BMEG3105: Data Analytics for Personalized Genomics and Precision Medicine

Lecture 9: Classification

Logistic Regression

Why logistic regression?

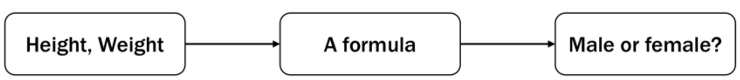
- KNN has some problems:
  - Need to store **all the data**
  - Need to calculate the **distance matrix**
  - Prediction is **slow**

  } Imagine that we have **one million** data points!!
- There is no need to calculate the distance matrix
  → Getting results with **a simple arithmetic calculation**

How to get a formula?

For example, we will use the same data that we previously classified P5 by KNN.

Person	Height	Weight	Gender
P1	0.625	0.875	M
P2	0	0	F
P3	0.25	0.375	M
P4	1	1	M
P5	0.4583	0.6667	??



1. It seems that if **H+W is large**, the person is very likely to be a **male**.
   Let's say if $H + W \geq 0.5$ → Male
   Thus, P5: $0.4583 + 0.6667 = 1.125 \geq 0.5$ → Male

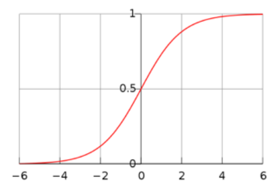
2. However, each attribute **might not be equally important**, and it **might not be 0.5** as well.
   So, we need to add **weights** and **bias** → w_h, w_w, and w_0 should be inferred from the training data.
   → At first, we use observation. → Then, we use mathematical calculation.

   $$H + W \geq 0.5 \rightarrow \text{Male} \quad \Longrightarrow \quad w_h H + w_w W + w_0 \geq 0.5$$

3. What if w_h, w_w, and w_0 are **large**?
   Then, we will adjust the formula to **logistic function**.

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

4. There are **two steps** for classification by using logistic function:
   - Training → fit the training data to get w_h, w_w, and w_0
   - Testing → run the formula to classify the unknown



$\frac{1}{1+e^{-t}} \geq 0.5$

<u>How to train the model?</u>

To fit the model to the training data → We are trying to make $\frac{1}{1+e^{-(w_h H + w_w W + w_0)}} \geq 0.5$ correct for the training data.

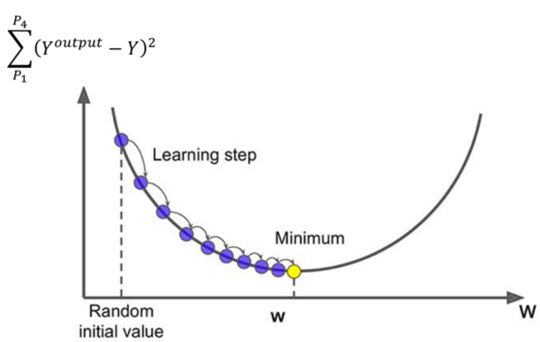
So, let $Y^{output} = \frac{1}{1+e^{-(w_h H + w_w W + w_0)}}$ → male => 1 and female => 0 as **Ground truth labels**,

and **Loss function** = $(Y^{output} - Y)^2$ which we would like to **minimize**.

For instance, P1 has loss function $(Y^{output} - Y)^2 = \left(1 - \frac{1}{1+e^{-(0.625 * w_h + 0.875 * w_w + w_0)}}\right)^2$

Total loss of the logistic function is $L = \sum_{P_1}^{P_4} (Y^{output} - Y)^2$, and our goal is to find *w*s that make L the **smallest**.

In order to find the **minimum value**, it is like in the **calculus**.
For each *w*, we want to find a value that makes function value the **smallest**.



→ Gradient descent algorithm

1. Initially, we random the values of w_h, w_w, and w_0.
2. We calculate Y^{output} and loss function of the first data point.
3. We update the weights.

$$w_i = w_i + \Delta w_i$$
$$\Delta w_i = 2 * \alpha (Y - Y^{output}) \frac{\partial Y^{output}}{\partial w_i}$$
α is a small constant

4. Repeat step 2-3 for every data point until there is no more updates.