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
  - \( p(x|\omega_j, \theta) \) are modeled explicitly; \( \theta \) are estimated via ML
  - Combined with estimates of \( p(\omega_j) \) are inverted via Bayes rule to arrive at \( p(\omega_j|x) \)
- **Regression approach**
  - \( p(\omega_j|x) \) are modeled explicitly
    - e.g., Logistic regression
- **Discriminative approach**
  - Try to model the decision boundary directly – i.e., a mapping from inputs \( x \) to one of the classes
  - Assume we know the form for the discriminant functions \( g_i(x) \)

Introduction

Building Classifiers (2)

- Classification is an easier problem than density estimation (Vapnik)
  - Why use density estimation as an intermediate step?
  - Remember likelihood ratio:
    \[
    \frac{p(x|\omega_1)}{p(x|\omega_2)} > \frac{\lambda_{12} - \lambda_{22}}{\lambda_{22} - \lambda_{11}} \times \frac{p(\omega_2)}{p(\omega_1)}
    \]
  - \( \Rightarrow \) we only need to know if \( \frac{p(\omega_1|x, \theta)}{p(\omega_2|x, \theta)} \)
  - i.e., only ratios matter!
Introduction
Linear Discriminant Functions

• Definition
  – Just a linear combination of the measurements of \( x \) written as
    \[ g(x) = w^T x + w_0 \]
  – \( w \) is the “weight” vector of the model
  – \( w_0 \) the “bias” or “threshold” weight

• Optimal if underlying distributions are “cooperative”
  – Gaussians with \( \Sigma_i \neq \Sigma_j \) 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 \( \phi(x) \)

Introduction
Linear Discriminant Functions (2)

• Decision rule - two-class case
  – Decide \( \omega_1 \) if \( g(x) > 0 \) and \( \omega_2 \) if \( g(x) < 0 \)
  – i.e., assign \( x \) to \( \omega_0 \) if \( w^T x \) exceeds threshold \( -w_0 \)
  – If \( g(x) = 0 \) assignment is undefined – i.e., can go either way

• Diagram of model

Introduction
Linear Discriminant Functions (3)

• Homogeneous form
  \[ g(x) = w_0 + \sum_{i=1}^{d} w_i x_i = \sum w_i x_i \quad \text{where } x_0 = 1 \]

• Augmented weight & feature vector
  \[ a = \begin{bmatrix} w_0 \\ w_1 \\ \vdots \\ w_d \end{bmatrix}, \quad y = \begin{bmatrix} 1 \\ x_1 \\ \vdots \\ x_d \end{bmatrix} \]

• We write \( g(x) = a^T y \)
**Introduction**

**Decision Surface (2)**

- \( g(x) \propto \text{distance from } x \text{ to } H \)
  - Express \( x \) as \( x = x_p + r \frac{w}{||w||} \)
  - because \( g(x_p) = 0 \)
    \[
g(x) = w'x + w_0 = g(x_p) + r w'w \]
    \[
    = ||w|| \Rightarrow r = \frac{g(x)}{||w||}
    \Rightarrow d(0, H) = ||w||
    \]
- Location of \( H \) is determined by \( w_0 \)

**Multiclass Case**

- One per class decomposition (linear machine)
  - i.e., \( C \) discriminant functions
    - \( \omega_i vs. \neg \omega_i \)
  - Decision boundaries \( H_{ij} \) defined by \( g_i(x) = g_j(x) \)
    - Number of \( H_{ij} \) is often fewer than \( C(C-1)/2 \)
    - Decision regions are convex and singly connected
      - Most suitable when \( p(x|\omega_j) \) is unimodal
        - Many exceptions!

**Multiclass Case (2)**

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

**Multiclass Case (3)**

- Without \emph{argmax}, ambiguous class assignments can arise
Linear Separable Case
Perceptron

- Simplifying normalization
  - Replace $\omega$ samples by their negatives
    \[ \Rightarrow \text{Find } a \text{ such that } ax > 0 \text{ for all samples} \]

- Note that $a$ is not unique!

Linear Separable Case
Perceptron (2)

- Criterion function
  - A scalar function $J(a)$ that is minimized if $a$ is a solution vector
  - Allows use of Gradient Descent methods:
    \[ a(k+1) = a(k) - \eta(k) \nabla J(a) \quad \text{or} \quad a(k+1) = a(k) - \mathbf{H}^{-1} \nabla J(a) \quad \text{(Newton)} \]
  - Idea 1: $J(a)$ is # of misclassified samples
  - Idea 2: $J_p(a) \propto \text{sum of distances to decision boundary}$

\[ J_p(a) = \sum_{i \in \text{Y(a) is misclassified set}} (a^T y_i) \]

Linear Separable Case
Perceptron (3)

- Fixed-increment, single-sample
  \[
  \begin{align*}
  k &\leftarrow 0 \\
  \text{do} & \{ \\
  k &\leftarrow k + 1 \\
  \text{if } (y^i \text{ is misclassified by } a) & \{ \\
  a &\leftarrow a + y^i \\
  \} \}
  \text{until (all patterns are properly classified)}
  \end{align*}
  \]

- 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 $a(k)$ ⇒ limited applicability

Linear Regression
Minimum Squared Error

- Criterion function
  \[ J(a) = ||Ya - b||^2 = \sum_{i=1}^{n} (a^T y_i - h_i)^2 \]
  - $Y$ is $n \times (d+1)$ augmented data matrix
  - $b$ indicator response vector (e.g., $b=1$)

- Rationale - minimizing the size of the error vector $e = Ya - b$

- Note that $Y$ is rectangular and $a$ is overdetermined
  - $Ya = b$ ordinarily has no exact solution

- $J(a)$ is quadratic — we can look for a single global minimum ($\nabla J = 0$)
Linear Regression
Minimum Squared Error (2)

- Closed-form solution
  \[ \nabla J = \sum_{i=1}^{n} 2(a^T y_i - b) y_i = 2Y^T (Ya - b) \]
  \[ \nabla J = 0 \Rightarrow Y^T a = Y^T b \]
  \[ a = (Y^T Y)^{-1} Y^T b \]
  \[ = Y^T b \]

- A more general definition of the pseudoinverse always exists: \( Y^+ = \lim_{\varepsilon \to 0} (Y^T Y + \varepsilon I)^{-1} Y^T \)

- We expect to obtain a useful discriminant in both the separable and the nonseparable cases
  - When \( c \) is large, sensitive to "masking" problem (Hastie)

Fisher Linear Discriminant
Low-Dimensional Projection

- Geometric interpretation of dot product
  - Length of the projection of \( x \) onto the (unit) vector \( w \)
    \[ w^T x = \| w \| \| x \| \cos \theta \]

- Searching for the \( w \) that best separates the projected data

Linear Regression
Minimum Squared Error (3)

- Example
  - \( X = \begin{bmatrix} 1 & 2 \\ 2 & 0 \\ 3 & 1 \\ 2 & 3 \end{bmatrix} \Rightarrow Y^+ = \begin{bmatrix} 1 & 1 & 2 \\ 1 & 2 & 0 \\ -1 & -3 & -1 \\ -1 & -2 & -3 \end{bmatrix} \)

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

Fisher Linear Discriminant
Low-Dimensional Projection (2)

- Criterion function
  - Idea 1: use the distance between the projected sample means
    \[ \| \tilde{m}_l - \tilde{m}_i \| = \| w^T (m_l - m_i) \| \]
    where \( m_l = \frac{1}{n_l} \sum x \in D_l x \)

    - Dependent on \( |w| \) could be made arbitrarily large

  - Idea 2: maximize ratio of between-class scatter (as above) to within-class scatter
    \[ J_1(w) = \frac{\| \tilde{m}_l - \tilde{m}_i \|^2}{S_1 + S_i} \]
    where \( S_1 = \sum_{x \in D_1} (w^T x - \tilde{m}_i)^2 \)

    - Clearly, \( 1/n (S_1^2 + S_i^2) \) 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
  \[
  w = S_1^{-1}(m_1 - m_2)
  \]
  where \( S_0 = S_1 + S_2 \)
  \[
  S_1 = \sum_{i \in I} (x_i - m_1)(x_i - m_2)'
  \]

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

• 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: \( P(\omega | x) = \phi(\beta_\omega + \beta' x) \) where \( \phi \) is the "logistic" function

\[
\phi(z) = \frac{e^z}{1 + e^z}
\]

– Two-class case: \( P(\omega_2 | x) = 1 - P(\omega_1 | x) = \frac{1}{1 + e^{\beta_1 x}} \)

• Log of “odds ratio” is linear

\[
\log \frac{P(\omega_1 | x)}{P(\omega_2 | x)} = \beta_1 + \beta' x \quad \Rightarrow \text{decision boundaries are linear}
\]
Logistic Regression
Comparison to LDA

• We had 
\[ g(x) = g_1(x) - g_2(x) = (w_1'x + w_{10}) - (w_2'x + w_{20}) \]
\[ = x\Sigma^{-1}(\mu_i - \mu_j) + (w_{10} - w_{20}) \]
\[ = \alpha' + \alpha'x \]

• Simply note that 
\[ g(x) = \log \frac{P(\omega_i|x)}{P(\omega_j|x)} \]

  – LR’s \( \beta \) computed directly not via \( \mu_i, \mu_j, \Sigma \)
  
  • 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 \( x \) are normal, then LDA is 30% more efficient

Logistic Regression
Comparison to LDA (2)

• If \( x \) 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