Tema 4.2:
FUNCIONES DISCRIMINANTES LINEALES y SV

Some Figures in these slides were taken from
Pattern Classification (2nd ed) by R. O. Duda, P. E. Hart and D. G. Stork, John Wiley & Sons, 2000
with the permission of the authors

Febrero-Mayo 2007

INDICE

1 INTRODUCCIÓN

Se asume:

- No se conocen las f.d.p.
- Se deciden las funciones discriminates salvo parámetros:
  - **Funciones lineales respecto al vector de características** *(como lqd: MAP con distribuciones gaussianas y matrices de covarianza iguales)*
  - No lineales: Funciones lineales respecto a conjunto de funciones que a su vez dependen del vector de características de forma no lineal
  - La función discriminante se diseña como resultado de minimizar una función criterio.

- **Propiedades de interés**:
  - Eficiencia computacional
  - Convergencia rápida con procedimientos de gradiente descendente, (al utilizar la base de datos de entrenamiento para estimar los parámetros de dichas funciones).

2 FUNCIONES DISCRIMINANTES LINEALES Y SUPERFICIES DE DECISIÓN

Definición de función lineal. $$g(\mathbf{x}) = \mathbf{w}^T \mathbf{x} + w_0$$

- Vector de características $\mathbf{x}$
- Vector de pesos $\mathbf{w}$
- Offset w_0

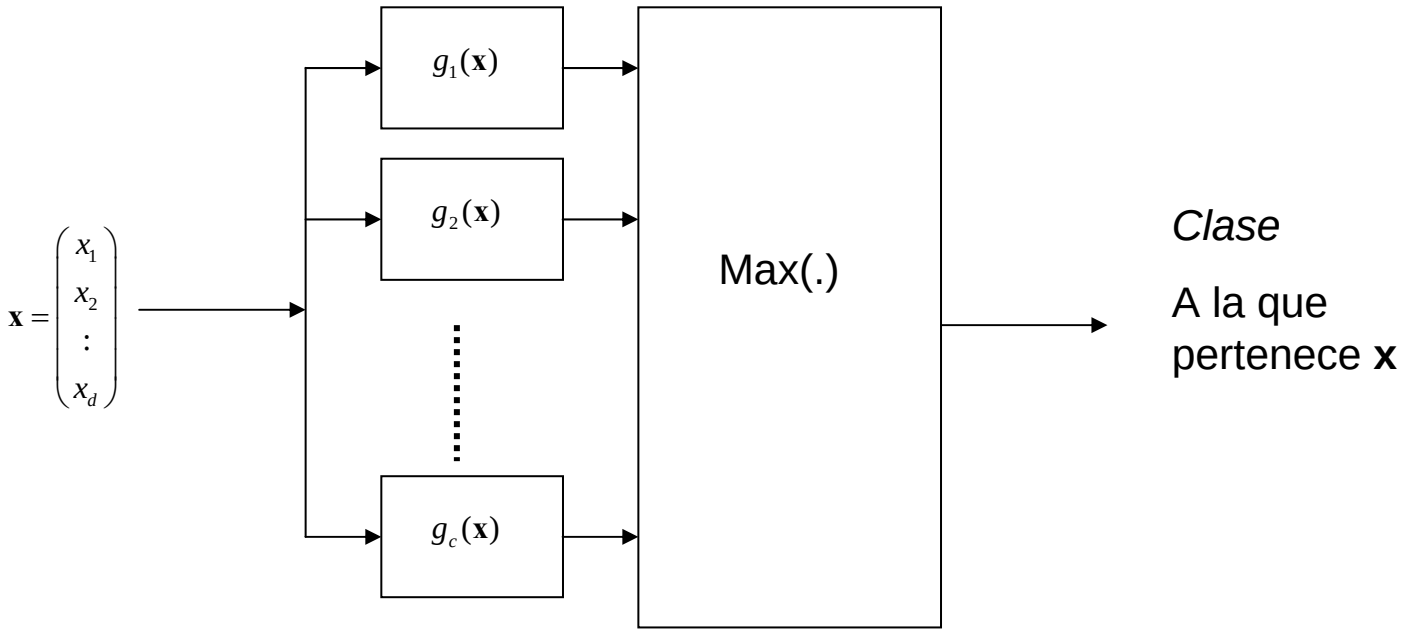
- Parámetros a ajustar $\begin{pmatrix} \mathbf{w} \\ w_0 \end{pmatrix}$

- C clases: C funciones discriminantes

$$g_i(\mathbf{x}) = \mathbf{w}_i^T \mathbf{x} + w_{0-i}$$
$$i = 1..c$$

2 FUNCIONES DISCRIMINANTES LINEALES Y SUPERFICIES DE DECISIÓN

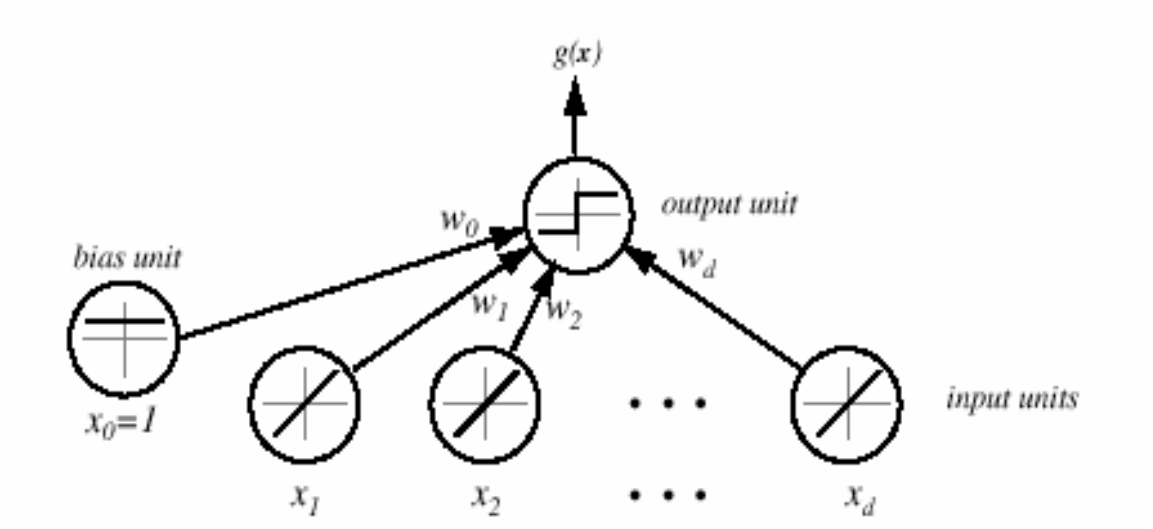
- Clasificador de c categorías:



2 FUNCIONES DISCRIMINANTES LINEALES Y SUPERFICIES DE DECISIÓN

- Clasificador de 2 categorías: DICOTOMIZADOR

$$g(\mathbf{x}) \begin{array}{c} > \omega_1 \\ < \omega_2 \end{array} 0$$



2 FUNCIONES DISCRIMINANTES LINEALES Y SUPERFICIES DE DECISIÓN

INTERPRETACIONES GEOMÉTRICAS:

- Las superficies de decisión son hiperplanos.
- El vector **w** es ortogonal al hiperplano que separa las zonas de decisión, y por tanto a través de él se determina la orientación de la superficie.
- El bias w_0 fija la superficie en un punto determinado.
- La función $g(\mathbf{x})$ da una medida de distancia del vector **x** al hiperplano:

- Hiperplano H_s:

$$\mathbf{x} \in \mathrm{H}_s \Rightarrow g(\mathbf{x}) = 0 \qquad \mathbf{x} = \mathbf{x}_p + r\frac{\mathbf{w}}{\|\mathbf{w}\|}$$

- Proyección de x sobre H_s: $\mathbf{x}_p$

$$g(\mathbf{x}) = \mathbf{w}^T\mathbf{x} + w_0 = \mathbf{w}^T\left(\mathbf{x}_p + r\frac{\mathbf{w}}{\|\mathbf{w}\|}\right) + w_0 =$$

- Distancia de **x** a H_s:

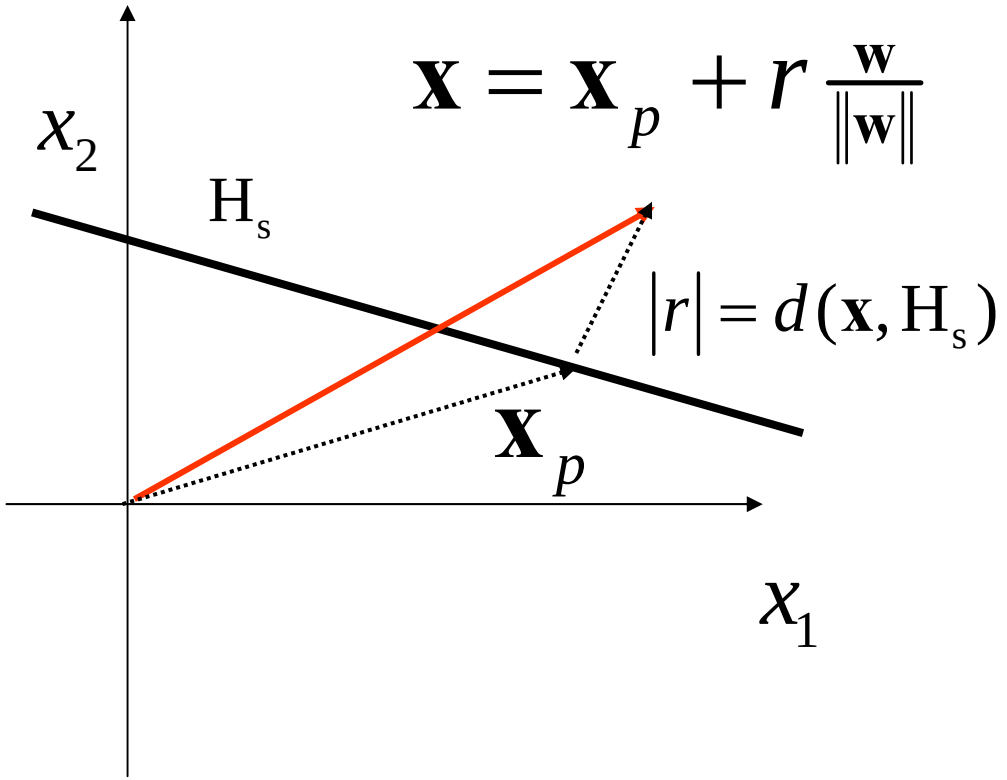
$$\mathbf{w}^T\mathbf{x}_p + w_0 + \mathbf{w}^T r\frac{\mathbf{w}}{\|\mathbf{w}\|} = g(\mathbf{x}_p) + r\frac{\|\mathbf{w}\|^2}{\|\mathbf{w}\|} = 0 + r\|\mathbf{w}\| =$$

$$\pm d(\mathbf{x}, \mathrm{H}_s)\|\mathbf{w}\|$$

$$r = \pm d(\mathbf{x}, \mathrm{H}_s)$$

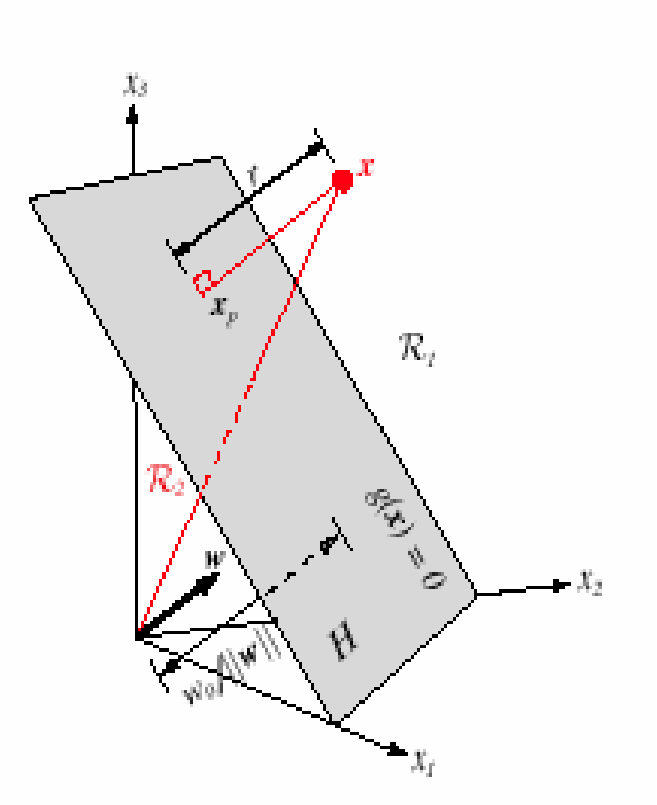
2 FUNCIONES DISCRIMINANTES LINEALES Y SUPERFICIES DE DECISIÓN

$$\mathbf{x} = \mathbf{x}_p + r\frac{\mathbf{w}}{\|\mathbf{w}\|}$$



2 FUNCIONES DISCRIMINANTES LINEALES Y SUPERFICIES DE DECISIÓN

- Medida de distancia del vector **x** al hiperplano:



$$x = x_p + r\frac{\mathbf{w}}{\|\mathbf{w}\|}$$

$$g(\mathbf{x}) = d(\mathbf{x}, \mathrm{H}_s)\|\mathbf{w}\|$$

2 FUNCIONES DISCRIMINANTES LINEALES Y SUPERFICIES DE DECISIÓN

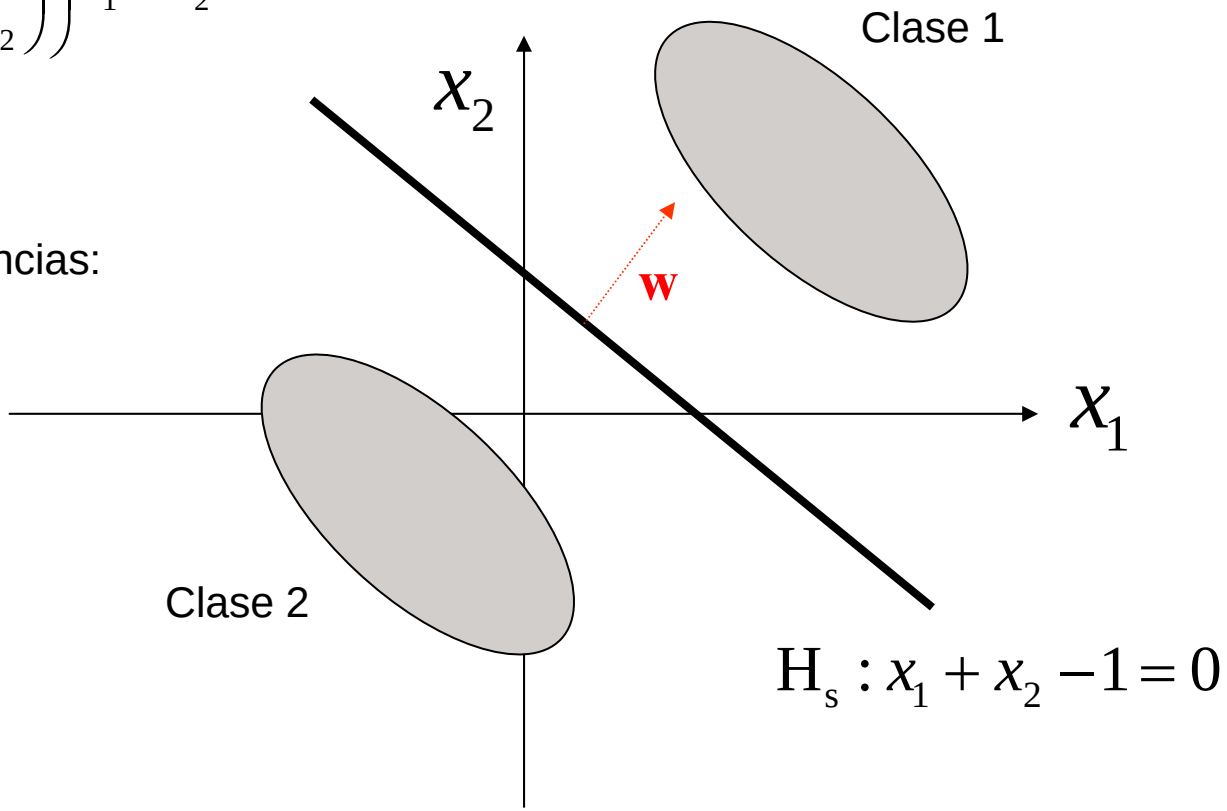
$$\left.\begin{array}{l}\mathbf{w}=\begin{pmatrix}1\\1\end{pmatrix}\\ w_0=-1\end{array}\right\} g\left(\begin{pmatrix}x_1\\x_2\end{pmatrix}\right) x_1+x_2-1=0$$

Cálculo de las siguientes distancias:

$$d((-2,0),\mathrm{H_s})$$

$$d((0,0),\mathrm{H_s})$$

$$d((-1,-1),\mathrm{H_s})$$



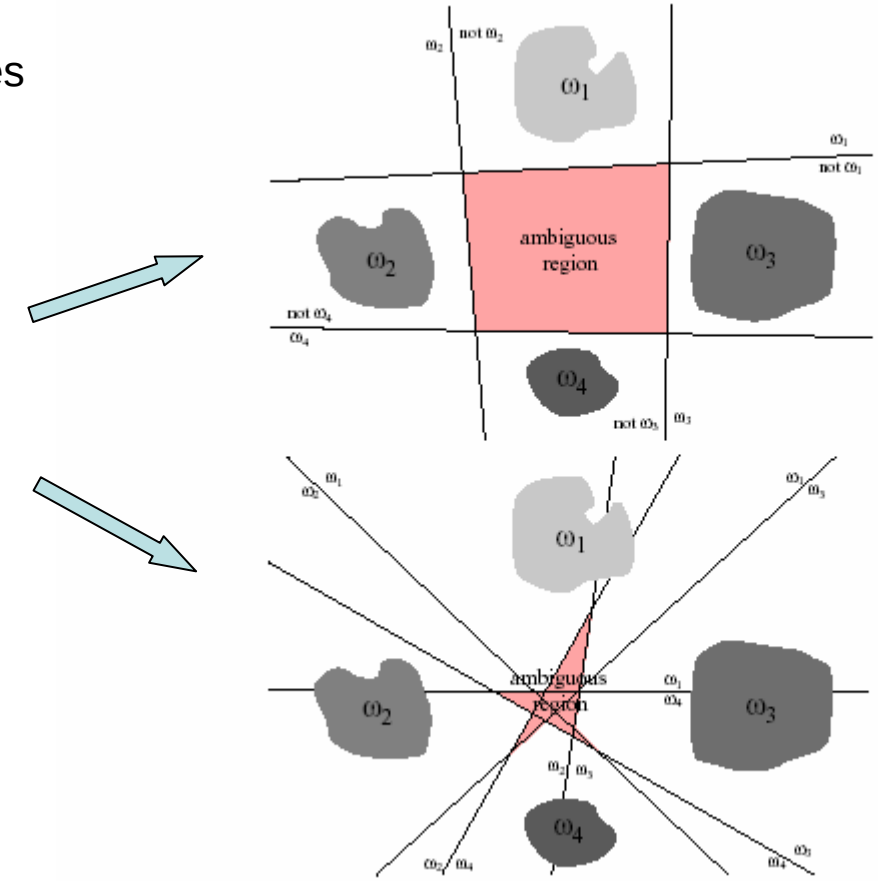
2 FUNCIONES DL y SD: MULTICATEGORY CASE

2 Formas de extender el problema a c>2 clases

Multicategory Case:

I) c Two class problem

II) c(c-1)/2 two class problems



2 FUNCIONES DL y SD: MULTICATEGORY CASE

Definitivamente el número de fronteras es <= c(c-1)/2 hiperplanos

$$\hat{\omega}_i = \max_i \left(g_i(\mathbf{x}) \right)$$

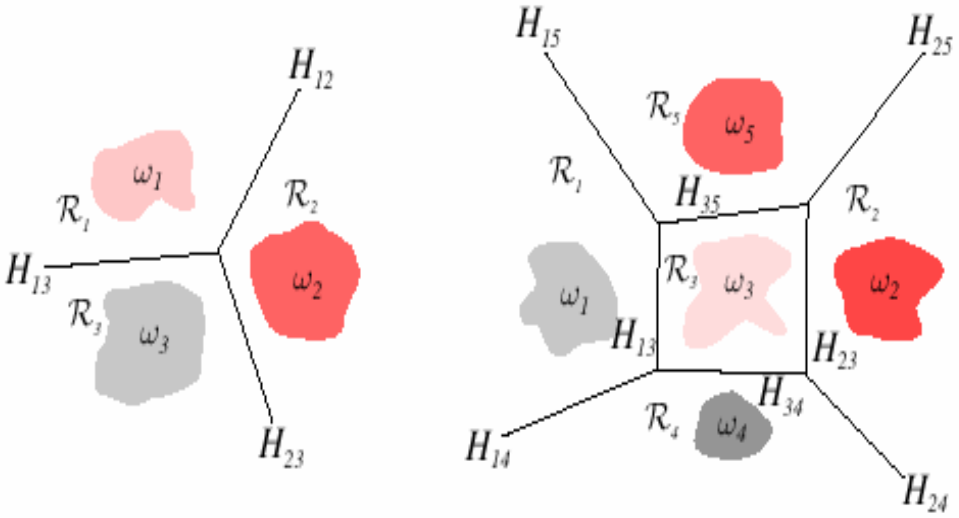


FIGURE 5.4. Decision boundaries produced by a linear machine for a three-class problem and a five-class problem. From: Richard O. Duda, Peter E. Hart, and David G. Stork, *Pattern Classification*. Copyright © 2001 by John Wiley & Sons, Inc.

3 CASO SEPARABLE DE 2 CATEGORÍAS

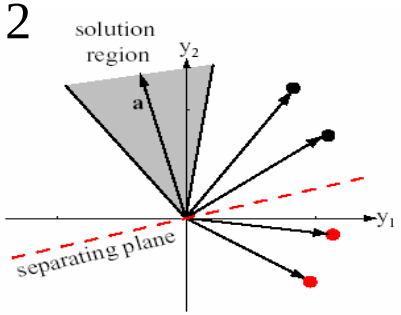
- Caso de dos categorías: ω_1, ω_2
- Augmented Feature Vector: Vector de características aumentado a dimensión+1

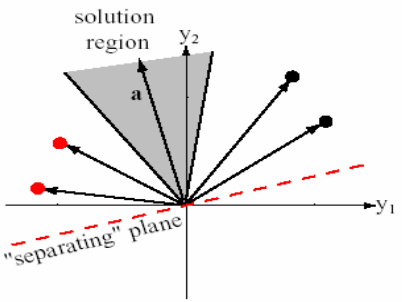
$$\hat{d} = d + 1 \quad \mathbf{y} = \begin{pmatrix} \mathbf{x} \\ 1 \end{pmatrix} \in \mathbb{R}^{\hat{d}} \quad \mathbf{a} = \begin{pmatrix} \mathbf{W} \\ w_0 \end{pmatrix}$$

- Función lineal $g(\mathbf{y}) = \mathbf{a}^T \mathbf{y}$
- Regla de Decisión: $\mathbf{a}^T \mathbf{y} \gtrless_{\omega_2}^{\omega_1} 0$
- Estrategia para simplificar el algoritmo de búsqueda de la solución.
  - Ejemplo : $d = 1, \hat{d} = 2$

$$\omega_1 \Rightarrow \mathbf{a}^T \mathbf{y}_k > 0$$

$$\omega_2 \Rightarrow \mathbf{a}^T \mathbf{y}_k < 0 \Rightarrow \mathbf{y}_k = -\mathbf{y}_k \Rightarrow \mathbf{a}^T \mathbf{y}_k > 0$$



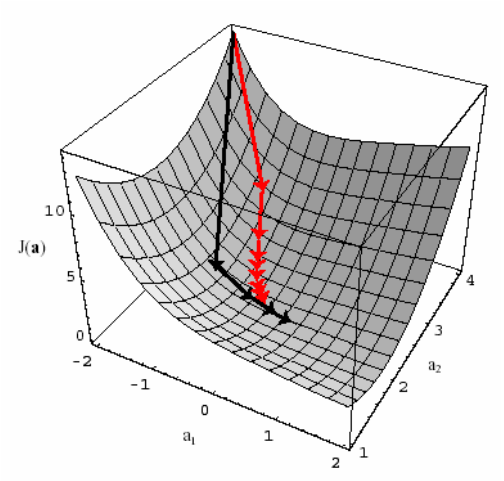


4 ALGORITMO DE GRADIENTE DESCENDENTE

- Estrategia de búsqueda del Vector solución o Vector separador:
- Minimizar una función que depende del vector **a** mediante un algoritmo de gradiente descendente:
- CORRECCIÓN PROPORCIONAL AL ERROR

$$\min_{\mathbf{a}}\left(J\left(\mathbf{a},\mathbf{y}_1..\mathbf{y}_N\right)\right)=\min_{\mathbf{a}}\left(J\left(\mathbf{a}\right)\right)$$

$$\mathbf{a}\left[k+1\right]=\mathbf{a}\left[k\right]-\eta\left[k\right]\nabla\left(J\left(\mathbf{a}\right)\right)$$



4.1 ALG. de GRAD. : FUNCIÓN PERCEPTRON

- Funciones a Minimizar: FUNCIÓN PERCEPTRON.
  - Adecuada para casos Separables
  - E: Conjunto de vectores mal clasificados por **a**).
  - Batch Algorithm:

$$J_p(\mathbf{a}) = \sum_{\mathbf{y}\in \mathrm{E}} \left(-\mathbf{a}^T \mathbf{y}\right) \Rightarrow \nabla J_p = \sum_{\mathbf{y}\in \mathrm{E}} (-\mathbf{y})$$

$$\mathbf{a}[k+1] = \mathbf{a}[k] + \eta[k] \sum_{\mathbf{y}\in \mathrm{E}} (\mathbf{y})$$

- SIMPLIFICACIÓN: Fixed increment (η constant) single sample perceptron

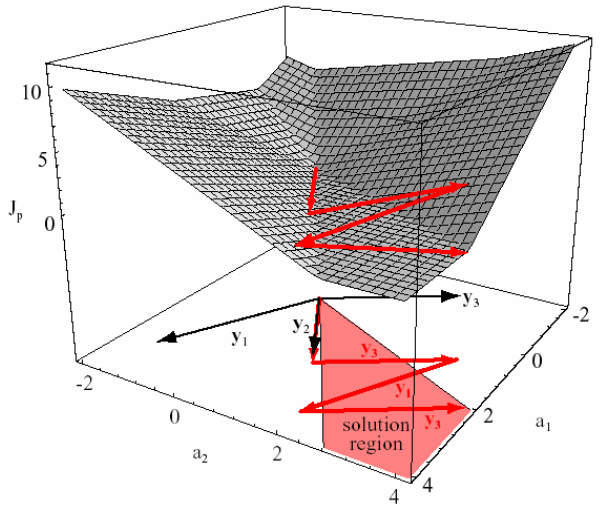
$$\mathbf{a}[k+1] = \begin{cases} \mathbf{a}[k] + \mathbf{y}[k] & \mathbf{y}[k] \in \mathrm{E} \\ \mathbf{a}[k] & \mathbf{y}[k] \notin \mathrm{E} \end{cases}$$

4.1 ALG. de GRAD. : FUNCIÓN PERCEPTRON

SIMPLIFICACIÓN: Fixed increment (η constant) single sample perceptron

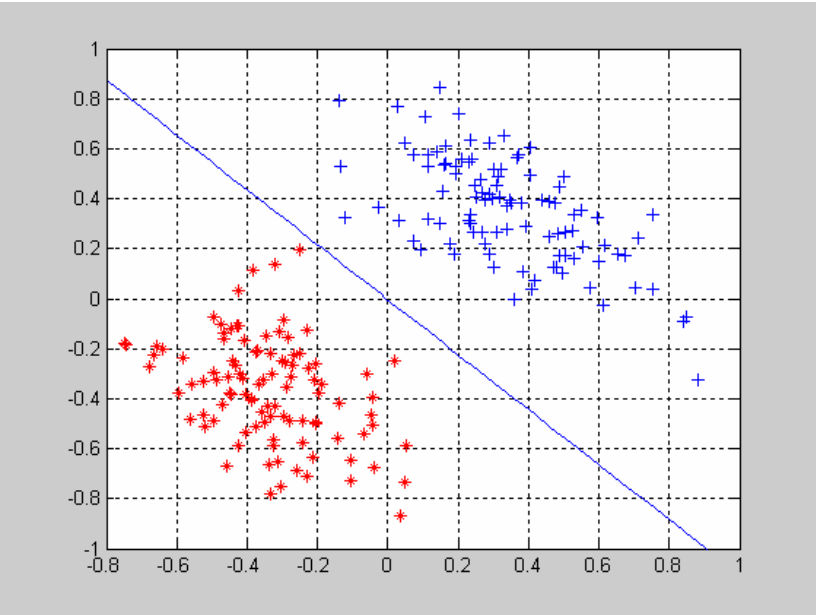
Algorithm 4 (Fixed-increment single-sample Perceptron)

```
1 begin initialize  a, k = 0
2         do  k ← (k + 1) mod n
3             if  y_k is misclassified by a  then  a ← a + y_k
4         until all patterns properly classified
5    return a
6 end
```

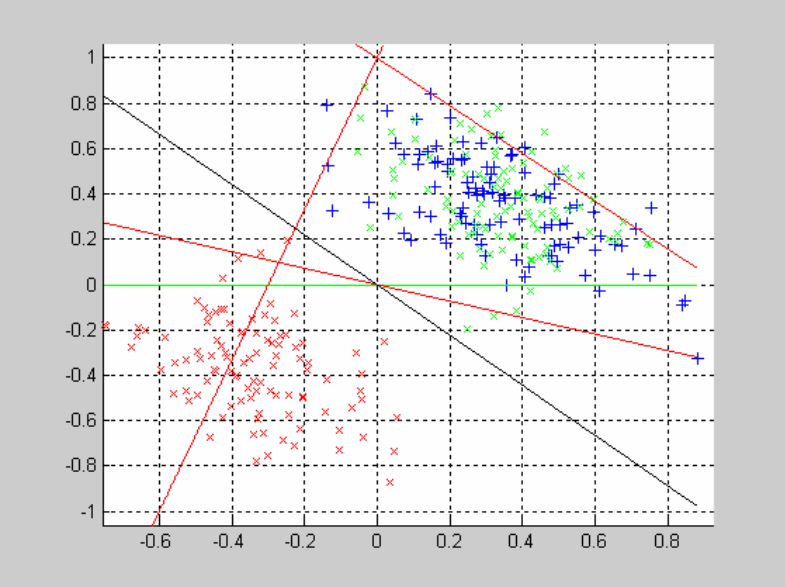


4.1 ALG. de GRAD. : FUNCIÓN PERCEPTRON

Fixed increment (η =1) single sample perceptron Ejemplo BPSK SNR=5dB



ldc

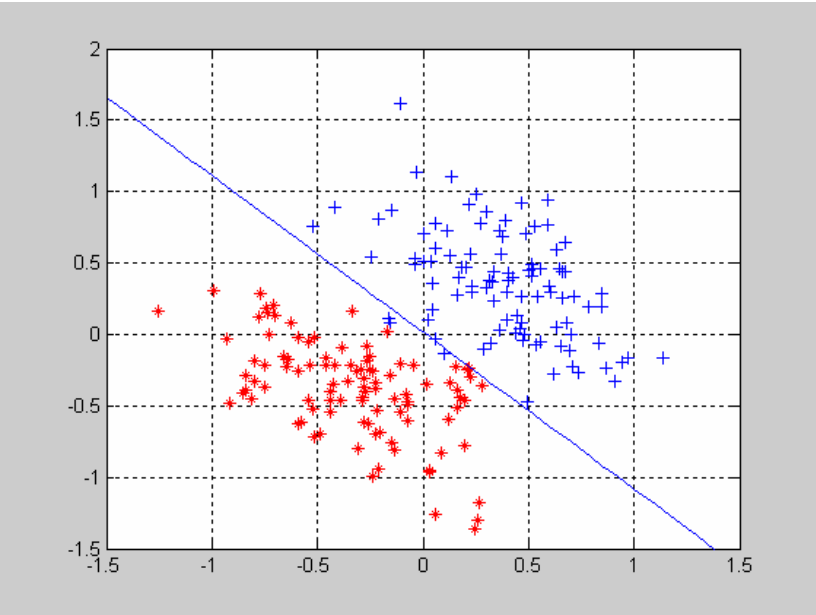


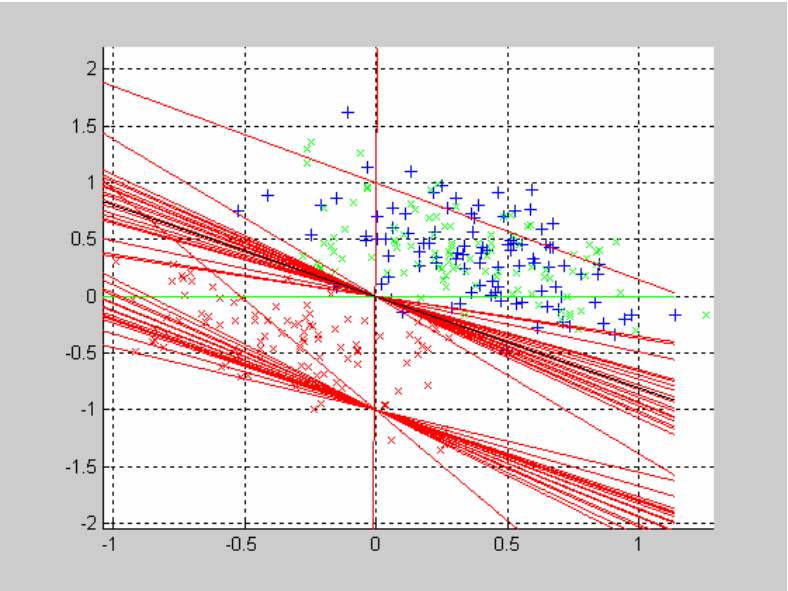
Fixed Increment Single Sample

Batch Perceptron (4 adapt.)

4.1 ALG. de GRAD. : FUNCIÓN PERCEPTRON

Fixed increment (η =1) single sample perceptron Ejemplo BPSK SNR=0dB





ldc

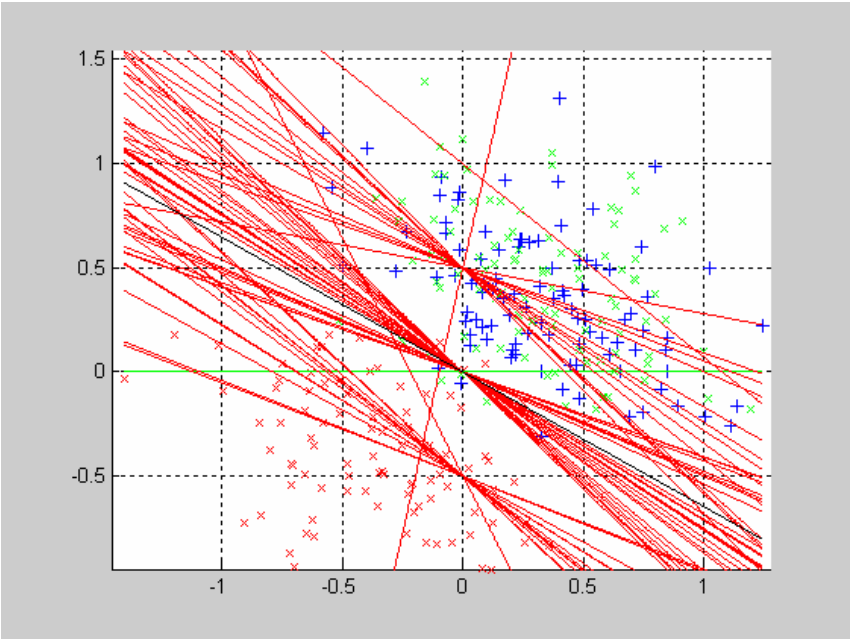
$$\eta(k) = \frac{\eta_0}{k}$$

Fixed Increment Single Sample

Batch Perceptron (38 adapt.)

4.1 ALG. de GRAD. : FUNCIÓN PERCEPTRON

Fixed increment (η =0.5) single sample perceptron Ejemplo BPSK SNR=0dB
(56 Iteraciones)



4.1 ALG. de GRAD. : FUNCIÓN PERCEPTRON

- FUNCIÓN PERCEPTRON: FIXED INCREMENT PARA C CLASES.

(c(c-1)/2 funciones)

$$\mathbf{y}(t)=\begin{pmatrix}1\\ \mathbf{x}(t)\end{pmatrix}$$

$$\mathbf{a}_i; i=1..c \Rightarrow \mathbf{a}_i^T\mathbf{y}(t)-\mathbf{a}_j^T\mathbf{y}(t)>0 \quad \text{with} \quad \mathbf{y}(t)\in D_i$$

- Fixed increment rule (Casos Separables)

$$\text{if } \mathbf{y}(t)\in D_i \quad \text{and} \quad \mathbf{a}_i^T\mathbf{y}(t)<\mathbf{a}_j^T\mathbf{y}(t)\Rightarrow$$

$$\mathbf{a}_i[t+1]=\mathbf{a}_i[t]+\mathbf{y}(t)$$

$$\mathbf{a}_j[t+1]=\mathbf{a}_j[t]-\mathbf{y}(t)$$

4.2 ALG. de GRA: MÍNIMO ERROR CUADRÁTICO

(A aplicar en casos no separables)

- Función Objetivo: $$J_s(\mathbf{a}) = \sum_{i=1}^{N} \left(\mathbf{a}^T \mathbf{y}[i] - b_i\right)^2 = \|\mathbf{Ya} - \mathbf{b}\|^2 = (\mathbf{Ya} - \mathbf{b})^T (\mathbf{Ya} - \mathbf{b})$$

- Matriz de vectores

$$\mathbf{Y} = \begin{pmatrix} 1_1 & \mathbf{X}_1 \\ -1_2 & -\mathbf{X}_2 \end{pmatrix} = \begin{pmatrix} 1 & x_1[1] & : & x_d[1] \\ : & : & : & : \\ 1 & x_1[N_1] & : & x_d[N_1] \\ -1 & -x_1[1] & : & -x_d[1] \\ : & : & : & : \\ -1 & -x_1[N_2] & : & -x_d[N_2] \end{pmatrix} = \begin{pmatrix} \mathbf{y}^T[1] \\ : \\ \mathbf{y}^T[N_1] \\ \mathbf{y}^T[N_1+1] \\ : \\ \mathbf{y}^T[N_1+N_2] \end{pmatrix}$$

- Vector de margen **b**

$$\mathbf{Ya} - \mathbf{b} \simeq \mathbf{0}$$

- Gradiente:

$$J_s(\mathbf{a}) = \mathbf{a}^T \mathbf{Y}^T \mathbf{Ya} - \mathbf{b}^T \mathbf{Ya} - \mathbf{a}^T \mathbf{Y}^T \mathbf{b} + \mathbf{bb}^T$$

$$\nabla J_s(\mathbf{a}) = 2\left(\mathbf{Y}^T \mathbf{Ya} - \mathbf{Y}^T \mathbf{b}\right) = 0 \Rightarrow \mathbf{a} = \left(\mathbf{Y}^T \mathbf{Y}\right)^{-1} \mathbf{Y}^T \mathbf{b} = \mathbf{Y}^{\#} \mathbf{b}$$

4.2 ALG. de GRA: MÍNIMO ERROR CUADRÁTICO

(A aplicar en casos no separables)

- Solución depende del vector **b**: Eligiendo el siguiente vector para definir el margen

$$\mathbf{Y}=\begin{pmatrix}1_1 & \mathbf{X}_1\\ -1_2 & -\mathbf{X}_2\end{pmatrix};\quad \mathbf{a}=\begin{pmatrix}w_0\\ \mathbf{w}\end{pmatrix};\quad \mathbf{b}=\begin{pmatrix}\frac{N}{N_1}1_{N_1}\\ \frac{N}{N_2}1_{N_2}\end{pmatrix}$$

- El resultado es equivalente a la Función Discriminante Lineal de Fisher (MDA)

$$w_0=-\tfrac{1}{2}\left(\mathbf{m}_1+\mathbf{m}_2\right)\mathbf{w} \qquad \mathbf{w}=\alpha\mathbf{S}_C^{-1}\left(\mathbf{m}_1-\mathbf{m}_2\right)=\alpha'\mathbf{S}_C^{-1}\mathbf{S}_B \qquad \mathbf{m}_i=\tfrac{1}{N_i}\sum_i\mathbf{x}_i$$

$$\mathbf{S}_T=\sum_{\mathbf{x}\in\{D_1,\ldots,D_c\}}\left(\mathbf{x}-\mathbf{m}\right)\left(\mathbf{x}-\mathbf{m}\right)^T=$$

$$\sum_{i=1}^{c}\mathbf{S}_{C,i}+\sum_{i=1}^{c}N_i\left(\mathbf{m}_i-\mathbf{m}\right)\left(\mathbf{m}_i-\mathbf{m}\right)^T=\mathbf{S}_C+\mathbf{S}_B$$

Suma de matrices de dispersión intra-clases

Matriz de distancia inter-clases

4.2 ALG. de GRA: MÍNIMO ERROR CUADRÁTICO

(A aplicar en casos no separables)

- Solución depende del vector **b**: Eligiendo el siguiente vector para definir el margen

$$\mathbf{Y} = \begin{pmatrix} 1_1 & \mathbf{X}_1 \\ -1_2 & -\mathbf{X}_2 \end{pmatrix}; \quad \mathbf{a} = \begin{pmatrix} w_0 \\ \mathbf{w} \end{pmatrix}; \quad \mathbf{b} = 1_N$$

- El resultado es equivalente a la solución MAP (lqd)

4.2 ALG. de GRA: MÍNIMO ERROR CUADRÁTICO

(A aplicar en casos no separables)

Aplicando el Algoritmo LMS:

- Función a minimizar $$J_s(\mathbf{a}) = \sum_{i=1}^{N} \left(\mathbf{a}^T \mathbf{y}[i] - b_i\right)^2 = \sum_{i=1}^{N} \left(e[i]\right)^2$$

- Estimación del gradiente $$\nabla J_{s,\mathbf{w}}[k] = 2\left(\mathbf{a}^T \mathbf{y}[k] - b_k\right)\mathbf{y}[k] = 2e[k]\mathbf{y}[k]$$

- Ecuación de adaptación

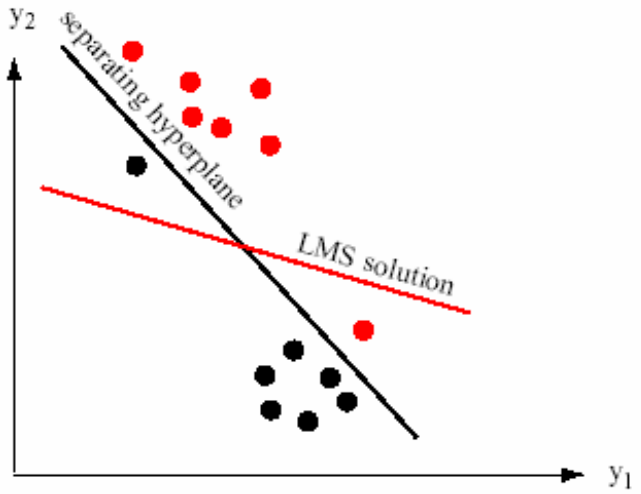
$$\mathbf{a}[k+1] = \mathbf{a}[k] - \eta[k]\nabla J_{s,\mathbf{w}}[k] =$$
$$\mathbf{a}[k] + \eta[k]\left(b_k - \mathbf{a}^T[k]\mathbf{y}[k]\right)\mathbf{y}[k]$$

4.2 ALG. de GRA: MÍNIMO ERROR CUADRÁTICO

(A aplicar en casos no separables)

Algorithm 10 (LMS)

1 **begin initialize** $\mathbf{a}, \mathbf{b}$, criterion $\theta, \eta(\cdot), k = 0$
2 **do** $k \leftarrow k + 1$
3 $\mathbf{a} \leftarrow \mathbf{a} + \eta(k)(b_k - \mathbf{a}^t\mathbf{y}^k)\mathbf{y}^k$
4 **until** $\eta(k)(b_k - \mathbf{a}^t\mathbf{y}^k)\mathbf{y}^k < \theta$
5 **return** a
6 **end**



5 SV: Support Vector Classifier

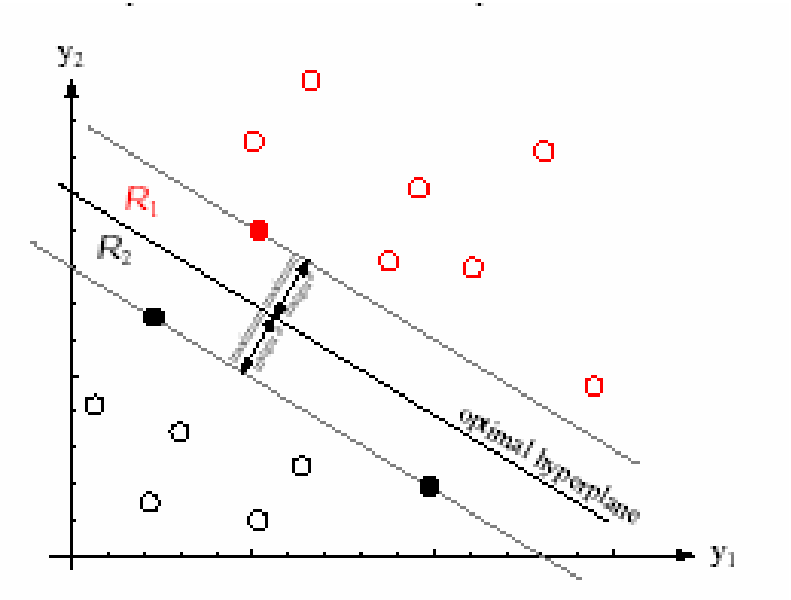
Con datos lineales, el SVC es muy similar al algoritmo que minimiza la función de mínimo error cuadrático medio. :

- Se elige la solución que maximiza el margen de separación entre la frontera y los datos.
- El margen se define como el ancho del “tubo” que no contiene muestras y se halla alrededor de la frontera de decisión.
- El nombre del algoritmo (Support Vector) se debe a que en la solución final hay muy pocas muestras multiplicadas por un factor (multiplicador de Lagrange para el caso no lineal) diferente de cero. Estas muestras corresponden a las más próximas a la frontera.

Referencias SV:

Pattern Recognition and Machine Learning, *Christopher M. BishopSpringer (2006). TEMAS 6y7*

F. Pérez-Cruz and O. Bousquet, (2004). “Kernel methods and their potential use in signal processing”. IEEE SignalProcessing Magazine, 21(3):57 - 65, May.



5.1 SVC: Caso lineal

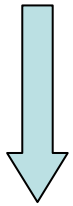
Función a diseñar: $g(\mathbf{x}) = \mathbf{w}^T \mathbf{x} + w_0$

Condiciones a cumplir

$$\begin{cases} \mathbf{w}^T \mathbf{x}_k + w_0 \geq 1; \quad \text{if} \quad \mathbf{x}_k \in D_1; \quad c_k = 1 \\ \mathbf{w}^T \mathbf{x}_k + w_0 \leq -1; \quad \text{if} \quad \mathbf{x}_k \in D_2; \quad c_k = -1 \end{cases}$$

Restricciones:

$$c_k \left(\mathbf{w}^T \mathbf{x}_k + w_0 \right) \geq 1$$



Se utilizan multiplicadores de Lagrange:

$$L = \tfrac{1}{2} \|\mathbf{w}\|^2 - \sum_{k=1}^{N} \alpha_k \left(c_k \left(\mathbf{w}^T \mathbf{x}_k + w_0 \right) - 1 \right); \qquad \alpha_k \geq 0$$

5.1 SVC: Caso lineal

En forma Matricial (1):

$$L = \tfrac{1}{2}\mathbf{w}^{\mathbf{T}}\mathbf{w} - \boldsymbol{\alpha}_c^{\mathbf{T}}\boldsymbol{\Phi}^{\mathbf{T}}\mathbf{w} + w_0\boldsymbol{\alpha}_c^{\mathbf{T}}\mathbf{1} - \sum_{k=1}^{N}\alpha_k;$$

$$\boldsymbol{\alpha}_c = \begin{pmatrix} \alpha_1 c_1 \\ : \\ \alpha_N c_N \end{pmatrix}; \qquad \boldsymbol{\Phi}^{\mathbf{T}} = \begin{pmatrix} \mathbf{x}_1^{\mathbf{T}} \\ : \\ \mathbf{x}_N^{\mathbf{T}} \end{pmatrix}$$

Gradiente respecto a los parámetros de lla función:

$$\nabla L_{\mathbf{w}} = \mathbf{w} - \boldsymbol{\alpha}_c^{\mathbf{T}}\boldsymbol{\Phi}^{\mathbf{T}} = 0 \Rightarrow \mathbf{w} = \boldsymbol{\alpha}_c^{\mathbf{T}}\boldsymbol{\Phi}^{\mathbf{T}}$$

$$\nabla L_{w_0} = \boldsymbol{\alpha}_c^{\mathbf{T}}\mathbf{1} = 0 \qquad (2)$$

Sustituyendo las expresiones anteriores en (1) se obtiene la forma dual que es la que se debe optimizar (minimizar) respecto a los multiplicadores de Lagrange y con la restricción (2)

$$L = \boldsymbol{\alpha}^{\mathbf{T}}\mathbf{1} - \tfrac{1}{2}\boldsymbol{\alpha}_c^{\mathbf{T}}\boldsymbol{\Phi}^{\mathbf{T}}\boldsymbol{\Phi}\boldsymbol{\alpha}_c$$

$$\boldsymbol{\alpha} = \begin{pmatrix} \alpha_1 \\ : \\ \alpha_N \end{pmatrix}$$

5.2 SVC: Funciones de Kernel.

Función de Kernel NO lineal:

- Previamente al diseño de la función discriminante, se realiza una transformación no lineal ϕ sobre los datos, llamada función base. Se pasa de un espacio vectorial a un espacio de Hilbert, de dimensión mayor *d*'>=*d*.

$$\boldsymbol{\varphi}(\mathbf{x}_n) = \begin{pmatrix} \phi_1(\mathbf{x}_n) \\ : \\ \phi_{d'}(\mathbf{x}_n) \end{pmatrix}; \qquad \boldsymbol{\Phi} = \begin{pmatrix} \boldsymbol{\varphi}^{\mathbf{T}}(\mathbf{x}_1) \\ : \\ \boldsymbol{\varphi}^{\mathbf{T}}(\mathbf{x}_N) \end{pmatrix}$$

- La función de Kernel es igual al producto escalar de los vectores tranformados y a la vez cumple la propiedad: de que depende del producto escalar (inner product) de los vectores de datos.

$$K : R^d \rightarrow R \qquad K(\mathbf{x}_k, \mathbf{x}_n) = \boldsymbol{\varphi}(\mathbf{x}_k)^{\mathbf{T}} \boldsymbol{\varphi}(\mathbf{x}_n) = f\left(\mathbf{x}_k{}^{\mathbf{T}} \mathbf{x}_n\right)$$

- La matriz de Kernel (NxN) contiene todos los productos escalares entre los N datos. En general la matriz de Kernel debe marcar diferencias entre productos de elementos de una misma clase y productos de elementos de clases diferentes.

$$\mathbf{K} = \begin{pmatrix} K(\mathbf{x}_1, \mathbf{x}_1) & : & K(\mathbf{x}_1, \mathbf{x}_N) \\ : & : & : \\ K(\mathbf{x}_N, \mathbf{x}_1) & : & K(\mathbf{x}_N, \mathbf{x}_N) \end{pmatrix} = \boldsymbol{\Phi}^{\mathbf{T}} \boldsymbol{\Phi}$$

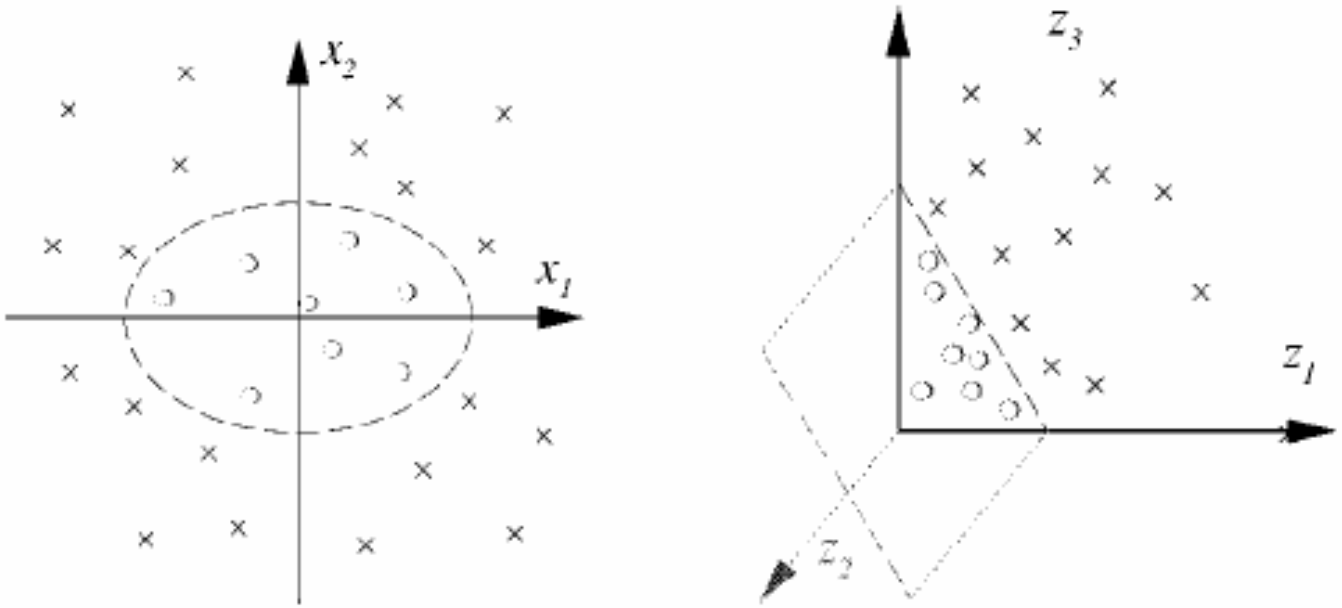


Fig. 4. Two-dimensional classification example. (a) Using the second-order monomials x_1^2, $\sqrt{2}x_1x_2$ and x_2^2 as features a separation in feature space can be found using a *linear* hyperplane. (b) In input space this construction corresponds to a *nonlinear* ellipsoidal decision boundary (figure from [48]).

B. Schölkopf and A. J. Smola, *Learning with Kernels*. Cambridge, MA: MIT Press, 2001.

5.2 SVC: Funciones de Kernel.

Ejemplos:

- Transformación no lineal ϕ sobre los datos Gaussiana: Función base polinomio de segundo grado

$$\boldsymbol{\varphi}(\mathbf{x}_n) = \boldsymbol{\varphi}\left(\begin{pmatrix} x_n[1] \\ x_n[2] \end{pmatrix}\right) = \begin{pmatrix} (x_n[1])^2 \\ \sqrt{2}x_n[1]x_n[2] \\ (x_n[d])^2 \end{pmatrix} \Rightarrow$$

$$K(\mathbf{x}_k, \mathbf{x}_n) = \boldsymbol{\varphi}^{\mathbf{T}}(\mathbf{x}_k)\boldsymbol{\varphi}(\mathbf{x}_n) = (\mathbf{x}_k^{\mathbf{T}}\mathbf{x}_n)^2$$

- Kernel Gaussiano (Función base radial ya que solo depende de la distancia eucídea).

$$K(\mathbf{x}_k, \mathbf{x}_n) = \boldsymbol{\varphi}^{\mathbf{T}}(\mathbf{x}_k)\boldsymbol{\varphi}(\mathbf{x}_n) = \exp\left(-\tfrac{1}{\sigma^2}\|\mathbf{x}_k - \mathbf{x}_n\|^2\right)$$

La mayor dificultad al aplicar SVC no lineal, consiste en hallar la función de Kernel decuada que trasnsforme los datos a un nuevo espacio en el que si sean separables linealmente. El tipo de función depende de la aplicación.

5.3 SVC: Non Linear Classifier

El clasificador SV no lineal se diseña en dos etapas:

1.- Transformación no lineal sobre los datos

2.- Desarrollo del algoritmo SV de forma lineal sobre los nuevos vectores de datos transformados.

Función a diseñar: $$g(\mathbf{x}) = \mathbf{w}^T \boldsymbol{\varphi}(\mathbf{x}) + w_0$$

Función a minimizar (1): $$L = \tfrac{1}{2}\mathbf{w}^{\mathrm{T}}\mathbf{w} - \boldsymbol{\alpha}_c^{\mathrm{T}}\boldsymbol{\Phi}^{\mathrm{T}}\mathbf{w} + w_0\boldsymbol{\alpha}_c^{\mathrm{T}}\mathbf{1} - \sum_{k=1}^{N}\alpha_k;$$

Operando de forma análoga que con el método lineal:

$$L = \boldsymbol{\alpha}^{\mathrm{T}}\mathbf{1} - \tfrac{1}{2}\boldsymbol{\alpha}_c^{\mathrm{T}}\boldsymbol{\Phi}^{\mathrm{T}}\boldsymbol{\Phi}\boldsymbol{\alpha}_c = \boldsymbol{\alpha}^{\mathrm{T}}\mathbf{1} - \tfrac{1}{2}\boldsymbol{\alpha}_c^{\mathrm{T}}\mathbf{K}\boldsymbol{\alpha}_c \qquad \boldsymbol{\alpha}_c^{\mathrm{T}}\mathbf{1} = 0$$

6 CONCLUSIONES

BÚSQUEDA DE DISCRIMINANTE LINEAL

- INTERPRETACIONES GEOMÉTRICAS: El discriminante lineal mide la distancia de los vectores al hiperplano separador
- C=2 categorías separables: Algoritmo de Gradiente descendente mediante la función perceptron, Generalizable a más categorías.
- C=2 categorías NO separables: Algoritmo de MMSE – LMS,Generalizable a más categorías
- C>2 Categorías: Se plantea el problema como c(c-1)/2 problemas de dos clases.

SV Classifier:

- NO Lineal: Es útil especialmente en casos no separables linealmente.
- Dificultad: Hallar la función de Kernel adecuada