Introduction to Pattern Recognition and Data Mining

Lecture 4: Linear Discriminant Functions

Instructor: Dr. Giovanni Seni

Department of Computer Engineering
Santa Clara University

Overview

- Introduction
  - Approaches to building classifiers
  - Linear discriminant functions: definition and surfaces
- Linear separable case – Perceptron criteria
- Other methods
  - Linear Discriminant Analysis (LDA)
    - Restricted Gaussian classifier (see Lecture 2)
  - Linear Regression -- Minimum Squared-Error (MSE) criteria
  - Fisher's geometric view of LDA
  - Logistic Regression

Introduction
Building Classifiers

- *Class-conditional* ("generative") approach
  - $p(\mathbf{x}|\omega_j, \theta_j)$ 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 $p(\omega_j|\mathbf{x})$
- *Regression* approach
  - $p(\omega_j|\mathbf{x})$ are modeled explicitly
  - e.g., Logistic regression
- *Discriminative* approach
  - Try to model the decision boundary directly – i.e., a mapping from inputs $\mathbf{x}$ to one of the classes
  - Assume we know the form for the discriminat functions $g_i(\mathbf{x})$

Introduction
Building Classifiers (2)

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

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

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

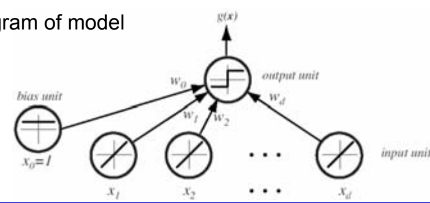
  - i.e., only ratios matter!

Introduction
Linear Discriminant Functions

- Definition
  - Just a linear combination of the measurements of $\boldsymbol{x}$ written as $g(\boldsymbol{x})=\boldsymbol{w}^t\boldsymbol{x}+w_0$
  - $\boldsymbol{w}$ is the "weight" vector of the model
  - w_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 $\varphi(\boldsymbol{x})$

Introduction
Linear Discriminant Functions (2)

- Decision rule - two-class case
  - Decide ω_1 if $g(\boldsymbol{x})>0$ and ω_2 if $g(\boldsymbol{x})<0$
  - i.e., assign $\boldsymbol{x}$ to ω_1 if $\boldsymbol{w}^t x$ exceeds threshold $-w_0$
  - If $g(\boldsymbol{x})=0$ 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

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

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

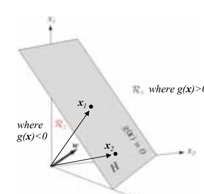
Introduction
Decision Surface

- Equation $g(\boldsymbol{x})=0$ defines surface that separates points assigned to the category ω_1 from points assigned to the category ω_2
  - $g(x)$ 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 $\boldsymbol{w}^t(\boldsymbol{x}_1-\boldsymbol{x}_2) = 0$

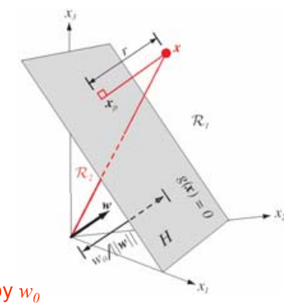
    $\Rightarrow$ $\boldsymbol{w}$ is normal to any vector lying in the hyperplane
  - Orientation of H is determined by $\boldsymbol{w}$



Introduction
Decision Surface (2)

- $g(\mathbf{x}) \propto$ distance from $\mathbf{x}$ to H
  - Express $\mathbf{x}$ as $\mathbf{x} = x_p + r\frac{w}{\|w\|}$
  - because $g(\mathbf{x}_p)=0$

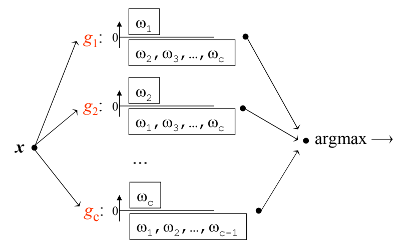
$$g(x) = w^t x + w_0 = g(x_p) + r\frac{w^t w}{\|w\|}$$
$$= r\|w\|$$
$$\Rightarrow r = \frac{g(x)}{\|w\|}$$
$$\Rightarrow d(0,H) = w_0 / \|w\|$$



- Location of H is determined by w_0

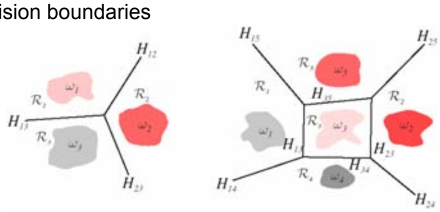
Introduction
Multiclass Case

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



Introduction
Multiclass Case (2)

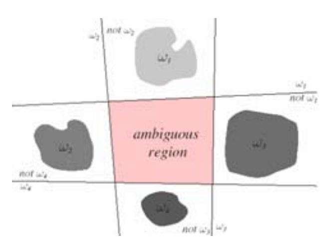
- Decision boundaries



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

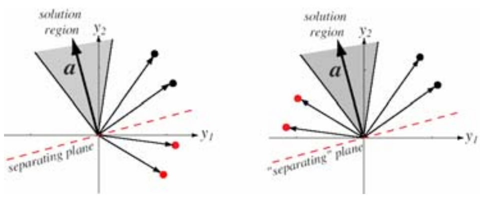
Introduction
Multiclass Case (3)

- 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



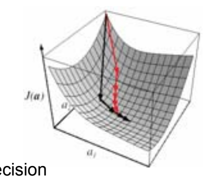
- Note that $\boldsymbol{a}$ is not unique!

Linear Separable Case
Perceptron (2)

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

$$J_p(\mathbf{a}) = \sum_{y \in Y}(-\mathbf{a}^t y) \quad \text{where } Y(\mathbf{a}) \text{ is misclassified set}$$

Linear Separable Case
Perceptron (3)

- Fixed-increment, single-sample

```
k ← 0
do {
  k ← k+1
  if (y^k is missclassified by a) {
    a ← a + y^k
  }
} 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
Minimum Squared Error

- Criterion function

$$J_s(\mathbf{a}) = \|\mathbf{Ya} - \mathbf{b}\|^2 = \sum_{i=1}^{n}(\mathbf{a}^t\mathbf{y}_i - b_i)^2$$

  - $\mathbf{Y}$ is $n \times (d+1)$ augmented data matrix
  - $\mathbf{b}$ indicator response vector (e.g., $b_i = 1$)
- Rationale - minimizing the size of the error vector $\mathbf{e} = \mathbf{Ya} - \mathbf{b}$
- Note that $\mathbf{Y}$ is rectangular and $\mathbf{a}$ is overdetermined
  - $\mathbf{Ya} = \mathbf{b}$ ordinarily has no exact solution
- $J_s(\boldsymbol{a})$ is quadratic – we can look for a single global minimum ($\nabla J_s = 0$)

Linear Regression
Minimum Squared Error (2)

- Closed-form solution

$$\nabla J_s = \sum_{i=1}^{n} 2(\mathbf{a}^t\mathbf{y}_i - b_i)\mathbf{y}_i = 2\mathbf{Y}^t(\mathbf{Ya}-\mathbf{b})$$

$$\nabla J_s = 0 \quad \Rightarrow \quad \mathbf{Y}^t\mathbf{Ya} = \mathbf{Y}^t\mathbf{b}$$

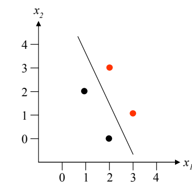
$$\mathbf{a} = (\mathbf{Y}^t\mathbf{Y})^{-1}\mathbf{Y}^t\mathbf{b}$$

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

- A more general definition of the *pseudoinverse* always exists: $\mathbf{Y}^+ \equiv \lim_{\varepsilon\to 0}(\mathbf{Y}^t\mathbf{Y}+\varepsilon\mathbf{I})^{-1}\mathbf{Y}^t$
- 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



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

- In R: Y.pi <-solve(t(Y) %*% Y) %*% t(Y)

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

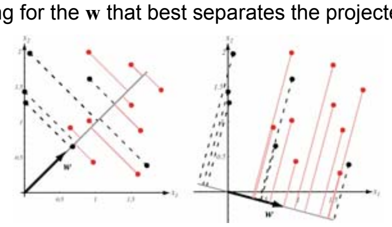
$$\Rightarrow \; g(\mathbf{x}) = \mathbf{a}^t\mathbf{y} = \frac{11}{3} - \frac{4}{3}x_1 - \frac{2}{3}x_2$$

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

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

    - Dependent on $\|\mathbf{w}\|$… could be made arbitrarily large
  - Idea 2: maximize ratio of between-class scatter (as above) to within-class scatter

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

    - 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

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

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

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

$$g(\mathbf{x}) = g_i(\mathbf{x}) - g_j(\mathbf{x}) = (\mathbf{w}_i^t\mathbf{x} + w_{i0}) - (\mathbf{w}_j^t\mathbf{x} + w_{j0})$$

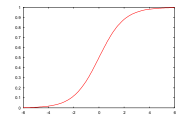
$$= \mathbf{x}^t \underbrace{\Sigma^{-1}(\mu_i - \mu_j)} + (w_{i0} - w_{j0}) \quad \text{since } \mathbf{w}_i = \Sigma^{-1}\mu_i$$

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

- 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)

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

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

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

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

Logistic Regression
Fitting Model (2)

- Differentiating again to obtain the Hessian:

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

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

- 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 $g(\mathbf{x}) = g_i(\mathbf{x}) - g_j(\mathbf{x}) = (\mathbf{w}_i^t\mathbf{x} + w_{i0}) - (\mathbf{w}_j^t\mathbf{x} + w_{j0})$

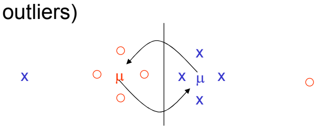
$$= \mathbf{x}^t\Sigma^{-1}(\mu_i - \mu_j) + (w_{i0} - w_{j0}) \quad \text{since } \mathbf{w}_i = \Sigma^{-1}\mu_i$$

$$= \alpha_0 + \alpha^t\mathbf{x}$$

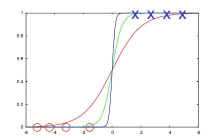
- Simply note that $g(\mathbf{x}) = \log \dfrac{P(\omega_i \mid \mathbf{x})}{P(\omega_j \mid \mathbf{x})}$
  - LR's β computed directly not via μ_i, μ_j, Σ
    - i.e., optimizing different criteria
  - LR holds also for some non-normal densities… it only needs the ratio to be of the logistic type
  - If x_i are normal, then LDA is 30% more efficient

Logistic Regression
Comparison to LDA (2)

- If 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 $||\beta|| = \infty$



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