Ridge and Lasso Review

Linear combination of features $\hat{y} = X\Theta \Rightarrow \Theta = X^{-1} y$ Pseduo inverse

$$\hat{y} = \Theta_0 + \Theta_1 X_1 + \Theta_2 X_1^2 + \dots + \Theta_d X_1^d$$

$$y = \Theta_0 + \Theta_1 z_1 + \dots + \Theta_d z_d = Z\Theta$$

$\Theta = Z^{-1}\Theta$ → Pseduo inverse

Overfitting issue: Θ_{solution} for $Z\Theta$ is the overfitted solution:

$$E(\Theta) = L(\underset{\text{overfitted}}{\Theta}) = \frac{1}{N}\sum(y_a - Z\Theta)^2 = 0 = \text{bias}^2 + \text{variance}$$

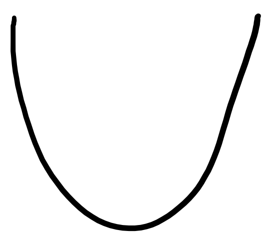
Ridge and Lasso Review

Regularization term

$\|\theta\|_2^2$

$$\text{Min } E(\theta) = \frac{1}{N}\sum_{i=1}^{N}(y_a - \hat{y}_p)^2$$

$$\text{s.t. } \theta^T\theta = C$$



$$\Rightarrow \boxed{\tilde{E}(\theta) = E(\theta) + \frac{\delta}{2N}\|\theta\|_2^2}$$

hyper Parameter ↓ Cross-Validation

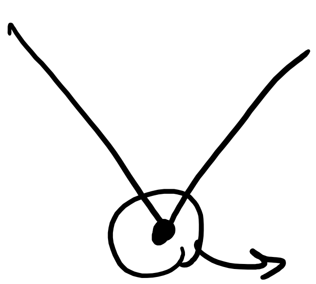
L_2 Regularize ↓ Ridge

$$\theta = (Z^TZ + \delta I)^{-1} Z^T y$$

$\|\theta\|_1 \Rightarrow$ LASSO L_1 Regularizer

$$\tilde{E}(\theta) = E(\theta) + \delta\|\theta\|_1$$

↓



→ Sign function

⇝ Sub-gradient ⇝ $\boxed{\text{feature selection}}$ → LASSO

Naïve Bayes and Logistic Regression

Mahdi Roozbahani
Georgia Tech

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

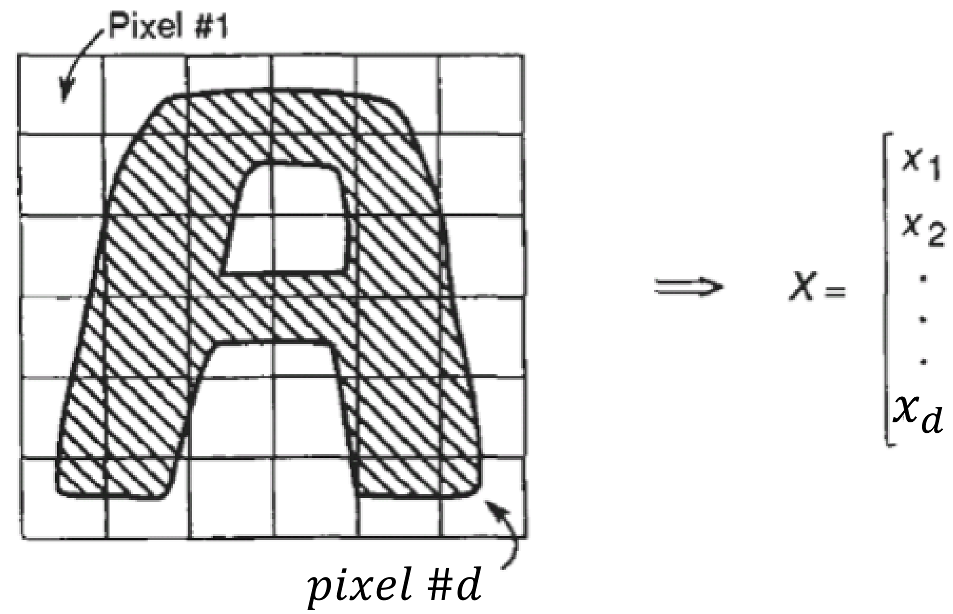

THE BEST WAY TO
EXPLAIN OVERFITTING

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

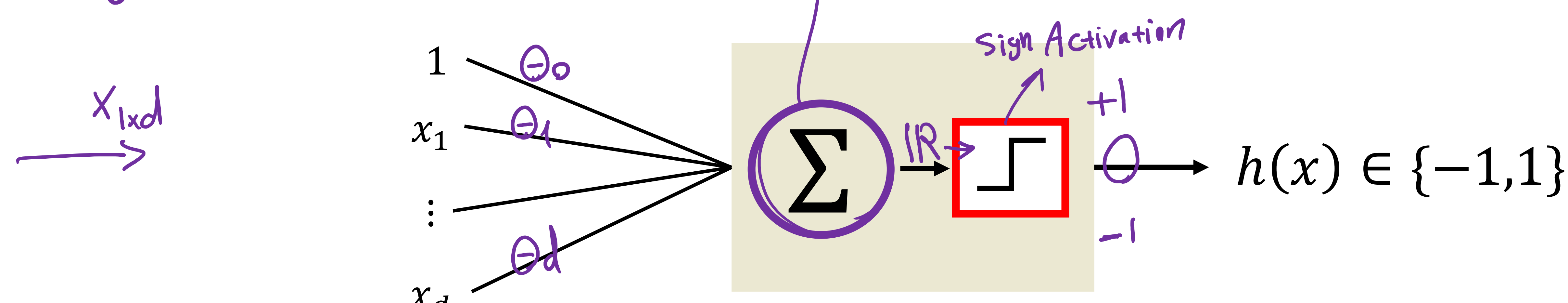


- A label is provided for each data point, eg., $y \in \{-1, +1\}$

$\text{Sum} = S = \Sigma = \theta_0 \times 1 + \theta_1 \times x_1 + \dots + \theta_d x_d = x\theta \in \mathbb{R}$

Learning block

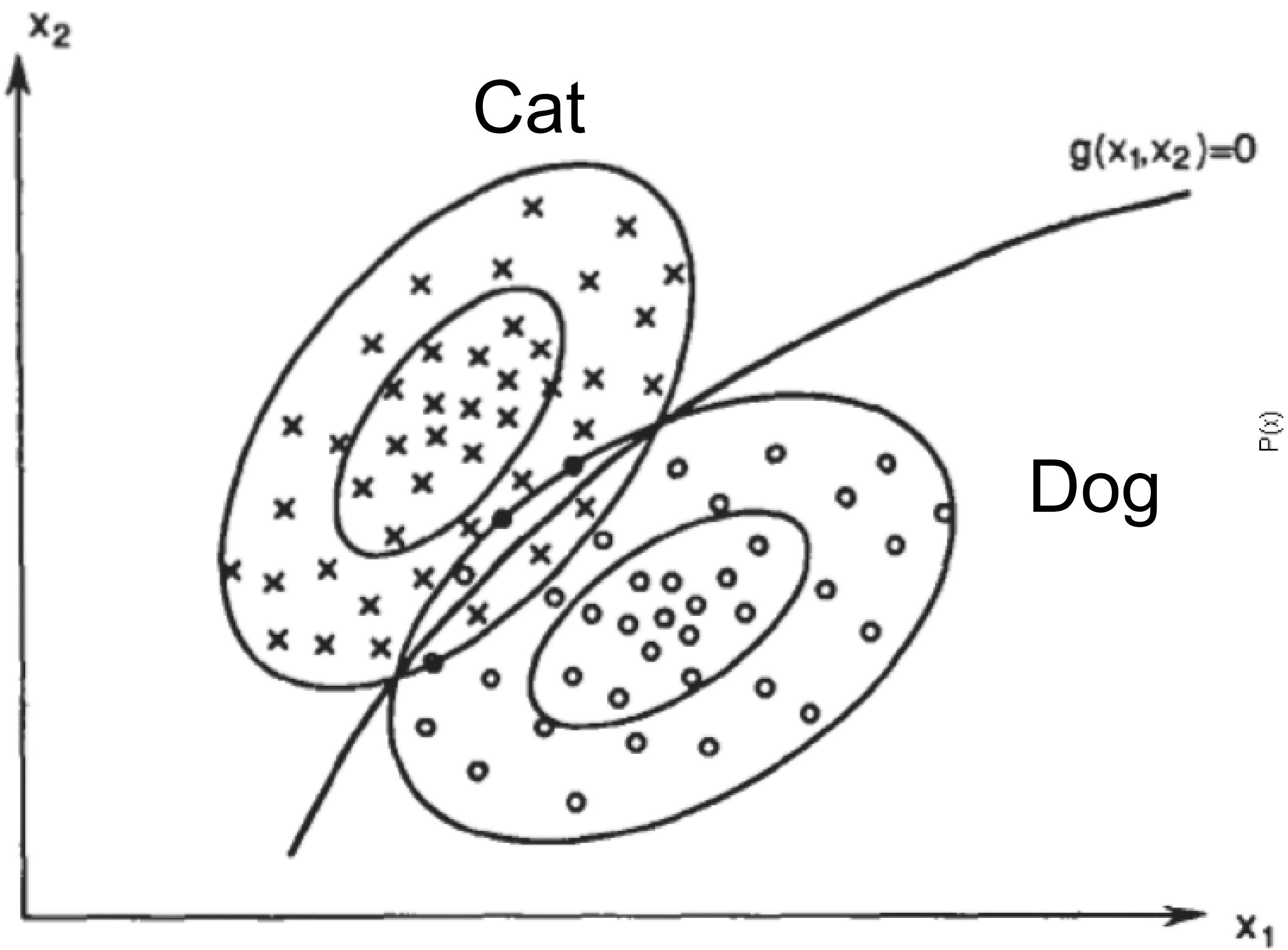
- Classifier

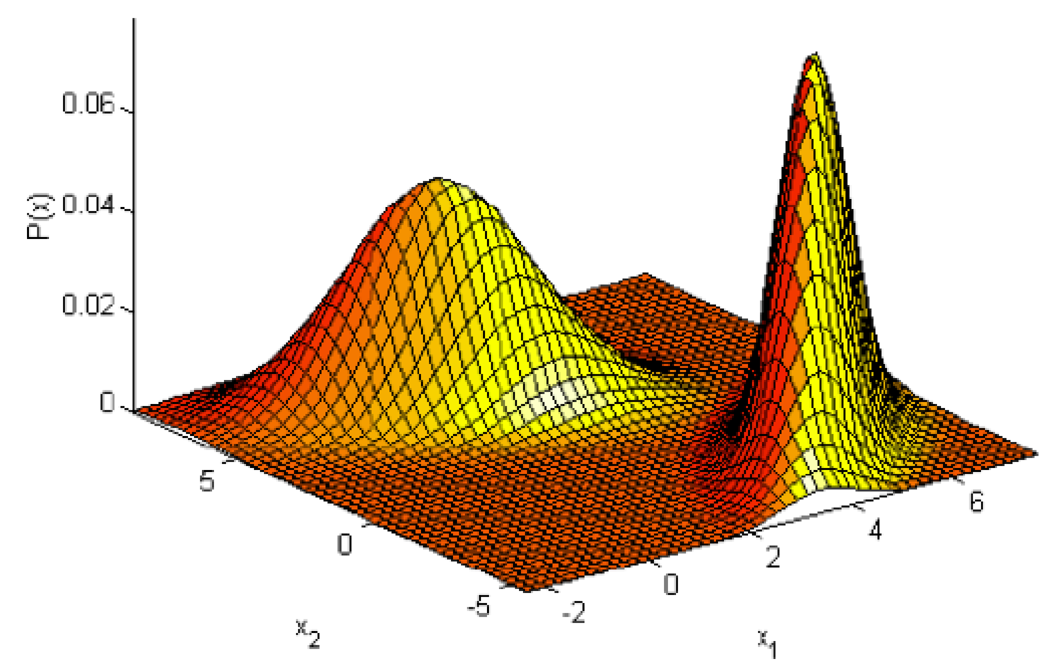


$h(x) = \; sign(x\theta) \rightarrow$ linear classification (Perceptron)

Decision Making: Dividing the Feature Space

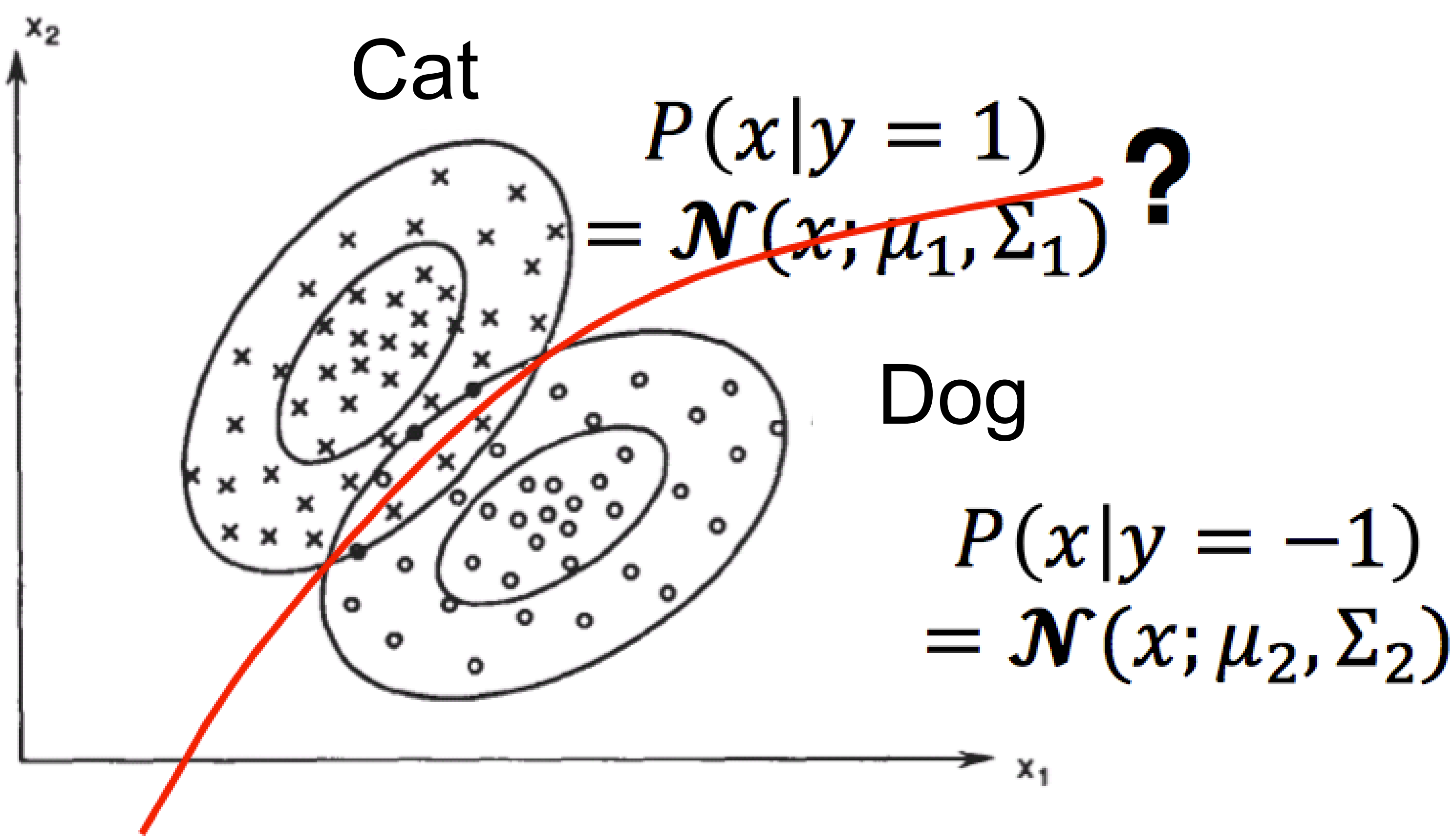
- Distributions of sample from normal (positive class) and abnormal (negative class) tissues





How to Determine the Decision Boundary?

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



Bayes Decision Rule

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

$y=1 \Rightarrow P(\cancel{x|y=1}) \cancel{P(y=1)}$

$P(y=1|x)$

$P(y=-1|x)$

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

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

$$P(y=1 \mid x) = \frac{\overbrace{P(x \mid y=1)}^{\text{Likelihood}} \; \overbrace{P(y=1)}^{\text{prior}}}{P(x) = \sum_{y \in \{-1,+1\}} P(x, y=y) = P(x, y=1) + P(x, y=-1)}$$

$$x_{1 \times d}$$

$$P(y=-1 \mid x) = \frac{P(x \mid y=-1)\, P(y=-1)}{P(x) = P(x, y=1) + P(x, y=-1)}$$

$$P(x \mid y=1) = \underbrace{P(x_1, x_2, \dots, x_d \mid y=1)}_{\text{Conditional independence}} = N(x \mid \mu, \Sigma) \qquad C = \big[\quad\big]_{d \times d}$$

Conditional independence

$$\underbrace{P(x_1 \mid y=1)}_{N(x \mid \mu, \sigma)} \; P(x_2 \mid y=1) \cdots P(x_d \mid y=1)$$

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 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:

$P(x_1 | y=1)$

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

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

$$= \cdots$$

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

Naïve Bayes assumption: let’s rewrite it as:

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

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

Gaussian naïve Bayes
A typical assumption

[Example]()

Discriminative Models

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

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

$= \sum_{y \in \{-1,+1\}} P(x, y=y)$

$P(x, y=1) + P(x, y=-1)$

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

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

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

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

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

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

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

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

$$\frac{a}{a+b} \quad \frac{a/a}{\frac{a+b}{a}} \quad \frac{1}{1+\frac{b}{a}}$$

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

constant

$$= X\theta$$

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

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

$$P(y=1|x) = \frac{P(x|y=1)P(y=1)}{P(x)}$$

Number of parameters:

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

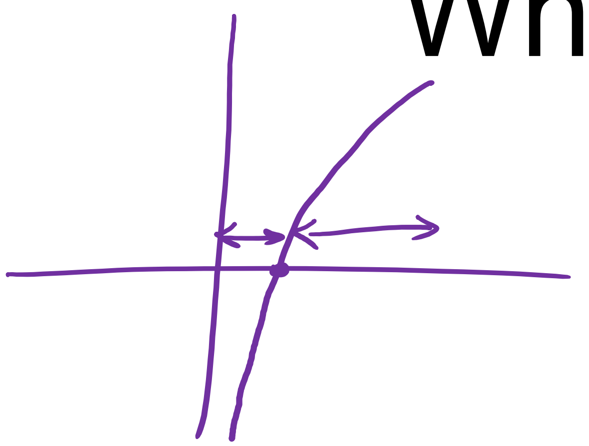
$$P(y=1|x) = \frac{1}{1+\exp[-(\sum_{i=1}^{d}(\theta_i x_i)+\theta_0)]} = \frac{1}{1+\exp(-s)}$$

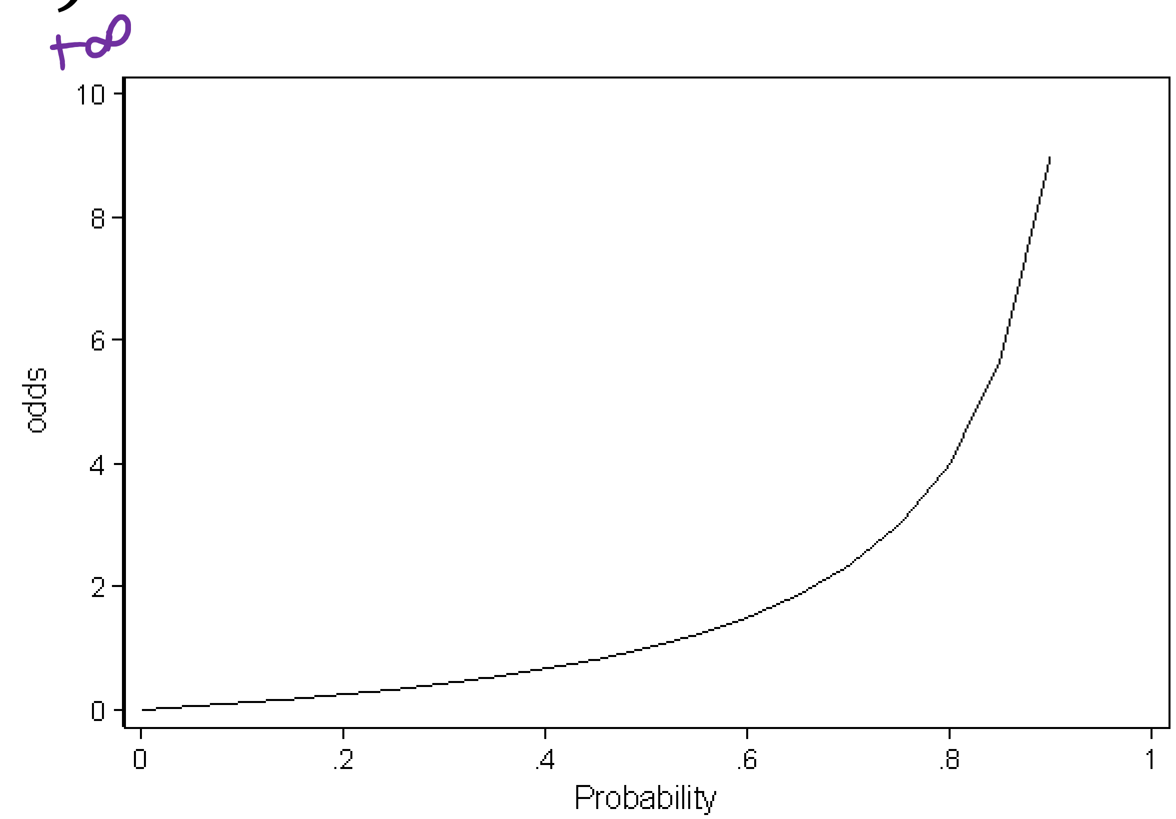
Number of parameters = $d+1 \rightarrow \theta_0, \theta_1, \theta_2, \ldots, \theta_d$

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

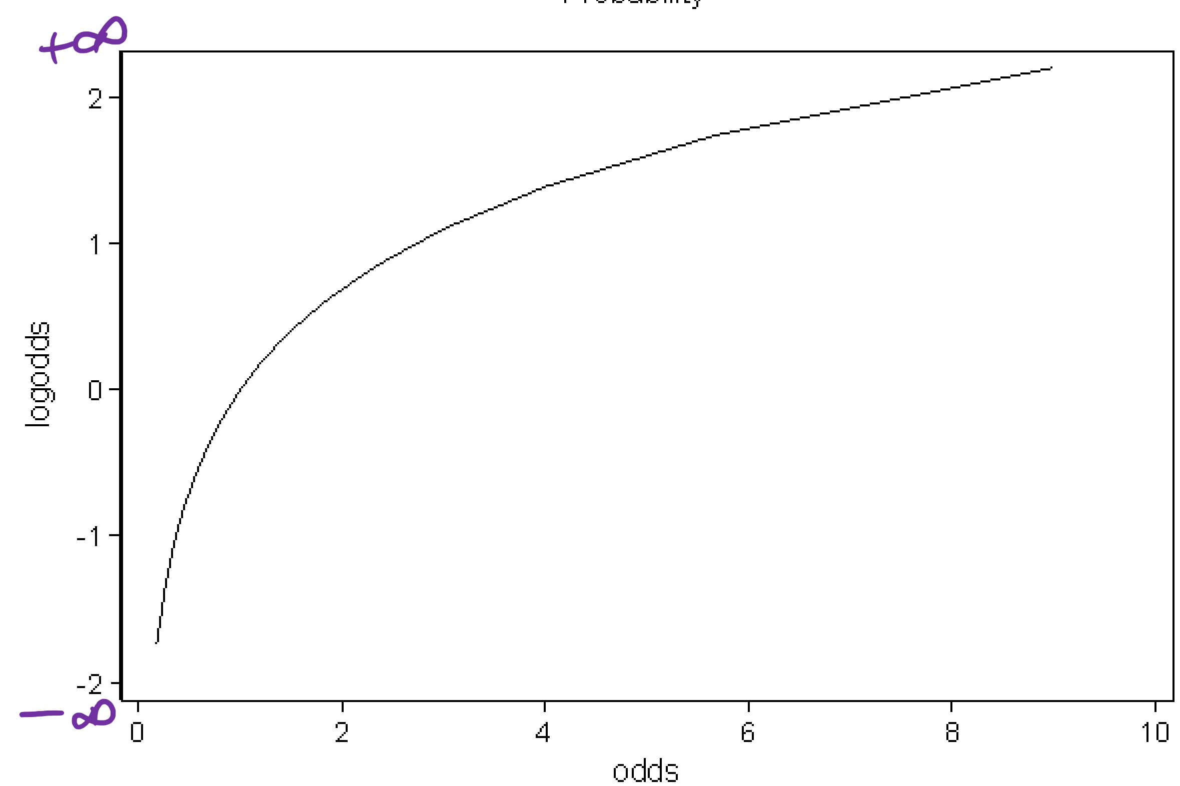
Gaussian Naïve Bayes is a subset of logistic regression

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?

$\log(odds) \in \mathbb{R}$ $\quad x\theta \in \mathbb{R}$

$x\theta = \log(odds)$

What is logit function?

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

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

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

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

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

Sigmoid

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

Logistic function for posterior probability

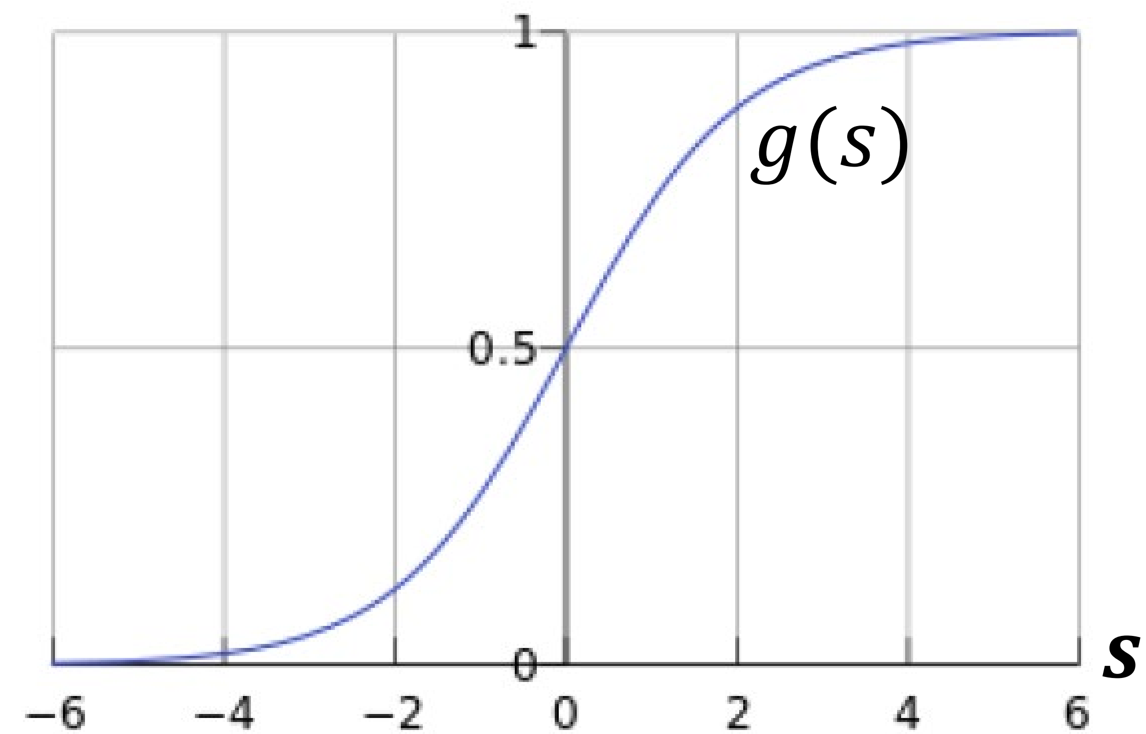
Many equations can give us this shape

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

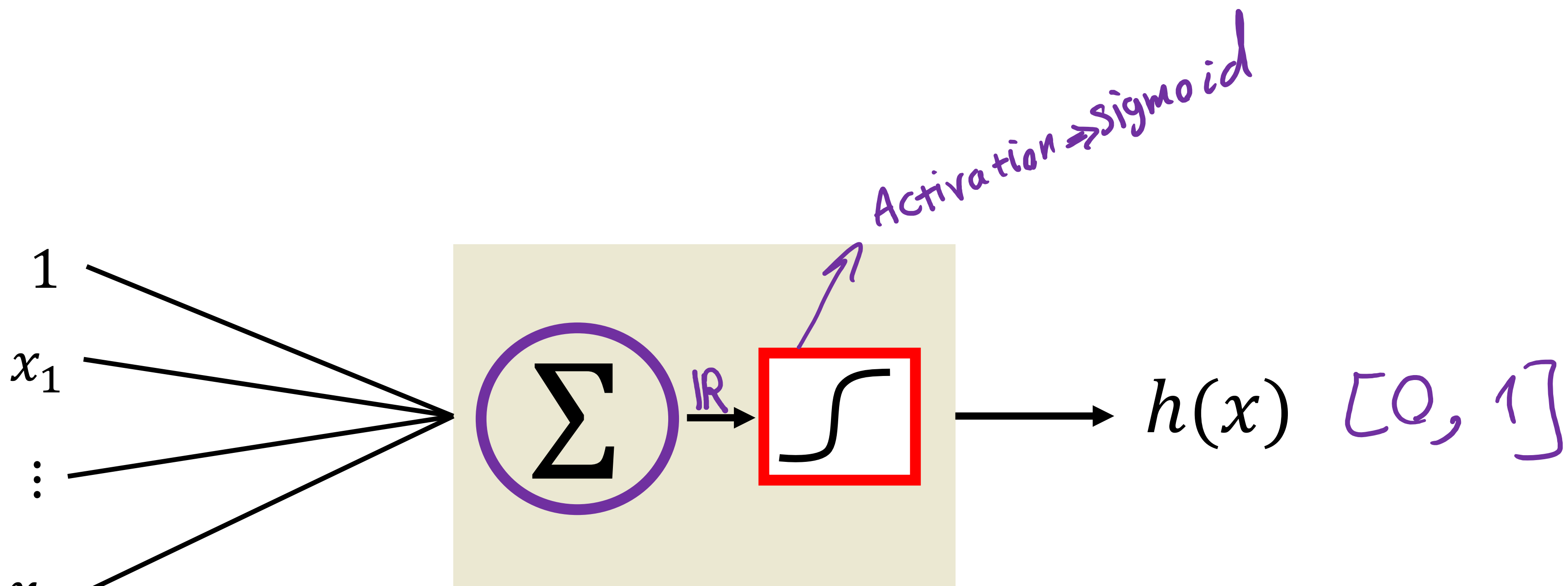
Is 0.5 threshold cut-off a good choice?

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

Sigmoid Function

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

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

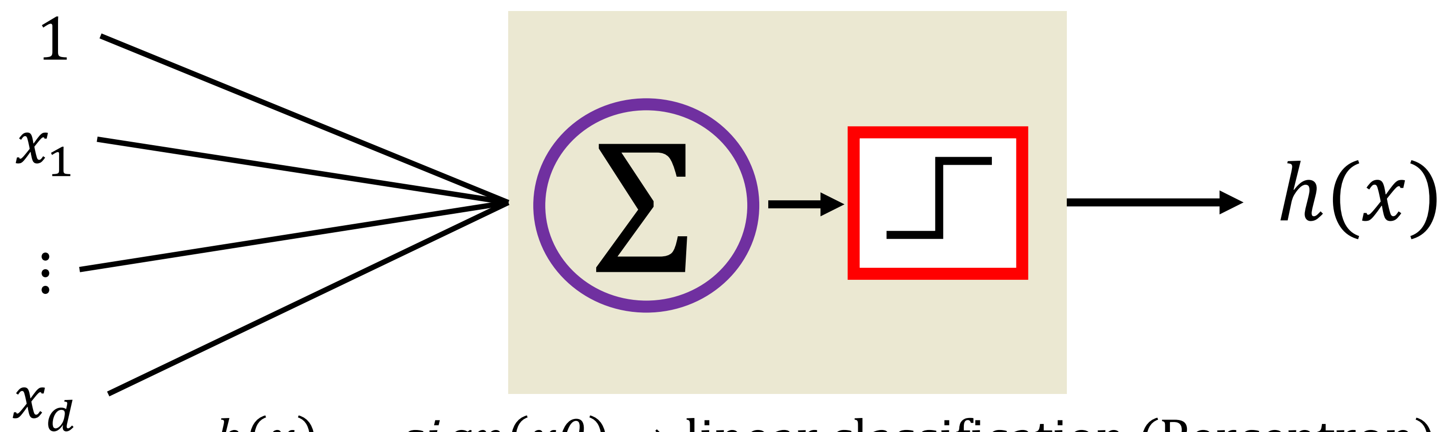


Soft classification
Posterior probability

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

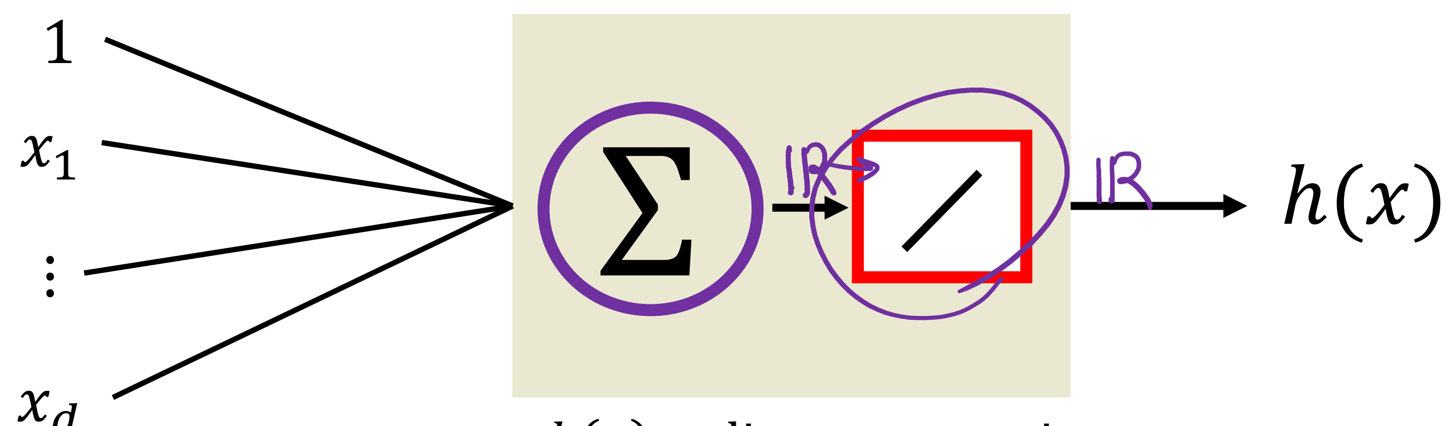
Three linear models

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

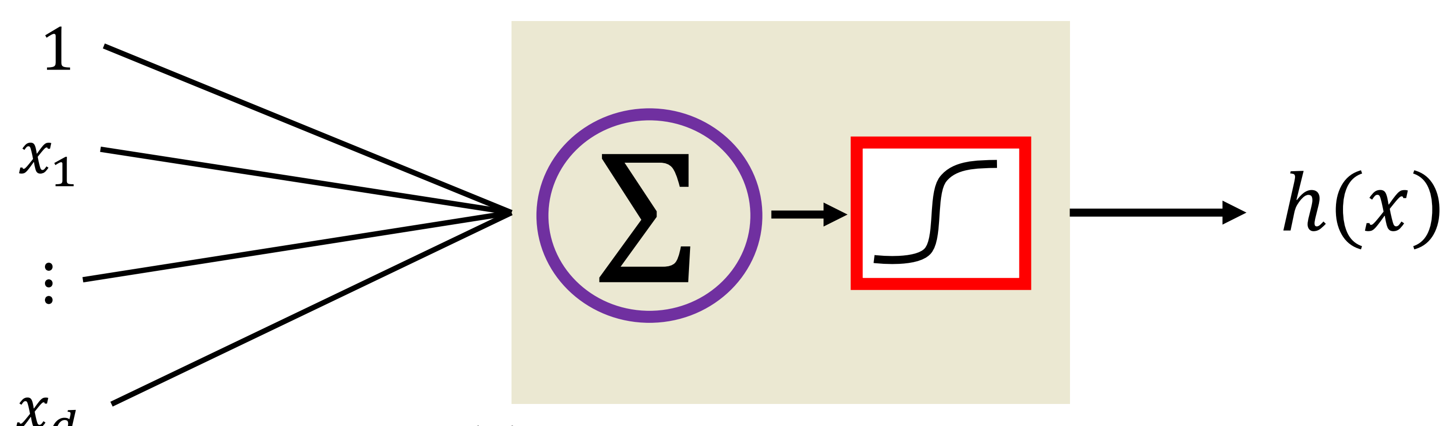


Hard classification

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



$h(x) \rightarrow$ linear regression



Soft classification
Posterior probability

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

$g(s)$ 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

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

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

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

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

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

[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

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

$$\theta^{\{t+1\}} \leftarrow \theta^{\{t\}} + \alpha \frac{\partial l(\theta)}{\partial \theta}$$

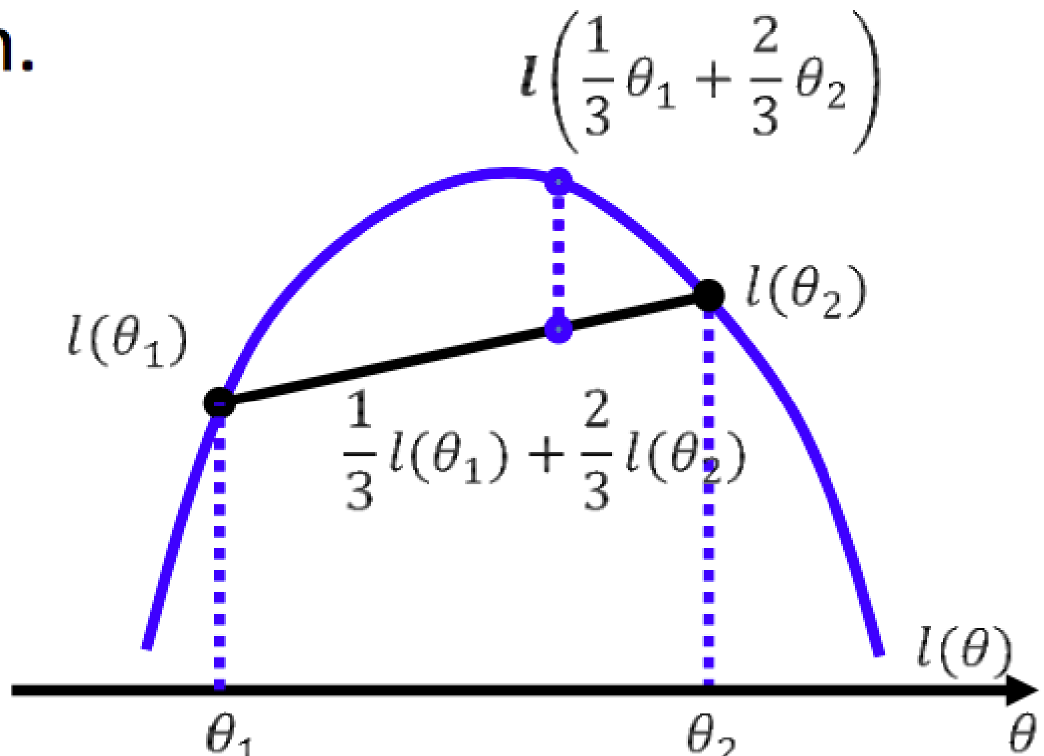
- Setting it to 0 does not lead to closed form solution

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.



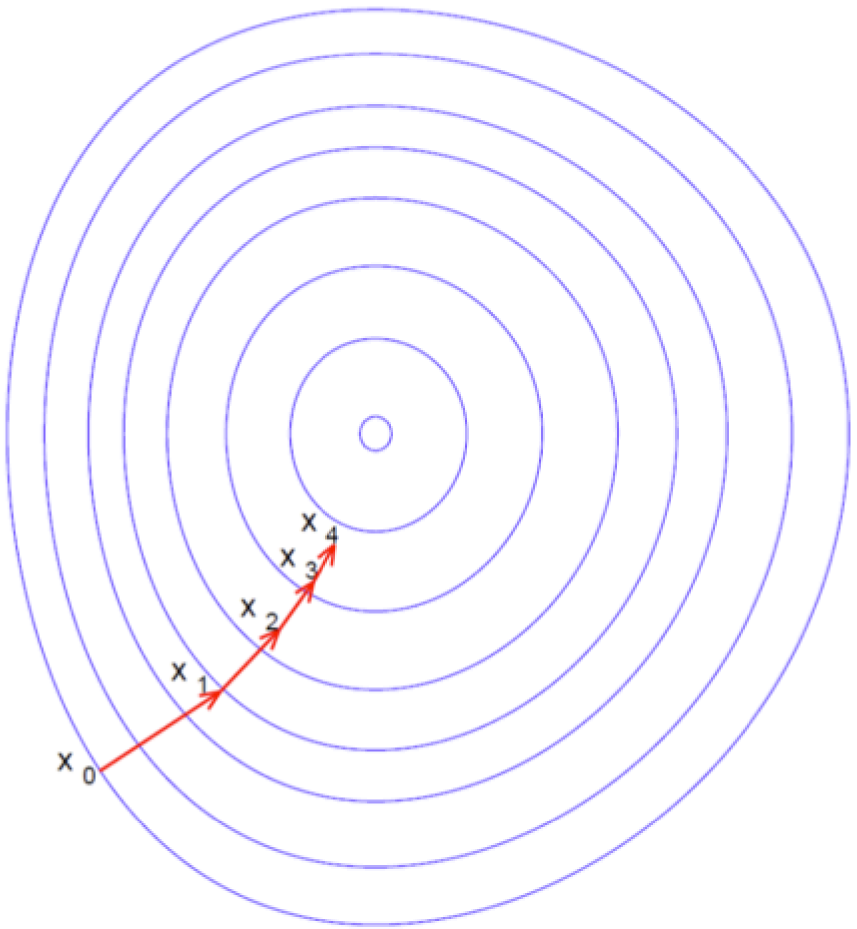
- Bad new: no closed form solution (resort to numerical method)

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



Gradient Ascent(concave)/Descent(convex) algorithm

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

Test data point

- Initialize parameter θ^0
- Do

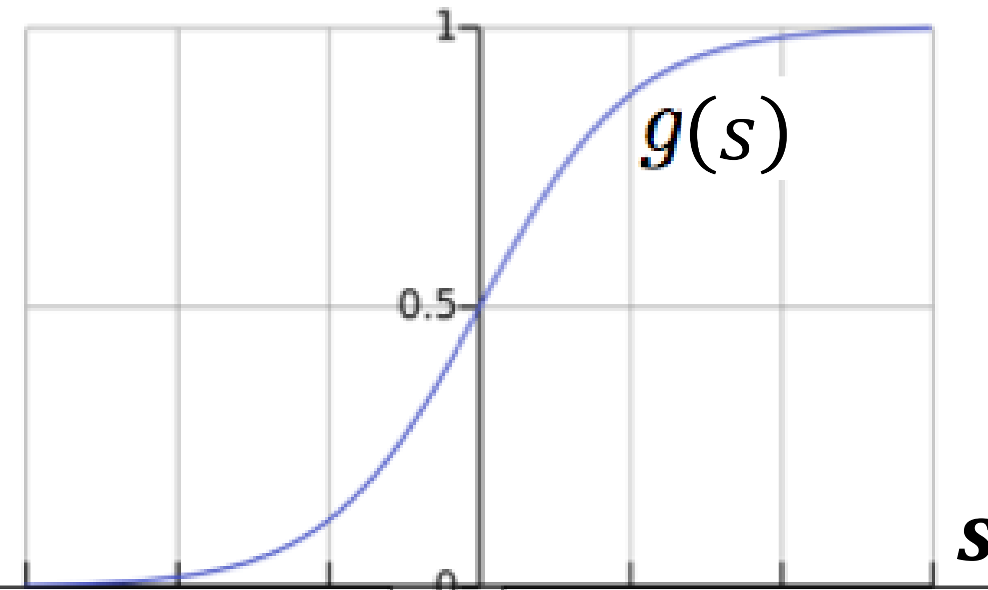
$$\theta^{t+1} \leftarrow \theta^t + \eta \sum_i x_i^T (y_i - 1) + x_i^T \frac{\exp(-x_i\theta)}{1+\exp(-x_i\theta)}$$

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

Logistic Regression

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

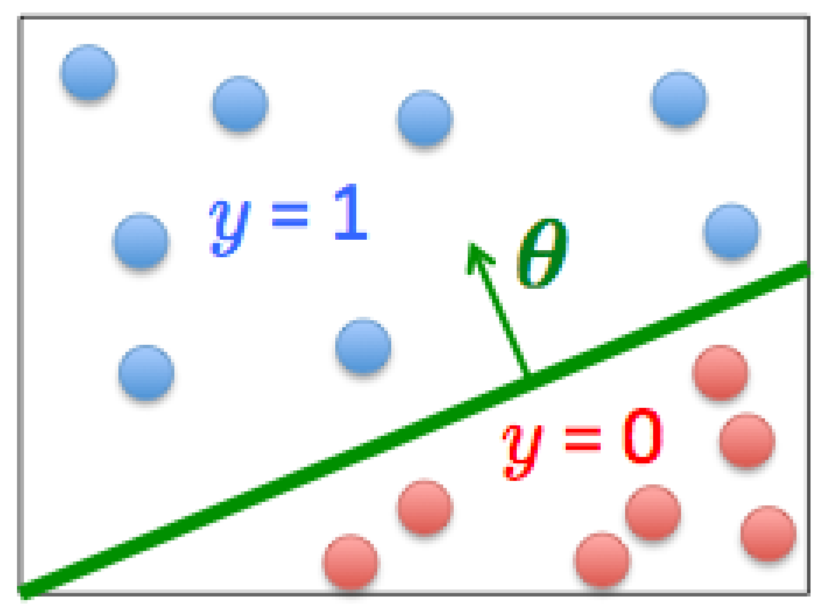
$$s = x\theta$$



$x\theta$ should be large <u>negative</u> values for negative instances

$x\theta$ should be large <u>positive</u> values for positive instances

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

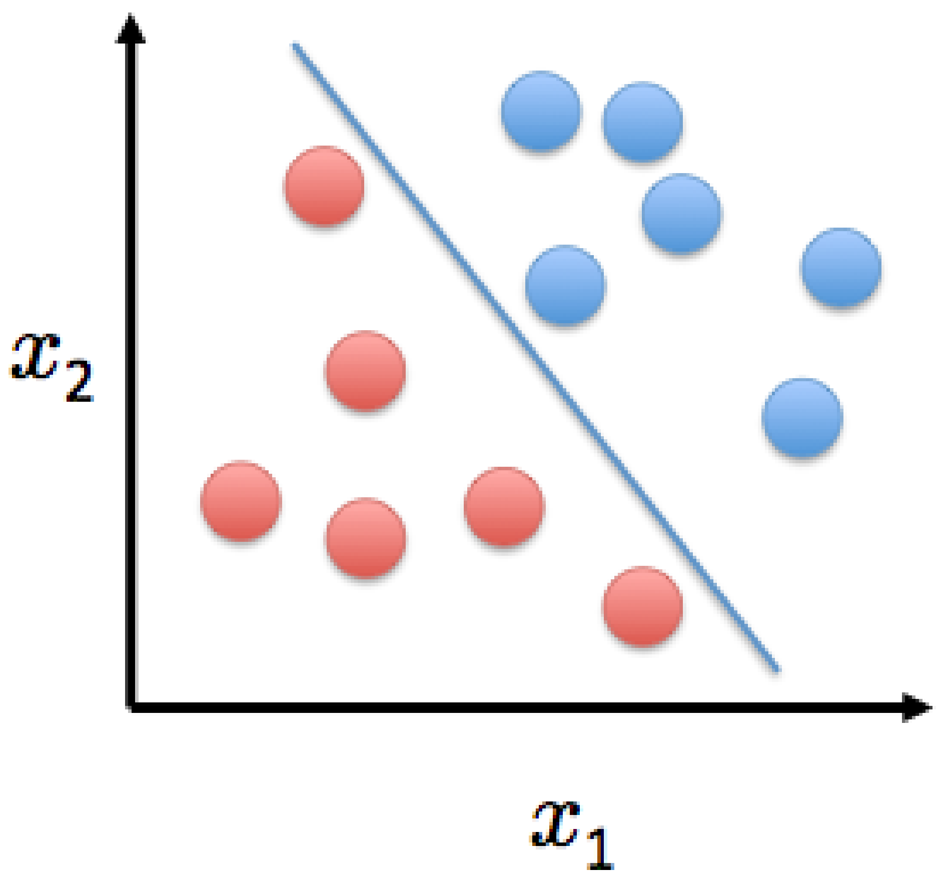


Outline

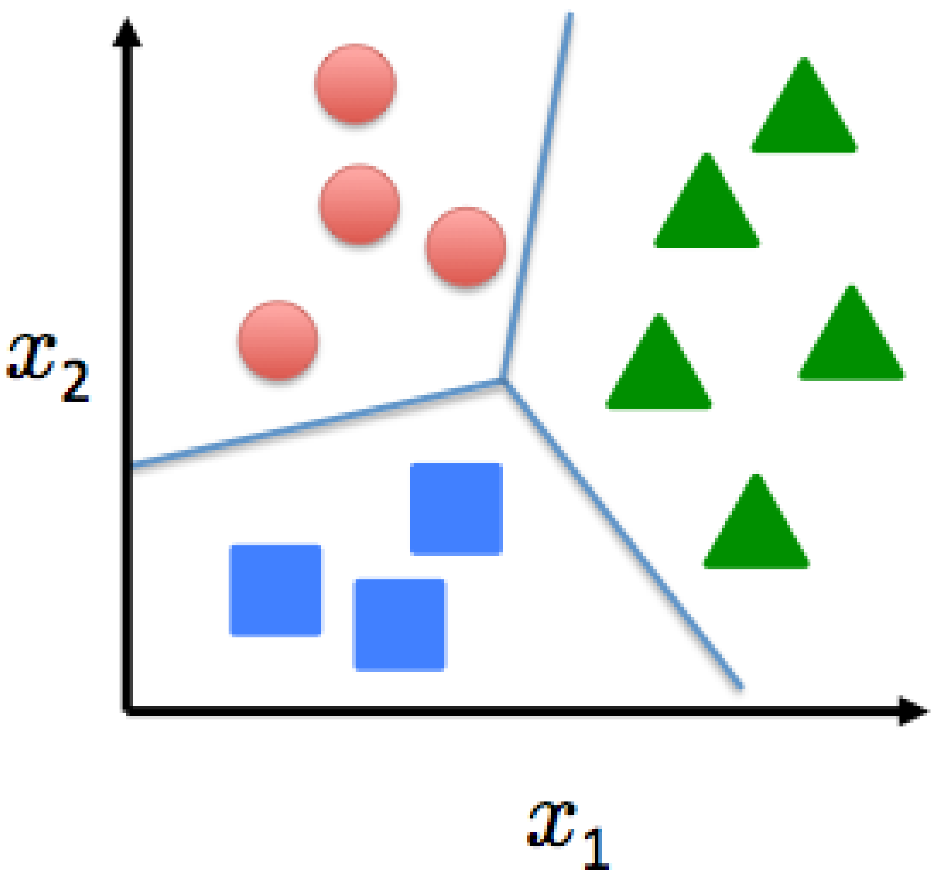
- Generative and Discriminative Classification
- The Logistic Regression Model
- Understanding the Objective Function
- Gradient Descent for Parameter Learning
- Multiclass Logistic Regression

Multiclass Logistic Regression

Binary classification:



Multi-class classification:

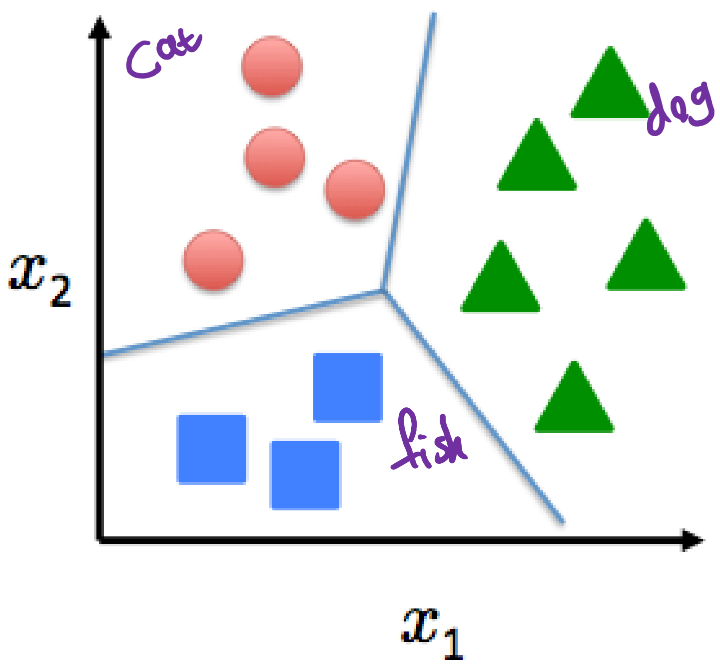


Disease diagnosis: healthy / cold / flu / pneumonia

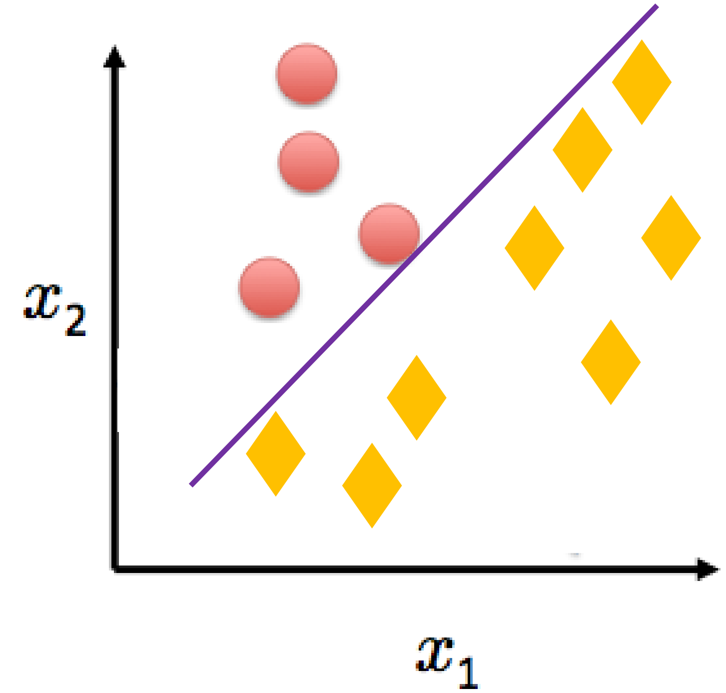
Object classification: desk / chair / monitor / bookcase

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

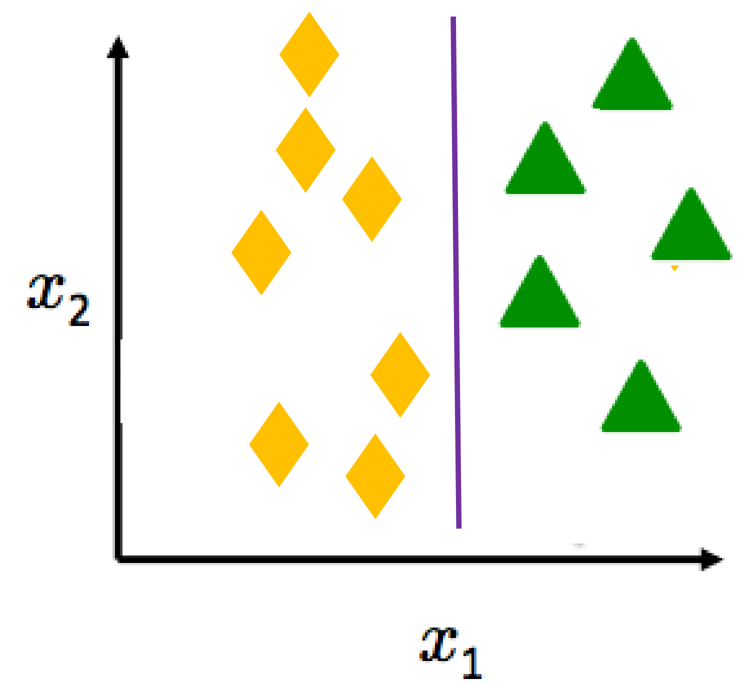
Multi-class classification:



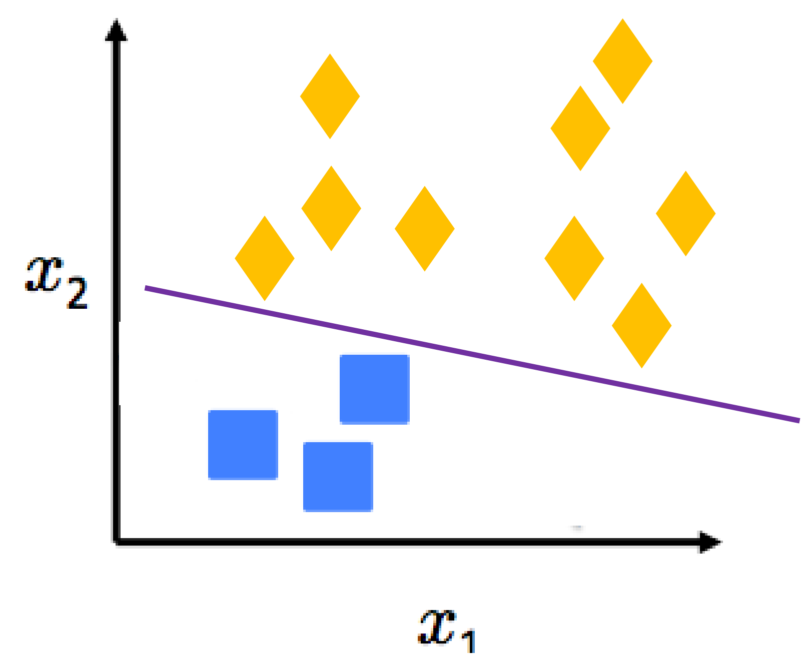
$$h_\theta^{(m)}(x) = p(y = 1|x, \theta) \ (m = 1,2,3)$$



$h_\theta^1(x) = \frac{1}{1+\exp(-x\theta)}$



$h_\theta^2(x)$



$h_\theta^3(x)$

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

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

$\Theta_{d \times k} =$

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

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

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

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

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

$d \times k$

Take-Home Messages

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