

CS 559: Linear Classification II & Kernel Method I

Lecture 4
Spring 2020

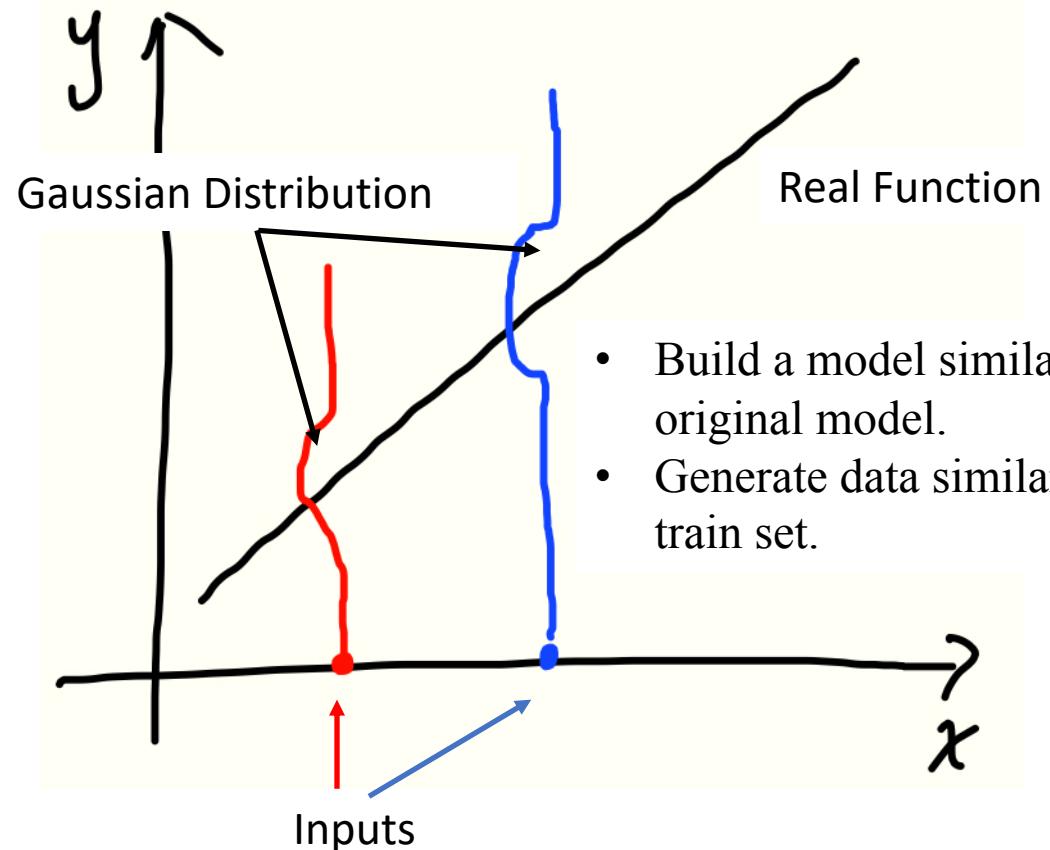
Outline

- Lecture 4-1: Linear Classification & Generalization
 - Probabilistic Generative Models
 - Probabilistic Discriminative Models (logistic regression)
 - Generalization
- Lecture 4-2: Kernel Methods
 - Dual Representations
 - Constructing Kernels

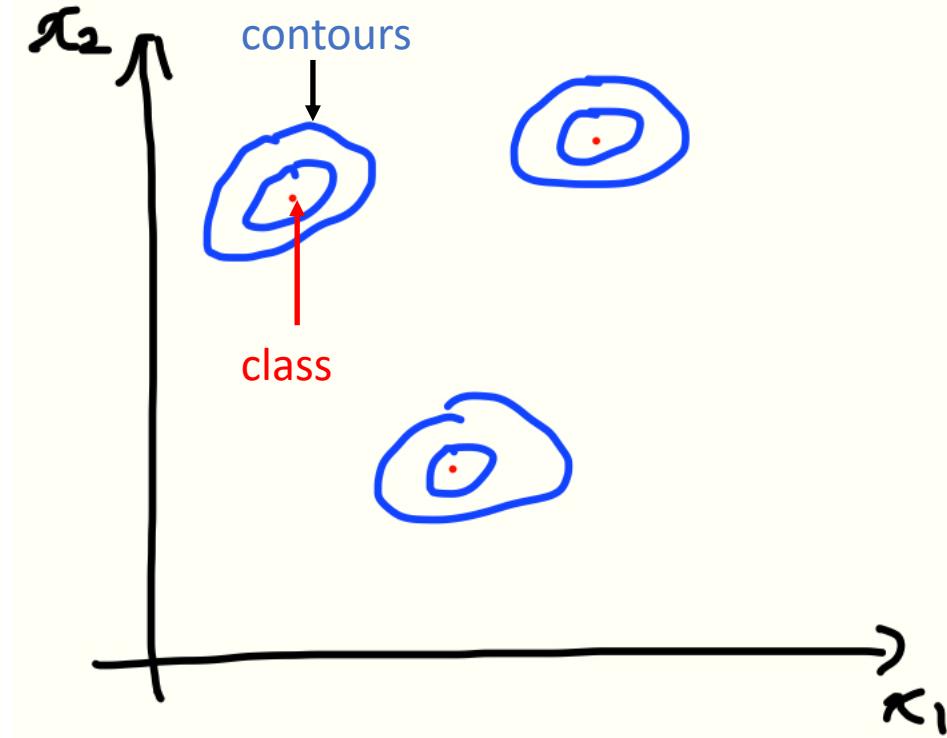
Linear Classification II

Probabilistic Generative Models

- We now turn to a probabilistic approach to classification.
- How models with linear decision boundaries arise from simple assumptions about the distribution of the data.



- Build a model similar to the original model.
- Generate data similar to the train set.



Generative Approach

- Infer the prior class probabilities $p(C_k)$.
- Solve the inference problem of estimating the class-conditional densities $p(\mathbf{x}|C_k)$ for each class C_k .
- Use Bayes' theorem to find the class posterior probabilities:

$$p(C_k|\mathbf{x}) = \frac{p(\mathbf{x}|C_k)p(C_k)}{p(\mathbf{x})} \quad (1)$$

where

$$p(\mathbf{x}) = \sum_k p(\mathbf{x}|C_k)p(C_k) \quad (2)$$

- Use decision theory to determine class membership for each new input \mathbf{x} .

Assumptions

- In classification, the number of classes is finite, so natural prior $p(C)$ is the multinomial
(e.g., coin or dice)

$$p(C = c_k) = \pi_k$$

- When $x \in \mathbb{R}^D$, then it is okay to assume that $p(x|C)$ is Gaussian.
- The **class-conditional densities** are Gaussian with the same covariance matrix:

$$p(x|C_k) = \frac{1}{(2\pi)^{D/2}} \left(\frac{1}{|\Sigma|^{1/2}} \right) \exp \left\{ -\frac{1}{2} (\mathbf{x} - \boldsymbol{\mu}_k)^T \boldsymbol{\Sigma}^{-1} (\mathbf{x} - \boldsymbol{\mu}_k) \right\} \quad (3)$$

General Posterior Distribution

$$p(C_k | \mathbf{x}) = \frac{p(\mathbf{x} | C_k) p(C_k)}{\sum_k p(\mathbf{x} | C_k) p(C_k)} \quad (1)$$

$\frac{1}{(2\pi)^{D/2}} \left(\frac{1}{|\Sigma|^{1/2}} \right)$ vanishes

$$= \frac{\pi_k \exp \left\{ -\frac{1}{2} (\mathbf{x} - \boldsymbol{\mu}_k)^T \boldsymbol{\Sigma}^{-1} (\mathbf{x} - \boldsymbol{\mu}_k) \right\}}{\sum_k \pi_k \exp \left\{ -\frac{1}{2} (\mathbf{x} - \boldsymbol{\mu}_k)^T \boldsymbol{\Sigma}^{-1} (\mathbf{x} - \boldsymbol{\mu}_k) \right\}} \quad (4-1)$$

$\mathbf{x}^T \boldsymbol{\Sigma}^{-1} \mathbf{x}$ is independent from C_k !

$$= \frac{\pi_k \exp \left\{ -\frac{1}{2} (\mathbf{x}^T \boldsymbol{\Sigma}^{-1} \mathbf{x} - 2\boldsymbol{\mu}_k^T \boldsymbol{\Sigma}^{-1} \mathbf{x} + \boldsymbol{\mu}_k^T \boldsymbol{\Sigma}^{-1} \boldsymbol{\mu}_k) \right\}}{\sum_k \pi_k \exp \left\{ -\frac{1}{2} (\mathbf{x}^T \boldsymbol{\Sigma}^{-1} \mathbf{x} - 2\boldsymbol{\mu}_k^T \boldsymbol{\Sigma}^{-1} \mathbf{x} + \boldsymbol{\mu}_k^T \boldsymbol{\Sigma}^{-1} \boldsymbol{\mu}_k) \right\}} \quad (4-2)$$

$$= \frac{\pi_k \exp \left\{ \left(\boldsymbol{\mu}_k^T \boldsymbol{\Sigma}^{-1} \mathbf{x} - \frac{1}{2} \boldsymbol{\mu}_k^T \boldsymbol{\Sigma}^{-1} \boldsymbol{\mu}_k \right) \right\}}{\sum_k \pi_k \exp \left\{ \left(\boldsymbol{\mu}_k^T \boldsymbol{\Sigma}^{-1} \mathbf{x} - \frac{1}{2} \boldsymbol{\mu}_k^T \boldsymbol{\Sigma}^{-1} \boldsymbol{\mu}_k \right) \right\}} \quad (4-3)$$

Two Classes - C_k and C_j

$$p(C_k | \mathbf{x}) = \frac{\pi_k \exp \left\{ \boldsymbol{\mu}_k^T \boldsymbol{\Sigma}^{-1} \mathbf{x} - \frac{1}{2} \boldsymbol{\mu}_k^T \boldsymbol{\Sigma}^{-1} \boldsymbol{\mu}_k \right\}}{\pi_j \exp \left\{ \boldsymbol{\mu}_j^T \boldsymbol{\Sigma}^{-1} \mathbf{x} - \frac{1}{2} \boldsymbol{\mu}_j^T \boldsymbol{\Sigma}^{-1} \boldsymbol{\mu}_j \right\} + \pi_k \exp \left\{ \boldsymbol{\mu}_k^T \boldsymbol{\Sigma}^{-1} \mathbf{x} - \frac{1}{2} \boldsymbol{\mu}_k^T \boldsymbol{\Sigma}^{-1} \boldsymbol{\mu}_k \right\}} \quad (5)$$

Using $\frac{a}{a+b} = \frac{1}{1+b/a}$ and $x = e^{\ln x}$

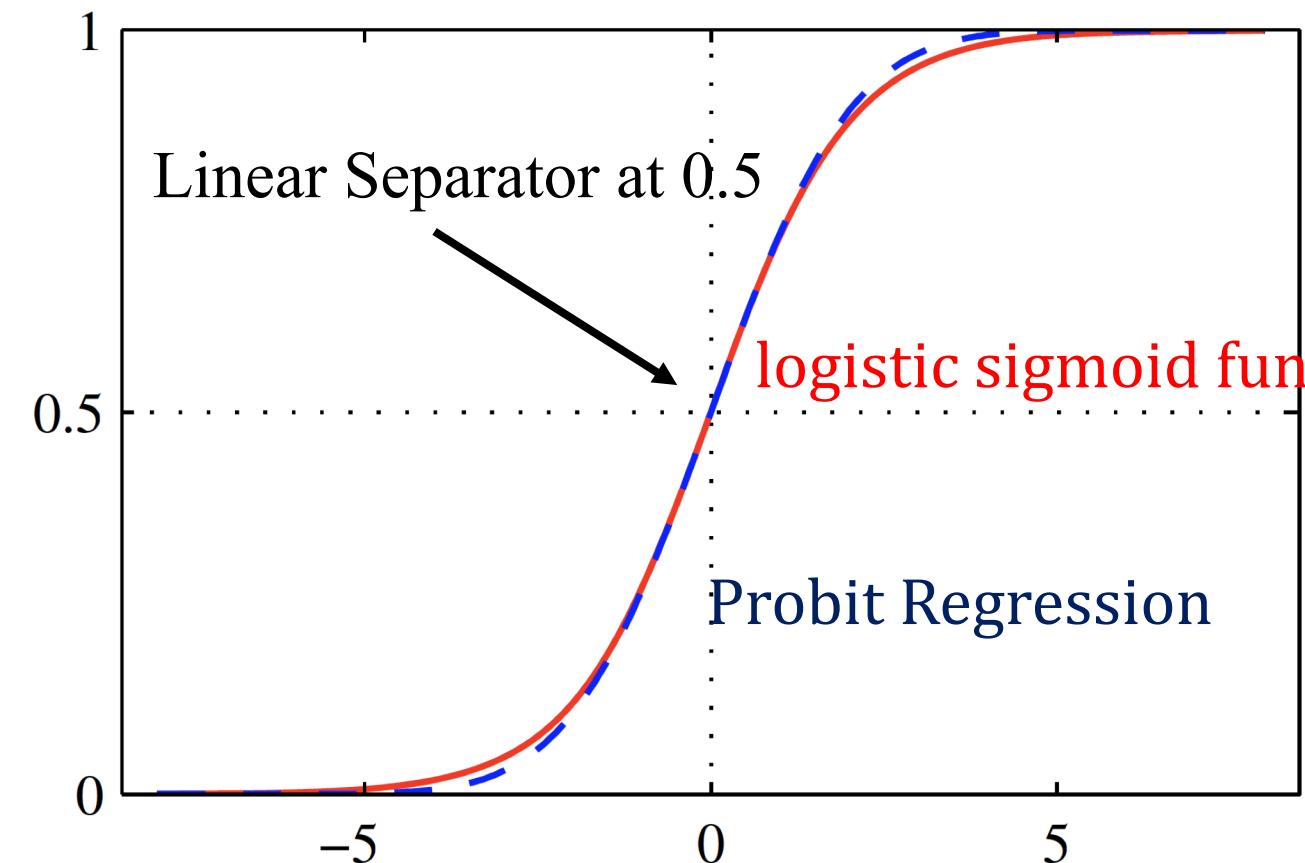
$$= \frac{1}{1 + \pi_j \exp \left\{ \boldsymbol{\mu}_j^T \boldsymbol{\Sigma}^{-1} \mathbf{x} - \frac{1}{2} \boldsymbol{\mu}_j^T \boldsymbol{\Sigma}^{-1} \boldsymbol{\mu}_j \right\} / \pi_k \exp \left\{ \boldsymbol{\mu}_k^T \boldsymbol{\Sigma}^{-1} \mathbf{x} - \frac{1}{2} \boldsymbol{\mu}_k^T \boldsymbol{\Sigma}^{-1} \boldsymbol{\mu}_k \right\}}$$

$$= \frac{1}{1 + \exp \left\{ - \left[(\underbrace{\boldsymbol{\mu}_k^T - \boldsymbol{\mu}_j^T}_{= \mathbf{w}^T} \boldsymbol{\Sigma}^{-1} \mathbf{x} - \underbrace{\frac{1}{2} \boldsymbol{\mu}_k^T \boldsymbol{\Sigma}^{-1} \boldsymbol{\mu}_k + \frac{1}{2} \boldsymbol{\mu}_j^T \boldsymbol{\Sigma}^{-1} \boldsymbol{\mu}_j + \ln \left(\frac{\pi_k}{\pi_j} \right)}_{= w_0} \right] \right\}} \quad (6)$$

Equation (6) simply becomes

$$= \frac{1}{1 + \exp \{ -(\mathbf{w}^T \mathbf{x} + w_0) \}} \quad \text{the logistic sigmoid function} \quad (7)$$

Logistic sigmoid function



The logistic sigmoid function $\sigma(a)$ is

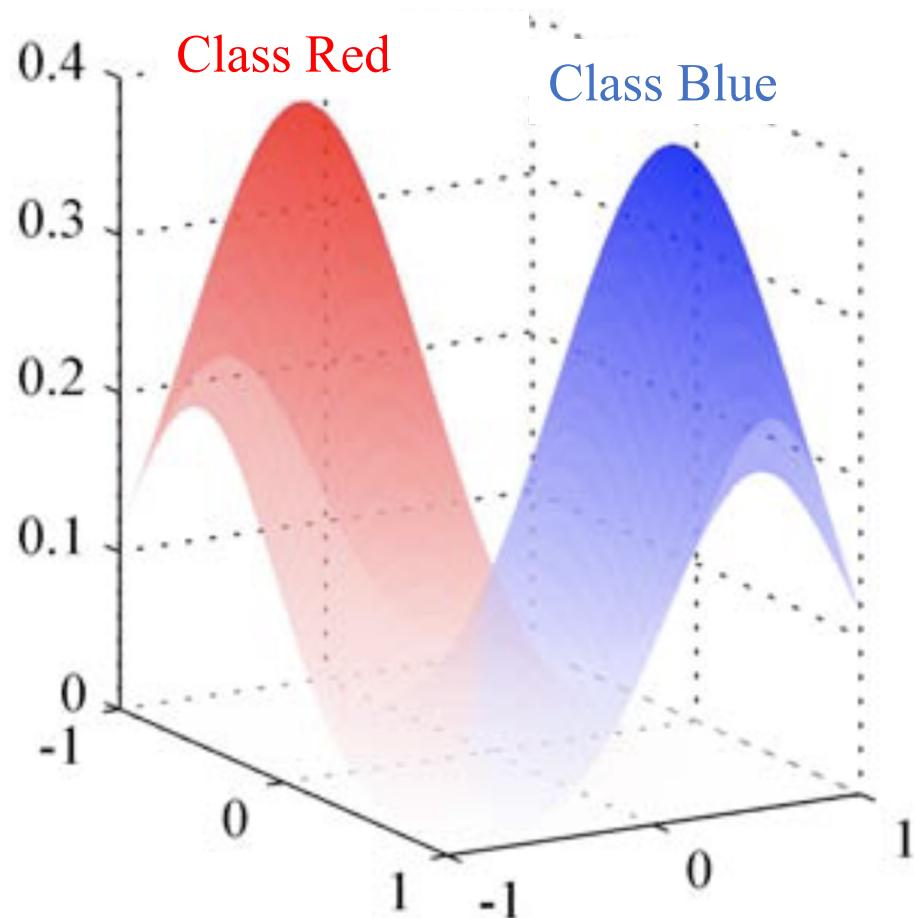
$$\sigma(a) = \frac{1}{1 + e^{-a}}$$

Then $p(C_k | \mathbf{x}) = \sigma(-\mathbf{w}^T \mathbf{x} + w_0)$

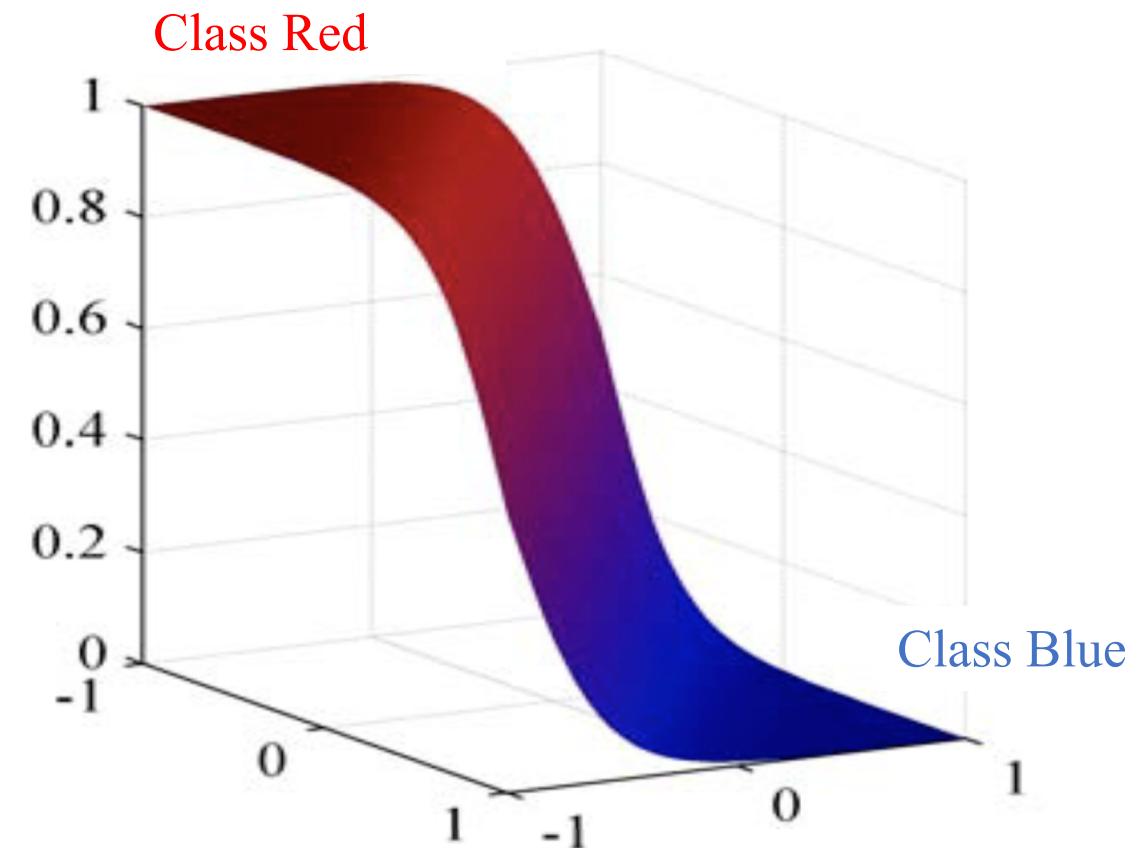
Logistic sigmoid function

Objective: find the probability of inputs likely being classified as Red.

Class Conditional

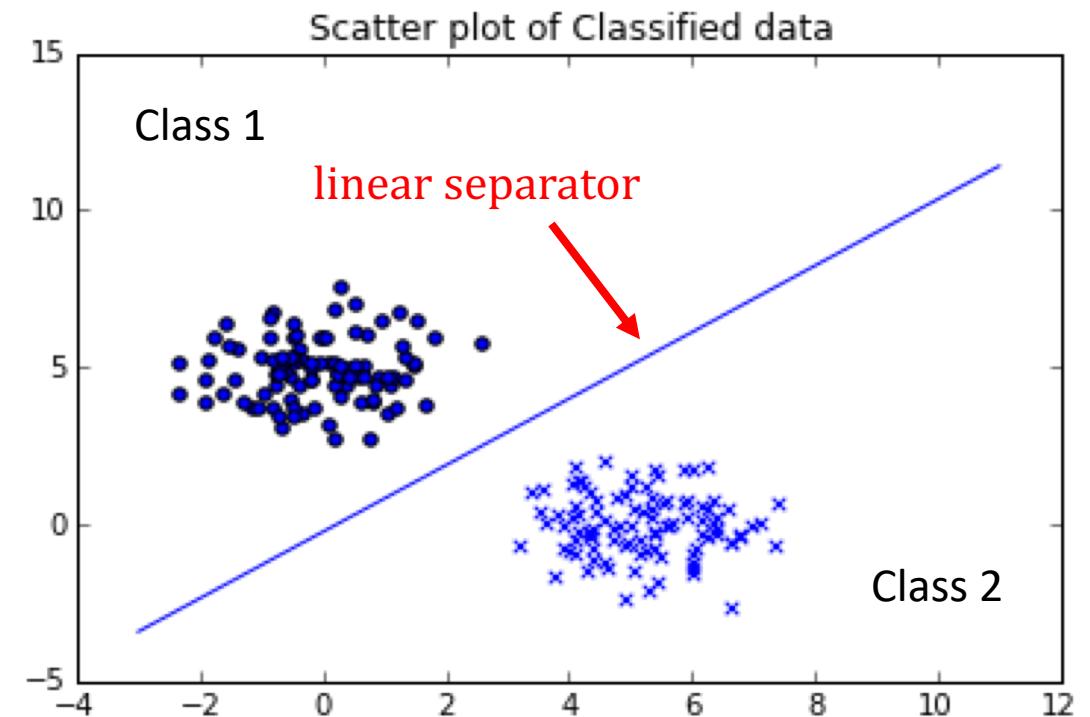


Posterior



Two Class Prediction

- best class = $\operatorname{argmax}_k P(C_k | \mathbf{x})$
- $= \begin{cases} c_1 & \sigma(a) \geq 0.5 \\ c_2 & \text{otherwise} \end{cases}$
- Class Boundary: $\sigma(w_k^T \mathbf{x} + w_0) = 0.5$
 $\Rightarrow e^{-(\mathbf{w}^T \mathbf{x} + w_0)} = 1$
 $\Rightarrow \mathbf{w}^T \mathbf{x} + w_0 = 0$
 \therefore linear separator



Posterior Distribution for Multi-Class: $k > 2$

- The normalized exponential:

$$p(C_k | \mathbf{x}) = \frac{\pi_k \exp \left\{ \boldsymbol{\mu}_k^T \boldsymbol{\Sigma}^{-1} \mathbf{x} - \frac{1}{2} \boldsymbol{\mu}_k^T \boldsymbol{\Sigma}^{-1} \boldsymbol{\mu}_k \right\}}{\sum_k \pi_k \exp \left\{ \boldsymbol{\mu}_k^T \boldsymbol{\Sigma}^{-1} \mathbf{x} - \frac{1}{2} \boldsymbol{\mu}_k^T \boldsymbol{\Sigma}^{-1} \boldsymbol{\mu}_k \right\}} \quad (4-3)$$

$$= \frac{\exp \left\{ \boldsymbol{\mu}_k^T \boldsymbol{\Sigma}^{-1} \mathbf{x} - \frac{1}{2} \boldsymbol{\mu}_k^T \boldsymbol{\Sigma}^{-1} \boldsymbol{\mu}_k + \ln \pi_k \right\}}{\sum_j \exp \left\{ \boldsymbol{\mu}_j^T \boldsymbol{\Sigma}^{-1} \mathbf{x} - \frac{1}{2} \boldsymbol{\mu}_j^T \boldsymbol{\Sigma}^{-1} \boldsymbol{\mu}_j + \ln \pi_j \right\}}$$

$$\text{softmax} \Rightarrow = \frac{e^{\mathbf{w}_k^T \bar{\mathbf{x}}}}{\sum_j e^{\mathbf{w}_j^T \bar{\mathbf{x}}}} \quad (8)$$

where $\mathbf{w}_k = \boldsymbol{\mu}_k^T \boldsymbol{\Sigma}^{-1}$ and $w_{0k} = -\frac{1}{2} \boldsymbol{\mu}_k^T \boldsymbol{\Sigma}^{-1} \boldsymbol{\mu}_k + \ln \pi_k$

Softmax

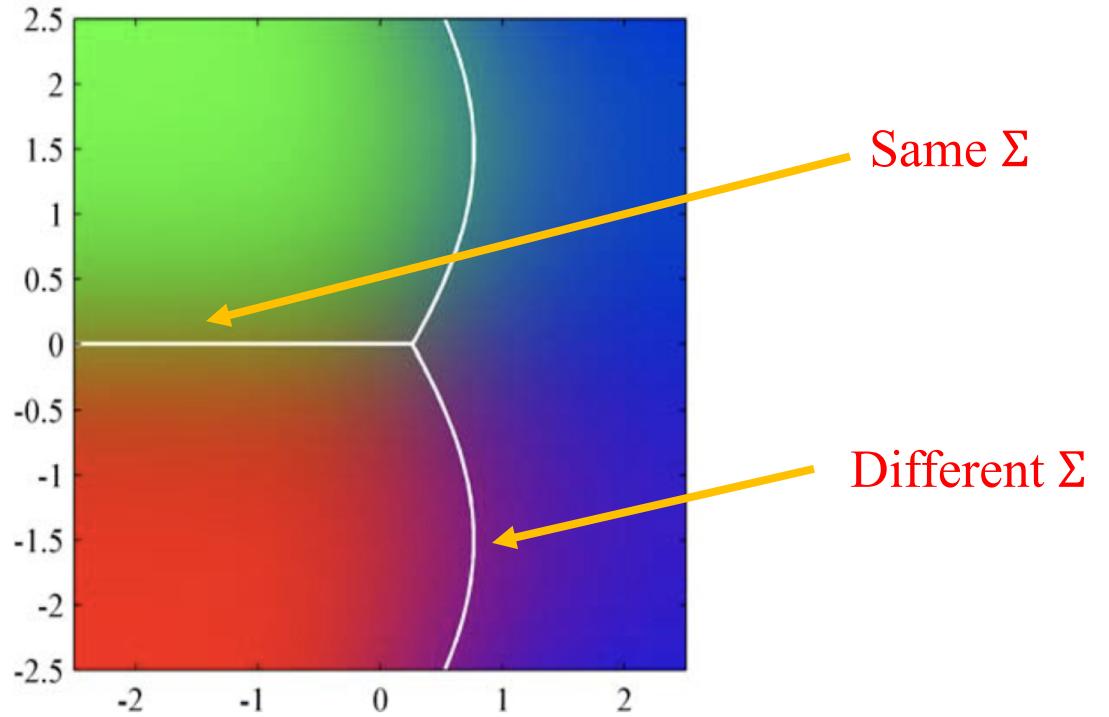
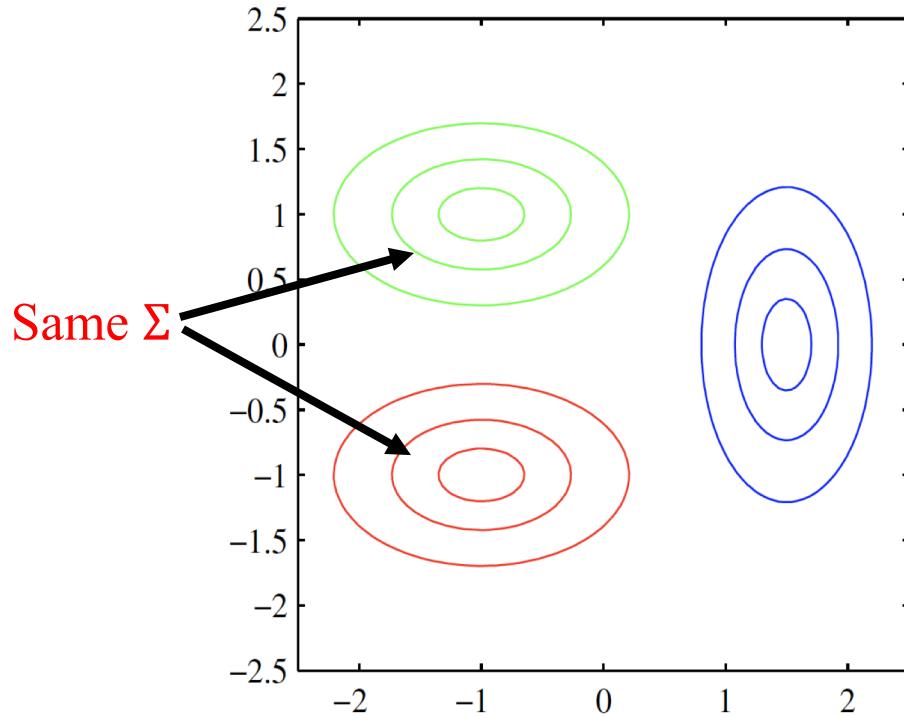
- The posterior is a softmax (generalization of the sigmoid)
- softmax distribution:

$$p(C_k | x) = \frac{\exp(f_k(x))}{\sum_j \exp(f_j(x))}$$

- argmax distribution

$$\begin{aligned} p(C_k | x) &= \begin{cases} 1 & \text{if } k = \operatorname{argmax}_j f_j(x) \\ 0 & \text{otherwise} \end{cases} \\ &= \lim_{base \rightarrow \infty} \frac{base^{f_k(x)}}{\sum_j base^{f_j(x)}} \\ &\approx \frac{\exp(f_k(x))}{\sum_j \exp(f_j(x))} \end{aligned} \tag{9}$$

Softmax



$$p(c_k | \mathbf{x}) = \frac{\pi_k \exp \left\{ -\frac{1}{2} (\mathbf{x}^T \Sigma_k^{-1} \mathbf{x} - 2\mu_k^T \Sigma_k^{-1} \mathbf{x} + \mu_k^T \Sigma_k^{-1} \mu_k) \right\}}{\sum_j \pi_j \exp \left\{ -\frac{1}{2} (\mathbf{x}^T \Sigma_j^{-1} \mathbf{x} - 2\mu_j^T \Sigma_j^{-1} \mathbf{x} + \mu_j^T \Sigma_j^{-1} \mu_j) \right\}} \quad (4-3)$$

Depends on class number j

Parameter Estimation

- We have a parametric function form for the class-conditional densities:

$$p(x|C_k) = \frac{1}{(2\pi)^{D/2}} \left(\frac{1}{|\Sigma|^{1/2}} \right) \exp \left\{ -\frac{1}{2} (\mathbf{x} - \boldsymbol{\mu}_k)^T \boldsymbol{\Sigma}^{-1} (\mathbf{x} - \boldsymbol{\mu}_k) \right\} \quad (3)$$

- We can estimate the parameters and the prior class probabilities using maximum likelihood.

- Two class case with shared covariance matrix.

- Training data:

- $\{x_n, y_n\}, n = 1, \dots, N$

- $y_n = 1$ denotes class C_1 ; $y_n = 0$ denotes class C_2 ;

- Priors: $p(C_1) = \pi, p(C_2) = 1 - \pi$

- For a data point x_n from class C_1 , we have $y_n = 1$ and therefore:

$$p(\mathbf{x}_n, C_1) = p(\mathbf{x}|C_1)p(C_1) = \pi \mathcal{N}(\mathbf{x}_n | \boldsymbol{\mu}_1, \boldsymbol{\Sigma}) \quad (9)$$

- For a data point x_n from class C_2 , we have $y_n = 0$ and therefore:

$$p(\mathbf{x}_n, C_2) = p(\mathbf{x}|C_2)p(C_2) = (1 - \pi) \mathcal{N}(\mathbf{x}_n | \boldsymbol{\mu}_2, \boldsymbol{\Sigma}) \quad (10)$$

Maximum Likelihood Solution

- Assuming observations are drawn independently, the likelihood function is as below:

$$L(\mathbf{x}, \mathbf{y}) = p(\mathbf{x}, \mathbf{y} | \pi, \mu_1, \mu_2, \Sigma) = \prod_{n=1}^N [p(\mathbf{x}_n, C_1)]^{y_n} [p(\mathbf{x}_n, C_2)]^{1-y_n} \quad (11-1)$$

Very common step

$$= \prod_{n=1}^N [\pi \mathcal{N}(\mathbf{x}_n | \boldsymbol{\mu}_1, \Sigma)]^{y_n} [(1 - \pi) \mathcal{N}(\mathbf{x}_n | \boldsymbol{\mu}_2, \Sigma)]^{1-y_n} \quad (11-2)$$


$$\ln L(\mathbf{x}, \mathbf{y}) = \sum_{n=1}^N [y_n \ln \pi + y_n \ln \mathcal{N}(\mathbf{x}_n | \boldsymbol{\mu}_1, \Sigma) + (1 - y_n) \ln(1 - \pi) - (1 - y_n) \ln \mathcal{N}(\mathbf{x}_n | \boldsymbol{\mu}_2, \Sigma)] \quad (12)$$

Maximum Likelihood Solution - parameter π

- We first maximize the log-likelihood with respect to π (set derivative to 0)

$$\frac{\partial}{\partial \pi} \left(\sum_{n=1}^N [y_n \ln \pi + (1 - y_n) \ln(1 - \pi)] \right) = 0 \quad (13)$$

- The maximum likelihood estimate of π is the fraction of points in class C_1 .
- For multi-class: maximum likelihood estimate for $p(C_k)$ is given by the fraction of points in the training set in C_k .

$$\Rightarrow \pi = \frac{1}{N} \sum_{n=1}^N y_n = \frac{N_1}{N} = \frac{N_1}{N_1 + N_2} \quad (14)$$

Maximum Likelihood Solution - parameter μ

- We then maximize the log-likelihood with respect to μ_1 (set derivative to 0)

$$\frac{\partial}{\partial \mu_1} \left(\sum_{n=1}^N y_n \ln \mathcal{N}(\mathbf{x}_n | \mu_1, \Sigma) \right) = 0$$
$$\frac{\partial}{\partial \mu_1} \left(\frac{1}{2} \sum_{n=1}^N y_n (\mathbf{x}_n - \mu_1)^T \Sigma^{-1} (\mathbf{x}_n - \mu_1) + \dots \right) = 0 \quad (15)$$

$$\mu_1 = \frac{1}{N_1} \sum_{n=1}^N y_n \mathbf{x}_n \quad (16)$$

- The maximum likelihood estimate of μ_1 is the sample mean of all input x_n in class C1.
- The maximum likelihood estimate of μ_2 is:

$$\mu_2 = \frac{1}{N_2} \sum_{n=1}^N (1 - y_n) \mathbf{x}_n \quad (17)$$

Maximum Likelihood Solution - parameter Σ

Maximize the log-likelihood w.r.t. Σ , we obtain

$$\Sigma = \frac{N_1}{N} S_1 + \frac{N_2}{N} S_2 \quad (18)$$

where

$$S_1 = \frac{1}{N_1} \sum_{n \in C_1} (x_n - \mu_1)(x_n - \mu_1)^T \quad (19)$$

$$S_2 = \frac{1}{N_2} \sum_{n \in C_2} (x_n - \mu_2)(x_n - \mu_2)^T \quad (20)$$

Probabilistic Discriminative Models

Logistic Regression & Generalization

Exponential Family

- Exponential family members - e.g., Gaussian, Bernoulli, Poisson, Beta, Dirichlet, Gamma, etc...
- For all, the posterior is in the format of:

$$P(x|\theta_k) = \exp\left(\theta_k^T T(x) - A(\theta_k) + B(x)\right) \quad (21)$$

- The posterior is a sigmoid logistic linear function \mathbf{x}

$$p(c_k|\mathbf{x}) = \sigma(\mathbf{w}^T \mathbf{x} + w_0)$$

Probabilistic Discriminative Models

- Probabilistic Generative Model - Learn prior and posterior by maximum likelihood and find posterior using Bayesian Theorem.
- The posterior is known - either logistic sigmoid or softmax
- Probabilistic Discriminative Model - Learn posterior directly by maximum likelihood.

Logistic Regression

$$L(\mathbf{x}, \mathbf{y}) = p(\mathbf{x}, \mathbf{y} | \pi, \mu_1, \mu_2, \Sigma) = \prod_{n=1}^N [\pi \mathcal{N}(\mathbf{x}_n | \boldsymbol{\mu}_1, \Sigma)]^{y_n} [(1 - \pi) \mathcal{N}(\mathbf{x}_n | \boldsymbol{\mu}_2, \Sigma)]^{1-y_n} \quad (11-2)$$

- The log-likelihood function is

$$\ln L(\mathbf{x}, \mathbf{y}) = \sum_{n=1}^N [y_n \ln \pi + y_n \ln \mathcal{N}(\mathbf{x}_n | \boldsymbol{\mu}_1, \Sigma) + (1 - y_n) \ln(1 - \pi) - (1 - y_n) \ln \mathcal{N}(\mathbf{x}_n | \boldsymbol{\mu}_2, \Sigma)] \quad (22)$$

- Use the negative log-likelihood function (cross-entropy error function) to find the parameters

$$E(\mathbf{w}) = -\ln L(\mathbf{w}) = -\sum_{n=1}^N [y_n \ln \sigma(\mathbf{w}^T \bar{\mathbf{x}}) + (1 - y_n) \ln (1 - \sigma(\mathbf{w}^T \bar{\mathbf{x}}))] \quad (23)$$

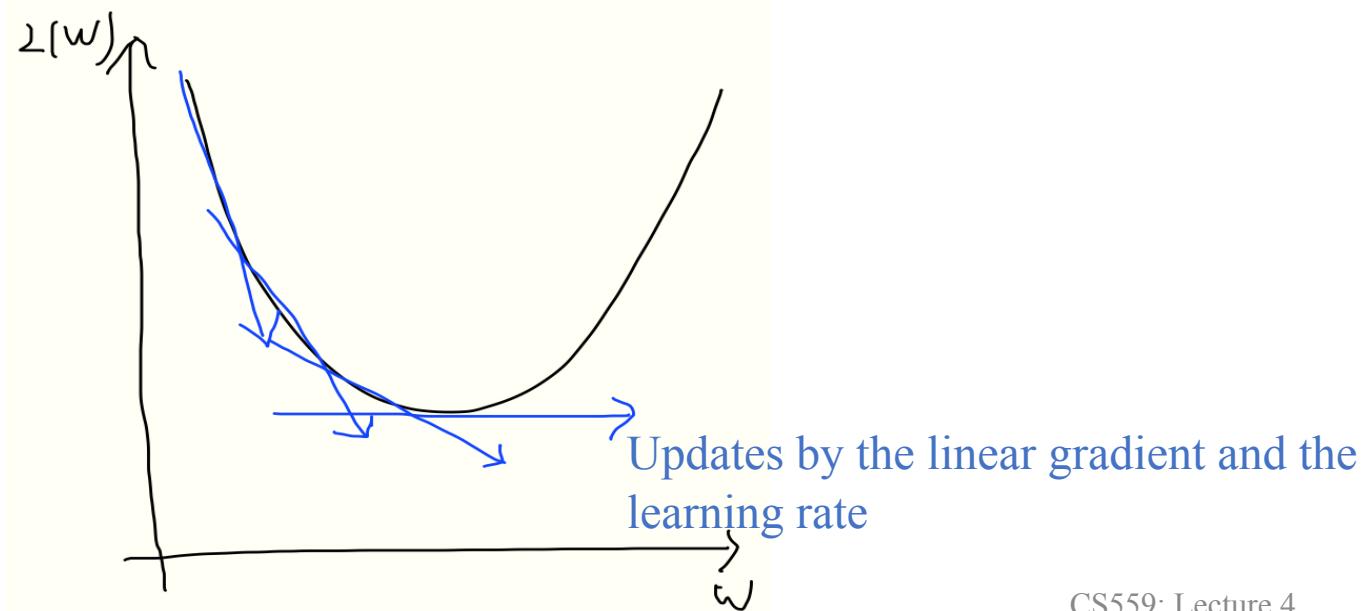
- The goal is to estimate the continuous posterior function.
- Logistic regression is a form of classification.

Maximum Likelihood

$$-\frac{\partial E(\mathbf{w})}{\partial w} = -\sum_n y_n \left(\frac{\sigma(\mathbf{w}^T \bar{\mathbf{x}}) (1 - \sigma(\mathbf{w}^T \bar{\mathbf{x}})) \bar{\mathbf{x}}_n}{\sigma(\mathbf{w}^T \bar{\mathbf{x}})} \right) - \sum_n \frac{(1 - y_n) (1 - \sigma(\mathbf{w}^T \bar{\mathbf{x}})) \sigma(\mathbf{w}^T \bar{\mathbf{x}}) (-\bar{\mathbf{x}}_n)}{1 - \sigma(\mathbf{w}^T \bar{\mathbf{x}})} \quad (24-1)$$

$$\begin{aligned} 0 &= -\sum_n y_n \bar{\mathbf{x}}_n - \sum_n y_n \sigma(\mathbf{w}^T \bar{\mathbf{x}})(\bar{\mathbf{x}}_n) + \sum_n \sigma(\mathbf{w}^T \bar{\mathbf{x}}) \bar{\mathbf{x}}_n + \sum_n y_n \sigma(\mathbf{w}^T \bar{\mathbf{x}})(\bar{\mathbf{x}}_n) \\ &= \sum_n [\sigma(\mathbf{w}^T \bar{\mathbf{x}}) - y_n] \bar{\mathbf{x}}_n \end{aligned} \quad (24-2)$$

- Can we estimate \mathbf{w} ? No
- Then how? Use an iterative method



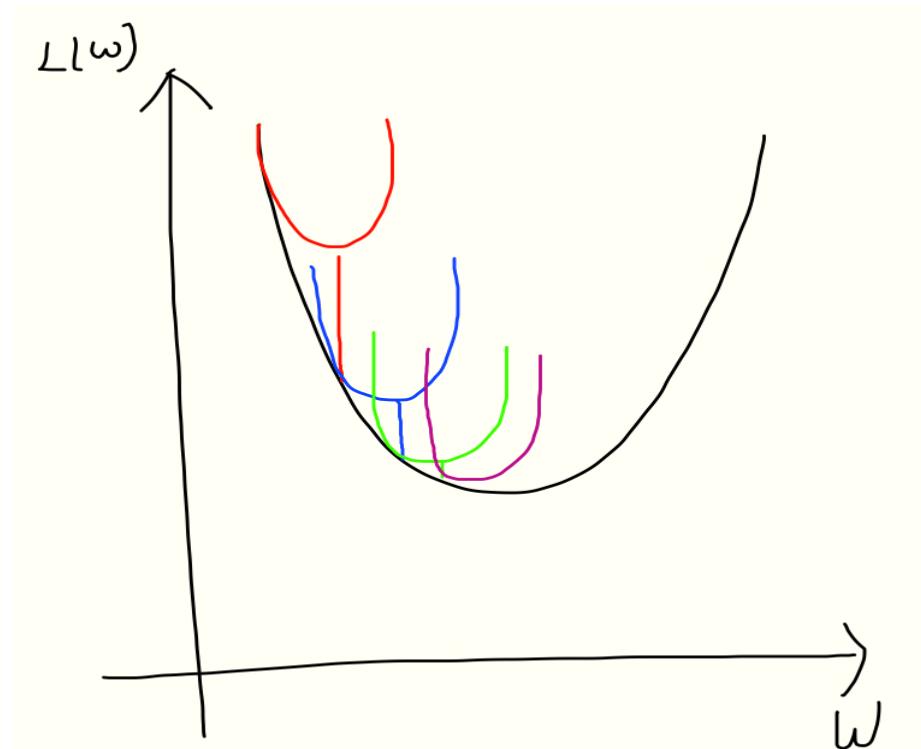
Newton's Method

Iterative reweighted least square:

$$\mathbf{w} \leftarrow \mathbf{w} - \mathbf{H}^{-1} \nabla E(\mathbf{w}) \quad (25)$$

where ∇E is the gradient (column vector) and \mathbf{H} is the Hessian (matrix)

$$\mathbf{H} = \begin{bmatrix} \frac{\partial^2 E}{\partial^2 w_0} & \cdots & \frac{\partial^2 E}{\partial w_0 \partial w_m} \\ \vdots & \ddots & \vdots \\ \frac{\partial^2 E}{\partial w_m \partial w_0} & \cdots & \frac{\partial^2 E}{\partial^2 w_m} \end{bmatrix} \quad (25-1)$$



Hessian

$$\mathbf{H} = \nabla(\nabla \ln L(w)) \quad (26)$$

$$\begin{aligned} &= \nabla \left(\sum_n [\sigma(\mathbf{w}^T \bar{\mathbf{x}}) - y_n] \bar{\mathbf{x}}_n \right) \\ &= \sum_{n=1}^N \sigma(\mathbf{w}^T \bar{\mathbf{x}}_n) (1 - \sigma(\mathbf{w}^T \bar{\mathbf{x}}_n)) \bar{\mathbf{x}}_n \bar{\mathbf{x}}_n^T \\ &= \bar{\mathbf{X}} \mathbf{R} \bar{\mathbf{X}}^T \end{aligned} \quad (27)$$

$$\text{where } \mathbf{R} = \begin{bmatrix} \sigma_1(1 - \sigma_1) & \cdots & 0 \\ \vdots & \ddots & \vdots \\ 0 & \cdots & \sigma_N(1 - \sigma_N) \end{bmatrix}$$

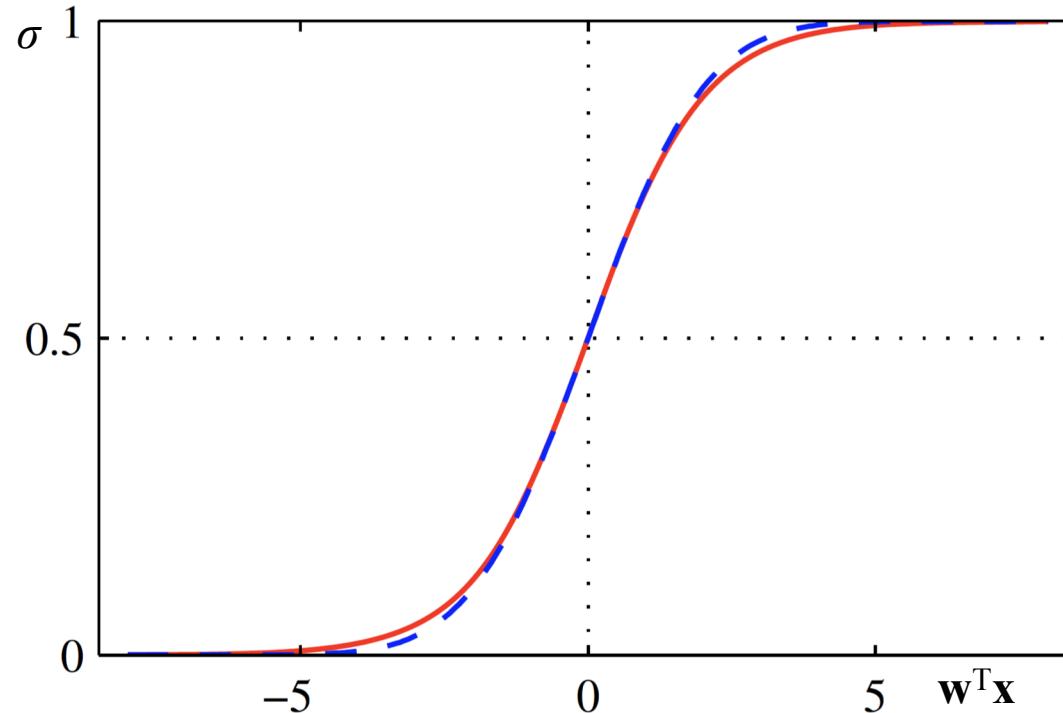
Issues

Issue: Maximum likelihood can exhibit overfitting: logistic regression can classify each data point arbitrarily well when data is small.

Reasons:

As the posterior gets toward 1, $\mathbf{w}^T \mathbf{x} \rightarrow \infty$ and therefore the magnitude of \mathbf{w} must be infinite.

If $\sigma=1$, then $(1-\sigma)=0$ so as $R=0$ and therefore H becomes singular.



Regularization

- We can penalize large weights
- How? Add the penalty term λ .

$$\min_w E(w) + \frac{1}{2} \lambda \|w\|^2 \quad (28)$$

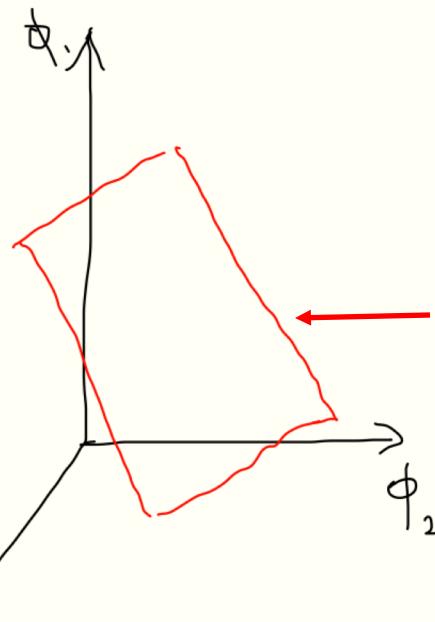
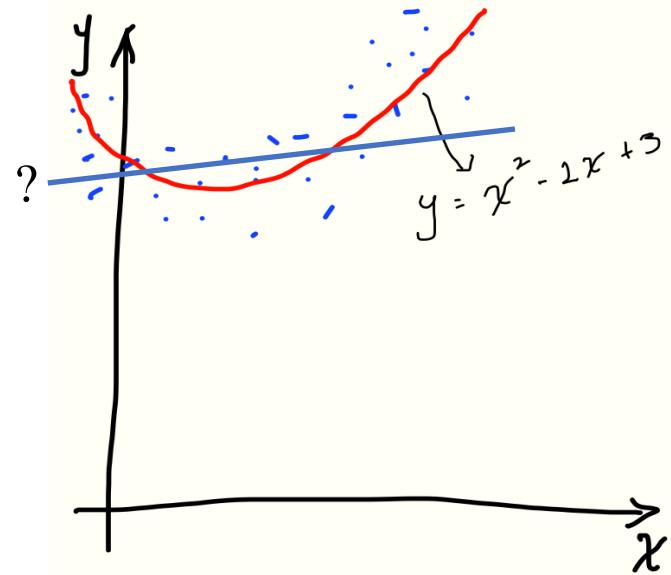
$$= \min_w \left\{ - \sum_{n=1}^N \left[y_n \ln \sigma(\mathbf{w}^T \bar{\mathbf{x}}) + (1 - y_n) \ln (1 - \sigma(\mathbf{w}^T \bar{\mathbf{x}})) \right] \right\} + \frac{1}{2} \lambda \mathbf{w}^T \mathbf{w} \quad (28-1)$$

- Hessian: $H = \bar{\mathbf{X}} \mathbf{R} \bar{\mathbf{X}}^T + \lambda \mathbf{I}$ (29)
- The term $\lambda \mathbf{I}$ ensures that H is not singular ($\text{eigenvalues} \geq \lambda$)

Generalized Linear Models

- Can we do non-linear classification?
- How can we do so? We can map inputs to a different space and so linear classification in that space.

Basis Functions



Use non-linear basis functions. Examples of basis functions are

polynomial: $\phi_j(x) = x^j$

Gaussian: $\phi_j(x) = e^{-\frac{(x-\mu_j)^2}{2s_j^2}}$

Sigmoid: $\phi(x) = \sigma\left(\frac{x - \mu_j}{s_j}\right)$

Other many as long as they seem appropriate.

The new space we have is $H = \{x \rightarrow \sum_i w_i \phi_i(x) \mid w_i \in \mathbb{R}\}$

Kernel Methods

Kernel Function

- Given: There are a large set of fixed linear basis functions.
 - Problem: It all depends on how complex the basis functions are.
 - Solution: Use “dual trick” that depends on the amount data than the function.
-
- We have a set of basis functions $\phi(x)$ that map inputs x to a feature space.
 - This feature space appears in the dot product $\phi(x)^T \phi(x')$ of input pairs, x, x' .
 - The kernel function $k(x, x') = \phi(x)^T \phi(x')$ in feature space.
 - We are interested in kernel function not the set of basis functions.

Dual Representations

The linear regression objective is the error.

$$E(\mathbf{w}) = \frac{1}{2} \sum_{n=1}^N [w^T \phi(x_n) - y_n]^2 + \frac{1}{2} \lambda \mathbf{w}^T \mathbf{w} \quad (30)$$

We set the gradient to 0.

$$E(\mathbf{w}) = \sum_n (\mathbf{w}^T \phi(x_n) - y_n) \phi(x_n) + \lambda \mathbf{w} = 0 \quad (31)$$

$$\mathbf{w} = -\frac{1}{\lambda} \sum_n (\mathbf{w}^T \phi(x_n) - y_n) \phi(x_n) \quad (32)$$

\mathbf{w} is linear combination of inputs in feature space

$$\{\phi(x_n) | 1 \leq n \leq N\}$$

Dual Representations

Substitute $\mathbf{w} = \Phi\mathbf{a}$

Where $\Phi = [\phi(x_1), \phi(x_2), \dots, \phi(x_N)]$

$$\mathbf{a} = \begin{bmatrix} a_1 \\ \vdots \\ a_N \end{bmatrix} \text{ and } a_n = -\frac{1}{\lambda}(\mathbf{w}^T \phi(x_n) - y_n)$$

Dual objective: minimize E with respect to a

$$E(a) = \frac{1}{2}a^T \Phi^T \Phi \Phi^T \Phi a - a^T \Phi^T \Phi y + \frac{y^T y}{2} + \frac{\lambda}{2} a^T \Phi^T \Phi a \quad (33)$$

Gram Matrix

The Gram Matrix: $K = \Phi^T \Phi$

Substitution K into (31)

$$E(a) = \frac{1}{2} a^T K K a - a^T K y + \frac{y^T y}{2} + \frac{\lambda}{2} a^T K a \quad (34)$$

$$\nabla E(a) = K K a - K y + \lambda K a$$

$$K y = K(K + \lambda I)a$$

$$a = (K - \lambda I)^{-1}y$$

Prediction $y_* = \phi(x_*)^T w = \phi(x_*)^T \Phi a = k(x_*, X)(K + \lambda I)^{-1}y$ (35)

where (X, y) is the training set and (x_*, y_*) is a test instance

- Primal Solution: depends on # of basis functions
- Dual solution: depends on amount of data
 - Advantage: can use very large # of basis functions

Just need to know k

Constructing Kernels

Possibilities:

- Find mapping Φ to feature space and let $K = \phi^T \phi$
- Directly specify K

A valid kernel must be positive semi-definite:

- k must factor into the product of a transpose matrix by itself
- or all eigenvalues ≥ 0
- problem? It is not always easy to verify.

Rules to construct Kernels

Let $k_1(x, x')$ and $k_2(x, x')$ be valid kernels.

The following kernels are also valid:

$$= ck_1(x, x') \text{ c is the constant } > 0$$

$$= f(x)k_1(x, x')f(x') \text{ f is the function}$$

$$= q(k_1(x, x')) \text{ q is the polynomial}$$

$$= \exp(k_1(x, x'))$$

$$k(x, x') = k_1(x, x') + k_2(x, x')$$

$$= k_3(\phi(x), \phi(x'))$$

$$x^T Ax' \text{ A is symmetric positive semi-definite}$$

$$= k_a(x_a, x'_a) + k_b(x_b, x'_b) \text{ } k_a \text{ & } k_b \text{ are valid kernel}$$

$$= k_a(x_a, x'_a)k_b(x_b, x'_b)$$

sets, strings, graphs, etc...

Example -

Lodhi, Saunders, Shawe-Taylor, Christianini, Watkins,

Text Classification Using String Kernels, JMLR, p. 419-444, 2002

Video Presentation

- Daniel Kadyrov – Logistic Regression
- Manoj Menon – Generalization (either classification/regression)
- Ravi Patel – kernel method (not SVM or Gaussian Processes)
- As the lecture is posted late, the posting presentation day and replying dates will be postponed accordingly. (**Due 2/20 Thursday**)