

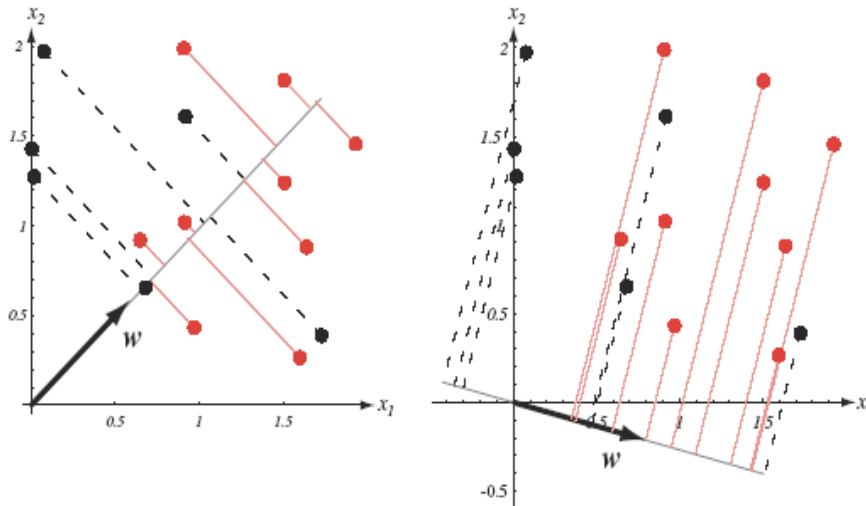
Linear Discriminant Functions and SVM

Seong-Wook Joo

1. Fisher linear discriminant

(1) Motivation

- Want dimensionality reduction in a classification problem.
- PCA seeks directions (principal directions) that best *represent* the original data.
- Fisher linear discriminant analysis (FDA) seeks directions that are efficient for *discrimination*. (below: which \mathbf{w} is better for discrimination?)



(2) Projection onto one direction \mathbf{w} , two-class problem

- Samples: n d -dimensional vectors $\mathbf{x}_1 \dots \mathbf{x}_n$, consisting of two subsets D_1, D_2
- Projected samples: $y = \mathbf{w}^t \mathbf{x}$ two subsets Y_1, Y_2
- Criterion: maximize the Fisher linear discriminant

$$J(\mathbf{w}) = \frac{|\tilde{m}_1 - \tilde{m}_2|^2}{\tilde{s}_1^2 + \tilde{s}_2^2} \quad \tilde{m}_1 = \text{mean of } y \in Y_1, \quad \tilde{s}_1^2 = \sum_{y \in Y_1} (y - \tilde{m}_1)^2 : \text{scatter of } Y_1$$

$$J(\mathbf{w}) = \frac{\mathbf{w}^t \mathbf{S}_B \mathbf{w}}{\mathbf{w}^t \mathbf{S}_W \mathbf{w}}$$

$$\mathbf{S}_B = (\mathbf{m}_1 - \mathbf{m}_2)(\mathbf{m}_1 - \mathbf{m}_2)^t : \text{between scatter matrix } (\mathbf{m}_i = \text{mean of } \mathbf{x} \in D_i)$$

$$\mathbf{S}_W = \mathbf{S}_1 + \mathbf{S}_2 : \text{within scatter matrix, } \mathbf{S}_i = \sum_{\mathbf{x} \in D_i} (\mathbf{x} - \mathbf{m}_i)(\mathbf{x} - \mathbf{m}_i)^t$$

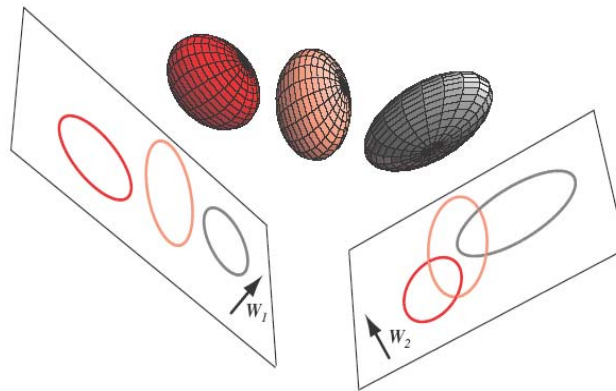
- Solution

$$\mathbf{w} = \mathbf{S}_W^{-1}(\mathbf{m}_1 - \mathbf{m}_2)$$

- Threshold (decision boundary)

Not given by FDA. Use “Bayesian decision theory” (minimize risk or error rate), or any general classifier.

(3) Projection onto multiple directions \mathbf{w}_i , c -class problem: Multiple Discriminant Analysis (MDA)



- Reduction to $(c-1)$ dimensions: There can be at most $c-1$ solutions for \mathbf{w}_i
- Projected samples: $y_i = \mathbf{w}_i^t \mathbf{x}$, $i=1, \dots, c-1$ or in matrix notation

$$\mathbf{y} = \mathbf{W}^t \mathbf{x}$$

- Generalization of scatter matrices

$$\mathbf{S}_W = \sum_{i=1}^c \mathbf{S}_i$$

$$\mathbf{S}_B = \sum_{i=1}^c n_i (\mathbf{m}_i - \mathbf{m})(\mathbf{m}_i - \mathbf{m})^t \quad \text{comes from the total (all classes) scatter matrix}$$

- Criterion: maximize the Fisher linear discriminant

$$J(\mathbf{W}) = \frac{|\mathbf{W}^t \mathbf{S}_B \mathbf{W}|}{|\mathbf{W}^t \mathbf{S}_W \mathbf{W}|}$$

where $||$ denotes matrix determinant (we want a scalar measure of matrix and determinant (product of eigenvals) of the scatter is a measure of scattering volume).

- Solution: solve the generalized eigenvector problem

$$\mathbf{S}_B \mathbf{w}_i = \lambda_i \mathbf{S}_W \mathbf{w}_i$$

Since \mathbf{S}_B is at most rank $c-1$, there can be at most $c-1$ nonzero eigenvalues λ 's and their corresponding \mathbf{w} 's.

2. Linear discriminant functions (linear classifiers [1])

(1) Problem formulation

- Linear discriminant function for a two-class (ω_1 or ω_2) problem

$$g(\mathbf{x}) = \mathbf{w}'\mathbf{x} + w_0$$

- The goal is to find \mathbf{w} : weight vector, w_0 : threshold (bias)
 - so that if $g(\mathbf{x}) > 0$ then \mathbf{x} is ω_1 , if $g(\mathbf{x}) < 0$ then \mathbf{x} is ω_2 .
 - Decision boundary is a hyperplane.
 - Note: The distance from \mathbf{x} to the hyperplane is $g(\mathbf{x})/\|\mathbf{w}\|$. The *margin* is the minimum of the distances.
- For a multi-category case, define a *linear machine*
 $g_i(\mathbf{x}) = \mathbf{w}_i'\mathbf{x} + w_{i0} \quad i = 1, \dots, c$
assign ω_i if $g_i(\mathbf{x}) > g_j(\mathbf{x})$ for all $j \neq i$.

- Generalized linear discriminant function

$$g(\mathbf{x}) = \mathbf{a}'\mathbf{y} + w_0$$

$$\mathbf{y} = \varphi(\mathbf{x})$$

- Allows for nonlinear decision boundary.
 - If \mathbf{y} is of higher dimension we can have multiply connected regions.
 - But beware of “the curse of dimensionality”
- Augmented feature vector

$$\mathbf{y} = \begin{bmatrix} 1 \\ \mathbf{x} \end{bmatrix}, \quad \mathbf{a} = \begin{bmatrix} w_0 \\ \mathbf{w} \end{bmatrix}$$

$$g(\mathbf{x}) = \mathbf{a}'\mathbf{y}$$

Note: The distance from \mathbf{y} to the hyperplane is $g(\mathbf{y})/\|\mathbf{a}\|$

- Normalization
 - Replace all $\mathbf{y} \in \{ \mathbf{y} \text{ is in } \omega_2 \}$ by $-\mathbf{y}$
 - Problem becomes finding \mathbf{a} such that $g(\mathbf{x}) = \mathbf{a}'\mathbf{y} > 0$ for all \mathbf{y}

(2) Criterion functions and algorithms

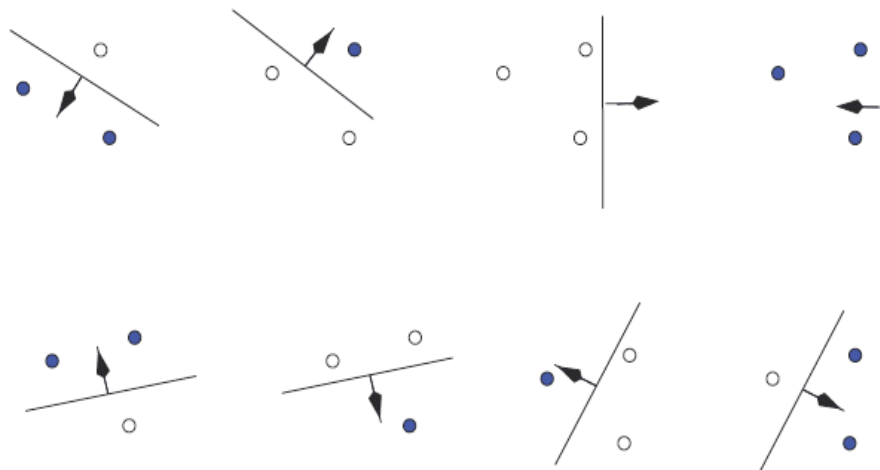
- Linearly separable case: Perceptron criterion
 - $J_P = \sum_{\mathbf{y} \in Y} (-\mathbf{a}'\mathbf{y})$, $Y = \{ \mathbf{y}'\text{'s misclassified by } \mathbf{a} \}$
 - Algorithm: keep adding misclassified \mathbf{y} to \mathbf{a}
 - Finite convergence only if linearly separable, else does not terminate

- Non-separable case: Minimum squared error
 - $J_S = \|\mathbf{Y}\mathbf{a} - \mathbf{b}\|^2 = \sum_{i=1}^n (\mathbf{a}^t \mathbf{y}_i - b_i)^2$ for some given \mathbf{b} (e.g., all ones)
 - Solution: $\mathbf{a} = (\mathbf{Y}^t \mathbf{Y})^{-1} \mathbf{Y}^t \mathbf{b}$
 - Resulting \mathbf{a} is same as FDA if $\mathbf{b} = [n/n_1 \dots (n_1 \text{ times}) \ n/n_2 \dots (n_2 \text{ times})]^t$
 - In addition this gives the threshold $w_0 = -\mathbf{m}^t \mathbf{w}$
- Ho-Kashyap procedure
 - Combination of MSE (both \mathbf{a} and \mathbf{b} are unknown) and Perceptron
 - Finite convergence if linearly separable, else provides proof of non-separability (but with no bound on the number of iterations needed)
- Linear programming
 - Finite convergence in both separable and non-separable (but solution useful only if separable?)

3. Support Vector Machines (SVM) [2]

(1) Capacity of a classifier

- Need to be *accurate* on the training samples but also *generalize* well on testing samples
- A classifier with too much *capacity* will not generalize well e.g., high degree polynomial
- “VC dimension” is a measure of the capacity of a class of function
 - definition of VC dimension: largest number of pts that can be shattered (classify all possible labeling combinations) by the function.
 - A line on \mathbf{R}^2 has VC dim = 3, hyperplane in \mathbf{R}^n has VC dim = $n+1$



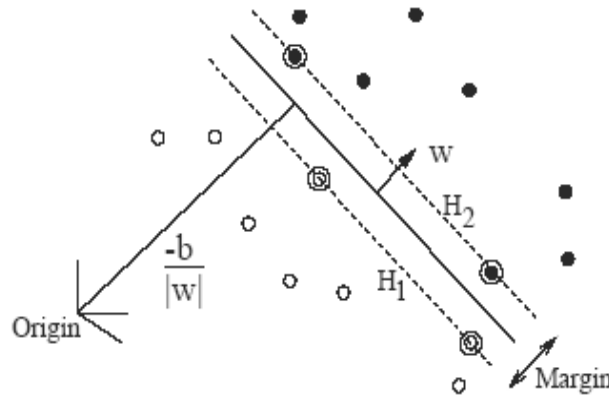
- VC bound
 - The expected risk (error) of a classifier is bounded by the following

$$R \leq R_{emp} + \phi(h, \dots)$$

$$R_{emp}: \text{training error, } h: \text{VC dimension}$$
 - $\phi(h, \dots)$ monotonically increases with h

(2) Linear SVM

- Training data: $\{(\mathbf{x}_i, y_i) \mid i=1, \dots, l\}$, \mathbf{x}_i : training samples, y_i : label (+1 or -1)
- SVM looks for the separating hyperplane (\mathbf{w}, b) that maximizes the margin



- Say (\mathbf{w}, b) is scaled so that

$$\begin{cases} \mathbf{x}_i \cdot \mathbf{w} + b \geq +1 & \text{for } y_i = +1 \\ \mathbf{x}_i \cdot \mathbf{w} + b \leq -1 & \text{for } y_i = -1 \end{cases}$$

Then margin = $2/\|\mathbf{w}\|$. Therefore minimize $\|\mathbf{w}\|^2$ subject to above constraint

- Solve for Lagrange multipliers $\alpha_i \geq 0$ from the “dual” problem (details omitted)
- $\mathbf{w} = \sum_i \alpha_i y_i \mathbf{x}_i$
- The *support vectors* are the \mathbf{x}_i 's satisfying the equality.
- The capacity decreases as margin increases [2]
- Slack variables ($\xi_i > 0$) can be added to allow for outliers

$$\begin{cases} \mathbf{x}_i \cdot \mathbf{w} + b \geq +1 - \xi_i & \text{for } y_i = +1 \\ \mathbf{x}_i \cdot \mathbf{w} + b \leq -1 + \xi_i & \text{for } y_i = -1 \end{cases}$$

with some penalty: minimize $\|\mathbf{w}\|^2 + C(\sum_i \xi_i)$

(3) Non-linear SVM

- Use non-linear mapping Φ that maps \mathbf{x} into higher (possibly infinite) dimensional space
- Note \mathbf{x}_i appears only in the form of *dot products* in the training phase
- If there exists a “kernel function” K such that $K(\mathbf{x}_i, \mathbf{x}_j) = \Phi(\mathbf{x}_i) \cdot \Phi(\mathbf{x}_j)$, then we don’t need to know what $\Phi(\mathbf{x})$ is.
- We don’t need to know what \mathbf{w} is in the testing phase either since
$$\mathbf{x} \cdot \mathbf{w} + b = \sum_i \alpha_i y_i (\mathbf{x} \cdot \mathbf{x}_i) + b$$
- Examples of such kernels

$$K(\mathbf{x}, \mathbf{y}) = (\mathbf{x} \cdot \mathbf{y} + 1)^p \quad (\text{polynomial})$$

$$K(\mathbf{x}, \mathbf{y}) = e^{-\|\mathbf{x}-\mathbf{y}\|^2 / 2\sigma^2} \quad (\text{Gaussian})$$

$$K(\mathbf{x}, \mathbf{y}) = \tanh(\kappa \mathbf{x} \cdot \mathbf{y} - \delta) \quad (\text{sigmoid})$$

(4) Notes

- SVM training always finds a global solution (neural network doesn’t)
- SVM depends on the choice of kernel. Best choice not known
- Speed and size (large datasets) are problems yet to be solved

References

- [1] Duda, Hart, and Stork, "Pattern Classification" Chapter 5, Wiley, 2000
- [2] C. J. C. Burges. “A Tutorial on Support Vector Machines for Pattern” Recognition. Knowledge Discovery and Data Mining, 2(2), 1998
- [3] Bernhard Scholkopf and Alexander J. Smola, "Learning with Kernels - Support Vector Machines, Regularization, Optimization, and Beyond", MIT Press, 2002, Chapter 1