsilon_t)}}$$

$$\epsilon_{t+1} = \frac{1}{2}$$

Thus, the new error rate is 0.5.

Programming Part Report

Problem (a)

1. Results as following
   (a) Classification Task (Accuracy)
      i. Logistic Regression Accuracy: 0.9333
      ii. Decision Tree Classifier Accuracy: 0.8889
      iii. Random Forest Classifier Accuracy: 0.9333
   (b) Regression Task (MSE)
      i. Linear Regression MSE: 63.0710
      ii. Decision Tree Regressor MSE: 28.3416
      iii. Random Forest Regressor MSE: 24.3225
   (c) Analysis
      i. Classification Task:
      Logistic Regression: This is a linear model and it performs well when the relationship between the features and the target is approximately linear. It seems to have performed well in this case.

      Random Forest Classifier: This is an ensemble of decision trees and generally performs well across various datasets due to its ability to handle complex relationships and reduce overfitting by averaging multiple trees.

      Decision Tree Classifier: Although powerful, individual decision trees can overfit to the training data, leading to slightly lower performance compared to the Random Forest which mitigates overfitting.
      ii. Regression Task:
      Random Forest Regressor: This model performs the best because it combines multiple decision trees and averages their predictions, reducing the variance and capturing more complex patterns in the data.

      Decision Tree Regressor: It performs better than Linear Regression because it can model nonlinear relationships, but it is still prone to overfitting, especially if not pruned properly.

      Linear Regression: This model assumes a linear relationship between features and target, which might not be the case in the dataset, leading to higher errors.

Problem (b)

1. Results as following (experiment done on logistic regression)
   (a) **Standardized** Logistic Regression Accuracy: 0.8889
   (b) **Normalized** Logistic Regression Accuracy: 0.9333
2. Analysis
   (a) Normalization
      i. Advantages: Helps in scenarios where features are on different scales and you want them to contribute equally.
      ii. Disadvantages: Sensitive to outliers as they can drastically affect the min and max values. Does not handle the data's variance and distribution properties.
   (b) Standardization
      i. Advantages: Reduces the impact of outliers.
      Features become comparable on the same scale, improving algorithm performance.
      ii. Disadvantages: Does not bound the data to a specific range, which might not be suitable for some models like neural networks requiring inputs between 0 and 1.

Problem (c)

1. Results as following

Configuration	**Learning Rate**	**Iterations**	**Accuracy**
Configuration 1	0.01	100	0.7556
Configuration 2	0.01	1000	0.9333
Configuration 3	0.01	10000	1.0
Configuration 4	0.1	1000	1.0
Configuration 5	0.001	1000	0.7556

Table 1: Performance of Logistic Regression with Different Hyperparameters

2. Analysis
   (a) Discussion on Learning Rate and Iterations
      i. Learning Rate: A high learning rate (e.g., 0.1) allows the model to converge faster but risks overshooting the optimal point, which can lead to divergence or suboptimal performance if not balanced correctly. A low learning rate (e.g., 0.001) ensures more stable convergence but requires significantly more iterations to reach the optimal point.
      ii. Number of Iterations: More iterations (e.g., 10000) generally allow the model to fully converge, especially with lower learning rates, leading to better performance. Fewer iterations (e.g., 100) may not be sufficient for the model to converge, particularly with moderate to low learning rates, resulting in suboptimal performance.

Problem (d)

1. Analysis
   (a) Number of Trees (n_estimators)
      i. Model Complexity: Increasing the number of trees generally improves the model's performance by reducing variance through averaging, leading to better generalization.
      ii. Generalization: More trees typically enhance the model's ability to generalize to unseen data, as the aggregation of multiple trees smooths out the predictions.
      iii. Overfitting: While a single decision tree is prone to overfitting, a random forest mitigates this by **averaging multiple trees**.
   (b) Maximum Depth (max_depth)
      i. Model Complexity: The maximum depth controls the complexity of the trees. Deeper trees can capture more details from the data, but this also increases the risk of overfitting.
      ii. Overfitting: Deeper trees are more likely to overfit, especially on smaller datasets. Setting a maximum depth helps in creating a more balanced model that generalizes well.

Problem (e)

1. Trade-offs Between Model Complexity, Interpretability, and Performance Model Complexity
   (a) Logistic Regression: Low complexity, suitable for large datasets and situations where interpretability is essential.
   (b) Decision Tree: Moderate complexity, can become complex if trees are deep. Random Forest: High complexity, involves training multiple trees and aggregating their results.
2. Interpretability
   (a) Logistic Regression: High interpretability; coefficients can be directly interpreted.
   (b) Decision Tree: Moderate interpretability; the model can be visualized as a series of rules.
   (c) Random Forest: Low interpretability; difficult to explain the combined effect of multiple trees.
3. Performance
   (a) Logistic Regression: Performs well on linear data but poorly on complex, nonlinear relationships.
   (b) Decision Tree: Can handle nonlinear data but prone to overfitting.
   (c) Random Forest: High performance on both linear and nonlinear data; reduces overfitting by averaging multiple trees.

Reference

ChatGPT