Scribing 2 for lecture 9

Chan Chun Ming Tony
1155156523

Scribing agenda

Logistic regression

Logistic regression model training

From logistic regression to neural networks

Performance evaluation

- Binary Classification Evaluation

Logistic regression

The problem of KNN

Suppose we have chosen the Euclidean distance and K=2

Person	Height (m)	Weight (kg)	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	??

Person	P5	Gender
P1	0.267	M
P2	0.809	F
P3	0.358	M
P4	0.636	M
P5	0	??

- Need to store all the data
- Need to calculate the distance matrix

- Predicting is slow

What if we have a formula?

(Height, Weight) → (A formula) → (Male or Female?)

No need to calculate the distance matrix Getting the results with simple arithmetic calculation

Person	Height (m)	Weight (kg)	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	??

It seems if H+W is large, the person is very likely to be a Male

$$H + W \geq 0.5 \rightarrow \text{Male}$$

$$\text{P5: } 0.4583 + 0.6667 = 1.125 \geq 0.5 \rightarrow \text{Male}$$

Adjust the formula

- Different attributes may not be equally important
- May not be 0.5
- Add weights and bias

- w_h and w_w, and w_0 should be inferred from the training data
- Observation → mathematical calculation

$$H + W \geq 0.5 \rightarrow \text{Male}$$

$$\Downarrow$$

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

Logistic function

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

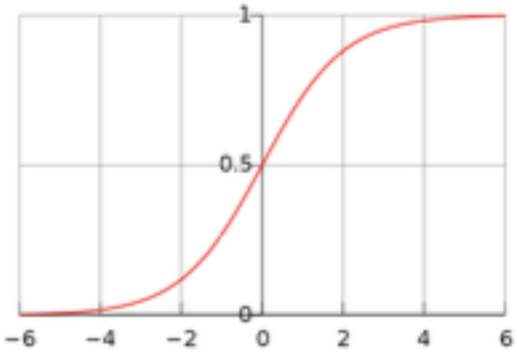
What is w_h, w_w and w_0 are large?

$$\frac{1}{1 + e^{-(whH + wwW + w0)}} \geq 0.5$$

Training: fit the training data

- To get w_h, w_w and w_0

Testing: run the formula Classification



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

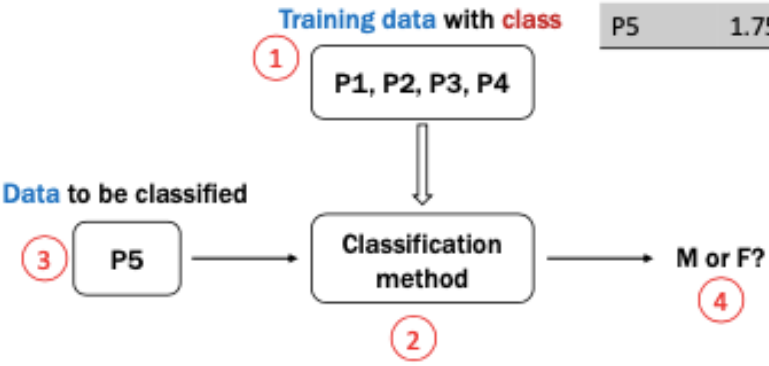
A working model

$$\frac{1}{1+e^{-(whH + wwW + w0\,)}} \geq 0.5$$

$w_h = 1$, $w_w = 1$ and $w_0 = -0.5$

How to do classification?

Person	Height(m)	Weight(kg)	Gender
P1	1.79	75	M
P2	1.64	54	F
P3	1.70	63	M
P4	1.88	78	M
P5	1.75	70	??



Logistic regression model training

How to train?

Training

- To get w_h, w_w and w_0

To make the model fit the training data

Make $\frac{1}{1+e^{-(whH + wwW + w0)}} \geq 0.5$ correct for the training data

$Y^{output} = \frac{1}{1+e^{-(whH + wwW + w0)}} \geq 0.5$

- 1 for male, 0 for female

Loss function

$(Y^{output} - Y)^2$ should be as small as possible

- Y: the true label we have for training data
- Loss function that we would like to minimize

$(Y^{output} - Y)^2$ is a function of ws

$Y^{output} = \frac{1}{1+e^{-(whH + wwW + w0)}}$

For P1

$(Y^{output} - Y)^2 = (1 - \frac{1}{1+e^{-(0.625*wh + 0.875ww + w0)}})^2$

$L = \sum_{P1}^{P4} (Y^{output} - Y)^2$ is a function of ws

- Goal: find ws to make L the smallest

Find out the minimum value

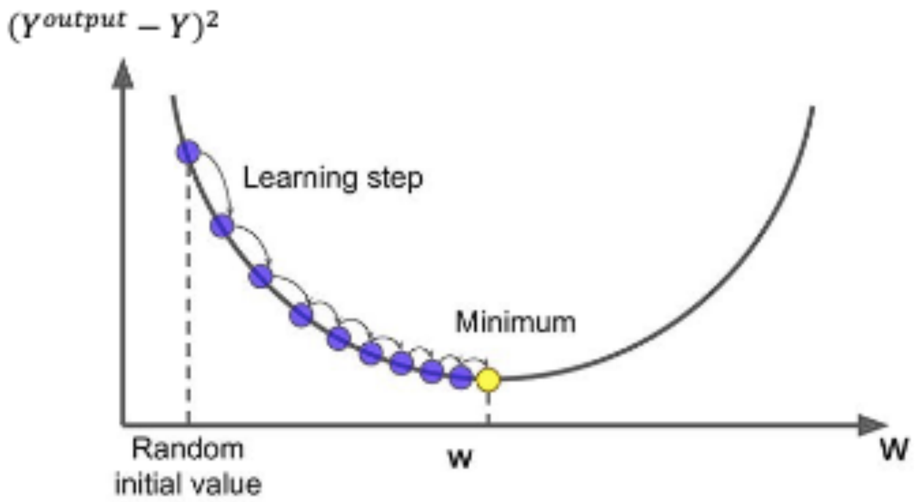
Calculus

What if the equation is not easy to resolve?

Gradient descent algorithm

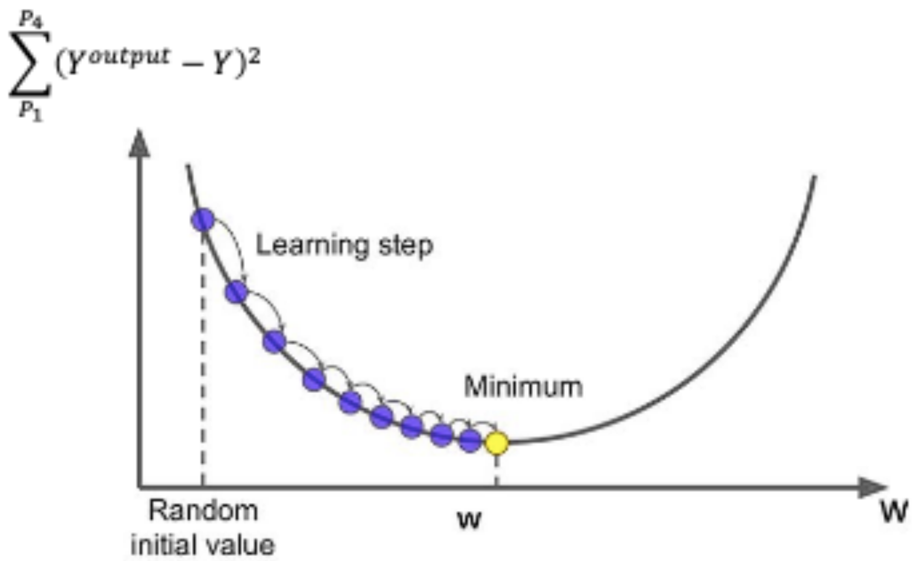
$(Y^{output} - Y)^2$ is a function of ws

For each w, we want to find a value to make the function value smallest



$L = \sum_{P1}^{P4} (Y^{output} - Y)^2$ is a function of *ws*

For each *w*, we want to find a value to make the function value smallest



To get the formula

Initialize w_h, w_w and w_0

- Random values

For P1, P2, P3, P4

- Calculate the output Y^{output}
- Update weights
  - $w_i = w_i + \Delta w_i$
  - $\Delta w_i = 2 * \alpha(Y - Y^{output})(\partial Y^{output} / \partial w_i)$

- α is a small constant

Repeat the above step

- Until no more to update

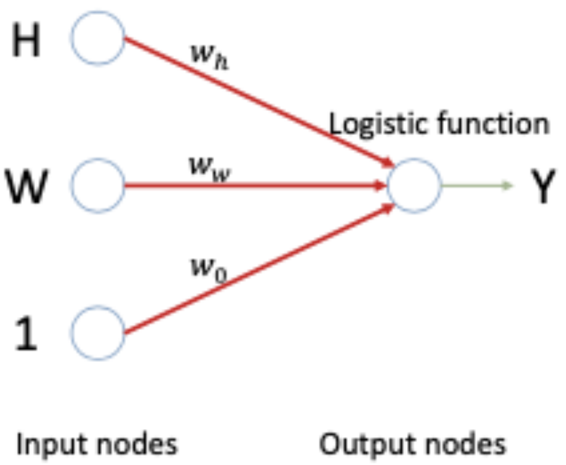
What are good w_h, w_W and w_0 ?

- The ones make $(Y^{output} – Y)^2$ the smallest

From logistic regression to neural networks

The simplest neural network

$$\frac{1}{1 + e^{-(whH + wwW + w0)}} \geq 0.5$$



$$Y^{output} = \frac{1}{1+e^{-(whH + wwW + w0)}}$$

From LR to NN

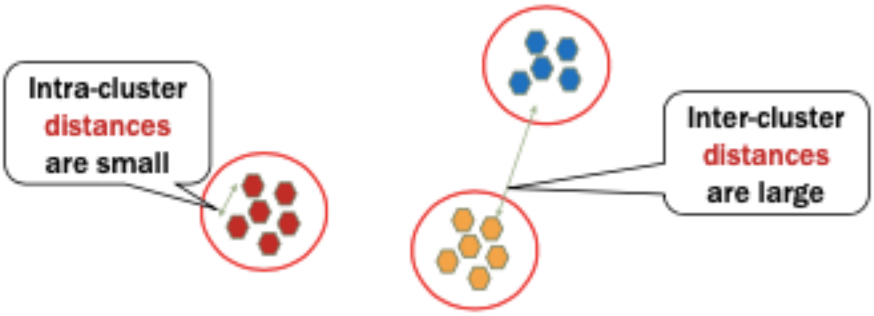
- Fast prediction
- Successful in real-life problems
- High tolerance to noisy data
- Long training time
- Poor interpretability

The most successful deep learning application

AlphaFold

Performance evaluation

Which clustering method is better?



Which classification method should we trust?

We need some quantitative values to summarize the performance of different methods

The purpose of model evaluation

Characterize the performance of a model

- Pinpoint the strong points and weak points of a method
- Method selection/Model selection

Classification performance evaluation

- Confusion matrix

	Predicted class		
Actual class		Class = Yes	Class = No
	Class = Yes	a (TP)	b (FN)
	Class = No	c (FP)	d (TN)

- TP: True Positive
- TN: True Negative
- FP: False Positive
- FN: False Negative

Person	Height (m)	Weight (kg)	Male?	Prediction
P1	1.79	75	Yes	Yes
P2	1.64	54	No	No
P3	1.70	63	Yes	No
P4	1.88	78	Yes	Yes
P5	1.75	70	Yes	No
P6	1.65	52	No	Yes

Most widely-used metric:

$$\text{Accuracy} = \frac{a+d}{a+b+c+d} = \frac{\text{TP}+\text{TN}}{\text{TP}+\text{TN}+\text{FP}+\text{FN}}$$

Accuracy Example:

	Predicted class		
Actual class		Class = Yes	Class = No
	Class = Yes	45 (TP)	4 (FN)
	Class = No	6 (FP)	45 (TN)

$$\text{Accuracy} = \frac{\text{TP}+\text{TN}}{\text{TP}+\text{TN}+\text{FP}+\text{FN}} = \frac{45+45}{45+45+4+6} = 0.9$$

What if we have a bad classifier and predict everything as Yes?

	Predicted class		
Actual class		Class = Yes	Class = No
	Class = Yes	49 (TP)	0 (FN)
	Class = No	51 (FP)	0 (TN)

$$\text{Accuracy} = \frac{\text{TP}+\text{TN}}{\text{TP}+\text{TN}+\text{FP}+\text{FN}} = \frac{49}{49+51} = 0.49$$

Accuracy: limitation

	Predicted class		
Actual class		Class = Yes	Class = No
	Class = Yes	4949 (TP)	0 (FN)
	Class = No	51 (FP)	0 (TN)

Imbalanced classes

$$\text{Accuracy} = \frac{\text{TP}+\text{TN}}{\text{TP}+\text{TN}+\text{FP}+\text{FN}} = \frac{4949}{4949+51} = 0.99$$

Maybe misleading for imbalanced data

Precision, recall, and F1 score

	Predicted class		
Actual class		Class = Yes	Class = No
	Class = Yes	a (TP)	b (FN)
	Class = No	c (FP)	d (TN)

$$\text{Precision} = \frac{a}{a + c}$$

$$Recall = \frac{a}{a + b}$$

$$F1\ score = \frac{2 * \text{precision} * Recall}{\text{Precision} + Recall}$$

Among the predicted positive samples, how many of them are correct?

How many actual positive samples are predicted to be positive?

The weighted average of precision and recall

Precision, recall, and F1 score: Example

	Predicted class		
Actual class		Class = Yes	Class = No
	Class = Yes	4949(TP)	0(FN)
	Class = No	51(FP)	0(TN)

$$\text{Precision} = \frac{a}{a + c} = \frac{4949}{4949 + 51} = 0.99 \qquad Recall = \frac{a}{a + b} = 1$$

$$F1\ score = \frac{2 * \text{precision} * Recall}{\text{Precision} + Recall} = 0.995$$

Still maybe misleading for imbalanced data

Balanced accuracy

	Predicted class		
Actual class		Class = Yes	Class = No
	Class = Yes	4949(TP)	0(FN)
	Class = No	51(FP)	0(TN)

$$Balanced\ accuracy = 0.5 * (\frac{TP}{TP + FN} + \frac{TN}{TN + FP}) = 0.5$$

Imbalanced dataset → Confusion matrix directly

Binary classification evaluation

	Predicted class		
Actual class		Class = Yes	Class = No
	Class = Yes	2 (TP)	0 (FN)

	Class = No	50 (FP)	50 (TN)

Value is not absolute. Context matters.