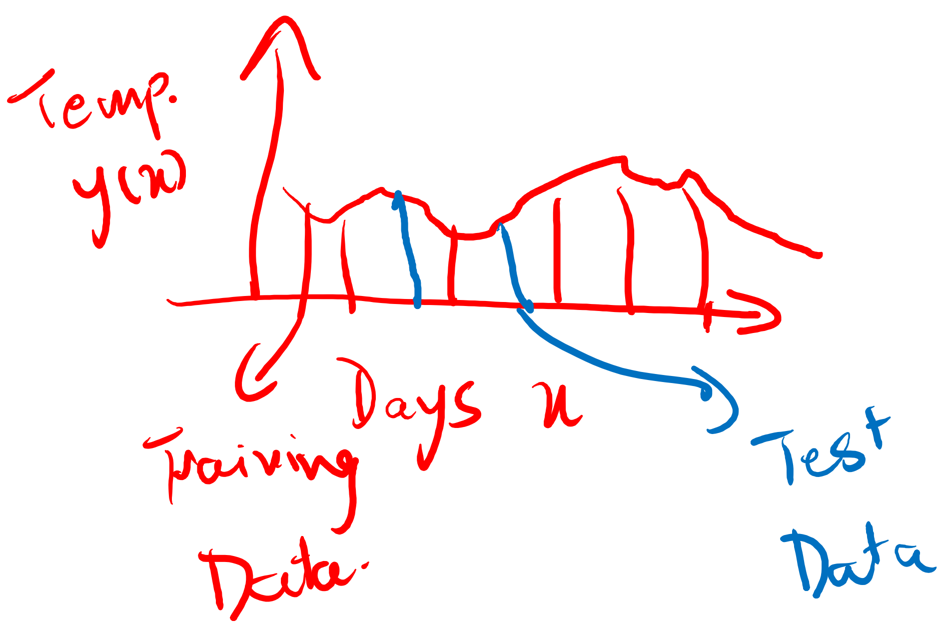
Naïve Bayes and Logistic Regression

Nakul Gopalan
Georgia Tech

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

Regression

$y(x)$



Mean Sq. Error

$$L(\theta) = \frac{1}{n}\sum_i^n (y(x_i) - \hat{y}(x_i))^2$$

Sparseness → LASSO

0.1 → 0.01

0.01 → 0.0001

Polynomial Regression

$$y(x) = 1 + a_1 x^1 + a_2 x^2 \dots a_n x^n$$

$$z_n = x^n$$

$$y(z) = 1 + a_1 z_1 + a_2 z_2 \dots + a_n z_n$$

Regularization → ~~of~~ overfitting & generalization

Lagrange Multiplier:

Ridge → $L(\theta) = \frac{1}{2n}\sum_n (y(x_i) - \hat{y}(x_i))^2 + \lambda\|\theta\|^2$

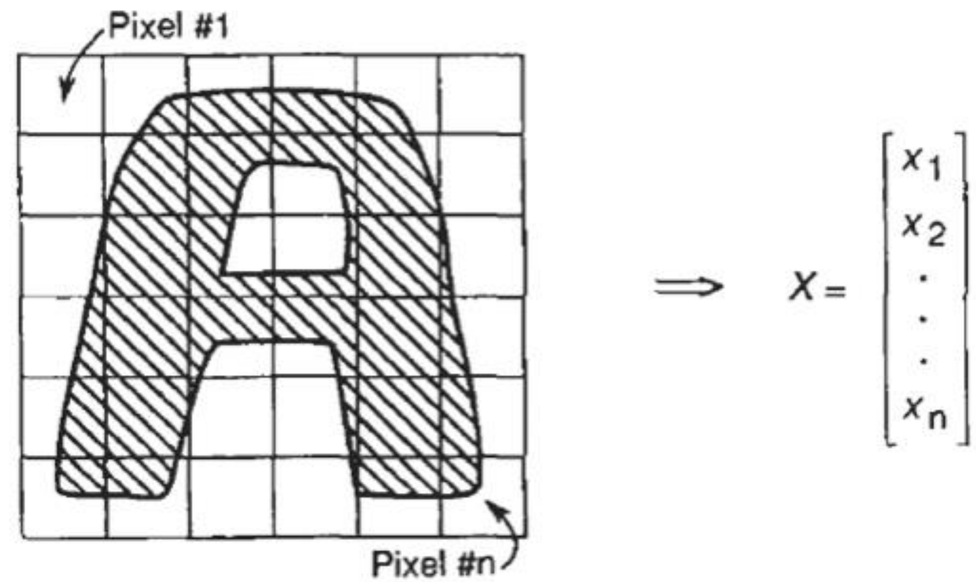
→ $L(\theta) = \frac{1}{n}\sum (y(x_i) - \hat{y}(x_i))^2 + \lambda\|\theta\|$

Outline

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

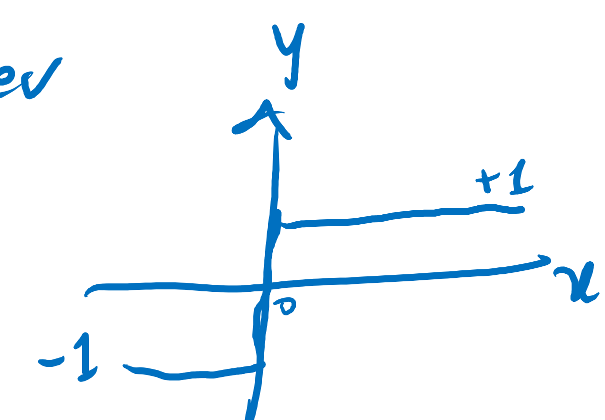
Classification

- Represent the data



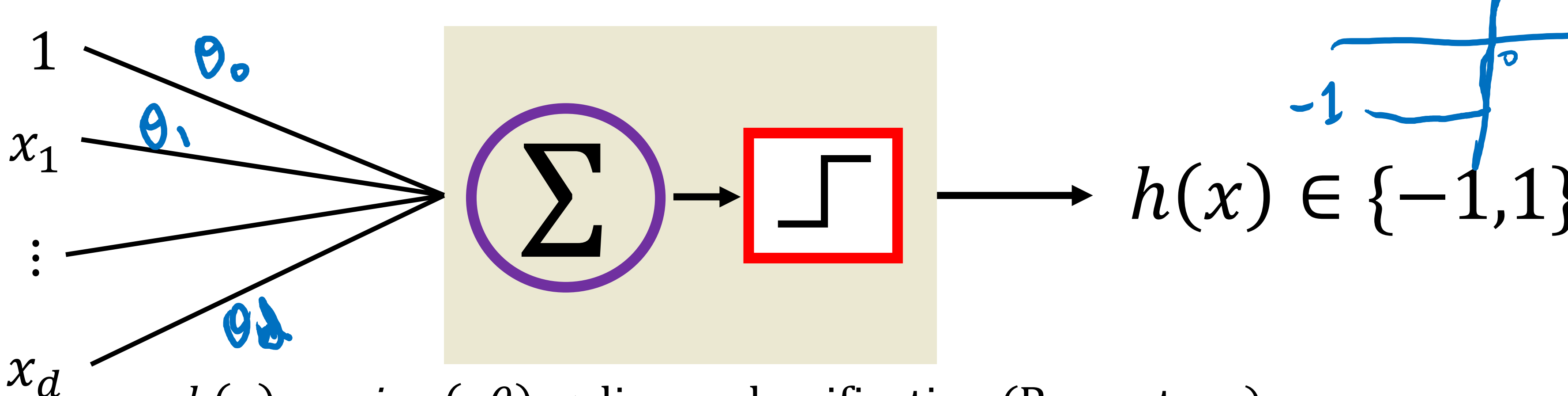
- A label is provided for each data point, eg., $y \in \{-1, +1\}$

  ↳ image of character



- Classifier

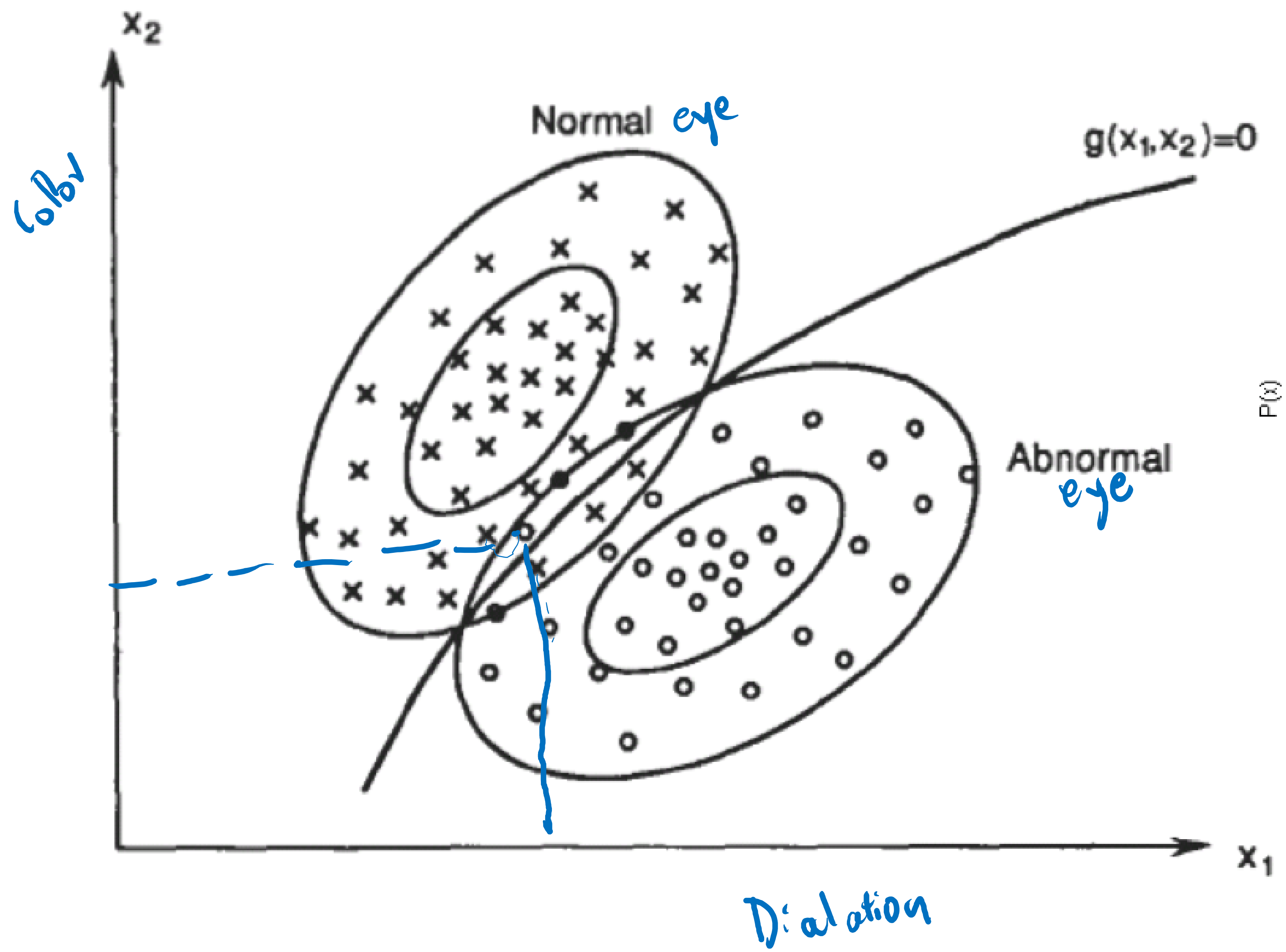
$\sum_{i \in d} x_i \theta_i$

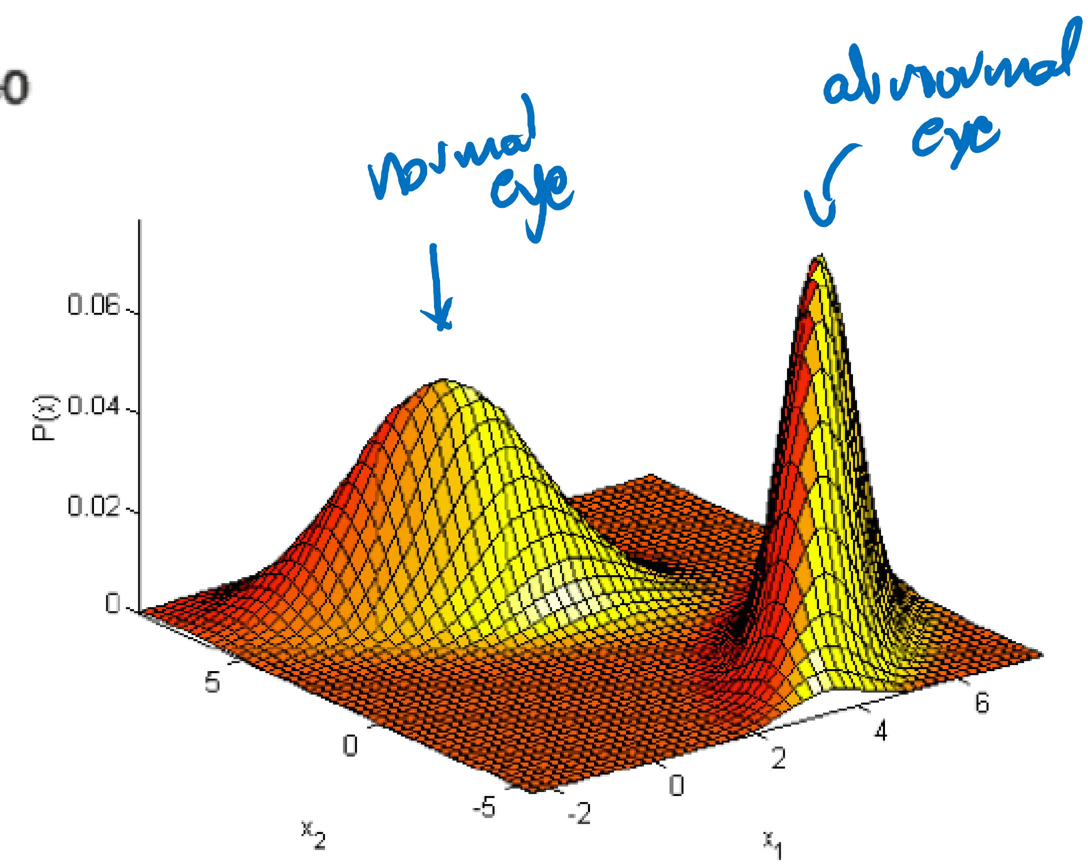


$h(x) = sign(x\theta) \rightarrow$ linear classification (Perceptron)

Decision Making: Dividing the Feature Space

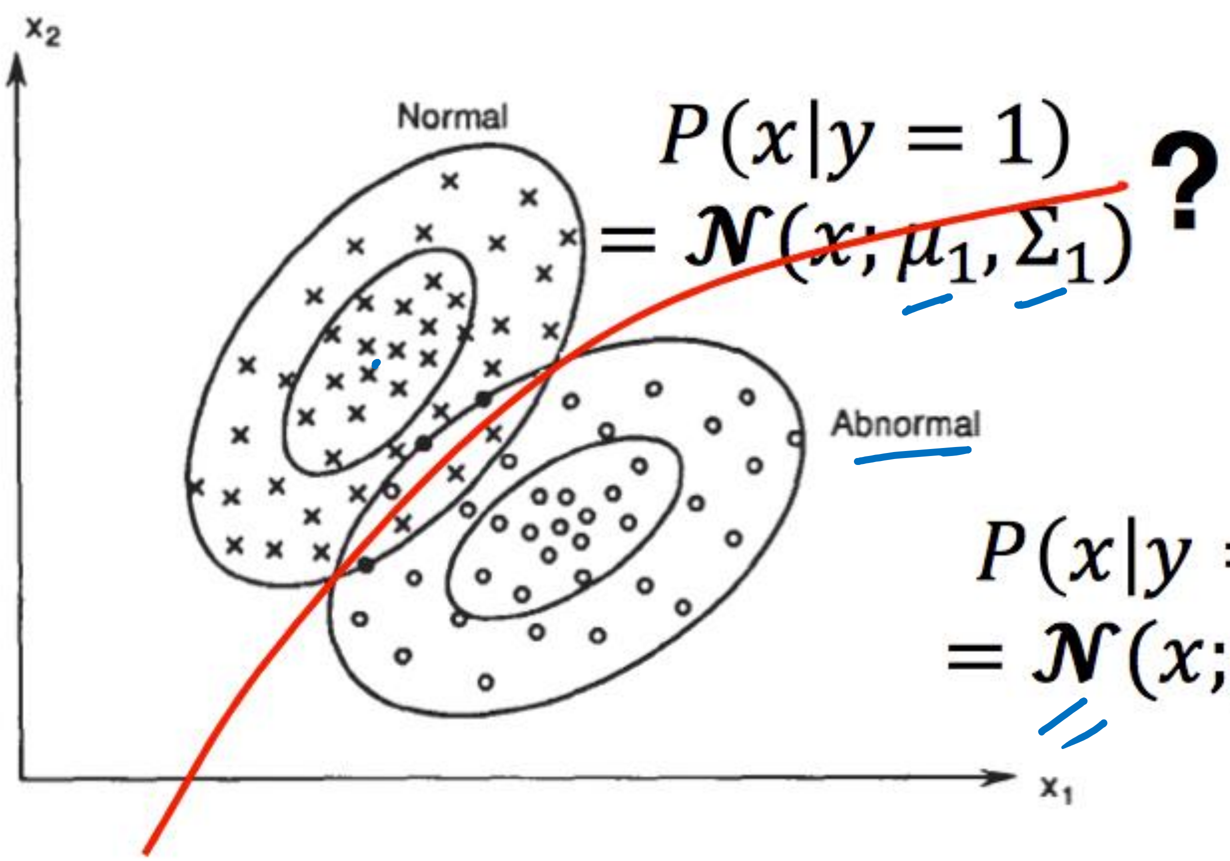
- Distributions of sample from normal (positive class) and abnormal (negative class) tissues





How to Determine the Decision Boundary?

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



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

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

$x_1 = x_{\text{pupil dilation}} = 0.05$ mm

$P(x_1 | y = \text{normal})$

$P(x_1 | y = \text{abnormal})$

Bayes Decision Rule

dialation intensity

$P(y)$
$P(x)$

$p(y|x)$ →

	x^1	x^2	Class
1	0.005	0.1	+1
	2	0.5	-1
⋮			⋮
1000	0.004	0.01	+1

Normal dist.

likelihood

Prior

Probability of normal eye given x_1 →

$x_1 = [0.005 \quad 0.1]$

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

→ marginalization

posterior

normalization constant

$P(x|y=+1) \sim \mu_1 = [x^1_\mu \quad x^2_\mu]^T$

$\Sigma = 2 [\quad]$

Prior: $P(y)$

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

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

Bayes Decision Rule

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

Posterior

- The poster probability of a test point

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

$+1/-1$

Def.

$q_i(x) > q_j(x)$

$P(x|y=i)P(y=i) \cancel{P(x)} > P(x|y=j)P(i) \cancel{P(x)}$

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

$1 > q_j / q_i$

$0 > \ln(q_j/q_i)$

- 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) $= -\ln\frac{q_i(x)}{q_j(x)}$ = 0 (decision boundary)

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

Mean, Var.
compute $p(x|y)$
↳ unseen points

- 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.

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

$$p(x = [x^1, x^2] \mid y = 1) = \prod_{i=1}^{2} (p(x^i \mid y = 1))$$

- 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:

$P(A,B) = p(A|B)\,p(B)$

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

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

$$= \cdots$$

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

Naïve Bayes assumption: let's rewrite it as:

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

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

Gaussian naïve Bayes

A typical assumption

[Example]()

Naïve Bayes cat vs dog!

y	x_1 wt ~~height~~ kg	x_2 ~~tail~~ ht. ft	x_3 tail length
cat	2	1	1
dog	6	2	2
dog	30	3	20
cat	1	0.5	0.5

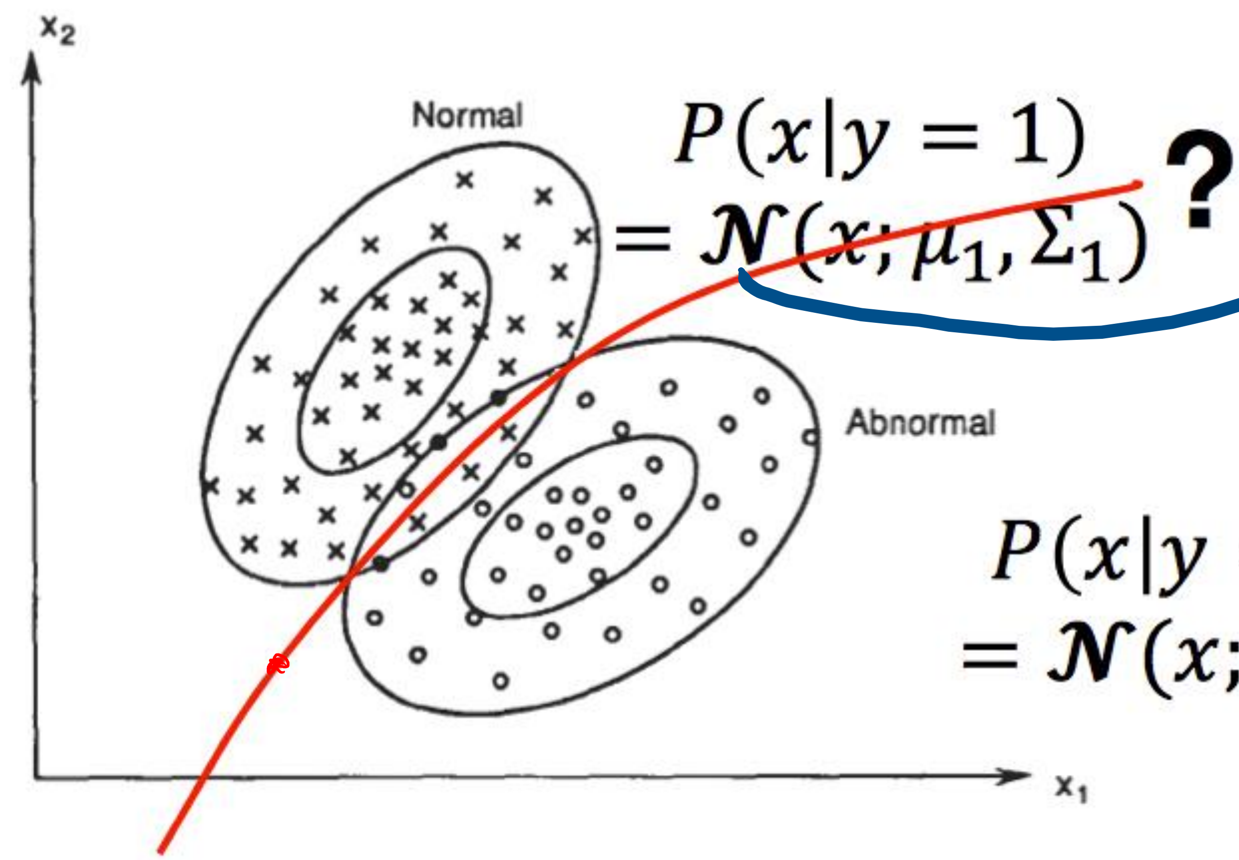
Joint dist.

$$= P(x = [10, 2, 0], y = \text{cat}) = P(y = \text{cat}) \cdot P(\overset{\text{wt}}{10} \mid \text{cat}) \cdot P(\text{ht} = 2 \mid \text{cat}) \cdot P(0 \mid \text{cat})$$

Administrative things

- Project team composition due this weekend
- Quiz out today. Let us know if you have problems with it, and take as many as you can, it will only help!!
- Homework due next week. Deadlines will start getting closeby from now.

Naïve Bayes



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

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

Boundary at
$P(y=\text{normal}|x) = P(y=\text{abnormal}|x)$

For cat vs dog :

$P(y=\text{cat}\,|\,x) = P(y=\text{dog}\,|\,x)$

$x \rightarrow$ features like tail length, height, weight.

Naïve Bayes

$$P(x \mid y) \approx \prod_i P(x_i \mid y)$$

independence assumption b/w features

$$P(x \mid y=cat) \approx P(\text{tail length}=1 \mid y=cat) \cdot P(wt=2 \mid y=cat) \cdot P(ht=1 \mid y=cat)$$

learning large conditional dist^n: hard

Actually

$$P(x \mid y=cat) = P(\text{tail length}=1 \mid wt=2, ht=1, y=cat) \cdot P(wt=2 \mid ht=1, y=cat) \cdot P(ht=1 \mid y=cat)$$

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

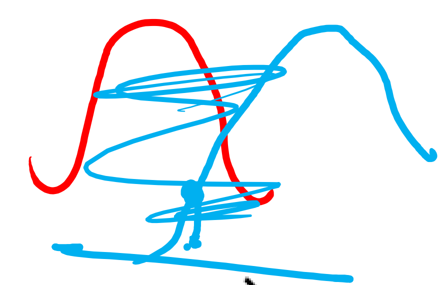
Gaussian Naïve Bayes

Prior   Naive ind. assumption

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

$$P(A|B)=\frac{P(B|A)\,P(A)}{P(B)}$$

Normal dist.



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

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

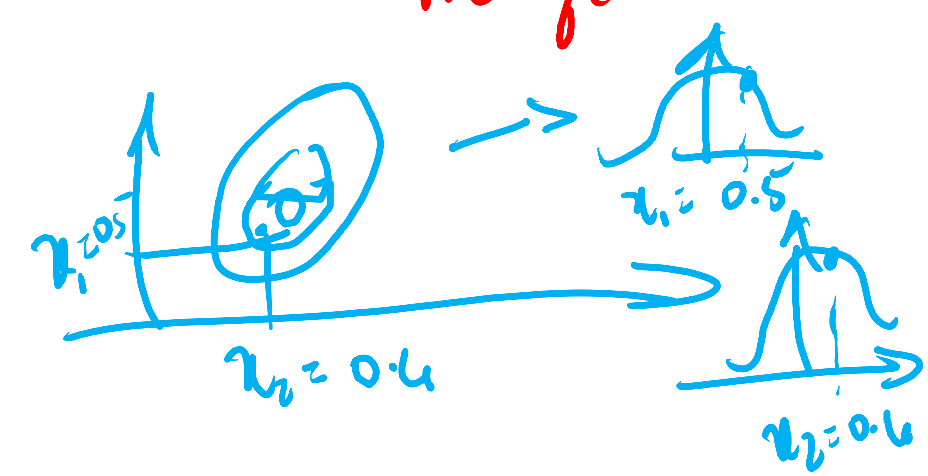
class   which feature

cat vs dog

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

π_1

π_2



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

$P(y|x) = P(x|y) \cdot P(y) / P(x)$

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

→ Previos slide

Marginalisation → by summing over y or labels

labels

π_k → $P(y_k)$

k → class

i → feature i.d.

$P(x) = \sum_{k \in \text{labels}} P(x, y_k)$

$= \sum_k P(x|y_i) \cdot P(y_k)$

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

$u = \exp(\ln(u))$

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

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

dividing by nr.

$-\mu_2 = -(-\mu_1)$

$$\frac{c_1}{\sum_{k \in 1,2} c_k} = \frac{c_1}{c_1+c_2} = \frac{1}{1+\frac{c_2}{c_1}}$$

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

$$\frac{\exp(c_2)}{\exp(c_1)} = \exp(c_2-c_1)$$

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

Number of parameters:

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

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

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?

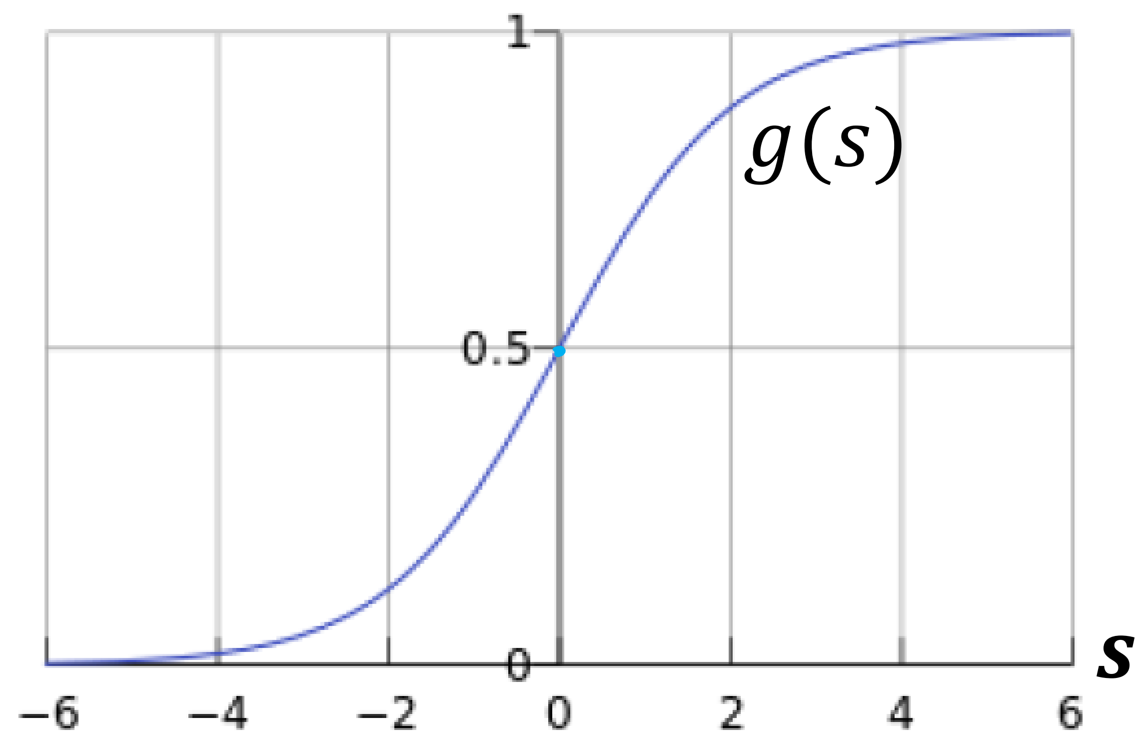
Gaussian Naïve Bayes is a subset of logistic regression

Logistic function for posterior probability

Many equations can give us this shape

Let's use the following function:

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



$\frac{1}{1+1} \because e^0 = 1$

This formula is called sigmoid function

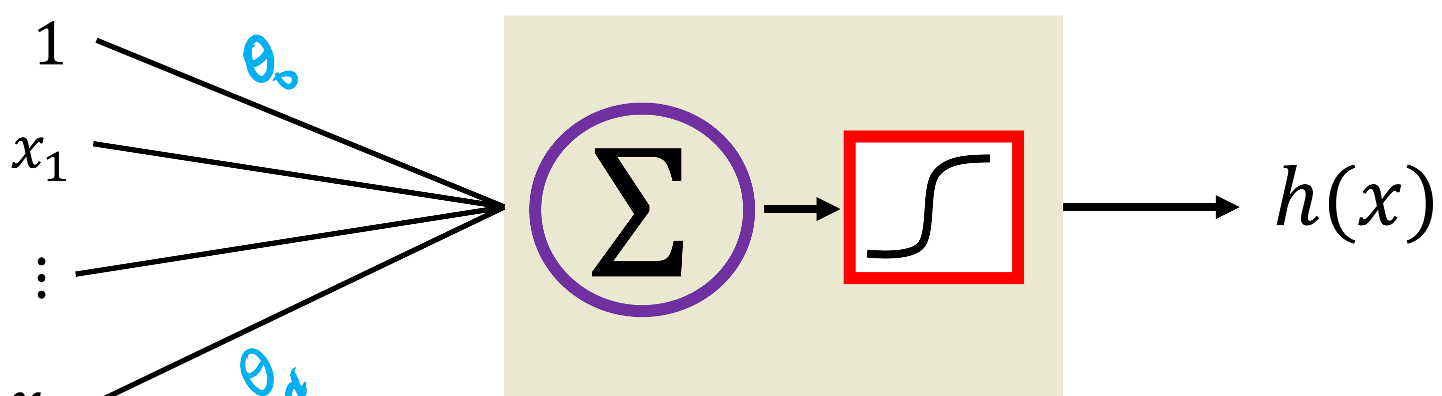
It is easier to use this function for optimization

derivative is hard to compute

Sigmoid Function

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

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

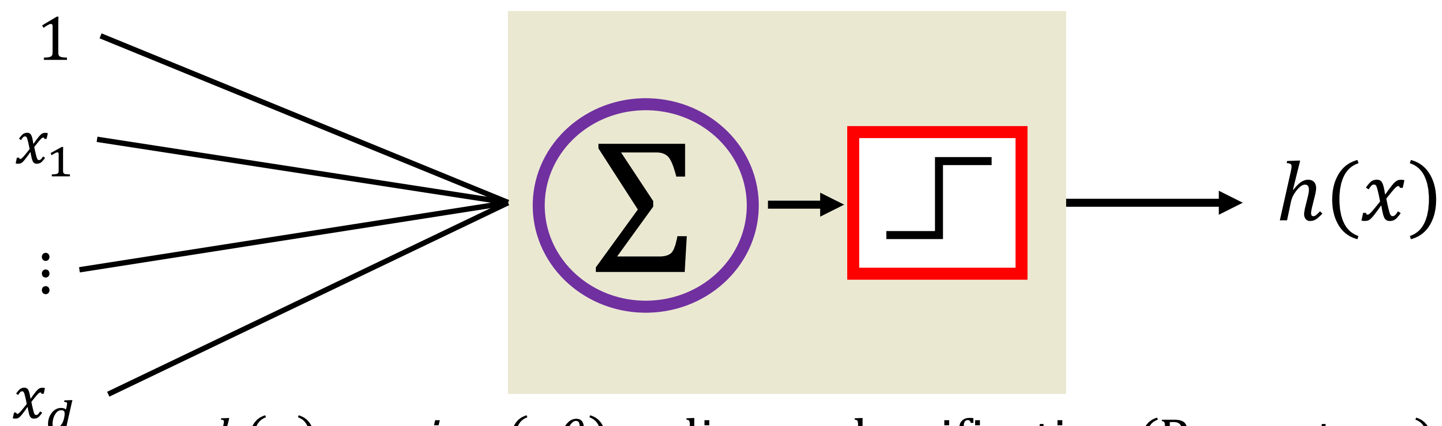


$h(x) = g(x\theta) \rightarrow$ logistic regression

Soft classification
Posterior probability

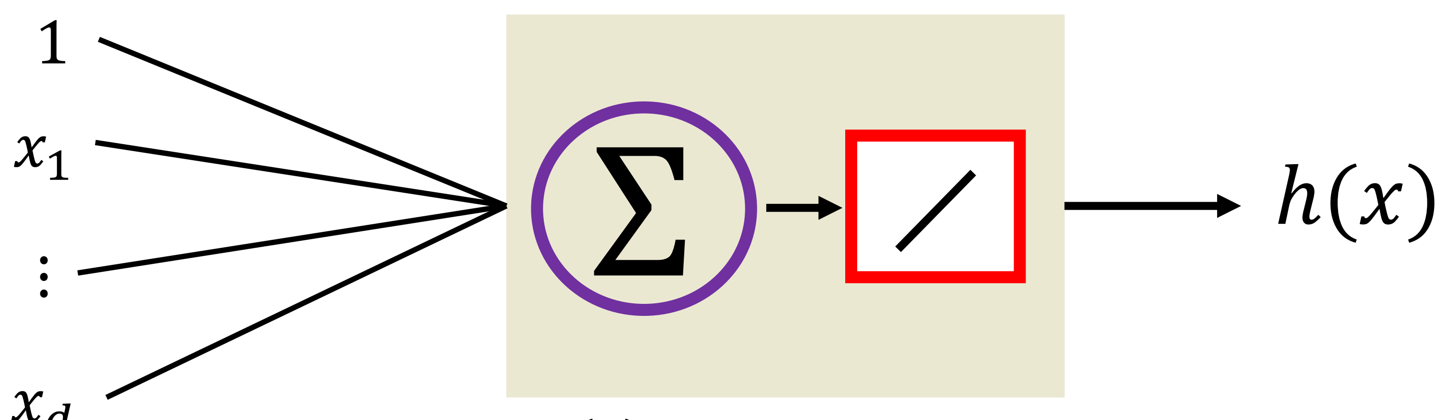
$$s = \sum_{i=0}^{d} x_i \theta_i = \theta_0 + \theta_1 x_1 + \cdots + \theta_d x_d$$

Three linear models

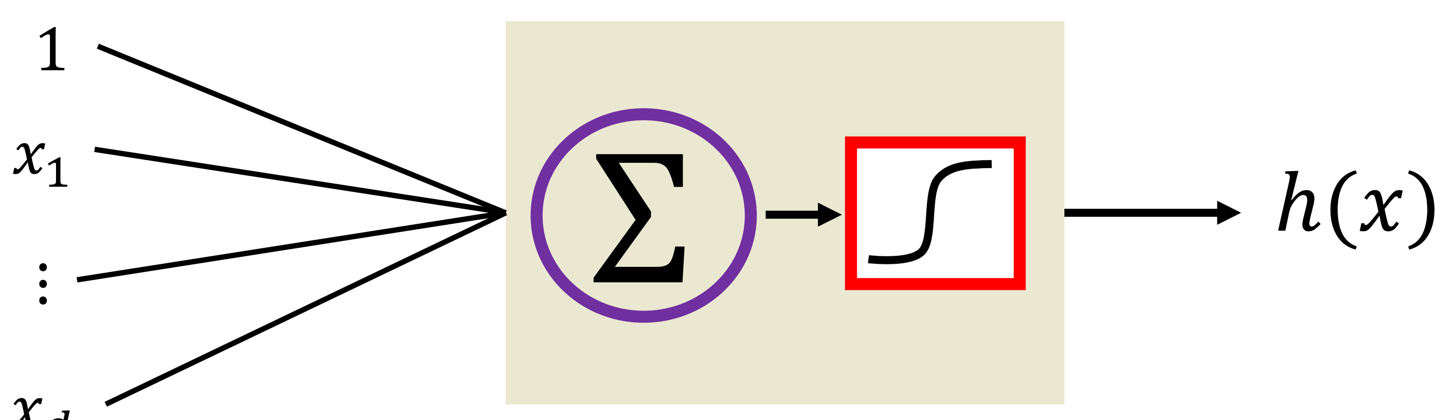


$h(x) = sign(x\theta) \rightarrow$ linear classification (Perceptron)

Hard classification



$h(x) \rightarrow$ linear regression



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

Soft classification

Posterior probability

$g(s)$ is interpreted as probability

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

We can't have a hard prediction here

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

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

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

Using posterior probability directly

Logistic regression model

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

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

$$l(\theta) := log \prod_{i=1}^{n \to \text{data points}} p(\underset{i\text{th class}}{y_i}, |\underset{i\text{th feature}}{x_i}, \theta)$$

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

[This form is concave, negative of this form is convex]()

The gradient of $l(\theta)$

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

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

$$\frac{d}{d\theta} \log(f(\theta)) = \frac{1}{f(\theta)} \cdot \frac{d f(\theta)}{d\theta}$$

- Gradient

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

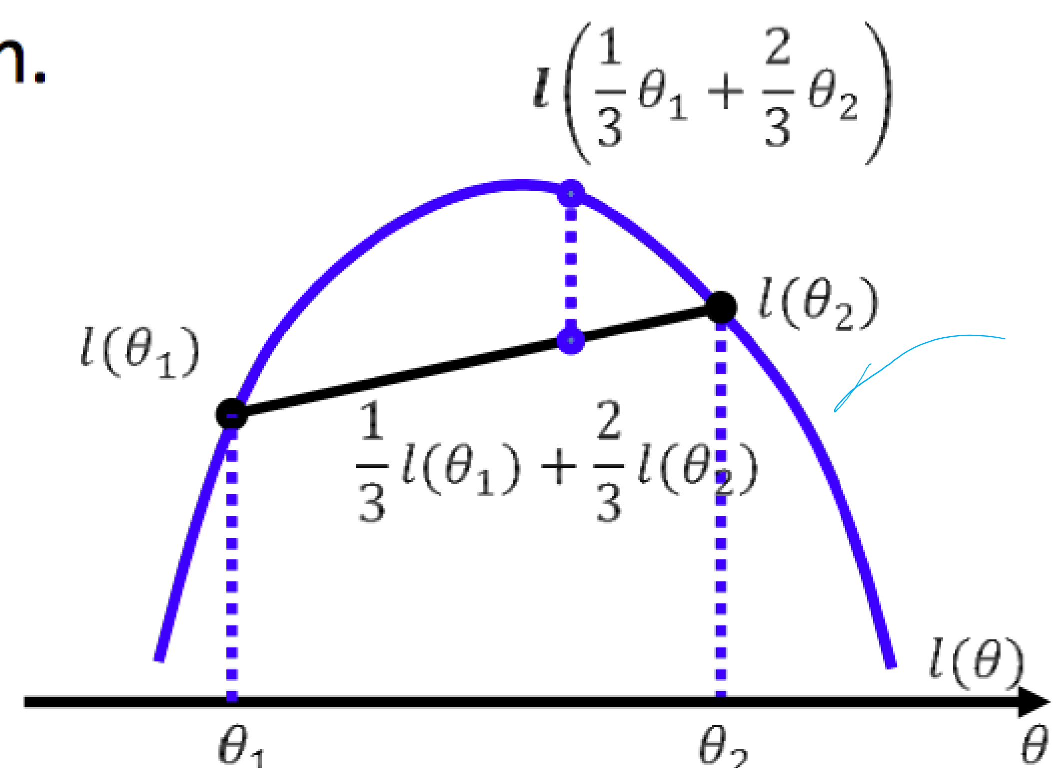
- Setting it to 0 does not lead to closed form solution

The Objective Function

- Find 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.



$a f(\theta_1) + b f(\theta_2) \leq$
$f(a\theta_1 + b\theta_2)$

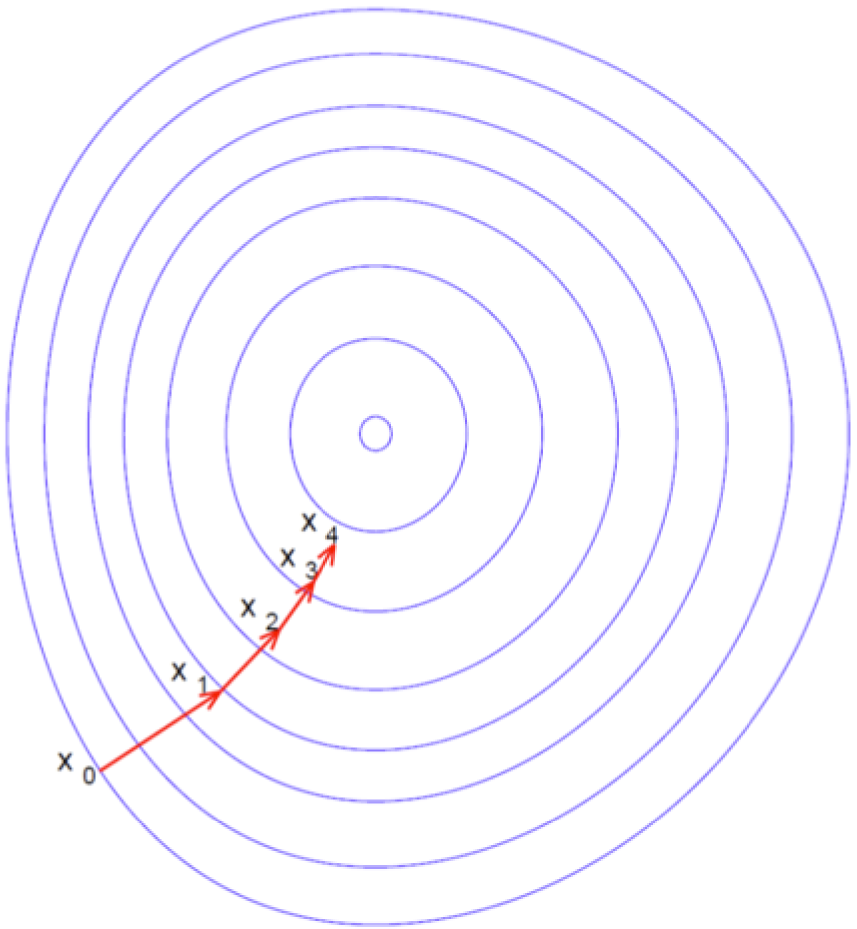
- Bad new: no closed form solution (resort to numerical method)

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



$\theta_{k+1} = \theta_k - \gamma_k \nabla f(\theta_k)$

- Update rule

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

γ_k is called the step size or learning rate

Gradient Ascent(concave)/Descent(convex) algorithm

- Initialize parameter θ^0

log liklihood of seeing n data points

$\frac{d\, l(\theta)}{d\theta}$

- Do

learning rate

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

params. for decision boundary.

we do not need:

P(x), P(y), P(x|y)

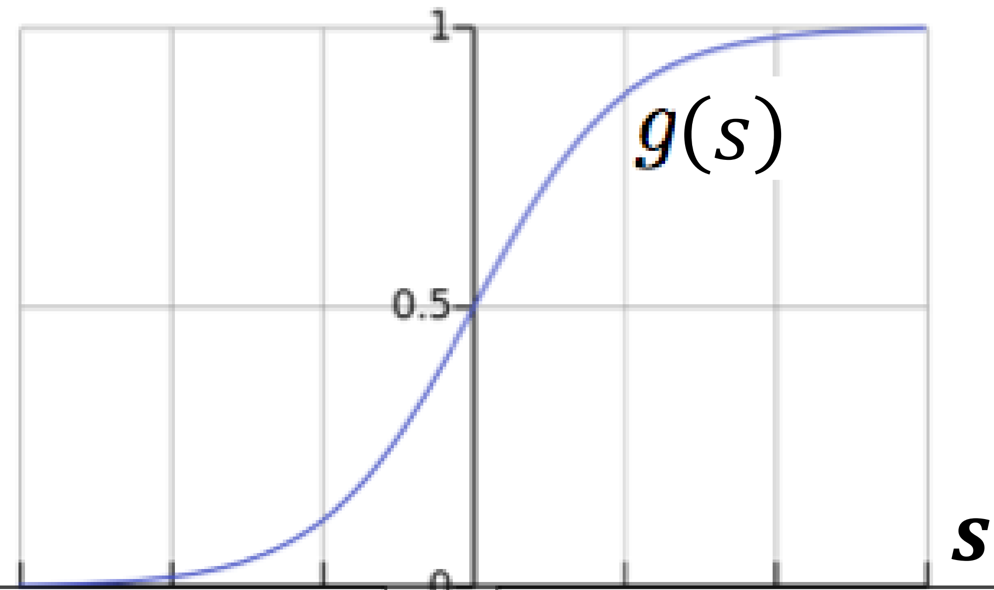
P(y|x)

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

Logistic Regression

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

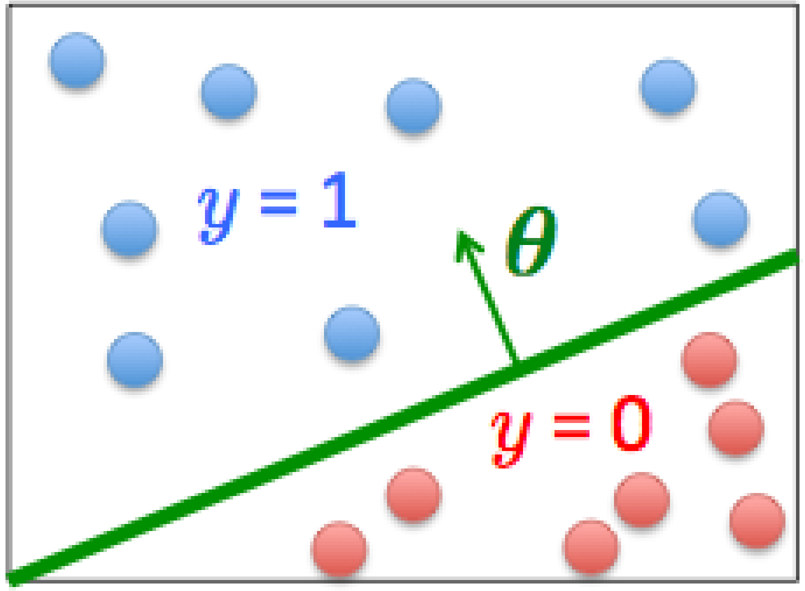
$$s = x\theta$$



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

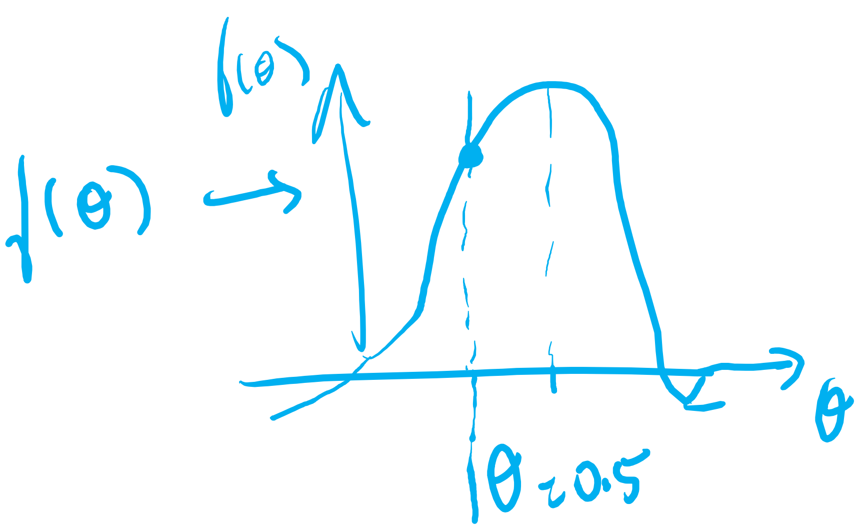
$x\theta$ should be large positive values for positive instances

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

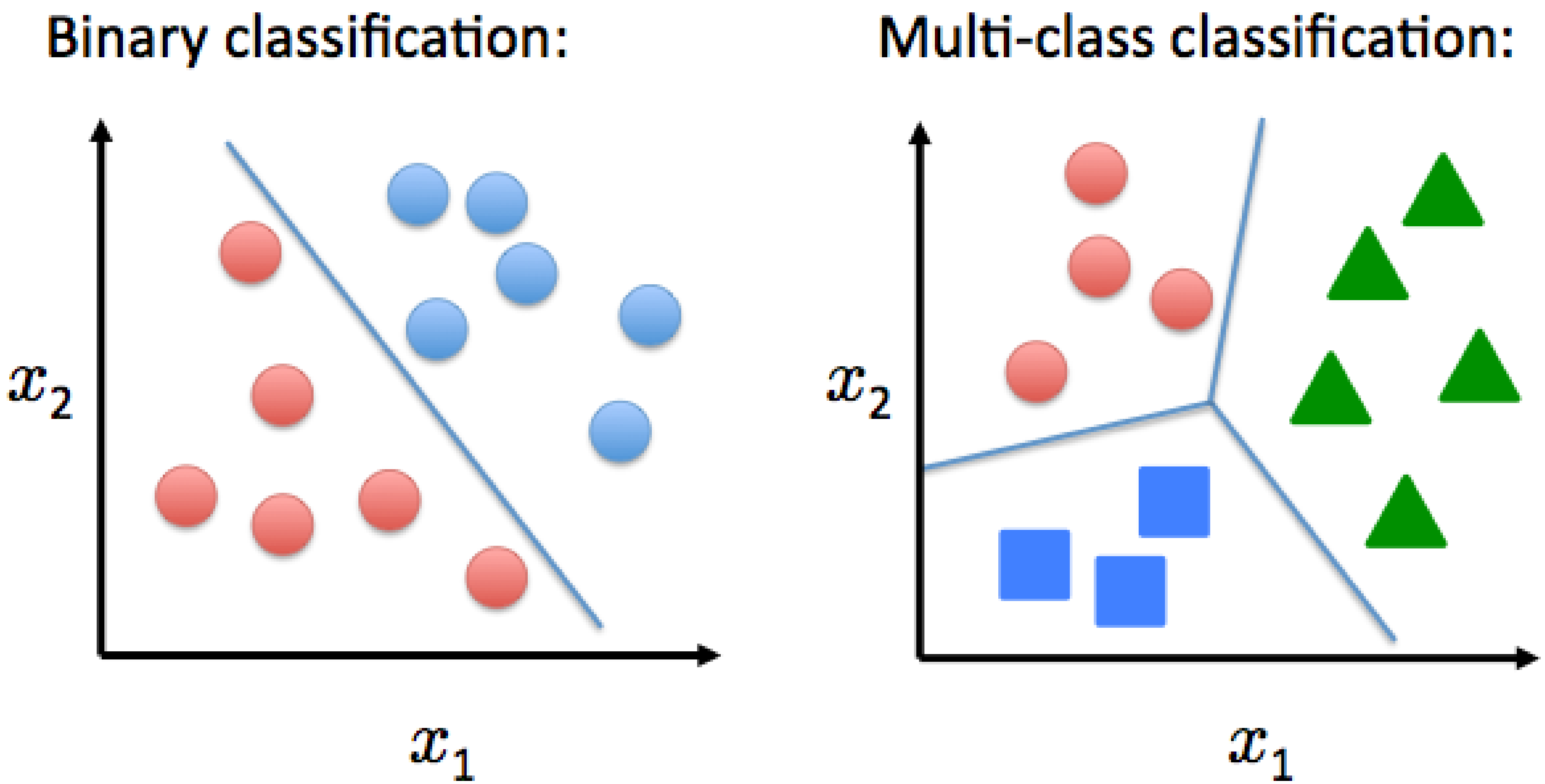


Outline

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



Multiclass Logistic Regression

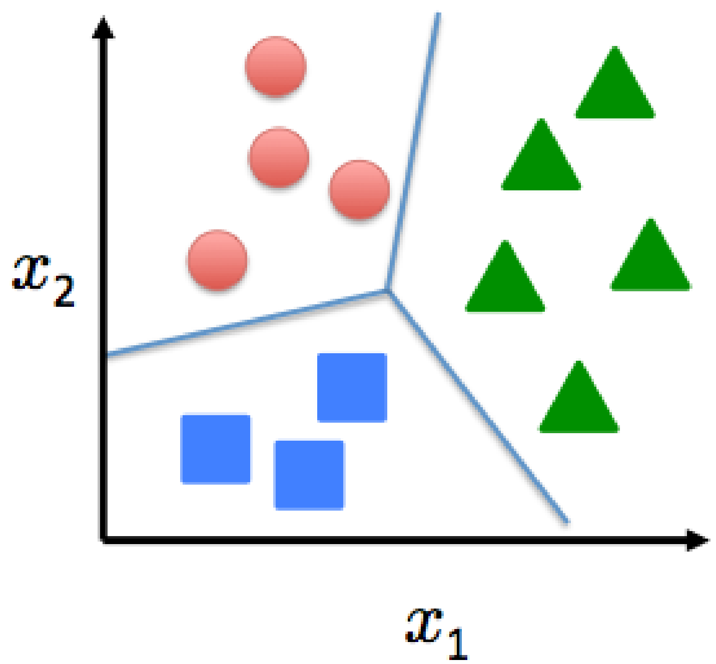


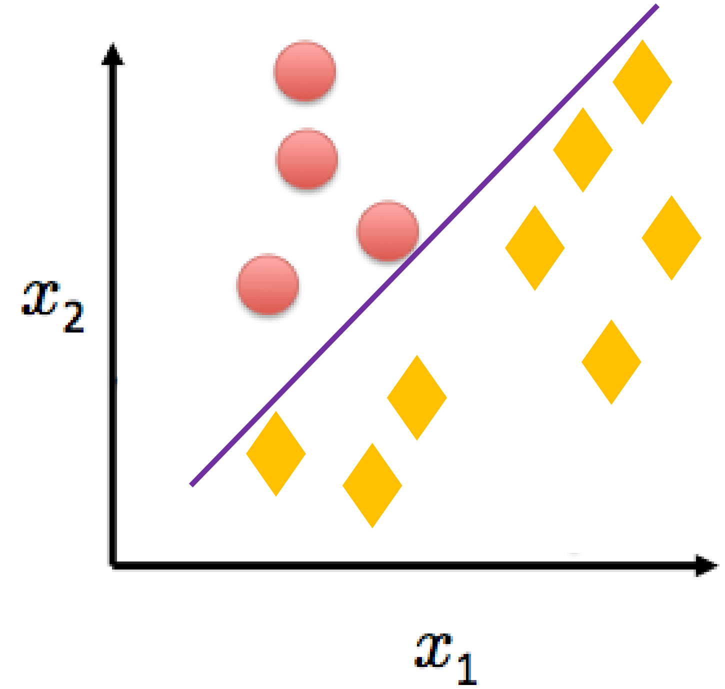
Disease diagnosis: healthy / cold / flu / pneumonia

Object classification: desk / chair / monitor / bookcase

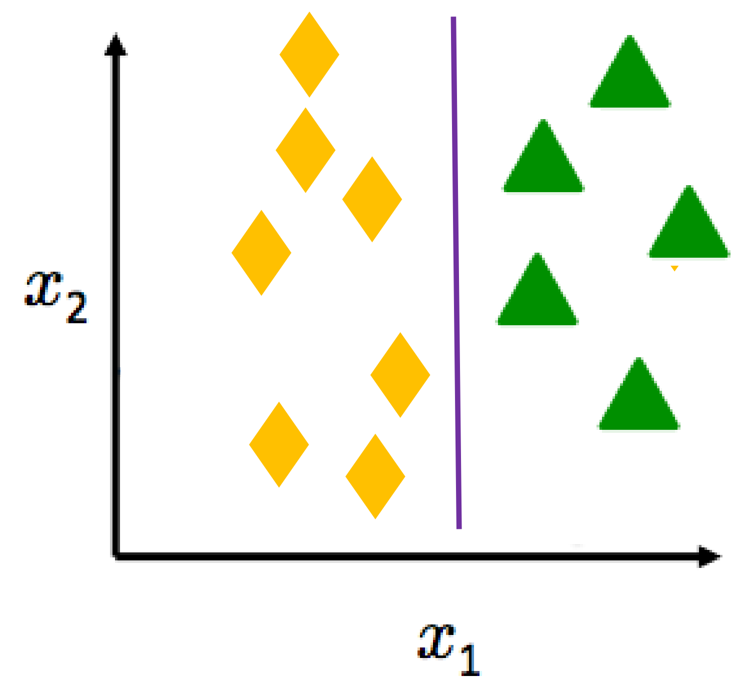
One-vs-all (one-vs-rest)

Multi-class classification:

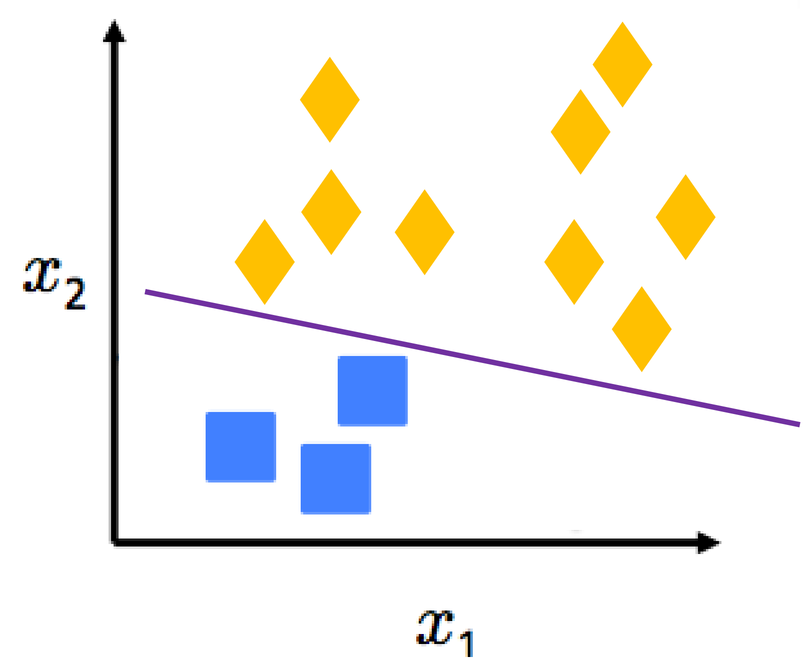




$h_\theta^1(x)$



$h_\theta^2(x)$



$h_\theta^3(x)$

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

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

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

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

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

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

Take-Home Messages

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