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.

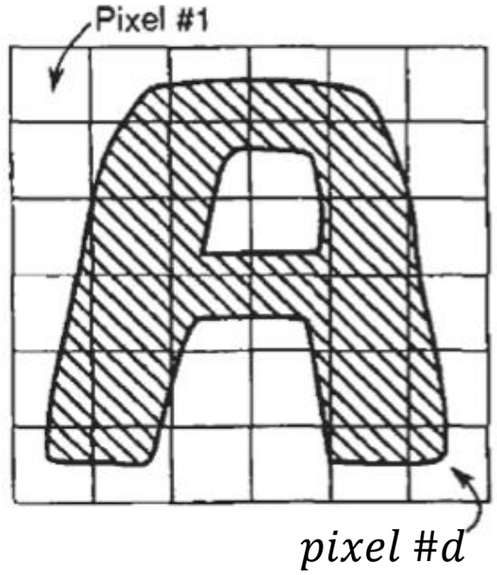
Georgia Tech

Outline

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

Classification

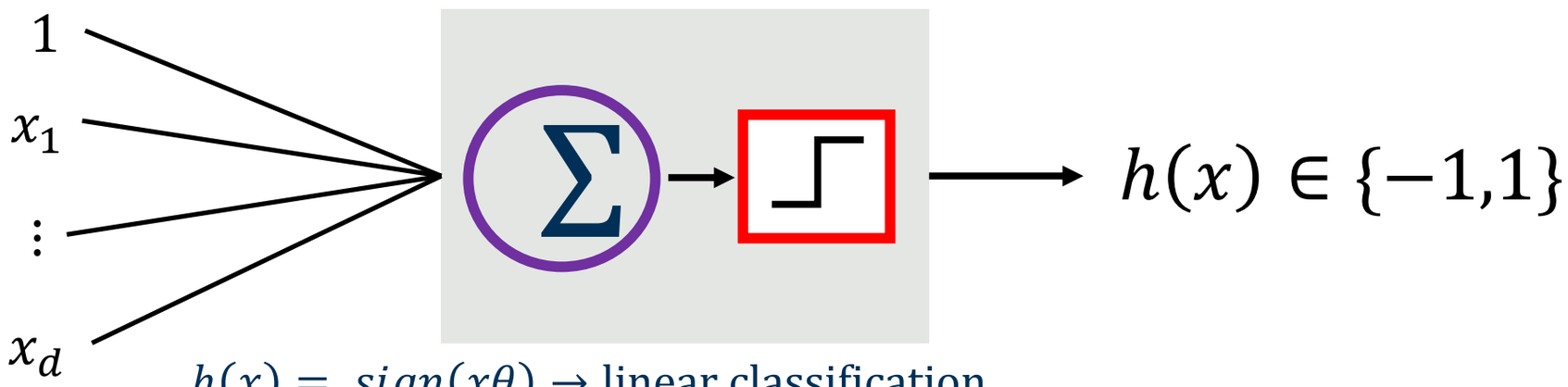
- Represent the data



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

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

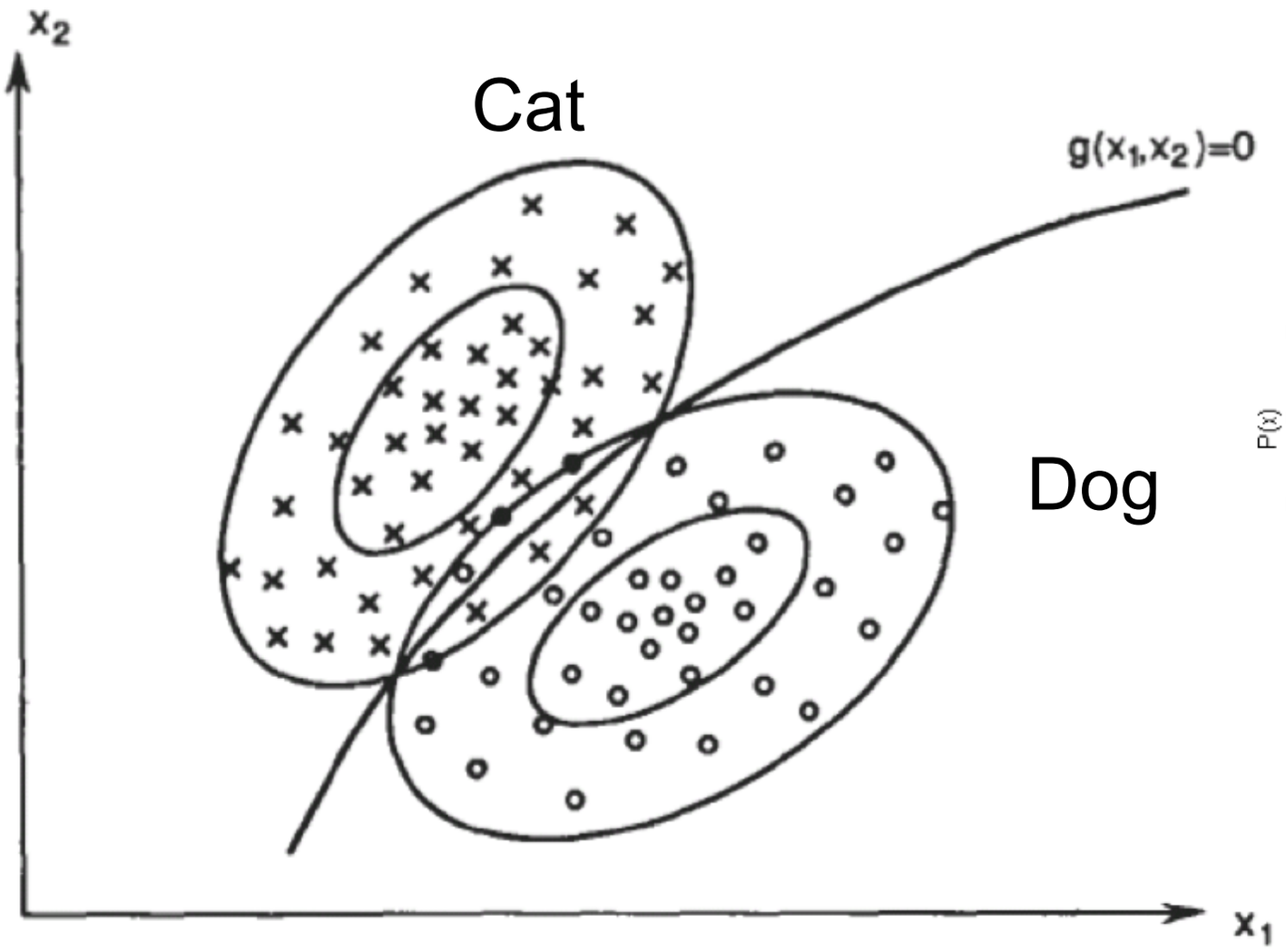
- Classifier

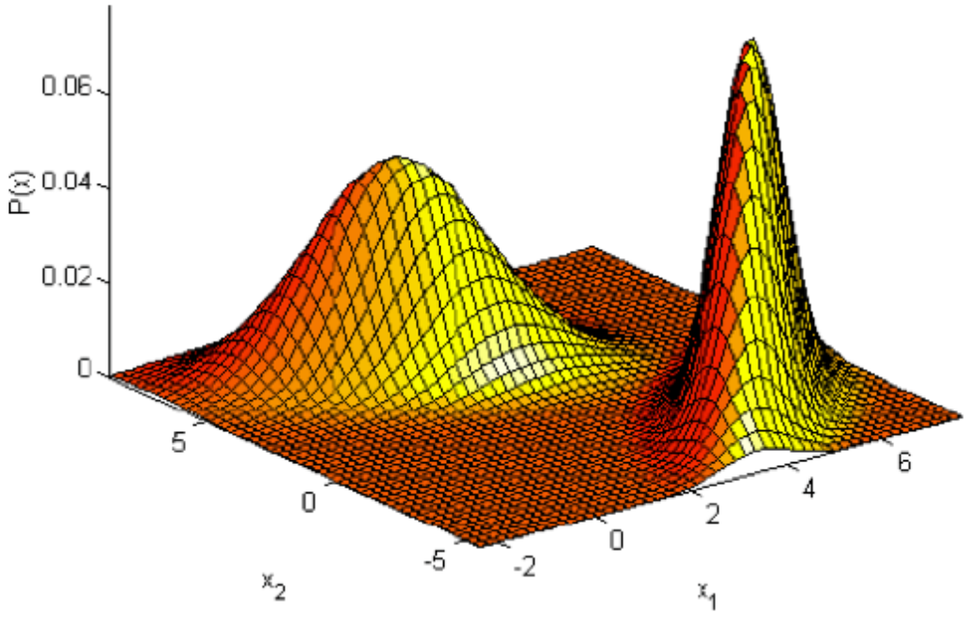


$h(x) = sign(x\theta) \rightarrow$ linear classification

Decision Making: Dividing the Feature Space

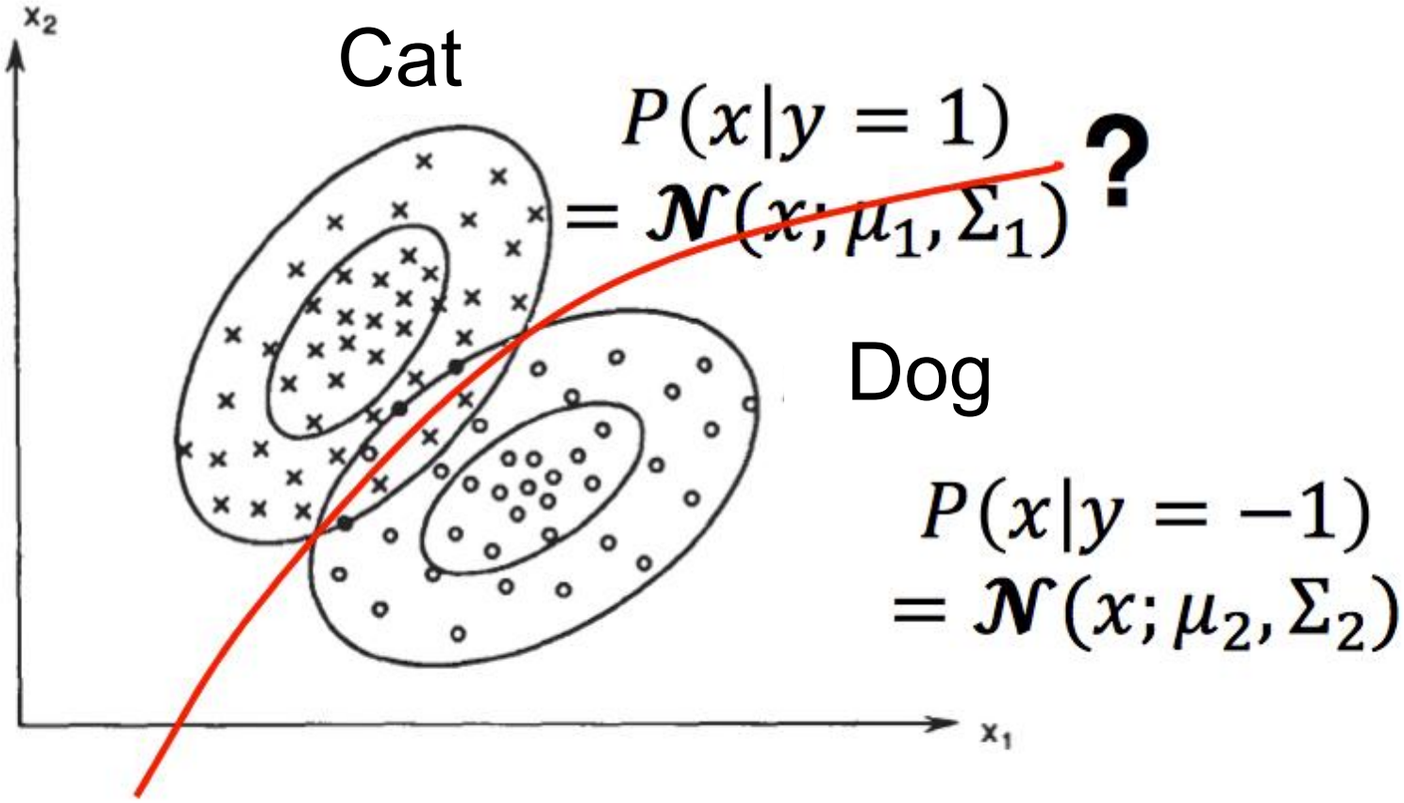
- Distributions of sample from normal (positive class) and abnormal (negative class) tissues





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

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

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

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

Georgia Tech

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:

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

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

Gaussian naïve Bayes
A typical assumption

Georgia Tech

“Naïve” conditional independence assumption

Real World Example

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

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

Outline

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

Recap

- Generative vs Discriminative

- Generative Example?
- Assumptions to calculate posterior

Gaussian Naïve Bayes

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

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

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

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

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

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

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

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

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

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

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

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

Gaussian Naïve Bayes is a subset of logistic regression

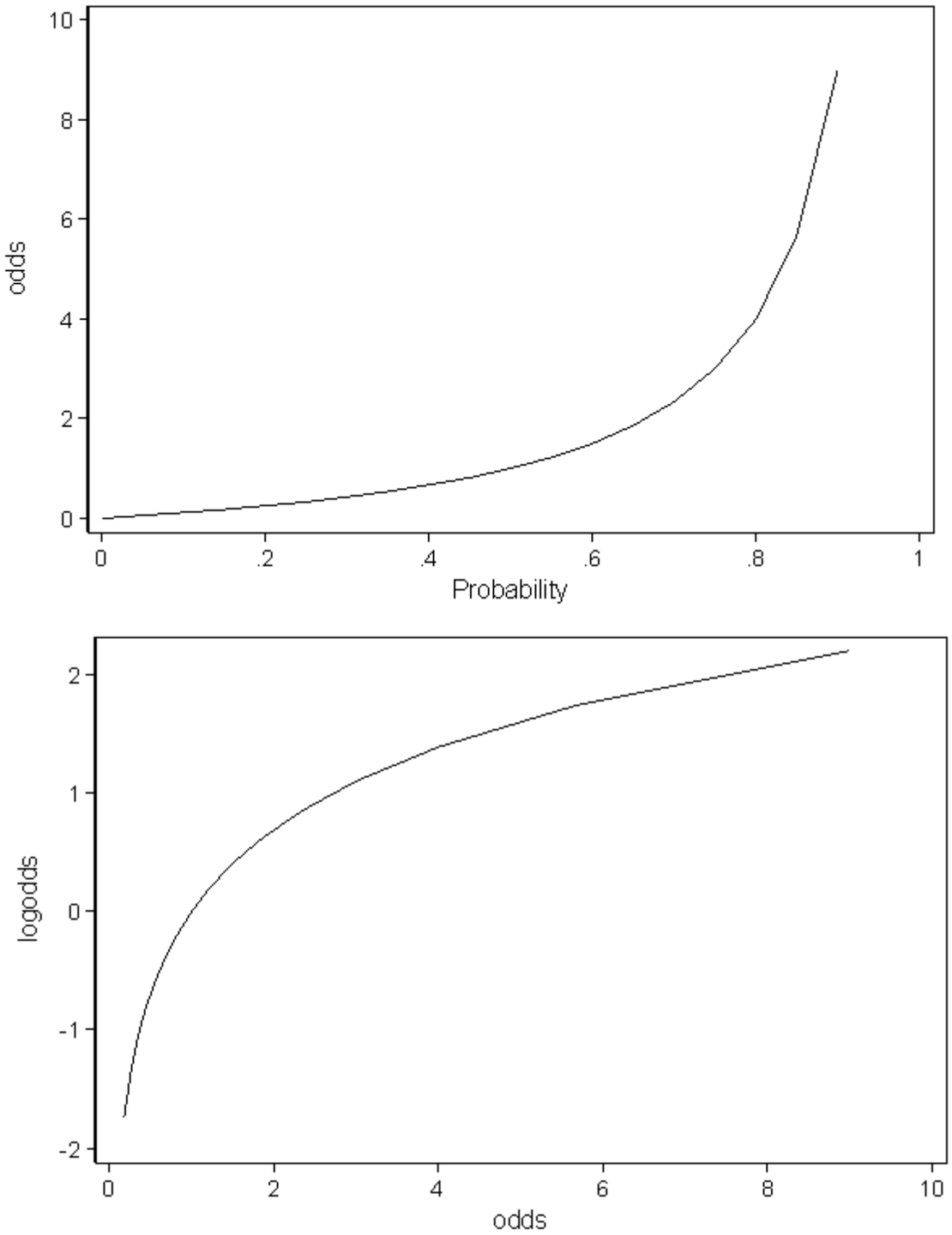
Georgia Tech

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

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

$log(odds)\ vs\ odds$

What could be $x\theta$ domain?



Georgia Tech

What is logit function?

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

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

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

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

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

Georgia Tech

Interpretation

Generative Naïve Bayes (model the data distribution) →
algebra simplification →
discriminative Logistic Regression (directly model the decision boundary).

In Naïve Bayes, we model how each class generates data — $P(x \mid y)$.
When we assume Gaussian distributions and equal variances and take logs, **the log of the posterior becomes linear in x.**

That's the same form as logistic regression — which means logistic regression is effectively learning the same function, but without assuming a Gaussian form.
So, Gaussian Naïve Bayes is a generative shortcut that leads us to the discriminative logistic regression formulation

Georgia Tech

Interpretation

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))

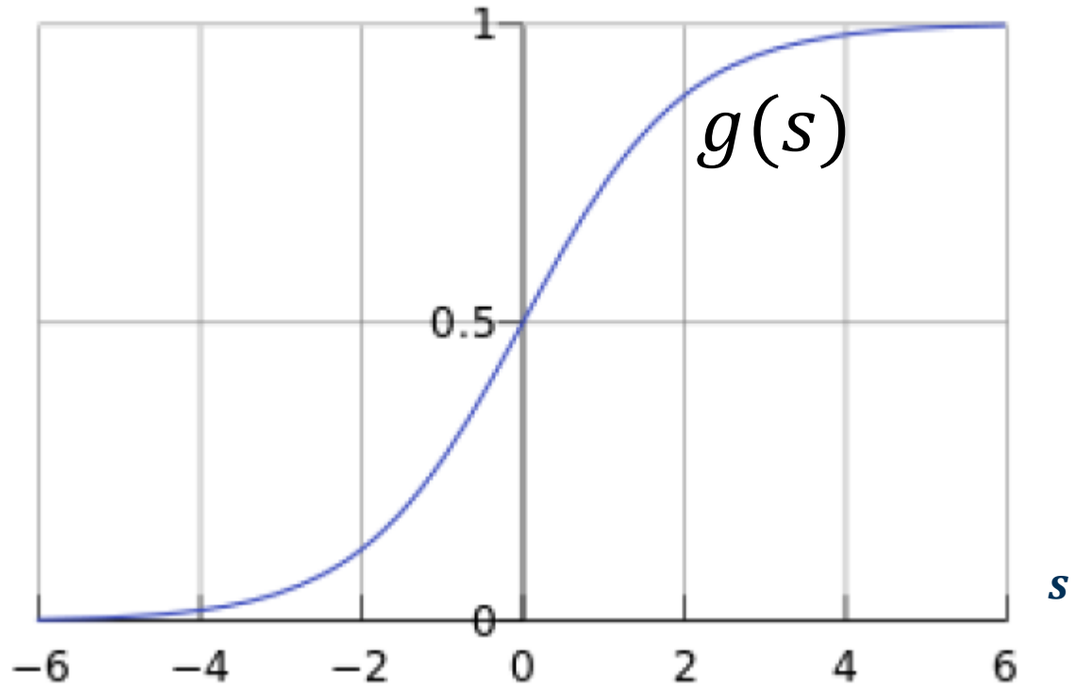
Logistic function for posterior probability

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)

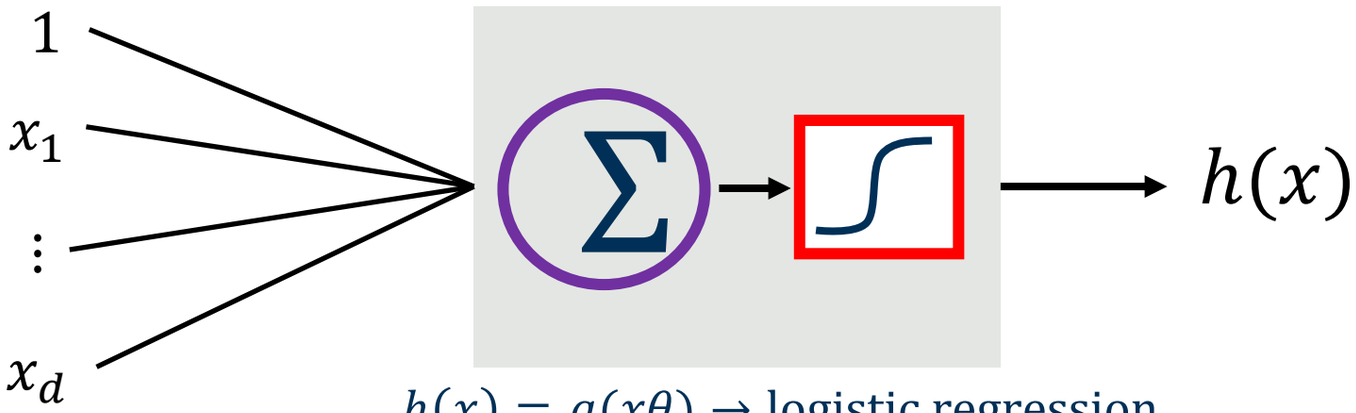
(Interactive)

Georgia Tech

Sigmoid Function

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

$$s = \sum_{i=0}^{d} x_i \theta_i = \theta_0 + {}_{\theta}1 x_1 + \cdots + {}_{\theta}d x_d$$

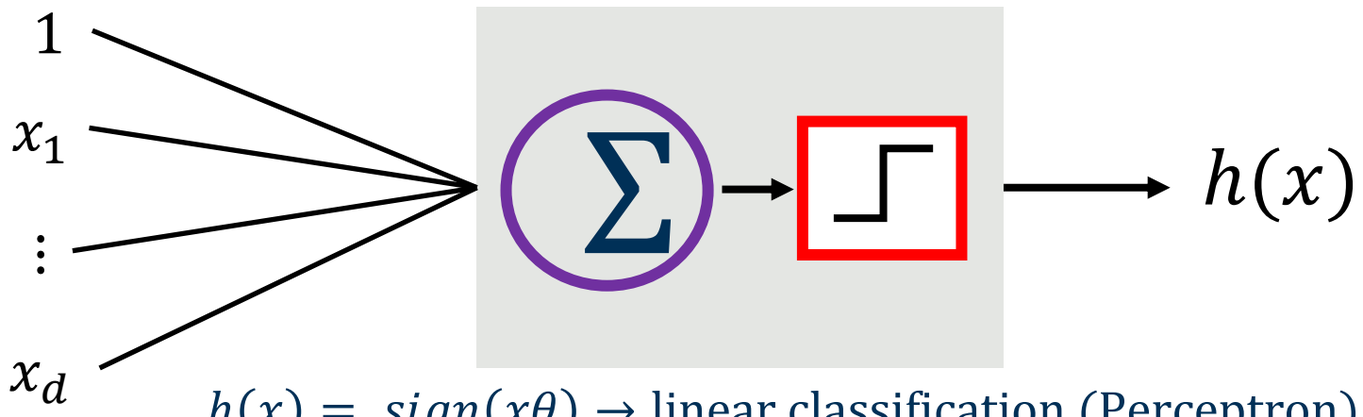


$h(x) = g(x\theta) \rightarrow$ logistic regression

Soft classification
Posterior probability

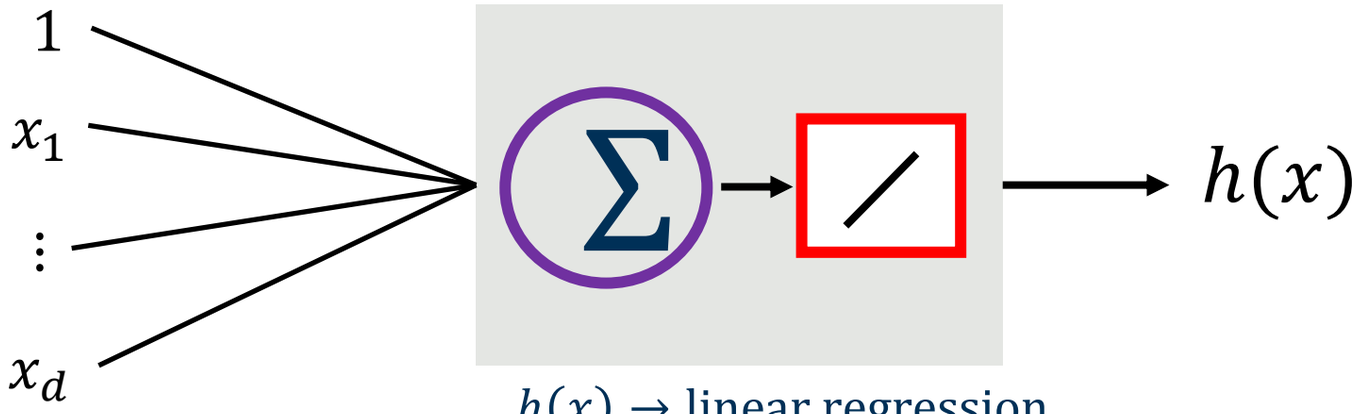
Three linear models

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

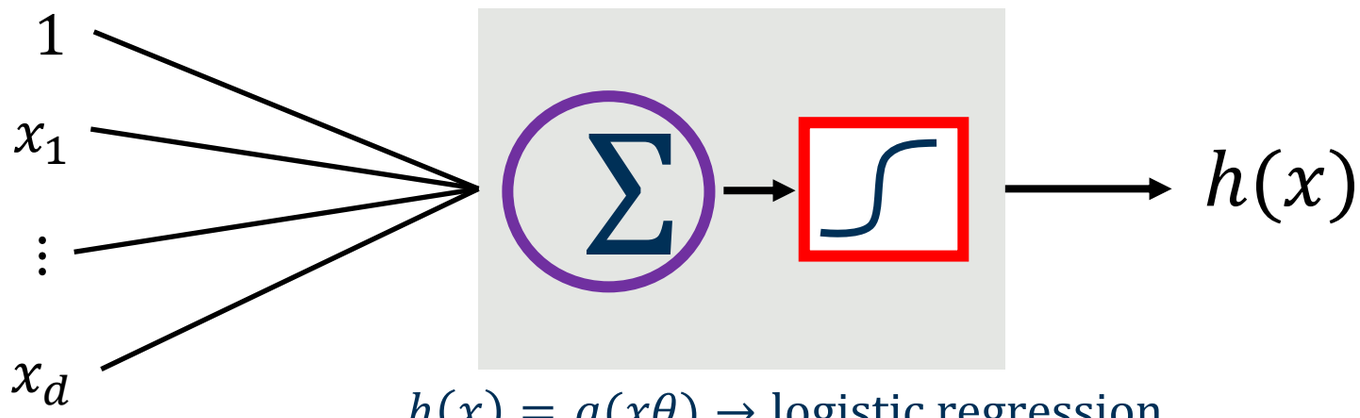


Hard classification

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



$h(x) \rightarrow$ linear regression



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

Soft classification
Posterior probability

Georgia Tech

Why Soft classification matters

Example: Prediction of heart attacks

Input $\boldsymbol{x}$: 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

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

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

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

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

This form is concave, negative of this form is convex

Georgia Tech

Georgia Tech

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

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

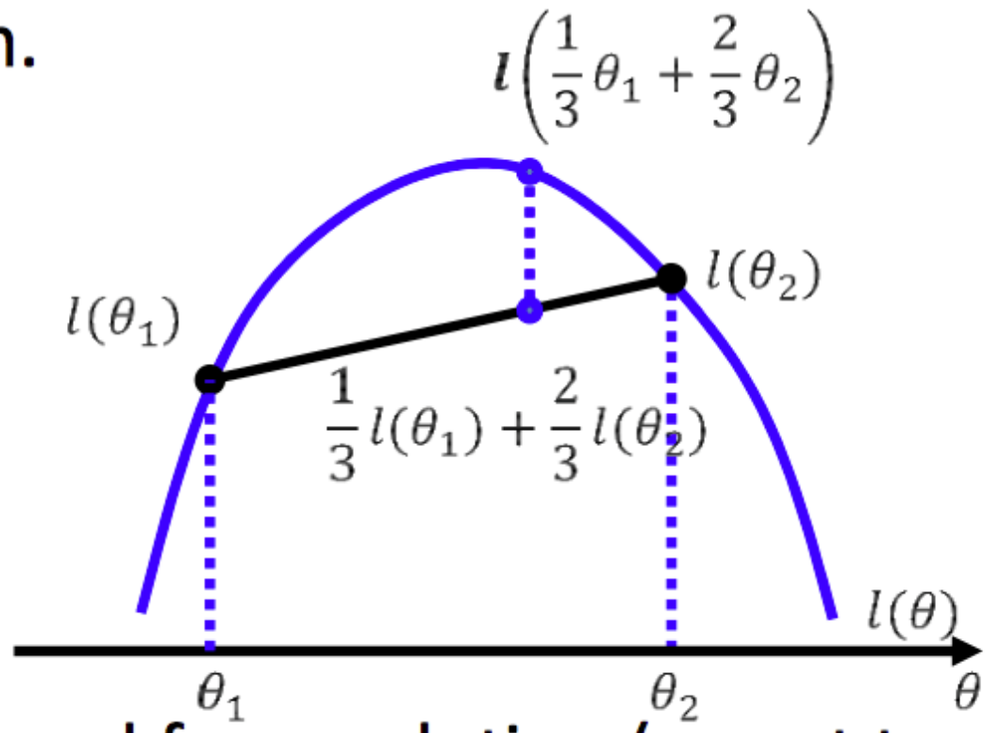
- Setting it to 0 does not lead to closed form solution

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

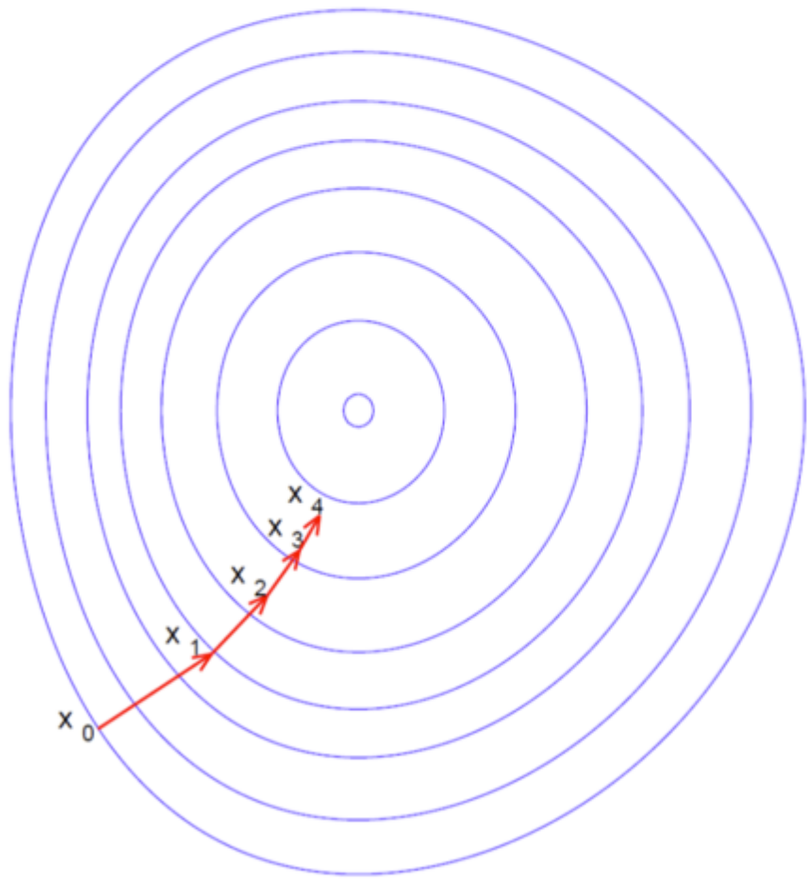
- Given an initial guess, we *iteratively* refine the guess by taking the direction of the negative gradient

- 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

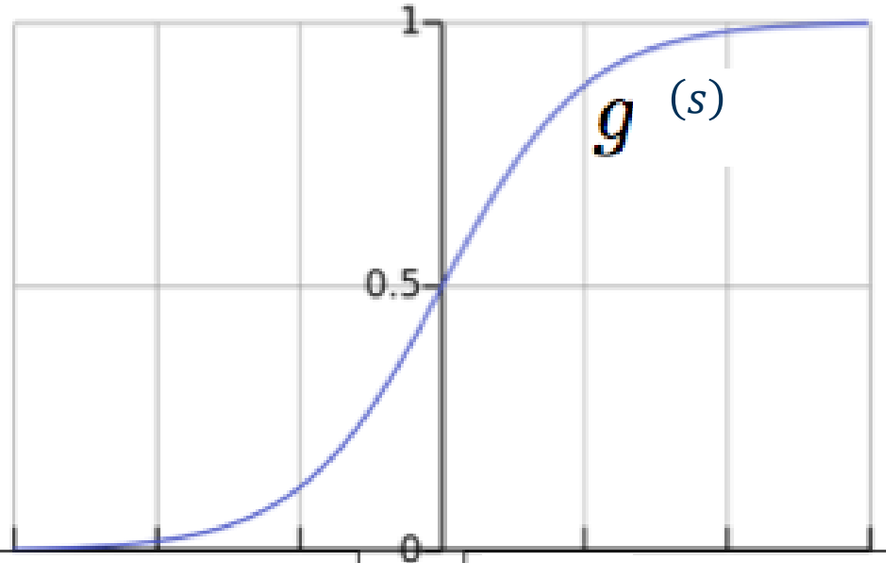
$$\theta^{t+1} \leftarrow \theta^t + \eta \sum_i (x^{\{i\}})^T \left(y^{\{i\}} - 1\right) + (x^{\{i\}})^T \frac{\exp\left(-x^{\{i\}}\theta\right)}{1 + \exp(-x^{\{i\}}\theta)}$$

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

Logistic Regression

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

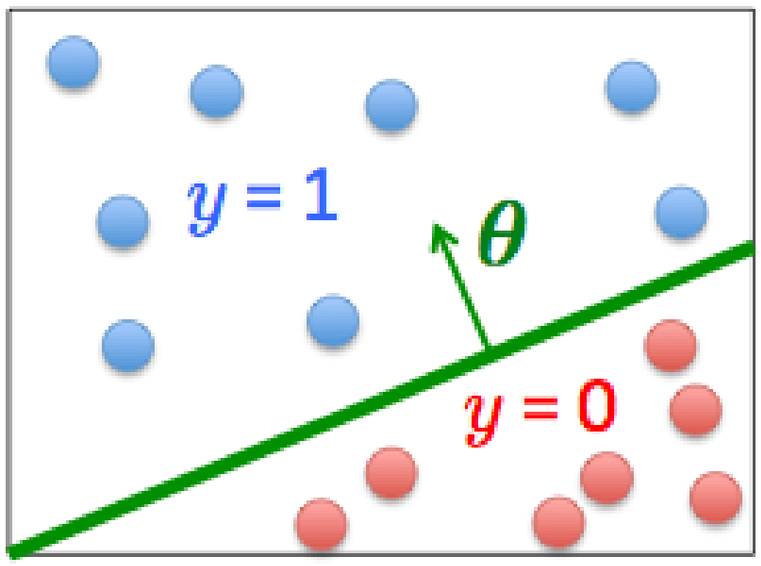
$$s = x\theta$$



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

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

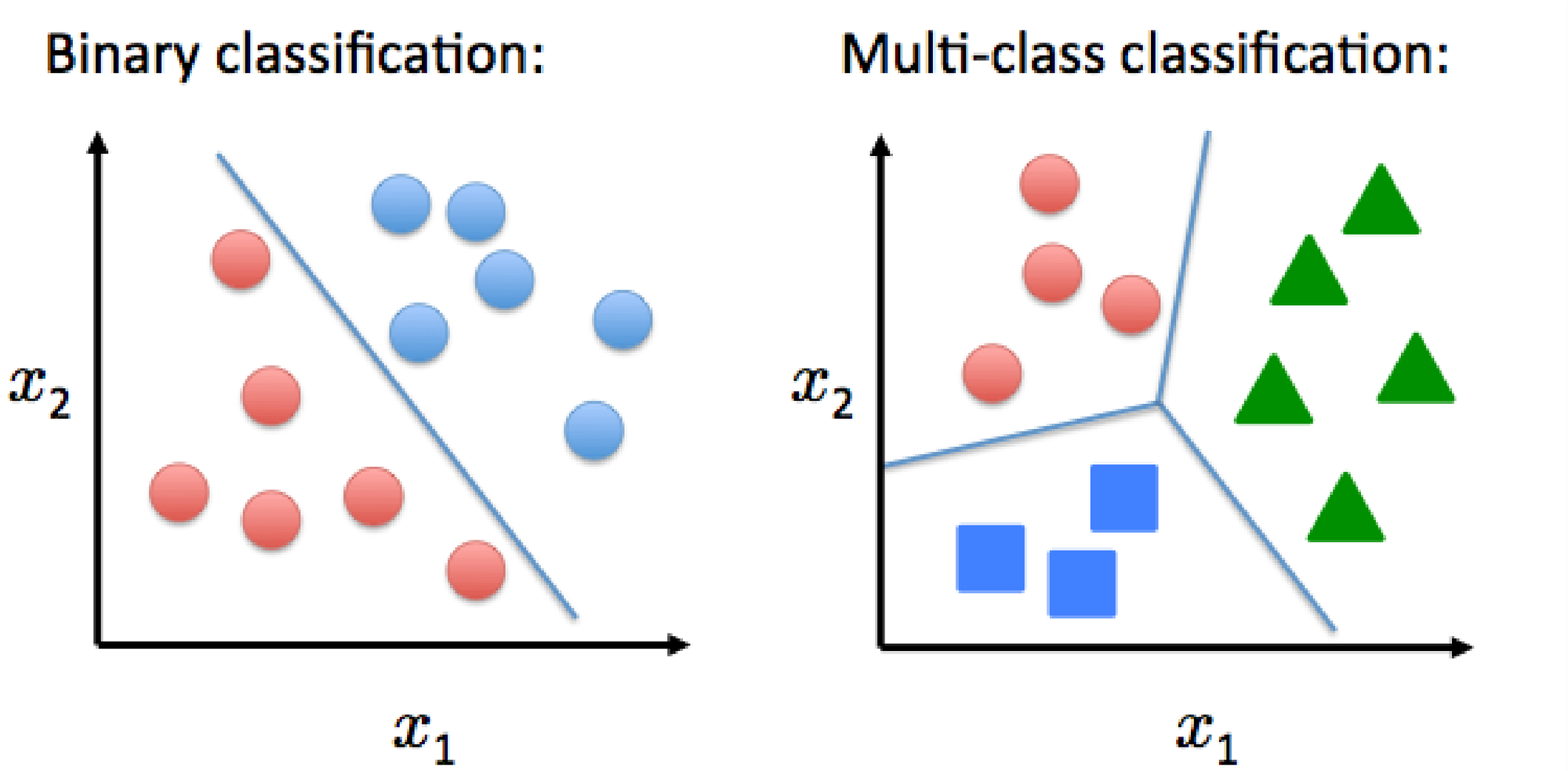
- Assume a threshold and...
  - Predict $y = 1$ if $g(s) \geq 0.5$
  - Predict $y = 0$ if $g(s) < 0.5$



Outline

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

Multiclass Logistic Regression

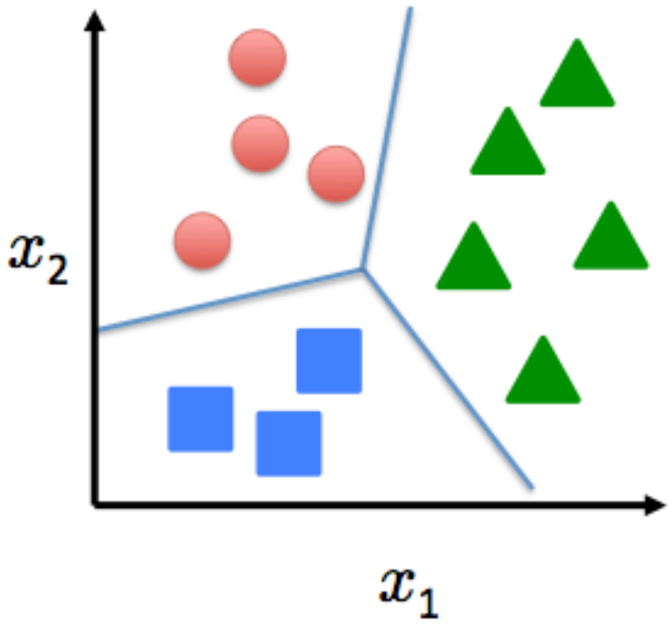


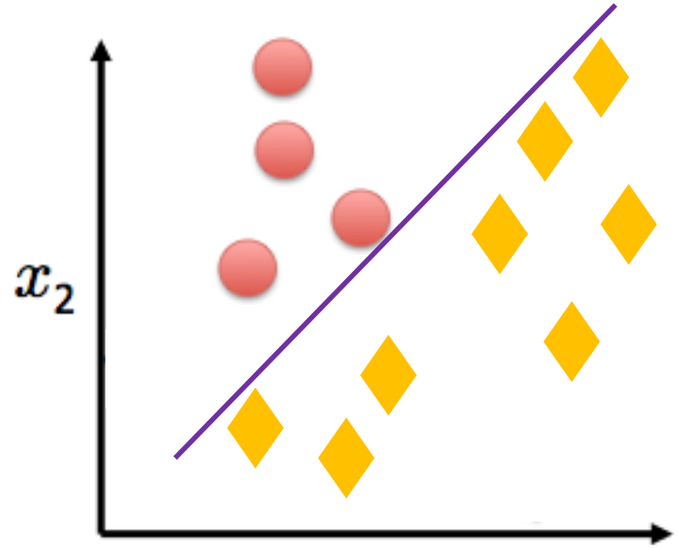
Disease diagnosis: healthy / cold / flu / pneumonia

Object classification: desk / chair / monitor / bookcase

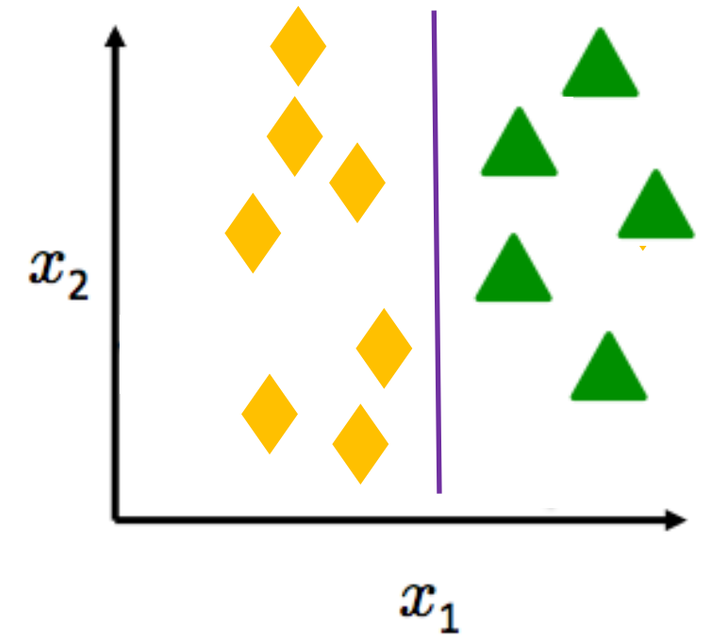
One-vs-all (one-vs-rest)

Multi-class classification:



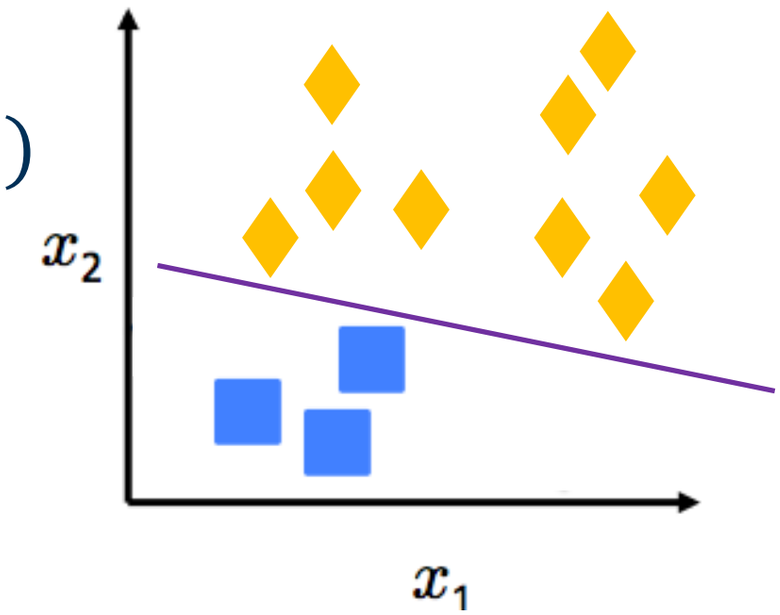


$h_\theta^1(x)$



$h_\theta^2(x)$

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



$h_\theta^3(x)$

Georgia Tech

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

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

$$y_a = [cat, dog, fish] = [1,0,0]$$
$$\Rightarrow there\ are\ M\ classes\ (M = 3\ in\ this\ example)$$

$$y_{p\ for\ class\ \boldsymbol{m}} = softmax(x\theta) = \frac{\exp(x\theta)_{\boldsymbol{m}}}{\sum_{j=0}^{M} \exp(x\theta)_j}$$

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

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

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

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
- 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
- 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