# CS 559: Linear Classification II & Kernel Method I

Lecture 4

Spring 2020

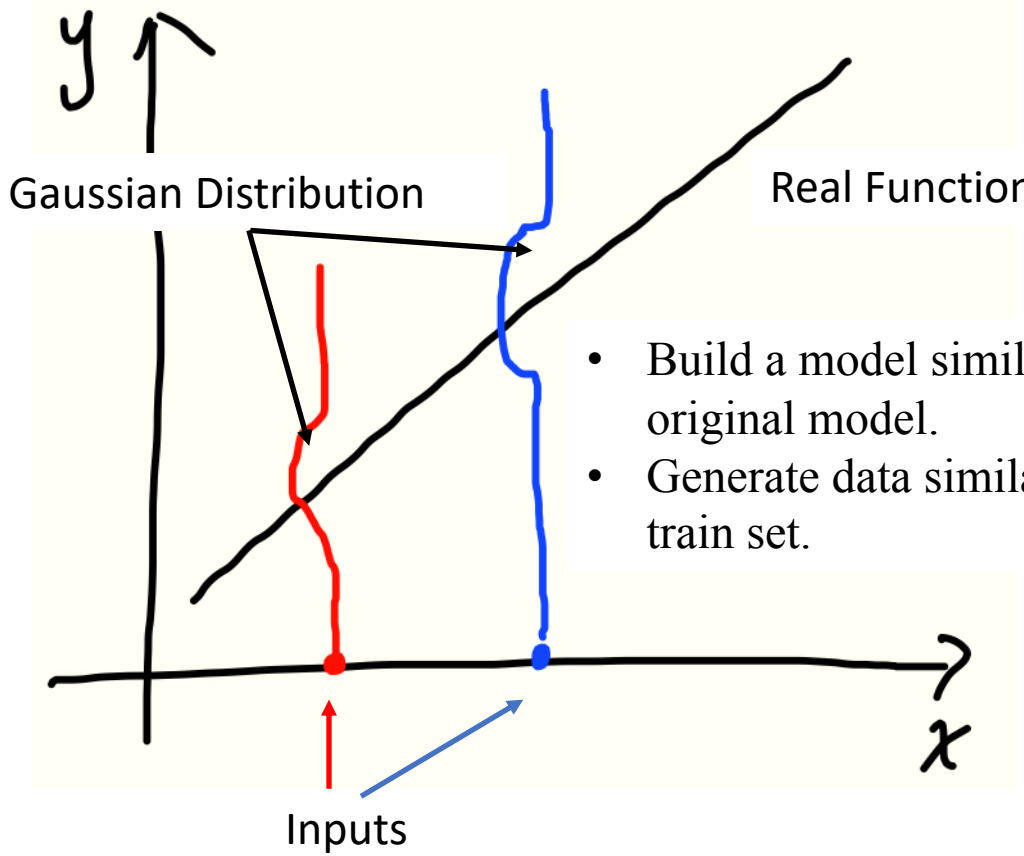
# Outline

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

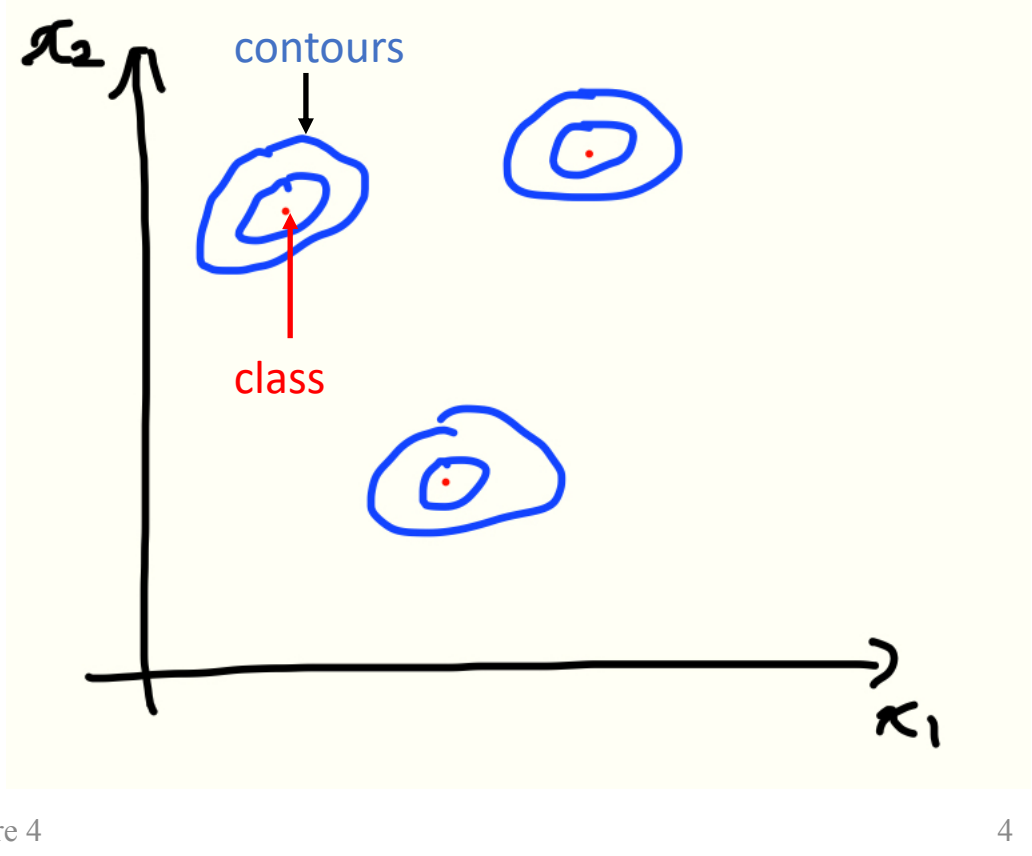
# Linear Classification II

# 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

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

- Use decision theory to determine class membership for each new input 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:

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

# General Posterior Distribution

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

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

$$= \frac{\pi_k \exp\left\{-\frac{1}{2}(\mathbf{x}-\boldsymbol{\mu_k})^T\boldsymbol{\Sigma}^{-1}(\mathbf{x}-\boldsymbol{\mu_k})\right\}}{\sum_k \pi_k \exp\left\{-\frac{1}{2}(\mathbf{x}-\boldsymbol{\mu_k})^T\boldsymbol{\Sigma}^{-1}(\mathbf{x}-\boldsymbol{\mu_k})\right\}} \quad (5\text{-}1)$$

Independent from $C_k$ !

$$= \frac{\pi_k \exp\left\{-\frac{1}{2}\left(\mathbf{x^T\Sigma^{-1}x} - 2\boldsymbol{\mu_k^T\Sigma^{-1}x} + \boldsymbol{\mu_k^T\Sigma^{-1}\mu_k}\right)\right\}}{\sum_k \pi_k \exp\left\{-\frac{1}{2}\left(\mathbf{x^T\Sigma^{-1}x} - 2\boldsymbol{\mu_k^T\Sigma^{-1}x} + \boldsymbol{\mu_k^T\Sigma^{-1}\mu_k}\right)\right\}} \quad (5\text{-}2)$$

$$= \frac{\pi_k \exp\left\{\frac{1}{2}\left(2\boldsymbol{\mu_k^T\Sigma^{-1}x} - \boldsymbol{\mu_k^T\Sigma^{-1}\mu_k}\right)\right\}}{\sum_k \pi_k \exp\left\{\frac{1}{2}\left(2\boldsymbol{\mu_k^T\Sigma^{-1}x} - \boldsymbol{\mu_k^T\Sigma^{-1}\mu_k}\right)\right\}} \quad (5\text{-}3)$$

# Two Classes - $C_k$ and $C_j$

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

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

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

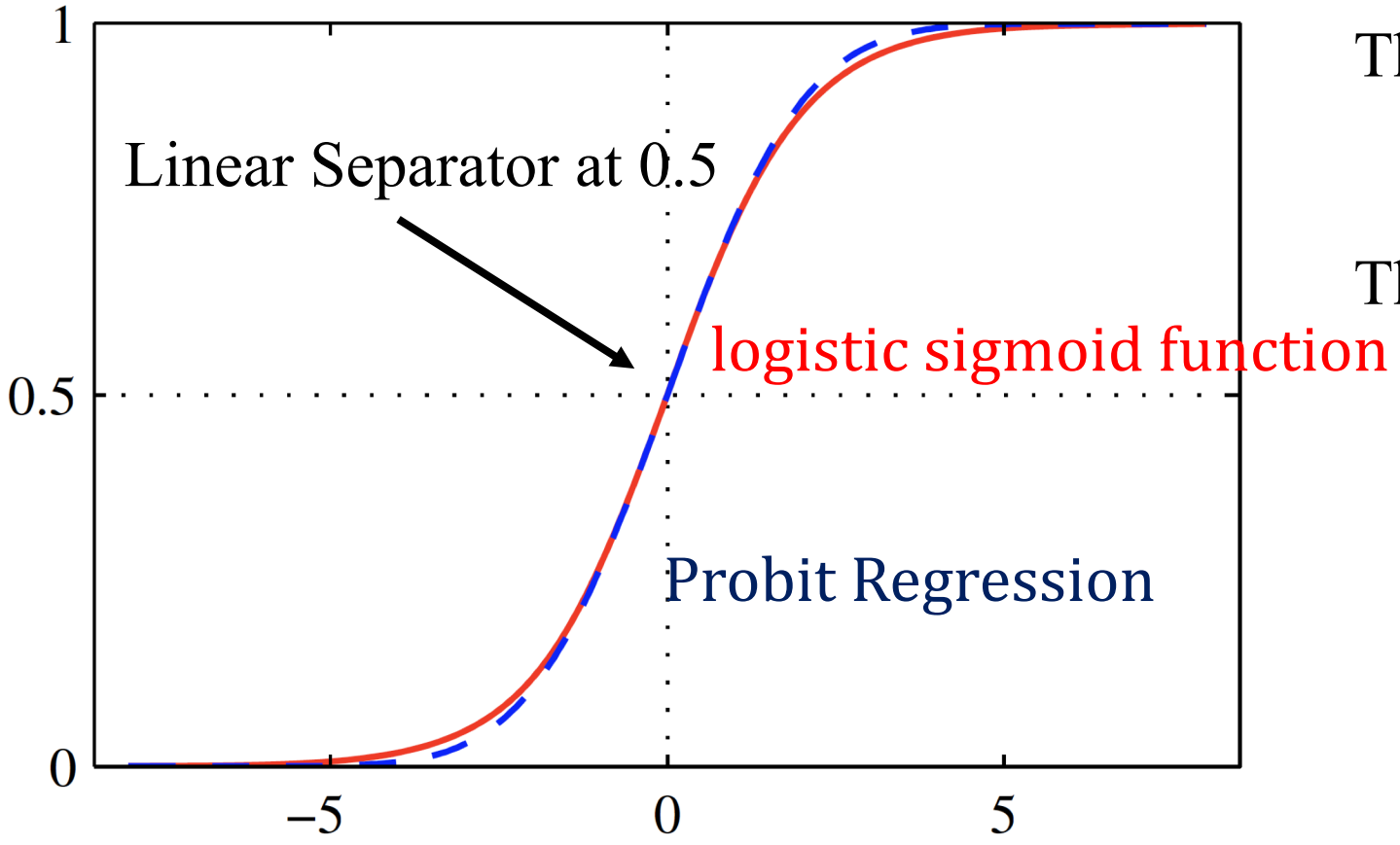
$$= \frac{1}{1 + \exp\left\{-\left[\underbrace{(\boldsymbol{\mu}_k^T - \boldsymbol{\mu}_j^T)\boldsymbol{\Sigma}^{-1}}_{=\boldsymbol{w}^T}\mathbf{x} \underbrace{- \frac{1}{2}\boldsymbol{\mu}_k^T\boldsymbol{\Sigma}^{-1}\boldsymbol{\mu}_k + \frac{1}{2}\boldsymbol{\mu}_j^T\boldsymbol{\Sigma}^{-1}\boldsymbol{\mu}_j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

Equation (6) simply becomes

$$= \frac{1}{1 + \exp\{-(\mathbf{w}^T\mathbf{x} + w_0)\}} \tag{7}$$

**the logistic sigmoid function**

# Logistic sigmoid function



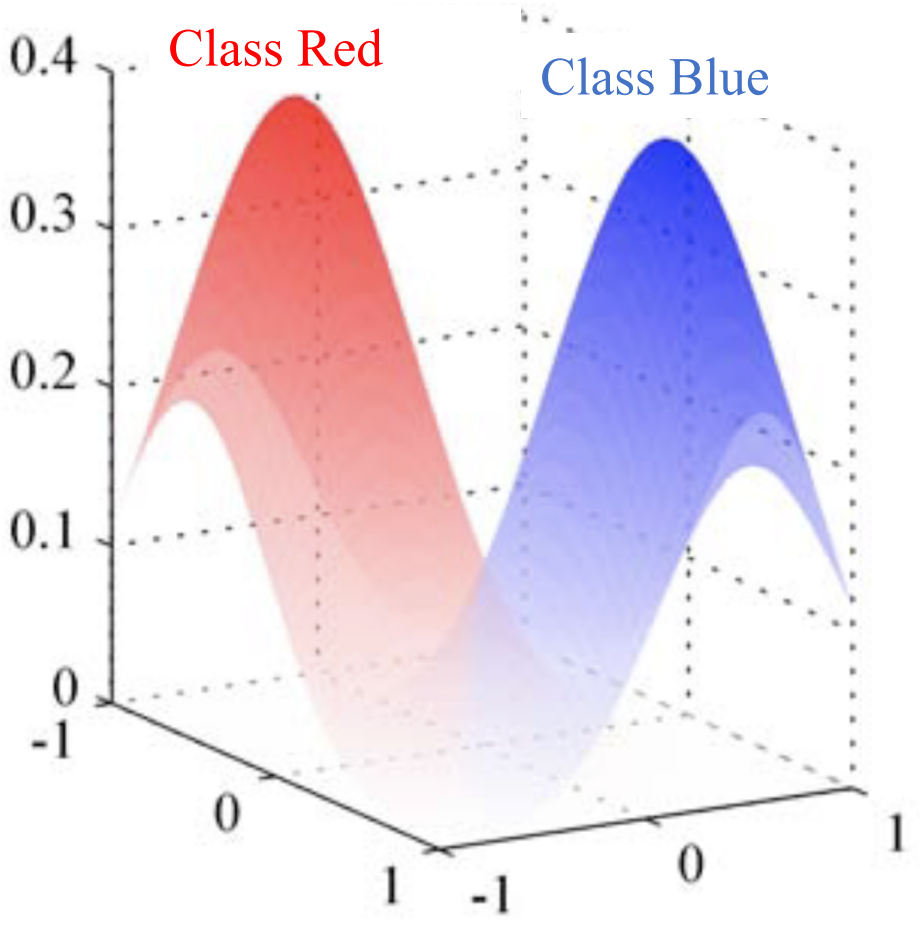

The logistic sigmoid function $\sigma(a)$ is

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

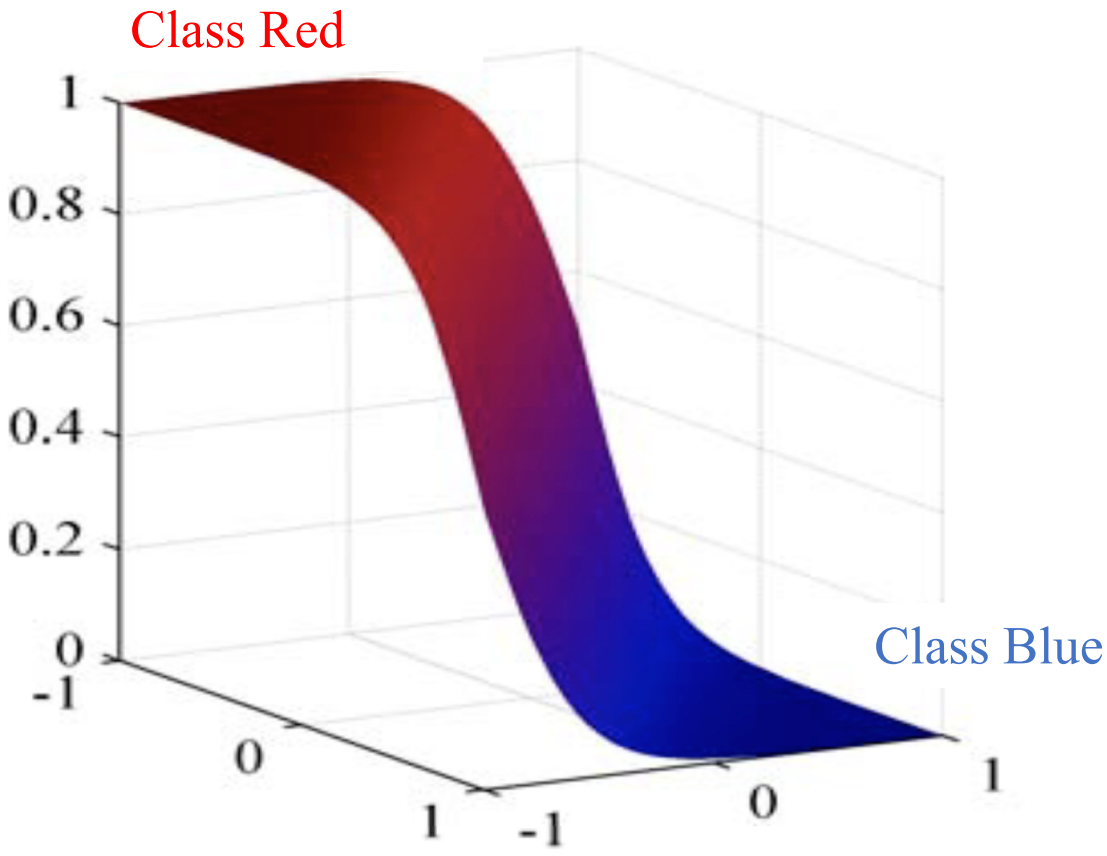
Then $p(C_k|\mathbf{x}) = \sigma(-\mathbf{w}^{\mathbf{T}}\mathbf{x} + w_0)$

# Logistic sigmoid function

Class Conditional




Posterior



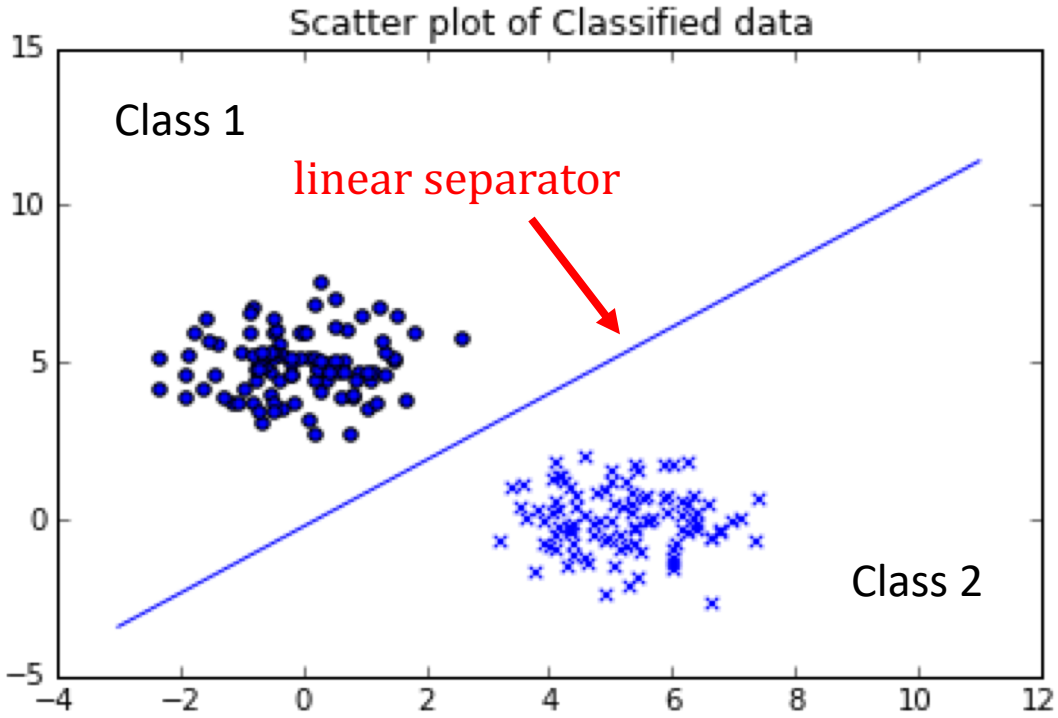
# Two Class Prediction

- $best\ class = argmax_k P(C_k|\mathbf{x})$
- $= \begin{cases} c_1 & \sigma(a) \geq 0.5 \\ c_2 & \text{otherwise} \end{cases}$
- Class Boundary: $\sigma\left(w_k^T x + w_0\right) = 0.5$

$$\implies e^{-(\mathbf{w^T x} + w_0)} = 1$$

$$\implies \mathbf{w^T x} + w_0 = 0$$

$\therefore$ linear separator



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

- The normalized exponential:

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

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

$$\text{softmax} \Longrightarrow = \frac{e^{\mathbf{w_k^T}\bar{\mathbf{x}}}}{\sum_j e^{\mathbf{w_j^T}\bar{\mathbf{x}}}} \tag{8}$$

where $\mathbf{w_k} = \boldsymbol{\mu}_k^T\boldsymbol{\Sigma}^{-1}$ and $w_0 = -\frac{1}{2}\boldsymbol{\mu}_k^T\boldsymbol{\Sigma}^{-1}\boldsymbol{\mu}_k + \ln \pi_k$

# Softmax

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

$$p(C_k \mid \boldsymbol{x}) = \frac{\exp\left(f_k(x)\right)}{\sum_j \exp(f_j(x))}$$

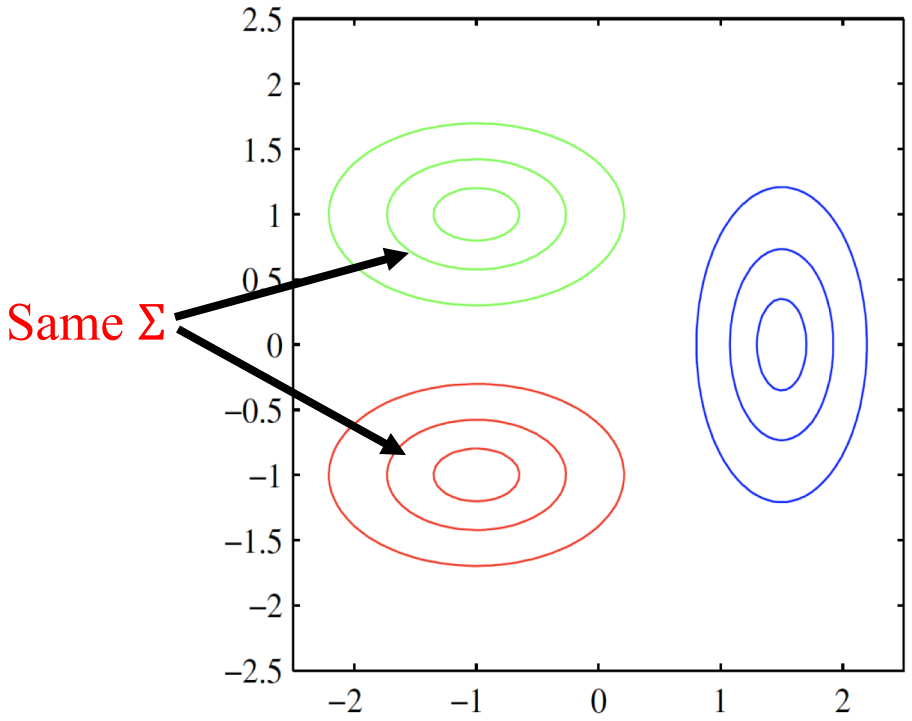
- argmax distribution

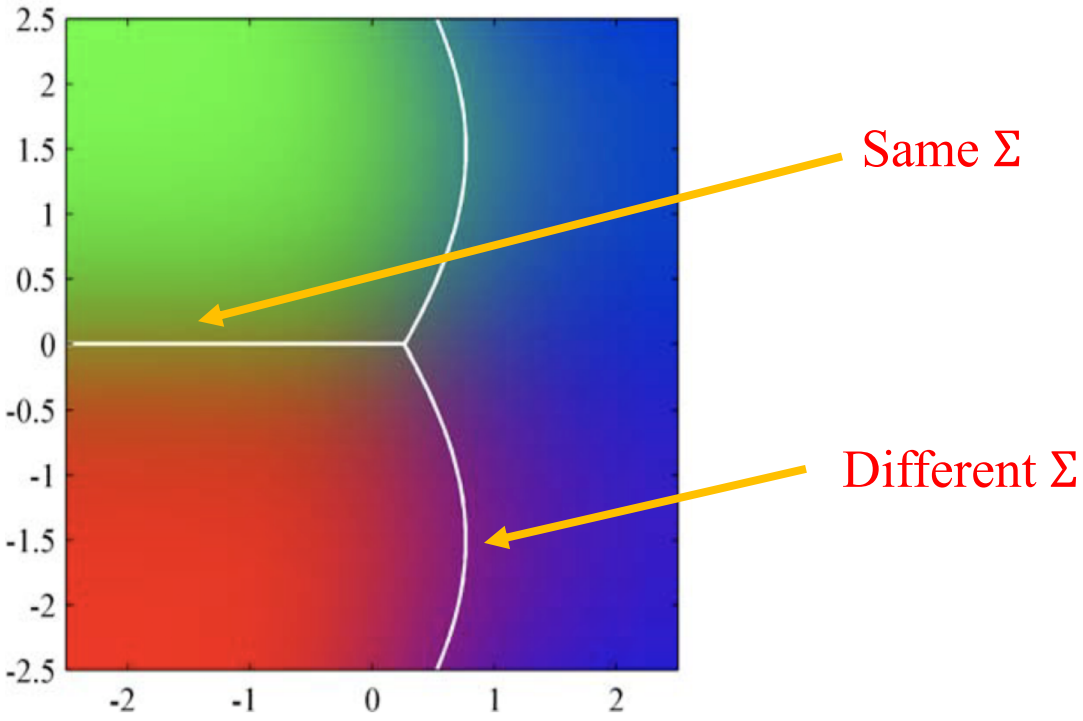
$$p(C_k \mid \boldsymbol{x}) = \begin{cases} 1 & \text{if } k = argmax_j\ f_j(x) \\ 0 & \text{otherwise} \end{cases}$$

$$= \lim_{base \to \infty} \frac{base^{f_k(x)}}{\sum_j base^{f_j(x)}} \qquad (10)$$

$$\approx \frac{\exp\left(f_k(x)\right)}{\sum_j \exp(f_j(x))}$$

# Softmax




$$p(c_k \mid \mathbf{x}) = \frac{\pi_k \exp\left\{-\frac{1}{2}\left(\mathbf{x}^{\mathbf{T}}\boldsymbol{\Sigma}_{\mathbf{k}}^{-1}\mathbf{x} - 2\boldsymbol{\mu}_k^T\boldsymbol{\Sigma}_{\mathbf{k}}^{-1}\mathbf{x} + \boldsymbol{\mu}_k^T\boldsymbol{\Sigma}_{\mathbf{k}}^{-1}\boldsymbol{\mu}_k\right)\right\}}{\sum_j \pi_j \exp\left\{-\frac{1}{2}\left(\mathbf{x}^{\mathbf{T}}\boldsymbol{\Sigma}_{\boldsymbol{j}}^{-1}\mathbf{x} - 2\boldsymbol{\mu}_j^T\boldsymbol{\Sigma}_{\mathbf{j}}^{-1}\mathbf{x} + \boldsymbol{\mu}_j^T\boldsymbol{\Sigma}_{\mathbf{j}}^{-1}\boldsymbol{\mu}_j\right)\right\}} \quad (5\text{-}2)$$

Depends on class number $j$

# Parameter Estimation

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

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

- We can estimate the parameters and the prior class probabilities using maximum likelihood.
  - Two class case with shared covariance matrix.
  - Training data:
  - $\{x_n, y_n\}, n = 1, \ldots, N$
  - $y_n = 1$ 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:

$$p(\mathbf{x_n}, C_1) = p(\mathbf{x}|C_1)p(C_1) = \pi\mathcal{N}(\mathbf{x_n}|\boldsymbol{\mu_1}, \Sigma) \quad (9)$$

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

$$p(\mathbf{x_n}, C_2) = p(\mathbf{x}|C_2)p(C_2) = (1-\pi)\mathcal{N}(\mathbf{x_n}|\boldsymbol{\mu_2}, \Sigma) \quad (10)$$

# Maximum Likelihood Solution

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

$$L(\mathbf{x}, \mathbf{y}) = p(\mathbf{x}, \mathbf{y}|\pi, \mu_1, \mu_2, \Sigma) = \prod_{n=1}^{N} [p(\mathbf{x_n}, C_1)]^{y_n} [p(\mathbf{x_n}, C_2)]^{1-y_n} \quad (13\text{-}1)$$

Very common step

$$= \prod_{n=1}^{N} [\pi \mathcal{N}(\mathbf{x_n}|\boldsymbol{\mu_1}, \Sigma)]^{y_n} [(1-\pi)\mathcal{N}(\mathbf{x_n}|\boldsymbol{\mu_2}, \Sigma)]^{1-y_n} \quad (13\text{-}2)$$

$$\ln L(\mathbf{x}, \mathbf{y}) = \sum_{n=1}^{N} [y_n \ln \pi + y_n \ln \mathcal{N}(\mathbf{x_n}|\boldsymbol{\mu_1}, \Sigma) + (1-y_n) \ln(1-\pi) - (1-y_n) \ln \mathcal{N}(\mathbf{x_n}|\boldsymbol{\mu_2}, \Sigma)] \quad (14)$$

# Maximum Likelihood Solution - parameter $\pi$

- We first maximize the log-likelihood with respect to $\pi$ (set derivative to 0)

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

- The maximum likelihood estimate of $\pi$ 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 $C_k$.

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

# Maximum Likelihood Solution - parameter $\mu$

- We them maximize the log-likelihood with respect to $\mu_1$ (set derivate to 0)

$$\frac{\partial}{\partial \mu_1}\left(\sum_{n=1}^{N} y_n \ln \mathcal{N}(\mathbf{x}_n|\mu_1, \Sigma)\right) = 0$$

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

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

- The maximum likelihood estimate of $\mu_1$ is the sample mean of all input xn in class C1.
- The maximum likelihood estimate of $\mu_2$ is:

$$\boldsymbol{\mu}_2 = \frac{1}{N_2}\sum_{n=1}^{N} (1 - y_n)\mathbf{x}_n \quad (17)$$

# Maximum Likelihood Solution - parameter Σ

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

$$\boldsymbol{\Sigma} = \frac{N_1}{N}\boldsymbol{S_1} + \frac{N_2}{N}\boldsymbol{S_2} \tag{18}$$

where

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

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

# 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:

$$P(x|\theta_k) = \exp\left(\theta_k^T T(x) - A(\theta_k) + B(x)\right) \quad (23)$$

- The posterior is a sigmoid logistic linear function x

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

# 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

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

- The log-likelihood function is

$$\ln L(\mathbf{x}, \mathbf{y}) = \sum_{n=1}^{N} [y_n \ln \pi + y_n \ln \mathcal{N}(\mathbf{x_n}|\boldsymbol{\mu_1}, \Sigma) + (1-y_n) \ln(1-\pi) - (1-y_n) \ln \mathcal{N}(\mathbf{x_n}|\boldsymbol{\mu_2}, \Sigma)] \quad (24)$$

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

$$E(\mathbf{w}) = -\ln L(\mathbf{w}) = -\sum_{n=1}^{N} \left[ y_n \ln \sigma\left(\mathbf{w^T \bar{x}}\right) + (1-y_n) \ln\left(1 - \sigma\left(\mathbf{w^T \bar{x}}\right)\right)\right] \quad (25)$$

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

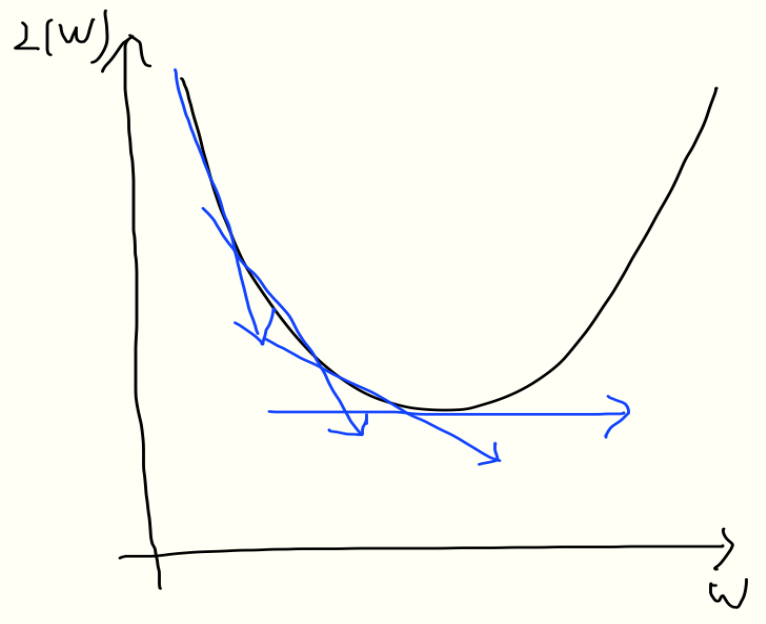
# Maximum Likelihood

$$-\frac{\partial E(\mathbf{w})}{\partial w} = -\sum_n y_n\left(\frac{\sigma\left(\mathbf{w}^{\mathbf{T}}\bar{\mathbf{x}}\right)\left(1-\sigma\left(\mathbf{w}^{\mathbf{T}}\bar{\mathbf{x}}\right)\right)\bar{\boldsymbol{x}}_n}{\sigma(\mathbf{w}^{\mathbf{T}}\bar{\mathbf{x}})} - \sum_n \frac{(1-y_n)\left(1-\sigma\left(\mathbf{w}^{\mathbf{T}}\bar{\mathbf{x}}\right)\right)\sigma\left(\mathbf{w}^{\mathbf{T}}\bar{\mathbf{x}}\right)(-\bar{\boldsymbol{x}}_n)}{1-\sigma(\mathbf{w}^{\mathbf{T}}\bar{\mathbf{x}})} \right. \quad (25\text{-}1)$$

$$0 = -\sum_n y_n\bar{x}_n - \sum_n y_n\sigma\left(\mathbf{w}^{\mathbf{T}}\bar{\mathbf{x}}\right)(\bar{\boldsymbol{x}}_n) + \sum_n \sigma\left(\mathbf{w}^{\mathbf{T}}\bar{\mathbf{x}}\right)\bar{x}_n + \sum_n y_n\sigma\left(\mathbf{w}^{\mathbf{T}}\bar{\mathbf{x}}\right)(\bar{\boldsymbol{x}}_n)$$

$$= \sum_n \left[\sigma\left(\mathbf{w}^{\mathbf{T}}\bar{\mathbf{x}}\right) - y_n\right]\bar{\boldsymbol{x}}_n \quad (25\text{-}2)$$

- Can we estimate **w**? No
- Then how? Use an iterative method



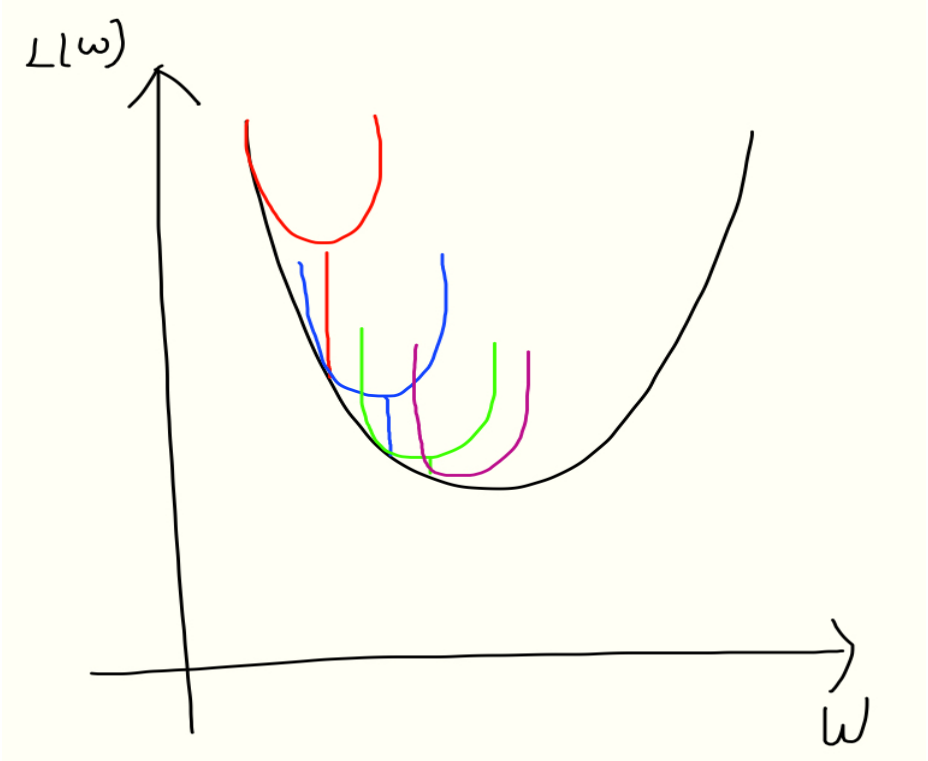
# Newton's Method

Iterative reweighted least square:

$$\mathbf{w} \leftarrow \mathbf{w} - \mathbf{H}^{-1}\nabla E(\mathbf{w}) \qquad (26)$$

where $\nabla E$ is the gradient (column vector) and H is the Hessian (matrix)

$$\mathbf{H} = \begin{bmatrix} \frac{\partial^2 E}{\partial^2 w_0} & \cdots & \frac{\partial^2 E}{\partial w_0 \partial w_m} \\ \vdots & \ddots & \vdots \\ \frac{\partial^2 E}{\partial w_m \partial w_0} & \cdots & \frac{\partial^2 E}{\partial^2 w_m} \end{bmatrix} \qquad (26\text{-}1)$$



# Hessian

$$\mathbf{H} = \nabla(\nabla \ln L(w)) \tag{27}$$

$$= \nabla\left(\sum_n \left[\sigma\left(\mathbf{w}^{\mathbf{T}}\bar{\mathbf{x}}\right) - y_n\right]\bar{\boldsymbol{x}}_{\boldsymbol{n}}\right)$$

$$= \sum_{n=1}^{N} \sigma\left(\mathbf{w}^{\mathrm{T}}\bar{\mathbf{x}}_n\right)\left(1 - \sigma\left(\mathbf{w}^{\mathbf{T}}\bar{\mathbf{x}}_n\right)\right)\bar{\mathbf{x}}_n\bar{\mathbf{x}}_n^{\mathrm{T}}$$

$$= \bar{\boldsymbol{X}}\boldsymbol{R}\bar{\boldsymbol{X}}^{T} \tag{27-1}$$

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

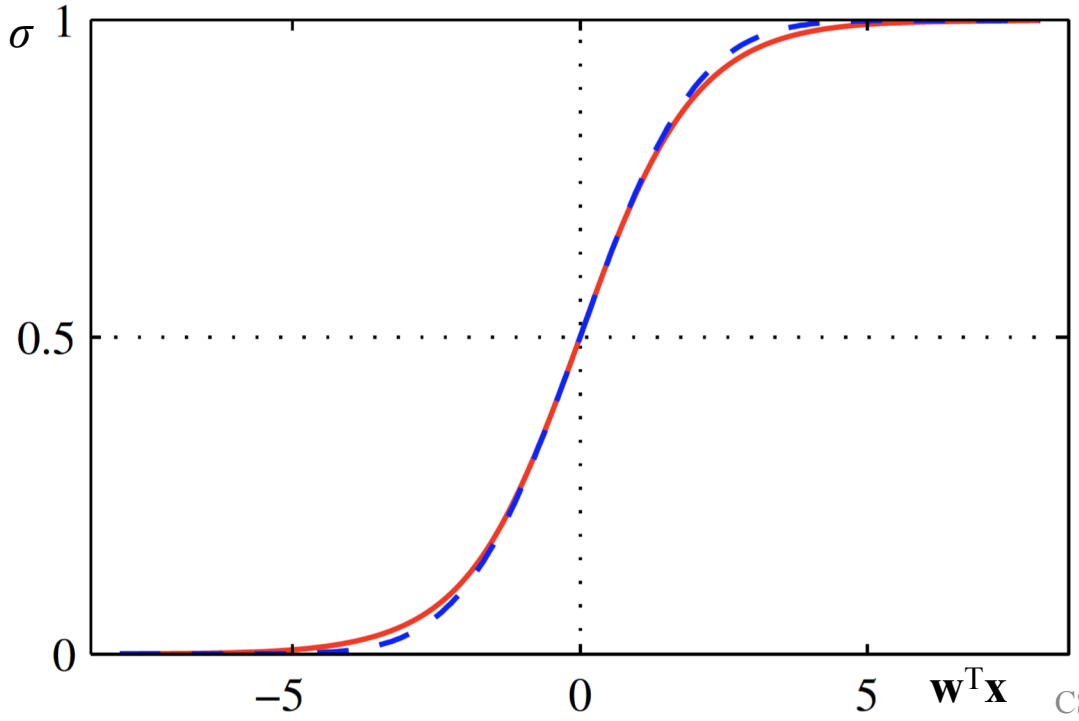
# Issues

**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 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.

Penalty Term

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

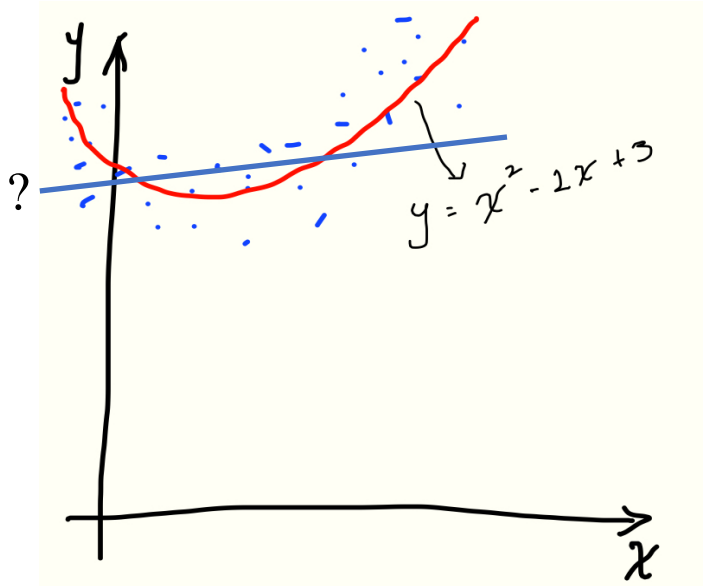
$$= \min_w \left\{ -\sum_{n=1}^{N} \left[ y_n \ln \sigma\left(\mathbf{w^T}\bar{\mathbf{x}}\right) + (1-y_n) \ln\left(1-\sigma\left(\mathbf{w^T}\bar{\mathbf{x}}\right)\right)\right]\right\} + \frac{1}{2}\lambda \mathbf{w^T w} \tag{28-1}$$

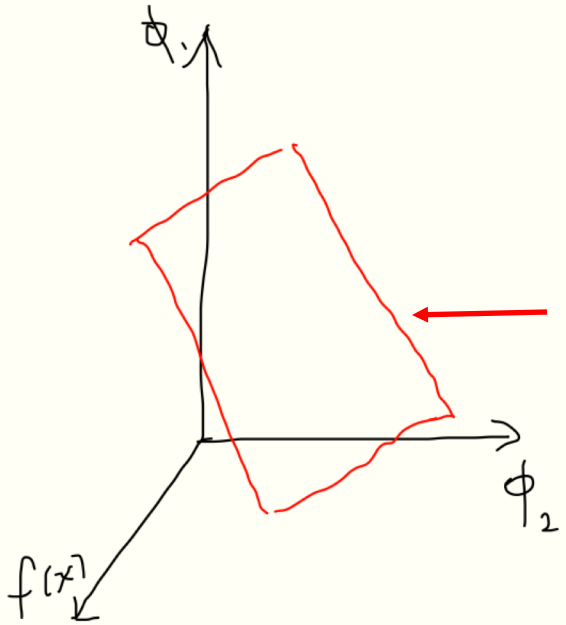
- Hessian: $$H = \bar{\mathbf{X}}\mathbf{R}\bar{\mathbf{X}}^T + \lambda\mathbf{I} \tag{29}$$
- The term $\lambda\mathbf{I}$ ensures that H is not singular (eigenvalues $\geq \lambda$)

# 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 $H = \{x \to \sum_i w_i \phi_i(x) \,|w_i \in \mathbb{R}\}$

← Linear hyperplane in the new space

# Kernel Methods

# Kernel Function

- Given: There are a large set of fixed linear basis functions.
- Problem: It is all depends on how complex the basis functions are.
- Solution: Use "dual trick" that is depends on the amount data than the function.

---

- We have a set of basis functions $\phi(\boldsymbol{x})$ 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.

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

We set the gradient to 0.

$$E(\mathbf{w}) = \sum_{n}\left(\mathbf{w}^T\phi(\boldsymbol{x}_n) - y_n\right)\phi(\boldsymbol{x}_n) + \lambda\mathbf{w} = 0 \tag{31}$$

$$\mathbf{w} = -\frac{1}{\lambda}\sum_{n}\left(\mathbf{w}^T\phi(\boldsymbol{x}_n) - y_n\right)\phi(\boldsymbol{x}_n) \tag{32}$$

w is linear combination of inputs in feature space

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

# Dual Representations

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

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

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

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)

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

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

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

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

Prediction $y_* = \phi(x_*)^T w = \phi(x_*)^T \Phi a = k(x_*, X)(K + \lambda I)^{-1} y \quad (34)$

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:

$k(x, x')$

$= ck_1(x, x')$ c is the constant > 0

$= f(x)k_1(x, x')f(x')$ f is the function

$= q\left(k_1(x, x')\right)$ q is the polynomial

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

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

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

$x^T Ax'$ A is symmetric positive semi-definite

$= k_a(x_a, x_a') + k_b(x_b, x_b')$ $k_a$ & $k_b$ are valid kernel

$= k_a(x_a, x_a')k_b(x_b, x_b')$

Other Application? sets, strings, graphs, etc…

Example -

Lodhi, Sunders, Shawe-Tylor, Christianini, Watkins,

**Text Classification Using Sting 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**)