


# Naïve Bayes and Logistic Regression

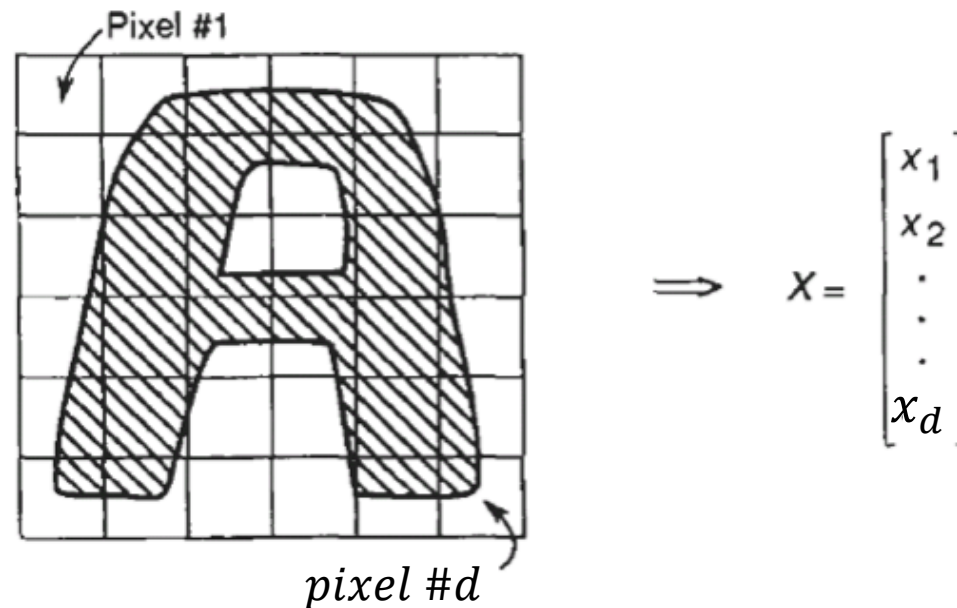
Mahdi Roozbahani  
Georgia Tech

# Outline

- Generative and Discriminative Classification ← 
- The Logistic Regression Model
- Understanding the Objective Function
- Gradient Descent for Parameter Learning
- Multiclass Logistic Regression

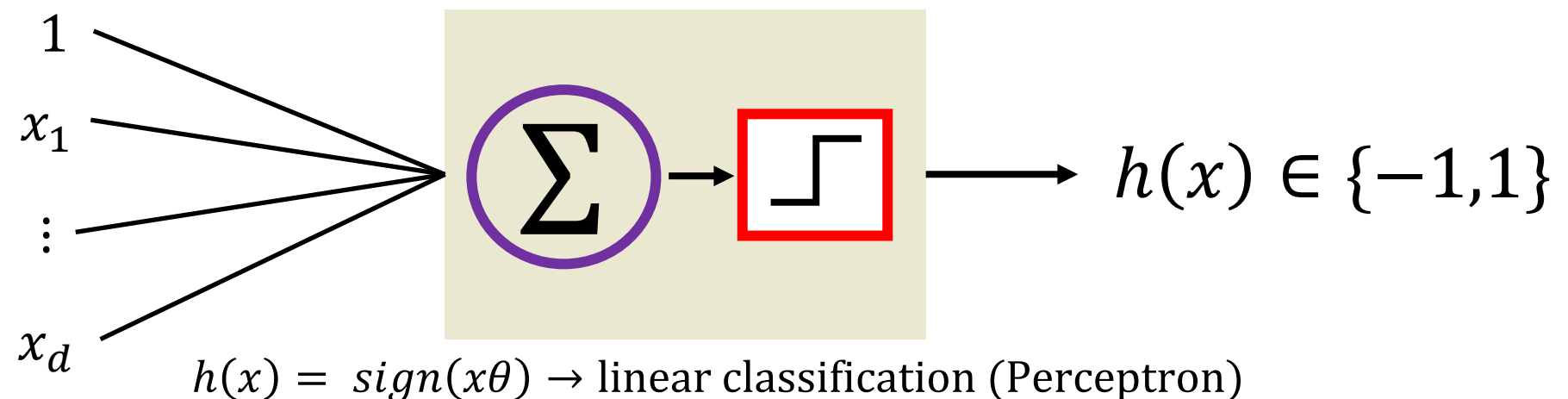
# Classification

- Represent the data



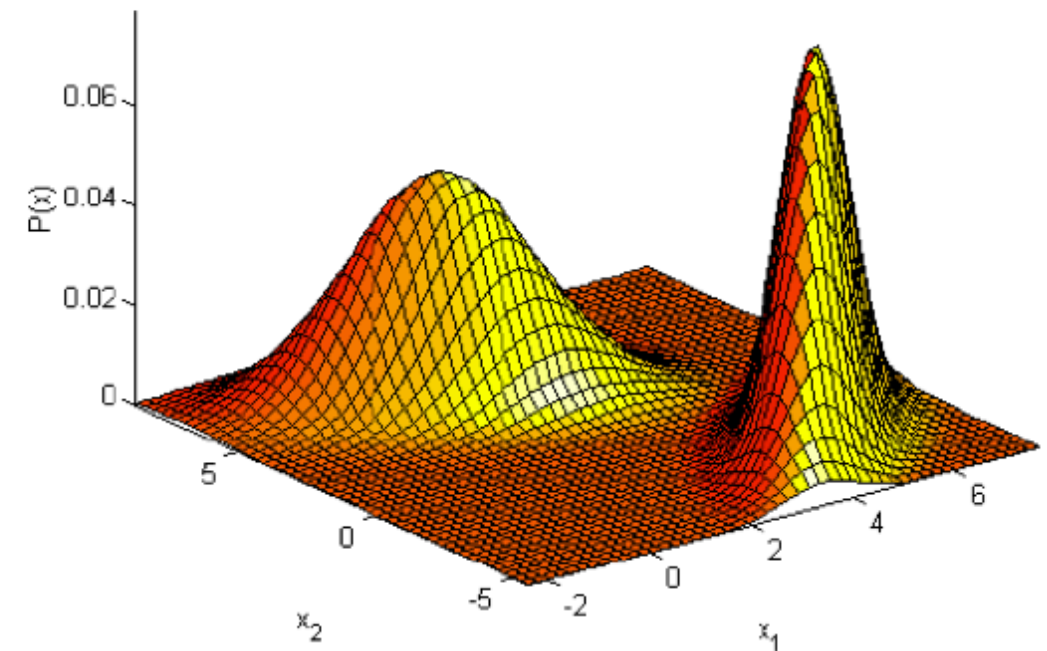
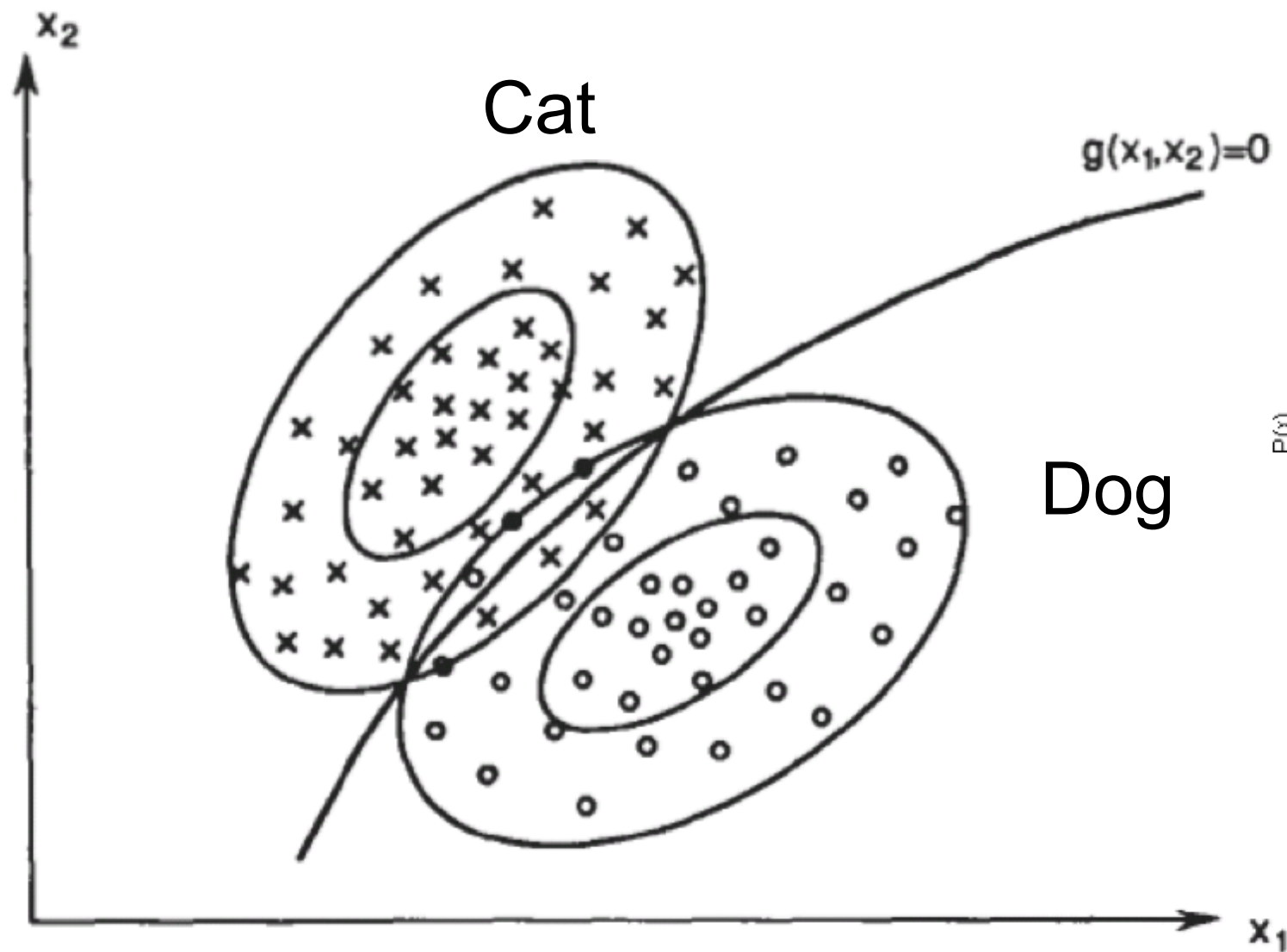
- A label is provided for each data point, eg.,  $y \in \{-1, +1\}$

- Classifier



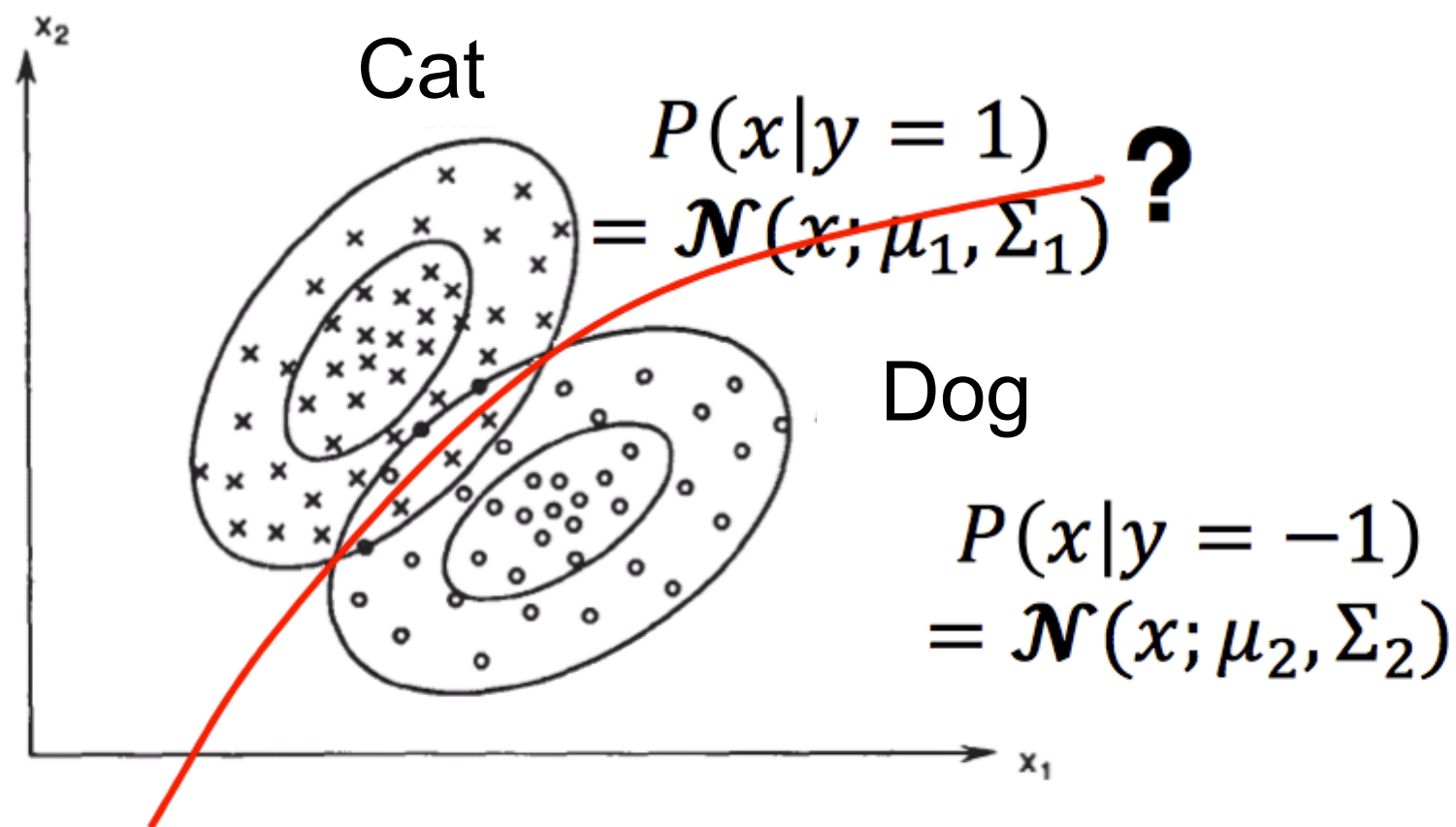
# Decision Making: Dividing the Feature Space

- Distributions of sample from normal (positive class) and abnormal (negative class) tissues



# How to Determine the Decision Boundary?

- Given class conditional distribution:  $P(x|y = 1), P(x|y = -1)$ , and class prior:  $P(y = 1), P(y = -1)$



# Bayes Decision Rule

likelihood

Prior

$$P(y|x) = \frac{P(x|y)P(y)}{P(x)} = \frac{P(x, y)}{\sum_z P(x, y)}$$

posterior

normalization constant

Prior:  $P(y)$

Likelihood (class conditional distribution :  $p(x|y) = \mathcal{N}(x|\mu_y, \Sigma_y)$

$$\text{Posterior: } P(y|x) = \frac{P(y)\mathcal{N}(x|\mu_y, \Sigma_y)}{\sum_y P(y)\mathcal{N}(x|\mu_y, \Sigma_y)}$$

# Bayes Decision Rule

- Learning: prior:  $p(y)$ , class conditional distribution :  $p(x|y)$

- The poster probability of a test point

$$q_i(x) := P(y = i|x) = \frac{P(x|y)P(y)}{P(x)}$$

- Bayes decision rule:

- If  $q_i(x) > q_j(x)$ , then  $y = i$ , otherwise  $y = j$

- Alternatively:

- If ratio  $l(x) = \frac{P(x|y=i)}{P(x|y=j)} > \frac{P(y=j)}{P(y=i)}$ , then  $y = i$ , otherwise  $y = j$

- Or look at the log-likelihood ratio  $h(x) = -\ln \frac{q_i(x)}{q_j(x)}$

# What do People do in Practice?

- Generative models
  - Model prior and likelihood explicitly
  - “Generative” means able to generate synthetic data points
  - Examples: Naive Bayes, Hidden Markov Models
- Discriminative models
  - Directly estimate the posterior probabilities
  - No need to model underlying prior and likelihood distributions
  - Examples: Logistic Regression, SVM, Neural Networks



# Generative Model: Naive Bayes

- Use Bayes decision rule for classification

$$P(y|x) = \frac{P(x|y)P(y)}{P(x)}$$

- But assume  $p(x|y = 1)$  is fully factorized: Dimensions are independent.

$$p(x|y = 1) = \prod_{i=1}^d p(x_i|y = 1)$$


- Or the variables corresponding to each dimension of the data are independent given the label

# “Naïve” conditional independence assumption

$$P(y|x) = \frac{P(x|y)P(y)}{P(x)} = \frac{P(x, y)}{P(x)}$$

Joint probability model:

$$P(x, y_{label=1}) = P(x_1, \dots, x_d, y_{label=1}) = P(x_1 | x_2, \dots, x_d, y_{label=1}) P(x_2, \dots, x_d, y_{label=1})$$


$$= P(x_1 | x_2, \dots, x_d, y_{label=1}) P(x_2 | x_3, \dots, x_d, y_{label=1}) P(x_3, \dots, x_d, y_{label=1})$$
$$= \dots$$

$$= P(x_1 | x_2, \dots, x_d, y_{label=1}) P(x_2 | x_3, \dots, x_d, y_{label=1}) \dots P(x_{d-1} | x_d, y_{label=1}) P(x_d | y_{label=1}) P(y_{label=1})$$

Naïve Bayes assumption: let's rewrite it as:


$$P(x, y_{label=1}) = P(x_1 | y_{label=1}) P(x_2 | y_{label=1}) \dots P(x_d | y_{label=1}) P(y_{label=1}) =$$

$$P(y_{label=1}) \prod_{i=1}^d P(x_i | y_{label=1})$$




Gaussian naïve Bayes  
A typical assumption

Example

# Discriminative Models

- Directly estimate decision boundary  $h(x) = -\ln \frac{q_i(x)}{q_j(x)}$  or posterior distribution  $p(y|x)$ 
  - Logistic regression, Neural networks
  - Do not estimate  $p(x|y)$  and  $p(y)$
- Why discriminative classifier?
  - Avoid difficult density estimation problem  Generative model
  - Empirically achieve better classification results

# Outline

- Generative and Discriminative Classification
- The Logistic Regression Model 
- Understanding the Objective Function 
- Gradient Descent for Parameter Learning 
- Multiclass Logistic Regression

# Gaussian Naïve Bayes

$$P(y = 1|x) = \frac{P(x|y = 1)P(y = 1)}{P(x)} = \frac{P(y = 1) \prod_{i=1}^d P(x_i|y = 1)}{P(x)}$$

$$\begin{aligned} \prod_{i=1}^d p(x_i|y = 1, \mu_{1i}, \sigma_{1i}) \\ = \prod_{i=1}^d \frac{1}{\sqrt{2\pi}\sigma_{1i}} \exp\left(-\frac{1}{2\sigma_{1i}^2} (x_{1i} - \mu_{1i})^2\right) \end{aligned}$$

Prior:  $p(y = 1) = \pi_1$

Posterior:  $p(y = 1 | x, \mu, \sigma, \pi)$

$$= \frac{\pi_1 \prod_{i=1}^d \frac{1}{\sqrt{2\pi}\sigma_{1i}} \exp\left(-\frac{1}{2\sigma_{1i}^2} (x_i - \mu_{1i})^2\right)}{\sum_{\substack{k=1 \\ \text{labels}}}^2 \pi_k \prod_{i=1}^d \frac{1}{\sqrt{2\pi}\sigma_{ki}} \exp\left(-\frac{1}{2\sigma_{ki}^2} (x_i - \mu_{ki})^2\right)}$$

get  $\exp(\ln(u))$  of numerator and denominator

$$= \frac{\exp\left(-\sum_{i=1}^d \left(\frac{1}{2\sigma_{1i}^2} (x_i - \mu_{1i})^2 + \log \sigma_{1i} + C\right) + \log \pi_1\right)}{\sum_{k=1}^2 \exp\left(-\sum_{i=1}^d \left(\frac{1}{2\sigma_{ki}^2} (x_i - \mu_{ki})^2 + \log \sigma_{ki} + C\right) + \log \pi_k\right)}$$

$$= \frac{\exp \left( -\sum_{i=1}^d \left( \frac{1}{2\sigma_i^2} (x_i - \mu_{1i})^2 + \log \sigma_i + C \right) + \log \pi_1 \right)}{\sum_{k=1}^2 \exp \left( -\sum_{i=1}^d \left( \frac{1}{2\sigma_i^2} (x_i - \mu_{ki})^2 + \log \sigma_i + C \right) + \log \pi_k \right)}$$

$$= \frac{1}{1 + \exp \left( \underbrace{-\sum_{i=1}^d \left( x_i \frac{1}{\sigma_i} (\mu_{1i} - \mu_{2i}) \right)}_{\sum_i \theta_i x_i} + \underbrace{\frac{1}{\sigma_i^2} (\mu_{1i}^2 - \mu_{2i}^2)}_{\theta_0} \right) + \log \frac{\pi_2}{\pi_1} \right)}$$

$$P(y = 1|x) = \frac{1}{1 + \exp \left( - \sum_{i=1}^d \left( x_i \frac{1}{\sigma_i} (\mu_{1i} - \mu_{2i}) + \frac{1}{\sigma_i^2} (\mu_{1i}^2 - \mu_{2i}^2) \right) + \log \frac{\pi_2}{\pi_1} \right)}$$

Number of parameters:

$2d + 1 \rightarrow d$  mean,  $d$  variance, and 1 for prior

$$P(y = 1|x) = \frac{1}{1 + \exp[-(\sum_{i=1}^d (\theta_i x_i) + \theta_0)]} = \frac{1}{1 + \exp(-s)}$$

Number of parameters =  $d + 1 \rightarrow \theta_0, \theta_1, \theta_2, \dots, \theta_d$

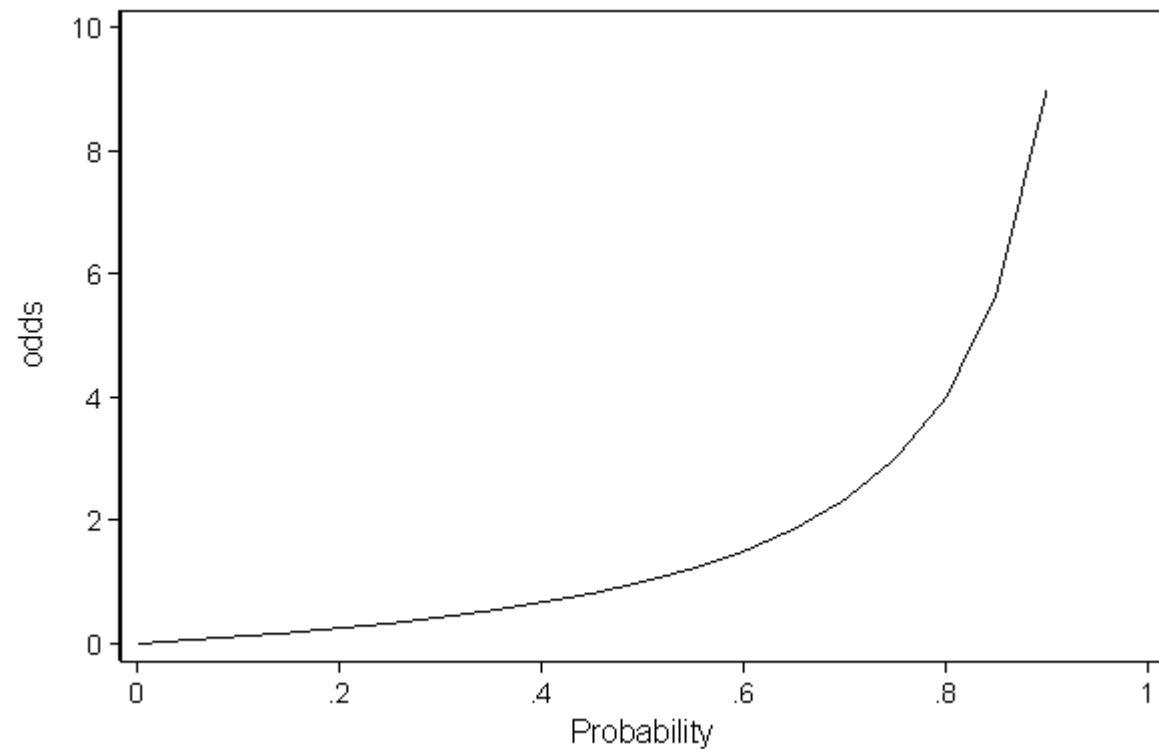
Why not directly learning  $P(y = 1|x)$  or  $\theta$  parameters?

Gaussian Naïve Bayes is a subset of logistic regression



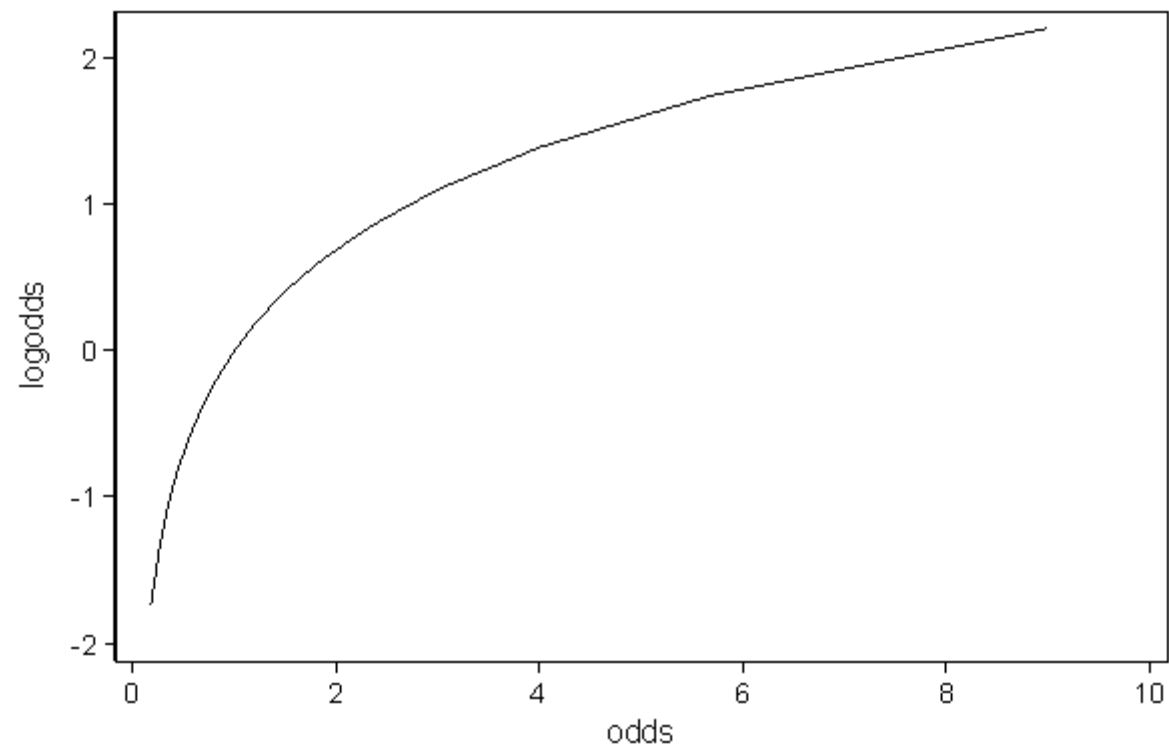
Why  $\frac{1}{1+\exp(-x\theta)}$  is a probability?

$\frac{P(y = 1|x)}{1-P(y = 1|x)}$  is called Odds



*log(odds) vs odds*

What could be  $x\theta$  domain?



# What is logit function?

$$\text{logit}(p) = \log(\text{odds}) = \log\left(\frac{p}{1-p}\right)$$

$$\log\left(\frac{p}{1-p}\right) = \theta_0 + \theta_1 x_1 + \cdots + \theta_d x_d = \sum_{i=0}^d x_i \theta_i = x\theta$$

$$\exp\left(\log\left(\frac{p}{1-p}\right)\right) = \exp(x\theta)$$

$$p = \frac{e^{x\theta}}{1 + e^{x\theta}} = \frac{1}{1 + e^{-x\theta}}$$

# Logistic function for posterior probability

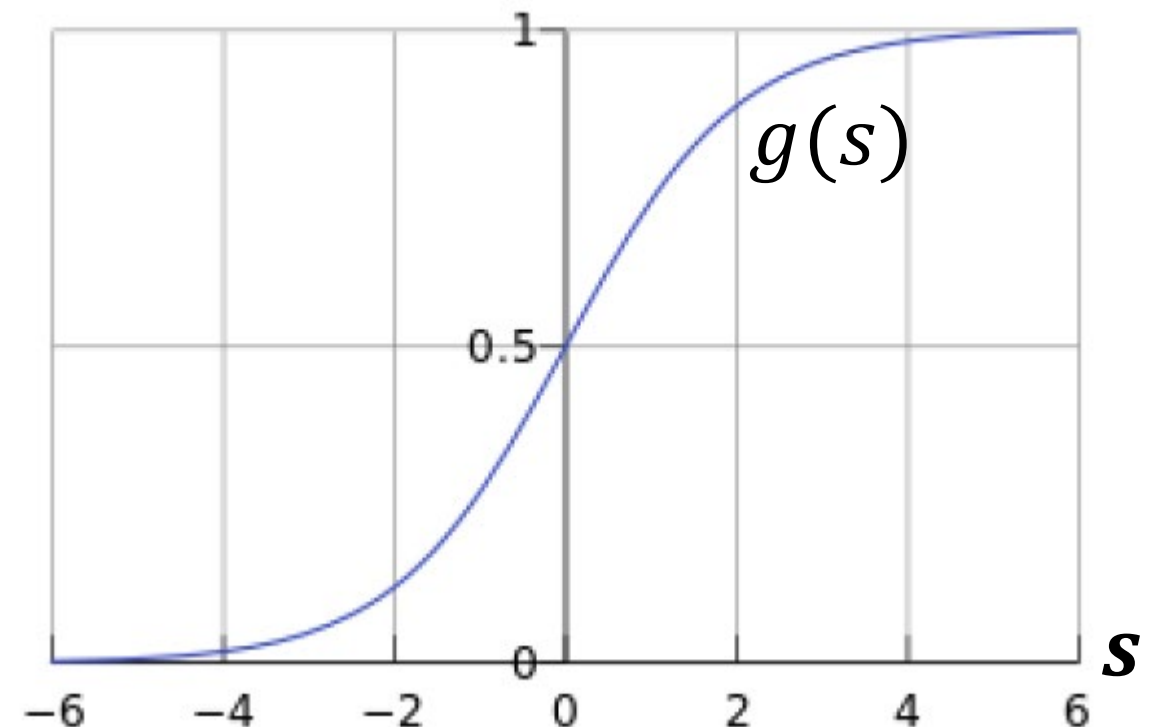
Many equations can give us this shape

Let's use the following function:

$$s = x\theta$$

$$g(s) = P(y = 1|x) = \frac{e^s}{1 + e^s} = \frac{1}{1 + e^{-s}}$$

This formula is called sigmoid function



It is easier to use this function for optimization

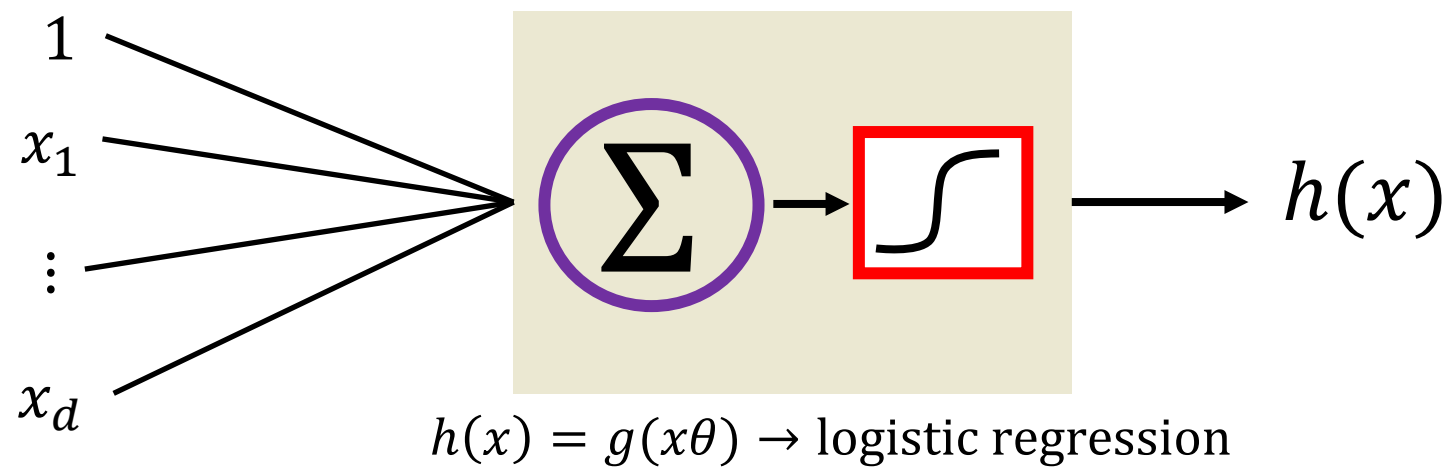
Is 0.5 threshold cut-off a good choice?

[Learn about ROC and AUC \(False positive rate and True positive rate\) \(Interactive\)](#)

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

# Sigmoid Function

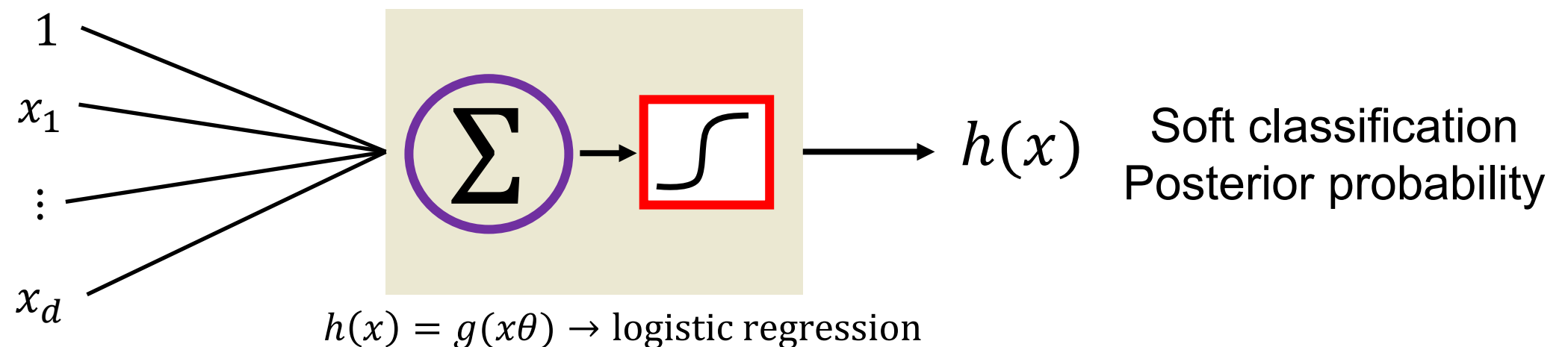
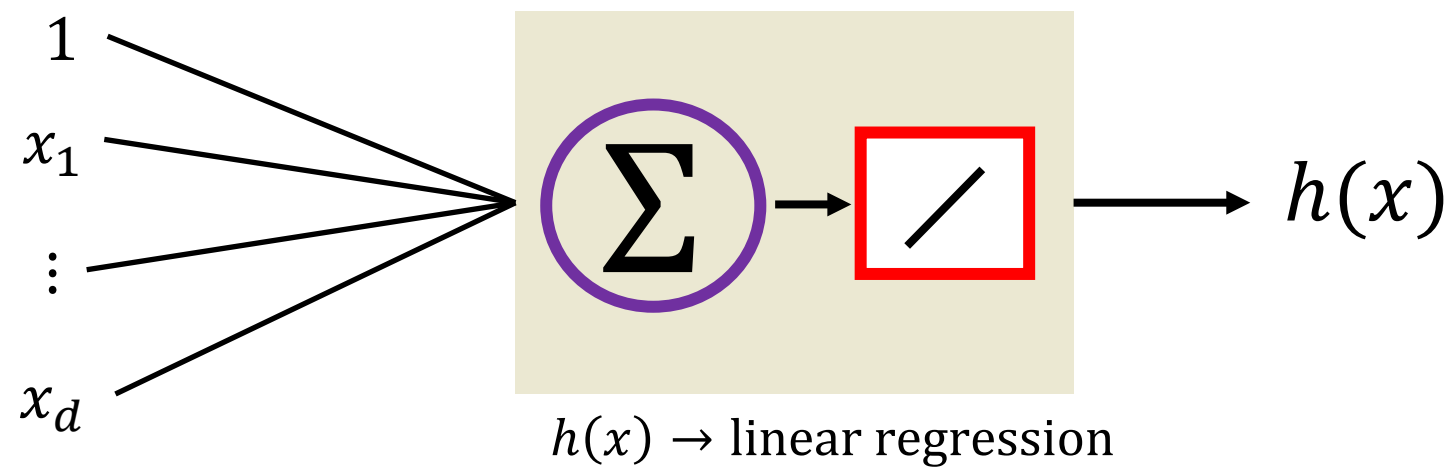
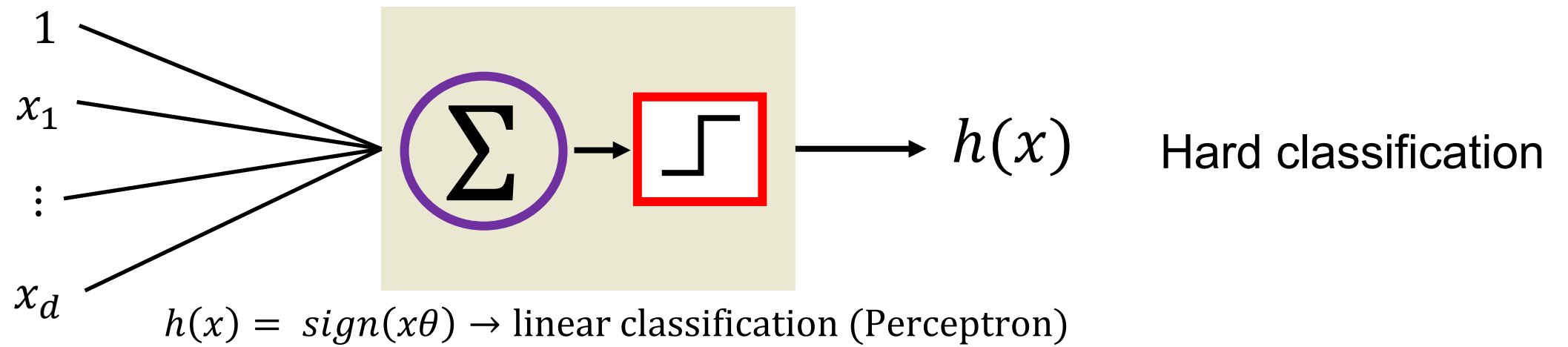
$$s = \sum_{i=0}^d x_i \theta_i = \theta_0 + \theta_1 x_1 + \cdots + \theta_d x_d$$



Soft classification  
Posterior probability

$$s = \sum_{i=0}^d x_i \theta_i = \theta_0 + \theta_1 x_1 + \dots + \theta_d x_d$$

## Three linear models



$g(s)$  is interpreted as probability

**Example:** Prediction of heart attacks

Input  $x$ : cholesterol level, age, weight, finger size, etc.

$g(s)$ : probability of heart attack within a certain time

We can't have a hard prediction here

$s = x\theta$       Let's call this risk score

$$h_{\theta}(x) = p(y|x) = \begin{cases} g(s), & y = 1 \\ 1 - g(s), & y = 0 \end{cases} \quad \text{Using posterior probability directly}$$

# Logistic regression model

$$p(y|x) = \begin{cases} \frac{1}{1 + \exp(-x\theta)} & y = 1 \\ 1 - \frac{1}{1 + \exp(-x\theta)} = \frac{\exp(-x\theta)}{1 + \exp(-x\theta)} & y = 0 \end{cases}$$

We need to find  $\theta$  parameters, let's set up log-likelihood for  $n$  datapoints

$$\begin{aligned} l(\theta) &:= \log \prod_{i=1}^n p(y_i, |x_i, \theta) \\ &= \sum_i \theta^T x_i^T (y_i - 1) - \log(1 + \exp(-x_i \theta)) \end{aligned}$$

This form is concave, negative of this form is convex

## The gradient of $l(\theta)$

$$l(\theta) = \log \prod_{i=1}^n p(y_i, |x_i, \theta)$$
$$= \sum_i \theta^T x_i^T (y_i - 1) - \log(1 + \exp(-x_i \theta))$$

- Gradient

$$\frac{\partial l(\theta)}{\partial \theta} = \sum_i x_i^T (y_i - 1) + x_i^T \frac{\exp(-x_i \theta)}{1 + \exp(-x_i \theta)}$$

- Setting it to 0 does not lead to closed form solution

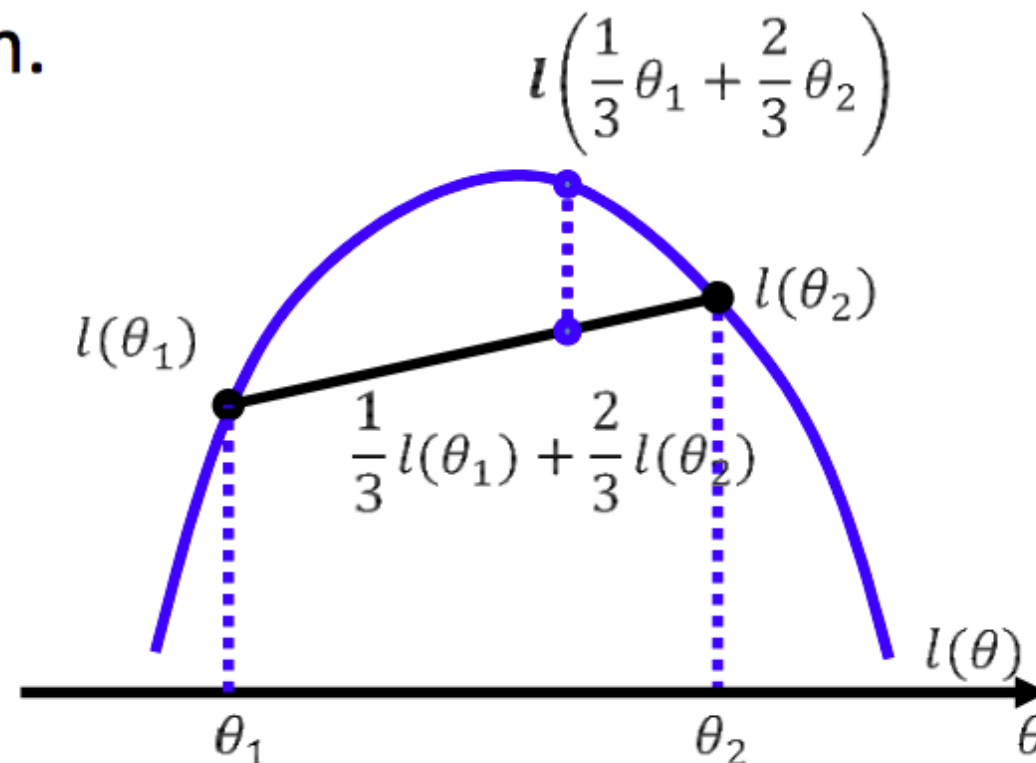


# The Objective Function

- Find  $\theta$ , such that the conditional likelihood of the labels is maximized

$$\max_{\theta} l(\theta) := \log \prod_{i=1}^n p(y_i, |x_i, \theta)$$

- Good news:  $l(\theta)$  is concave function of  $\theta$ , and there is a single global optimum.



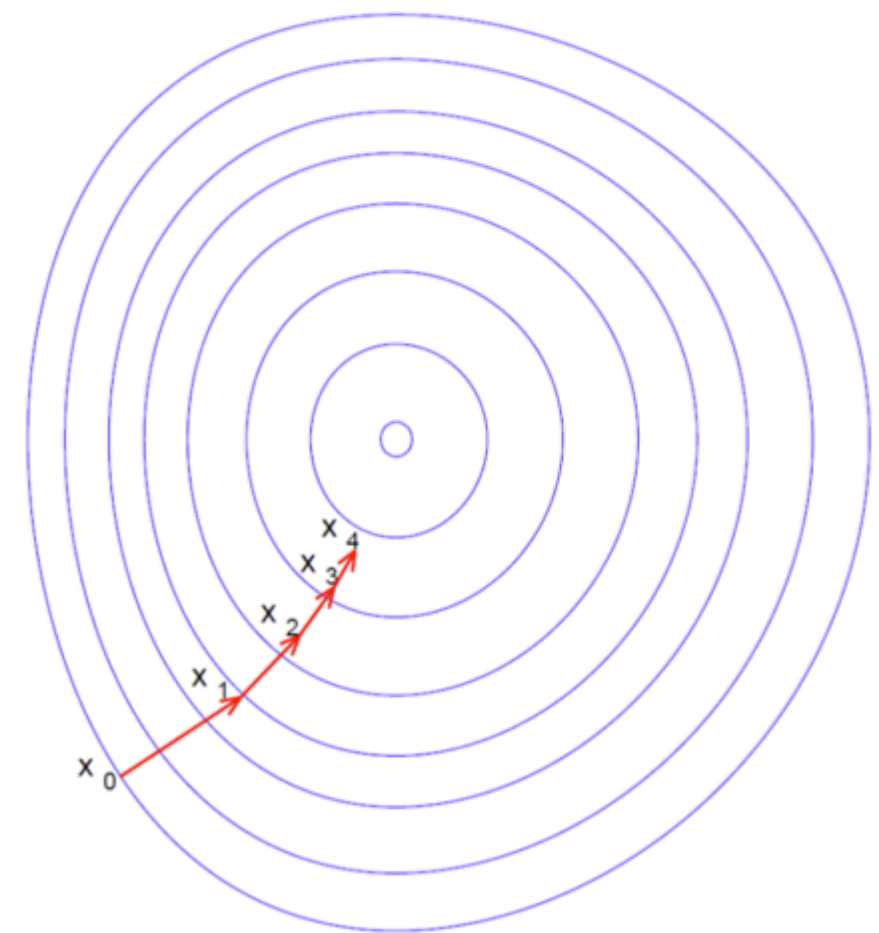
- Bad new: no closed form solution (resort to numerical method)

# Gradient Descent

- One way to solve an *unconstrained* optimization problem is gradient descent
- Given an initial guess, we *iteratively* refine the guess by taking the direction of the negative gradient
- Think about going down a hill by taking the steepest direction at each step
- Update rule

$$x_{k+1} = x_k - \gamma_k \nabla f(x_k)$$

$\gamma_k$  is called the step size or learning rate



# Gradient Ascent(concave)/Descent(convex) algorithm

- Initialize parameter  $\theta^0$

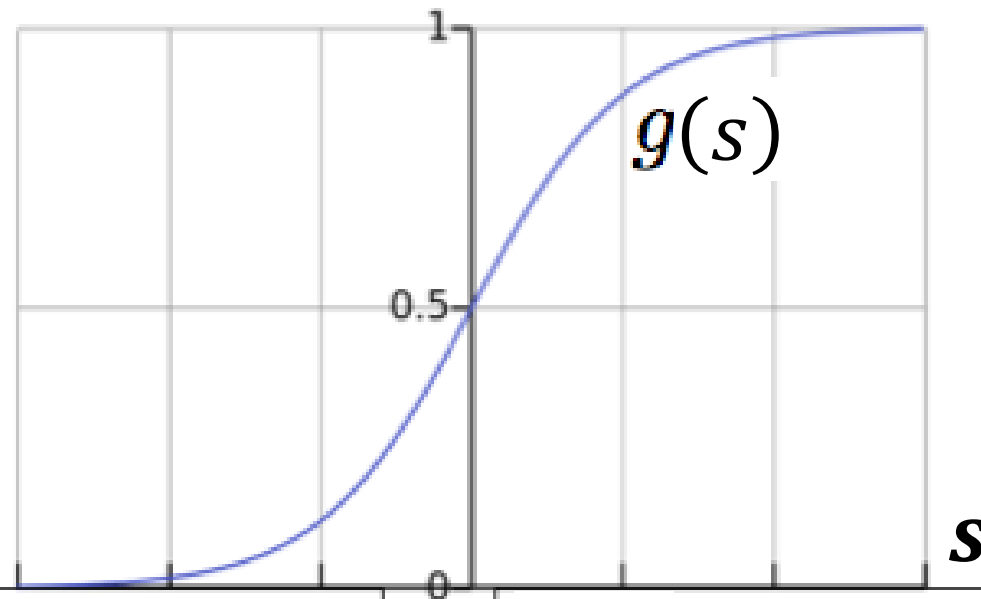
- Do

$$\theta^{t+1} \leftarrow \theta^t + \eta \sum_i x_i^T (y_i - 1) + x_i^T \frac{\exp(-x_i \theta)}{1 + \exp(-x_i \theta)}$$

- While the  $||\theta^{t+1} - \theta^t|| > \epsilon$

# Logistic Regression

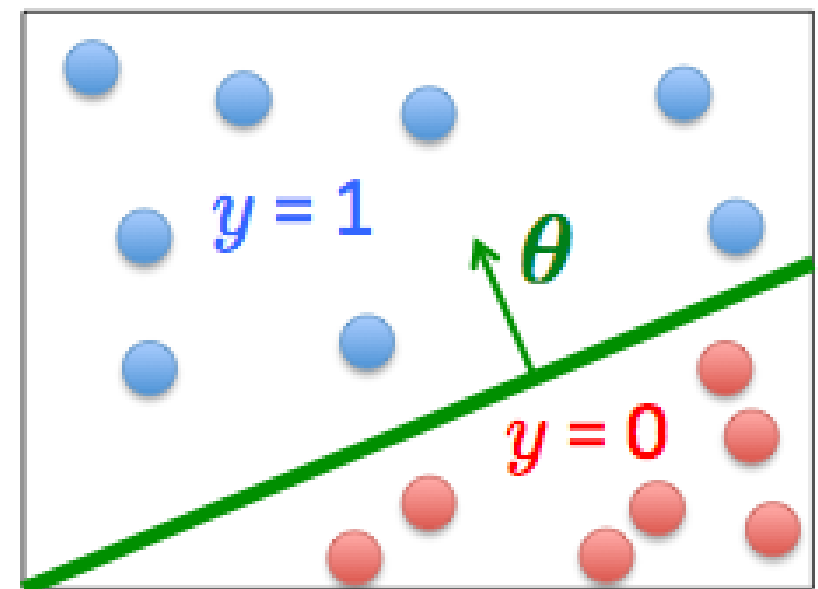
$$g(s) = \frac{e^s}{1 + e^s} = \frac{1}{1 + e^{-s}}$$
$$s = x\theta$$



$x\theta$  should be large negative  
values for negative instances

$x\theta$  should be large positive  
values for positive instances

- Assume a threshold and...
  - Predict  $y = 1$  if  $g(s) \geq 0.5$
  - Predict  $y = 0$  if  $g(s) < 0.5$

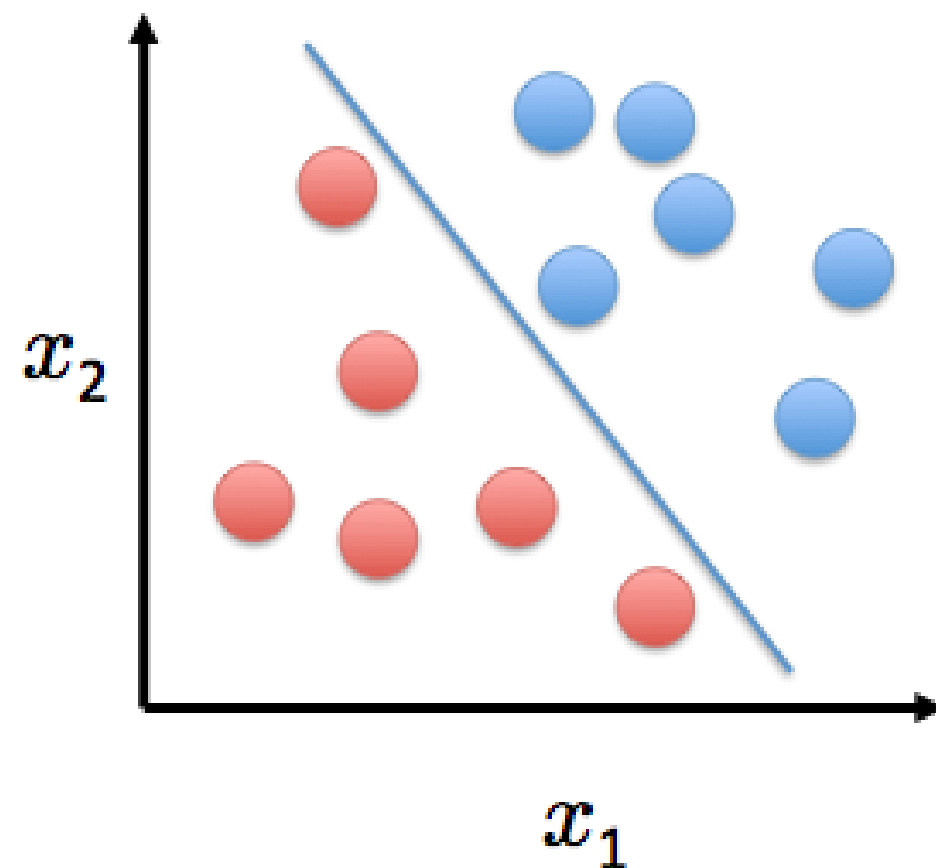


# Outline

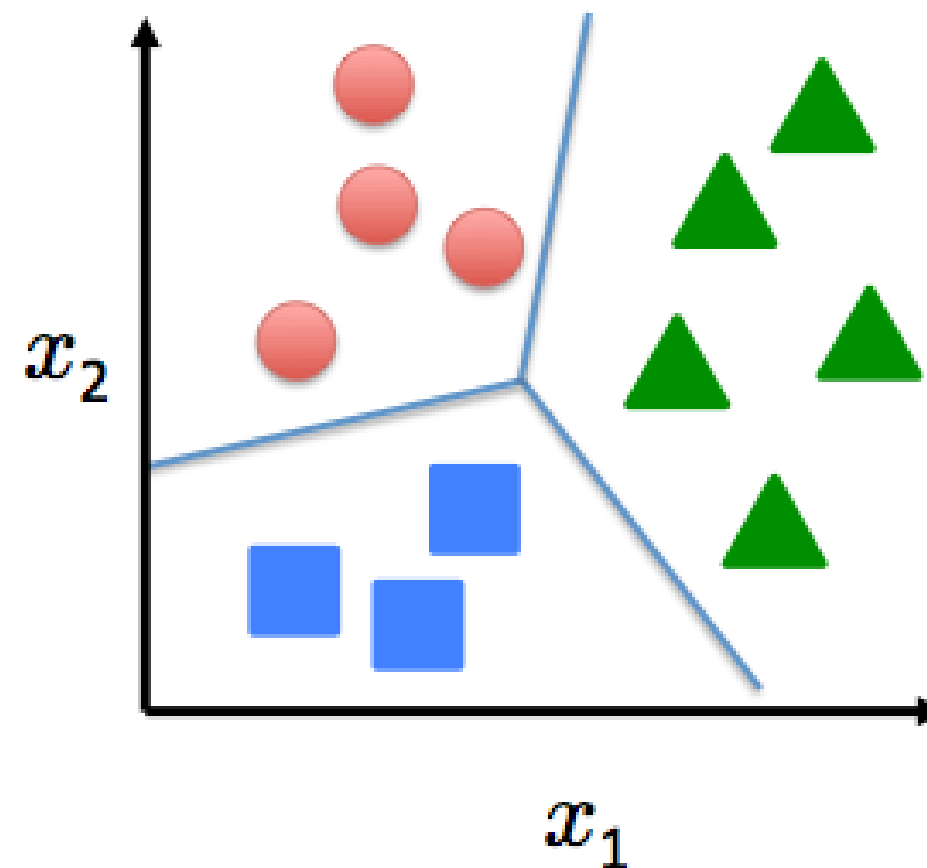
- Generative and Discriminative Classification
- The Logistic Regression Model
- Understanding the Objective Function
- Gradient Descent for Parameter Learning
- Multiclass Logistic Regression ←

# Multiclass Logistic Regression

Binary classification:



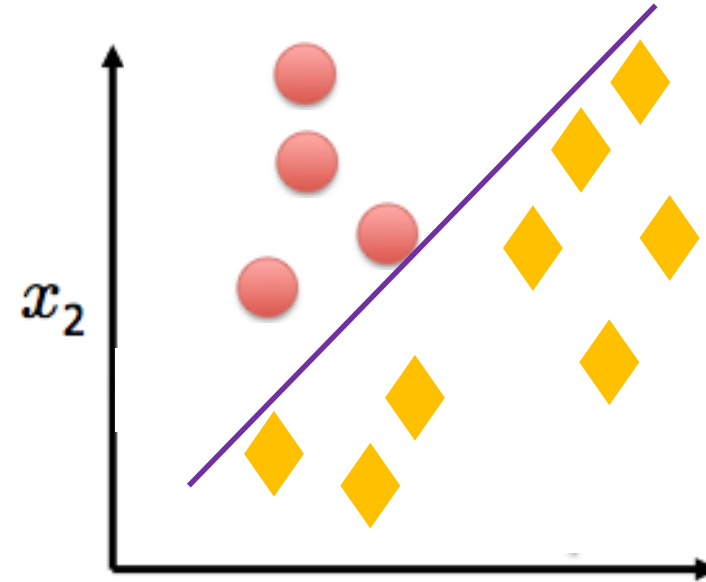
Multi-class classification:



Disease diagnosis: healthy / cold / flu / pneumonia

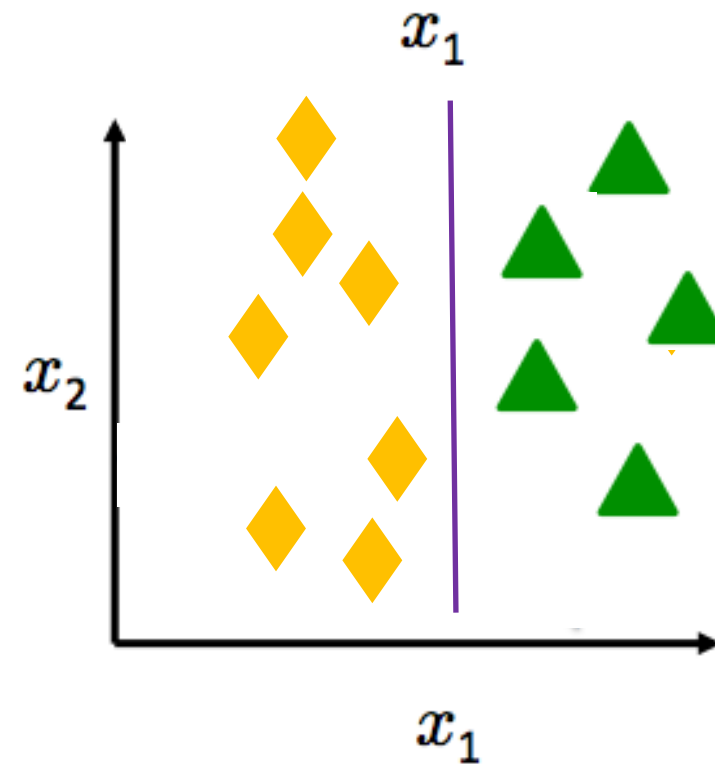
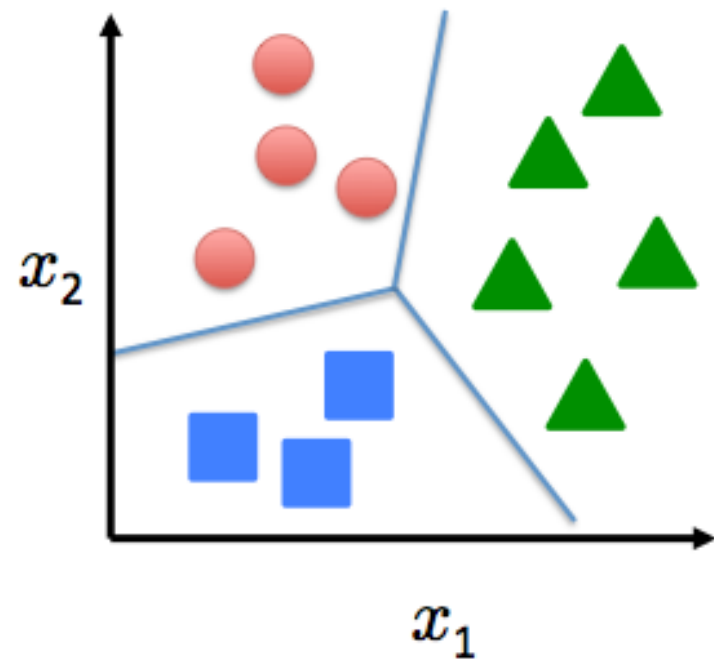
Object classification: desk / chair / monitor / bookcase

# One-vs-all (one-vs-rest)

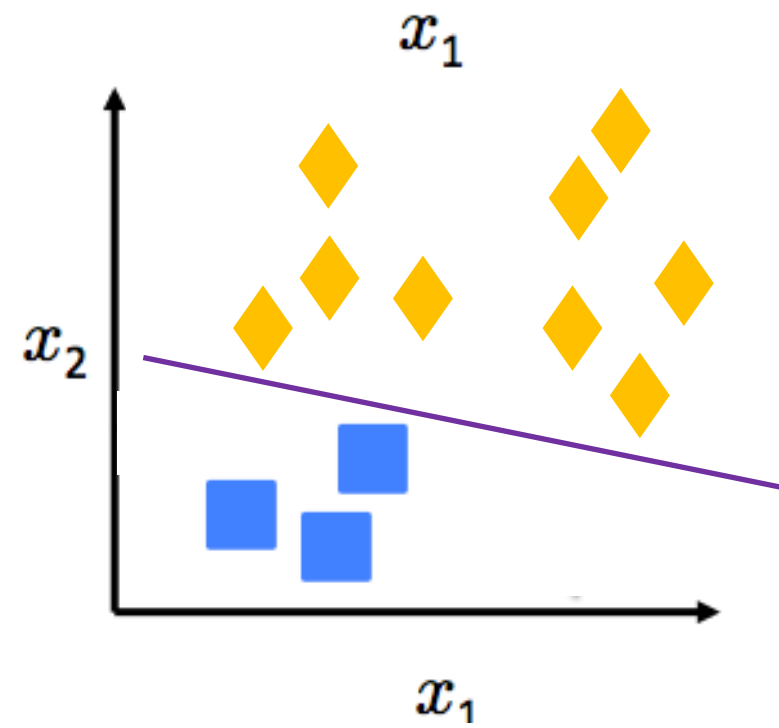


$$h_{\theta}^1(x)$$

Multi-class classification:



$$h_{\theta}^2(x)$$



$$h_{\theta}^3(x)$$

$$h_{\theta}^{(m)}(x) = p(y = 1|x, \theta) \quad (m = 1, 2, 3)$$

## One-vs-all (one-vs-rest)

Train a logistic regression  $h_{\theta}^{(m)}(x)$  for each class  $m$

To predict the label of a new input  $x$ , pick class  $m$  that maximizes:

$$\max_i h_{\theta}^{(m)}(x)$$



# Using Softmax

$$L(\theta) = - \sum_{i=1}^N y_a * \log(y_p)$$

$$y_a = [cat, dog, fish] = [1, 0, 0]$$

$\Rightarrow$  there are  $k$  classes ( $k = 3$  in this example)

$$y_p \text{ for class } m = \text{softmax}(x\theta) = \frac{\exp(x\theta)_m}{\sum_{j=0}^k \exp(x\theta)_j}$$

$$y_p = [0.6, 0.3, 0.1]$$

$$SGD \Rightarrow \theta^{t+1} \leftarrow \theta^t - \alpha \nabla L(\theta)$$

$$\theta^{t+1} \leftarrow \theta^t - \alpha x^T (y_p - y_a)$$

# Take-Home Messages

- Generative and Discriminative Classification
- The Logistic Regression Model
- Understanding the Objective Function
- Gradient Descent for Parameter Learning
- Multiclass Logistic Regression