

CS60050

Machine Learning

Logistic Regression (Classification?)

Somak Aditya
Assistant Professor

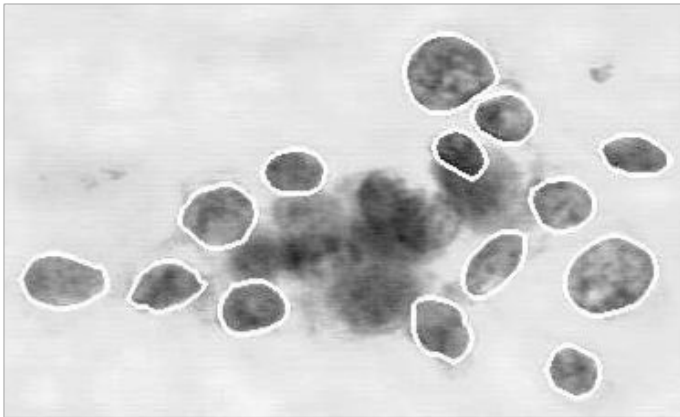
Sudeshna Sarkar

Department of CSE, IIT Kharagpur
August 2, 2024



Example: Breast cancer classification

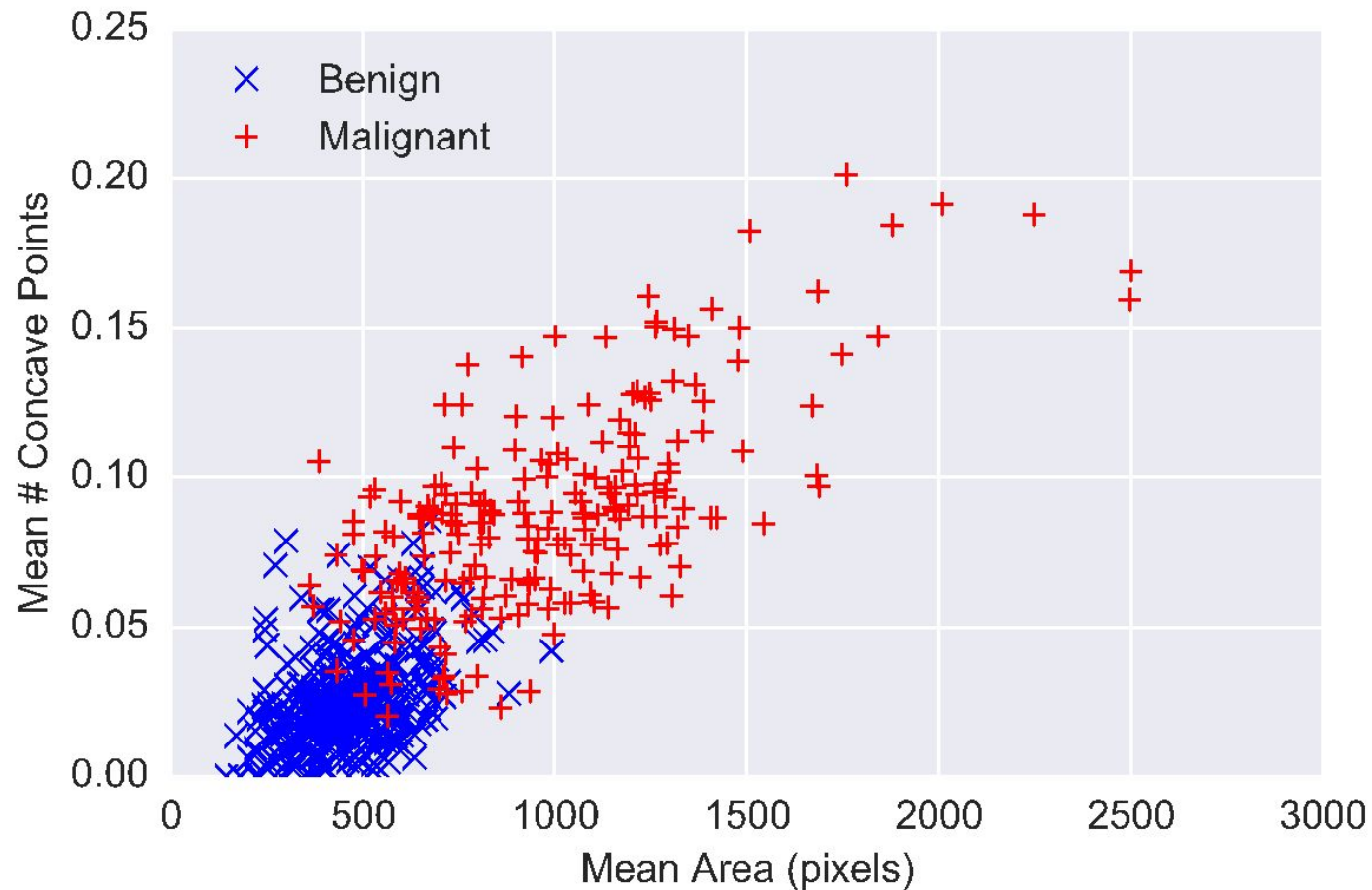
- Well-known classification example: using machine learning to diagnose whether a breast tumor is benign or malignant [Street et al., 1992]
- Setting: doctor extracts a sample of fluid from tumor, stains cells, then outlines several of the cells (image processing refines outline)



- System computes features for each cell such as area, perimeter, concavity, texture (10 total); computes mean/std/max for all features

Example: Breast cancer classification

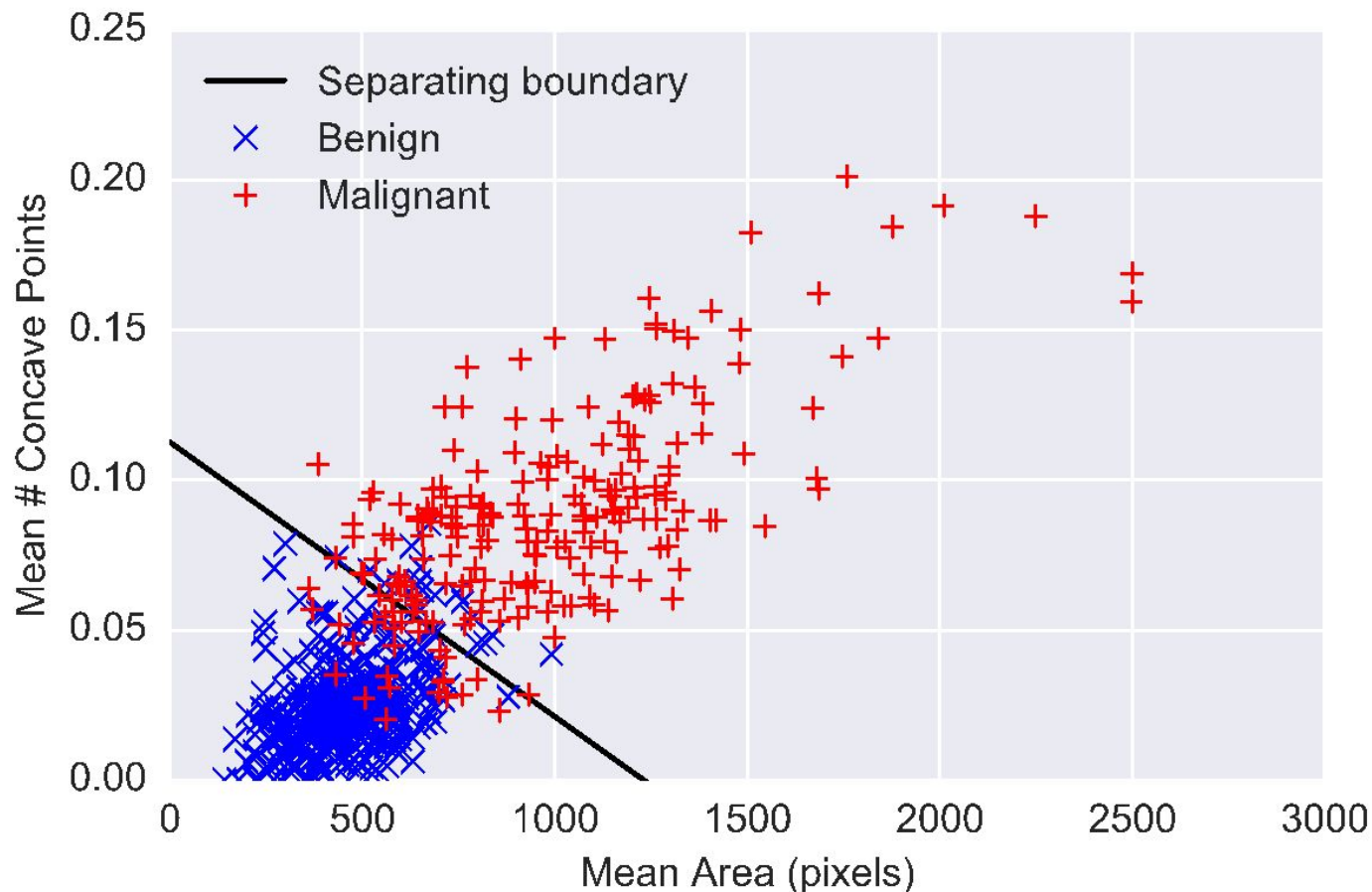
- Plot of two features: mean area vs. mean concave points, for two classes



Logistic regression for classification

- Linear classification: linear decision boundary

Probabilistic classification: provide $P(Y = 1 \mid x)$ rather than just $\hat{y} \in \{0, 1\}$



Building on a Linear Model

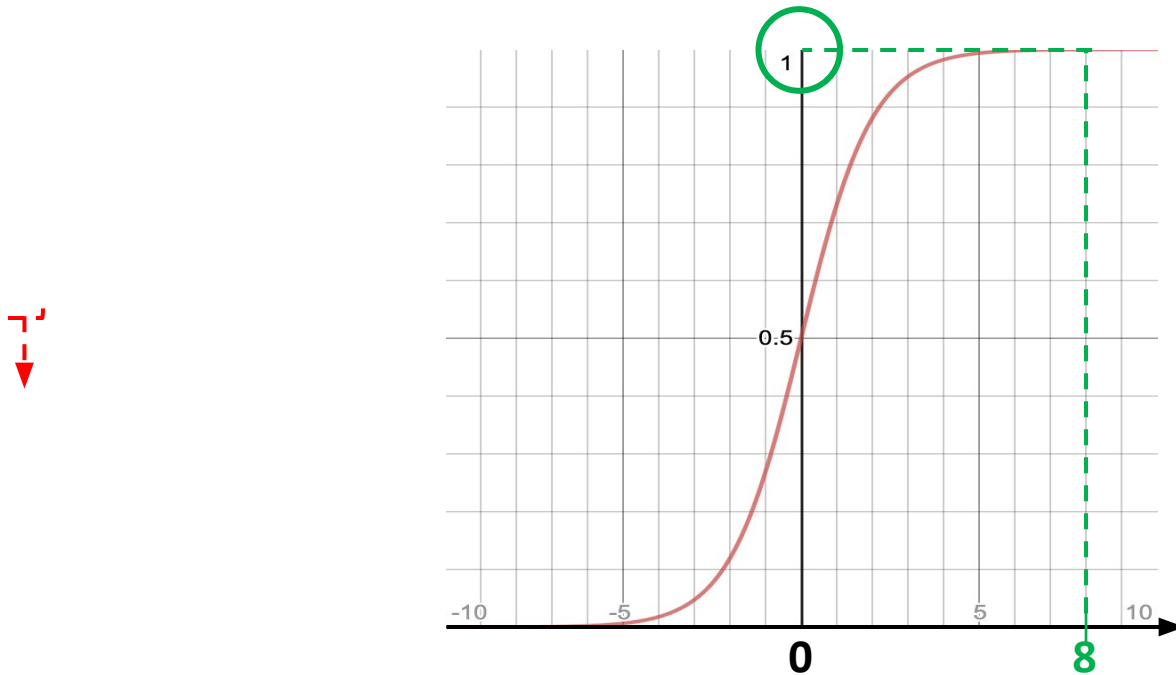
- Linear vs Thresholded Linear vs Logistic Linear

Building on a Linear Model

- Linear vs Thresholded Linear vs Logistic Linear

Regression vs. Classification

• We want the possible outputs of $f_{\theta}(x) = \theta^T x$ to be discrete-valued
Use an **activation function** (e.g., **sigmoid or logistic function**)



Classification



Thresholding:

predict “y = 1” if $f_{\theta}(x) \geq 0.5$

$$z = \theta^T x \geq 0$$

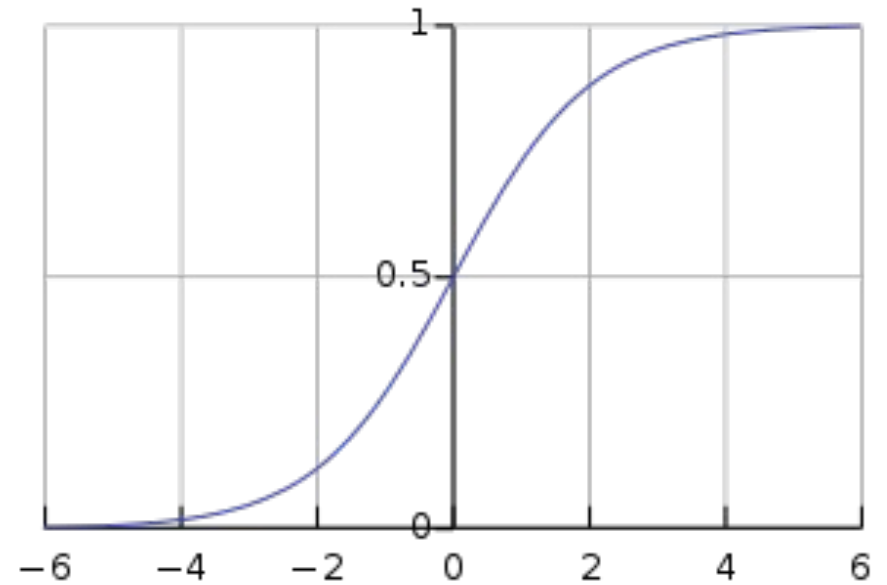
predict “y = 0” if $f_{\theta}(x) < 0.5$

$$z = \theta^T x < 0$$



$$f_{\theta}(x) \in [0,1]$$

$$f_{\theta}(x) = g(\theta^T x)$$
$$g(z) = \frac{1}{1 + e^{-z}}$$



Classification



Thresholding:

predict “y = 1” if $f_{\theta}(x) \geq 0.5$

$$z = \theta^T x \geq 0$$

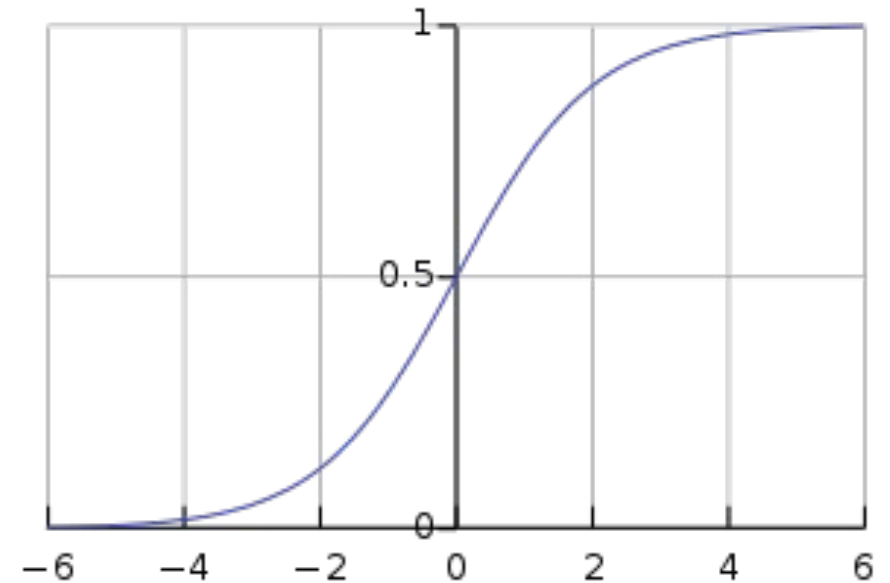
predict “y = 0” if $f_{\theta}(x) < 0.5$

$$z = \theta^T x < 0$$



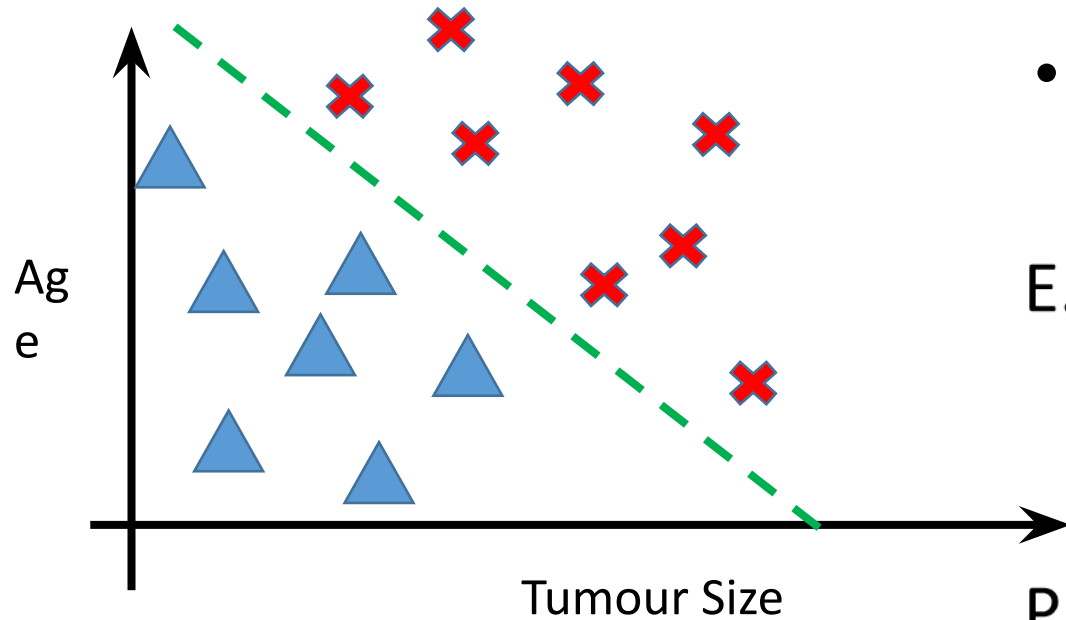
$$f_{\theta}(x) \in [0,1]$$

$$f_{\theta}(x) = g(\theta^T x)$$
$$g(z) = \frac{1}{1 + e^{-z}}$$



Alternative Interpretation: $f_{\theta}(x)$ =
estimated probability that $y = 1$ on input x
Will come back to it shortly!

Decision boundary



- $$f_{\theta}(x) = g(\theta_0 + \theta_1 x_1 + \theta_2 x_2)$$

E.g., $\theta_0 = -3$, $\theta_1 = 1$, $\theta_2 = 1$

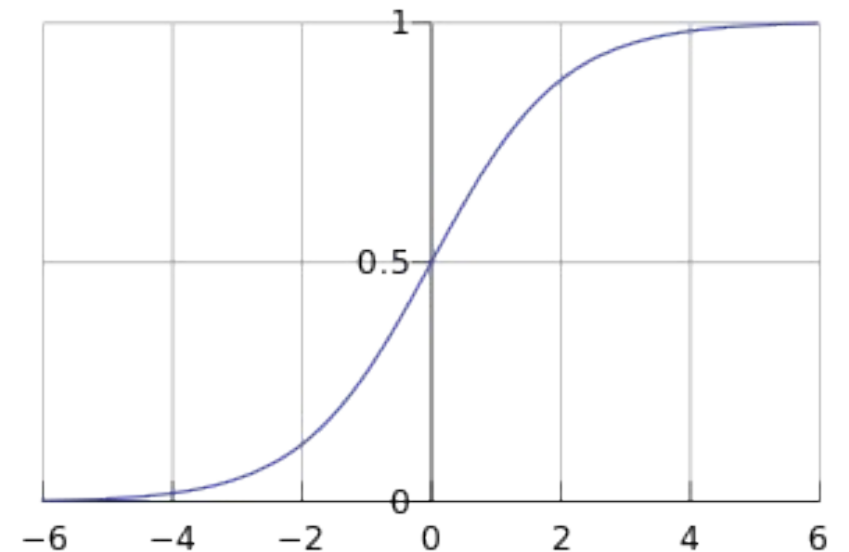
Predict “ $y = 1$ ” if $-3 + x_1 + x_2 \geq 0$

Poll 1

• For a point \mathbf{x} on the decision boundary of logistic regression, does $g(\mathbf{w}^T \mathbf{x} + b) = \mathbf{w}^T \mathbf{x} + b$?

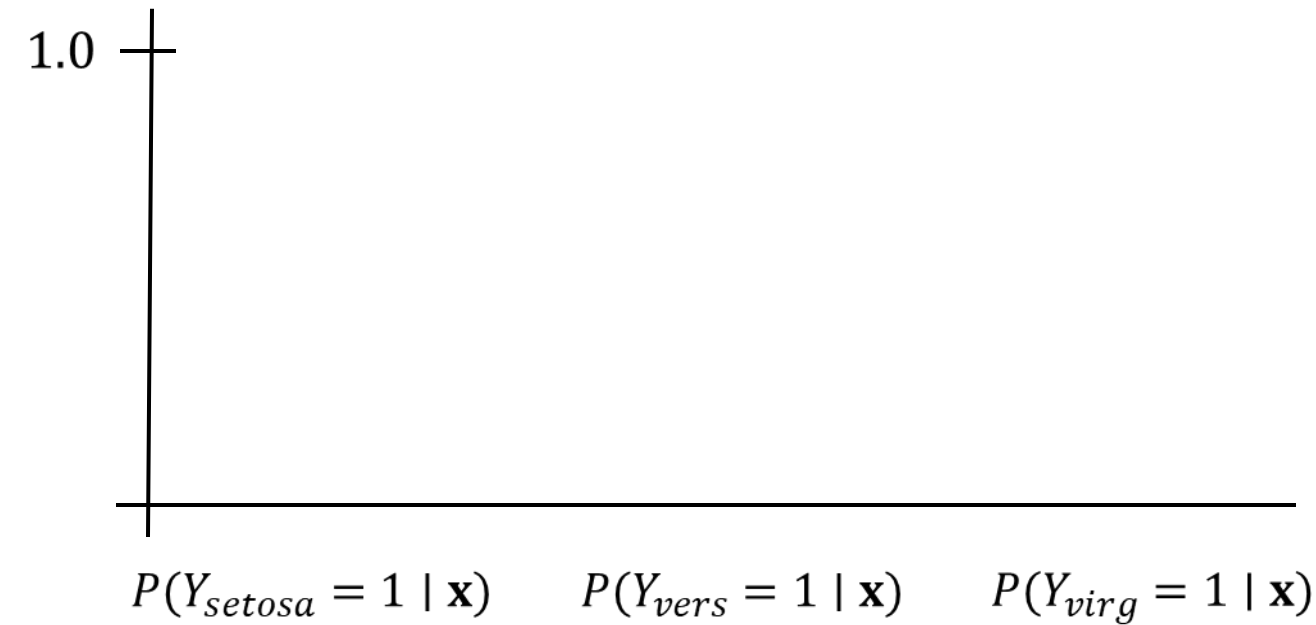
- A) Yes
- B) No
- C) I have no idea

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

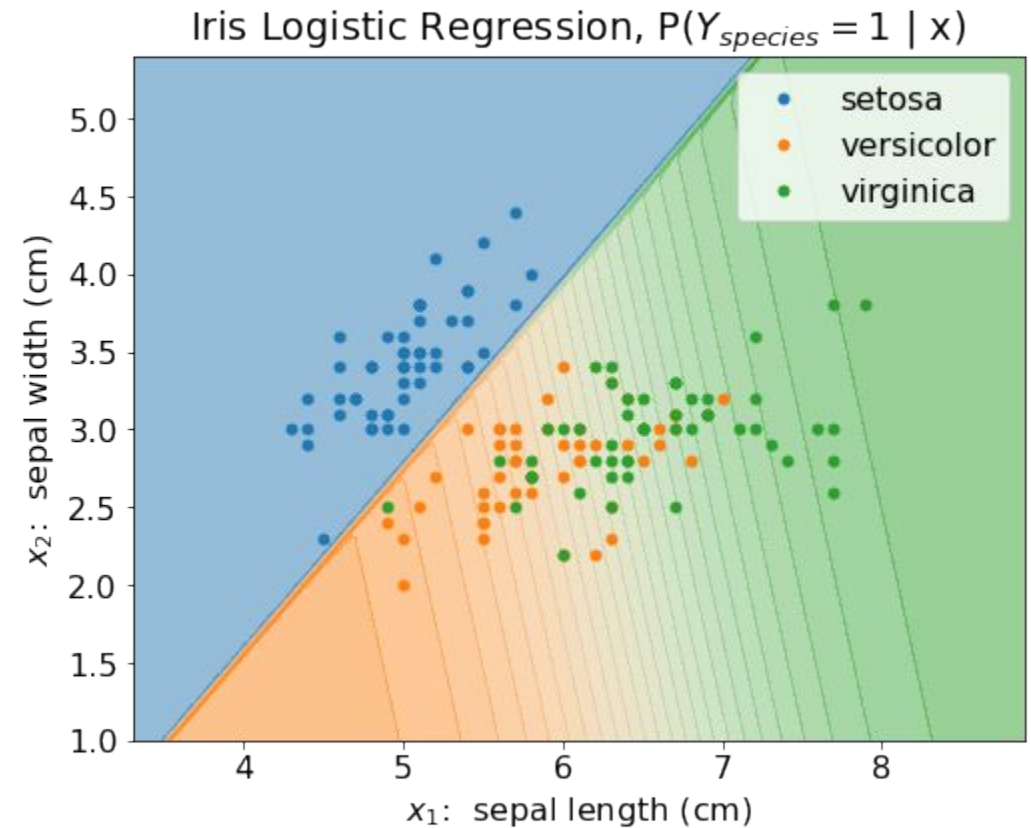


Pre-Reading: Classification “Probability”

- Constructing a model that can return the probability of the output being a specific class could be incredibly useful

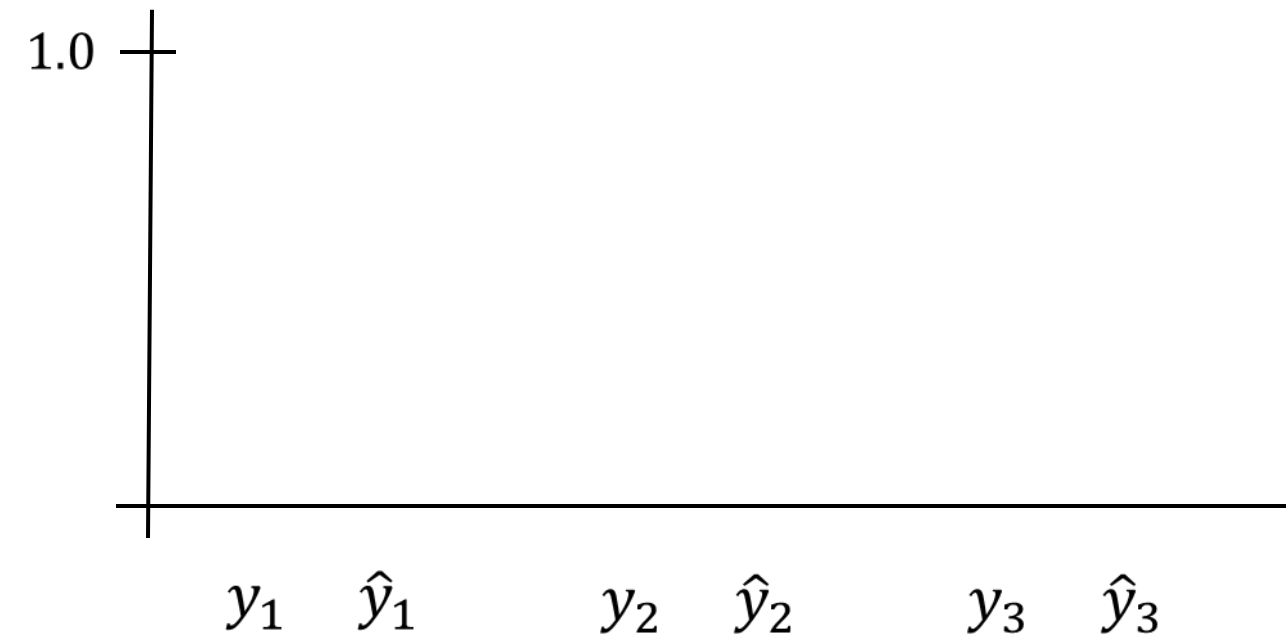


We can still make decisions, .e.g,
$$\operatorname{argmax}_k P(Y_k = 1 | \mathbf{x})$$



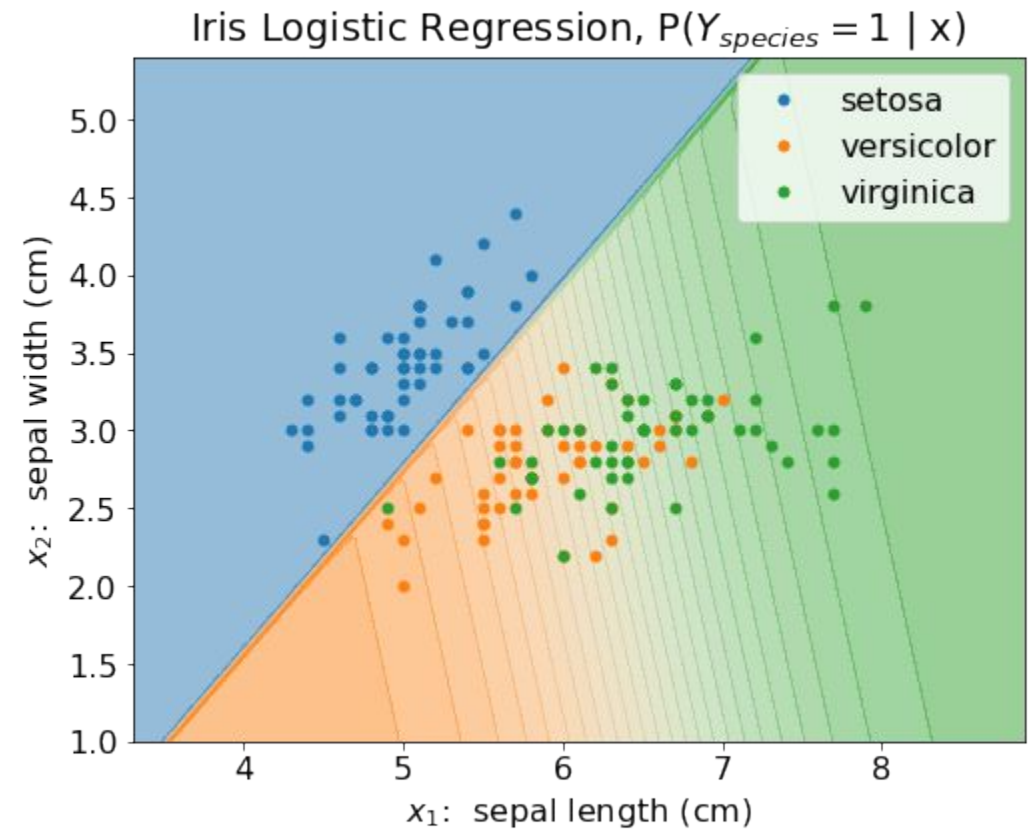
Pre-Reading: Loss for Probability Distributions

- We need a way to compare how good/bad each prediction is



Cross-entropy loss

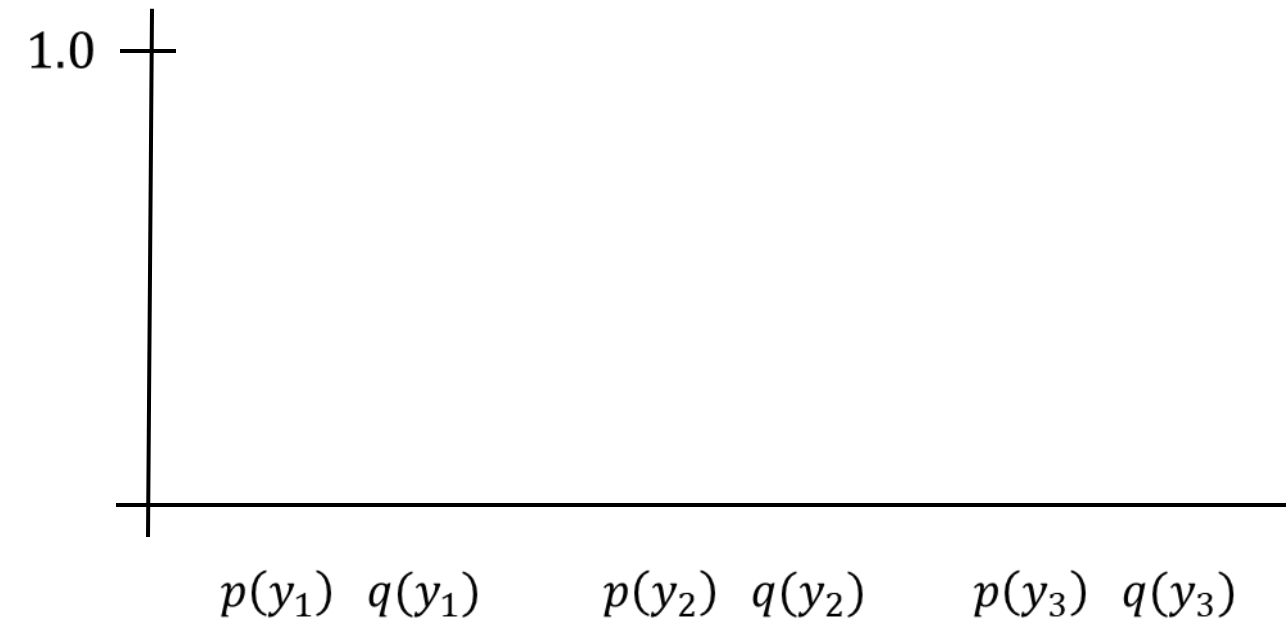
$$\ell(\mathbf{y}, \hat{\mathbf{y}}) = -\sum_{k=1}^K y_k \log \hat{y}_k$$



Pre-Reading: Loss for Probability Distributions

- Cross-entropy more generally is a way to compare any two probability distributions*

*when used in logistic regression
 \mathbf{y} is always a one-hot vector



Cross-entropy loss

$$H(P, Q) = - \sum_{k=1}^K p(y_k) \log q(y_k)$$

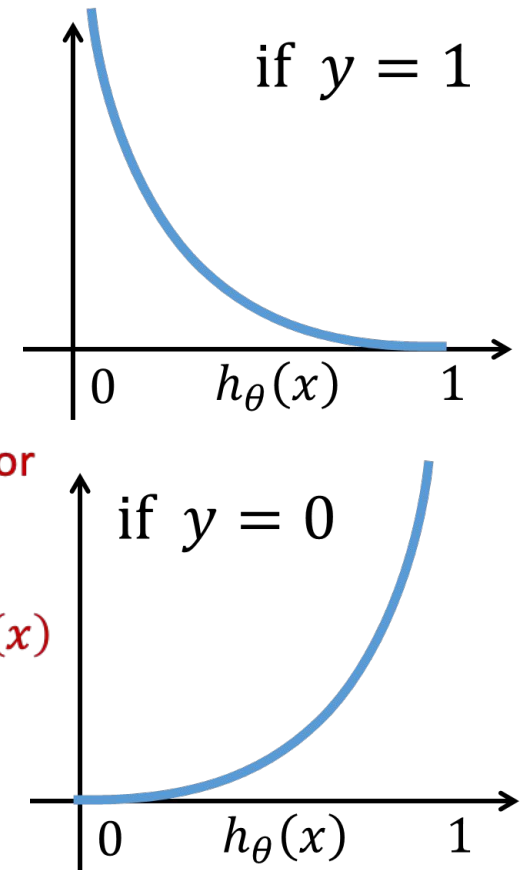
Cost function for Logistic Regression

Logistic Regression

$$\begin{aligned} \bullet \text{ Cost}(f_{\theta}(x), y) &= \begin{cases} -\log(f_{\theta}(x)) & \text{if } y = 1 \\ -\log(1 - f_{\theta}(x)) & \text{if } y = 0 \end{cases} \\ &= -y \log(f_{\theta}(x)) - (1 - y) \log(1 - f_{\theta}(x)) \end{aligned}$$

Functional Interpretation:
Maximize $f_{\theta}(x)$ for $y = 1$

Maximize $1 - f_{\theta}(x)$ for $y = 0$



$$\begin{aligned} J(\theta) &= \frac{1}{m} \sum_{i=1}^m \text{Cost}(f_{\theta}(x^{(i)}), y^{(i)}) \\ &= -\frac{1}{m} \left[\sum_{i=1}^m y^{(i)} \log(f_{\theta}(x^{(i)})) + (1 - y^{(i)}) \log(1 - f_{\theta}(x^{(i)})) \right] \end{aligned}$$

Binary Logistic Regression

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

- Objective: Special case for binary logistic regression

$$\hat{y} = g(\boldsymbol{\theta}^T \mathbf{x})$$

$$J(\boldsymbol{\theta}) = -\frac{1}{N} \sum_i \sum_k y_k^{(i)} \log y_k^{(i)}$$

$$= -\frac{1}{N} \sum_i (y^{(i)} \log \hat{y}^{(i)} + (1 - y^{(i)}) \log(1 - \hat{y}^{(i)}))$$

Solve Logistic Regression

$$\hat{y} = g(\boldsymbol{\theta}^T \mathbf{x}) \quad g(z) = \frac{1}{1 + e^{-z}} \quad \frac{dg}{dz} = g(z)(1 - g(z))$$

$$J^{(i)}(\boldsymbol{\theta}) = -[y^{(i)} \log \hat{y}^{(i)} + (1 - y^{(i)}) \log(1 - \hat{y}^{(i)})]$$

$$\frac{\partial J^{(i)}}{\partial \boldsymbol{\theta}} = -(y^{(i)} - \hat{y}^{(i)}) \mathbf{x}^{(i)}$$

Solve Logistic Regression

$$\hat{y} = g(\boldsymbol{\theta}^T \mathbf{x}) \quad g(z) = \frac{1}{1 + e^{-z}} \quad \frac{dg}{dz} = g(z)(1 - g(z))$$

$$J^{(i)}(\boldsymbol{\theta}) = -[y^{(i)} \log \hat{y}^{(i)} + (1 - y^{(i)}) \log(1 - \hat{y}^{(i)})]$$

$$\frac{\partial J^{(i)}}{\partial \boldsymbol{\theta}} = -(y^{(i)} - \hat{y}^{(i)}) \mathbf{x}^{(i)}$$

Solve Logistic Regression

$$\hat{y} = g(\boldsymbol{\theta}^T \mathbf{x}) \quad g(z) = \frac{1}{1+e^{-z}}$$

$$J(\boldsymbol{\theta}) = -\frac{1}{N} \sum_i (y^{(i)} \log \hat{y}^{(i)} + (1 - y^{(i)}) \log(1 - \hat{y}^{(i)}))$$

$$\nabla_{\boldsymbol{\theta}} J(\boldsymbol{\theta}) = -\frac{1}{N} \sum_i (y^{(i)} - \hat{y}^{(i)}) \mathbf{x}^{(i)}$$

$$\nabla_{\boldsymbol{\theta}} J(\boldsymbol{\theta}) = 0?$$

No closed form solution ☹

Back to iterative methods. Solve with (stochastic) gradient descent, Newton's method, or Iteratively Reweighted Least Squares (IRLS)

Gradient descent

$$J(\theta) = -\frac{1}{m} \left[\sum_{i=1}^m y^{(i)} \log(f_{\theta}(x^{(i)})) + (1 - y^{(i)}) \log(1 - f_{\theta}(x^{(i)})) \right]$$

Goal: $\min_{\theta} \text{loss}(\theta)$

Good news: Convex function!

Bad news: No analytical solution

Gradient descent

$$\bullet \text{ } loss(\theta) = -\frac{1}{m} \left[\sum_{i=1}^m y^{(i)} \log(f_{\theta}(x^{(i)})) + (1 - y^{(i)}) \log(1 - f_{\theta}(x^{(i)})) \right]$$

$$\frac{\partial}{\partial \theta_j} loss(\theta) = \frac{1}{m} \sum_{i=1}^m (f_{\theta}(x^{(i)}) - y^{(i)}) x_j^{(i)}$$

Gradient descent

Repeat {

$$\theta_j := \theta_j - \alpha \frac{\partial}{\partial \theta_j} \text{loss}(\theta)$$

}

(Simultaneously update all θ_j)

$$\frac{\partial}{\partial \theta_j} l(\theta) = \frac{1}{m} \sum_{i=1}^m (f_{\theta}(x^{(i)}) - y^{(i)}) x_j^{(i)}$$

- **Gradient descent for Linear Regression**

Repeat {

$$\theta_j := \theta_j - \alpha \frac{1}{m} \sum_{i=1}^m (f_{\theta}(x^{(i)}) - y^{(i)}) x_j^{(i)}$$

}

$$f_{\theta}(x) = \theta^{\top} x$$

Gradient descent for Logistic Regression

Repeat {

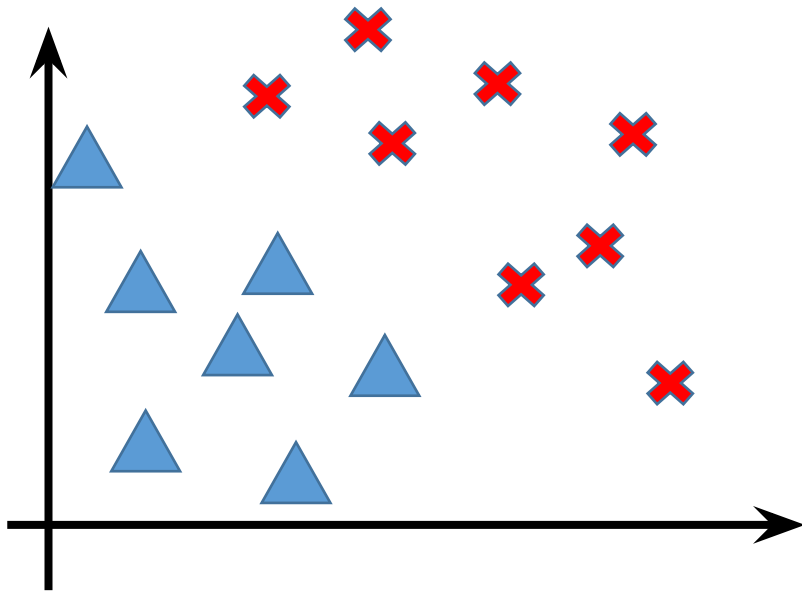
$$\theta_j := \theta_j - \alpha \frac{1}{m} \sum_{i=1}^m (f_{\theta}(x^{(i)}) - y^{(i)}) x_j^{(i)}$$

}

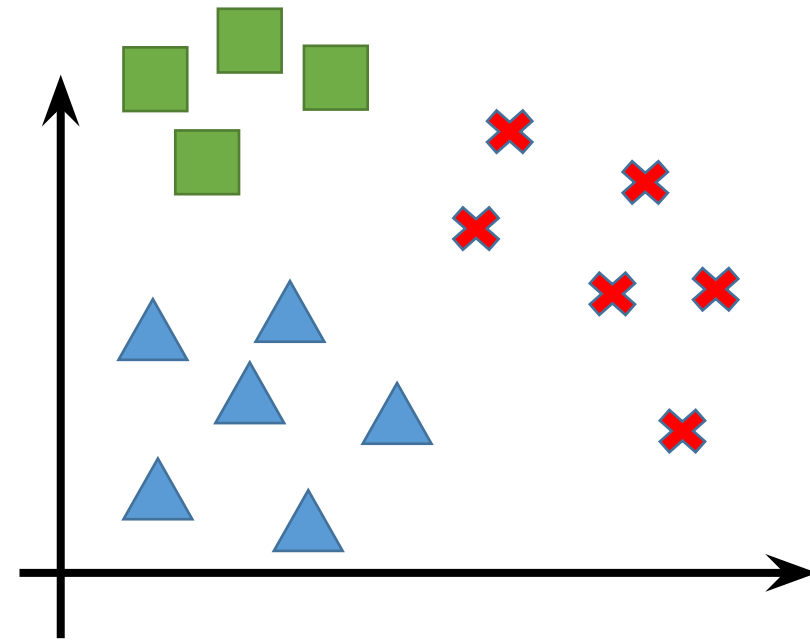
$$f_{\theta}(x) = \frac{1}{1 + e^{-\theta^{\top} x}}$$

Multiclass classification

Binary classification



Multiclass classification



Multi-class Logistic Regression

- Cross-entropy loss
- $\ell(\mathbf{y}, \hat{\mathbf{y}}) = -\sum_{k=1}^K y_k \log \hat{y}_k$
- Model

$$\hat{\mathbf{y}} = h(\mathbf{x}) = g_{softmax}(\mathbf{z})$$

$$\mathbf{z} = \Theta \mathbf{x}$$

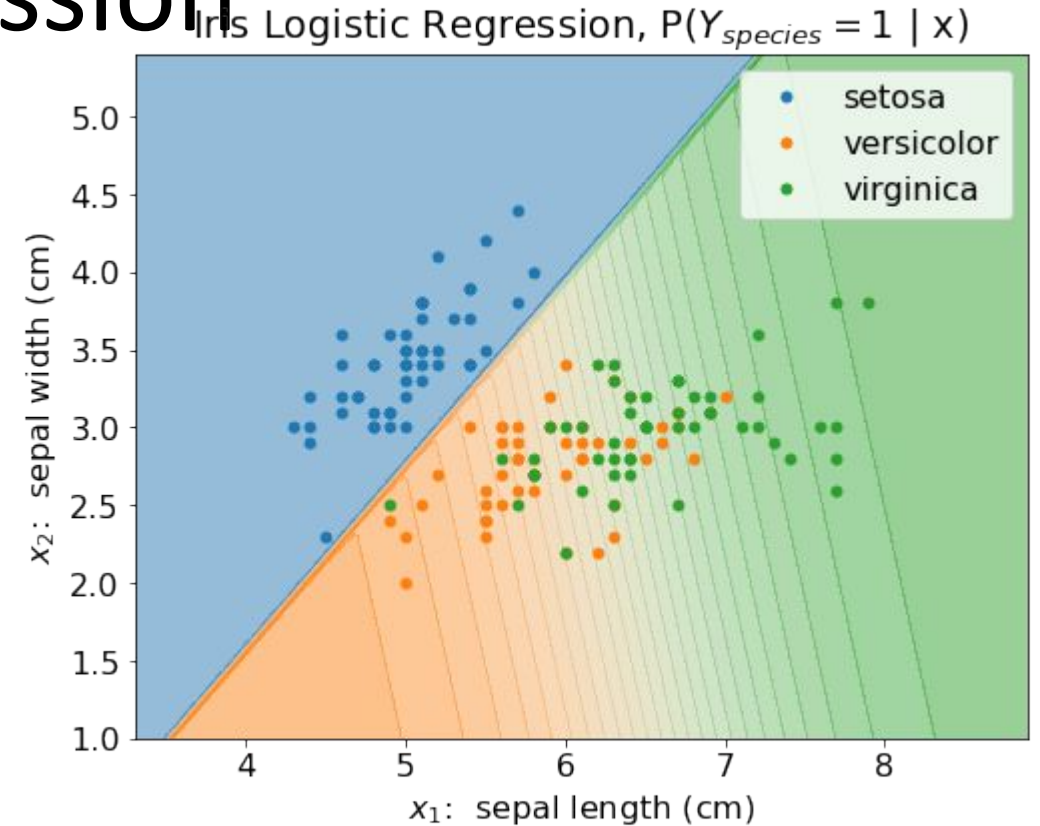
One vector of
parameters for
each class

$$z_k = \boldsymbol{\theta}_k \mathbf{x}$$

$$\mathbf{x} = \begin{bmatrix} 1 \\ x_1 \\ x_2 \end{bmatrix}$$

$$\boldsymbol{\theta}_k = \begin{bmatrix} b_k \\ w_{k,1} \\ w_{k,2} \end{bmatrix}$$

$$\Theta = \begin{bmatrix} - & \boldsymbol{\theta}_1^T & - \\ - & \boldsymbol{\theta}_2^T & - \\ - & \boldsymbol{\theta}_3^T & - \end{bmatrix} = \begin{bmatrix} b_1 & w_{1,1} & w_{1,2} \\ b_2 & w_{2,1} & w_{2,2} \\ b_3 & w_{3,1} & w_{3,2} \end{bmatrix}$$



Stacked into a matrix of $K \times M$ parameters

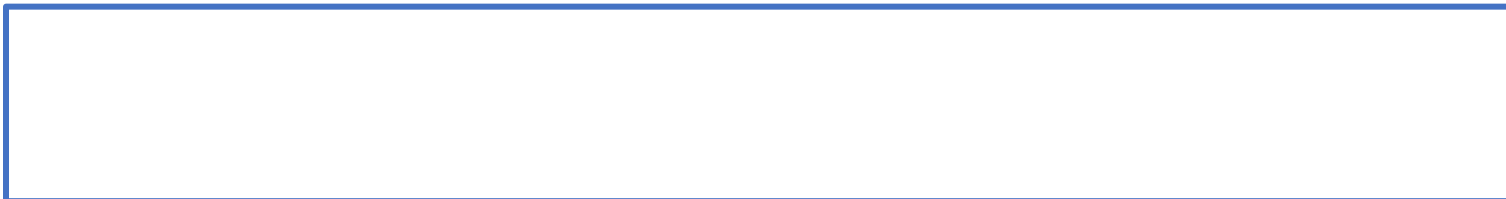
Multi-class Classification

- Multi-class Classification: y can take on K different values $\{1, 2, \dots, k\}$
- $f_{\theta}(x)$ estimates the probability of belonging to each class

$$P(y = k|x, \theta) \propto \exp(\theta_k^T x)$$

$$\theta = \begin{bmatrix} \vdots & \vdots & \vdots \\ \theta_1 & \theta_2 & \theta_k \\ \vdots & \vdots & \vdots \end{bmatrix}$$

$$P(y = k|x, \theta) = \frac{\exp(\theta_k^T x)}{\sum_{j=1}^K \exp(\theta_j^T x)}$$



Multiclass Predicted Probability

- Multiclass logistic regression uses the parameters learned across all K classes to predict the discrete conditional probability distribution of the output Y given a specific input vector \mathbf{x}

$$\bullet \begin{bmatrix} p(Y = 1 \mid \mathbf{x}, \boldsymbol{\theta}_1, \boldsymbol{\theta}_2, \boldsymbol{\theta}_3) \\ p(Y = 2 \mid \mathbf{x}, \boldsymbol{\theta}_1, \boldsymbol{\theta}_2, \boldsymbol{\theta}_3) \\ p(Y = 3 \mid \mathbf{x}, \boldsymbol{\theta}_1, \boldsymbol{\theta}_2, \boldsymbol{\theta}_3) \end{bmatrix} = \begin{bmatrix} e^{\boldsymbol{\theta}_1^T \mathbf{x}} \\ e^{\boldsymbol{\theta}_2^T \mathbf{x}} \\ e^{\boldsymbol{\theta}_3^T \mathbf{x}} \end{bmatrix} \cdot \frac{1}{\sum_{k=1}^K e^{\boldsymbol{\theta}_k^T \mathbf{x}}}$$