COEN281 -- Introduction to Pattern Recognition and Data Mining

Lecture 5: Linear Discriminant Functions

Instructor: Dr. Giovanni Seni
GSeni@scu.edu

Department of Computer Engineering
Santa Clara University

Fall/18

Syllabus

Week 1	Introduction; R (Ch.1)
Week 2	Bayesian Decision Theory (Ch.2; DHS: 2.1-2.6, 2.9) Parameter Estimation (DHS: 3.1-3.4)
Week 3	**Linear Discriminant Functions** (Ch.3&4; DHS: 3.8.2, 5.1-5.8) Regularization (Ch.6; SE: Ch.3)
Week 4	Neural Networks (DHS: 6.1-6.6, 6.8); Deep Learning
Week 5	Support Vector Machines (Ch.9)
Week 6	Decision Trees (Ch. 8.1; DHS: 8.3; Ch 2 SE)
Week 7	Ensemble Methods (Ch. 8.2; SE: Ch 4, 5)
Week 8	Clustering (Ch. 10; DHS: 10.6, 10.7) Clustering (DHS: 10.9); How many clusters are there? (DHS: 10.10)
Week 9	Non-metric: Association Rules Collaborative Filtering
Week 10	Text Retrieval; Other topics

Overview

- Introduction
  - Approaches to building classifiers
  - Geometry of linear discriminant functions
- Linear separable case – Perceptron criteria
- Other methods
  - Linear Discriminant Analysis (LDA)
    - Restricted Gaussian classifier (see Lecture 2)
  - Linear Regression – Least Squares (LS) solution
  - Fisher's geometric view of LDA
  - Logistic Regression

Introduction
Building Classifiers

1) *Class-conditional* ("generative") approach
   - $p_{\theta_j}(\boldsymbol{x}|\omega_j)$ are modeled explicitly; $\hat{\theta}_j$ are estimated via ML
   - Combined with estimates of $P(\omega_j)$ are inverted via Bayes rule to arrive at $P(\omega_j|\boldsymbol{x})$
2) *Regression* approach
   - $P(\omega_j|\boldsymbol{x})$ are modeled explicitly
   - e.g., Logistic regression
3) *Discriminative* approach
   - Try to model the decision boundary directly – i.e., a mapping from inputs $\boldsymbol{x}$ to one of the classes
   - Assume we know the form for the discriminant functions $g_i(\boldsymbol{x})$

Introduction
Building Classifiers (2)

- Classification is an easier problem than density estimation (Vapnik)
  - Why use density estimation as an intermediate step?
  - Remember likelihood ratio:

$$\frac{p(\mathbf{x}\mid\omega_1)}{p(\mathbf{x}\mid\omega_2)} > \frac{\lambda_{12}-\lambda_{22}}{\lambda_{21}-\lambda_{11}} \times \frac{P(\omega_2)}{P(\omega_1)}$$

  $\Rightarrow$ we only need to know if $\frac{P(\omega_i)\cdot p(\mathbf{x}\mid\omega_i)}{P(\omega_j)\cdot p(\mathbf{x}\mid\omega_j)} > 1$

  - i.e., only ratios matter!

Introduction
Linear Discriminant Functions

- Definition
  - Just a linear combination of the measurements of $\boldsymbol{x}$ written as $g_\theta(x) = \boldsymbol{\theta}^t\boldsymbol{x} + \theta_0$
  - $\boldsymbol{\theta}$ is the "weight"[T] (or coefficient) vector of the model
  - θ_0 the "bias" or "threshold" weight
- Optimal if underlying distributions are "cooperative"
  - Gaussians with $\Sigma_i = \sigma^2 I$ or $\Sigma_i = \Sigma$ (LDA - see Lecture 2)
  - Simplicity makes them attractive for initial, trial classifiers
  - Can be generalized to be linear in some given set of functions $\varphi(\boldsymbol{x})$

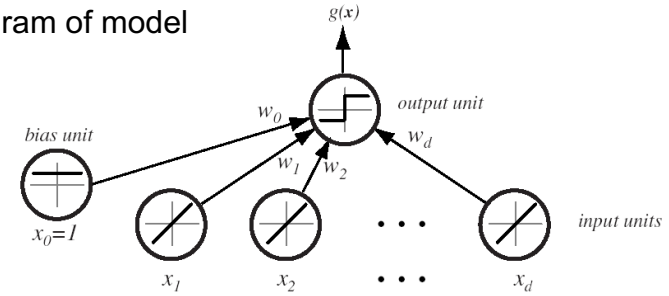
[T] sometimes we write $\boldsymbol{w}$ to refer to $\boldsymbol{\theta}$

Introduction
Linear Discriminant Functions (2)

- Decision rule - two-class case
  - Decide ω_1 if $g(\mathbf{x})>0$ and ω_2 if $g(\mathbf{x})<0$

    i.e., assign $\mathbf{x}$ to ω_1 if $\mathbf{w}^t\mathbf{x}$ exceeds threshold $-w_0$
  - If $g(\mathbf{x})=0$ assignment is undefined – i.e., can go either way

- Diagram of model



Introduction
Linear Discriminant Functions (3)

- Homogeneous form

$$g(\mathbf{x}) = w_0 + \sum_{i=1}^{d} w_i x_i = \sum_{i=0}^{d} w_i x_i \qquad \text{where } x_0 = 1$$

- Augmented weight & feature vector

$$\mathbf{a} = \begin{bmatrix} w_0 \\ w_1 \\ \vdots \\ w_d \end{bmatrix} \qquad \mathbf{x} = \begin{bmatrix} 1 \\ x_1 \\ \vdots \\ x_d \end{bmatrix}$$

- We write $g(\mathbf{x})=\mathbf{a}^t\mathbf{x}$

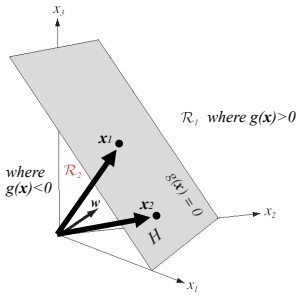
Introduction
Decision Surface

- Equation $g(\boldsymbol{x})=0$ defines surface that separates points assigned to the category ω_1 from points assigned to the category ω_2
  - $g(\boldsymbol{x})$ linear $\Rightarrow$ surface is a *hyperplane* H
  - Consider $\boldsymbol{x}_1$ and $\boldsymbol{x}_2$ both on the decision surface:

    $\boldsymbol{w}^t\boldsymbol{x}_1+w_0 = \boldsymbol{w}^t\boldsymbol{x}_2+w_0$

    or $\boldsymbol{w}^t(\boldsymbol{x}_1-\boldsymbol{x}_2) = 0$

    $\Rightarrow$ $\boldsymbol{w}$ is normal to any vector lying in the hyperplane
  - Orientation of H is determined by $\boldsymbol{w}$



Introduction
Decision Surface (2)

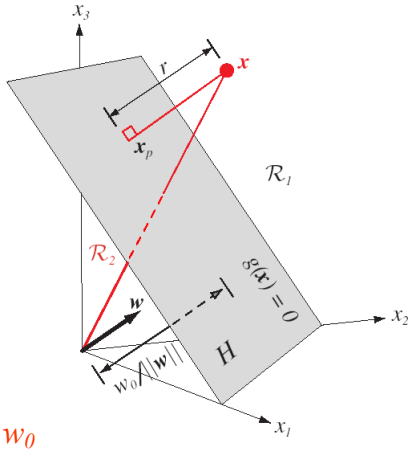
- $g(\boldsymbol{x}) \propto$ distance from $\boldsymbol{x}$ to H
  - Express $\boldsymbol{x}$ as $\boldsymbol{x} = \boldsymbol{x}_p + r\frac{\boldsymbol{w}}{\|\boldsymbol{w}\|}$

    because $g(\boldsymbol{x}_p)=0$

    $$g(x) = w^t x + w_0 = g(x_p) + r\frac{w^t w}{\|w\|}$$

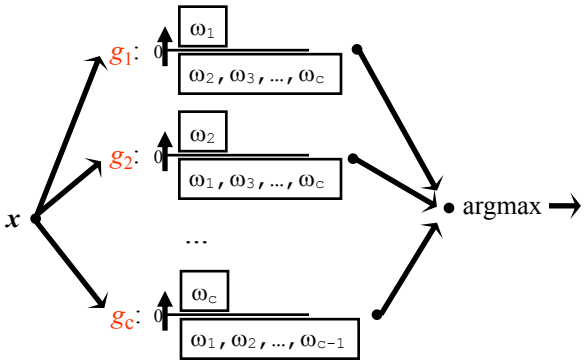
    $$= r\|w\|$$

    $$\Rightarrow r = \frac{g(x)}{\|w\|}$$
  - Also, $d(0,H) = w_0/\|w\|$
- Location of H is determined by w_0



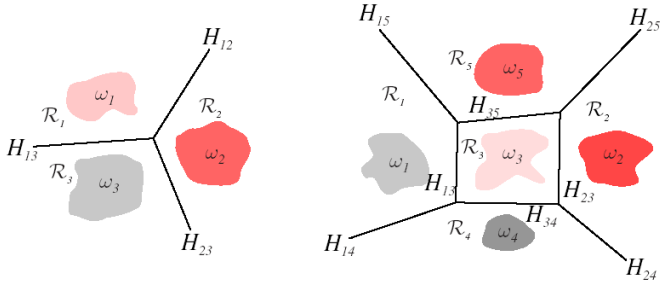
Introduction
Multiclass Case

- One per class decomposition (*linear machine*)
  - i.e., C discriminant functions
  - ω_i vs. $\neg\omega_i$



Introduction
Multiclass Case (2)

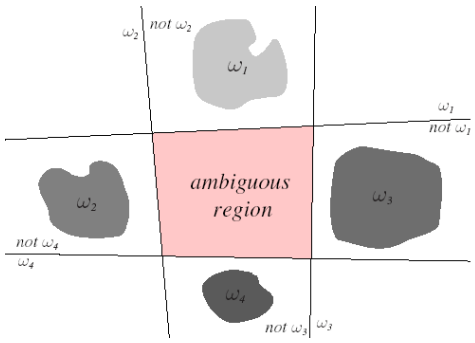
- Decision boundaries



  - H_{ij} defined by $g_i(\boldsymbol{x}) = g_j(\boldsymbol{x})$
  - Number of H_{ij} is often fewer than $c(c\text{-}1)/2$
  - Decision regions are convex and singly connected
    - Most suitable when $p(\boldsymbol{x}|\omega_j)$ is unimodal
      - *Many exceptions!*

Introduction
Multiclass Case (3)

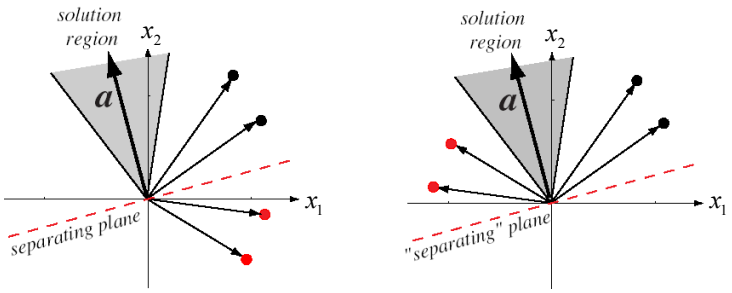
- Without *argmax*, ambiguous class assignments can arise



Linear Separable Case
Perceptron

- Simplifying normalization
  - Replace ω_2 samples by their negatives

    $\Rightarrow$ Find $\boldsymbol{a}$ such that $\boldsymbol{a}^t x > 0$ for all samples



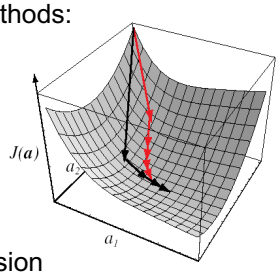
- Note that $\boldsymbol{a}$ is not unique!

Linear Separable Case
Perceptron (2)

- Score function
  - A scalar function $J(\boldsymbol{a})$ that is minimized if $\boldsymbol{a}$ is a solution vector
  - Allows use of *Gradient Descent* (search) methods:

$$\mathbf{a}(k+1) = \mathbf{a}(k) - \eta(k)\nabla J(\mathbf{a}) \quad \text{or}$$

$$\mathbf{a}(k+1) = \mathbf{a}(k) - \mathbf{H}^{-1}\nabla J(\mathbf{a}) \quad \text{(Newton)}$$



  - Idea 1: $J(\boldsymbol{a})$ is # of misclassified samples
  - Idea 2: $J_p(\boldsymbol{a})$ is $\propto$ to sum of distances to decision boundary

$$J_P(\mathbf{a}) = \sum_{\mathbf{x}\in Miss(\mathbf{a})} (-\mathbf{a}^t\mathbf{x})$$ where $Miss(\mathbf{a})$ is misclassified set

Linear Separable Case
Perceptron (3)

- Fixed-increment, single-sample

```
k ← 0
do {
  k ← k+1
  if (x^k is missclassified by a) {
    a ← a + x^k        → -∇J_p
  }
} until (all patterns are properly classified)
```

- Convergence Theorem – Perceptron algorithm is guaranteed to find a solution if samples are linearly separable
- In nonseparable case, error-correcting algorithm produces an infinite sequence $\boldsymbol{a}(k) \Rightarrow$ limited applicability

Linear Regression

Least Squares Error

- Class encoding: $y = 1 \;\; if \; class(\mathbf{x}) = \omega_1, \;$ else $\; y = -1$
- Model: $\hat{y} = \hat{w}_0 + \sum_{j=1}^{p} x_j \hat{w}_j = \mathbf{x}^t \hat{\mathbf{w}}$
- Score function: $J_s(\mathbf{w}) = RSS(\mathbf{w}) = \sum_{i=1}^{n} (y_i - \mathbf{x}_i^t \mathbf{w})^2$
  - Rationale - minimizing the size of the error vector $\mathbf{e} = \mathbf{Y} - \hat{\mathbf{Y}}$ (see Lecture 4)
- Note that $\mathbf{X}$ is rectangular and $\mathbf{w}$ is overdetermined
  - $\mathbf{Y} = \mathbf{X}\mathbf{w}$ ordinarily has no exact solution
- $J_s(\mathbf{w})$ is quadratic – we can look for a single global minimum ($\nabla J_s = 0$)

Linear Regression

Least Squares Error (2)

- Closed-form solution

$$\nabla J_s = -\sum_{i=1}^{n} 2(y_i - \mathbf{x}_i^t \mathbf{w})\mathbf{x}_i = -2\mathbf{X}^t(\mathbf{Y} - \mathbf{X}\mathbf{w})$$

$$\nabla J_s = 0 \quad \Rightarrow \quad \mathbf{X}^t\mathbf{Y} = \mathbf{X}^t\mathbf{X}\mathbf{w}$$

$$\hat{\mathbf{w}} = (\mathbf{X}^t\mathbf{X})^{-1}\mathbf{X}^t\mathbf{Y}$$

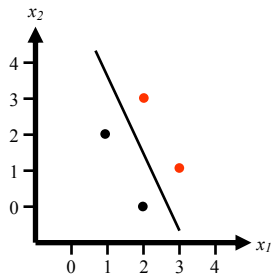
$$= \boxed{\mathbf{X}^+\mathbf{Y}}$$

- A more general definition of the *pseudoinverse* always exists: $\mathbf{X}^+ \equiv \lim_{\varepsilon \to 0} (\mathbf{X}^t\mathbf{X} + \varepsilon\mathbf{I})^{-1}\mathbf{X}^t$
- We expect to obtain a useful discriminant in both the separable and the nonseparable cases
  - When c is large, sensitive to "masking" problem (Hastie)

Linear Regression

Minimum Squared Error (3)

- Example



$$\Rightarrow \quad \mathbf{X} = \begin{bmatrix} 1 & 1 & 2 \\ 1 & 2 & 0 \\ 1 & 3 & 1 \\ 1 & 2 & 3 \end{bmatrix} \qquad \mathbf{Y} = \begin{bmatrix} -1 \\ -1 \\ 1 \\ 1 \end{bmatrix}$$

- In R: X.plus <-solve(t(X) %*% X) %*% t(X)

$$\mathbf{X}^{+} = (\mathbf{X}^t\mathbf{X})^{-1}\mathbf{X}^t = \begin{bmatrix} 5/4 & 13/12 & -3/4 & -7/12 \\ -1/2 & -1/6 & 1/2 & 1/6 \\ 0 & -1/3 & 0 & 1/3 \end{bmatrix} \Rightarrow \mathbf{X}^{+}\mathbf{Y} = \mathbf{w} = \begin{bmatrix} -11/3 \\ 4/3 \\ 2/3 \end{bmatrix}$$

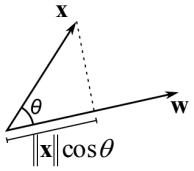
$$\Rightarrow \quad g(\mathbf{x}) = \mathbf{w}^t\mathbf{x} = -\frac{11}{3} + \frac{4}{3}x_1 + \frac{2}{3}x_2$$

Fisher Linear Discriminant

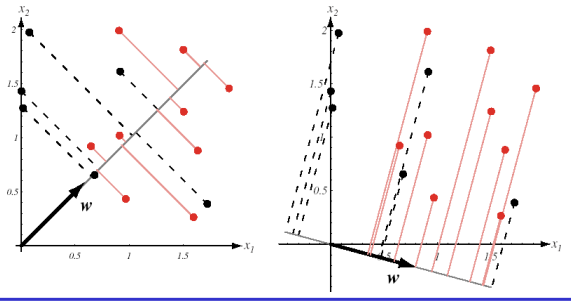
Low-Dimensional Projection

- Geometric interpretation of dot product
  - Length of the projection of $\mathbf{x}$ onto the (unit) vector $\mathbf{w}$

$$\mathbf{w}^t\mathbf{x} = \|\mathbf{w}\|\|\mathbf{x}\|\cos\theta$$



- Searching for the $\mathbf{w}$ that best separates the projected data



Fisher Linear Discriminant

Low-Dimensional Projection (2)

- Criterion function
  - Idea 1: use the distance between the projected sample means

$$\left|\tilde{m}_1 - \tilde{m}_2\right| = \left|\mathbf{w}^t(\mathbf{m}_1 - \mathbf{m}_2)\right| \qquad \text{where} \quad \mathbf{m}_i = \frac{1}{n_i}\sum_{\mathbf{x}\in D_i}\mathbf{x}$$

    - Dependent on ||**w**||… could be made arbitrarily large

  - Idea 2: maximize ratio of between-class scatter (as above) to within-class scatter

$$J_F(\mathbf{w}) = \frac{\left|\tilde{m}_1 - \tilde{m}_2\right|^2}{\tilde{S}_1^2 + \tilde{S}_2^2} \qquad \text{where} \quad S_i^2 = \sum_{\mathbf{x}\in D_i}(\mathbf{w}^t\mathbf{x} - \mathbf{w}^t\mathbf{m}_i)^2$$

    - Clearly, $(1/n)(\tilde{S}_1^2 + \tilde{S}_2^2)$ is an estimate of the variance of the pooled data

Fisher Linear Discriminant

Low-Dimensional Projection (3)

- **w** that optimizes $J_F()$ can be shown to be

$$\mathbf{w} = \mathbf{S}_w^{-1}(\mathbf{m}_1 - \mathbf{m}_2) \qquad \text{where} \quad \mathbf{S}_w = \mathbf{S}_1 + \mathbf{S}_2$$

$$\mathbf{S}_i = \sum_{\mathbf{x}\in D_i}(\mathbf{x} - \mathbf{m}_i)(\mathbf{x} - \mathbf{m}_i)^t$$

  - $\mathbf{S}_w$ is called the within class scatter matrix

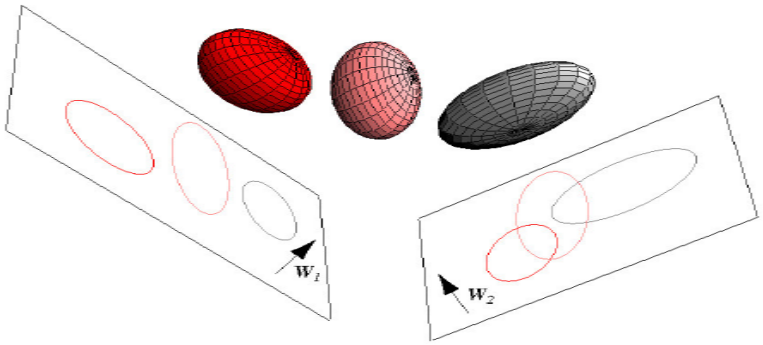
- Connection to LDA -- $p(\mathbf{x}|\omega_i) \sim N(\boldsymbol{\mu}_i, \boldsymbol{\Sigma})$

$$\begin{aligned} g(\mathbf{x}) &= g_i(\mathbf{x}) - g_j(\mathbf{x}) \\ &= (\mathbf{w}_i^t\mathbf{x} + w_{i0}) - (\mathbf{w}_j^t\mathbf{x} + w_{j0}) \\ &= \mathbf{x}^t\Sigma^{-1}(\mu_i - \mu_j) + (w_{i0} - w_{j0}) \quad \text{since } \mathbf{w}_i = \Sigma^{-1}\mu_i \end{aligned}$$ (see Lecture 2)

Fisher Linear Discriminant

Low-Dimensional Projection (4)

- For the c-class problem, $c-1$ functions are required
  - Projection is from a d to a $(c-1)$ dimensional space $(d > c)$



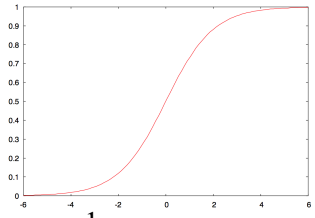
  - Sacrifice performance for the advantage of lower-dimensional space

Logistic Regression

Modeling Posteriors

- Model form: $P(\omega_1 \mid \mathbf{x}) = \phi(\beta_0 + \beta^t \mathbf{x})$ where ϕ is the "logistic" function

$$\phi(z) = \frac{e^z}{1+e^z} = \frac{1}{1+e^{-z}}$$



  - Two-class case: $P(\omega_2 \mid \mathbf{x}) = 1 - P(\omega_1 \mid \mathbf{x}) = \frac{1}{1+e^{\beta_0+\beta^t \mathbf{x}}}$

- Log of "odds ratio" is linear

$$\log \frac{P(\omega_1 \mid \mathbf{x})}{P(\omega_2 \mid \mathbf{x})} = \beta_0 + \beta^t \mathbf{x} \qquad \Rightarrow \text{decision boundaries are linear}$$

Logistic Regression
Fitting Model

- ϕ' is given by:

$$\phi'(z) = \frac{e^{-z}}{(1+e^{-z})^2} = \frac{e^{-z}}{1+e^{-z}}\frac{1}{1+e^{-z}} = \frac{1}{1+e^{z}}\frac{e^{z}}{1+e^{z}} = \phi(z)(1-\phi(z))$$

- Log-likelihood (two-class case)

$$l(\beta) = \sum_{i=1}^{n} b_i \ln P(\mathbf{x}_i;\beta) + (1-b_i)\ln(1-P(\mathbf{x}_i;\beta)) \qquad b_i = \begin{cases} 1 & x \in \omega_1 \\ 0 & \text{otherwise} \end{cases}$$

$$\partial l / \partial \beta_r = \sum_{i=1}^{n}\left(\frac{b_i}{P_i} - \frac{1-b_i}{1-P_i}\right)\phi'(\beta^t \mathbf{x}_i)x_{ir}$$

$$\partial l / \partial \boldsymbol{\beta} = \sum_{i=1}^{n}\left(\frac{b_i}{P_i} - \frac{1-b_i}{1-P_i}\right)P_i(1-P_i)\mathbf{x}_i = \sum_{i=1}^{n}[b_i(1-P_i) - P_i(1-b_i)]\mathbf{x}_i$$

$$= \sum_{i=1}^{n}(b_i - P_i)\mathbf{x}_i = \mathbf{X}^t(\mathbf{b}-\mathbf{P})$$

Logistic Regression
Fitting Model (2)

- Differentiating again to obtain the Hessian:

$$\partial^2 l / \partial\beta_s \partial\beta_r = \sum_{i=1}^{n}\partial\beta_s(b_i - P_i)x_{ir} = -\sum_{i=1}^{n}\phi'(\beta^t\mathbf{x}_i)x_{ir}x_{is} = -\sum_{i=1}^{n}P_i(1-P_i)x_{ir}x_{is}$$

$$\mathbf{H} = -\mathbf{X}^t\mathbf{W}\mathbf{X} \qquad \text{where } \mathbf{W} = \begin{pmatrix} P_1(1-P_1) & \cdots & 0 \\ \vdots & \ddots & \vdots \\ 0 & \cdots & P_n(1-P_n) \end{pmatrix}$$

- Newton steps is:

$$\boldsymbol{\beta}(k+1) = \boldsymbol{\beta}(k) - \mathbf{H}^{-1}\nabla J(\boldsymbol{\beta})$$
$$= \boldsymbol{\beta}(k) + [\mathbf{X}^t\mathbf{W}\mathbf{X}]^{-1}\mathbf{X}^t(\mathbf{b}-\mathbf{P})$$

Logistic Regression
Comparison to LDA

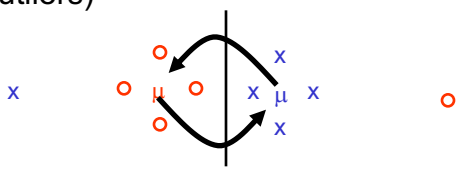
- We had $g(\mathbf{x}) = g_i(\mathbf{x}) - g_j(\mathbf{x}) = (\mathbf{w}_i^t\mathbf{x} + w_{i0}) - (\mathbf{w}_j^t\mathbf{x} + w_{j0})$
$= \mathbf{x}^t\Sigma^{-1}(\mu_i - \mu_j) + (w_{i0} - w_{j0}) \quad \text{since } \mathbf{w}_i = \Sigma^{-1}\mu_i$
$= \alpha_0 + \alpha^t\mathbf{x}$

- Simply note that $g(\mathbf{x}) = \log\dfrac{P(\omega_i \mid \mathbf{x})}{P(\omega_j \mid \mathbf{x})}$

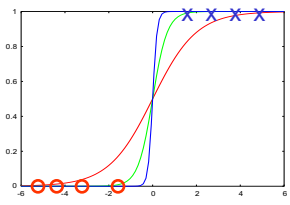
  – LR's $\boldsymbol{\beta}$ computed directly not via μ_i, μ_j, Σ
    - i.e., optimizing different criteria
  – LR holds also for some non-normal densities… it only needs the ratio to be of the logistic type
  – If $\boldsymbol{x}_i$ are normal, then LDA is 30% more efficient

Logistic Regression
Comparison to LDA (2)

- If $\boldsymbol{x}_i$ are not normal, then LDA can be much worse (e.g., extreme outliers)



- LR can be degenerate on separable data
  – Numerical issues when ||β|| = ∞



- In general, LR is a safer, more robust bet, but often similar results