CS60050
Machine Learning

Logistic Regression (Classification?)

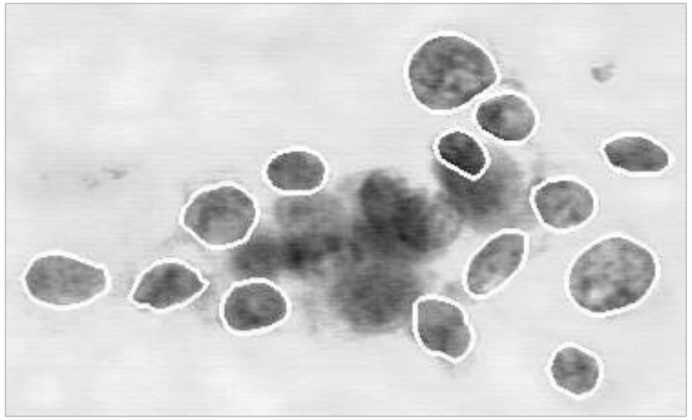
Somak Aditya
Assistant Professor

Sudeshna Sarkar

Department of CSE, IIT Kharagpur
August 2, 2024

Example: Breast cancer classification

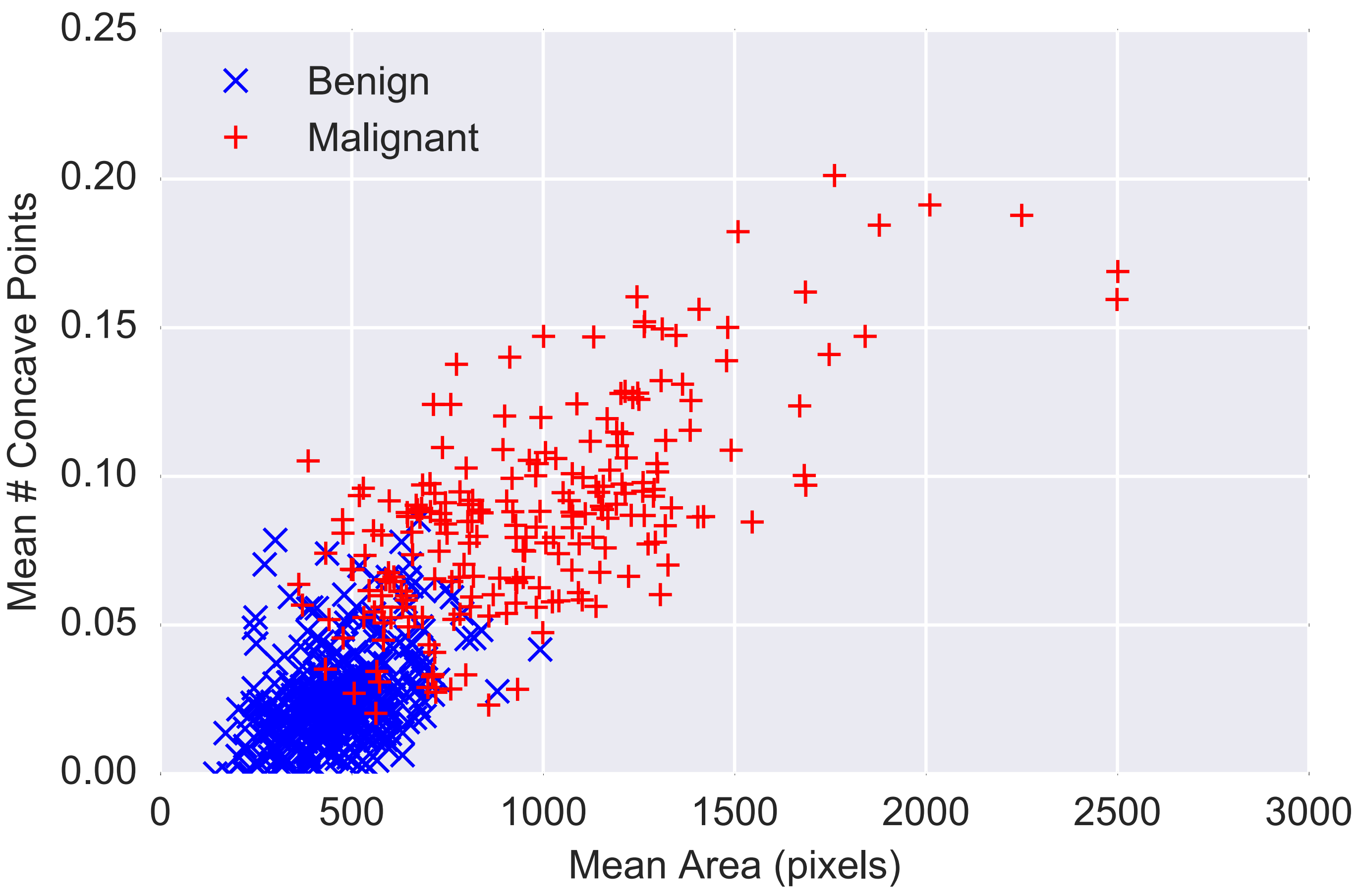
- Well-known classification example: using machine learning to diagnose whether a breast tumor is benign or malignant [Street et al., 1992]
- Setting: doctor extracts a sample of fluid from tumor, stains cells, then outlines several of the cells (image processing refines outline)



- 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

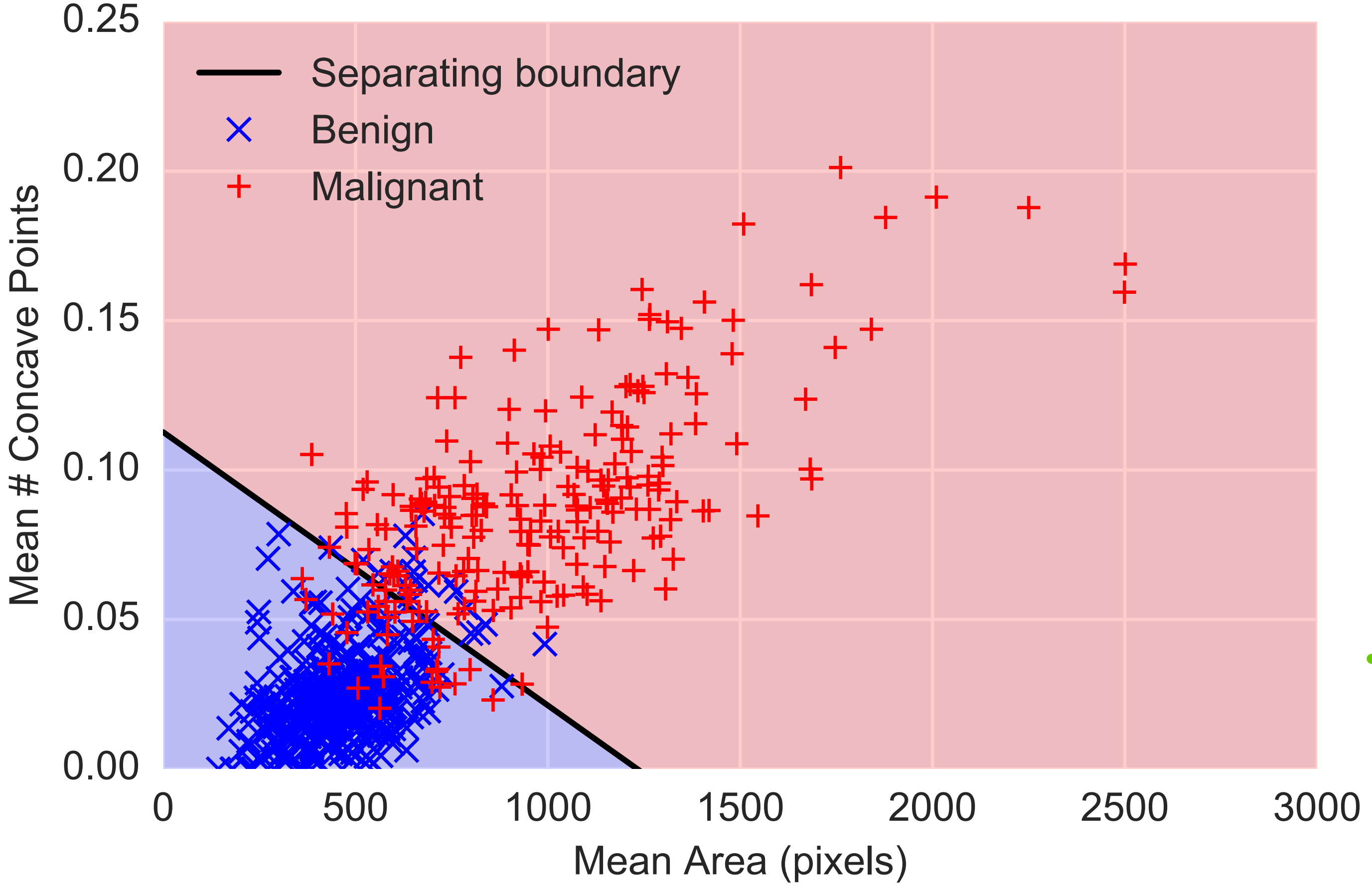
- Plot of two features: mean area vs. mean concave points, for two classes



Slide credit: CMU AI Zico Kolter

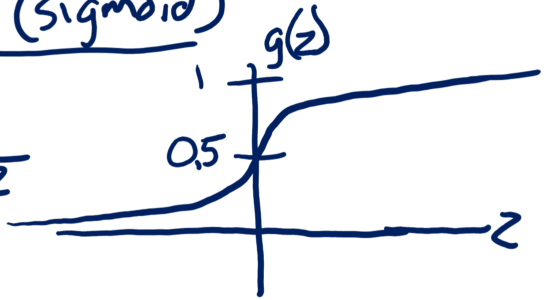
Logistic regression for classification

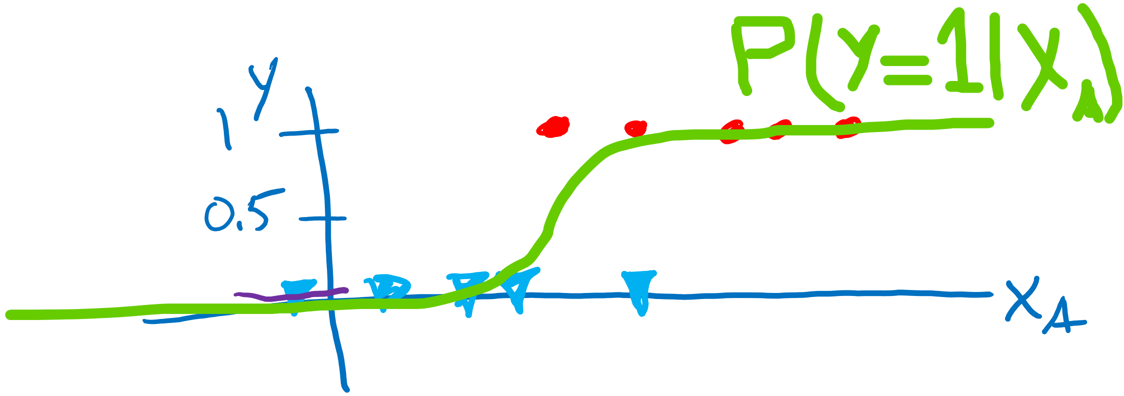
- Linear classification: linear decision boundary
- Probabilistic classification: provide $P(Y = 1 \mid x)$ rather than just $\hat{y} \in \{0, 1\}$



logistic function (sigmoid)

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

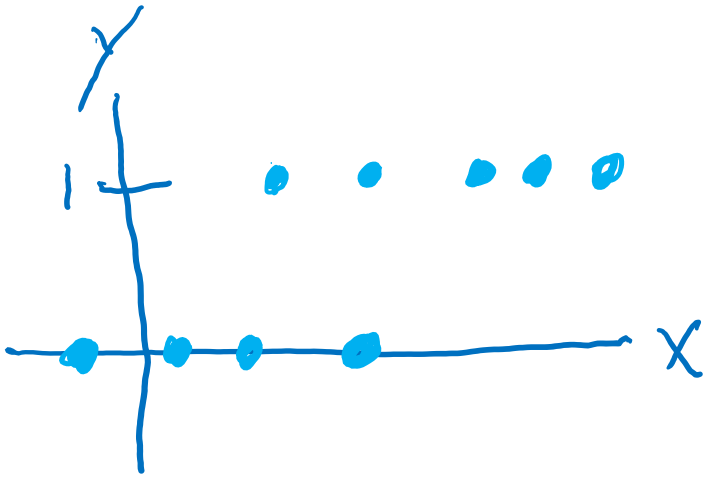


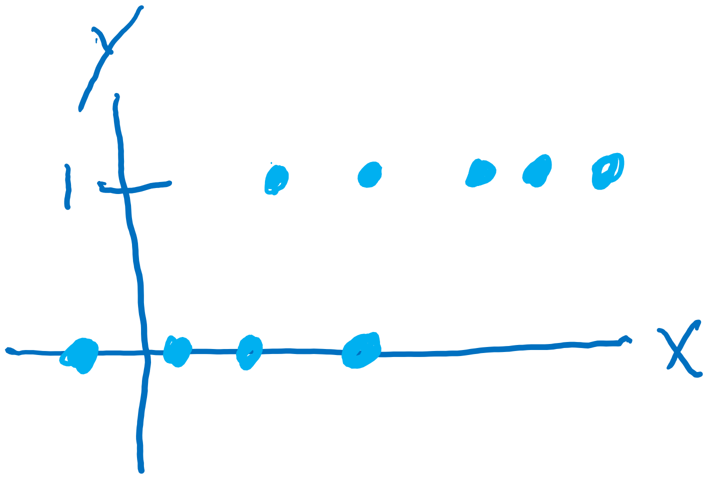


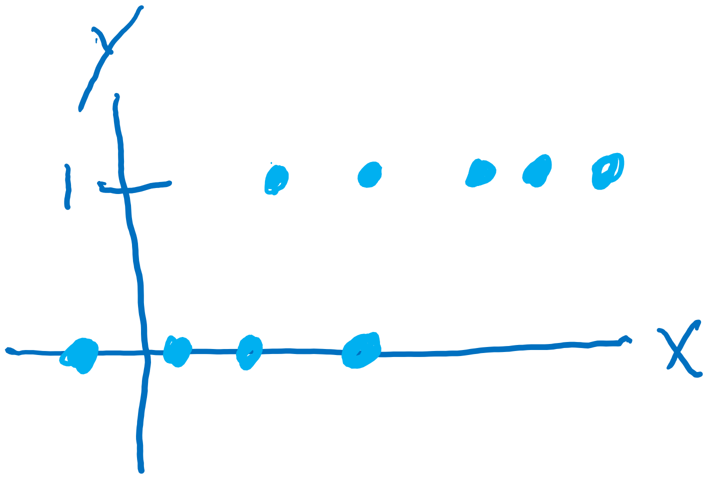
Slide credit: CMU AI Zico Kolter

Building on a Linear Model

- Linear vs Thresholded Linear vs Logistic Linear

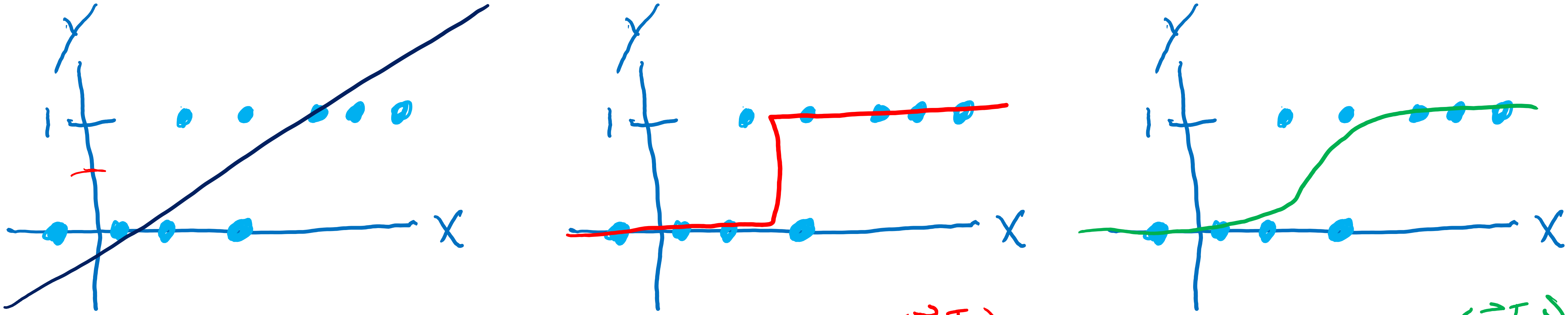






Building on a Linear Model

- Linear vs Thresholded Linear vs Logistic Linear



$\hat{y} = \vec{\theta}^T \vec{x}$

not classification

$\hat{y} = g_{thresh}(\vec{\theta}^T \vec{x})$

classification only (0/1)

zero derivatives

$\hat{y} = g_{logistic}(\vec{\theta}^T \vec{x})$

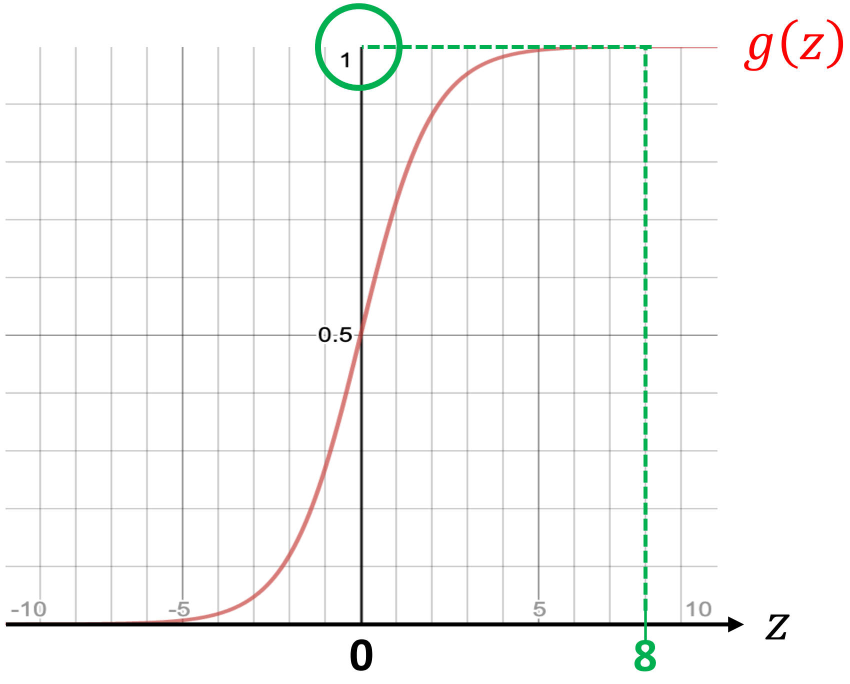
Regression vs. Classification

We want the possible outputs of $\boldsymbol{f_\theta(x)} = \boldsymbol{\theta^T x}$ to be discrete-valued

Use an ***activation function*** (e.g., ***sigmoid or logistic function***)

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

$z \in \mathbb{R}, but$
$g(z) \in [0,1]$



If y = **1**, we want $g(z) \approx 1$ (i.e., we want a correct prediction)
For this to happen, $\boldsymbol{z} \gg \mathbf{0}$

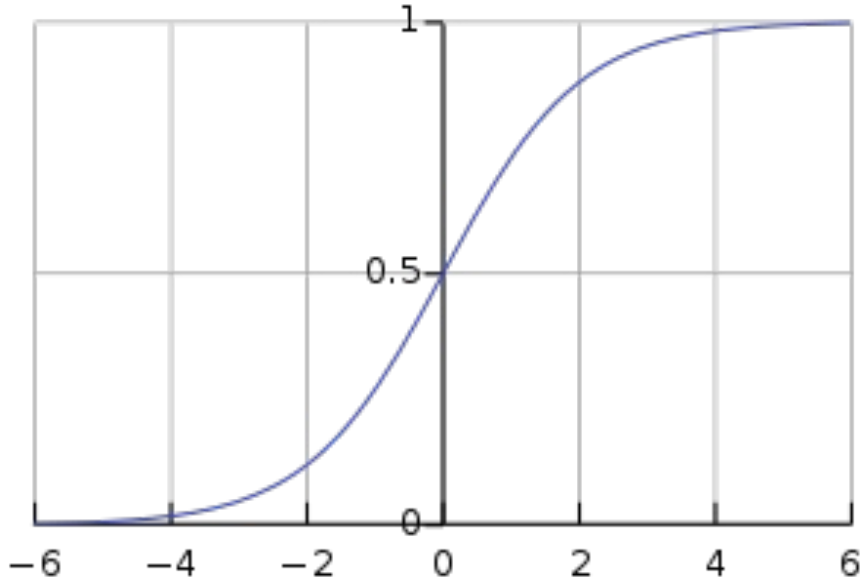
If y = **0**, we want $g(z) \approx 0$ (i.e., we want a correct prediction)
For this to happen, $\boldsymbol{z} \ll \mathbf{0}$

Classification

$\boldsymbol{x} = [\boldsymbol{x}_0, \dots, \boldsymbol{x}_m]$ → $g(\boldsymbol{\theta}^T \boldsymbol{x})$ → $\boldsymbol{f}_{\boldsymbol{\theta}}(\boldsymbol{x}) \in [0,1]$

$$f_\theta(x) = g(\theta^\top x)$$

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



Thresholding:
predict "y = 1" if $f_\theta(x) \geq 0.5$

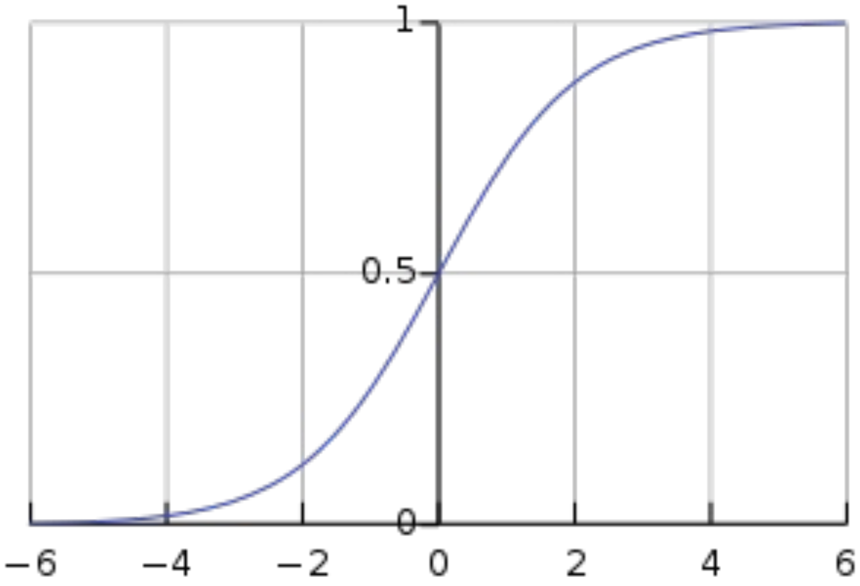
predict "y = 0" if $f_\theta(x) < 0.5$

Classification

$\boldsymbol{x} = [\boldsymbol{x}_0, \dots, \boldsymbol{x}_m] \rightarrow \boldsymbol{g(\theta^T x)} \rightarrow \boldsymbol{f_\theta(x)} \in [0,1]$

$$f_\theta(x) = g(\theta^\top x)$$

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



Thresholding:

predict "y = 1" if $f_\theta(x) \geq 0.5$

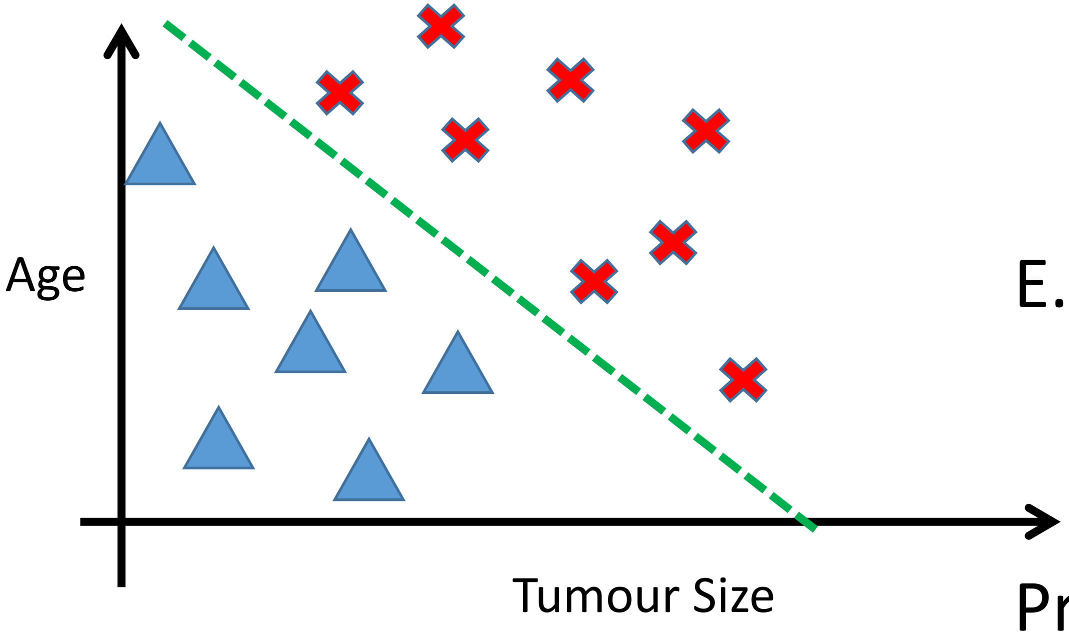
$$\boldsymbol{z = \theta^\top x \geq 0}$$

predict "y = 0" if $f_\theta(x) < 0.5$

$$\boldsymbol{z = \theta^\top x < 0}$$

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

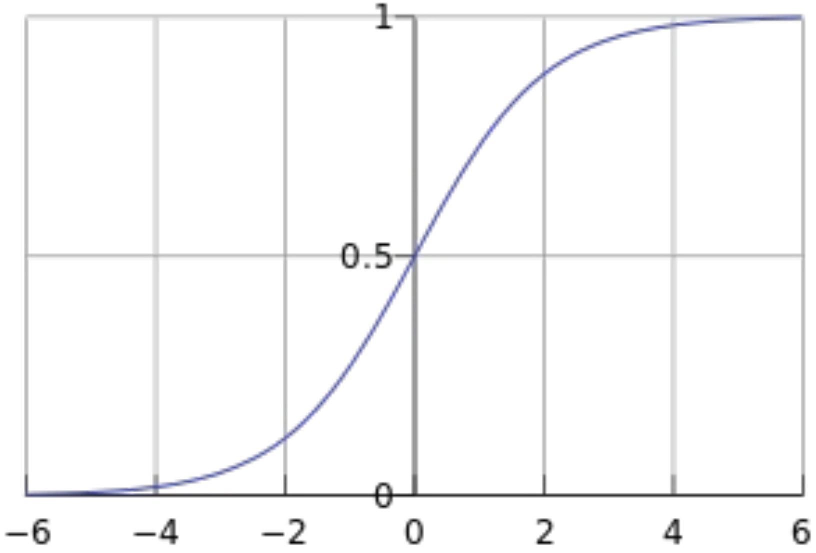
- For a point $\mathbf{x}$ on the decision boundary of logistic regression, does $g(\mathbf{w}^T\mathbf{x} + b) = \mathbf{w}^T\mathbf{x} + b$?

A) Yes

B) No

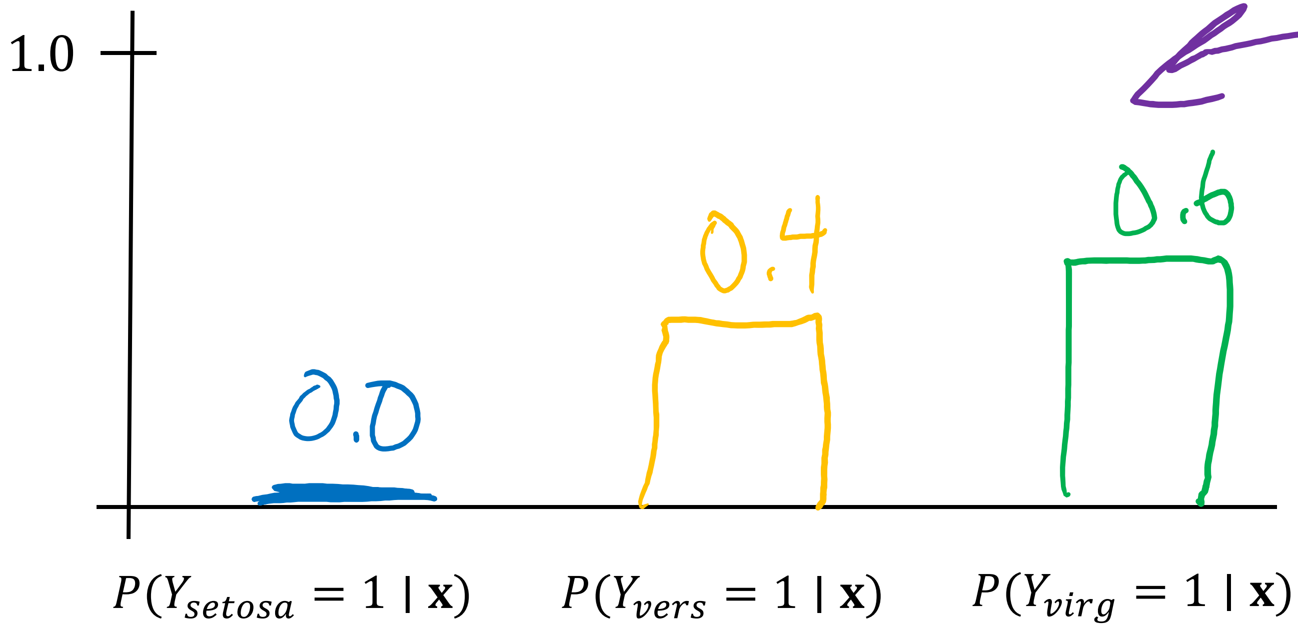
C) I have no idea

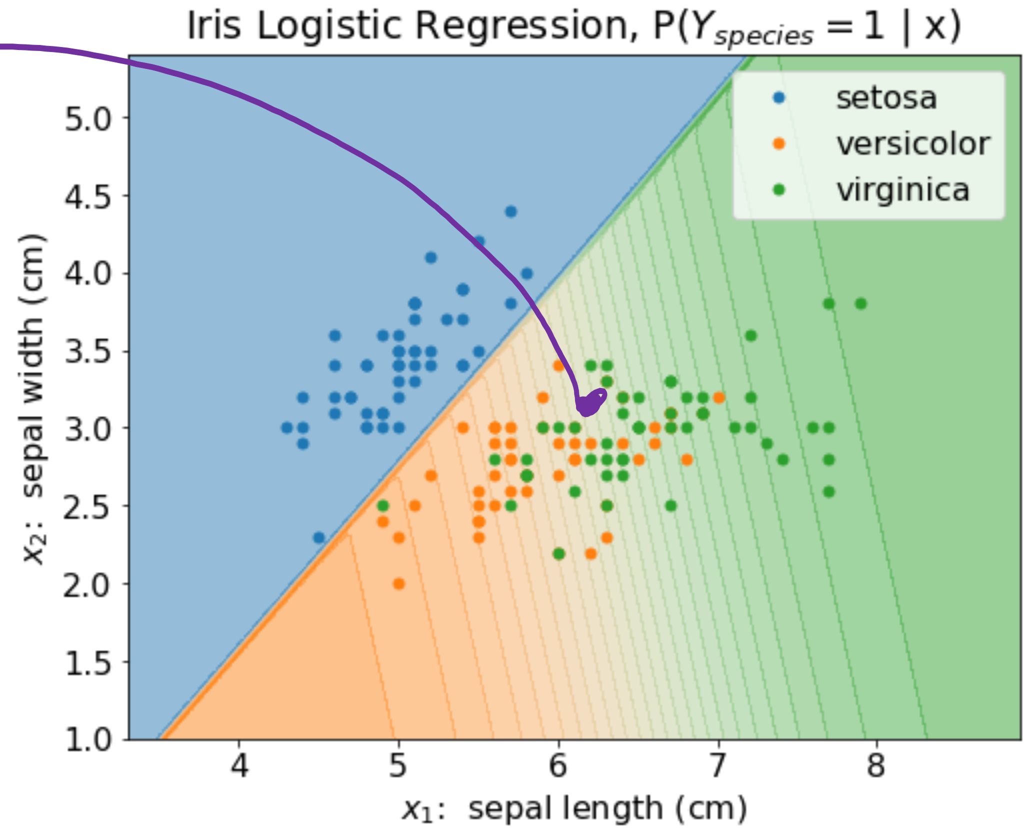
$$g(z) = \frac{1}{1 + e^{-z}}$$



Pre-Reading: Classification "Probability"

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





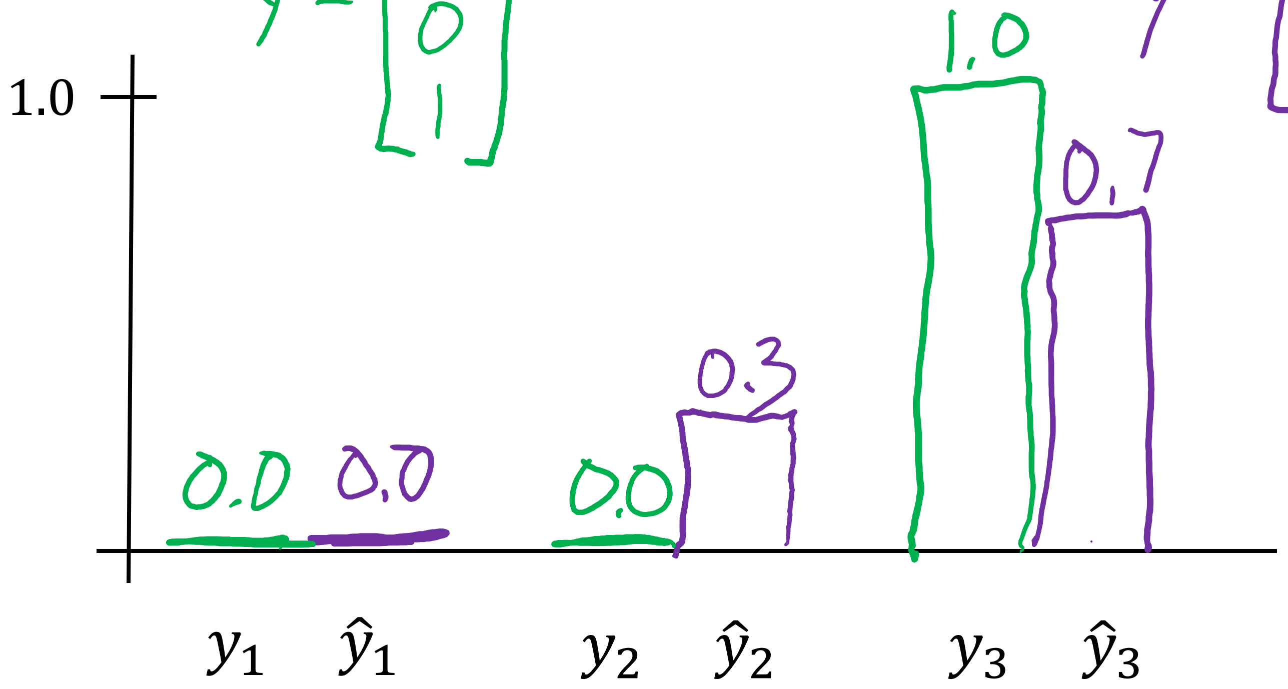
We can still make decisions, .e.g,

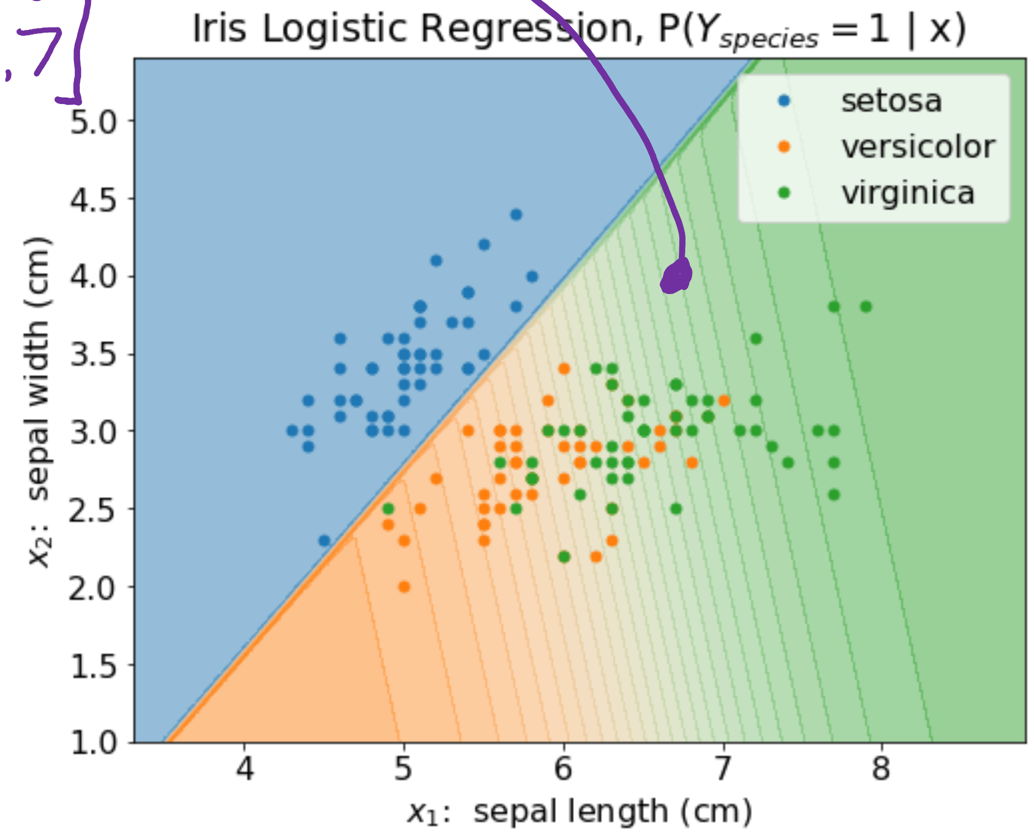
$$\operatorname*{argmax}_k P(Y_k = 1 \mid \mathbf{x})$$

Pre-Reading: Loss for Probabilty Disributions

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

$$y = \begin{bmatrix} 0 \\ 0 \\ 1 \end{bmatrix} \qquad \hat{y} = \begin{bmatrix} 0.0 \\ 0.3 \\ 0.7 \end{bmatrix}$$





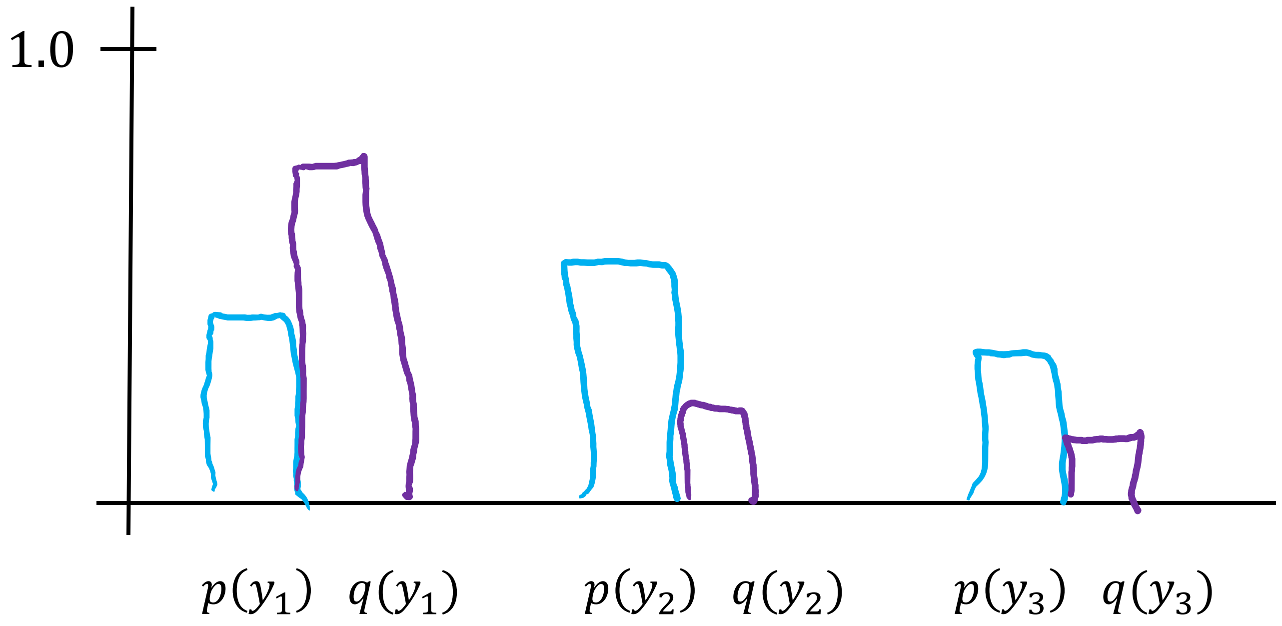
Cross-entropy loss

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

Pre-Reading: Loss for Probabilty Disributions

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

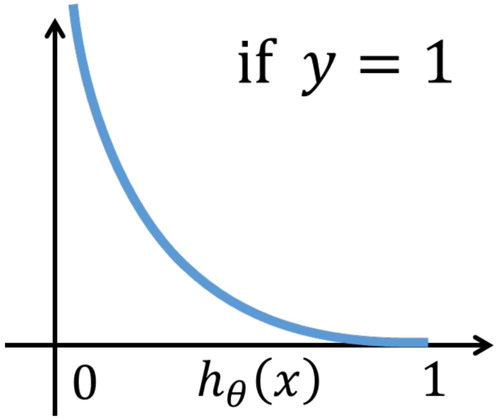
*when used in logistic regression
$\mathbf{y}$ is always a one-hot vector



Cross-entropy loss

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

Cost function for Logistic Regression



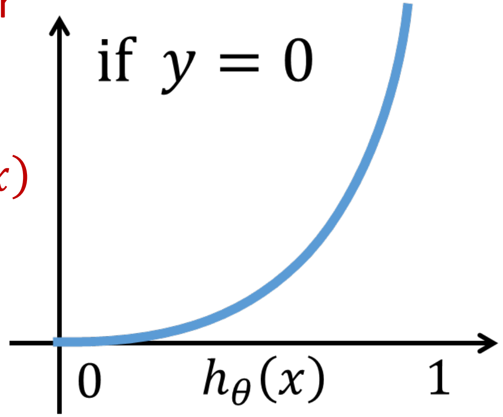
Logistic Regression

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

$$= -y \ \log(f_\theta(x)) - (1 - y) \log(1 - f_\theta(x))$$

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

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



$$J(\theta) = \frac{1}{m} \sum_{i=1}^{m} \mathrm{Cost}(f_\theta(x^{(i)}), y^{(i)}))$$

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

Binary Logistic Regression

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

- Objective: Special case for binary logistic regression

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

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

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

Solve Logistic Regression

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

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

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

Solve Logistic Regression

$z = \theta^T x$

$\hat{y} = g(z)$

$\frac{d}{du} u^T v = v$ or v^T

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

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

$$\boxed{\frac{dg}{dz} = g(z)\big(1 - g(z)\big)}$$

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

$\ell(y^{(i)}, \hat{y}^{(i)})$

$$\frac{\partial J}{\partial \theta} = \frac{\partial \ell}{\partial \hat{y}} \frac{\partial \hat{y}}{\partial \theta}$$

$$= -\left[\frac{y}{\hat{y}} - \frac{1-y}{1-\hat{y}}\right] \frac{\partial \hat{y}}{\partial z} \frac{\partial z}{\partial \theta}$$

$$= -\left[\frac{y}{\hat{y}} - \frac{1-y}{1-\hat{y}}\right] \hat{y}(1-\hat{y}) \vec{x}$$

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

Solve Logistic Regression

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

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

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

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

No closed form solution ☹

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

Gradient descent

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

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

Good news: Convex function!

Bad news: No analytical solution

Gradient descent

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

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

Gradient descent

Repeat {

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

}

(Simultaneously update all θ_j)

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

Gradient descent for Linear Regression

Repeat {

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

}

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

Gradient descent for Logistic Regression

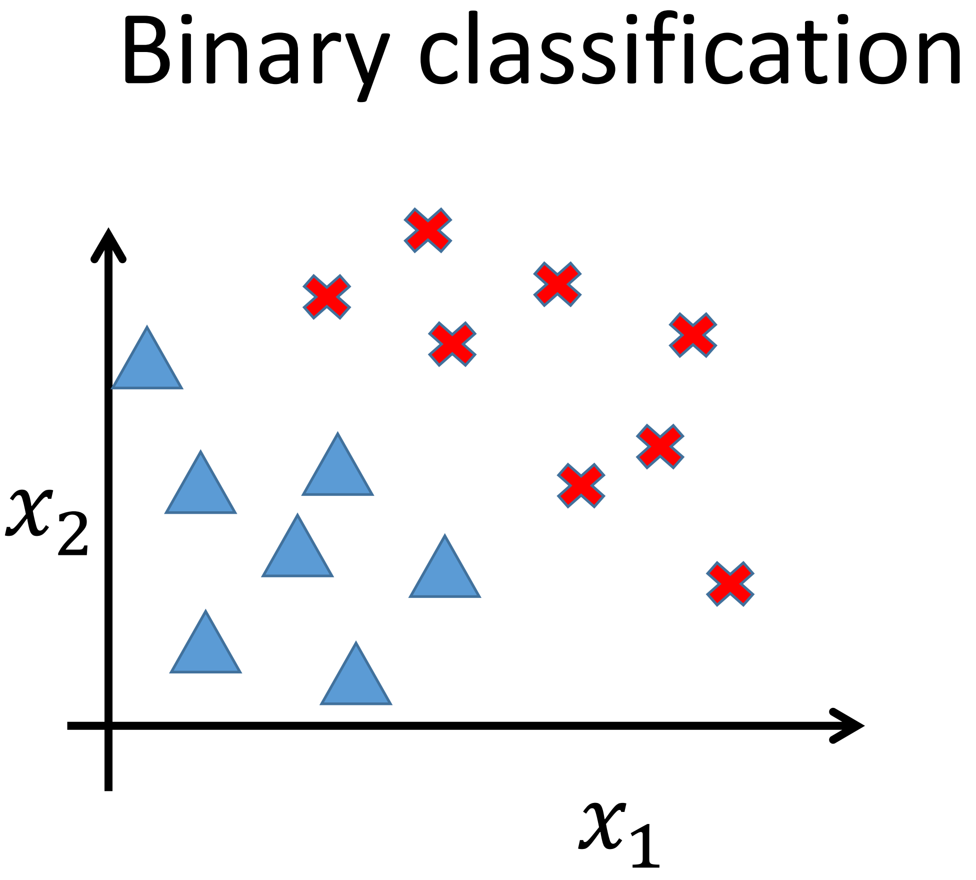
Repeat {

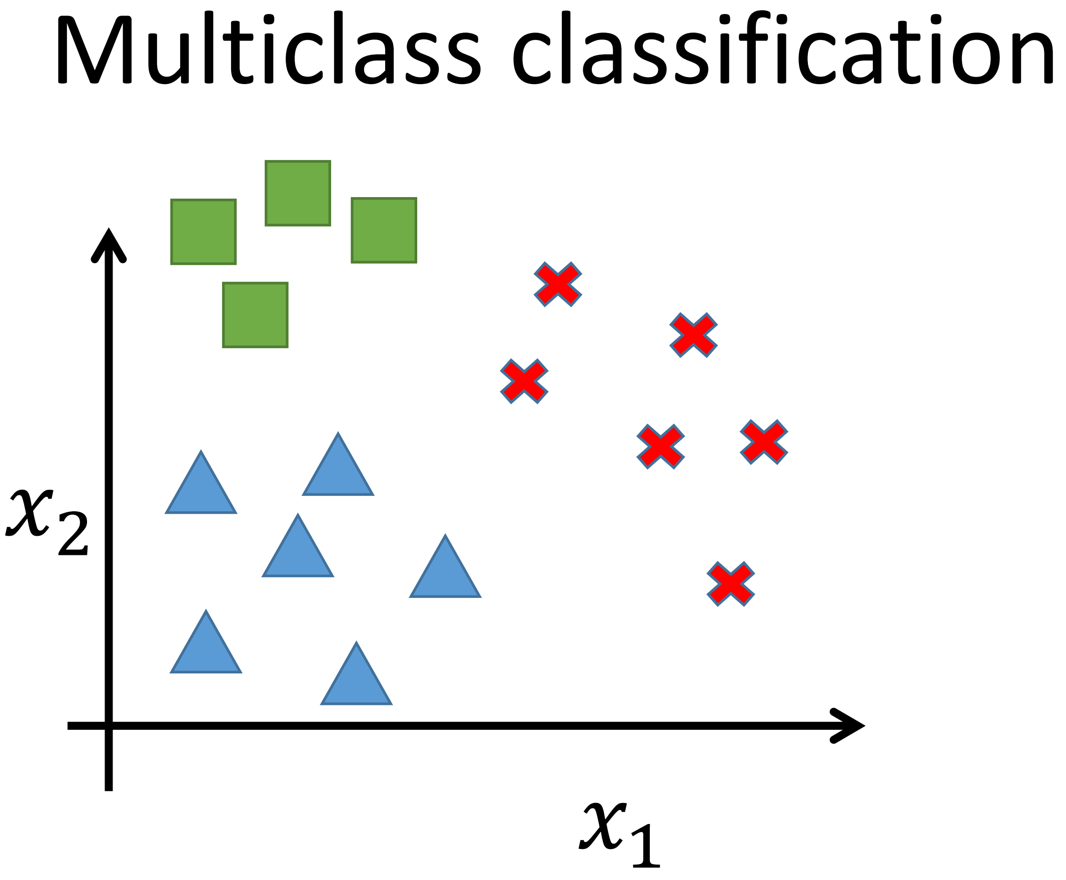
$$\theta_j := \theta_j - \alpha \frac{1}{m} \sum_{i=1}^{m} \left(f_\theta\left(x^{(i)}\right) - y^{(i)}\right) x_j^{(i)}$$

}

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

Multiclass classification

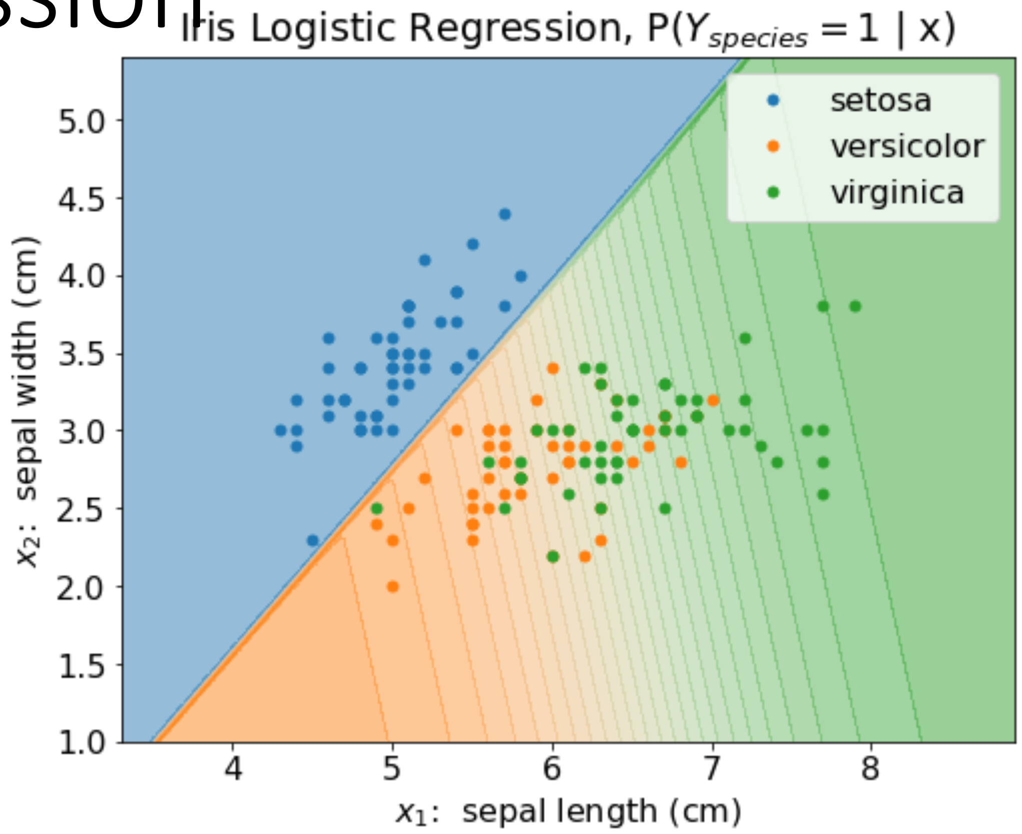




Multi-class Logistic Regression

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

- Model



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

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

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

One vector of parameters for each class

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

$$\mathbf{x} = \begin{bmatrix} 1 \\ \mathrm{x}_1 \\ \mathrm{x}_2 \end{bmatrix} \qquad \boldsymbol{\theta}_k = \begin{bmatrix} b_k \\ w_{k,1} \\ w_{k,2} \end{bmatrix}$$

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

Multi-class Classification

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

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

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

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

$$\boxed{\mathrm{J}(\theta) = -\left[\sum_{i=1}^m \sum_{j=1}^K 1\{y^{(i)} = k\} \log \frac{\exp(\theta_k^T x^{(i)})}{\sum_{j=1}^K \exp(\theta_j^T x^{(i)})}\right]}$$

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 $\mathbf{x}$

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