BMEG3105 Data Analytics for Personalized Genomics and Precision Medicine

Lecture 10: Classification + Performance Evaluation

Wednesday, 8th October 2025

Lecturer : Yu LI (李煜)

Scriber : VERMINSHU, Britney Alexandra (1155206525)

Agenda(s) :

1	Recap from last lecture
2	Logistic regression
3	Logistic regression model training
4	From logistic regression to neural networks
5	Performance evaluation

1. Recap

a. Classification

- Method to assign the class of previously unseen records based on the other attributes and the training set.

b. K-nearest neighbor classification

- Classification method by identifying the K most similar data to find the mode class and assign it to the unclassified data.
- KNN procedure:
  - Normalize all data to be on the same scale.
  - Compute distances between the data points.
  - Identify the K-nearest neighbors.
  - Assign the most frequent class among them to the test record.
- Example: predicting the gender of a new person given the records of other individuals' height, weight, and gender.

c. Clustering vs Classification

- Clustering groups similar data; classification assigns a class to new data.
- Unknown number of classes when doing clustering; known number of classes for classification.

2. Logistic regression

a. *Problems with KNN*

- Storage: need spaces to store all data.

- Computation: calculating the distance matrix for each data point is time consuming, especially for large data sets.

b. *Solution*

- Use a formula to simplify the class prediction of a new data, e.g., if height + weight ≥ 0.5 → male (faster prediction process).
- Since different attributes (e.g., height/H and weight/W) may have different importance, the formula can be adjusted with weights (w_1 & w_2) and bias (w_0).
- Using the previous example, the formula is adjusted into:

$$H + W \geq 0.5 \rightarrow w_h H + w_w W + w_0 \geq 0.5$$

- When w_1, w_2, and w_0 are large (overflow), use the logistic function to control the output to be between 0 and 1, defined through the previous example as:

$$\frac{1}{1+e^{-t}} \rightarrow \frac{1}{1+e^{-(w_h H + w_w W + w_0)}} \geq 0.5$$

- To do classification with logistic function:
    - Training: fit the training data that performs well to get the weights and bias.
    - Testing: run the formula to classify the testing data

3. Logistic regression model training

- Using the previous example, let the output of the logistic function be Y^{output}, where the output 1 → male and 0 → female.

a. *Loss function*

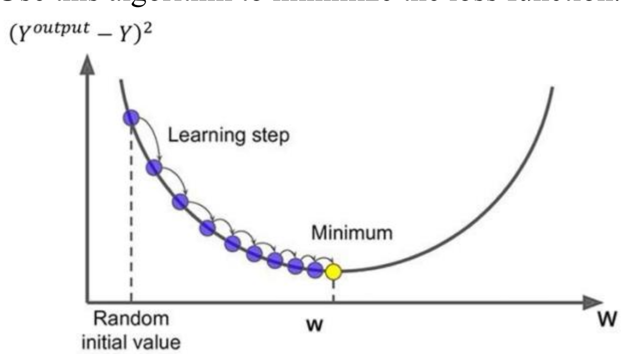
- Fit the model to the training data: by minimizing loss function $(Y^{output} - Y)^2$, where Y is the true label for the training data.
- Loss function measures the error between the model's prediction and the actual value.
- Example: for a dataset of height, weight, and gender of x individuals:

$$L = \sum_{P1}^{Px} (Y^{output} - Y)^2$$

where L is a function of ws, and the goal is to find the ws to make L the smallest.

b. *Gradient descent algorithm*

- Use this algorithm to minimize the loss function.



- Steps:
    - Initialize random values to the weights and bias.
    - Calculate the output value (e.g., Y^{output}).
    - Update weights using $w_i = w_i + \Delta w_i$, where Δw_i is proportional to the error.
    - Repeat these steps until the model converges (i.e., the value doesn't decrease significantly).

4. From logistic regression (LR) to neural networks (NN)

- Neural networks extend logistic function by stacking multiple layers of calculations.
- Advantages: fast prediction and high tolerance to noisy data, making NN successful in real-life applications.
- Limitations: NNs require long training time and has poor interpretability.
- From NN to Deep Learning, deep learning builds on neural networks by adding more layers and complex architectures, e.g., AlphaFolds (most successful deep learning application).

5. Performance evaluation

- Purpose: to characterize the performance of a model (i.e., pinpoint the strong and weak points) with quantitative values (method/model selection).

a. *Confusion matrix*

- Contains TP/True Positive (correctly predicted as positive), TN/True Negative (correctly predicted as negative), FP/False Positive (incorrectly predicted as positive), and FN/False Negative (incorrectly predicted as negative).
- Accuracy (most widely-used metric), represented as:

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

  *) Accuracy can be misleading for imbalanced data.

Disclaimer: all figures are adapted from Prof. Li BMEG3105 Lecture 10 Notes