

COMS 4030A/7047A

Adaptive Computation and Machine Learning

Hima Vadapalli

Semester I, 2022

Session 1 (refer to course notes):
Linear Regression – Recap
Locally Weighted Regression

Session 2 (slides):
Classification
Logistic Regression

Classification

Logistic Regression

Classification

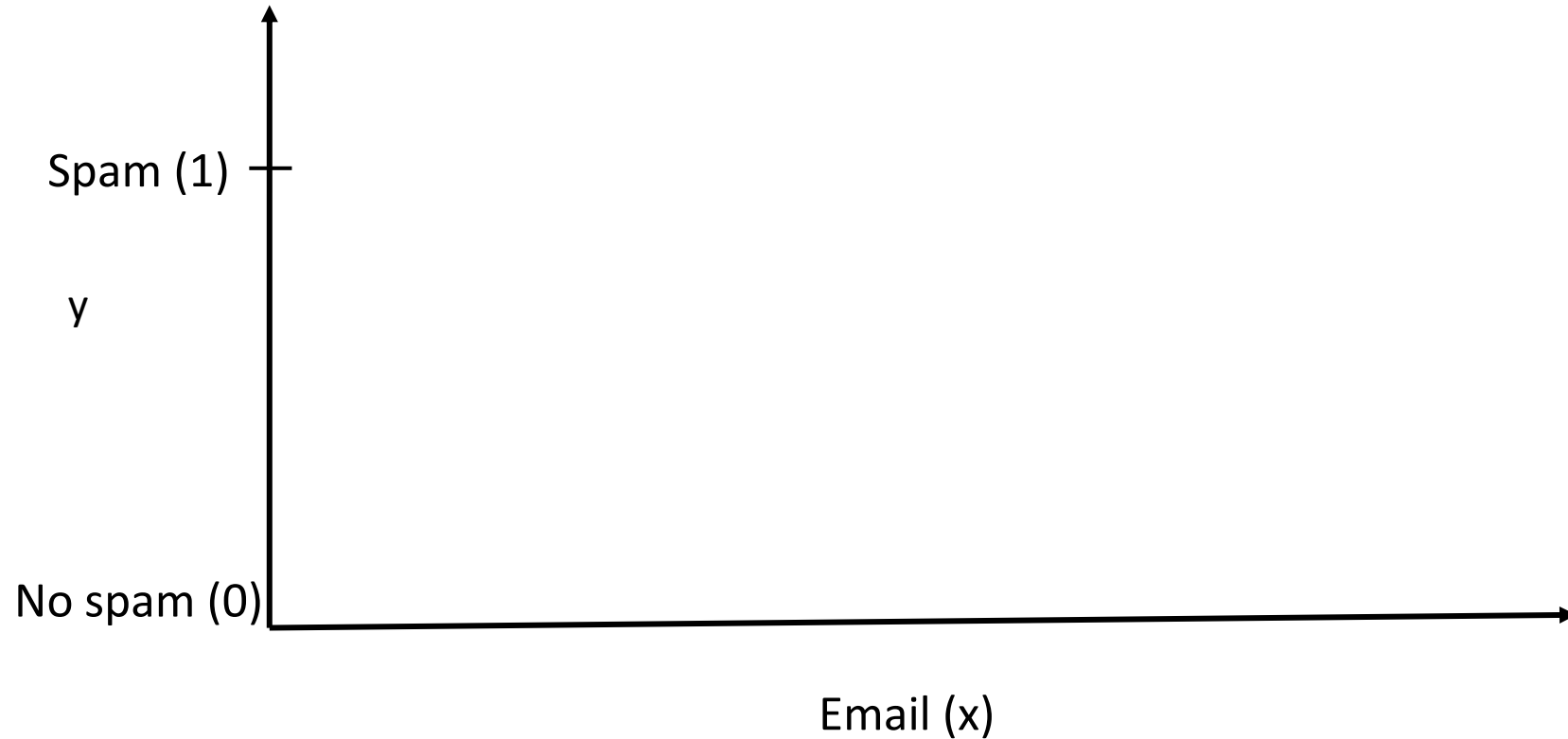
Just like regression problem except that values of y are a set of discrete values

$Y \in \{0,1\}$: Classification (Binary)

- Spam / no spam
- Face / no face

1 – positive class (label), 0 – negative class (label)

Why can't we use linear regression?



Classification Based on Probability

- Instead of just predicting the class, give the probability of the instance being that class
 - i.e., learn $p(y \mid \boldsymbol{x})$

- Recall that:

$$0 \leq p(\text{event}) \leq 1$$

$$p(\text{event}) + p(\neg \text{event}) = 1$$

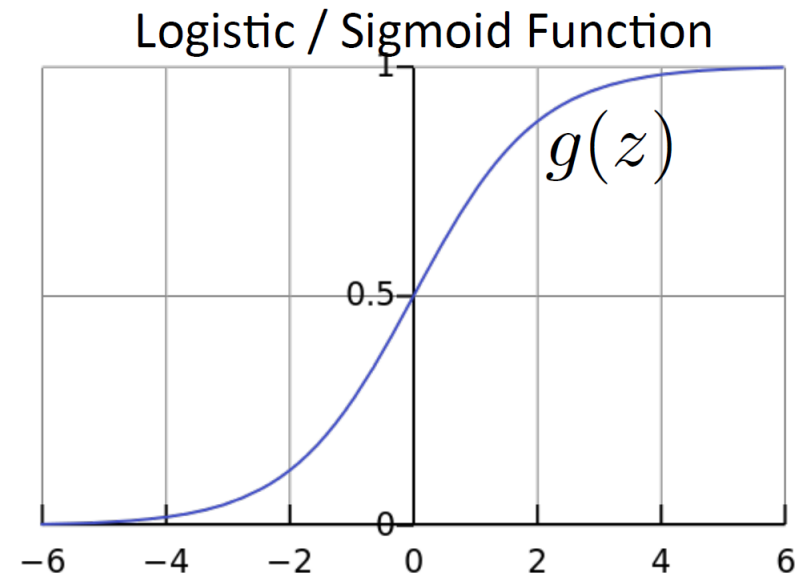
Logistic Regression

- Takes a probabilistic approach to learning discriminative functions (i.e., a classifier)
- $h_{\theta}(\mathbf{x})$ should give $p(y = 1 \mid \mathbf{x}; \theta)$
 - Want $0 \leq h_{\theta}(\mathbf{x}) \leq 1$
- Logistic regression model:

$$h_{\theta}(\mathbf{x}) = g(\theta^{\top} \mathbf{x})$$

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

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



Interpretation of Hypothesis Output

$$h_{\theta}(\mathbf{x}) = \text{estimated } p(y = 1 \mid \mathbf{x}; \theta)$$

Example: Cancer diagnosis from tumor size

$$\mathbf{x} = \begin{bmatrix} x_0 \\ x_1 \end{bmatrix} = \begin{bmatrix} 1 \\ \text{tumorSize} \end{bmatrix}$$

$$h_{\theta}(\mathbf{x}) = 0.7$$

→ Tell patient that 70% chance of tumor being malignant

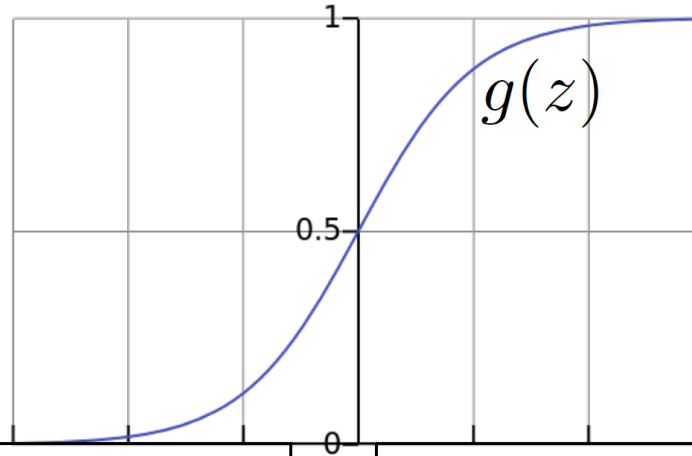
Note that: $p(y = 0 \mid \mathbf{x}; \theta) + p(y = 1 \mid \mathbf{x}; \theta) = 1$

Therefore, $p(y = 0 \mid \mathbf{x}; \theta) = 1 - p(y = 1 \mid \mathbf{x}; \theta)$

Logistic Regression

$$h_{\theta}(\mathbf{x}) = g(\theta^{\top} \mathbf{x})$$

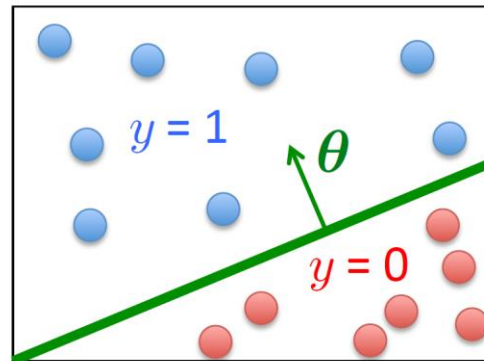
$$g(z) = \frac{1}{1 + e^{-z}}$$



$\theta^{\top} \mathbf{x}$ should be large negative values for negative instances

$\theta^{\top} \mathbf{x}$ should be large positive values for positive instances

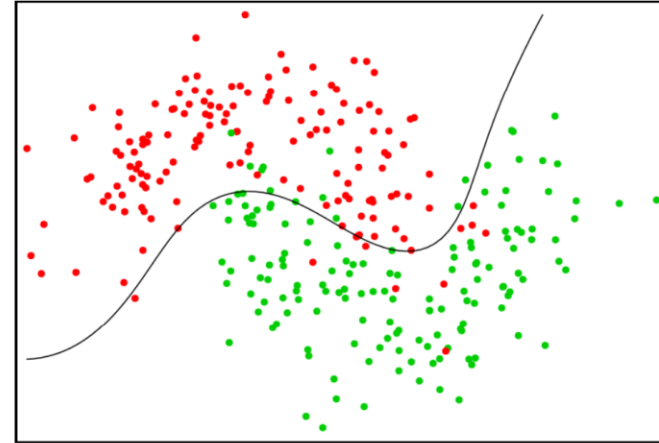
- Assume a threshold and...
 - Predict $y = 1$ if $h_{\theta}(\mathbf{x}) \geq 0.5$
 - Predict $y = 0$ if $h_{\theta}(\mathbf{x}) < 0.5$



Non-Linear Decision Boundary

- Can apply basis function expansion to features, same as with linear regression

$$\mathbf{x} = \begin{bmatrix} 1 \\ x_1 \\ x_2 \end{bmatrix} \rightarrow \begin{bmatrix} 1 \\ x_1 \\ x_2 \\ x_1 x_2 \\ x_1^2 \\ x_2^2 \\ x_1^2 x_2 \\ x_1 x_2^2 \\ \vdots \end{bmatrix}$$



Logistic Regression

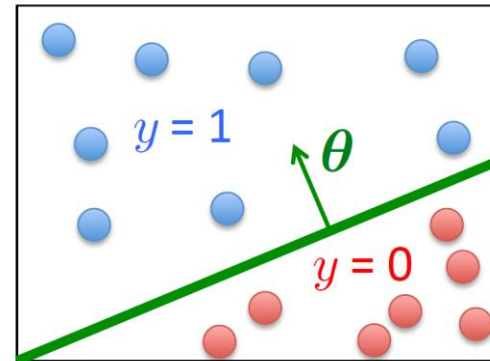
- Given $\left\{ \left(\mathbf{x}^{(1)}, y^{(1)} \right), \left(\mathbf{x}^{(2)}, y^{(2)} \right), \dots, \left(\mathbf{x}^{(n)}, y^{(n)} \right) \right\}$

where $\mathbf{x}^{(i)} \in \mathbb{R}^d$, $y^{(i)} \in \{0, 1\}$

- Model: $h_{\theta}(\mathbf{x}) = g(\theta^{\top} \mathbf{x})$

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

- Assume a threshold and...
 - Predict $y = 1$ if $h_{\theta}(\mathbf{x}) \geq 0.5$
 - Predict $y = 0$ if $h_{\theta}(\mathbf{x}) < 0.5$



Logistic Regression Objective Function

- Can't just use squared loss as in linear regression:

$$J(\boldsymbol{\theta}) = \frac{1}{2n} \sum_{i=1}^n \left(h_{\boldsymbol{\theta}} \left(\boldsymbol{x}^{(i)} \right) - y^{(i)} \right)^2$$

- Using the logistic regression model

$$h_{\boldsymbol{\theta}}(\boldsymbol{x}) = \frac{1}{1 + e^{-\boldsymbol{\theta}^T \boldsymbol{x}}}$$

results in a non-convex optimization

Objective Function

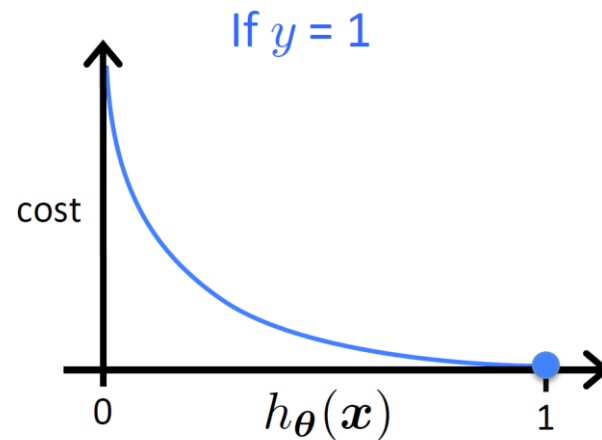
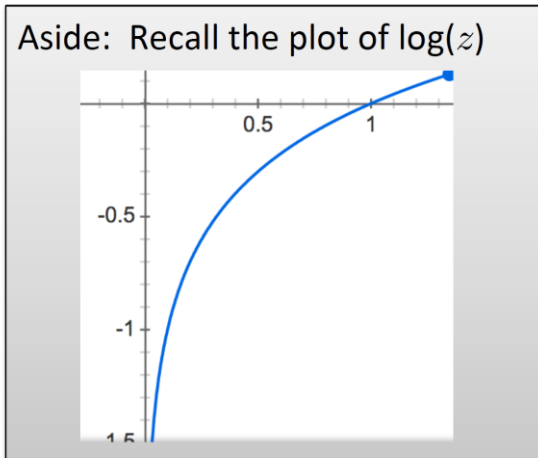
$$J(\boldsymbol{\theta}) = \frac{1}{2n} \sum_{i=1}^n \left(h_{\boldsymbol{\theta}}(\mathbf{x}^{(i)}) - y^{(i)} \right)^2$$

$$\text{cost}(h_{\boldsymbol{\theta}}(\mathbf{x}), y) = \begin{cases} -\log(h_{\boldsymbol{\theta}}(\mathbf{x})) & \text{if } y = 1 \\ -\log(1 - h_{\boldsymbol{\theta}}(\mathbf{x})) & \text{if } y = 0 \end{cases}$$

$$J(\boldsymbol{\theta}) = \sum_{i=1}^n \text{cost}(h_{\boldsymbol{\theta}}(\mathbf{x}^{(i)}), y^{(i)})$$

Intuition Behind the Objective

$$\text{cost}(h_{\theta}(\mathbf{x}), y) = \begin{cases} -\log(h_{\theta}(\mathbf{x})) & \text{if } y = 1 \\ -\log(1 - h_{\theta}(\mathbf{x})) & \text{if } y = 0 \end{cases}$$



If $y = 1$

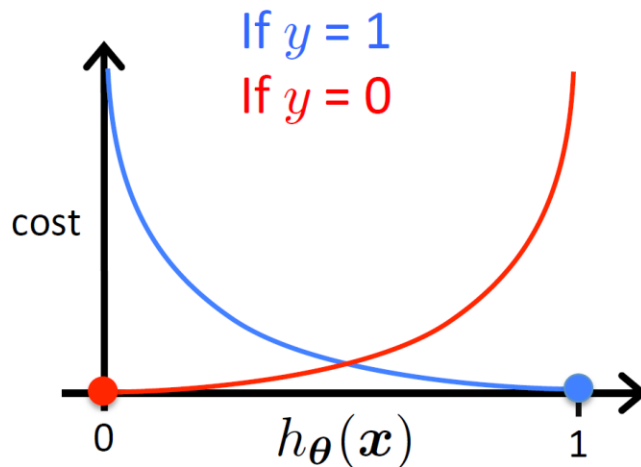
- Cost = 0 if prediction is correct
- As $h_{\theta}(\mathbf{x}) \rightarrow 0$, cost $\rightarrow \infty$
- Captures intuition that larger mistakes should get larger penalties
 - e.g., predict $h_{\theta}(\mathbf{x}) = 0$, but $y = 1$

Intuition Behind the Objective

$$\text{cost}(h_{\theta}(\mathbf{x}), y) = \begin{cases} -\log(h_{\theta}(\mathbf{x})) & \text{if } y = 1 \\ -\log(1 - h_{\theta}(\mathbf{x})) & \text{if } y = 0 \end{cases}$$

If $y = 0$

- Cost = 0 if prediction is correct
- As $(1 - h_{\theta}(\mathbf{x})) \rightarrow 0$, $\text{cost} \rightarrow \infty$
- Captures intuition that larger mistakes should get larger penalties



Objective Function

Compact way:

$$\text{cost}(h_{\theta}(x), y) = -y \log(h_{\theta}(x)) - (1 - y) \log(1 - h_{\theta}(x))$$

$$J(\boldsymbol{\theta}) = - \sum_{i=1}^n \left[y^{(i)} \log h_{\boldsymbol{\theta}}(\mathbf{x}^{(i)}) + (1 - y^{(i)}) \log (1 - h_{\boldsymbol{\theta}}(\mathbf{x}^{(i)})) \right]$$

Gradient Descent for Logistic Regression

To fit parameters θ

Want $\min_{\theta} J(\theta)$

- Initialize θ
- Repeat until convergence

$$\theta_j \leftarrow \theta_j - \alpha \frac{\partial}{\partial \theta_j} J(\theta)$$

simultaneous update
for $j = 0 \dots d$

Regularized Logistic Regression

$$J(\boldsymbol{\theta}) = - \sum_{i=1}^n \left[y^{(i)} \log h_{\boldsymbol{\theta}}(\mathbf{x}^{(i)}) + (1 - y^{(i)}) \log (1 - h_{\boldsymbol{\theta}}(\mathbf{x}^{(i)})) \right]$$

- We can regularize logistic regression exactly as before:

$$\begin{aligned} J_{\text{regularized}}(\boldsymbol{\theta}) &= J(\boldsymbol{\theta}) + \lambda \sum_{j=1}^d \theta_j^2 \\ &= J(\boldsymbol{\theta}) + \lambda \|\boldsymbol{\theta}_{[1:d]}\|_2^2 \end{aligned}$$

Gradient Descent for Logistic Regression

$$J_{\text{reg}}(\boldsymbol{\theta}) = - \sum_{i=1}^n \left[y^{(i)} \log h_{\boldsymbol{\theta}}(\mathbf{x}^{(i)}) + (1 - y^{(i)}) \log (1 - h_{\boldsymbol{\theta}}(\mathbf{x}^{(i)})) \right] + \lambda \|\boldsymbol{\theta}_{[1:d]}\|_2^2$$

Want $\min_{\boldsymbol{\theta}} J(\boldsymbol{\theta})$

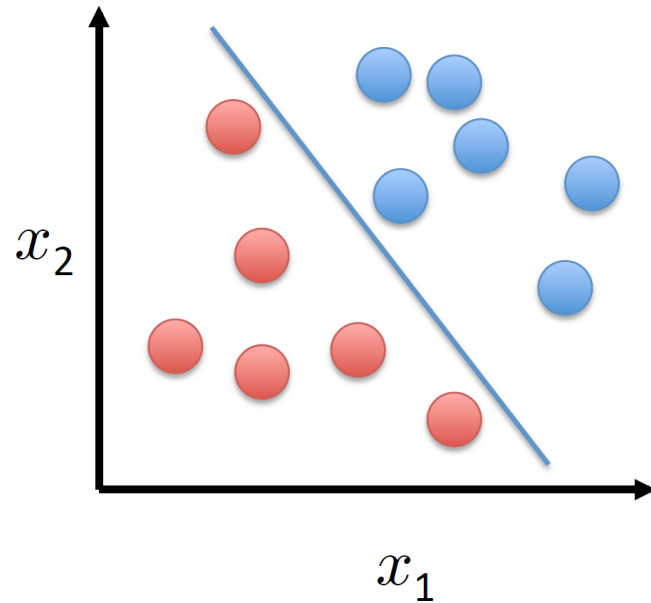
- Initialize $\boldsymbol{\theta}$
- Repeat until convergence (simultaneous update for $j = 0 \dots d$)

$$\theta_0 \leftarrow \theta_0 - \alpha \sum_{i=1}^n \left(h_{\boldsymbol{\theta}}(\mathbf{x}^{(i)}) - y^{(i)} \right)$$

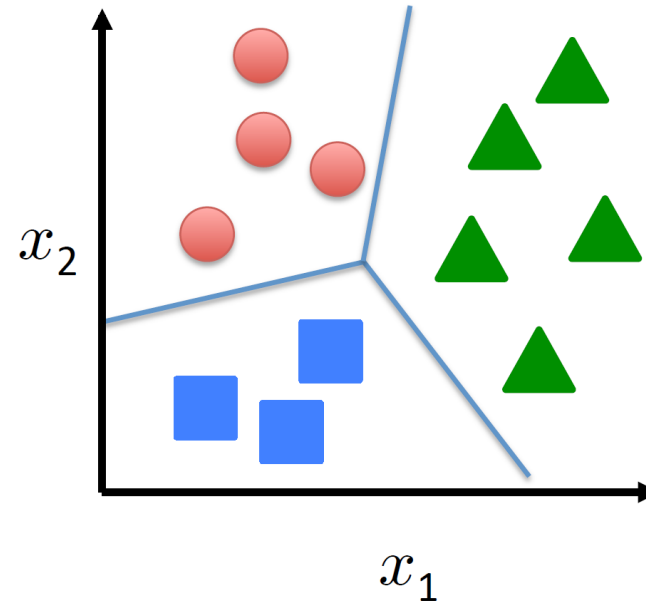
$$\theta_j \leftarrow \theta_j - \alpha \left[\sum_{i=1}^n \left(h_{\boldsymbol{\theta}}(\mathbf{x}^{(i)}) - y^{(i)} \right) x_j^{(i)} - \frac{\lambda}{n} \theta_j \right]$$

Multi-Class Classification

Binary classification:



Multi-class classification:



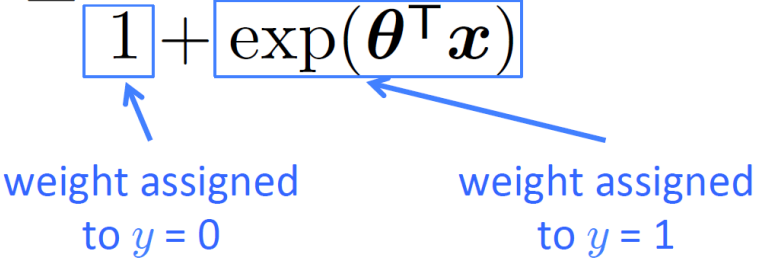
Disease diagnosis: healthy / cold / flu / pneumonia

Object classification: desk / chair / monitor / bookcase

Multi-Class Logistic Regression

- For 2 classes:

$$h_{\theta}(\mathbf{x}) = \frac{1}{1 + \exp(-\theta^{\top} \mathbf{x})} = \frac{\exp(\theta^{\top} \mathbf{x})}{\boxed{1} + \boxed{\exp(\theta^{\top} \mathbf{x})}}$$



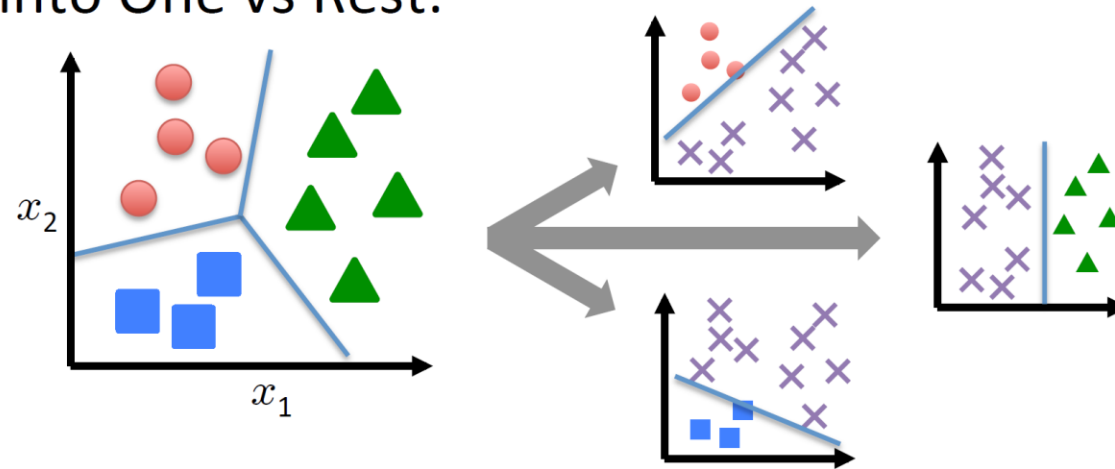
- For C classes $\{1, \dots, C\}$:

$$p(y = c \mid \mathbf{x}; \theta_1, \dots, \theta_C) = \frac{\exp(\theta_c^{\top} \mathbf{x})}{\sum_{c=1}^C \exp(\theta_c^{\top} \mathbf{x})}$$

– Called the **softmax** function

Multi-Class Logistic Regression

Split into One vs Rest:



- Train a logistic regression classifier for each class i to predict the probability that $y = i$ with

$$h_c(\mathbf{x}) = \frac{\exp(\boldsymbol{\theta}_c^\top \mathbf{x})}{\sum_{c=1}^C \exp(\boldsymbol{\theta}_c^\top \mathbf{x})}$$

Implementing Multi-Class Logistic Regression

- Use $h_c(\mathbf{x}) = \frac{\exp(\boldsymbol{\theta}_c^\top \mathbf{x})}{\sum_{c=1}^C \exp(\boldsymbol{\theta}_c^\top \mathbf{x})}$ as the model for class c
- Gradient descent simultaneously updates all parameters for all models
 - Same derivative as before, just with the above $h_c(\mathbf{x})$
- Predict class label as the most probable label

$$\max_c h_c(\mathbf{x})$$