Reading
- Linear Discriminant Functions
  - DHS Chap. 5.5-5.8
- Review of vector derivative and chain rule
- Discriminant Functions with Higher Dimensions
  - DHS Chap. 5.3

Grading options
- Option A: complete HW#5-8, no project required
- Option B: complete a project on image classification, no more HWs
- Final exam required for either option

Class schedules
- No classes on
  - 10/31 (M), 11/7 (M, Uni. Holiday), 11/9 (W), 11/14 (M)
- Long lectures (start at 12 noon)
  - 11/2 (W), 11/16 (W), 11/21 (M)
Linear Discriminant Classifiers

\[ g(x) = w'x + w_0 \Rightarrow \text{find weight } w \text{ and bias } w_0 \]

- Augmented Vector
  \[ y = \begin{bmatrix} 1 \\ x_1 \\ \vdots \\ x_d \end{bmatrix}, \quad a = \begin{bmatrix} w_0 \\ w_1 \\ \vdots \\ w_d \end{bmatrix} \Rightarrow g(x) = g(y) = a'y \]

map \( y \) to class \( \omega_1 \) if \( g(y) > 0 \), otherwise class \( \omega_2 \)

distance from \( y \) to boundary in \( y \) space:
\[ r = \frac{g(y)}{\|w\|} \]

- Normalization
  \( \forall y_i \text{ in class } \omega_2, y_i \leftarrow -(y_i) \)
- Design Objective
  \[ a'y > b, \quad \forall y_i \]
  - Each \( y_i \) defines a half plane in the weight space (a).
  - Note we search weight solutions in the a-space

---

Gradient Decent Search with Different Criterion Functions

- \( J_p(a) = \sum_{y \in Y_p} (-a'y) \)
- \( J_q(a) = \sum_{y \in Y_q} (a'y)^2 \)
- \( J_r(a) = \frac{1}{2} \sum_{y \in Y_r} (a'y - b)^2 / \|b\|^2 \)

- 
  - \# misclassified
  - \( GD \) not applicable
  - Not differentiable
  - Smooth, but solutions may be trapped to boundaries
  - Solutions moved away from boundaries
Example: GD based on perceptron criterion

\[ J_p(a) = \sum_{y \in Y} (-a'y), \quad \text{where } Y \text{ is the set of misclassified samples} \]

\[ \nabla J_p(a) = \sum_{y \in Y} (-y) \quad \text{GD: } a(k+1) = a(k) - \eta(k)\nabla J(a(k)) \]

- **Batch Perceptron Update**
  - Initialize \( a(1) \), choose rate \( \eta(\cdot) \), and stop criterion \( \theta \)
  - Loop \( a(k+1) = a(k) + \eta(k) \sum_{y \in Y} y \)
  - Add sum of misclassified samples

- **Theorem:**
  - If samples are separable, then a solution can always be found within finite steps.

Relaxation Procedure

- **Problems with Quadratic Criterion**
  - Too smooth, solution trapped at boundaries
  - Dominated by large mis-classified sample

- **Relaxation Criterion**

\[ J_q(a) = \frac{1}{2} \sum_{y \in Y} a'y^2 b \]

\[ \nabla J_q(a) = \sum_{y \in Y} \frac{(a'y - b)}{\|y\|^2} y \]

- **Gradient Decent with single sample** \( y^k \)

\[
\begin{align*}
a(k+1) &= a(k) + \eta(k) \left( b - a'(k)y^k \right) y^k \\
&= a(k) + \eta(k) \left( \frac{b - a'(k)y^k}{\|y^k\|^2} y^k \right) y^k
\end{align*}
\]

- **Move** \( a \) **towards boundary** \( a'(k)y^k = b \)
  - \( 0 < \eta < 1 \): underrelaxation
  - \( 1 < \eta < 2 \): overrelaxation
Vector Derivative (Gradient) and Chain Rule

Consider scalar function of vector input: \( J(x) \)

- **Vector derivative (gradient)** \( \nabla_x J(x) = [\partial J / \partial x_1, \partial J / \partial x_2, \ldots, \partial J / \partial x_j] \)

- **Inner product** \( J = a^t b = \sum_k a_k b_k \quad \partial J / \partial a_i = b_i \)

  \[ \Rightarrow \nabla_x a^t b = b \quad \nabla_x a^t b = \nabla_x b^t a = a \]

- **Hermitian** \( J = x^t A x = \sum_i \sum_j x_i A_{ij} x_j \Rightarrow \nabla_x x^t A x = A x + A^t x \)

  if \( A \) is symmetric, then \( \nabla_x J = 2A x \)

  if \( A = I \), then \( \nabla_x J = 2x \)

- **Generalized chain rule**

  now consider \( x = A x' \), i.e. \( x_i = \sum_j A_{ij} x_j' \Rightarrow \delta x_i / \delta x_j' = A_{ij} \)

  \[ \nabla_x J = \begin{pmatrix} \delta x_j' \\ \delta x_j' \end{pmatrix} \nabla_{x'} J \Rightarrow \nabla_x J = A' \nabla_{x'} J \]

Example of gradient chain rule

if \( x = A x' \) then \( \nabla_x J = A' \nabla_{x'} J \)

example (mean squared error) \( J = \| y - b \|^2 = (y - b)^t (y - b) \)

Let \( x = y - b \), \( x' = a \)

\[ \Rightarrow x = y x' - b, \quad \nabla_x J = y' \nabla_{x'} J \] **chain rule of gradient**

note \( J_x = x' x \) \[ \Rightarrow \nabla_x J = 2x = 2(y - b) \]

\[ \Rightarrow \nabla_{x'} J = y' \nabla_{x'} J = 2y'(y - b) \]

\[ \therefore \nabla_{x'} J = 2y'(y - b) \]
**Minimal Squared-Error Solution**

Training sample matrix

\[
Y = \begin{bmatrix}
y_1' \\
y_2' \\
\vdots \\
y_n'
\end{bmatrix}
\]

Objective: \( a'y_i = b \), \( \forall y_i \)

\[\Rightarrow \text{define } J_s = \sum_{i=1}^{n} (a'y_i - b)^2 \]

\[= \|Ya - b\|^2 = (Ya - b)(Ya - b) \]

\[
\nabla_a J_s = 2Y'(Ya - b) = 0 \implies Y'Ya = Y'b
\]

if \( Y'Y \) is nonsingular \( \Rightarrow a = (Y'Y)^{-1}Y'b = Y^\dagger b \)

\[Y^\dagger = (Y'Y)^{-1}Y'\]  pseudo-inverse : \((d+1) \times n\)

**Example**

Training samples:

class \( \omega_1 : (1,2)' \), \( (2,0)' \) class \( \omega_2 : (3,1)' , (2,3)' \)

\[Y = \begin{bmatrix}
1 & 1 & 2 \\
1 & 2 & 0 \\
-1 & -3 & -1 \\
-1 & -2 & -3
\end{bmatrix}
\quad b = \begin{bmatrix}
1 \\
1 \\
1 \\
1
\end{bmatrix}
\]

find \( Y^\dagger \), then compute \( a^* = Y^\dagger b \)

(see figure in textbook)

---

**Generalized Linear Discriminant Functions**

- Include more than just the linear terms

\[g(x) = w_0 + \sum_{j=1}^{d} w_j x_j + \sum_{j=1}^{d} \sum_{j=1}^{d} w_{ij} x_j x_j = w_0 + x'Wx \]

- Shape of decision boundary

- ellipsoid, hyperhyperboloid, lines etc

- In general \( g(x) = \sum_{i=1}^{d} a_i y_i(x) = a'y \)

**Example**

\[g(x) = a_1 + a_2 x + a_3 x^2 \quad g(x) = a_1 x_1 + a_2 x_2 + a_3 x_1 x_2 \]

\[= \begin{bmatrix} a_1 & a_2 & a_3 \end{bmatrix} \begin{bmatrix} 1 & x & x^2 \end{bmatrix}' \]

**SVM**

- learning all the parameters is hard (curse of dim.)
- instead, try to maximize margins