Multiclass Classification

Swati Mishra

Applications of Machine Learning (4AL3)

Fall 2024

McMaster University

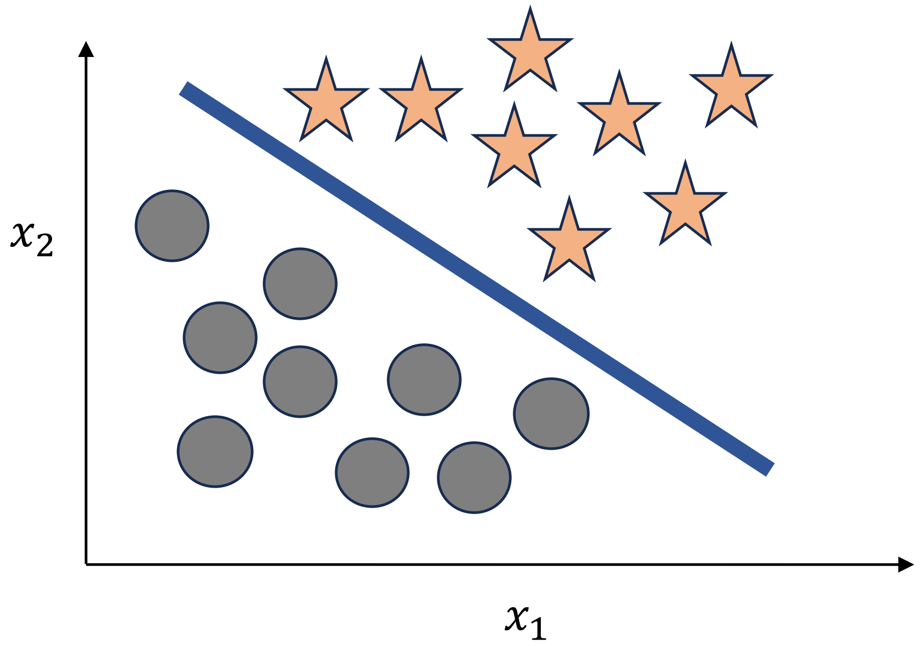
ENGINEERING

Review

- Data Model vs Concept Model
- Correlations – Compute, Visualize, Decide
- Data Cleaning , Feature Scaling
- Creating Test Sets - Random 80-20 Split, Stratified Split

Classification

- Binary Classification – 2 classes

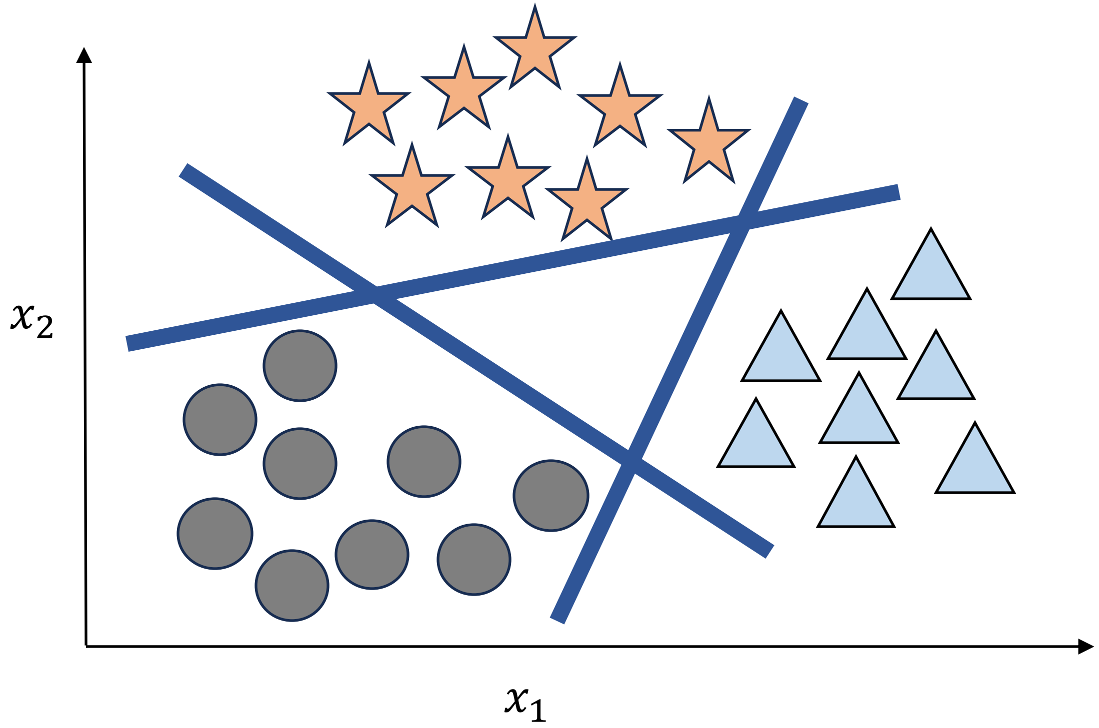


Example Tasks:

- Storm/No storm
- Buy/Sell
- Lend/don’t lend

Classification

- Multiclass Classification – More than 2 classes



Example Tasks:

- Dog, Cat, Tiger, Wolf
- Politics, Sports, Entertainment
- Positive, negative, neutral

Binary Logistic Regression - Review

- Logistic Regression **Model**: Given a set of features (x) of a data instance, compute the probability of the instance belonging to class 1 (or 0).

$$P(Y|x) = \begin{cases} p(x) & , if\ \mathrm{Y} = 1 \\ 1 - p(x) & , if\ Y = 0 \end{cases}$$

- Logistic Regression **Equation** is

$$p(Y = 1|x) = \sigma(b + \boldsymbol{W}.x)$$

- Training objective is to learn parameters W and b to maximize the log probability of correct label $p(Y \middle| x)$ using training dataset.

- Logit is called Log ratio of probability $\quad log\ (\frac{p(x)}{1 - p(x)})$

Binary Logistic Regression - Review

- Logistic Regression **Model:** Given a set of features (x) of a data instance, compute the probability of the instance belonging to class 1 (or 0).

$$P(Y|x) = \begin{cases} p(x) & , if\ Y = 1 \\ 1 - p(x) & , if\ Y = 0 \end{cases}$$

- Logistic Regression **Equation** is

$$p(Y = 1|x) = \sigma(b + \boldsymbol{W}.x)$$

- Training objective is to learn parameters W and b to maximize the log probability of correct label $p(Y|x)$ using training dataset.

- Logit is called Log ratio of probability $\quad log\ (\frac{p(x)}{1 - p(x)})$

McMaster University

Binary Logistic Regression - Review

- Logistic Regression **Model:** Given a set of features (x) of a data instance, compute the probability of the instance belonging to class 1 (or 0).

$$P(Y|x) = \begin{cases} p(x) & , if\ Y = 1 \\ 1 - p(x) & , if\ Y = 0 \end{cases}$$

- Logistic Regression **Equation** is

$$p(Y = 1|x) = \sigma(b + \boldsymbol{W}.x)$$

- Training objective is to learn parameters W and b to maximize the log probability of correct label $p(Y|x)$ using training dataset.

- Logit is called Log ratio of probability $log\ (\frac{p(x)}{1 - p(x)})$

McMaster University

Binary Logistic Regression - Review

- Logistic Regression **Model:** Given a set of features (x) of a data instance, compute the probability of the instance belonging to class 1 (or 0).

$$P(Y|x) = \begin{cases} p(x) & , if\ Y = 1 \\ 1 - p(x) & , if\ Y = 0 \end{cases}$$

- Logistic Regression **Equation** is

$$p(Y = 1|x) = \sigma(b + \boldsymbol{W}.x)$$

- Training objective is to learn parameters W and b to maximize the log probability of correct label $p(Y|x)$ using training dataset.

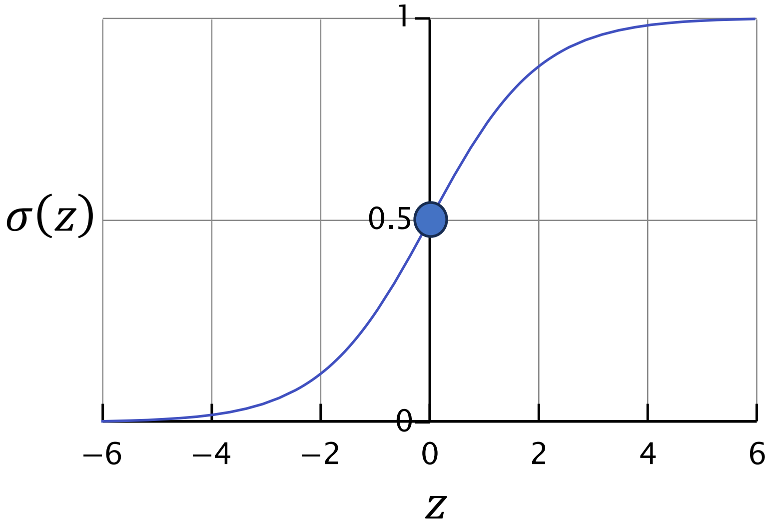
- Logit is called Log ratio of probability $\quad log\ (\frac{p(x)}{1 - p(x)})$

Binary Logistic Regression - Review

- Logistic function is

$$\sigma(z) = \frac{1}{1 + e^{-z}}$$

- It is also called **sigmoid** function
  - If z is large, $\sigma(z) \rightarrow 1$
  - If z is small, $\sigma(z) \rightarrow 0$



Threshold = 0.5

- After applying logistic regression function,

$$P(y = 1) = \frac{1}{1 + e^{-(b+\mathbf{w}.\mathbf{x})}}$$

Picture Source: https://en.wikipedia.org/wiki/Sigmoid_function

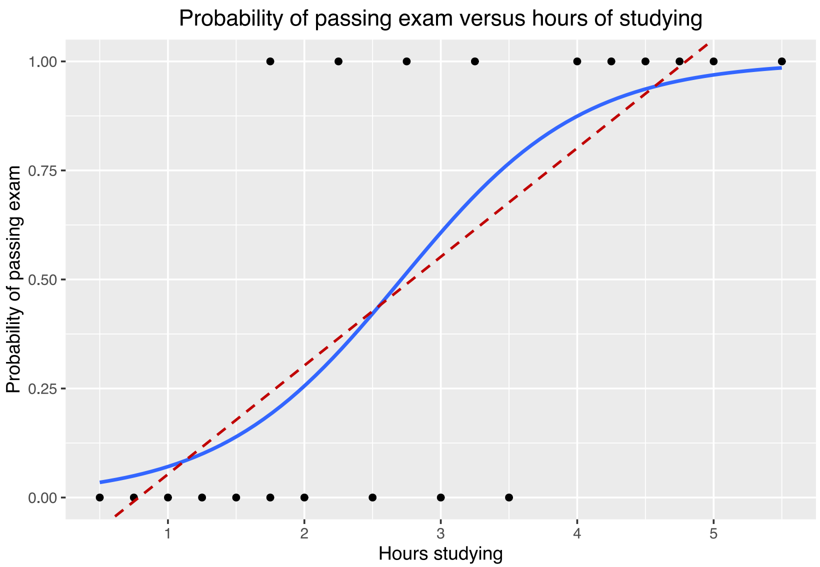
Binary Logistic Regression - Review

x_k	0.50	0.75	1.00	1.25	1.50	1.75	1.75	2.00	2.25	2.50	2.75	3.00	3.25	3.50	4.00	4.25	4.50	4.75	5.00	5.50
y_k	0	0	0	0	0	0	1	0	1	0	1	0	1	0	1	1	1	1	1	1

- Logistic function is

$$\sigma(z) = \frac{1}{1 + e^{-z}}$$

- It is also called **sigmoid** function
  - If z is large, $\sigma(z) \rightarrow 1$
  - If z is small, $\sigma(z) \rightarrow 0$



x_k = Hours Studied
y_k = Will Pass

Logistic regression does better than linear regression!

- After applying logistic regression function,

$$P(y = 1) = \frac{1}{1 + e^{-(b+\mathbf{w}.\mathbf{x})}}$$

Picture Source: https://en.wikipedia.org/wiki/Logistic_regression

McMaster University

Binary Logistic Regression -Review

Binary Logistic Regression with multiple features : $p(Y = 1|x) = \sigma(b + \boldsymbol{W}.x)$

$$p(Y = 1|\boldsymbol{X}) = \sigma(\boldsymbol{b} + \boldsymbol{W}.\boldsymbol{X}) = \sigma(b + w_1\, x_{i1} + w_2\, x_{i2} + w_3\, x_{i3} + \ldots + w_m\, x_{in})$$

m = number of observations
n = number of features

Binary Logistic Regression -Review

Binary Logistic Regression with multiple features : $p(Y = 1|x) = \sigma(b + \boldsymbol{W}.x)$

$$p(Y = 1|\boldsymbol{X}) = \sigma(\boldsymbol{b} + \boldsymbol{W}.\boldsymbol{X}) = \sigma(b + w_1\ x_{i1} + w_2\ x_{i2} + w_3\ x_{i3} + \ldots + w_m\ x_{in})$$

... and with multiple observations :

$$y = \begin{vmatrix} y_1 \\ y_2 \\ \vdots \\ y_m \end{vmatrix} \qquad \boldsymbol{X} = \begin{vmatrix} x_{11} & x_{12} \ldots & x_{1n} \\ x_{21} & x_{22} \ldots & x_{2n} \\ \vdots & \vdots \quad \ldots & \vdots \\ x_{m1} & x_{m2} \ldots & x_{mn} \end{vmatrix} \qquad \boldsymbol{W} = \begin{vmatrix} w_1 \\ w_2 \\ \vdots \\ w_m \end{vmatrix} \qquad \boldsymbol{b} = \begin{vmatrix} b_1 \\ b_2 \\ \vdots \\ b_m \end{vmatrix}$$

$$m * 1 \qquad\qquad \mathrm{m} * n \qquad\qquad m * 1 \qquad\qquad m * 1$$

m = number of observations
n = number of features

Binary Logistic Regression -Review

Binary Logistic Regression with multiple features : $p(Y = 1|x) = \sigma(b + \boldsymbol{W}.x)$

$$p(Y = 1|\boldsymbol{X}) = \sigma(\boldsymbol{b} + \boldsymbol{W}.\boldsymbol{X}) = \sigma(b + w_1\ x_{i1} + w_2\ x_{i2} + w_3\ x_{i3} + \ldots + w_m\ x_{in})$$

... and with multiple observations :

$$y = \begin{vmatrix} y_1 \\ y_2 \\ \vdots \\ y_m \end{vmatrix} \qquad \boldsymbol{X} = \begin{vmatrix} x_{11} & x_{12}\ldots & x_{1n} \\ x_{21} & x_{22}\ldots & x_{2n} \\ \vdots & \vdots\ \ \ldots & \vdots \\ x_{m1} & x_{m2}\ldots & x_{mn} \end{vmatrix} \qquad \boldsymbol{W} = \begin{vmatrix} w_1 \\ w_2 \\ \vdots \\ w_m \end{vmatrix} \qquad \boldsymbol{b} = \begin{vmatrix} b_1 \\ b_2 \\ \vdots \\ b_m \end{vmatrix}$$

$$m * 1 \qquad \mathrm{m} * n \qquad m * 1 \qquad m * 1$$

m = number of observations
n = number of features

$b_1 = b_2 = b_3 = \ldots = b_m$
b is same for all observations

Binary Logistic Regression -Review

Training logistic regression requires Cross Entropy Loss Function which is defined as below :

$$L_i = -(y_i log(\sigma(\boldsymbol{b} + \boldsymbol{W}.\boldsymbol{X}) \; + \; (1 - y_i) \, log(\, 1 - \sigma(\boldsymbol{b} + \boldsymbol{W}.\boldsymbol{X})))$$

- If the predicted label is wrong the loss is large
- If the predicted label is right the the loss is small.

- Training goal is to minimize the average loss

- Why entropy? CE Loss measures the number of bits we send.

Binary Logistic Regression -Review

Training logistic regression requires Cross Entropy Loss Function which is defined as below :

$$L_i = -(y_i log(\sigma(\boldsymbol{b} + \boldsymbol{W}.\boldsymbol{X}) \ + \ (1 - y_i) \ log(\ 1 - \sigma(\boldsymbol{b} + \boldsymbol{W}.\boldsymbol{X})))$$

- If the predicted label is wrong the loss is large
- If the predicted label is right the the loss is small.

- Training goal is to minimize the average loss which is given by $\frac{1}{n}\sum_{i=1}^{n} L_i$

- Why entropy? CE Loss measures the number of bits we send.

Binary Logistic Regression -Review

Training logistic regression requires Cross Entropy Loss Function which is defined as below :

$$L_i = -(y_i log(\sigma(\boldsymbol{b} + \boldsymbol{W}.\boldsymbol{X}) \; + \; (1 - y_i) \; log(\; 1 - \sigma(\boldsymbol{b} + \boldsymbol{W}.\boldsymbol{X})))$$

- If the predicted label is wrong the loss is large
- If the predicted label is right the the loss is small.

- Training goal is to minimize the average loss which is given by $\frac{1}{n}\sum_{i=1}^{n} L_i$

- Why entropy? CE Loss measures the number of bits we send.

Binary Logistic Regression

What does output of Sigmoid function look like:

Let us consider features:

- Count of positive words
- Count of negative words

Let us consider labels:

- Positive
- Negative

- Input vector

Data instance 1: "4AL3 is awesome and wonderful"

Feature Vector for 1 instance = [2,0]

Label Vector for 1 instance = [1]

$w_1 = w_1 = b = 0$

$\eta = 0.1$

$$\nabla L = \begin{vmatrix} (\sigma(b + \boldsymbol{W}.x) - y)x_1 \\ (\sigma(b + \boldsymbol{W}.x) - y)x_2 \\ (\sigma(b + \boldsymbol{W}.x) - y) \end{vmatrix} = \begin{vmatrix} (\sigma(0) - 1)x_1 \\ (\sigma(0) - 1)x_2 \\ (\sigma(0) - 1) \end{vmatrix}$$

$\beta' = \beta$ - $\eta \nabla L$

Binary Logistic Regression -Review

Training logistic regression requires Cross Entropy Loss Function which is defined as below :

$$L_i = -(y_i log(\sigma(\boldsymbol{b} + \boldsymbol{W}.\boldsymbol{X}) \ + \ (1 - y_i) \ log(1 - \sigma(\boldsymbol{b} + \boldsymbol{W}.\boldsymbol{X})))$$

- If the predicted label is wrong the loss is large
- If the predicted label is right the the loss is small.

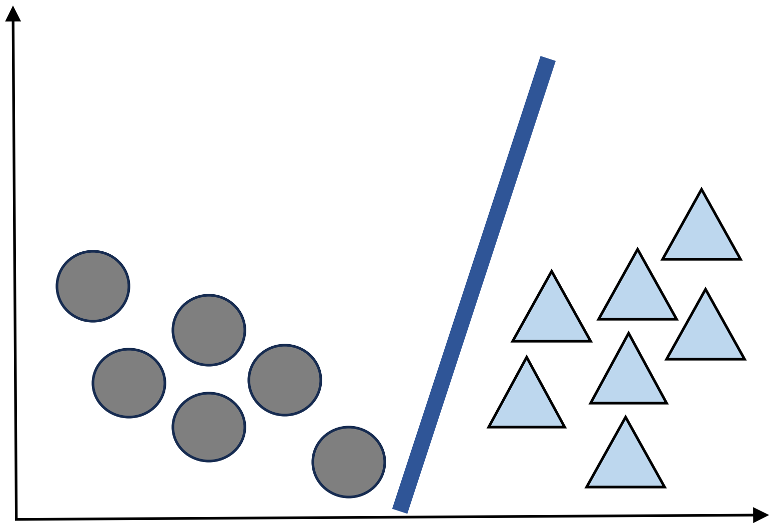
- Training goal is to minimize the average loss

What happens if there are more than 2 classes ?

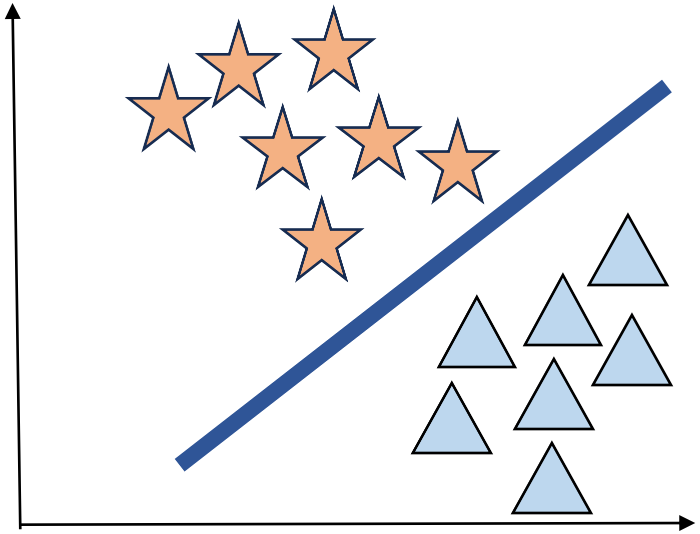
Multinomial Logistic Regression

Approach 1: For each class, we predict the probability that of observation belonging to the class.

$p(Y = 0|\boldsymbol{X})$



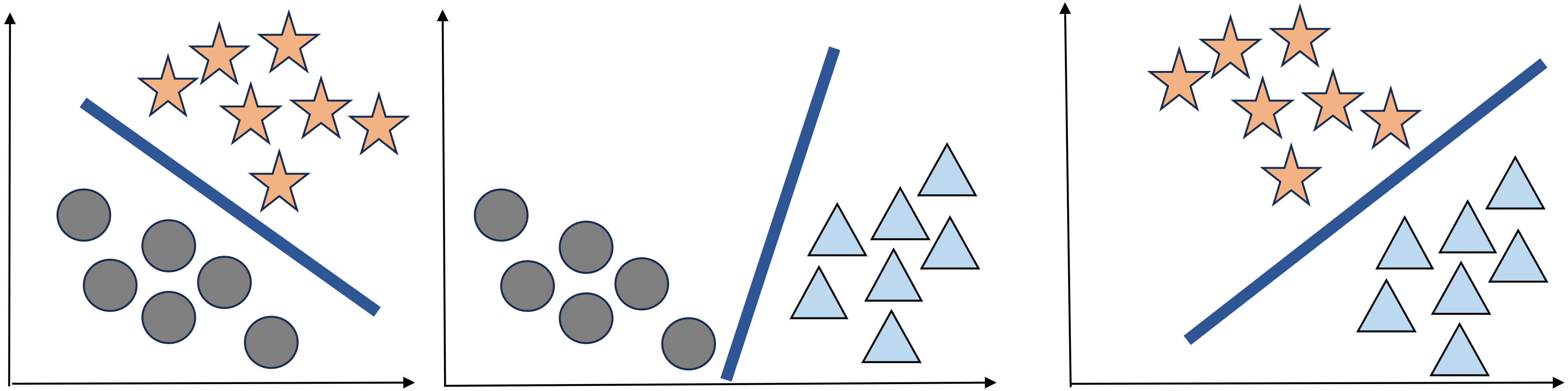
$p(Y = 1|\boldsymbol{X})$



$p(Y = 2|\boldsymbol{X})$

Multinomial Logistic Regression

Approach 1: For each class, we predict the probability that of observation belonging to the class.



$p(Y = 0|\boldsymbol{X})$ $p(Y = 1|\boldsymbol{X})$ $p(Y = 2|\boldsymbol{X})$

Final class is the one with the highest probability

Multinomial Logistic Regression

Approach 1: For each class, we predict the probability that of observation belonging to the class.

Limitations:

- Computation can be intense for large classes
- We might not need all possible computations

Multinomial Logistic Regression

Approach 2: Use Softmax Regression to compute the probability that a data point belongs to each class by using below softmax function

$$softmax(\, e^{z}) \;=\; \frac{exp(z_i)}{\sum_{\mathrm{i}=1}^{K} exp(z_i)} \;1 \leq \; i \;\leq \; K$$

Multinomial Logistic Regression

Approach 2: Use Softmax Regression to compute the probability that a data point belongs to each class by using below softmax function

$$softmax(\,e^{z}) \;=\; \frac{exp(z_i)}{\sum_{\mathrm{i}=1}^{K} exp(z_i)}\; 1 \leq\; i\; \leq\; K$$

- The softmax function of an input vector z = [$z_1, z_1, \ldots, z_k$] is given by :

$$softmax(\,z) \;=\; \frac{exp(z_1)}{\sum_{\mathrm{i}=1}^{K} exp(z_i)} + \frac{exp(z_2)}{\sum_{\mathrm{i}=1}^{K} exp(z_i)}, \ldots, \frac{exp(z_K)}{\sum_{\mathrm{i}=1}^{K} exp(z_i)}$$

Multinomial Logistic Regression

Approach 2: Use Softmax Regression to compute the probability that a data point belongs to each class by using below softmax function

$$softmax(e^z) = \frac{exp(z_i)}{\sum_{i=1}^{K} exp(z_i)} \quad 1 \leq i \leq K$$

- The softmax function of an input vector z = [$z_1, z_1, \ldots, z_k$] is given by :

$$softmax(z) = \frac{exp(z_1)}{\sum_{i=1}^{K} exp(z_i)} + \frac{exp(z_2)}{\sum_{i=1}^{K} exp(z_i)}, \ldots, \frac{exp(z_K)}{\sum_{i=1}^{K} exp(z_i)}$$

- The denominator $\sum_{i=1}^{K} exp(z_i)$ is used to normalize all the values into probabilities.

Multinomial Logistic Regression

Approach 2: Use Softmax Regression to compute the probability that a data point belongs to each class by using below softmax function

$$softmax(\, e^{z}) \;=\; \frac{exp(z_i)}{\sum_{\mathrm{i}=1}^{K} exp(z_i)} \; 1 \leq \, i \, \leq \, K$$

- The softmax function of an input vector z = [$z_1, z_1, \ldots, z_k$] is given by :

$$softmax(\, z) \;=\; \frac{exp(z_1)}{\sum_{\mathrm{i}=1}^{K} exp(z_i)} + \frac{exp(z_2)}{\sum_{\mathrm{i}=1}^{K} exp(z_i)}, \ldots, \frac{exp(z_K)}{\sum_{\mathrm{i}=1}^{K} exp(z_i)}$$

- The denominator $\sum_{\mathrm{i}=1}^{K} exp(z_i)$ is used to normalize all the values into probabilities.
- Like the sigmoid, the softmax has the property of squashing values toward 0 or 1.

McMaster University

Multinomial Logistic Regression

Approach 2: Use Softmax Regression to compute the probability that a data point belongs to each class by using below softmax function

$$softmax(\,e^{z}) \;=\; \frac{exp(z_i)}{\sum_{\mathrm{i}=1}^{K} exp(z_i)} \; 1 \le \; i \; \le \; K$$

- The softmax function of an input vector z = [$z_1, z_1, \ldots, z_k$] is given by :

$$softmax(\,z) \;=\; \frac{exp(z_1)}{\sum_{\mathrm{i}=1}^{K} exp(z_i)} + \frac{exp(z_2)}{\sum_{\mathrm{i}=1}^{K} exp(z_i)}, \ldots, \frac{exp(z_K)}{\sum_{\mathrm{i}=1}^{K} exp(z_i)}$$

z = is the vector of score is called the logit.

- The denominator $\sum_{\mathrm{i}=1}^{K} exp(z_i)$ is used to normalize all the values into probabilities.
- Like the sigmoid, the softmax has the property of squashing values toward 0 or 1.

Multinomial Logistic Regression

Approach 2: Use Softmax Regression to compute the probability that a data point belongs to each class by using below softmax function

- When applying softmax to logistic regression, the probability of each output corresponding to class k is given by:

$$P(y_k = 1|x) = \frac{exp(w_k.x + b_k)}{\sum_{\mathrm{i}=1}^{K} exp(w_i.x + b_i)}$$

Multinomial Logistic Regression

Approach 2: Use Softmax Regression to compute the probability that a data point belongs to each class by using below softmax function

- Let's say probability of instance x belonging to class Y=1 for a given β is given by: $P(Y = 1|x;\ \beta)$
- Then using Softmax means:

$$\begin{vmatrix} P(Y = 1|x;\ \beta) \\ P(Y = 2|x;\ \beta) \\ \vdots \\ P(Y = K|x;\ \beta) \end{vmatrix} = \frac{1}{\sum_{i=1}^{K} exp(w_i.x + b_i)} \begin{vmatrix} exp(w_1.x + b_1) \\ exp(w_2.x + b_2) \\ \vdots \\ exp(w_k.x + b_k) \end{vmatrix}$$

Normalization (sum probabilities to 1)

Separate for each class

Predict class with highest probability

Multinomial Logistic Regression

Approach 2: Use Softmax Regression to compute the probability that a data point belongs to each class by using below softmax function

- Let's say probability of instance x belonging to class Y=1 for a given β is given by: $P(Y = 1|x;\ \beta)$
- Then using Softmax means:

$$\begin{bmatrix} P(Y=1|x;\ \beta) \\ P(Y=2|x;\ \beta) \\ \vdots \\ P(Y=K|x;\ \beta) \end{bmatrix} = \frac{1}{\sum_{\mathrm{i}=1}^{K} exp(w_i.x + b_i)} \begin{bmatrix} exp(w_1.x + b_1) \\ exp(w_2.x + b_2) \\ \vdots \\ exp(w_k.x + b_k) \end{bmatrix}$$

Normalization (sum probabilities to 1)

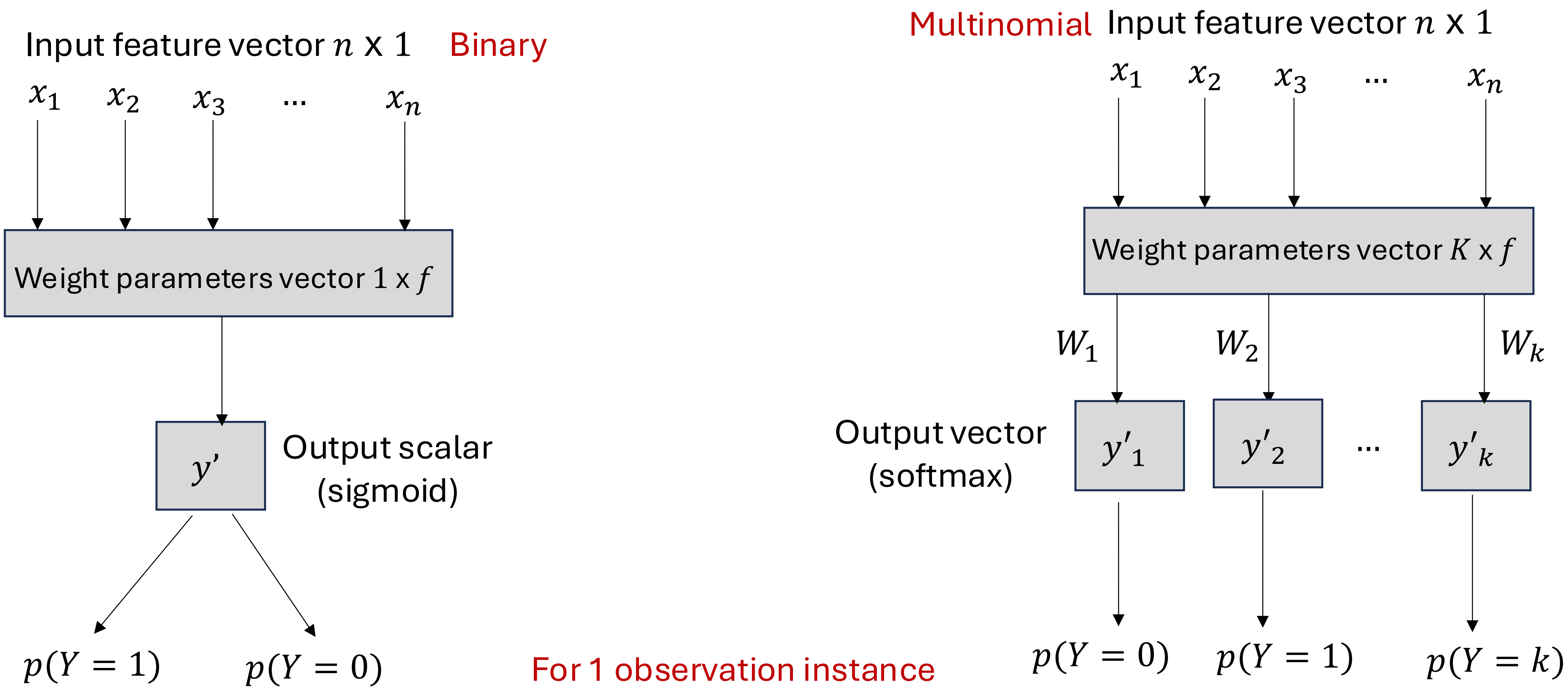
Separate for each class

Predict class with highest probability

What happens for K= 2 ?

McMaster University

Binary vs Multinomial Regression



Multinomial Logistic Regression

What does output of Softmax function look like:

- One hot encoding vector of size k for k classes

Example: 5 classes, 3 data instances

Positive	Negative	Neutral	Too Positive	Too Negative
0	0	0	0	0

Multinomial Logistic Regression

What does output of Softmax function look like:

- One hot encoding vector of size k for k classes

Example: 5 classes, 3 data instances

	Positive	Negative	Neutral	Too Positive	Too Negative
Data instance 1: Professor is awesome	1	0	0	0	0

Multinomial Logistic Regression

What does output of Softmax function look like:

- One hot encoding vector of size k for k classes

Example: 5 classes, 3 data instances

	Positive	Negative	Neutral	Too Positive	Too Negative
Data instance 1: Professor is awesome	1	0	0	0	0
Data instance 2: Professor is horrible	0	0	0	0	1

Multinomial Logistic Regression

What does output of Softmax function look like:

- One hot encoding vector of size k for k classes

Example: 5 classes, 3 data instances	Positive	Negative	Neutral	Too Positive	Too Negative
Data instance 1: Professor is awesome	1	0	0	0	0
Data instance 2: Professor is horrible	0	0	0	0	1
Data instance 3: Professor is meh	0	0	1	0	0

Multinomial Logistic Regression

Where else Softmax function is used:

- Neural Networks:
  - Softmax is often used as the final layer of NN.
  - Such networks are commonly under log loss (or cross-entropy).

- Reinforcement Learning:
  - Softmax function can convert action value corresponding to expected reward into probabilities.

Evaluating Classifiers

Predicted outcome of classifiers can belong to either of these categories:

Actual Value	**Predicted Value**	
	Predicted Positive	Predicted Negative
Positive (P)	True Positive (TP)	False Negative (FN)
Negative (N)	False Positive (FP)	True Negative (TN)

Evaluating Classifiers

Predicted outcome of classifiers can belong to either of these categories:

Actual Value	**Predicted Value**	
	Predicted Positive	Predicted Negative
Positive (P)	True Positive (TP)	False Negative (FN)
Negative (N)	False Positive (FP)	True Negative (TN)

Accuracy = (TP + TN) / (P+N)

Precision = TP / TP+FP

Recall = TP / TP+FN

Sensitivity = True Positive Rate = TP/P

Specifity = True Negative Rate = TN/N

Evaluating Classification Models

Predicted outcome of Binary Classifier:

		Predicted condition	
	Total 8 + 4 = 12	Cancer 7	Non-cancer 5
Actual condition	Cancer 8	6	2
	Non-cancer 4	1	3

Picture Source : https://en.wikipedia.org/wiki/Confusion_matrix

Evaluating Classification Models

Predicted outcome of Multiclass Classifier:

		PREDICTED classification				
	Classes	a	b	c	d	Total
ACTUAL classification	a	6	0	1	2	9
	b	3	9	1	1	14
	c	1	0	10	2	13
	d	1	2	1	12	16
	Total	11	11	13	17	**52**

Picture Source : https://ar5iv.labs.arxiv.org/html/2008.05756

Readings

Required Readings:

Introduction to Statistical Learning

1. Chapter 4 – Section 4.3 Page 138 – 144
2. Chapter 2 – Section 2.2.3 Page 34 – 40

Supplemental Readings (Not required but recommended):

1. Online: [Metrics for Multiclass Classification]

Thank You

McMaster University