BMEG3105 - Fall 2024

Lecture 10: Performance Evaluation

Lecture Date: 09 Oct. Deadline: 16 Oct. 11:59 p.m.

Lecturer: Prof. LI Yu *Scribe: LIU Linqi*

1 Recap from Last Lecture

1.1 K-Nearest Neighbors (KNN)

1.1.1 Challenges and K Selection

1. KNN requires storing all data points and calculating distances, which is time-consuming, space-consuming and computationally expensive.
2. To select an optimal K, part of the training data is used as testing data, and average accuracy across folds is computed.

1.1.2 Standard Procedure

1. Choose a distance metric and K value.
2. Normalize data and compute distances.
3. Find the K nearest neighbors and predict the class based on the majority vote.

1.2 Logistic Regression

1.2.1 Logistic Function

The logistic function is defined as:

$$\sigma(t) = \frac{1}{1+e^{-t}}, \quad t = w_h H + w_w W + w_0 \tag{1}$$

where H and W are height and weight, respectively.

1.2.2 Gradient Descent Algorithm

The gradient descent algorithm is used to minimize the loss function by iteratively adjusting the weights in the direction that reduces error.

1. **Initialization**:

- Initialize weights w_h, w_w, and w_0 to random values.

2. **Iteration Over Data Points**:
   - For each data point P_i in the dataset (e.g., P_1, P_2, P_3, P_4):
     (a) **Calculate the Output**:

$$Y_{\text{output}} = \frac{1}{1 + e^{-(w_h H + w_w W + w_0)}}$$

     where H and W represent the input features (e.g., height and weight).
     (b) **Update the Weights**:

$$w_i = w_i + \Delta w_i$$

$$\Delta w_i = \alpha \cdot (Y_{\text{true}} - Y_{\text{output}}) \cdot \frac{\partial Y_{\text{output}}}{\partial w_i}$$

     Here:
     - α is a small constant known as the learning rate.
     - Y_{true} is the true label for the data point.
     - $\frac{\partial Y_{\text{output}}}{\partial w_i}$ is the partial derivative of the output with respect to the weight w_i.

3. **Repeat the Above Steps**:
   - Continue iterating over all data points and updating weights until the change in the weights is negligible, indicating convergence.

The objective of the gradient descent algorithm is to find the set of weights w_h, w_w, and w_0 that minimizes the loss function L, thereby improving model accuracy.

1.2.3 LR, NN and Deep Learning

- **From LR to NN**:
  - Neural networks (NN) extend logistic regression (LR) by incorporating multiple interconnected neurons.
  - Enable faster predictions and better tolerance to noisy data, making NNs successful in real-life applications.
  - Drawbacks: NNs typically require longer training times and has poor interpretability.
- **From NN to Deep Learning**:
  - Deep learning builds on neural networks by adding more layers and complex architectures.
  - Example: *AlphaFold* is the most successful deep learning application.

2 Performance Evaluation

2.1 Perpose of Model Evaluation

1. Pinpoint the strong points and weak points of a method.
2. Method selection / Model selection.

2.2 Confusion Matrix

The confusion matrix contains:

- **True Positive (TP)**: The number of instances correctly predicted as positive by the model.
- **False Positive (FP)**: The number of instances incorrectly predicted as positive when they are actually negative.
- **True Negative (TN)**: The number of instances correctly predicted as negative by the model.
- **False Negative (FN)**: The number of instances incorrectly predicted as negative when they are actually positive.

2.3 Accuracy

$$\text{Accuracy} = \frac{TP + TN}{TP + TN + FP + FN} \tag{2}$$

Limitations: Accuracy can be misleading for imbalanced classes.

2.4 Precision, Recall, and F1 Score

2.4.1 Precision

$$\text{Precision} = \frac{TP}{TP + FP} \tag{3}$$

Definition: Indicates how many of the predicted positive samples are actually correct.

2.4.2 Recall

$$\text{Recall} = \frac{TP}{TP + FN} \tag{4}$$

Definition: Indicates how many actual positive samples are correctly predicted by the model, reflects the model's ability to identify all relevant instances.

2.4.3 F1 Score

$$F1 = 2 \times \frac{\text{Precision} \times \text{Recall}}{\text{Precision} + \text{Recall}} \tag{5}$$

Definition: The weighted average of precision and recall.

These three evaluation matrices may still be misleading for imbalanced data.

2.5 Balanced Accuracy

$$\text{Balanced Accuracy} = \frac{1}{2}\left(\frac{TP}{TP+FN} + \frac{TN}{TN+FP}\right) \tag{6}$$

Balanced accuracy helps in evaluating models on imbalanced datasets.

2.6 Binary Classification Evaluation

Value is not absolute, instead, context matters. Example: rare cancer pre-screening.

3 Cross-Validation

3.1 n-Fold Cross-Validation

Idea: train multiple times, leaving out a disjoint subset of data each time for validation. Average the validation set accuracies.

- **Process:**
  1. Randomly partition the dataset into n disjoint subsets (folds).
  2. For each fold i (from 1 to n):
     - Set the i-th subset as the validation data.
     - Train the classifier h on the remaining $n-1$ subsets.
     - Evaluate the accuracy of the trained classifier h on the i-th validation subset, denoted as Accuracy(i).
- **Final Accuracy:**

$$\text{Final Accuracy} = \frac{1}{n}\sum_{i=1}^{n} \text{Accuracy}(i) \tag{7}$$

  The final accuracy is computed as the mean of the n accuracies obtained from each fold.

3.2 Leave-One-Out Cross-Validation

Idea: a special case of n-fold cross-validation, where n = N.

- **Process:**
  1. Partition the dataset into N disjoint subsets, where each subset contains exactly one data point.
  2. For each data point i (from 1 to N):
     - Set the i-th data point as the validation data.
     - Train the classifier h on all data points except the i-th point.
     - Evaluate the accuracy of the trained classifier h on the i-th validation point, denoted as Accuracy(i).
- **Final Accuracy:**

$$\text{Final Accuracy} = \frac{1}{N}\sum_{i=1}^{N}\text{Accuracy}(i) \tag{8}$$

  The final accuracy is computed as the mean of the N accuracies obtained from each data point serving as a validation set.

4 Multi-Class Classification

For KNN, there is no need to change the algorithm; for logistic regression, handling multi-class classification requires building a separate logistic regression model for each class. During prediction, the class with the highest predicted value is assigned.

4.1 Macro-Averaging and Micro-Averaging

For multi-class classification, metrics are extended as follows:

$$\text{Macro-Averaged Accuracy} = \frac{1}{k}\sum_{i=1}^{k}\text{Accuracy}_i \tag{9}$$

$$\text{Micro-Averaged Accuracy} = \frac{\sum_{i=1}^{k}(TP_i + TN_i)}{\sum_{i=1}^{k}(TP_i + TN_i + FP_i + FN_i)} \tag{10}$$

5 Clustering Evaluation

5.1 Rand Index

The Rand Index is a measure used to evaluate the similarity between two clusterings by considering all pairs of data points. It is calculated as follows:

$$R = \frac{a+d}{a+b+c+d} \tag{11}$$

where a is the number of correctly clustered pairs, and d is the number of correctly separated pairs.

$$R = \frac{a+d}{\text{Total Number of Pairs}} = \frac{a+d}{\binom{n}{2}} = \frac{a+d}{\frac{n(n-1)}{2}} \tag{12}$$

where n is the total number of data points, and $\binom{n}{2} = \frac{n(n-1)}{2}$ is the number of all possible pairs of data points in the dataset.

5.2 Confusion Matrix

		Predicted clusters	
Actual clusters		The same	Not the same
	The same	a(TP)	b(FN)
	Not the same	c(FP)	d(TN)

Figure 1: Confusion Matrix for Clustering [1]

6 Python Implementation and Resources

For practical implementation, Python's scikit-learn library offers various tools for classification and clustering performance evaluation. Recommended resources include:

- `classification_report` for classification metrics.
- `cross_validate` for model evaluation.

References

[1] Li, Yu (2024). *BMEG3105: Data Analytics for Personalized Genomics and Precision Medicine - Clustering and Classification Performance Evaluation*.