Linear Discriminant Functions

Jacob Hays
Amit Pillay
James DeFelice

5.8, 5.9, 5.11

Minimum Squared Error

- Previous methods only worked on linear separable cases, by looking at misclassified samples to correct error
- MSE looks at all samples, using linear equations to find estimate

Minimum Squared Error

- **x** space mapped to **y** space.
- For all samples $\mathbf{x_i}$ in dimension d, there exists a $\mathbf{y_i}$ of dimension d^
- Find vector **a** making all $\mathbf{a^t y_i > 0}$
- All samples $\mathbf{y_i}$ in matrix **Y**, dim n x d^,
- **Ya = b** (b is vector of positive constants)
- b is our margin for error

$$\begin{pmatrix} y_{10} & y_{11} & \dots & y_{1d} \\ y_{20} & y_{21} & \dots & y_{2d} \\ \dots & \dots & & \dots \\ y_{n0} & y_{n1} & \dots & y_{nd} \end{pmatrix} \begin{pmatrix} a_0 \\ a_1 \\ . \\ a_d \end{pmatrix} = \begin{pmatrix} b_1 \\ b_2 \\ \dots \\ \dots \\ b_n \end{pmatrix}$$

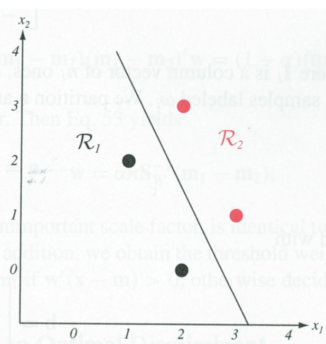
Minimum Squared Error

- **Y** is rectangular (n x d^), so it does not have a direct inverse to solve **Ya = b**
- **Ya – b = e** – gives error, minimize it
- Square error $\|e\|^2$ $\quad J_s(a) = \|Ya - b\|^2 = \sum_{i=1}^{n} (a^t y_i - b_i)^2$
- Take Gradient $\quad \nabla J_s = \sum_{i=1}^{n} 2(a^t y_i - b_i) y_i = 2Y^t (Ya - b)$
- Gradient should goto Zero $\quad Y^t Y a = Y^t b$

Minimum Squared Error

- $\mathbf{Y^tYa = Y^tb}$ goes to $\mathbf{a = (Y^tY)^{-1}Y^tb}$
- $\mathbf{(Y^tY)^{-1}Y^t}$ is the psuedo-inverse of Y, dimension d^ x n, can be written as $\mathbf{Y^\dagger}$
- $\mathbf{Y^\dagger Y = I}$ $\quad\quad$ $\mathbf{YY^\dagger \neq I}$
- $\mathbf{a = Y^\dagger b}$ gives us a solution with b being a margin.

Minimum Squared Error



Four training points and the decision boundary $\mathbf{a}^t \begin{pmatrix} 1 \\ x_1 \\ x_2 \end{pmatrix} = 0$, where $\mathbf{a}$ was found by means of a pseudoinverse technique.

Our matrix $\mathbf{Y}$ is therefore

$$\mathbf{Y} = \begin{pmatrix} 1 & 1 & 2 \\ 1 & 2 & 0 \\ -1 & -3 & -1 \\ -1 & -2 & -3 \end{pmatrix}$$

Fisher's Linear Discriminant

- Based on projection of d-dimensional data onto a line.
- Loses a lot of data, but some orientation of the line might give a good split

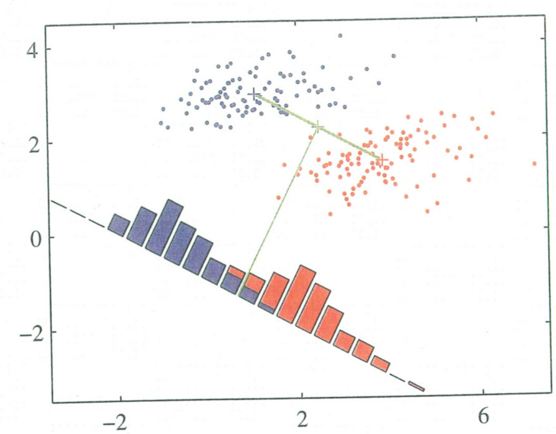
$$y = \mathbf{w}^t\mathbf{x}, \quad \|w\| = 1$$

- y_i is projection of x_i onto line **w**
- **Goal:** Find best **w** to separate them
- Highly overlapping data performs poorly

Fisher's Linear Discriminant

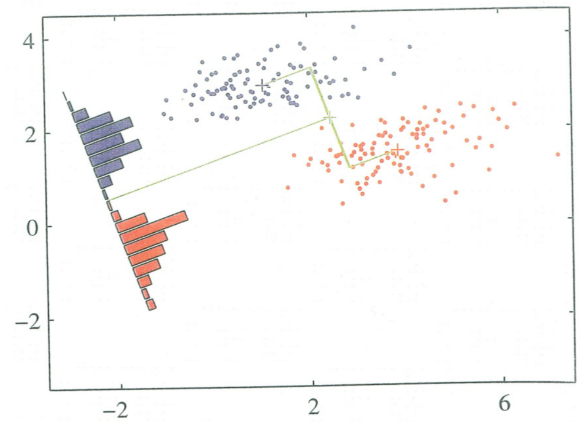
- Mean of each class D_i
- $w = m_1 - m_2 / \| m_1 - m_2 \|$

$$m_i = \frac{1}{n_i} \sum_{x \in D_i} x$$



Fisher's Linear Discriminant

- Scatter Matrices $S_i = \sum_{x \in D_i} (x - m_i)(x - m_i)^t$
- $S_W = S_1 + S_2$ $w = S_W^{-1}(m_1 - m_2)$



Fisher's Relation to MSE

- MSE and Fisher equivalent for specific b
  - n_i = number of $x \in D_i$
  - 1_i is column vector of n_i full of ones

$$Y = \begin{bmatrix} 1_1 & X_1 \\ -1_2 & -X_2 \end{bmatrix} \quad a = \begin{bmatrix} w_0 \\ \mathbf{w} \end{bmatrix} \quad b = \begin{bmatrix} \frac{n}{n_1} 1_1 \\ \frac{n}{n_2} 1_2 \end{bmatrix}$$

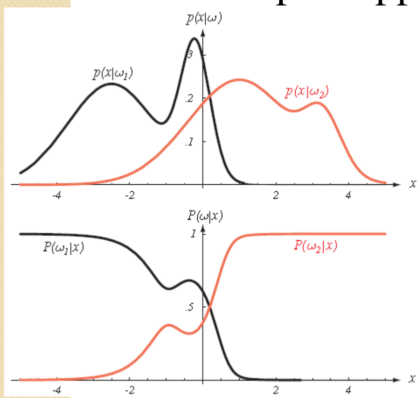
- Plug into $\mathbf{Y^tYa = Y^tb}$

$$\begin{bmatrix} 1_1 & -1_1 \\ X_1^t & -X_2^t \end{bmatrix} \begin{bmatrix} 1_1 & X_1 \\ -1_2 & -X_2 \end{bmatrix} \begin{bmatrix} w_0 \\ \mathbf{w} \end{bmatrix} = \begin{bmatrix} 1_1 & -1_1 \\ X_1^t & -X_2^t \end{bmatrix} \begin{bmatrix} \frac{n}{n_1} 1_1 \\ \frac{n}{n_1} 1_2 \end{bmatrix}$$

$$w = \alpha n S_W^{-1}(m_1 - m_2)$$

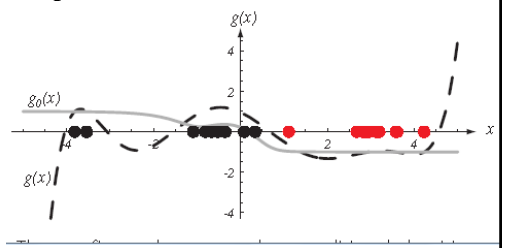
Relation to Optimal Discriminant

- If you set $b = 1_n$, MSE approaches the optimal Bayes discriminant g_0 as number of samples approaches infinity. (see 5.8.3)



$$g_0(x) = P(\omega_2 \mid x) - P(\omega_2 \mid x)$$

g(x) is MSE estimation



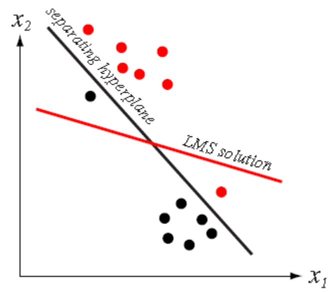
Widrow-Hoff / LMS

- LMS – Least Mean Squared
- Still solves when Y^tY is singular

```
a,b, threshold θ, step η(.), k = 0
begin
   do
        k = (k + 1) mod n
        a = a + η(k)(b_k – a^t y^k) y^k
   until | η(k) )(b_k – a^t y^k) y^k | < θ
   return a
end
```

Widrow-Hoff / LMS

- LMS not guaranteed to converge to a separating plane, even if one exists.



Procedural differences

- Perceptron, relaxation
  - If samples linearly separable, we can find a solution
  - Otherwise, we do not converge to a solution
- MSE
  - Always yields a weight vector
  - May not be the best solution
    - Not guaranteed to be a separating vector

Choosing b

- Arbitrary b, MSE minimizes $||Ya - b||^2$
- If linearly separable, we can more smartly choose b
  - Define â and ß such that

    Yâ = ß > 0
  - Every component of ß is positive

Modified MSE

- $J_s(a,b) = ||Ya - b||^2$
- a, b allowed to vary
- Subject to b > 0
- Min of J_s is zero
- a that achieves min J_s is the separating vector

Ho-Kashyap/Descent prodecure

$$\nabla_a J_s = 2Y^t(Ya - b)$$

$$\nabla_b J_s = -2(Ya - b)$$

- For any b

$$a = Y^{\dagger} b$$

$$\textbf{So, } \nabla_a J_s = 0 \textbf{ and we're done? no...}$$

- – Must avoid b = 0
- – Must avoid b < 0

Ho-Kashyap/Descent Procedure

- Pick positive b
- Don't allow reduction of b's components
- Set all positive components of $\nabla_a J_s$ to zero
  - b(k+1) = b(k) - ηc

$$c = \begin{cases} \nabla_b J_s & \textbf{if } \nabla_b J_s \le 0 \\ \textbf{0} & \textbf{otherwise} \end{cases}$$

$$c = \frac{1}{2}\left(\nabla_b J_s - \left|\nabla_b J_s\right|\right)$$

Ho-Kashyap/Descent Procedure

$$\nabla_b J_s = -2(Ya - b)$$

$$e = Ya - b$$

$$b_{k+1} = b_k - \eta \frac{1}{2}\left[\nabla_b J_s - \left|\nabla_b J_s\right|\right]$$

$$b_{k+1} = b_k + 2\eta_k e_k^+ \qquad e_k^+ = \frac{1}{2}\left(e_k - \left|e_k\right|\right)$$

$$a_k = Y^\dagger b_k$$

Ho-Kashyap

- Algorithm 11
  - Begin initialize a, b, η() < 1, threshold b_{min}, k_{max}
    - do k = k+1 mod n
      - e = Ya – b
      - $e^+ = ½(e+abs(e))$
      - $b = b + 2η(k)e^+$
      - $a = Y^†b$
      - if abs(e) <= b_{min} then return a,b and exit
    - Until k = k_{max}
    - Print “NO SOLUTION”
  - End
- When e(k) = 0 → we have solution
- When e(k) <= 0 → samples not linearly separable

Convergence (separable case)

- If $0 < \eta < 1$, and linearly separable
  - Solution vector exists
  - We will find in finite k steps
- Two possibilities
  - $e(k) = 0$ for some finite k_0
  - No zero in e()
- If $e(k_0)$
  - a(k), b(k), e(k) stop changing
  - $Ya(k) = b(k) > 0$ for all $k > k_0$
  - If we find k_0, algorithm terminates with solution vector

Convergence (separable)

- e() never zero for finite k
- If samples are linearly separable
  - $Ya = b, b > 0$
- Because b is positive, either
  - e(k) is zero, or
  - e(k) is positive
- Since e(k) cannot be zero (first bullet), it must be positive

Convergence (separable)

- $\frac{1}{4}(\|e_k\|^2 - \|e_{k+1}\|^2) = \eta(1-\eta)\|e^+_k\|^2 + \eta^2 e^{+t}_k YY^\dagger e^+_k$
  - $YY^\dagger$ is symmetric, positive semi-definite
  - $0 < \eta < 1$
- Therefore, $\|e_k\|^2 > \|e_{k+1}\|^2$ if $0 < \eta < 1$
  - $\|e\|$ will eventually converge to zero
  - a will eventually converge to solution vector

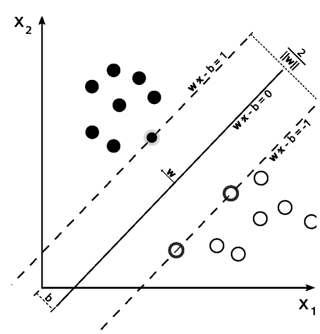
Convergence (non-separable)

- If not linearly separable, may obtain a non-zero error vector without positive components
- Still have
  - $\frac{1}{4}(\|e_k\|^2 - \|e_{k+1}\|^2) = \eta(1-\eta)\|e^+_k\|^2 + \eta^2 e^{+t}_k YY^\dagger e^+_k$
  - So limiting $\|e\|$ cannot be zero
    - Will converge to a non-zero value
- Convergence says that
  - $e^+_k = 0$ for some finite k (separable)
  - *e^+_k will converge to zero while $\|e\|$ is bounded away from zero (non-separable)*

Support Vector Machines (SVMs)

SVMs

- Representing data in higher dimensions space, SVM will construct a separating hyperplane in that space, one which maximizes margin between the two data sets.



Application

- Face detection, verification, and recognition
- Object detection and recognition
- Handwritten character and digit recognition
- Text detection and categorization
- Speech and speaker verification, recognition
- Information and image retrieval

Formalization

- We are given some training data, a set of points of the form

$$\mathcal{D} = \{(\mathbf{x}_i, c_i) | \mathbf{x}_i \in \mathbb{R}^p, c_i \in \{-1, 1\}\}_{i=1}^n$$

Equation of separating hyperplane:

$$\mathbf{w} \cdot \mathbf{x} - b = 0.$$

The vector **w** is a normal vector. The parameter b/||w|| determines the offset of the hyperplane from the origin along the normal vector

Formalization cont…

- Defining two hyperplanes given by equations:

H1: $\mathbf{w} \cdot \mathbf{x} - b = 1$

H2: $\mathbf{w} \cdot \mathbf{x} - b = -1.$

- These hyperplanes are defined in such a way that no points lies between them
- To prevent data points falling between these hyperplanes, following two constraints are defined:

$$\mathbf{w} \cdot \mathbf{x}_i - b \geq 1$$
$$\mathbf{w} \cdot \mathbf{x}_i - b \leq -1$$

Formulation cont…

- This can be rewritten as:

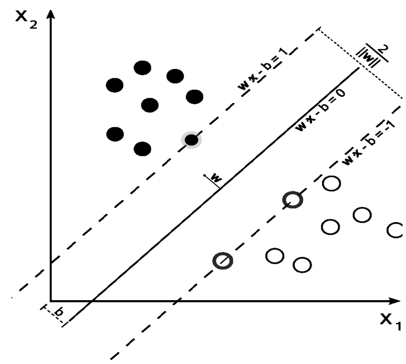
$$c_i(\mathbf{w} \cdot \mathbf{x}_i - b) \geq 1, \quad \text{for all } 1 \leq i \leq n$$

- So the formulation of the optimization problem is
  - Choose **w, b** to minimize ||**w**||

    subject to

$$c_i(\mathbf{w} \cdot \mathbf{x_i} - b) \geq 1, \quad \text{for all } 1 \leq i \leq n$$

SVM Hyperplane Example



SVM Training

- Langrange Optimization problem
- Reformulated Optimization Problem is given as:

$$L_P \equiv \frac{1}{2}||\mathbf{w}||^2 - \sum_{i=1}^{l} \alpha_i y_i (\mathbf{x}_i \cdot \mathbf{w} + b) + \sum_{i=1}^{l} \alpha_i$$

- Thus the new optimization problem is to minimize $\mathbf{L_P}$ w.r.t **w** and **b** subject to:

$$\alpha_i \geq 0$$

SVM Training cont…

- Dual of Langrange formulation

  The dual of Langrange states that the gradient descent of $\mathbf{L_P}$ with respect to **w** and **b** vanishes.

  so we have the dual as:

$$L_D = \sum_i \alpha_i - \frac{1}{2}\sum_{i,j} \alpha_i \alpha_j y_i y_j \mathbf{x}_i \cdot \mathbf{x}_j$$

The optimization problem w.r.t dual is to maximize L_D subject to:

$$\alpha_i \geq 0$$

- From the above optimization equation we have:

$$\mathbf{w} = \sum_i \alpha_i y_i \mathbf{x}_i$$

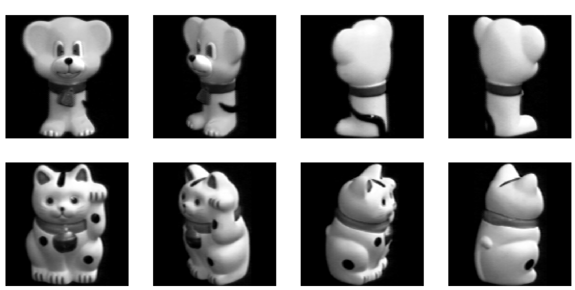
- This shows that the solution is the inner product of input points
- Most of the points have α to be zero and for those points for which α is not zero are the closest points to the separating hyperplane. These points are called support vectors.

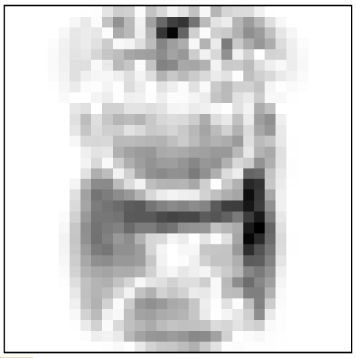
Advantages & Disadvantages of SVM

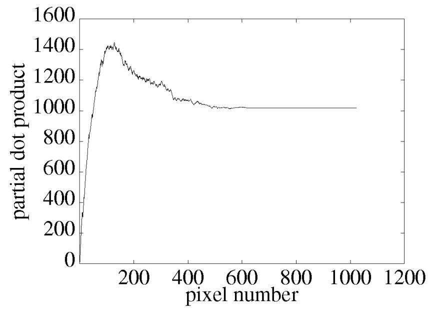
- Advantages
  - Gives high generalization performance
  - Complexity of SVM classifier is characterized by number of support vectors rather than the dimensionality of transformed space.
- Disadvantages
  - The training time scales somewhere between quadratic and cubic with respect to the number of training samples

Recognition of 3D-Objects

- Experiment involved recognition of 3D objects from the COIL db
- Each coil image is transformed into eight-bit vector of 32X32 = 1024 components







References

- Pontil, M.; Verri, A., *"Support vector machines for 3D object recognition*," Pattern Analysis and Machine Intelligence, IEEE Transactions Vol. 20, Issue 6, June 1998, pp. 637 – 646
- Christopher J. C. Burges, *"A tutorial on support vector machines for pattern recognition" (1998)*
- R.O. Duda, P. E. Hart and D. Stork, Wiley , *Pattern Classification* (2nd Edition) by 2001
- R.J. Schalkoff. (1992) *Pattern Recognition: Statistical, Structural, and Neural Approaches*, Wiley.*