Georgia Tech

Machine Learning CS 4641-7641

Naïve Bayes and Logistic Regression

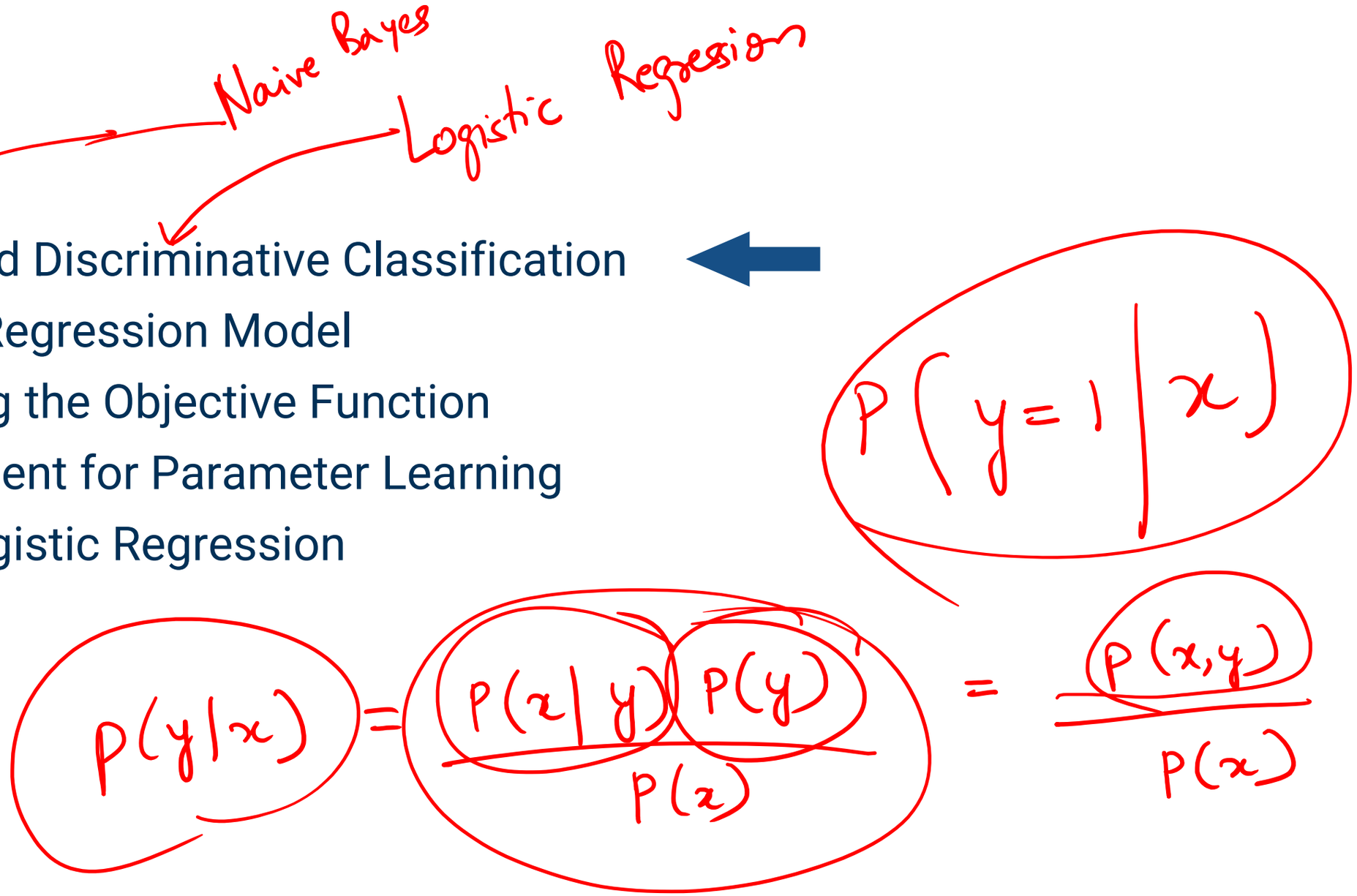
Nimisha Roy
Georgia Tech

These slides are inspired based on slides from Mahdi Roozbahani, Le Song, Eric Eaton, and Chao Zhang.

Georgia Tech

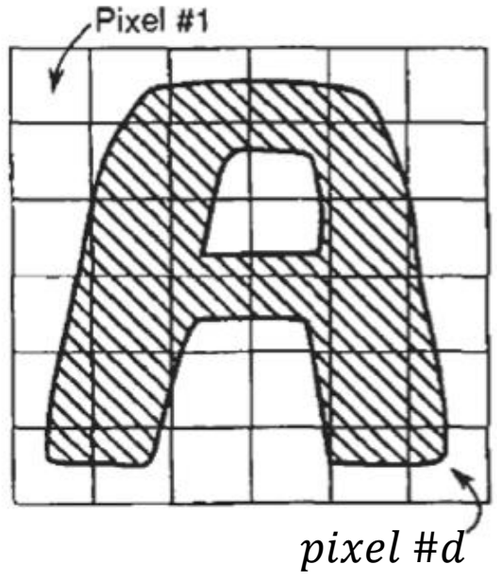
Outline

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



Classification

- Represent the data



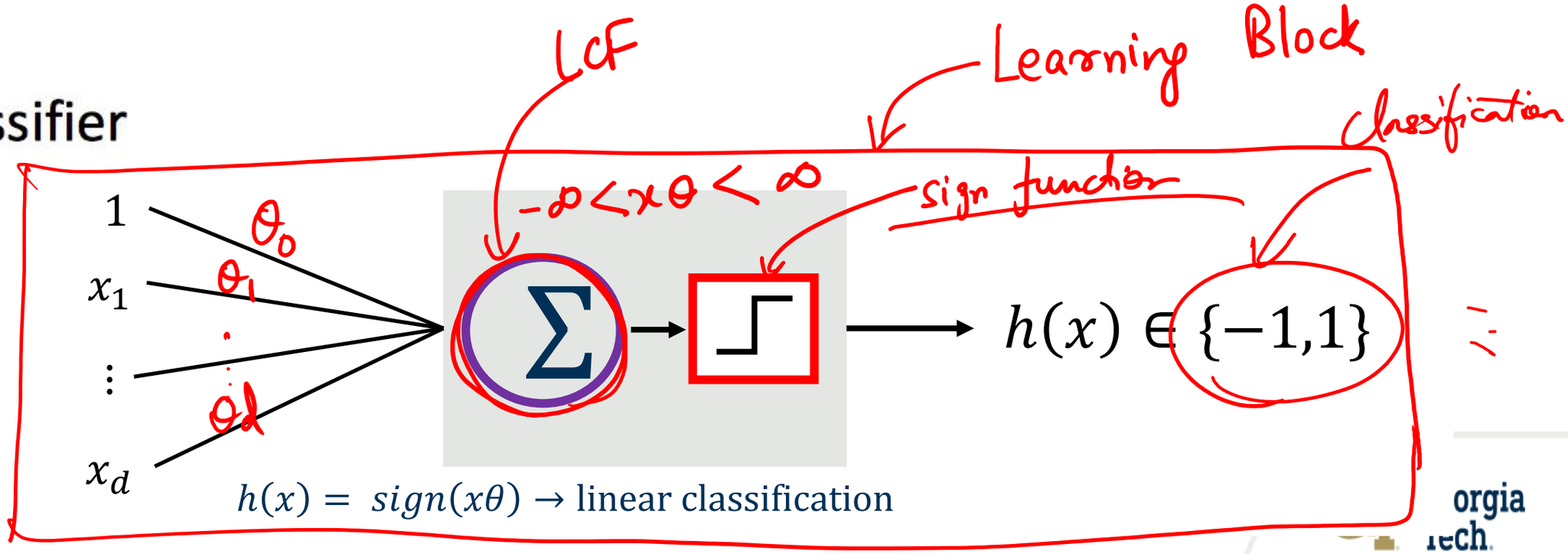
$$\Rightarrow \quad X = \begin{bmatrix} x_1 \\ x_2 \\ \cdot \\ \cdot \\ \cdot \\ x_d \end{bmatrix}$$

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

A

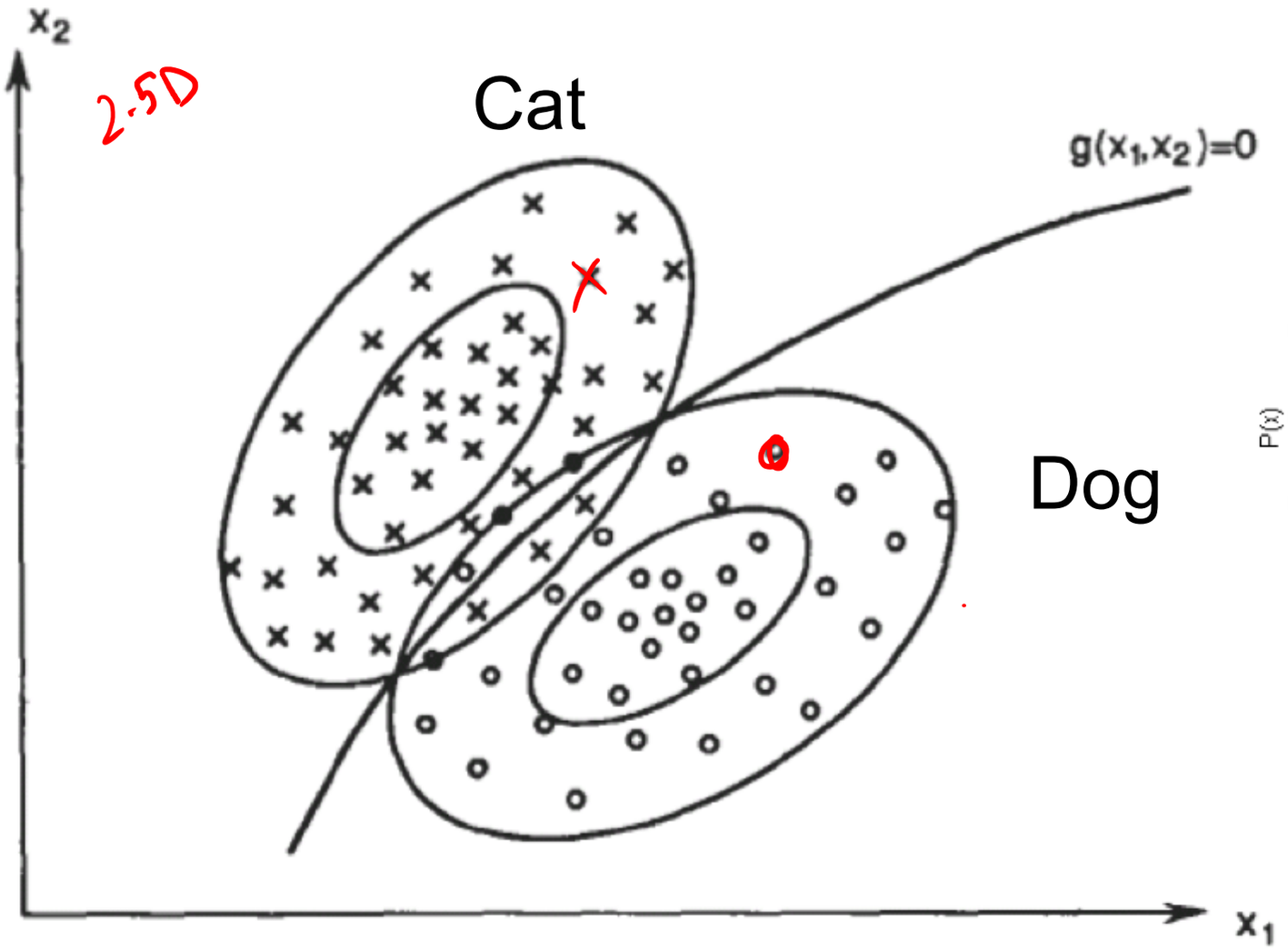
not A

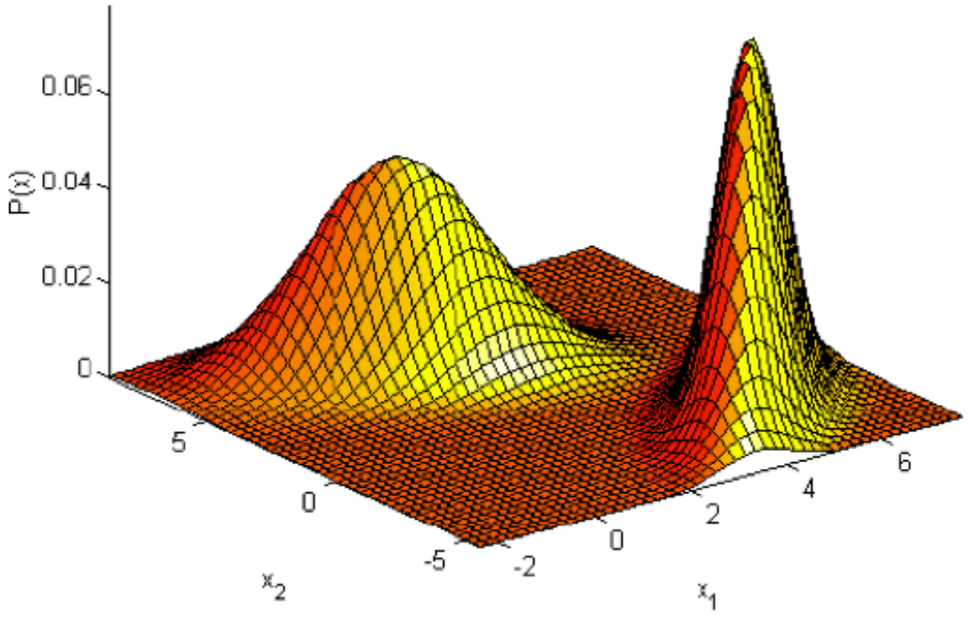
- Classifier



Decision Making: Dividing the Feature Space

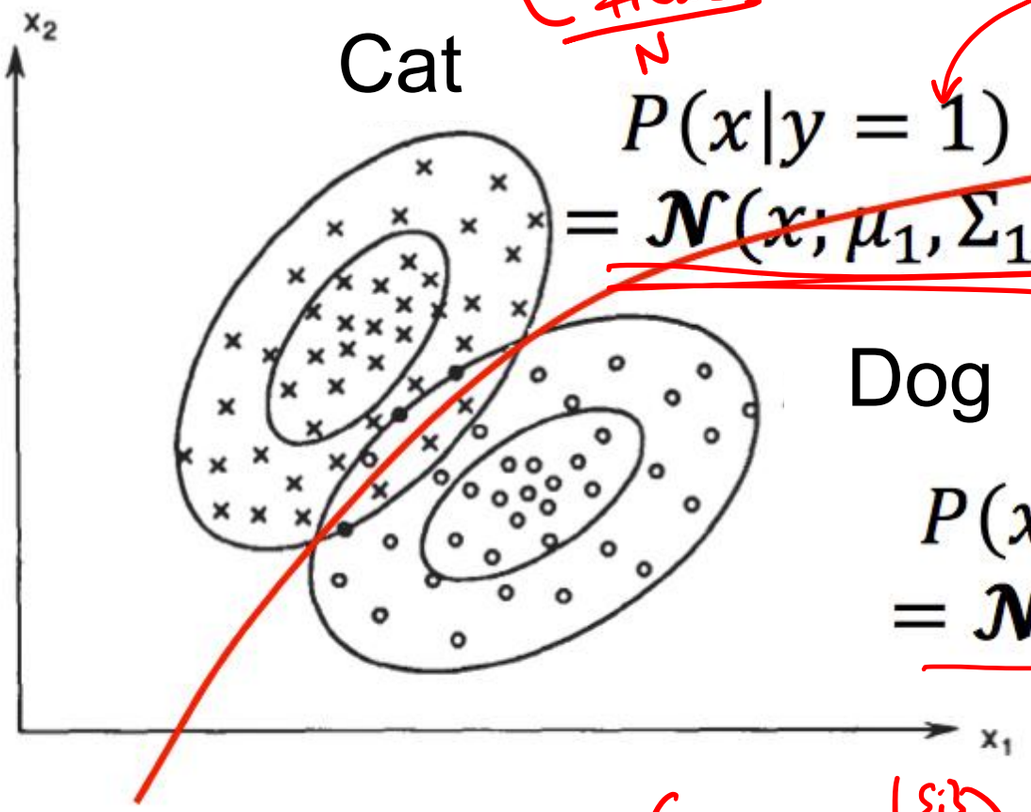
- Distributions of sample from normal (positive class) and abnormal (negative class) tissues





How to Determine the Decision Boundary?

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



$$P(x|y=1) = \mathcal{N}(x; \mu_1, \Sigma_1) \quad ?$$

$$P(x|y=-1) = \mathcal{N}(x; \mu_2, \Sigma_2)$$

$P(y=1|x^{(i)})$ & $P(y=-1|x^{(i)})$

Bayes Decision Rule

$p(y=1|x)$ vs. $p(y=-1|x)$

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

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

$$P(y=\pm 1 \mid x^{\{i\}}) = \frac{P(x^{\{i\}} \mid y=\pm 1)\cdot P(y=\pm 1)}{P(x^{\{i\}})}$$

$$= \frac{P(x_1^{\{i\}}, x_2^{\{i\}}, x_3^{\{i\}} \dots x_d^{\{i\}} \mid y=\pm 1)\cdot P(y=\pm 1)}{P(x^{\{i\}})}$$

$$= \frac{\frac{1}{2\pi|\Sigma|^{1/2}} \exp\left(-\frac{1}{2}(x-\mu)\Sigma^{-1}(x-\mu)^T\right)\cdot P(y=\pm 1)}{P(x^{\{i\}})}$$

Naive Bayes → Assume Conditional independence between features

univariate gaussian

$$= \frac{P(x_1^{\{i\}} \mid y=\pm 1)\cdot P(x_2^{\{i\}} \mid y=\pm 1) \dots P(x_d^{\{i\}} \mid y=\pm 1)\cdot P(y=\pm 1)}{P(x^{\{i\}})}$$

We move from computing Σ^{-1} which is very computationally expensive to computing product of univariate gaussian

Bayes Decision Rule

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

- The poster probability of a test point

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

- 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 $\mathrm{h(x)} = -\ln \frac{q_i(x)}{q_j(x)}$

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 conditionally 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:

$$P(x, y_{label=1}) = \mathrm{P}(\mathrm{x}_1, \ldots, \mathrm{x}_\mathrm{d}, \mathrm{y}_{\mathrm{label}=1}) = \mathrm{P}(\mathrm{x}_1|\mathrm{x}_2, , \ldots, \mathrm{x}_\mathrm{d}, \mathrm{y}_{\mathrm{label}=1})P(\mathrm{x}_2, \ldots, \mathrm{x}_\mathrm{d}, \mathrm{y}_{\mathrm{label}=1})$$

$$= \mathrm{P}(\mathrm{x}_1|\mathrm{x}_2, , \ldots, \mathrm{x}_\mathrm{d}, \mathrm{y}_{\mathrm{label}=1})P(\mathrm{x}_2|\mathrm{x}_3 \ldots, \mathrm{x}_\mathrm{d}, \mathrm{y}_{\mathrm{label}=1})P(\mathrm{x}_3, \ldots, \mathrm{x}_\mathrm{d}, \mathrm{y}_{\mathrm{label}=1})$$

$$= \ldots$$

$$= \mathrm{P}(\mathrm{x}_1|\mathrm{x}_2, , \ldots, \mathrm{x}_\mathrm{d}, \mathrm{y}_{\mathrm{label}=1})P(\mathrm{x}_2|\mathrm{x}_3 \ldots, \mathrm{x}_\mathrm{d}, \mathrm{y}_{\mathrm{label}=1}) \ldots$$
$$P(\mathrm{x}_{\mathrm{d}-1}|x_d, y_{label=1})P(x_d|y_{label=1})\mathrm{P}(y_{label=1})$$

Naïve Bayes assumption: let's rewrite it as:

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

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

→ Gaussian naïve Bayes
A typical assumption

Georgia Tech

“Naïve” conditional independence assumption

Real World Example

What do People do in Practice?

$p(x,y)$

- 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

Discriminative Models

- Directly estimate decision boundary $\mathrm{h}(\mathrm{x}) = -\ln \frac{q_i(x)}{q_j(x)}$ 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

Recap

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

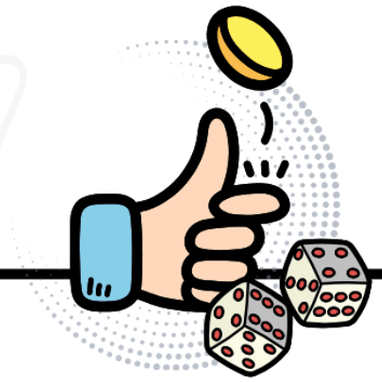
$P(y|x)$ → learn this directly

- Generative vs Discriminative
- Generative Example? → Naive Bayes
- Assumptions to calculate posterior → all features are conditionally independent

$$P(x_1, x_2 \ldots x_d \mid y) \longleftarrow \text{Likelihood}$$

$$(\varepsilon^{-1}) \quad \| \quad P(x_1|y) \cdot P(x_2|y) \cdots P(x_d|y)$$

Gaussian/Multinomial/Bernoulli Naïve Bayes



Bernoulli Naive Bayes

For binary or boolean features.

EXAMPLE

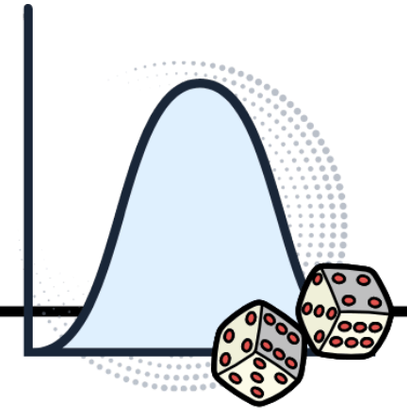
HAS 4 LEGS?	CAN FLY?	LAY EGGS?
1	0	1
0	1	1
0	1	1
0	1	0
0	0	1

Multinomial Naive Bayes

For discrete features (like word count)

EXAMPLE

WORD the	WORD but	WORD is
4	1	2
3	0	0
3	0	0
1	0	4
0	2	3



Gaussian Naive Bayes

For continuous, real-valued attributes.

EXAMPLE

AGE	WEIGHT	HEIGHT
17.4	56.5	145.2
25.4	71.2	170.4
18.8	70.3	164.5
21.1	51.2	140.5
20.9	81.5	182.2

https://towardsdatascience.com/bernoulli-naive-bayes-explained-a-visual-guide-with-code-examples-for-beginners-aec39771ddd6/

Georgia Tech

Outline

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

2 classes

Gaussian Naïve Bayes

fraction of datapoints with $y=1$

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

$$\prod_{i=1}^{d} p(x_i|y=_d 1, \mu_{1i}, \sigma_{1i})$$

$$= \prod_{i=1} \frac{1}{\sqrt{2\pi}\sigma_{1i}} \exp\left(-\frac{1}{2\sigma_{1i}^2}(x_{1i}-\mu_{1i})^2\right)$$

Prior: $p(y=1) = \pi_1 = \frac{\#\text{ in class 1}}{\text{All data points.}}$

Georgia Tech

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

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

labels

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

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

$$P(y=1|x) = \frac{\exp(a)}{\exp(a) + \exp(b)}$$

$$= \frac{1}{1 + \frac{\exp(b)}{\exp(a)}} =$$

2 class

$$\frac{1}{1 + \exp(b-a)}$$

Assumption: All classes have same variance

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

$\exp(b-a)$ includes: $(x_i - \mu_{2i})^2 - (x_i - \mu_{1i})^2$

This will lead to no x_i^2 term

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

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

$$\frac{1}{1+\exp\left(\theta_0+\sum_i\theta_i x_i\right)} \qquad \sum_i \theta_i x_i$$

sigmoid function

θ_0

$$P(y = 1 \mid x) = \frac{1}{1 + \exp\left(-(\theta_0 + \sum_i \theta_i x_i)\right)}$$

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

LCF= $\theta_0 + \sum_i \theta_i x_i$

After all that Gaussian algebra, We started with a generative model…
but we ended up with this.

This is exactly the same form as **logistic regression.**

Georgia Tech

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

Pros:

1. less parameters
2. Make assumptions. Dist. of each class
3. Compute likelihood

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 = $d+1 \rightarrow \theta_0, \theta_1, \theta_2, \ldots, \theta_d$

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

Gaussian Naïve Bayes is a special case/subset of logistic regression

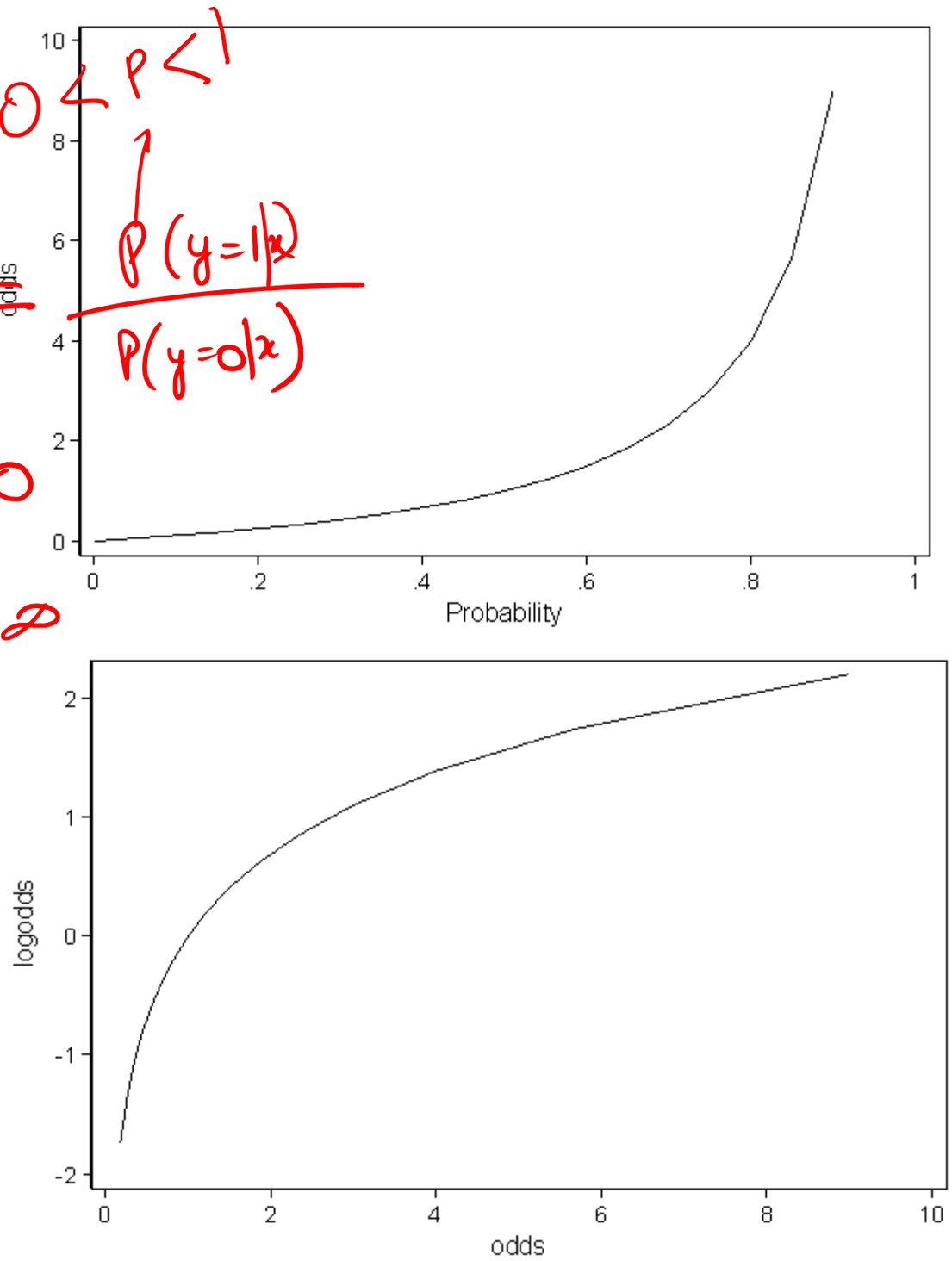
Georgia Tech

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

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

$0 < odds < \infty$

$-\infty < \log(odds) < \infty$



$log(odds)\ vs\ odds$

What could be $x\theta$ domain?

Georgia Tech

What is logit function?

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

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

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

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

Sigmoid function - maps any real-valued $x\theta$ to a probability between 0 and 1.

Georgia Tech

Interpretation: Generative and Discriminative Thinking

Naive Bayes (Generative):

Models $P(x \mid y)$

↓

With Gaussian + equal variance:

$$\log \frac{P(y=1 \mid x)}{P(y=-1 \mid x)} = \theta_0 + \sum_i \theta_i x_i$$

↓

Logistic Regression (Discriminative):

$$P(y=1 \mid x) = \frac{1}{1 + e^{-\left(\theta_0 + \sum_i \theta_i x_i\right)}}$$

Georgia Tech

Interpretation

$P(y=1|x) = 0.5 = P(y=-1|x)$

$\frac{1}{1+\exp(-s)} = 0.5 \Rightarrow 1+\exp(-s) = 2$

$\Rightarrow \exp(-s) = 1$

$\Rightarrow -s = 0$

$\Rightarrow s = 0 \Rightarrow x\theta = 0$

Linear line

Concept	Gaussian Naïve Bayes	Logistic Regression
Type	Generative	Discriminative
Learns	P(x,y) → P(y|x) and P(x)	P(y|x)
Distribution Assumption	Gaussian features	No distribution assumption
Decision boundary	Linear (after log transformation and equal variance assumption)	Linear
Parameters	Means, variances, and priors ((2d+1))	Coefficients and bias ((d+1))

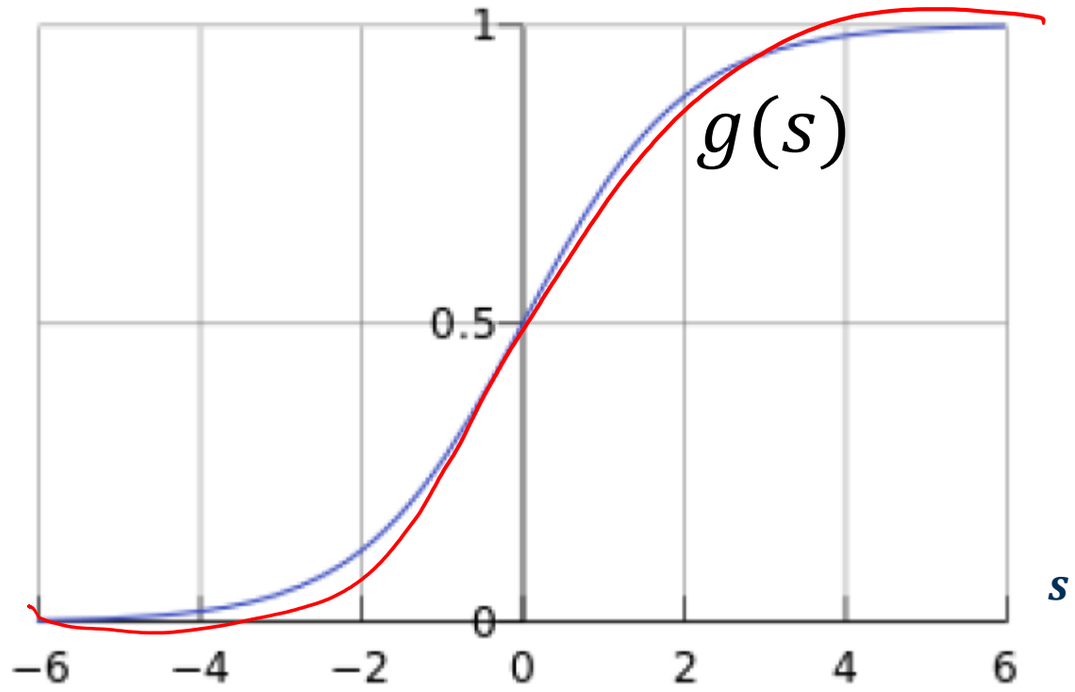
Logistic function for posterior probability

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)

Georgia Tech

Why Soft classification matters

Flw OR B/w feature Selection

Example: Prediction of heart attacks

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

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

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

p=0.3

We can't have a hard prediction here

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

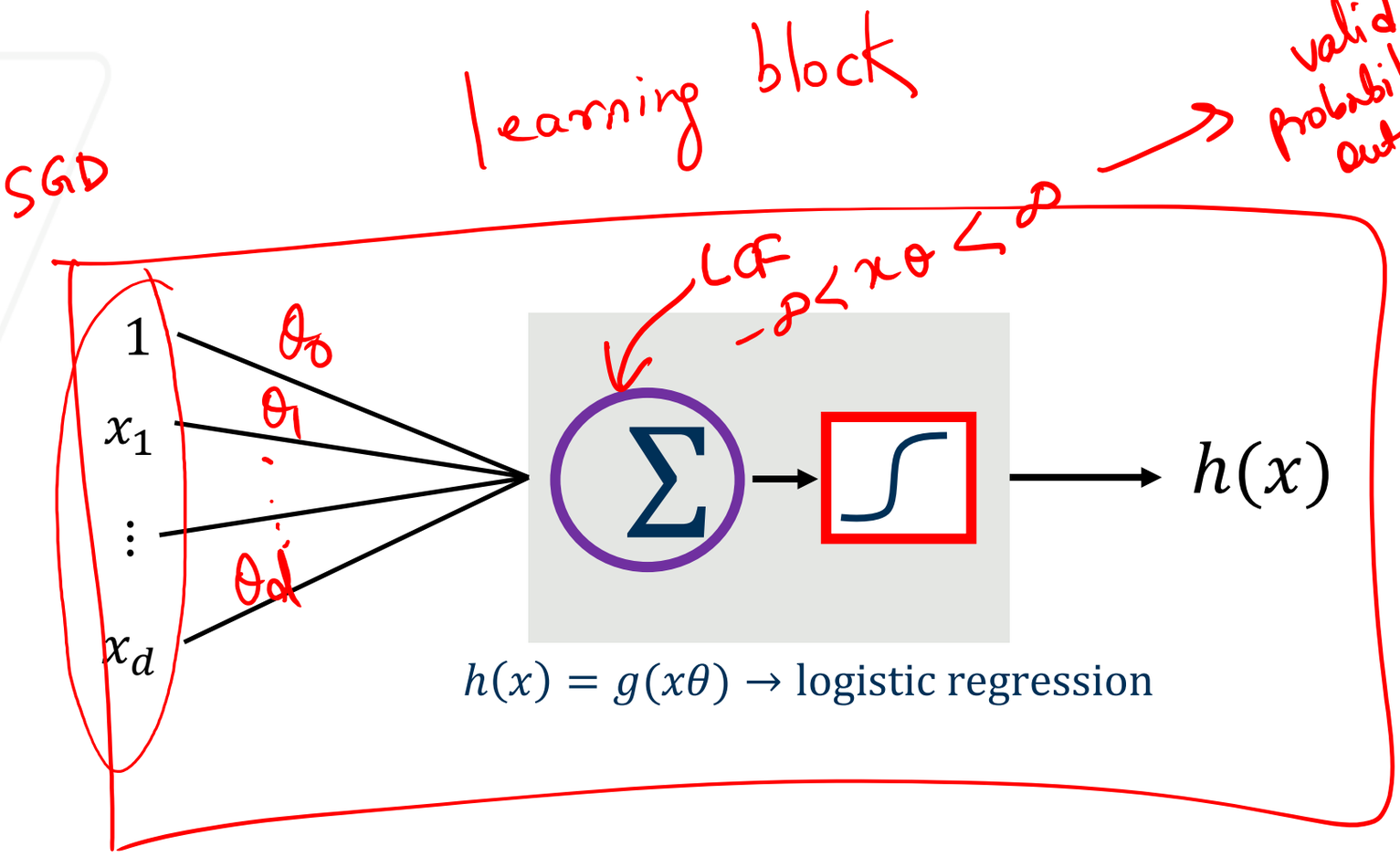
Using posterior probability directly

Georgia Tech

Sigmoid Function

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

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

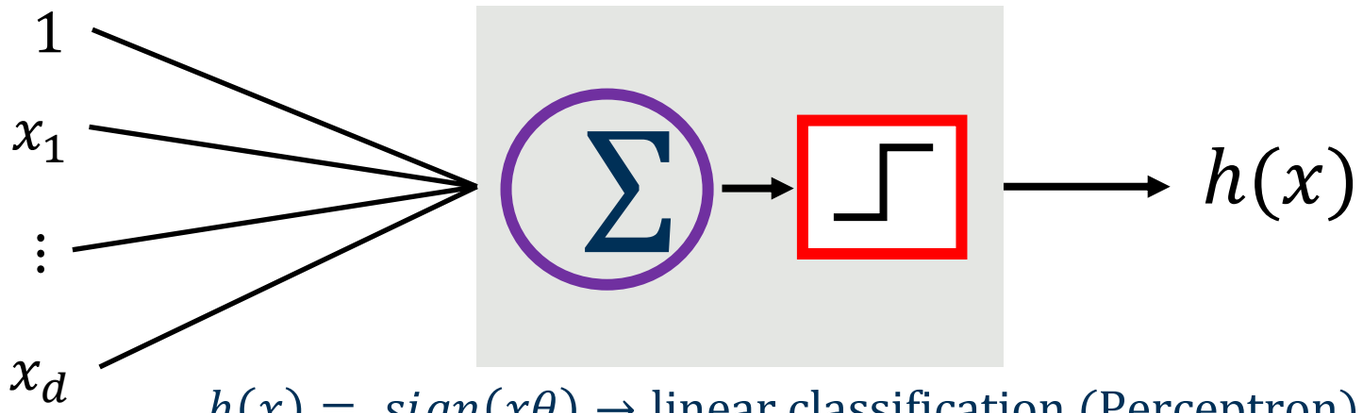


Soft classification
Posterior probability

Georgia Tech

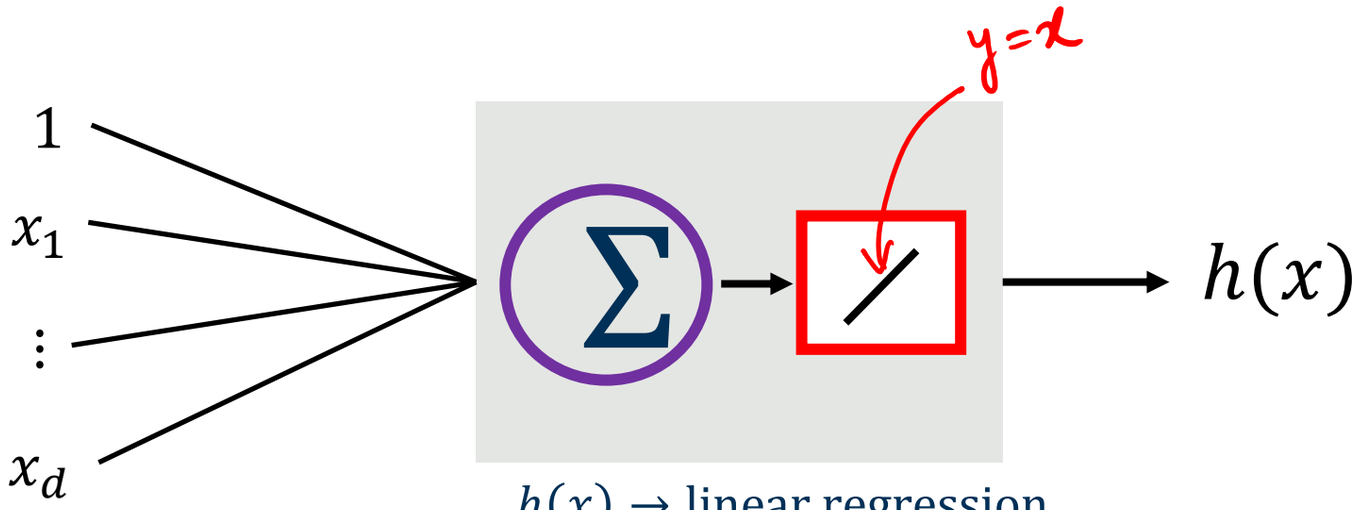
Three linear models

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

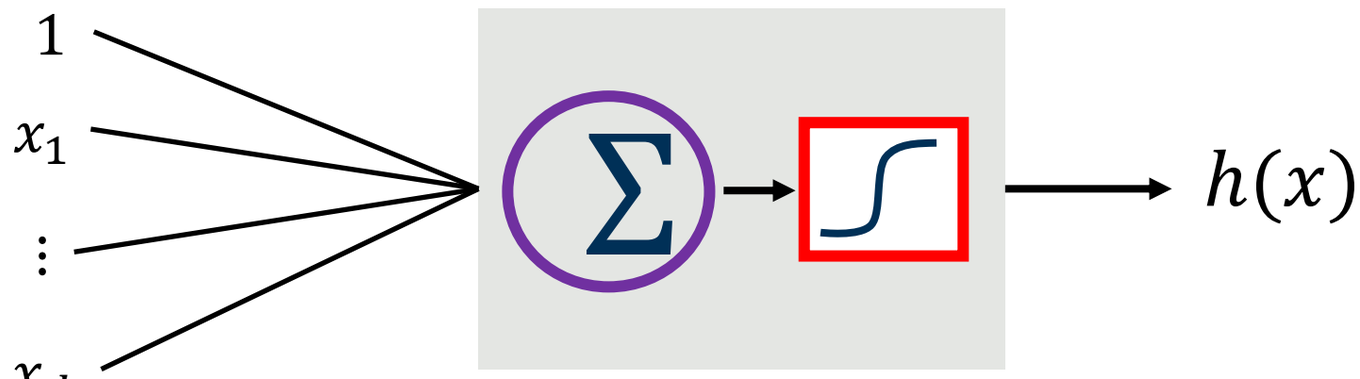


Hard classification

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



$h(x) \rightarrow$ linear regression



Soft classification
Posterior probability

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

Georgia Tech

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, what should we use as the objective function?

MLE

We are maximizing the probability of the *observed labels given the inputs*.
We choose θ so that the model assigns high probability to the correct labels.

Georgia Tech

Logistic regression model

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

$$p(y|x) = \left(\frac{1}{1+\exp(-x\theta)}\right)^{y} \left(\frac{\exp(-x\theta)}{1+\exp(-x\theta}\right)^{1-y}$$

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

i.i.d

$$l(\theta) := log \prod_{i=1}^{n} p(y^{\{i\}}|x^{\{i\}}, \theta)$$

$$= \sum_{i} \theta^T (x^{\{i\}})^T \left(y^{\{i\}} - 1\right) - \log(1 + \exp(-x^{\{i\}}\theta))$$

This form is concave, negative of this form is convex

Georgia Tech

$$\log \prod_i p\left(y^{\{i\}} \mid x^{\{i\}}, \theta\right)$$

$$\log\left(\left(\frac{1}{1+\exp(-x\theta)}\right)^{y}\left(\frac{\exp(-x\theta)}{1+\exp(-x\theta)}\right)^{1-y}\right)$$

$$y\left(\log 1 - \log(1+\exp(-x\theta)\right) + (1-y)\log\left(\exp(-x\theta)\right) - (1-y)\log(1+\exp(-x\theta))$$

$$-y\log(1+\exp(-x\theta)) + \log(\exp(-x\theta) - y(-x\theta) - \log(1+\exp(-x\theta)) + y\log(1+\exp(-x\theta))$$

$$-x\theta + xy\theta - \log(1+\exp(-x\theta))$$

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

$$\frac{\partial l(\theta)}{\partial \theta} = \sum_{i} (x^{\{i\}})^T \left(y^{\{i\}} - 1\right) + (x^{\{i\}})^T \frac{\exp\left(-x^{\{i\}}\theta\right)}{1 + \exp(-x^{\{i\}}\theta)}$$

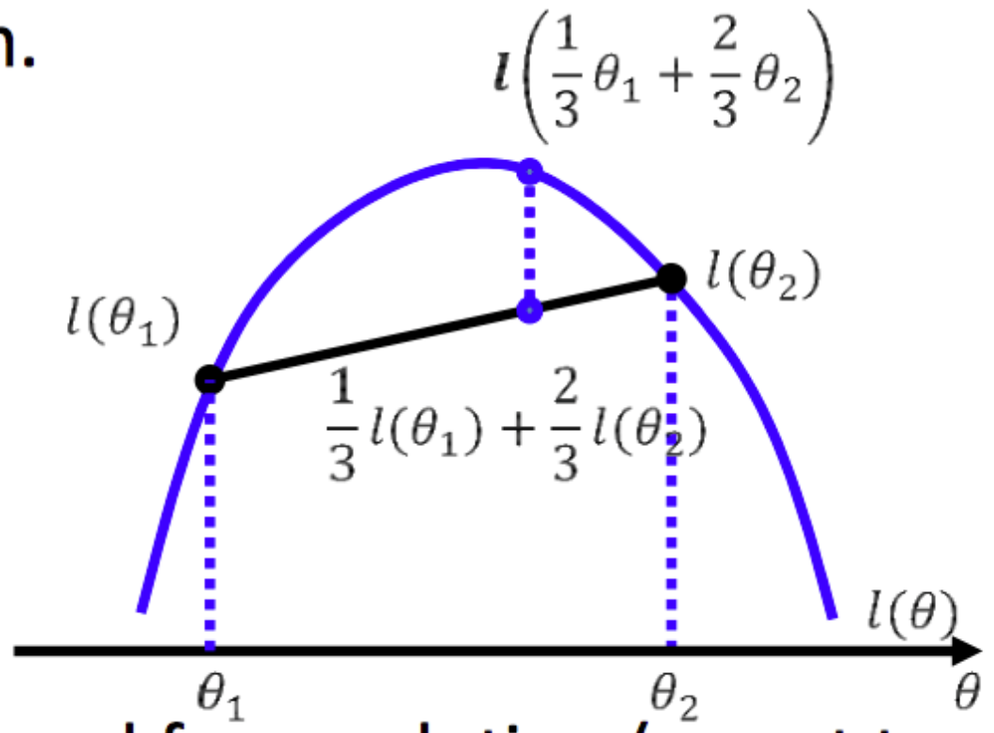
- Setting it to 0 does not lead to closed form solution

The Objective Function

- Find θ, 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 θ, 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

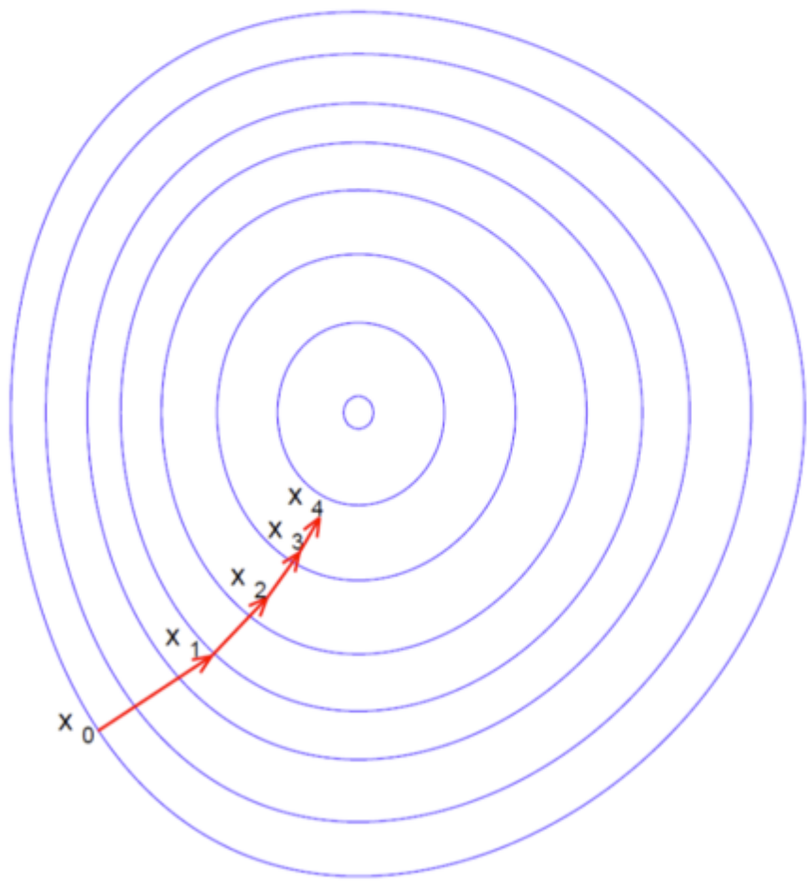
- Given an initial guess, we *iteratively* refine the guess by taking the direction of the negative gradient

- Think about going down a hill by taking the steepest direction at each step

- Update rule

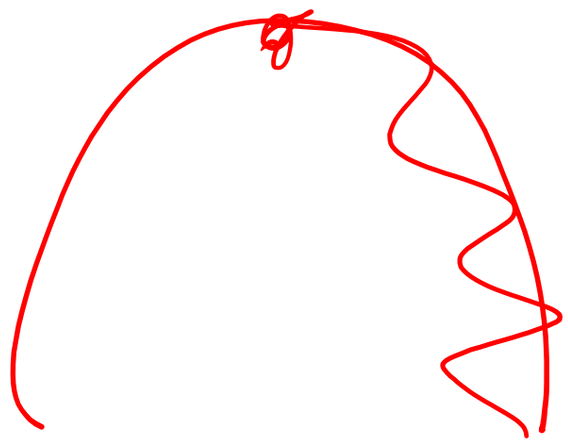
$$x_{k+1} = x_k - \gamma_k \nabla f(x_k)$$

γ_k is called the step size or learning rate



Tech

Gradient Ascent(concave)/Descent(convex) algorithm



- Initialize parameter θ^0
- Do

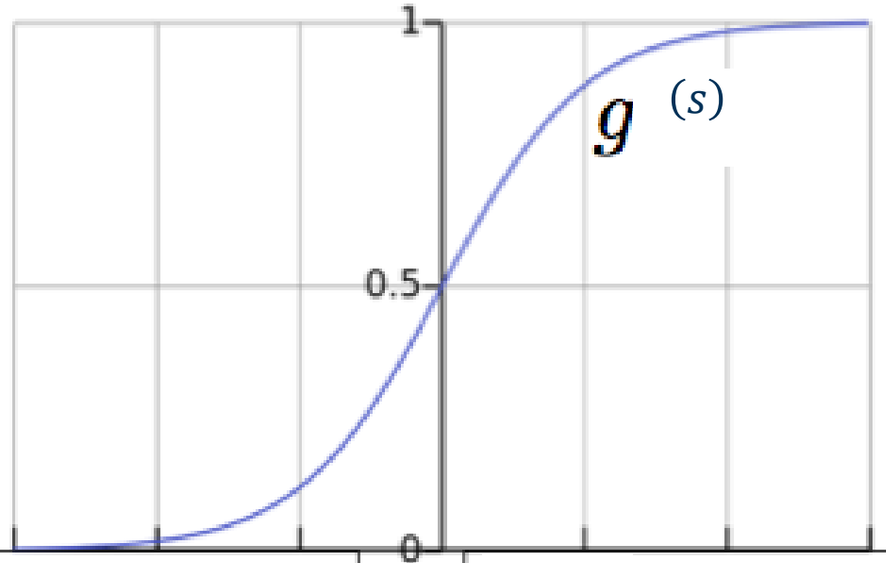
$$\theta^{t+1} \leftarrow \theta^t + \eta \sum_i (x^{\{i\}})^T \left(y^{\{i\}} - 1\right) + (x^{\{i\}})^T \frac{\exp\left(-x^{\{i\}}\theta\right)}{1 + \exp(-x^{\{i\}}\theta)}$$

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

Logistic Regression

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

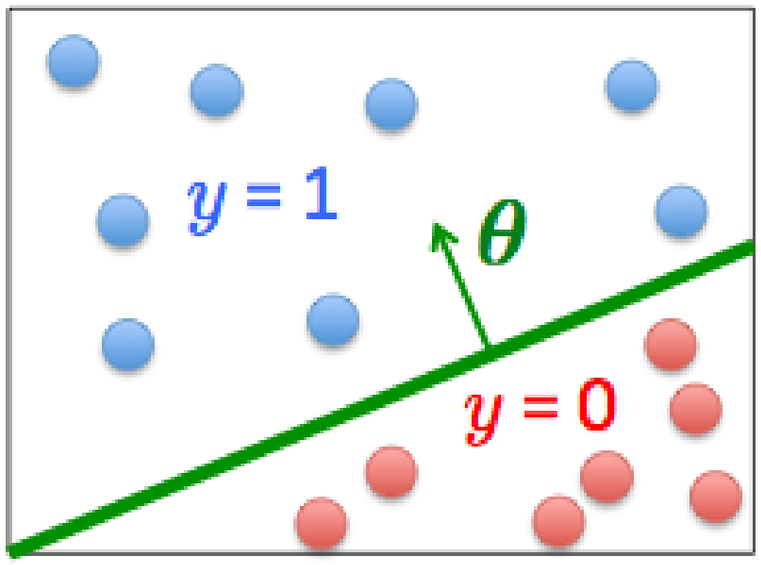
$$s = x\theta$$



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

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

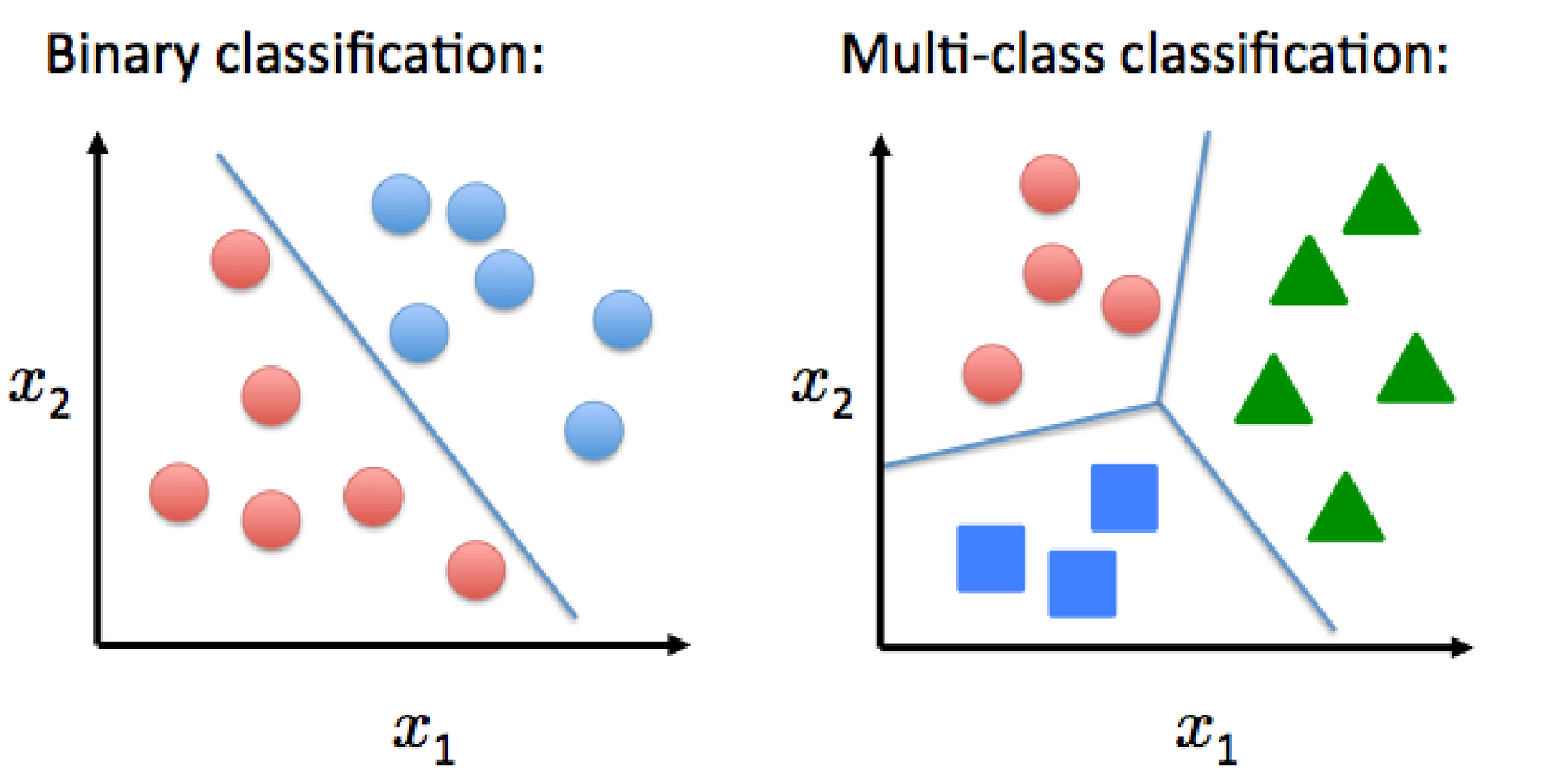
- Assume a threshold and...
  - Predict $y = 1$ if $g(s) \geq 0.5$
  - Predict $y = 0$ if $g(s) < 0.5$



Outline

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

Multiclass Logistic Regression

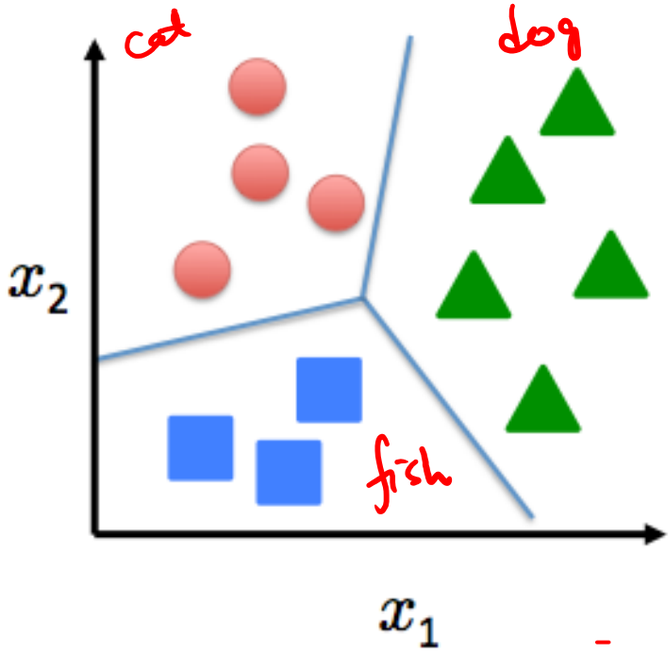


Disease diagnosis: healthy / cold / flu / pneumonia

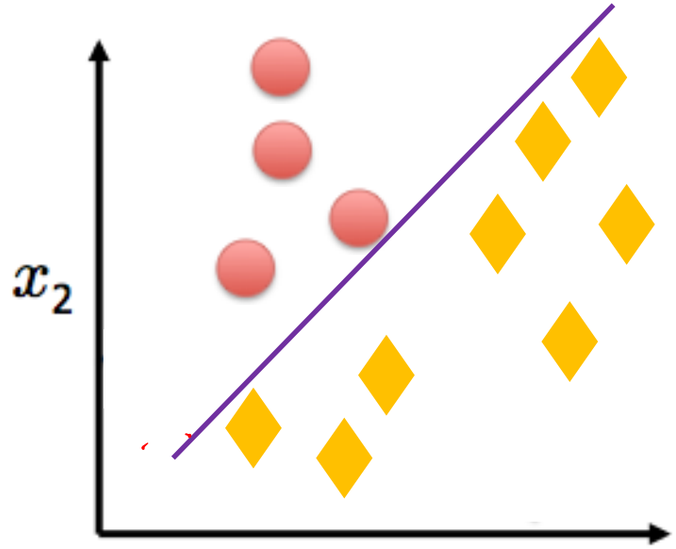
Object classification: desk / chair / monitor / bookcase

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

Multi-class classification:

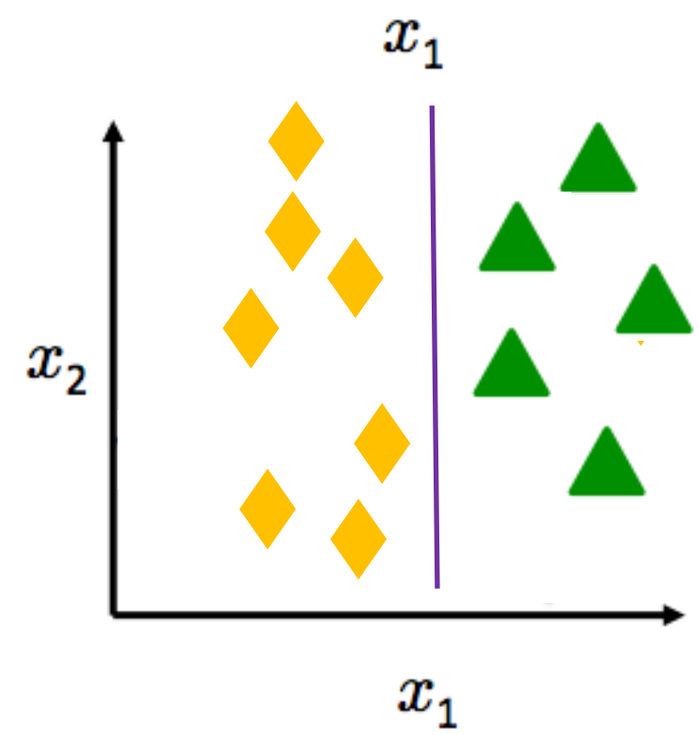


$h_\theta^{(m)}(x) = p(y = 1|x, \theta)\ (m = 1,2,3)$



$h_\theta^1(x)$

{ 0.7 cat 0.3 not cat }



$h_\theta^2(x)$

{ 0.6 dog 0.4 not dog }

$h_\theta^3(x)$

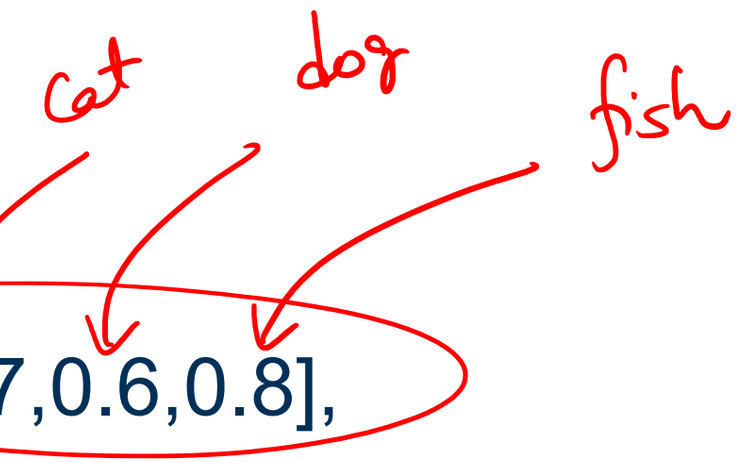
{ 0.8 fish 0.2 not fish }

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:

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



Key properties:

- Output is not a true distribution e.g. [0.7,0.6,0.8], because Models are independent
- Need to train m models

Approach 2: Using Cross Entropy + Softmax

Idea:

Train **one model**
All classes **compete together**

Output:

$$p_m = \frac{\exp\left((x\theta)_m\right)}{\sum_j \exp\left((x\theta)_j\right)}$$

Key properties:
Outputs:

$[0.6, 0.3, 0.1]$ (*sums to 1*)

True **probability distribution**

Georgia Tech

Approach 2: Using Cross Entropy + Softmax

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

Objective function - Make the probability of the correct class high

Sigmoid (binary case)

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

Outputs: One probability p. Other class is $1 - p$

Softmax (multiclass case)

$$p_m = \frac{\exp(s_m)}{\sum_j \exp\left(s_j\right)}$$

Outputs: One probability per class, All sum to 1

Georgia Tech

[cat, dog, fish] d=2 [ ]

$x^{(i)} = [2, 1]$

Using Cross Entropy + Softmax

Setup (Dimensions)

$x \in \mathbb{R}^d$ input features

$\theta \in \mathbb{R}^{d \times M}$ parameters

$s \in \mathbb{R}^M$ scores = $x.\theta$ for each class

$y_a \in \mathbb{R}^M$)one-hot label

$y_p \in \mathbb{R}^M$)predicted probabilities

$\theta_{2\times3} = \begin{bmatrix} 1 & 0 & 1 \\ 0 & 1 & 1 \end{bmatrix}$

Step 1: Compute scores (LCF)

$$s = x^T\theta \ (\in \mathbb{R}^M)$$

$$(2\ 1)\begin{bmatrix} 1 & 0 & -1 \\ 0 & 1 & 1 \end{bmatrix} = \begin{bmatrix} 2 & 1 & -1 \end{bmatrix}$$

cat dog fish

Step 2: Convert to probabilities (Softmax)

$$p_m = \frac{\exp(s_m)}{\sum_j \exp(s_j)} \ for\, m = 1, \ldots, M$$

$$y_p \in \mathbb{R}^M, \sum_m p_m = 1$$

$P_{dog} = 0.25$

$P_{fish} = 0.05$

$$P_{cat} = \frac{\exp(S_{cat})}{\exp(S_{cat}) + \exp(S_{dog}) + \exp(S_{fish})}$$

$$= \frac{\exp(2)}{\exp(2) + \exp(1) + \exp(-1)}$$

$= 0.7$

Georgia Tech

Using Cross Entropy + Softmax

$y_a = [1 \quad 0 \quad 0]$

$\hat{y}_p = p_m = [0.7 \quad 0.25 \quad 0.05]$

Step 3: Compute loss (Cross-Entropy)

- $L = -\sum_{m=1}^{M} y_a^{(m)} \log(p_m)$
- For one datapoint: $L = -\log(p_{\textit{correct class}})$

$L = -\log(0.7)$

Step 4: Gradient

- $\nabla_\theta L = x(y_p - y_a)^T \; (\in \mathbb{R}^{d \times M})$

2×1 3×1 3×1

$\nabla_\theta L$ is of size 2×3, $\theta_{2\times3}$

$\begin{bmatrix} 2 \\ 1 \end{bmatrix} \cdot \begin{bmatrix} 1-0.7 & 0-0.25 & 0-0.05 \end{bmatrix}$

Step 5: SGD Update

- $\theta \leftarrow \theta - \alpha x(y_p - y_a)^T$

2×3, 2×3, hyp., 2×3

$\theta_{new} = \begin{bmatrix} 1.5 & -0.5 & -0.01 \\ 0.5 & 0.5 & 0.2 \end{bmatrix}$

Prediction (after training)

$\hat{y} = \arg\max_m \quad p_m$

Georgia Tech

Take-Home Messages

- Generative and Discriminative Classification
- The Logistic Regression Model
- Understanding the Objective Function – Log Likelihood with sigmoid
- Gradient Descent for Parameter Learning
- Multiclass Logistic Regression – Using CE as objective function

Quick Knowledge Check

- Which of the following best describes a generative model? A. Directly models $P(y|x)$ B. Models $P(x|y)$ and $P(y)$, then uses Bayes' rule for $P(y|x)$ C. Uses cross-entropy to learn the decision boundary D. Requires no assumptions about data distribution

- Logistic Regression is an example of a: A. Generative model that reconstructs data distribution B. Hybrid model using both priors and likelihoods C. Discriminative model that directly estimates $P(y|x)$ D. Model that approximates $P(x|y)$ with Gaussian likelihoods

$$\text{Odds} = \frac{P(y=1)}{P(y=0)} = \text{Logit}$$

- What does the logit function represent? A. The derivative of the sigmoid B. The log of the odds ratio $\log \frac{P(y=1|x)}{1-P(y=1|x)}$ C. The posterior probability itself D. The exponential of the decision boundary

- In binary logistic regression, the model uses the sigmoid $g(x)= \frac{1}{1+e^{-x\theta}}$ to predict class probabilities. The objective function typically: A. Maximizes the joint likelihood $P(x,y)$ B. Minimizes squared error between predictions and labels C. Minimizes the negative log-likelihood based on the sigmoid output D. Minimizes the variance of the Gaussian features

Georgia Tech

Quick Knowledge Check

- In multiclass logistic regression using softmax, the objective function: A. Computes separate sigmoids for each class independently B. Minimizes the mean squared error across classes C. Maximizes the total log-likelihood using cross-entropy over all classes

- Gradient descent in logistic regression is used to: A. Optimize parameters by minimizing or maximizing the objective function B. Update weights in the direction of the gradient of log-likelihood C. Compute priors and posteriors explicitly D. Eliminate class imbalance by reweighting samples

- True or False: Gaussian Naïve Bayes and Logistic Regression yield identical decision boundaries under equal variance and Gaussian feature assumptions. A. True B. False

Georgia Tech