Applied Text Analytics & Natural Language Processing

with Dr. Mahdi Roozbahani
& Wafa Louhichi

Logistic Regression – part 1

Learning Objectives

In this lesson, you will learn another linear text classifier

- Discriminative model
- Probability output
- Sigmoid function

Generative vs Discriminative

- 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 vs Discriminative

- 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

Let's Start with the Math Concept Again: Bayes Decision Rule

Posterior

Likelihood

Prior

$$\underbrace{P(y|x)}_{\text{Posterior}} = \frac{P(x|y)P(y)}{P(x)} = \frac{P(x,y)}{\sum_y P(x,y)}$$

Normalization Constant

Generative: We need to calculate likelihood and prior explicitly

Discriminative: Can we calculate Posterior directly without using Bayes equation?

Logistic Function for Posterior Probability

Let's use the following function:

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

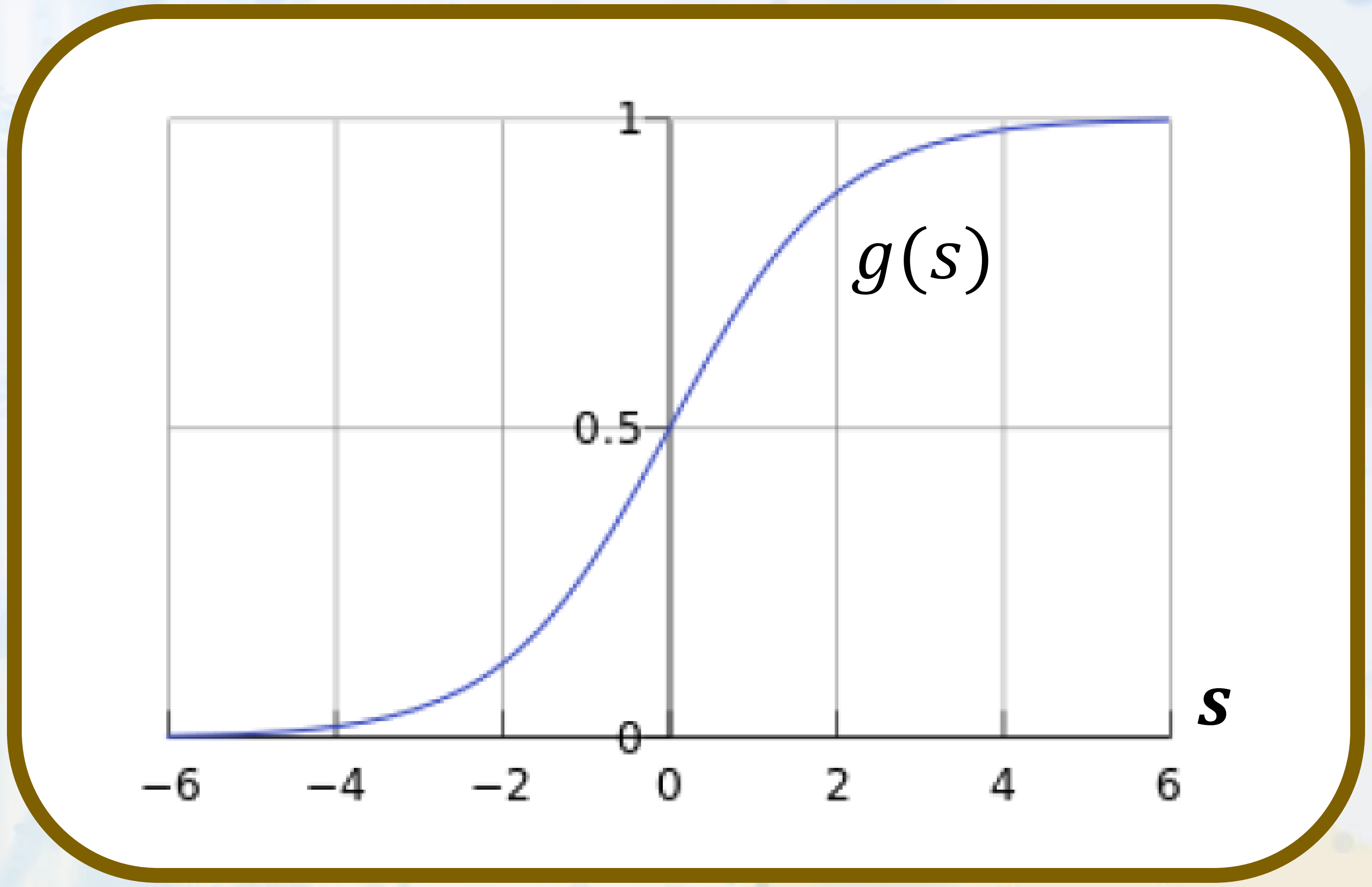
$$s = x\theta$$

This formula is called sigmoid function

It is easier to use this function for optimization

Is 0.5 threshold cut-off a good choice?

Many equations can give us this shape



Logistic Function for Posterior Probability

Let's use the following function:

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

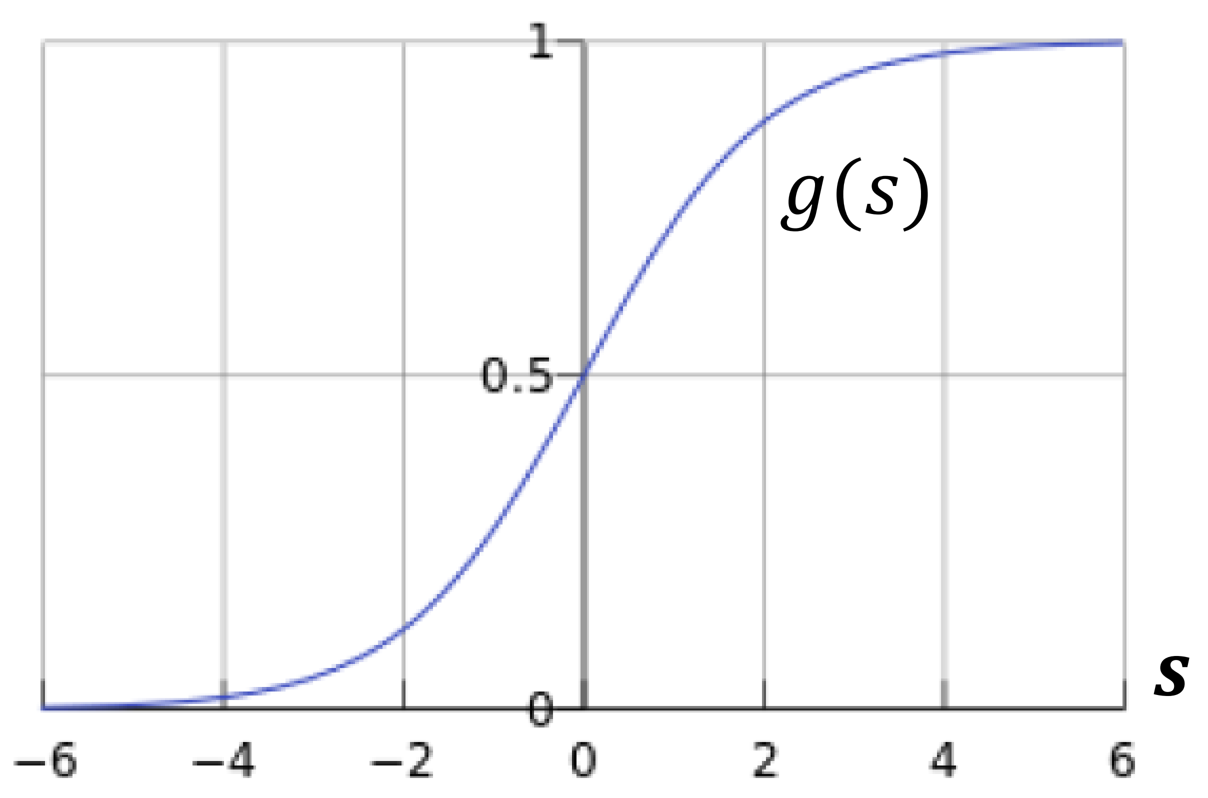
$$s = x\theta$$

This formula is called sigmoid function

It is easier to use this function for optimization

Is 0.5 threshold cut-off a good choice?

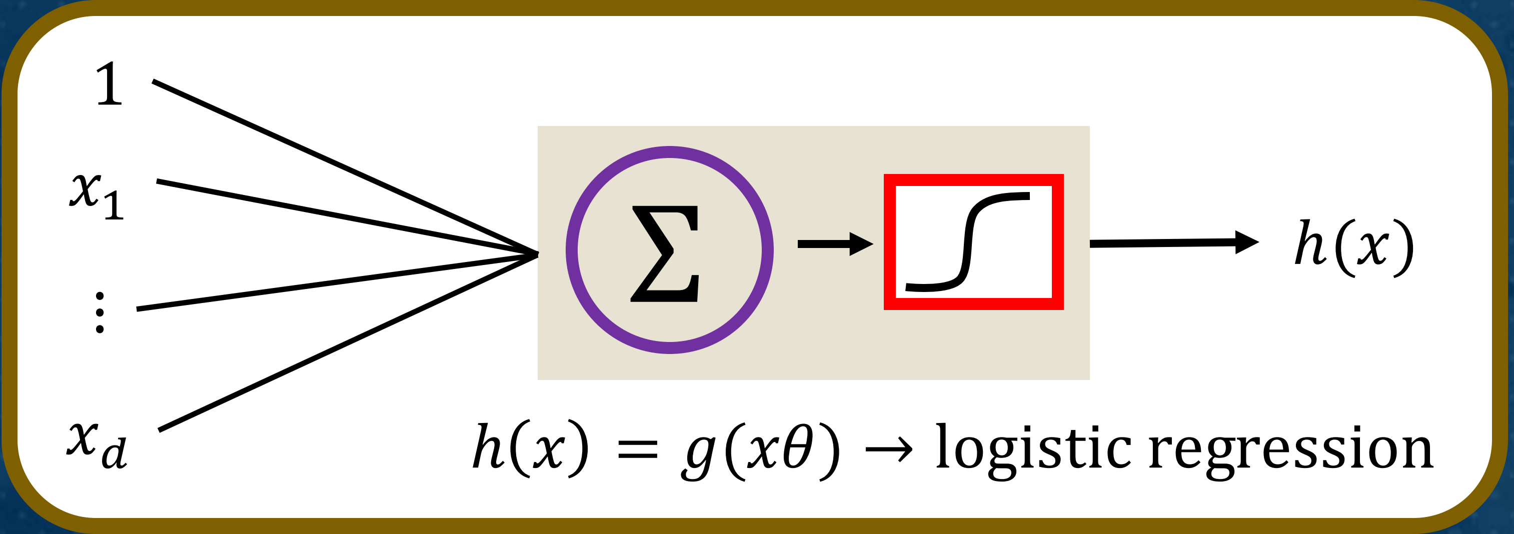
Many equations can give us this shape



Sigmoid Function

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

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



Soft classification
Posterior probability

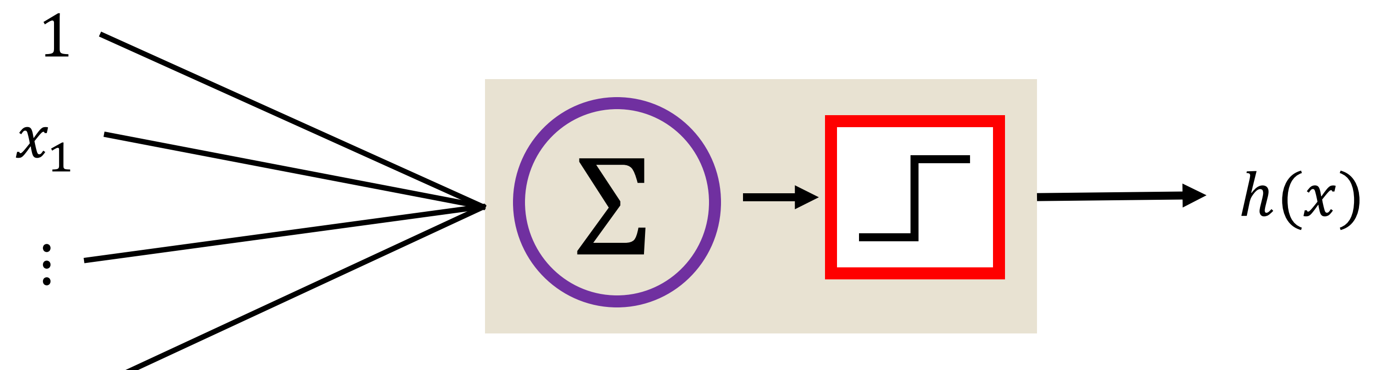
Applied Text Analytics & Natural Language Processing

with Dr. Mahdi Roozbahani
& Wafa Louhichi

Logistic Regression – part 2

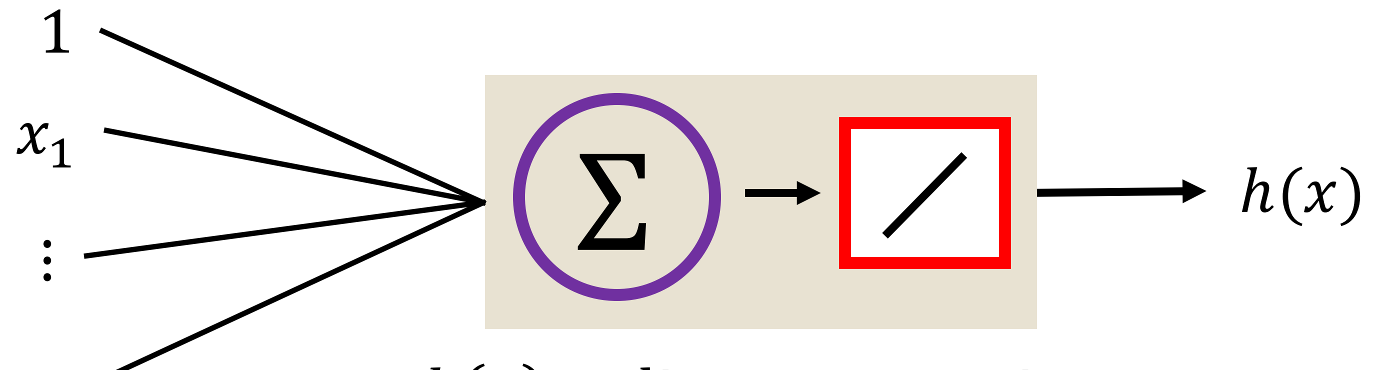
Three Linear Models

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

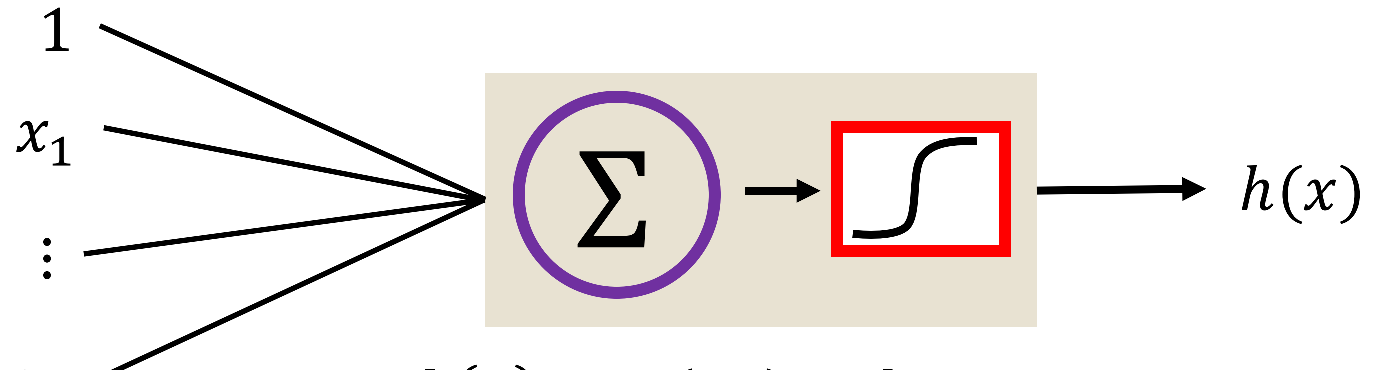


$h(x) = sign(x\theta) \rightarrow$ linear classification (Perceptron)

Hard classification



$h(x) \rightarrow$ linear regression



$h(x) = g(x\theta) \rightarrow$ logistic regression

Soft classification
Posterior probability

$Sigmoid$ is Interpreted as Probability

Example: Prediction of whether a customer likes a product based on the customer written feedback

Input $\boldsymbol{x}$: a BoW or TF-IDF of a document that contains a customer's feedback

$g(s)$: probability of whether a customer likes the product or not

We can't have a hard prediction here

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

$$h_\theta(x) = p(y|x) = \begin{cases} g(s), & y = 1 \\ 1 - g(s), & y = 0 \end{cases}$$

Using posterior probability directly

$Sigmoid$ is Interpreted as Probability

Example: Prediction of whether a customer likes a product based on the customer written feedback

Input $\boldsymbol{x}$: a BoW or TF-IDF of a document that contains a customer's feedback

$g(s)$: probability of whether a customer likes the product or not

We can't have a hard prediction here

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

$$h_\theta(x) = p(y|x) = \begin{cases} g(s), & y = 1 \\ 1 - g(s), & y = 0 \end{cases}$$

Using posterior probability directly

Logistic Regression Model

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

We need to find θ parameters, let's set up log-likelihood for **n** datapoints

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

$$= \sum_i \theta^T x_i^T (y_i - 1) - \log(1 + \exp(-x_i\theta))$$

The Gradient of $l(\theta)$

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

$$= \sum_i \theta^T x_i^T (y_i - 1) - \log(1 + \exp(-x_i \theta))$$

- Gradient

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

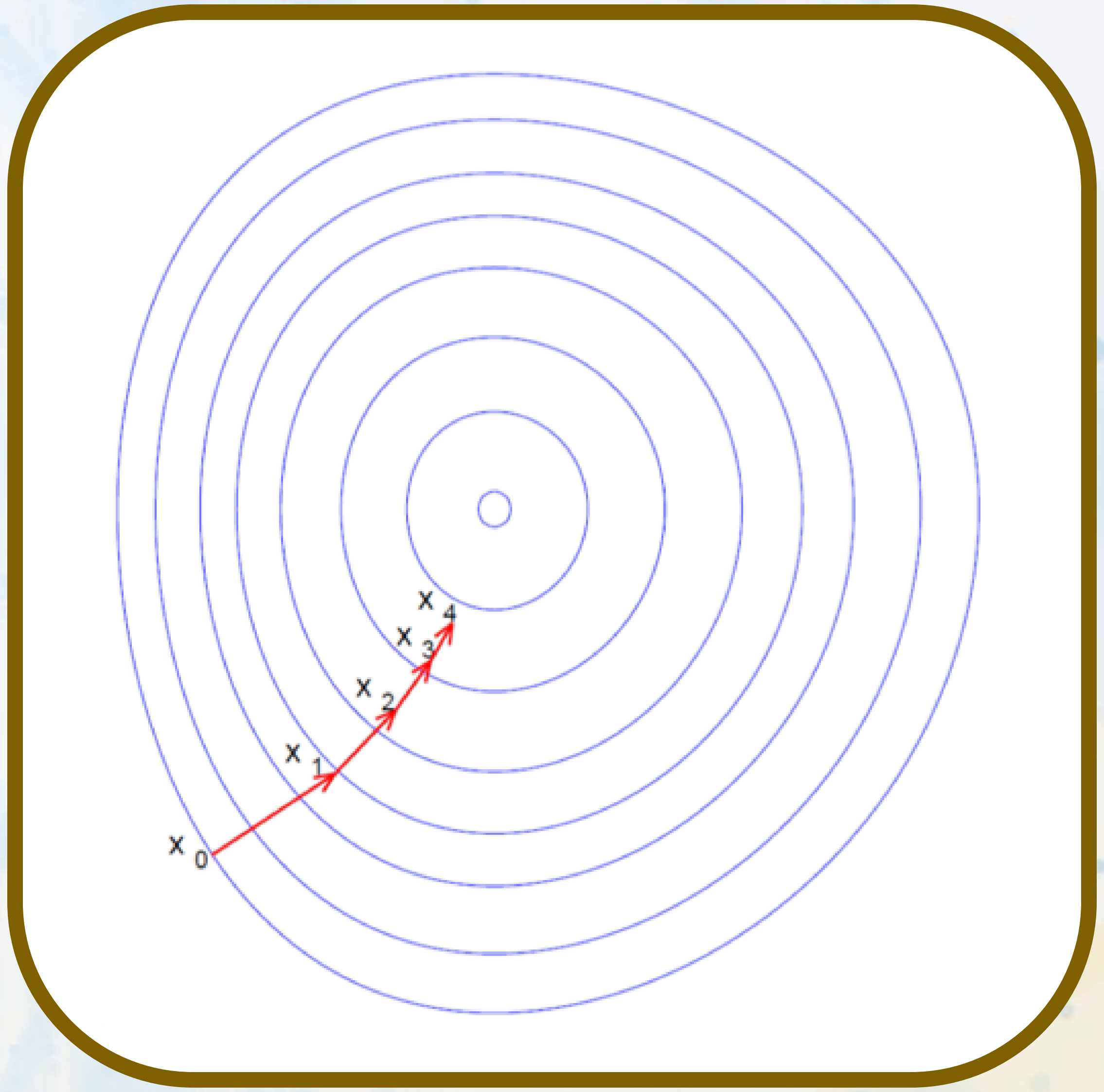
- Setting it to 0 does not lead to closed form solution

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

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

η_k is called the step size or learning rate



Gradient Ascent (concave) / Descent (convex) Algorithm

- Initialize parameter θ^0
- Do

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

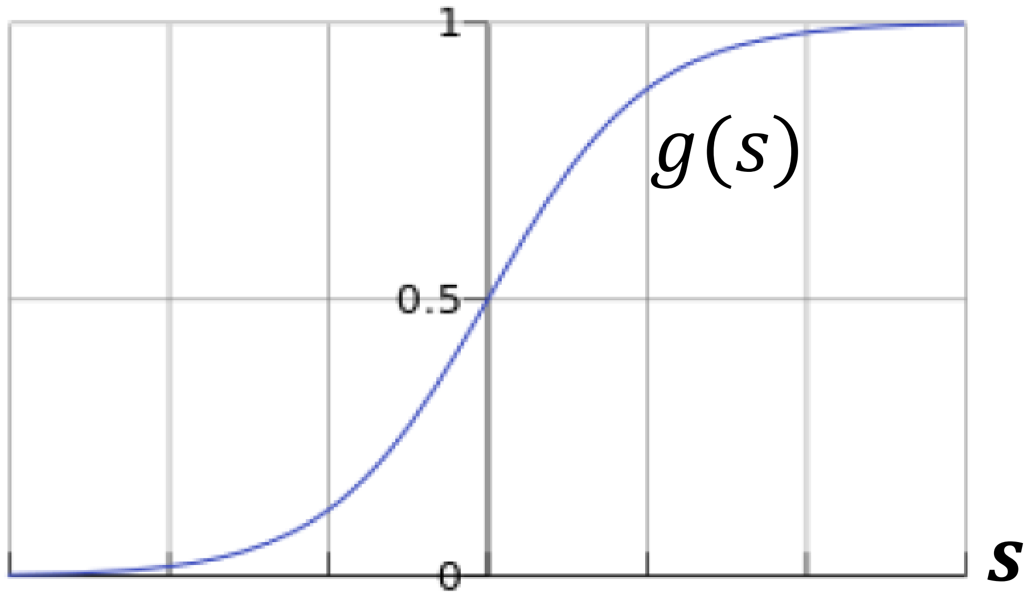
- While the $||\theta^{t+1} - \theta^t|| > \in$

Logistic Regression

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

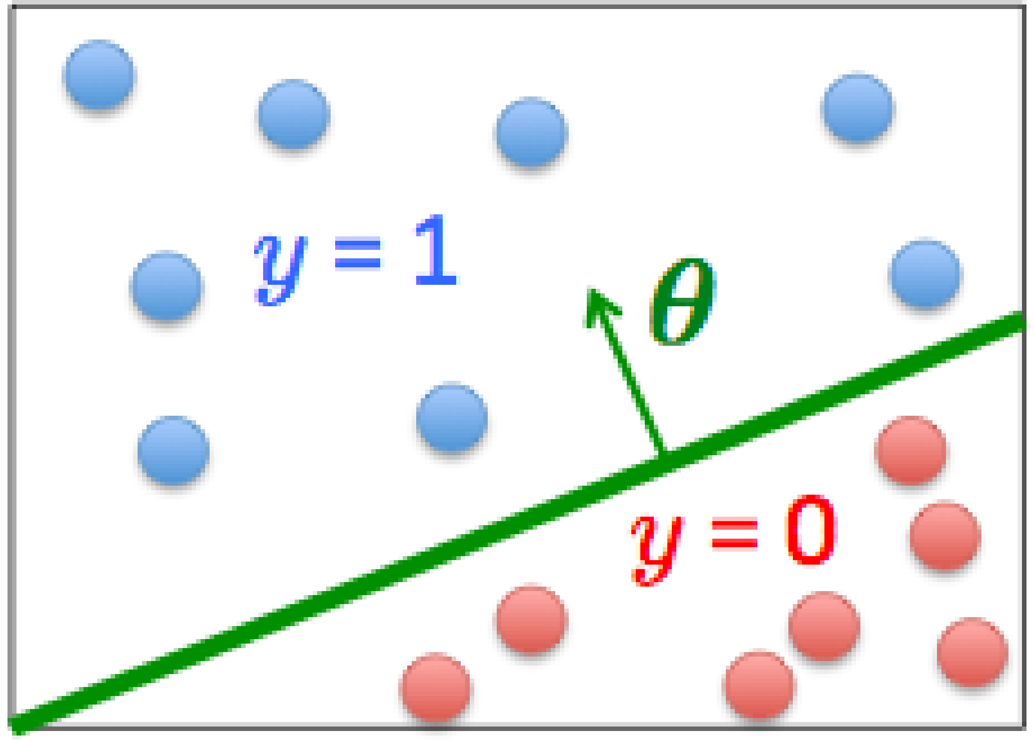
$$s = x\theta$$

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



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

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



Advantages and Disadvantages of Logistic Regression

- Advantages:
  - Simple algorithm
  - Does not need to model prior or likelihood
  - It provides a probability output
- Disadvantages:
  - We have the discriminative model assumption
  - Model needs to be optimized using a numerical approach

Summary

- We learned about discriminative model
- We know how logistic regression works and how we calculate posterior probability directly