Sistemas Inteligentes

Escuela Técnica Superior de Informática

Universitat Politècnica de València

Tema B2T2

Aprendizaje de funciones discriminantes: Perceptrón

SIN

Índice

Índice

Extracción de carácterísticas y Espacio de Representación

Espacio de representación:

- Espacio, frecuentemente vectorial, donde se representan los objetos
- La representación de un objeto se obtiene mediante técnicas de *preproceso* y *extracción de propiedades o carácterísticas*

Extracción de características; propiedades deseables:

- *Continuidad y capacidad de discriminación*: La similitud entre las representaciones de dos objetos debe estar en relación directa con la similitud con la que se suelen *percibir* dichos objetos:

  objetos de la misma clase deberían tener representaciones similares, mientras que objetos de clases distintas deberían tener representaciones claramente diferentes.

- *Invarianza a las transformaciones y distorsiones usuales*: Diferentes instancias de un mismo objeto deberían tener representaciones similares.

Un ejemplo clásico: clasificación de irisáceas

- Problema "académico" tradicional, introducido por *Fisher* en 1936
- Se trata de clasificar correctamente flores de la familia *iris* (liliáceas) en base a las dimensiones de sus *pétalos* y *sépalos*.
- El conjunto de datos consta de las mediciones de 150 ejemplares de tres subclases; *Setosa, Versicolor* y *Virgínica*.
- Se utiliza frecuentemente como tarea-ejemplo para comparar las prestaciones y posibilidades de distintos métodos de Análisis de Datos, Reconocimiento de Formas y Aprendizaje Automático.


Setosa


Versicolor


Virgínica

Ejemplares de tres variedades de irisáceas

Setosa

Versicolor

Virgínica

Iris: Espacio de Representación

Vectores de características en $\mathbb{R}^4$ de tres clases de irisáceas. Componentes:

LONGITUD-SÉPALOS ANCHURA-SÉPALOS LONGITUD-PÉTALOS ANCHURA-PÉTALOS

Iris Setosa

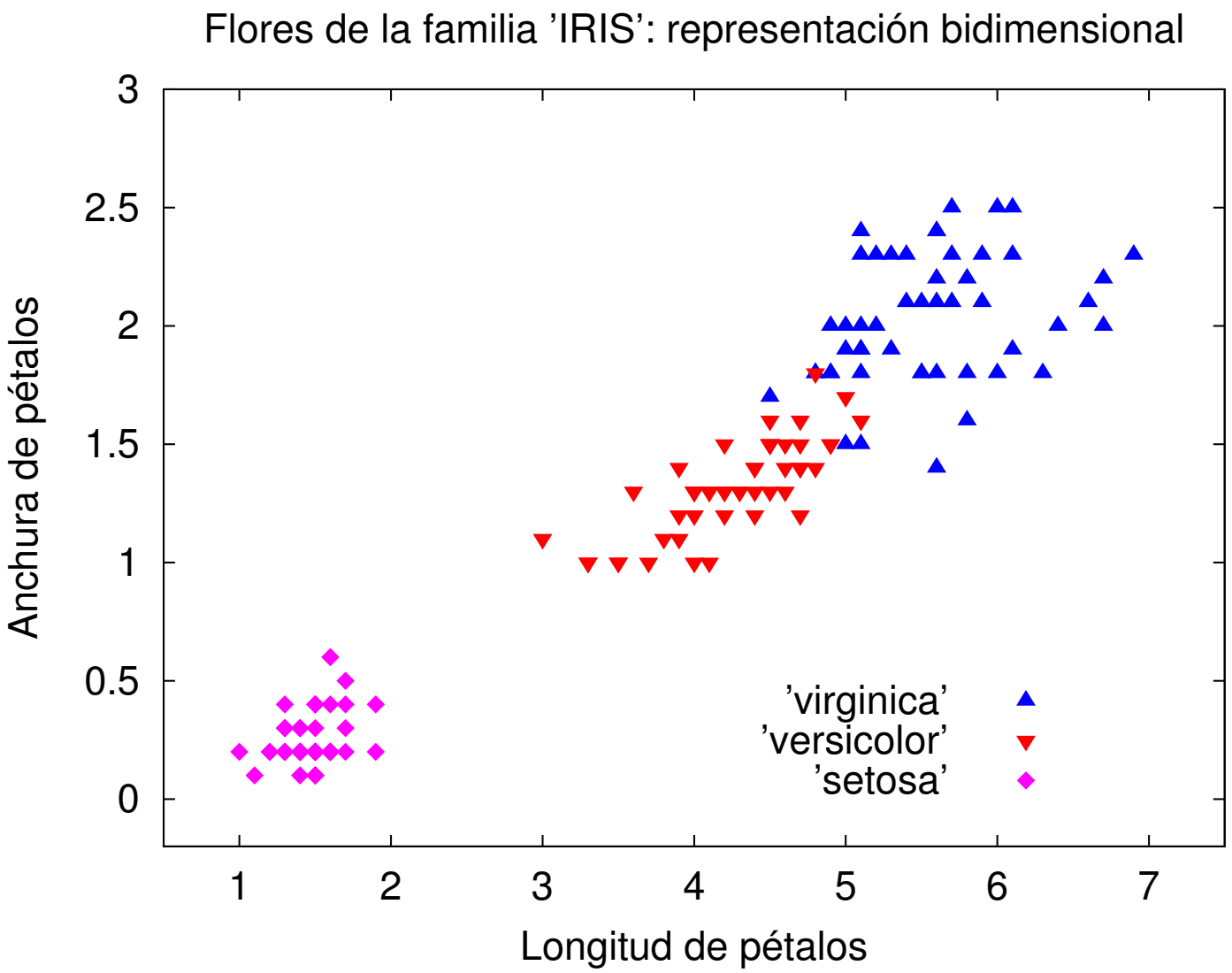
5.1 3.5 1.4 0.2
4.9 3.0 1.4 0.2
4.7 3.2 1.3 0.2
4.6 3.1 1.5 0.2
5.0 3.6 1.4 0.2
...

Iris Versicolor

7.0 3.2 4.7 1.4
6.4 3.2 4.5 1.5
6.9 3.1 4.9 1.5
5.5 2.3 4.0 1.3
6.5 2.8 4.6 1.5
...

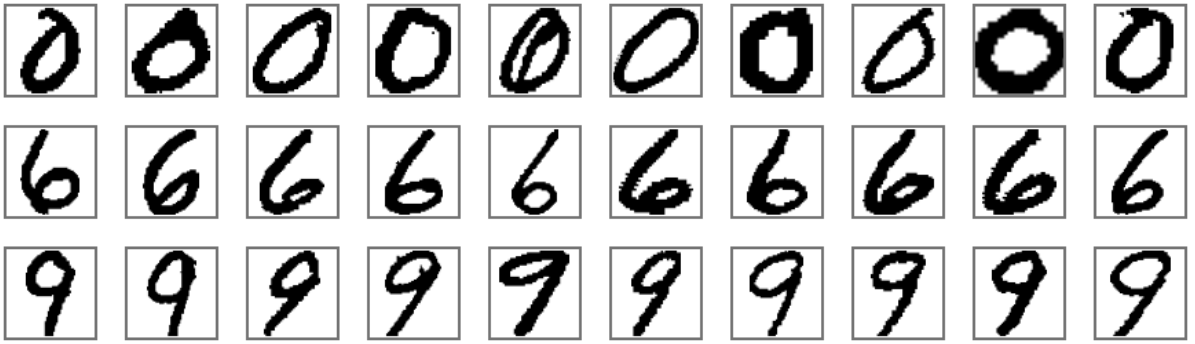
Iris Virgínica

6.3 3.3 6.0 2.5
5.8 2.7 5.1 1.9
7.1 3.0 5.9 2.1
6.3 2.9 5.6 1.8
6.5 3.0 5.8 2.2
...

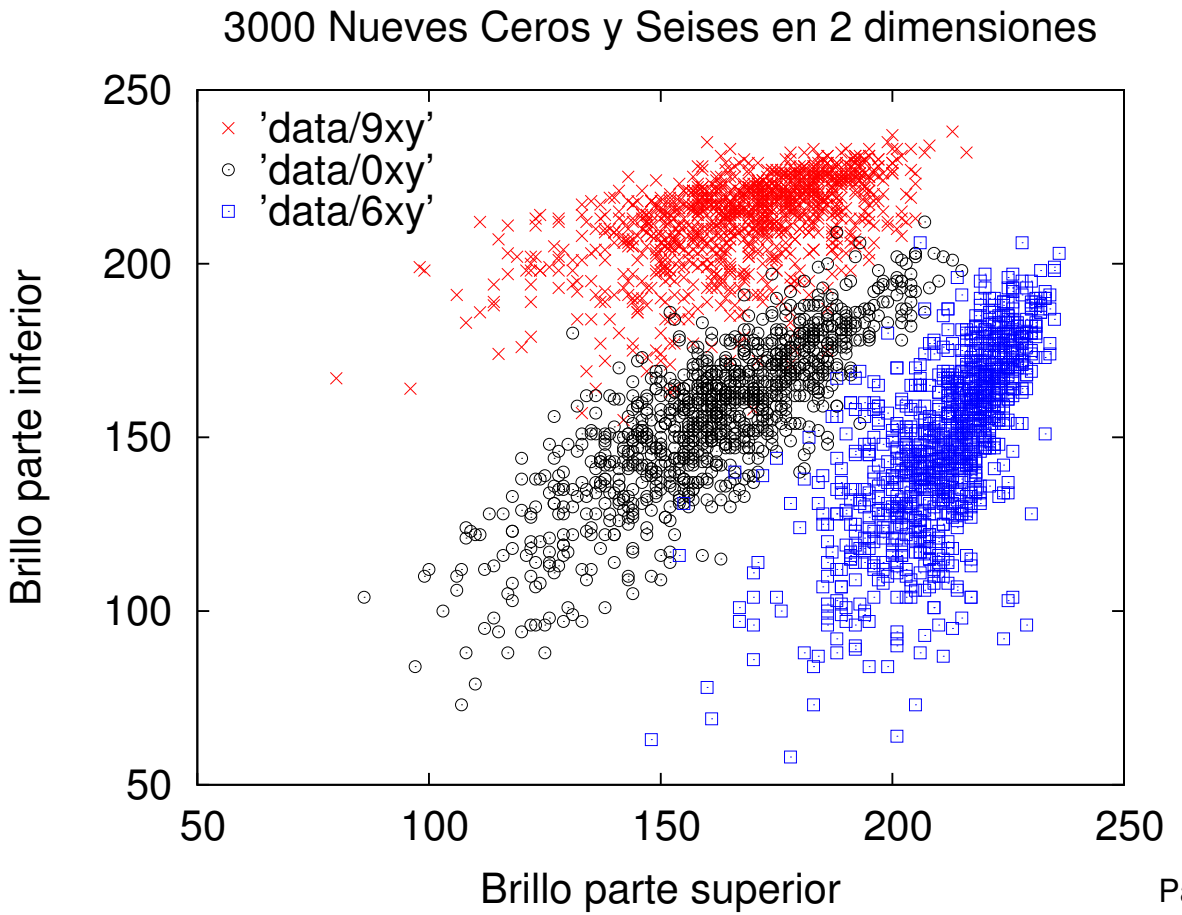


Otro ejemplo: Imágenes de digitos manuscritos

Ejemplos de imágenes normalizadas de los dígitos manuscritos *cero, seis* y *nueve*. Las imágenes de *seis* son mas *claras* por arriba y las de *nueve* por abajo.



3000 ejemplos en un espacio de representación $\mathbb{R}^2$ correspondiente a la extracción de dos propiedaes de cada imagen: *brillo en la mitad superior* y *brillo en la mitad inferior*.



Clases, espacio de representación y clasificador: notación

- *Clases*: $\mathbb{C}$, $\mathbb{C} = \{1, 2, \ldots, C\}$ si no se dice lo contrario
  - Cada *objeto* (o su señal) se manifiesta en un *Espacio Primario* o "Universo", U
  - Suponemos que cada objeto $x \in U$ pertenece a una única *clase* $c(x) \in \mathbb{C}$
  - $\mathbb{C}$ denota el conjunto de posibles *identificadores* o *etiquetas de clase*

Clases, espacio de representación y clasificador: notación

- *Clases*: $\mathbb{C}$, $\mathbb{C} = \{1, 2, \ldots, C\}$ si no se dice lo contrario
  - Cada *objeto* (o su señal) se manifiesta en un *Espacio Primario* o "Universo", U
  - Suponemos que cada objeto $x \in U$ pertenece a una única *clase* $c(x) \in \mathbb{C}$
  - $\mathbb{C}$ denota el conjunto de posibles *identificadores* o *etiquetas de clase*
- *Espacio de representación*: E, generalmente $E = \mathbb{R}^D$
  - Sea $\boldsymbol{y} = \boldsymbol{y}(x)$ el resultado del preproceso y extracción de características aplicados a un objeto cualquiera $x \in U$
  - E incluye todos los posibles resultados: $\{\boldsymbol{y} \ : \ \boldsymbol{y} = \boldsymbol{y}(x), \, x \in U\} \subset E$
  - Dado que dos objetos distintos en U pueden tener la misma representación en E, no se garantiza que cada punto en E corresponda a una única clase

Clases, espacio de representación y clasificador: notación

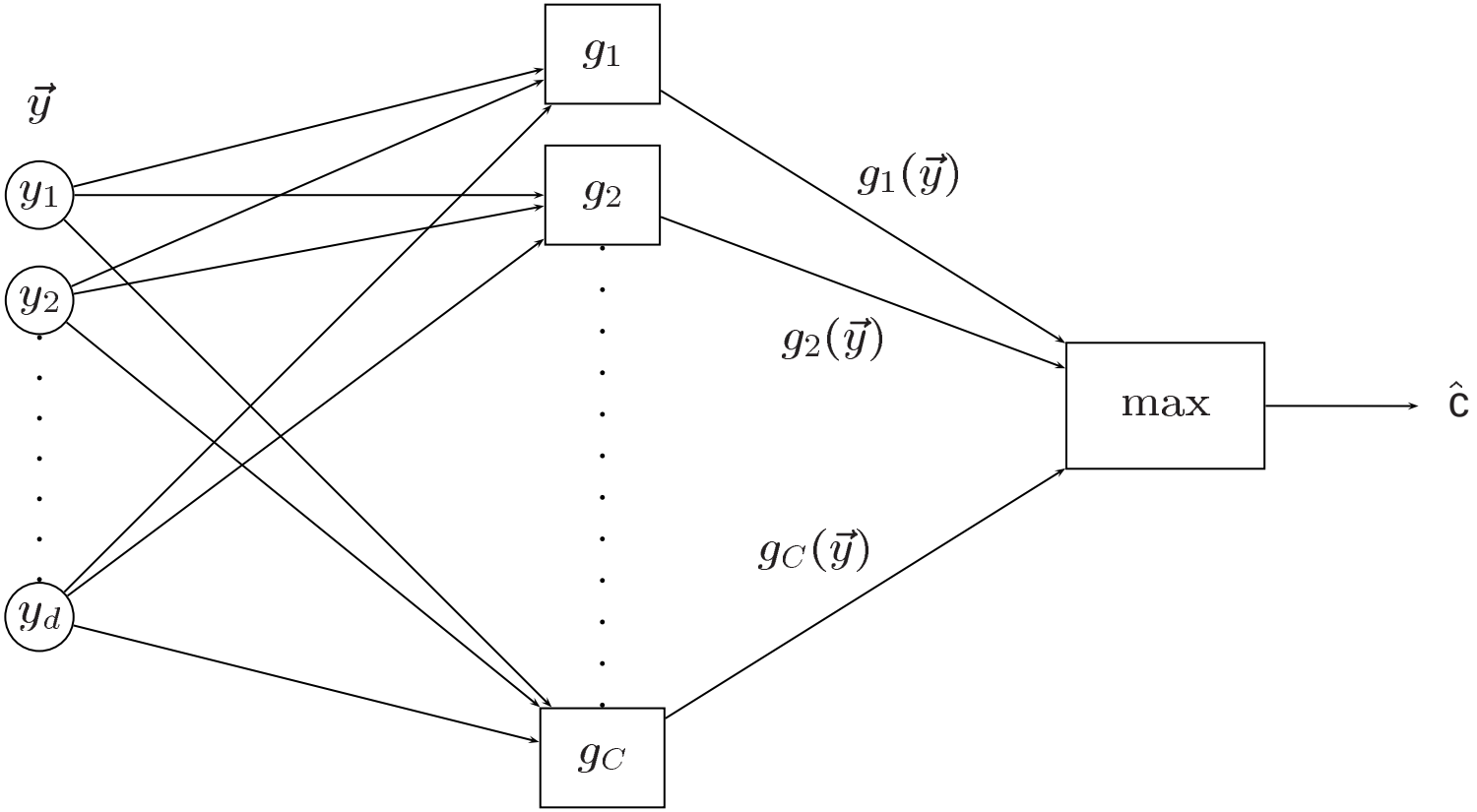
- *Clases*: $\mathbb{C}$, $\mathbb{C} = \{1, 2, \ldots, C\}$ si no se dice lo contrario
  - Cada *objeto* (o su señal) se manifiesta en un *Espacio Primario* o "Universo", U
  - Suponemos que cada objeto $x \in U$ pertenece a una única *clase* $c(x) \in \mathbb{C}$
  - $\mathbb{C}$ denota el conjunto de posibles *identificadores* o *etiquetas de clase*
- *Espacio de representación*: E, generalmente $E = \mathbb{R}^D$
  - Sea $\boldsymbol{y} = \boldsymbol{y}(x)$ el resultado del preproceso y extracción de características aplicados a un objeto cualquiera $x \in U$
  - E incluye todos los posibles resultados: $\{\boldsymbol{y} \ : \ \boldsymbol{y} = \boldsymbol{y}(x), x \in U\} \subset E$
  - Dado que dos objetos distintos en U pueden tener la misma representación en E, no se garantiza que cada punto en E corresponda a una única clase
- *Clasificador*: $G : E \to \mathbb{C}$
  - G se aprende con N *muestras etiquetadas* $(\boldsymbol{y}_1, c_1), \ldots, (\boldsymbol{y}_N, c_N) \in E \times \mathbb{C}$
  - Para un nuevo objeto $x \in U$ se estima su clase como $\hat{c} = \hat{c}(x) = G(\boldsymbol{y}(x))$. El objetivo es acertar la clase correcta; es decir, $\hat{c} = c(x)$, el mayor número de veces posible.

Índice

Clasificadores y Funciones Discriminantes (FD)

Todo clasificador G en C clases puede expresarse mediante C *funciones discriminantes* $g_c : E \to \mathbb{R}, \quad 1 \leq c \leq C,$ y la correspondiente *regla de clasificación*:

$$G = (g_1, g_2, \ldots, g_C), \qquad \hat{c} = G(\boldsymbol{y}) \equiv \underset{1 \leq c \leq C}{\text{argmax}} \; g_c(\boldsymbol{y})$$



Fronteras de decisión o de clasificación

Un clasificador divide el espacio de representación en C *regiones de decisión*, $R_1, \ldots, R_C$:

$$R_j = \{\mathbf{y} \in E : \quad g_j(\mathbf{y}) > g_i(\mathbf{y}) \quad i \neq j,\ 1 \leq i \leq C\}$$

- *Frontera de Decisión entre dos clases* i, j:
  Lugar geométrico de los puntos $\mathbf{y} \in E$ para los que $g_i(\mathbf{y}) = g_j(\mathbf{y})$

  En general son *Hipersuperficies* definidas por las ecuaciones:

$$g_i(\mathbf{y}) - g_j(\mathbf{y}) = 0 \qquad i \neq j, \ \ 1 \leq i, j \leq C$$

  - Si $E \equiv \mathbb{R}^3$ las fronteras son superficies (ej. *planos*)
  - Si $E \equiv \mathbb{R}^2$ las fronteras son líneas (ej. *rectas*)
  - Si $E \equiv \mathbb{R}$ las fronteras son puntos

- *Frontera de Decisión de una clase* i:
  Lugar geométrico de los puntos $\mathbf{y} \in E$ para los que:

$$g_i(\mathbf{y}) \ = \ \max_{j \neq i} \ g_j(\mathbf{y})$$

$g_s(\vec{y}) = 4$

$g_v(\vec{y}) = y_1 + 2y_2$

Eq. frontera: $g_s(\vec{y}) = g_v(\vec{y})$

$4 = y_1 + 2y_2$

$y_1 + 2y_2 - 4 = 0$

$y_1 = 0 \rightarrow y_2 = \frac{4}{2} = 2$

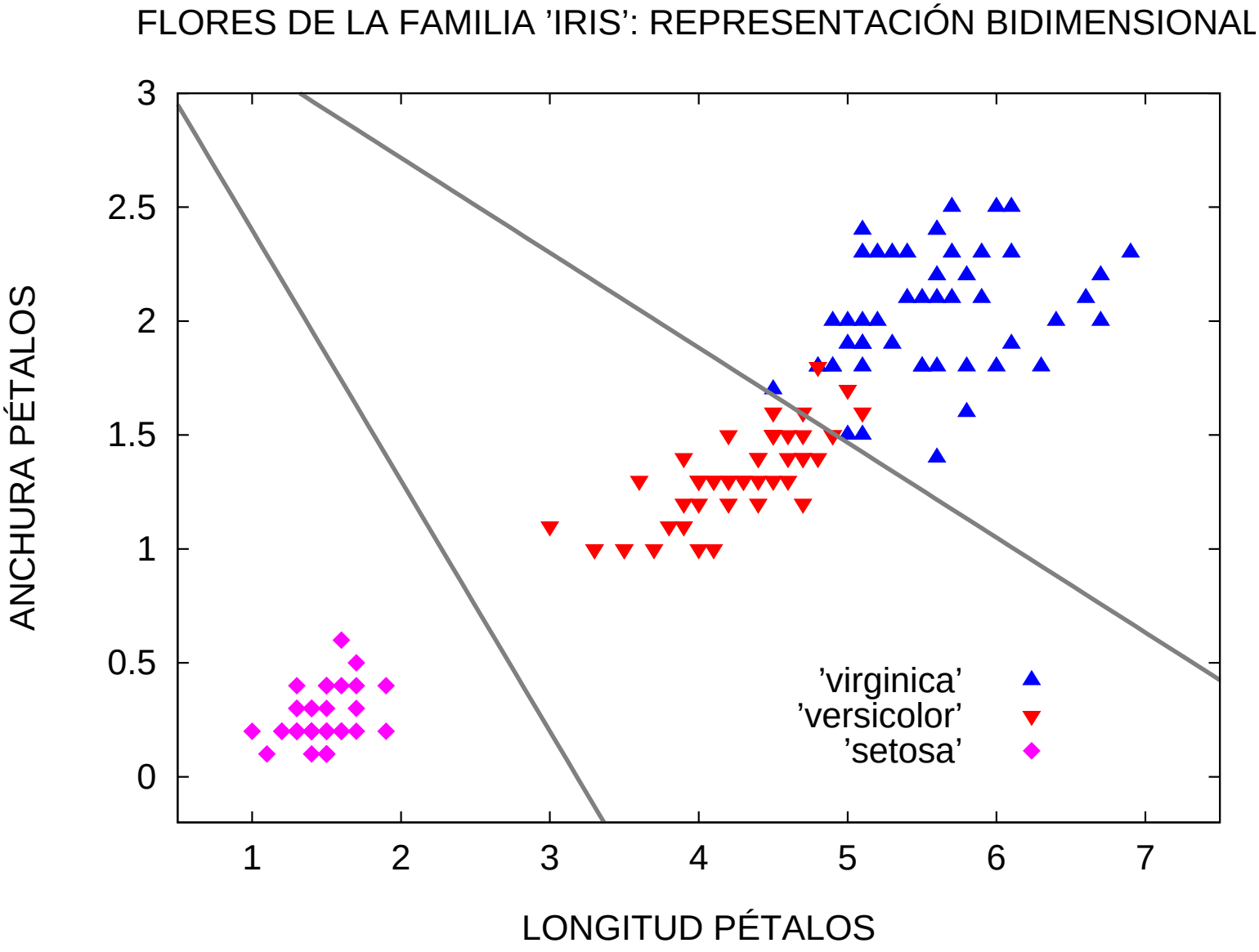
$y_2 = 0 \rightarrow y_1 = 4$

$G(\vec{y}') = G\begin{pmatrix}1.5\\0.5\end{pmatrix} = \hat{c}(\vec{y}') = \rightarrow$ $g_1(\vec{y}') = 4$, $g_2(\vec{y}') = 1.5 + 1 = 2.5$ $\}$ $\max(4, 2.5) = 4$

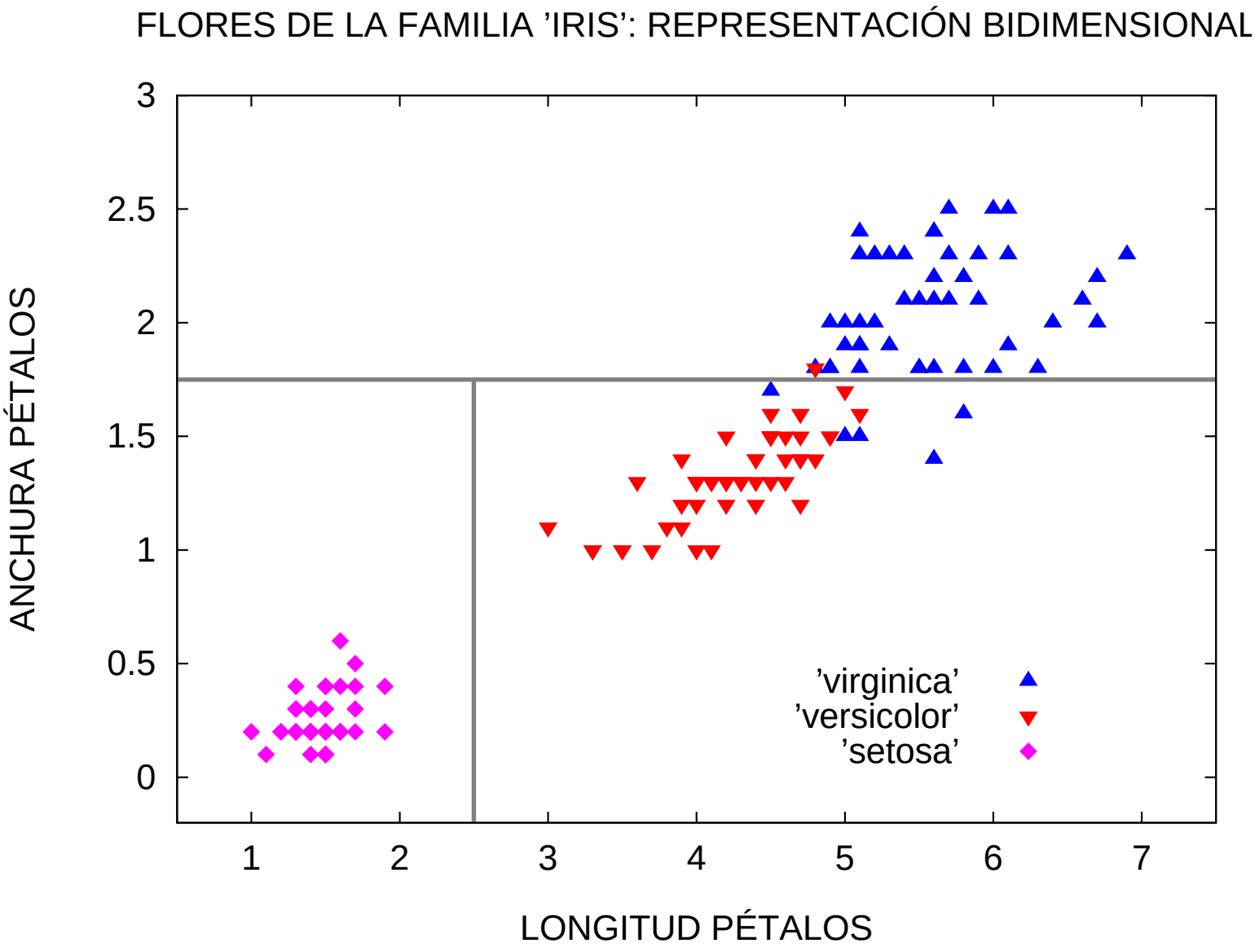
$G(\vec{y}'') = G\begin{pmatrix}3\\1\end{pmatrix} = \hat{c}(\vec{y}'') =$ asignamos $g_s(\vec{y}') = 5 \equiv \hat{c}(\vec{y}')$ a $G(5,1)$.

$g_1(\vec{y}'') = 4$, $g_2(\vec{y}'') = 3 + 2 = 5$ $\}$ $\rightarrow \hat{c}(\vec{y}'') = 1$

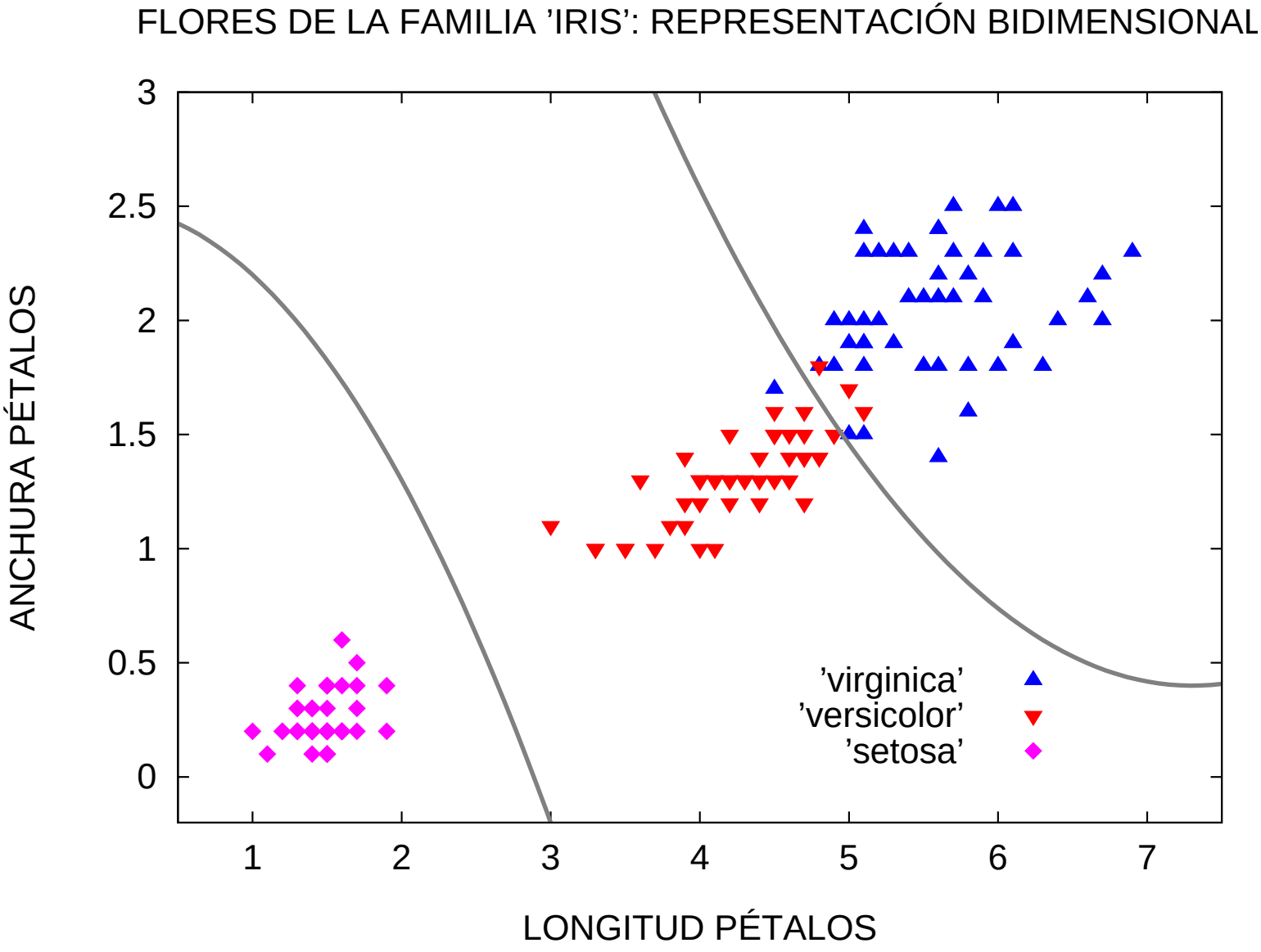
Fronteras de decisión en el problema de las irisáceas: Fronteras lineales



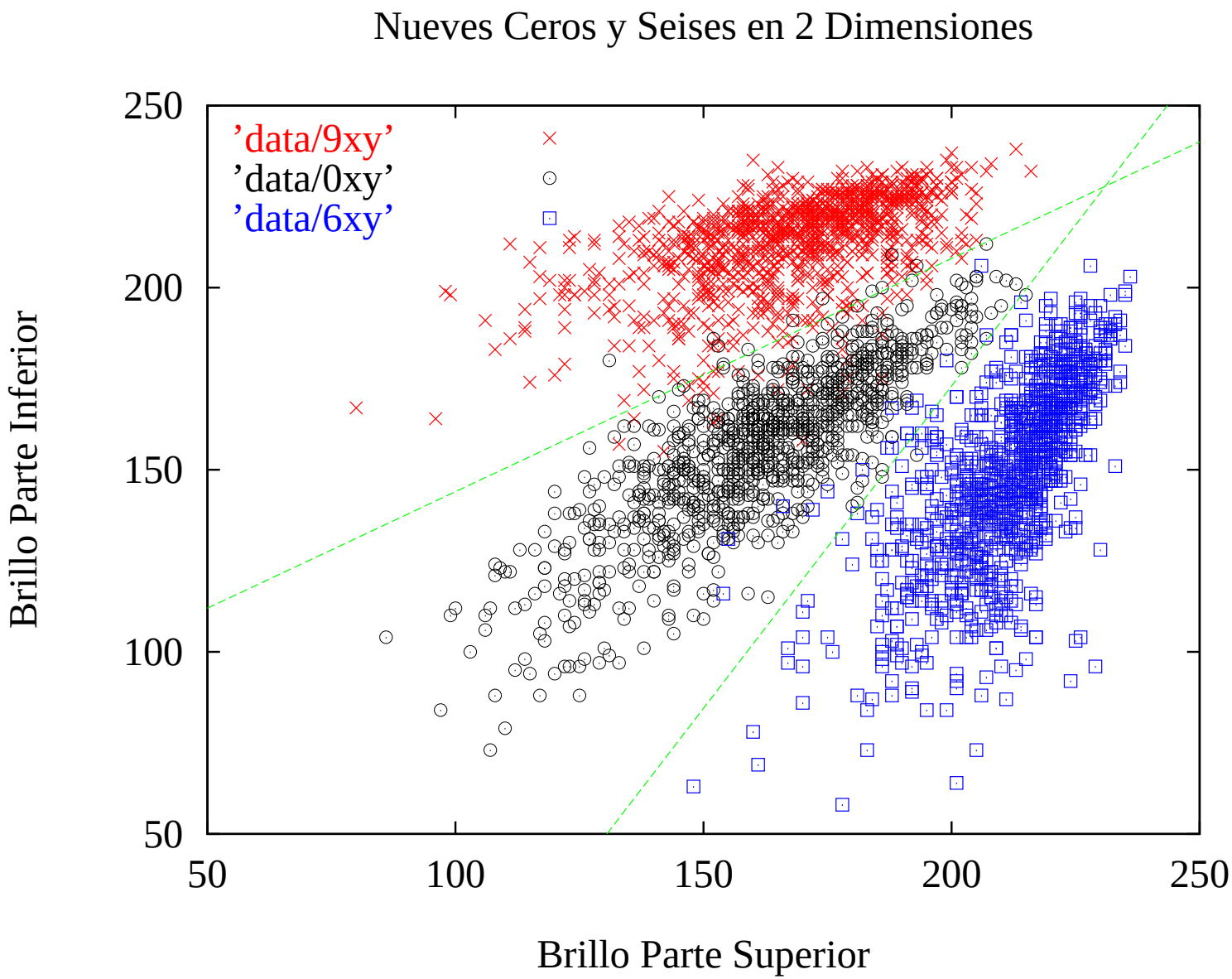
Fronteras de decisión en el problema de las irisáceas: Fronteras lineales paralelas a los ejes



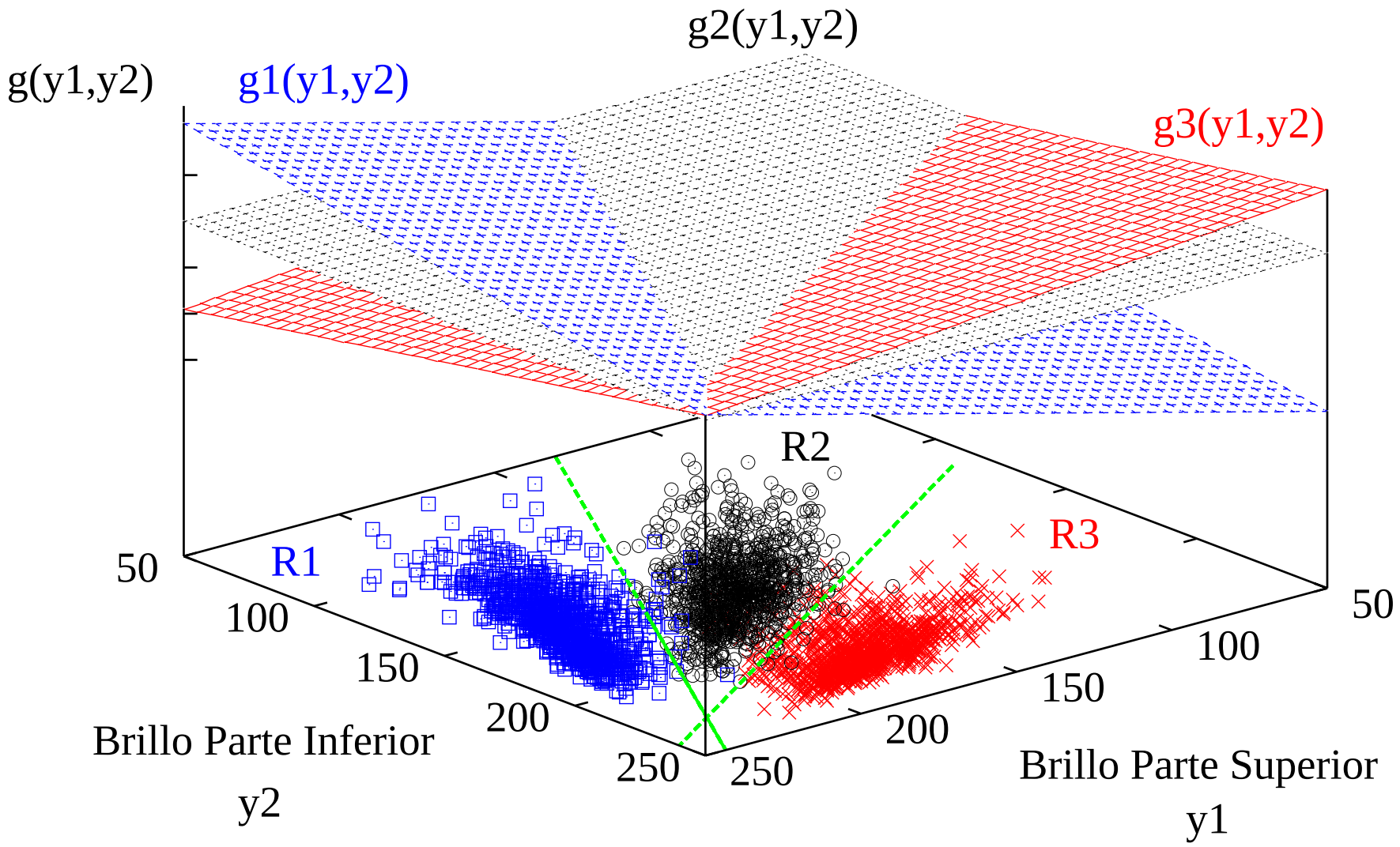
Fronteras de decisión en el problema de las irisáceas: Fronteras no lineales



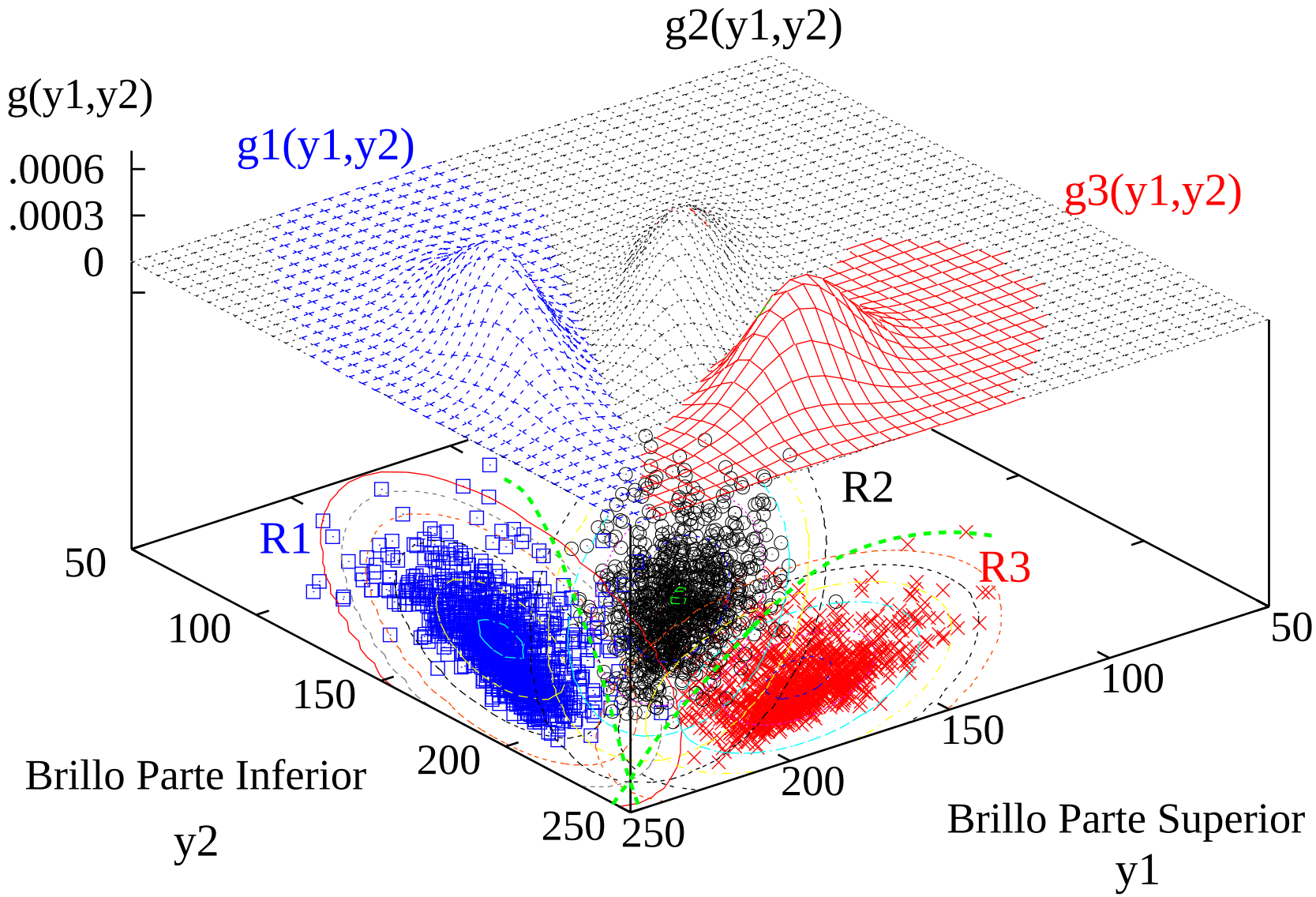
Fronteras de decisión de un clasificador de imágenes de dígitos manuscritos representadas en dos dimensiones



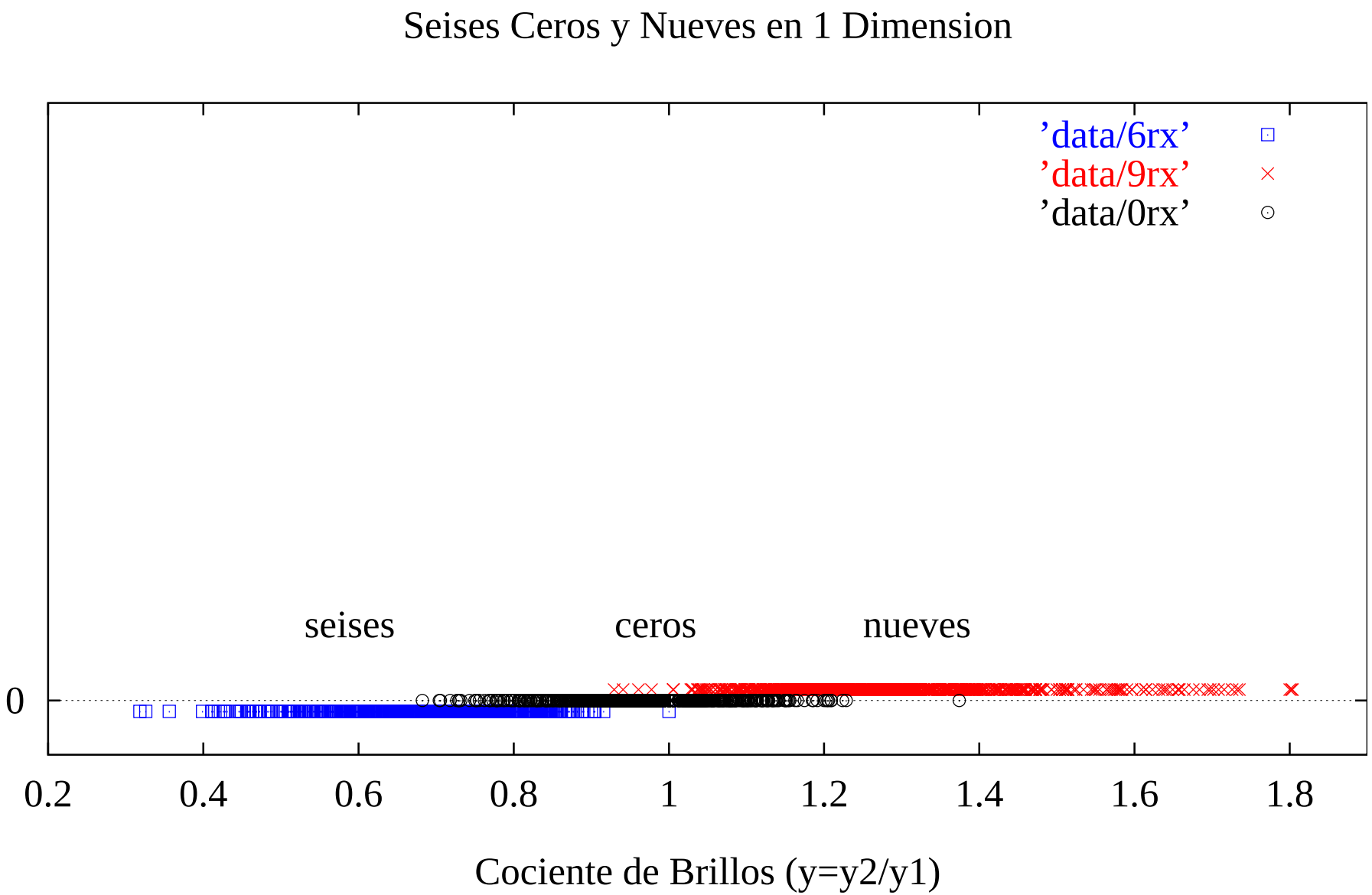
Fronteras de de Decisión y Funciones Discriminantes (lineales)



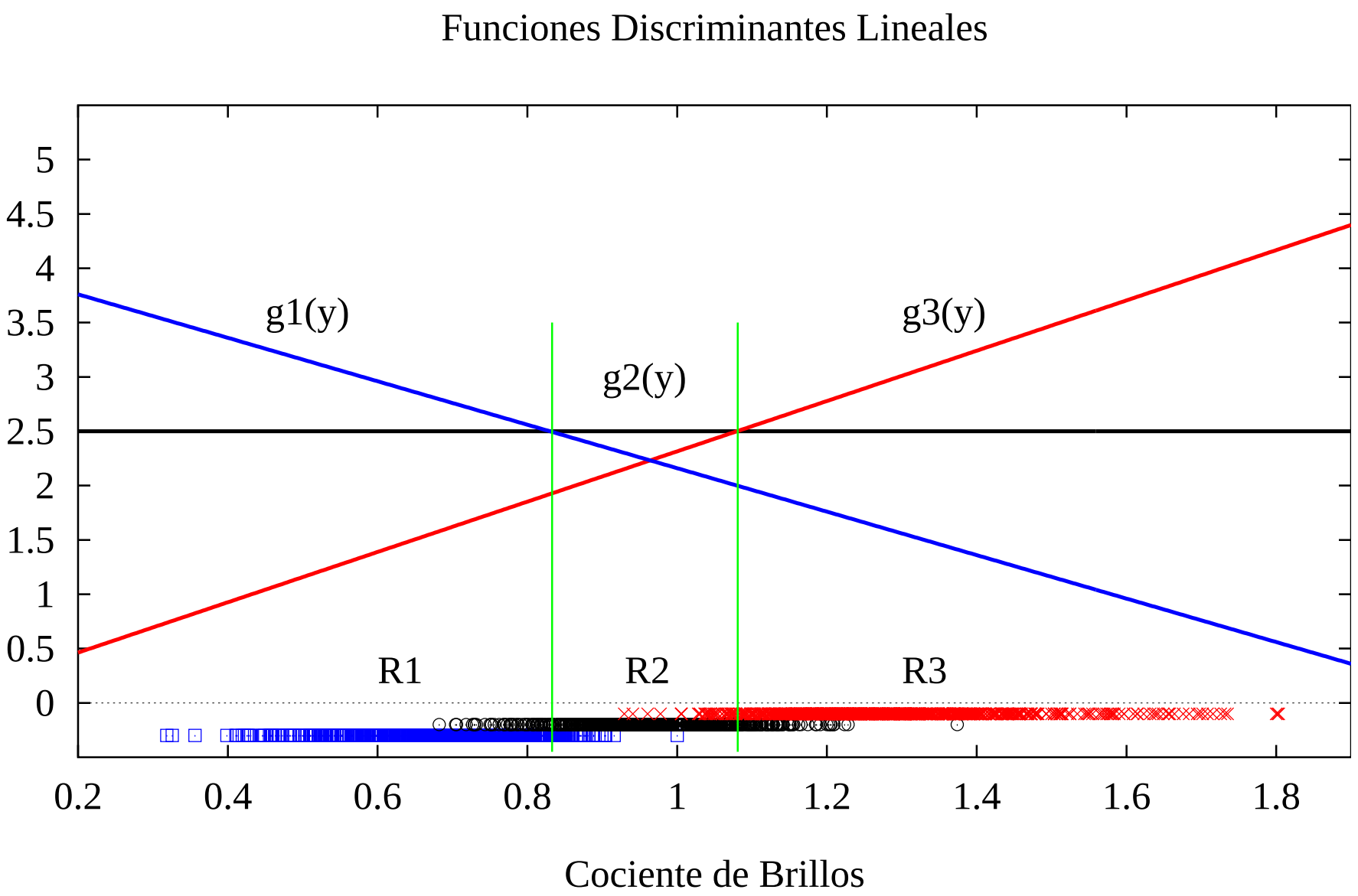
Fronteras de Decisión y Funciones Discriminantes (Gaussianas)



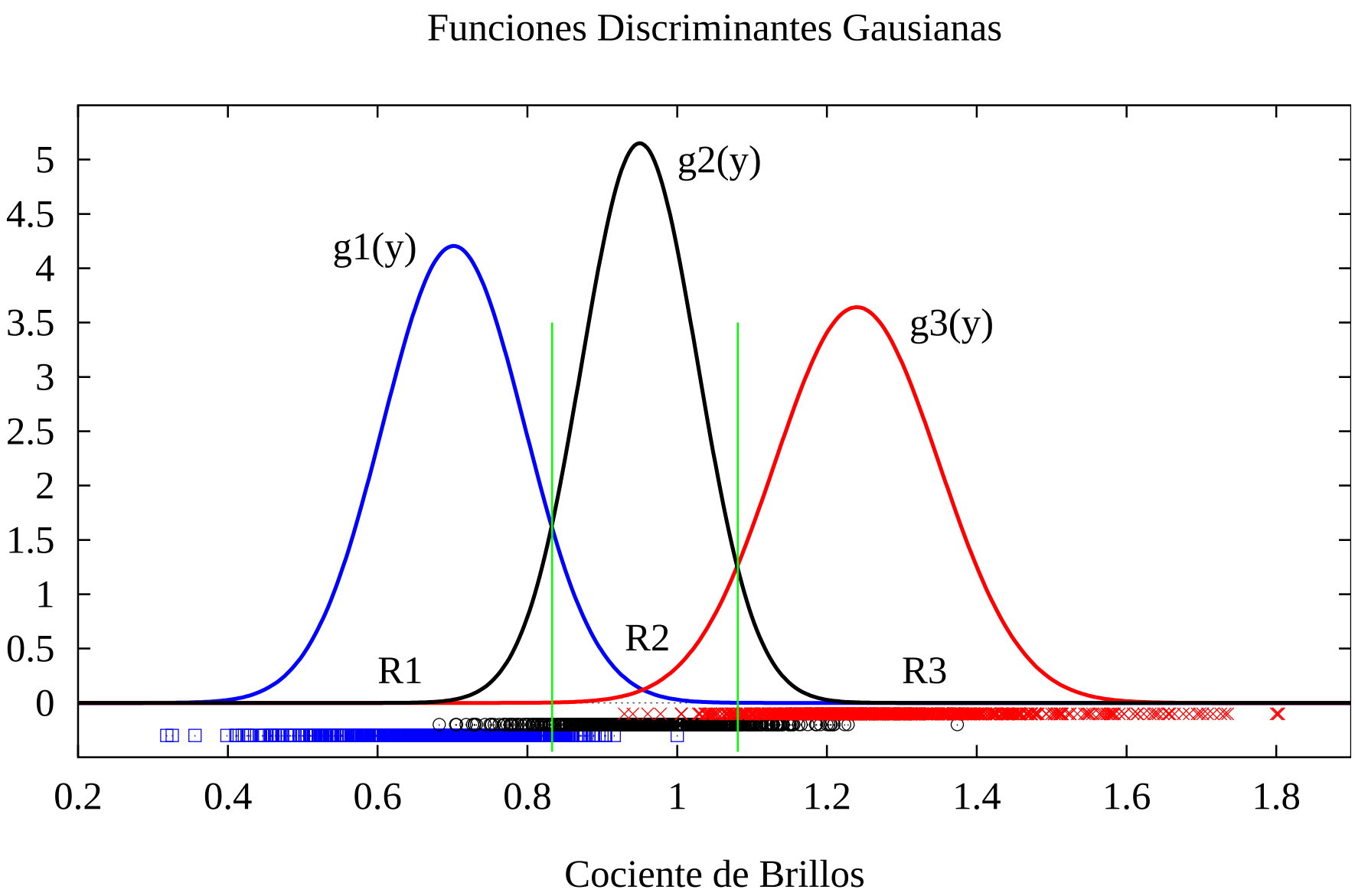
Ejemplos en una dimensión



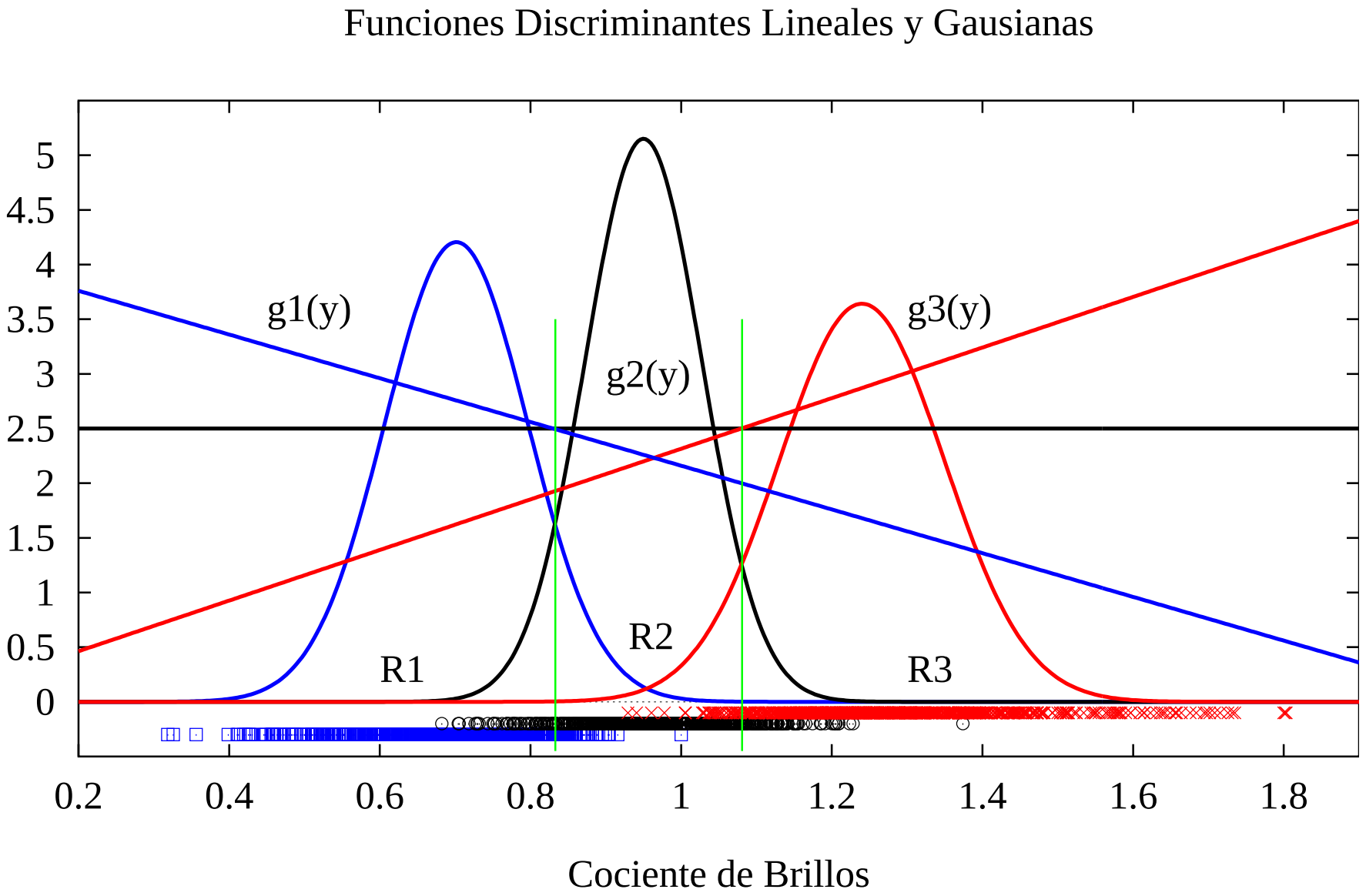
Funciones Discriminantes Lineales y sus Fronteras de Decisión



Funciones Discriminantes Gausianas y sus Fronteras de Decisión



Equivalencia de funciones discriminantes



Clasificadores equivalentes

Dos clasificadores $(g_1, \ldots, g_C)$ y $(g'_1, \ldots, g'_C)$ son equivalentes si inducen las mismas fronteras de decisión; es decir:

$$g_i(\mathbf{y}) > g_j(\mathbf{y}) \Leftrightarrow g'_i(\mathbf{y}) > g'_j(\mathbf{y}) \qquad \forall j \neq i, \;\; \forall \mathbf{y} \in E$$

Sea $f : \mathbb{R} \to \mathbb{R}$ cualquier función monótona creciente. Entonces los siguientes clasificadores son equivalentes:

$$(g_1, \ldots, g_C), \qquad (f(g_1), \ldots, f(g_C))$$

Ejemplos:

$f(z) = az + b$ con $a > 0$

$f(z) = \log z$ con $g_i(\mathbf{y}) > 0$, $1 \leq i \leq C$

Funciones discriminantes lineales y cuadráticas

Funciones discriminantes *lineales*:

$$g(\mathbf{y}) = \sum_{i=1}^{d} a_i y_i + a_0 = \mathbf{a}^t \mathbf{y} + a_0$$

$\mathbf{a}$ se denomina *vector de pesos* y a_0 *peso umbral*.

- Número de parámetros: $d + 1$ (*lineal* con la dimensión)
- En general, *las fronteras de decisión son hiperplanos*.

Funciones discriminantes *cuadráticas*:

$$g(\mathbf{y}) = \sum_{i=1}^{d} \sum_{j=1}^{d} a_{ij} y_i y_j + \sum_{i=1}^{d} a_i y_i + a_0 = \mathbf{y}^t A \mathbf{y} + \mathbf{a}^t \mathbf{y} + a_0$$

A se denomina *matriz de pesos* y a_0 *peso umbral*.

- Número de parámetros: *cuadrático* con la dimensión d.
- En general, *las fronteras de decisión son hipercuádricas*.

Índice

Funciones Discriminantes Lineales (FDL)

Un clasificador es *lineal* si sus Funcions Discriminantes son *funciones lineales* de las componentes de los vectores de E. Sea $\boldsymbol{y} \in E \equiv \mathbb{R}^D$ la representación de un objeto cualquiera.

$$g_c(\boldsymbol{y}) \;=\; \sum_{j=1}^{D} a_{cj} \cdot y_j + a_{c0} \;=\; \boldsymbol{a}_c^t \boldsymbol{y} + a_{c0}, \;\; 1 \leq c \leq C$$

Funciones Discriminantes Lineales (FDL)

Un clasificador es *lineal* si sus Funcions Discriminantes son *funciones lineales* de las componentes de los vectores de E. Sea $\boldsymbol{y} \in E \equiv \mathbb{R}^D$ la representación de un objeto cualquiera.

$$g_c(\boldsymbol{y}) \;=\; \sum_{j=1}^{D} a_{cj} \cdot y_j + a_{c0} \;=\; \boldsymbol{a}_c^t \boldsymbol{y} + a_{c0}, \;\; 1 \leq c \leq C$$

Notación homogénea (*notar los cambios de letra* $\boldsymbol{a} \rightarrow \mathbf{a}$, $\boldsymbol{y} \rightarrow \mathbf{y}$):

$$\mathbf{y} = (1, y_1, ..., y_D)^t, \;\; \mathbf{a}_c = (a_{c0}, a_{c1}, \cdots, a_{cD})^t$$

$$g_c(\boldsymbol{y}) = \mathbf{a}_c^t \mathbf{y}$$

Funciones Discriminantes Lineales (FDL)

Un clasificador es *lineal* si sus Funcions Discriminantes son *funciones lineales* de las componentes de los vectores de E. Sea $\boldsymbol{y} \in E \equiv \mathbb{R}^D$ la representación de un objeto cualquiera.

$$g_c(\boldsymbol{y}) \;=\; \sum_{j=1}^{D} a_{cj} \cdot y_j + a_{c0} \;=\; \boldsymbol{a}_c^t \boldsymbol{y} + a_{c0}, \;\; 1 \leq c \leq C$$

Notación homogénea (*notar los cambios de letra* $\boldsymbol{a} \rightarrow \mathbf{a}$, $\boldsymbol{y} \rightarrow \mathbf{y}$):

$$\mathbf{y} = (1, y_1, ..., y_D)^t, \;\; \mathbf{a}_c = (a_{c0}, a_{c1}, \cdots, a_{cD})^t$$

$$g_c(\boldsymbol{y}) = \mathbf{a}_c^t \mathbf{y}$$

Regla de clasificación:

$$\hat{c} = G(\boldsymbol{y}) \equiv \underset{1 \leq c \leq C}{\text{argmax}} \;\; \mathbf{a}_c^t \mathbf{y}$$

Funciones Discriminantes Lineales (FDL)

Un clasificador es *lineal* si sus Funcions Discriminantes son *funciones lineales* de las componentes de los vectores de E. Sea $\boldsymbol{y} \in E \equiv \mathbb{R}^D$ la representación de un objeto cualquiera.

$$g_c(\boldsymbol{y}) \;=\; \sum_{j=1}^{D} a_{cj} \cdot y_j + a_{c0} \;=\; \boldsymbol{a}_c^t \boldsymbol{y} + a_{c0}, \;\; 1 \leq c \leq C$$

Notación homogénea (*notar los cambios de letra* $\boldsymbol{a} \rightarrow \mathbf{a}$, $\boldsymbol{y} \rightarrow \mathbf{y}$):

$$\mathbf{y} = (1, y_1, ..., y_D)^t, \;\; \mathbf{a}_c = (a_{c0}, a_{c1}, \cdots, a_{cD})^t$$

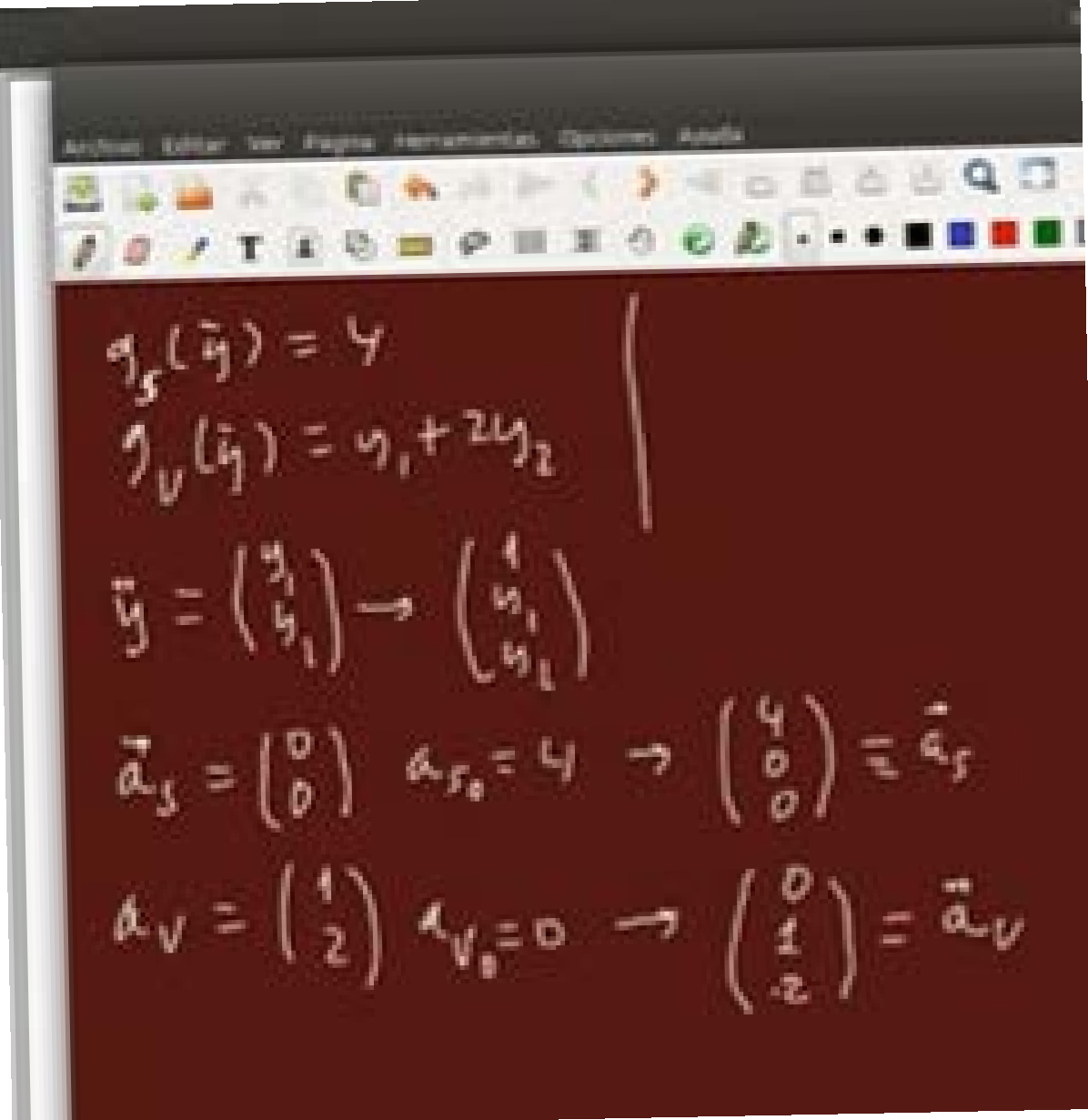
$$g_c(\boldsymbol{y}) = \mathbf{a}_c^t \mathbf{y}$$

Regla de clasificación:

$$\hat{c} = G(\boldsymbol{y}) \equiv \underset{1 \leq c \leq C}{\text{argmax}} \;\; \mathbf{a}_c^t \mathbf{y}$$

Frontera de decisión entre cada par de clases i, j: $\mathbf{a}_i^t \mathbf{y} = \mathbf{a}_j^t \mathbf{y}$

Fronteras lineales o *hiperplanos* de dimensión D (rectas si $D = 2$).


Archivo Editar Ver Página Herramientas Opciones Ayuda
$g_s(\vec{y}) = 4$
$g_v(\vec{y}) = y_1 + 2y_2$
$\vec{y} = \begin{pmatrix} y \\ y_1 \end{pmatrix} \rightarrow \begin{pmatrix} 1 \\ y_1 \\ y_2 \end{pmatrix}$
$\vec{a}_s = \begin{pmatrix} 0 \\ 0 \end{pmatrix}$ $a_{s_0} = 4 \rightarrow \begin{pmatrix} 4 \\ 0 \\ 0 \end{pmatrix} = \vec{a}_s$
$a_v = \begin{pmatrix} 1 \\ 2 \end{pmatrix}$ $a_{v_0} = 0 \rightarrow \begin{pmatrix} 0 \\ 1 \\ 2 \end{pmatrix} = \vec{a}_v$

Índice

Aprendizaje de FDLs [1]

Dadas N muestras de aprendizaje $(\mathbf{y}_1, c_1), \ldots, (\mathbf{y}_N, c_N)$, encontrar C vectores de pesos $\mathbf{a}_j$, $1 \leq j \leq C$, C que clasifiquen (lo más) correctamente (posible) las muestras de aprendizaje dadas; es decir:

$$\mathbf{a}_{c_1}^t \mathbf{y}_1 > \mathbf{a}_j^t \mathbf{y}_1, \ \forall j \neq c_1$$

$$\mathbf{a}_{c_2}^t \mathbf{y}_2 > \mathbf{a}_j^t \mathbf{y}_2, \ \forall j \neq c_2$$

$$\ldots$$

$$\mathbf{a}_{c_N}^t \mathbf{y}_N > \mathbf{a}_j^t \mathbf{y}_N, \ \forall j \neq c_N$$

Una solución: Ajustar iterativamente unos *pesos iniciales* mediante el *Algoritmo Perceptrón*, introducido en 1957 por Frank Rosenblatt.

Los clasificadores basados en el Algoritmo Perceptrón pueden considerarse como los ejemplos más sencillos de *redes neuronales*.

Problemas de convergencia con clases no *linealmente separables*. Soluciones: “margen” y/o algoritmo “Pocket-Perceptron”.

Aprendizaje de FDLs: Algoritmo Perceptrón

// Sean: $\mathbf{a}_j,\ 1 \leq j \leq C,\ \ C$ vectores de pesos iniciales;
// $(\mathbf{y}_1, c_1), \ldots, (\mathbf{y}_N, c_N),\ \ N$ muestras de aprendizaje;
// $\alpha \in \mathbb{R}^{>0},$ "factor de aprendizaje";
// $b \in \mathbb{R},$ "margen" (para ajustar la convergencia).

do $\{$
  $m = 0$ // número de muestras bien clasificadas
  for $(n = 1;\ n \leq N;\ n$++$)$ $\{$
    $i = c_n$**;** $g = \mathbf{a}_i^t\, \mathbf{y}_n$ **;** error=*false*
    for $(j = 1;\ j \leq C;\ j$++$)$ ***if*** $(j \neq i)$
      $\{$***if*** $(\mathbf{a}_j^t\, \mathbf{y}_n + b > g)$ $\{\mathbf{a}_j = \mathbf{a}_j - \alpha\, \mathbf{y}_n$ **;** error=*true*$\}\}$
    if (error) $\mathbf{a}_i = \mathbf{a}_i + \alpha\, \mathbf{y}_n$ **;** **else** $m = m + 1$
  $\}$
$\}$ ***while*** $(m < N)$

En caso de error se "corrigen" tanto los vectores de pesos de todas las clases que causan el error ($j:\ \mathbf{a}_j^t\, \mathbf{y}_n + b > \mathbf{a}_i^t\, \mathbf{y}_n$), como el de la clase correcta (i).

Algoritmo Perceptrón: Ejercicio (para hacer en clase)

En un problema de clasificación en 2 clases, para objetos representados mediante vectores de características bidimensionales, se tienen dos muestras de entrenamiento: $\boldsymbol{y}_1 = (0,0)^t$, $\boldsymbol{y}_2 = (1,1)^t$ de clases $c_1 = 1$, $c_2 = 2$, respectivamente.

Mostrar una traza de ejecución del algoritmo Perceptrón, con *vectores de pesos iniciales* nulos, *factor de aprendizaje* $\alpha = 1$ y *margen* $b = 0{,}1$. La traza debe incluir las sucesivas actualizaciones de los vectores de pesos de las clases.

Algoritmo Perceptrón: Ejercicio (para hacer en clase)

En un problema de clasificación en 2 clases, para objetos representados mediante vectores de características bidimensionales, se tienen dos muestras de entrenamiento: $\boldsymbol{y}_1 = (0,0)^t$, $\boldsymbol{y}_2 = (1,1)^t$ de clases $c_1 = 1$, $c_2 = 2$, respectivamente.

Mostrar una traza de ejecución del algoritmo Perceptrón, con *vectores de pesos iniciales* nulos, *factor de aprendizaje* $\alpha = 1$ y *margen* $b = 0{,}1$. La traza debe incluir las sucesivas actualizaciones de los vectores de pesos de las clases.

Solución resumida (solo la secuencia de pesos):

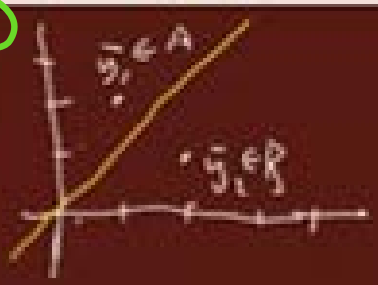
El algoritmo realiza 3 iteraciones del bucle más externo, produciendo la secuencia de vectores de pesos (en notación homogénea):

$$\mathbf{a}_1: \quad \begin{pmatrix}0\\0\\0\end{pmatrix} \rightarrow \begin{pmatrix}1\\0\\0\end{pmatrix}, \quad \begin{pmatrix}0\\-1\\-1\end{pmatrix} \rightarrow \begin{pmatrix}1\\-1\\-1\end{pmatrix}$$

$$\mathbf{a}_2: \quad \begin{pmatrix}0\\0\\0\end{pmatrix} \rightarrow \begin{pmatrix}-1\\0\\0\end{pmatrix}, \quad \begin{pmatrix}0\\1\\1\end{pmatrix} \rightarrow \begin{pmatrix}-1\\1\\1\end{pmatrix}$$

$\vec{y}_1 = \begin{pmatrix} 1 \\ 2 \end{pmatrix}$ $\vec{y}_2 = \begin{pmatrix} 2 \\ 1 \end{pmatrix}$ $S = \{(\vec{y}_1, A), (\vec{y}_2, B)\}$ $\delta = 0.1$

$\vec{y}_1 = \begin{pmatrix} 1 \\ 1 \\ 2 \end{pmatrix}$ $\vec{y}_2 = \begin{pmatrix} 1 \\ 2 \\ 1 \end{pmatrix}$; $\vec{a}_A = \begin{pmatrix} 0 \\ 0 \\ 0 \end{pmatrix}$ $\vec{a}_B = \begin{pmatrix} 0 \\ 0 \\ 0 \end{pmatrix}$ $\alpha = 1$



S: $\vec{y}_1 = \begin{pmatrix} 1 \\ 1 \\ 2 \end{pmatrix}$ $\vec{y}_2 = \begin{pmatrix} 1 \\ 2 \\ 1 \end{pmatrix}$ $\vec{y}_1 = \begin{pmatrix} 1 \\ 1 \\ 2 \end{pmatrix}$ $\vec{y}_2$

g_A	0	5	1	-1
g_B	0	-5	-1	1
Conf.	$0+0.1>0$	$5+1.1>-5$	$-1+0.1<1$	$-1+0.1<1$

$\vec{a}_A \begin{pmatrix} 0 \\ 1 \\ 0 \end{pmatrix} + \vec{y}_1 = \begin{pmatrix} 1 \\ 1 \\ 2 \end{pmatrix} - \vec{y}_2 = \begin{pmatrix} 0 \\ -1 \\ 1 \end{pmatrix} \rightarrow \begin{pmatrix} 0 \\ -1 \\ 1 \end{pmatrix} \rightarrow \begin{pmatrix} 0 \\ -1 \\ 1 \end{pmatrix}$ $y_1 - y_1 = 0$

$y_2 = y_1$

$\vec{a}_B \begin{pmatrix} 0 \\ 1 \\ 0 \end{pmatrix} - \vec{y}_1 = \begin{pmatrix} -1 \\ -1 \\ -2 \end{pmatrix} + \vec{y}_1 = \begin{pmatrix} 0 \\ 1 \\ -1 \end{pmatrix} \rightarrow \begin{pmatrix} 0 \\ 1 \\ -1 \end{pmatrix} \rightarrow \begin{pmatrix} 0 \\ 1 \\ -1 \end{pmatrix}$

$g_A(\vec{y}) = \vec{a}_A^t \vec{y}$
$g_B(\vec{y}) = \vec{a}_B^t \vec{y}$
$\Big\} \; \vec{a}_A^t \vec{y} = \vec{a}_B^t \vec{y}$: $(0, -1, 1)\begin{pmatrix} 1 \\ y_1 \\ y_2 \end{pmatrix} = (0, 1, -1)\begin{pmatrix} 1 \\ y_1 \\ y_2 \end{pmatrix} \Rightarrow -y_1 + y_2 = y_1 - y_2$

Algoritmo Perceptrón para *dos* clases [1]

Regla de clasificación para $C = 2$, con pesos $\mathbf{a}_1,\ \mathbf{a}_2$:

$$\hat{c} = G(\mathbf{y}) \ = \ \begin{cases} 1 & si \quad \mathbf{a}_1{}^t\mathbf{y} > \mathbf{a}_2{}^t\mathbf{y} \\ 2 & sino \ \ (\mathbf{a}_1{}^t\mathbf{y} \leq \mathbf{a}_2{}^t\mathbf{y}) \end{cases}$$

Este clasificador, su frontera de decisión, el criterio de éxito del aprendizaje y las ecuaciones de actualización de pesos se simplifican notablemente etiquetando las clases $\{1, 2\}$ como $\{+1, -1\}$ y usando un único vector de pesos $\mathbf{a} = \mathbf{a}_1 - \mathbf{a}_2$:

Clasificador:	$G(\mathbf{y}) \ = \ \text{sgn}(\mathbf{a}^t\mathbf{y})$
Frontera de decisión:	$\mathbf{a}^t\mathbf{y} \ = \ 0$
Criterio de éxito:	$c_n\, \mathbf{a}^t\mathbf{y}_n \ > \ 0, \ \ n = 1, \ldots, N$
Aprendizaje:	$\mathbf{a} \ = \ \mathbf{a} + \alpha\,(c_n - G(\mathbf{y}_n))\,\mathbf{y}_n, \ \ n = 1, \ldots, N$

Convergencia y calidad de resultados del Algoritmo Perceptrón

Tres parámetros controlan la convergencia y calidad de resultados:

α, b *(margen),* M *(máximo número de iteraciones)*

α determina el tamaño de las correcciones y por tanto la *velocidad de aprendizaje*. En general, $\alpha << \Rightarrow$ convergencia suave, pero con más iteraciones.

Convergencia y calidad de resultados del Algoritmo Perceptrón

Tres parámetros controlan la convergencia y calidad de resultados:

α, b *(margen),* M *(máximo número de iteraciones)*

α determina el tamaño de las correcciones y por tanto la *velocidad de aprendizaje*. En general, $\alpha << \Rightarrow$ convergencia suave, pero con más iteraciones.

- *Comportamiento con conjuntos de entrenamiento linealmente separables:*
  - Converge en un número finito de iteraciones $\forall \alpha > 0$
  - $b = 0$: Las fronteras de decisión pueden tener poca "holgura"; es decir, pueden resultar demasiado cerca de algunos datos
  - $b > 0$: si b es bastante grande, se obtienen fronteras de decisión "centradas" entre las regiones de decisión (típicamente mucho mejores que con $b = 0$)

Convergencia y calidad de resultados del Algoritmo Perceptrón

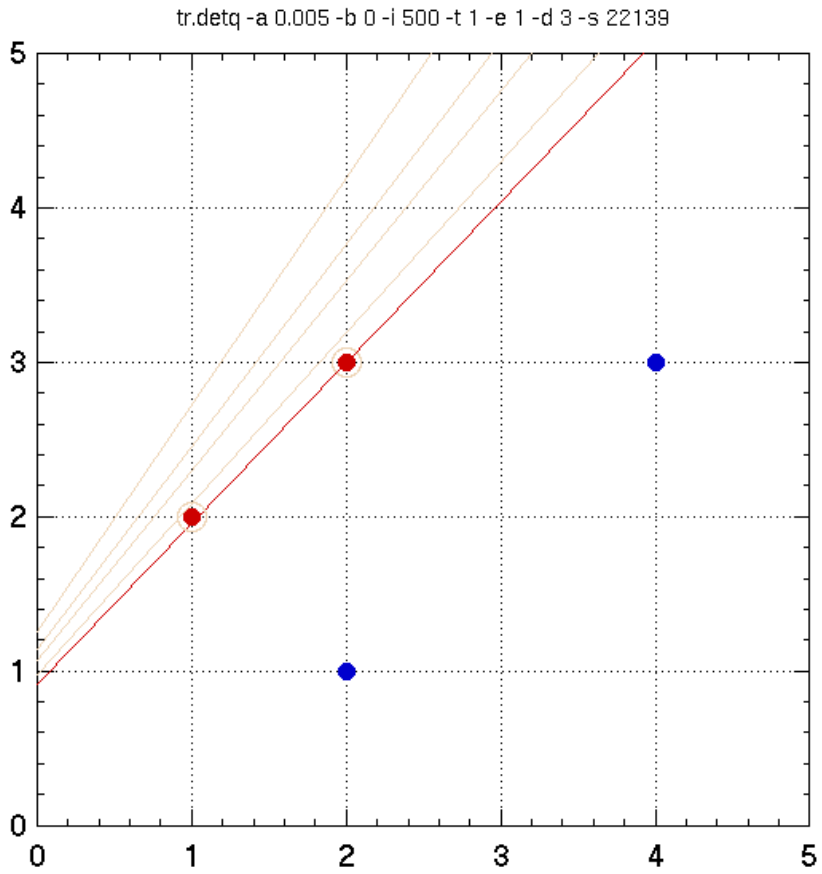
Tres parámetros controlan la convergencia y calidad de resultados:

α, b *(margen)*, M *(máximo número de iteraciones)*

α determina el tamaño de las correcciones y por tanto la *velocidad de aprendizaje*. En general, $\alpha << \Rightarrow$ convergencia suave, pero con más iteraciones.

- *Comportamiento con conjuntos de entrenamiento linealmente separables:*
  - Converge en un número finito de iteraciones $\forall \alpha > 0$
  - $b = 0$: Las fronteras de decisión pueden tener poca "holgura"; es decir, pueden resultar demasiado cerca de algunos datos
  - $b > 0$: si b es bastante grande, se obtienen fronteras de decisión "centradas" entre las regiones de decisión (típicamente mucho mejores que con $b = 0$)
- *Comportamiento con conjuntos de entrenamiento* ***no*** *linealmente separables:*
  - $b = 0$: ninguna garantía de convergencia ni de calidad del resultado
  - $b > 0$: no hay convergencia pero, con b y M suficientemente grandes, se obtienen buenas fronteras de decisión; generalmente (casi-)*óptimas*, en el sentido de minimizar el error de clasificación del conjunto de entrenamiento

Ilustración del funcionamiento del Perceptrón: Ejemplo simple separable con *margen nulo*



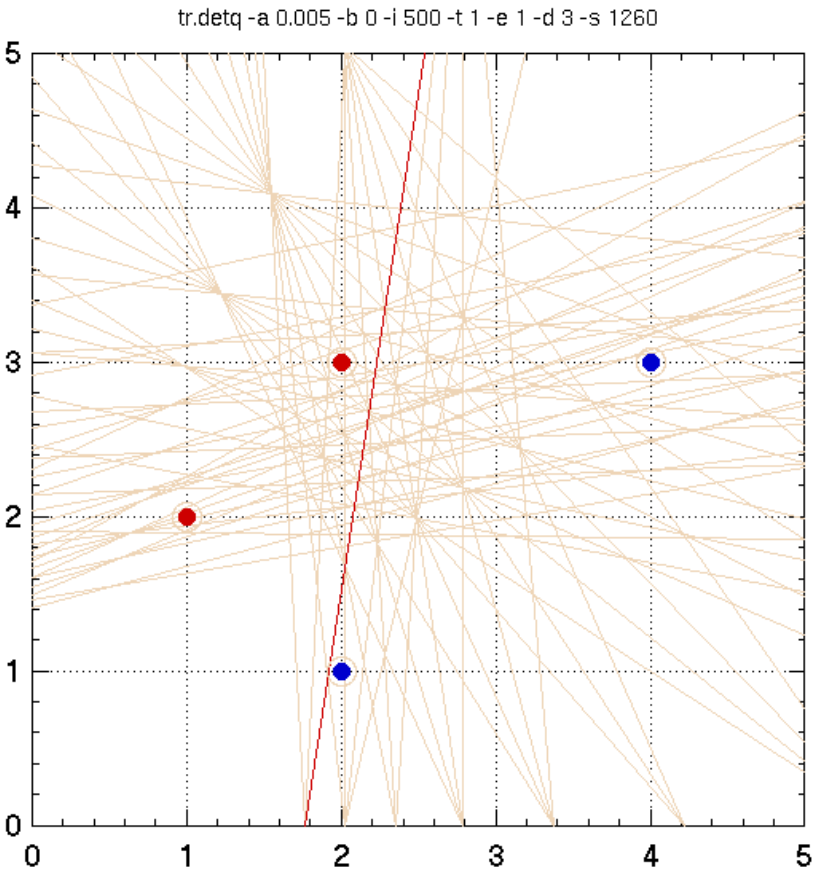
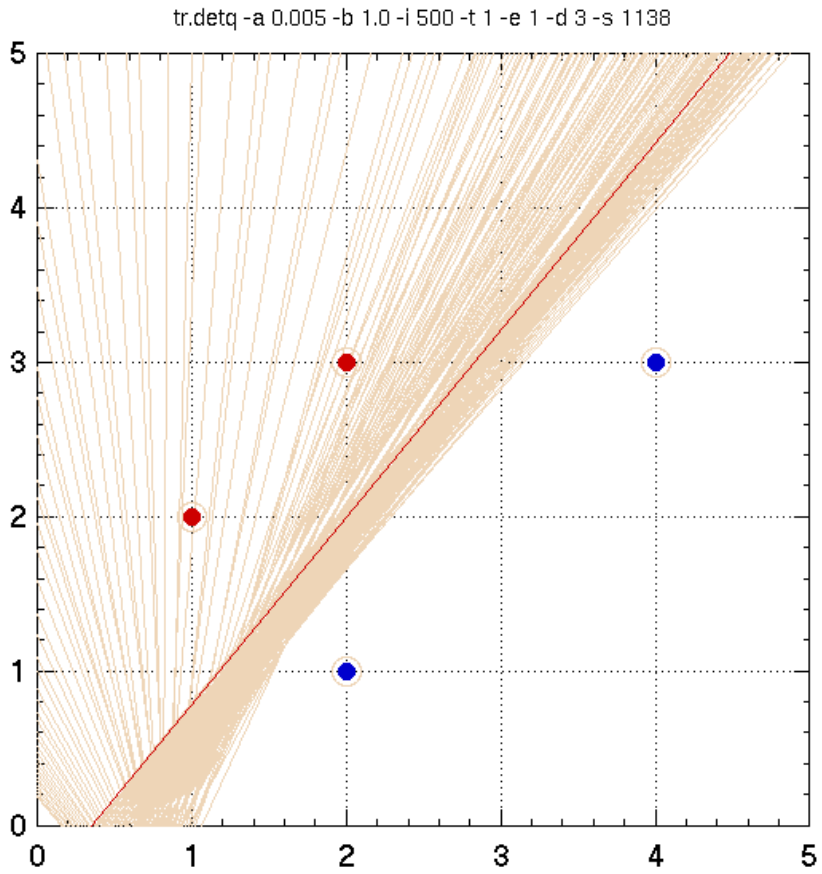


Ilustración del funcionamiento del Perceptrón: Ejemplo simple separable con *margen positivo*



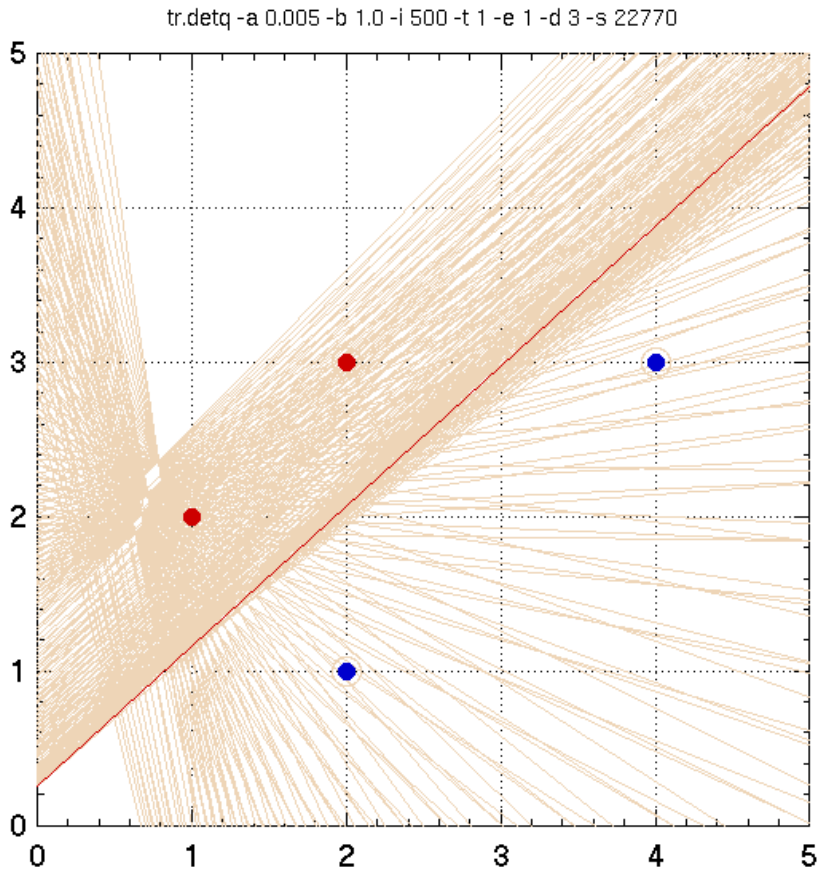
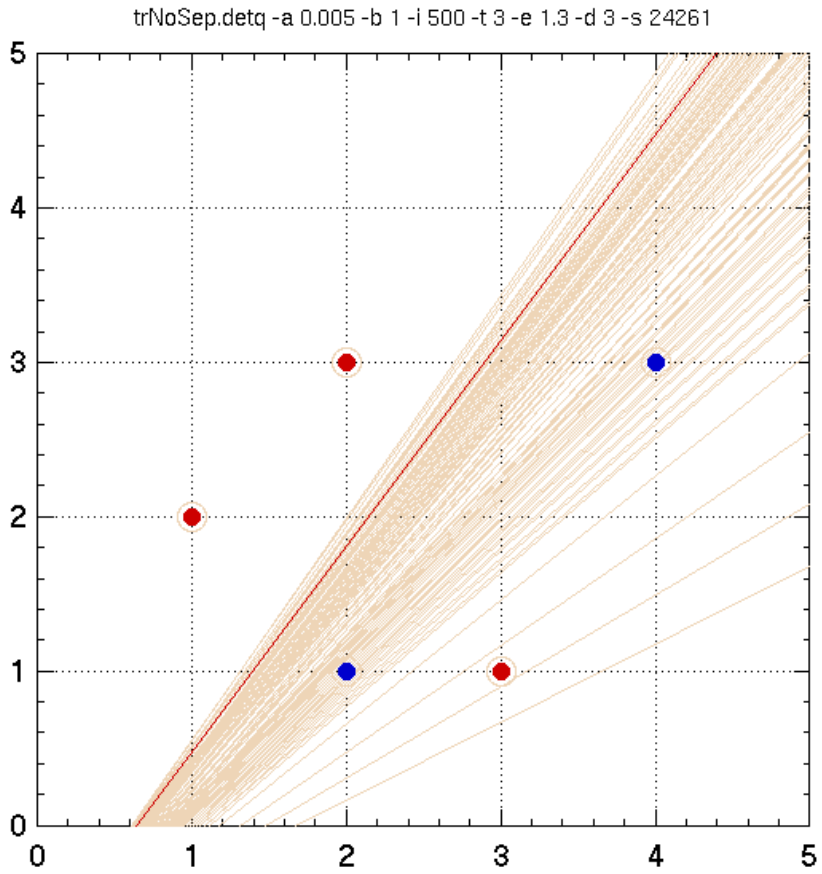


Ilustración del funcionamiento del Perceptrón: Ejemplo simple no separable con *margen positivo*



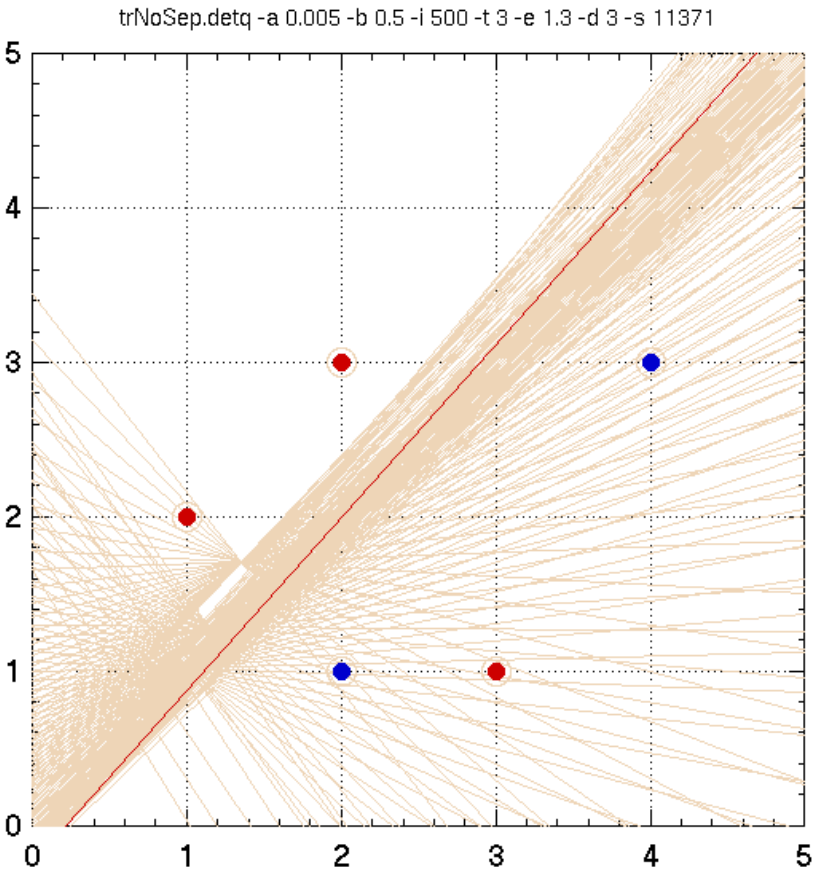
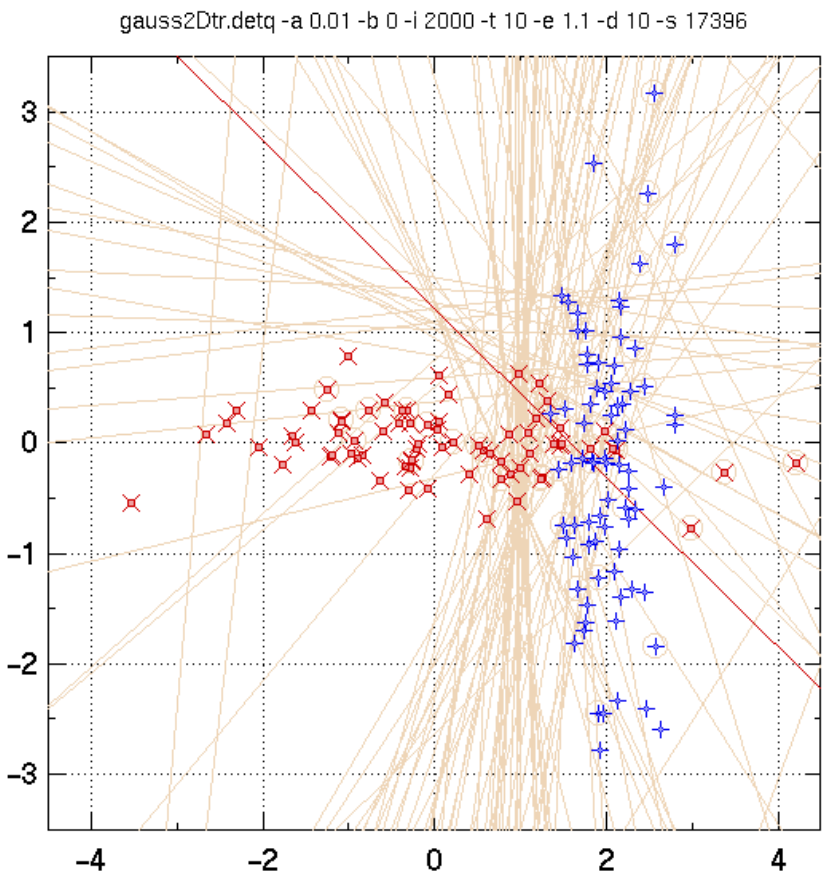


Ilustración del funcionamiento del Perceptrón: Gauss2D (no separable)

margen nulo



margen positivo

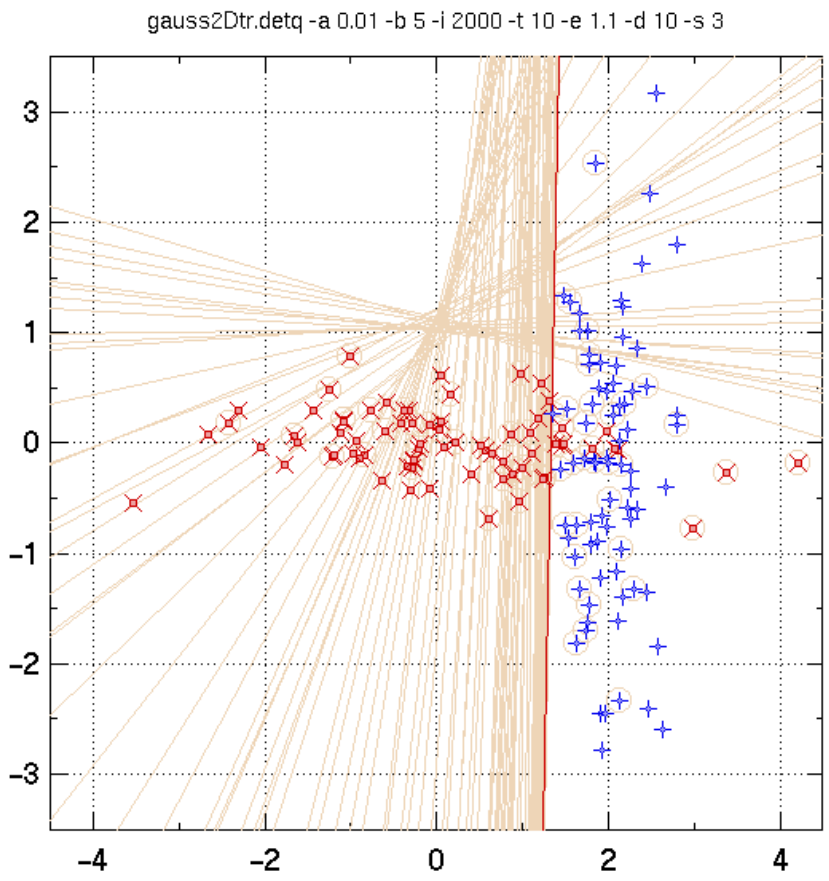
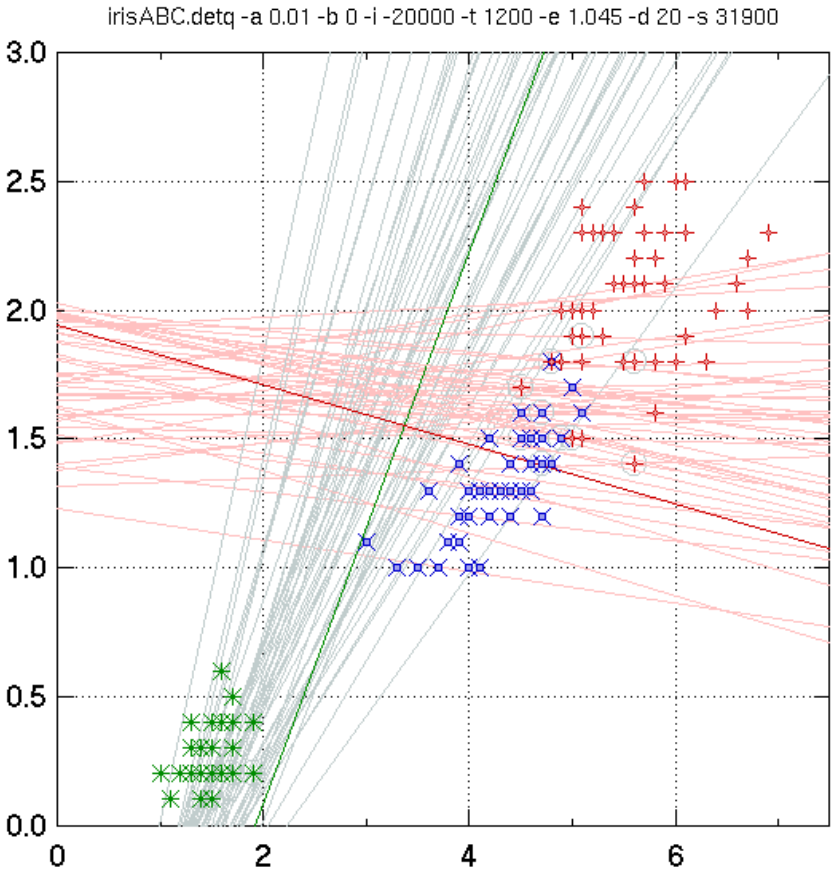
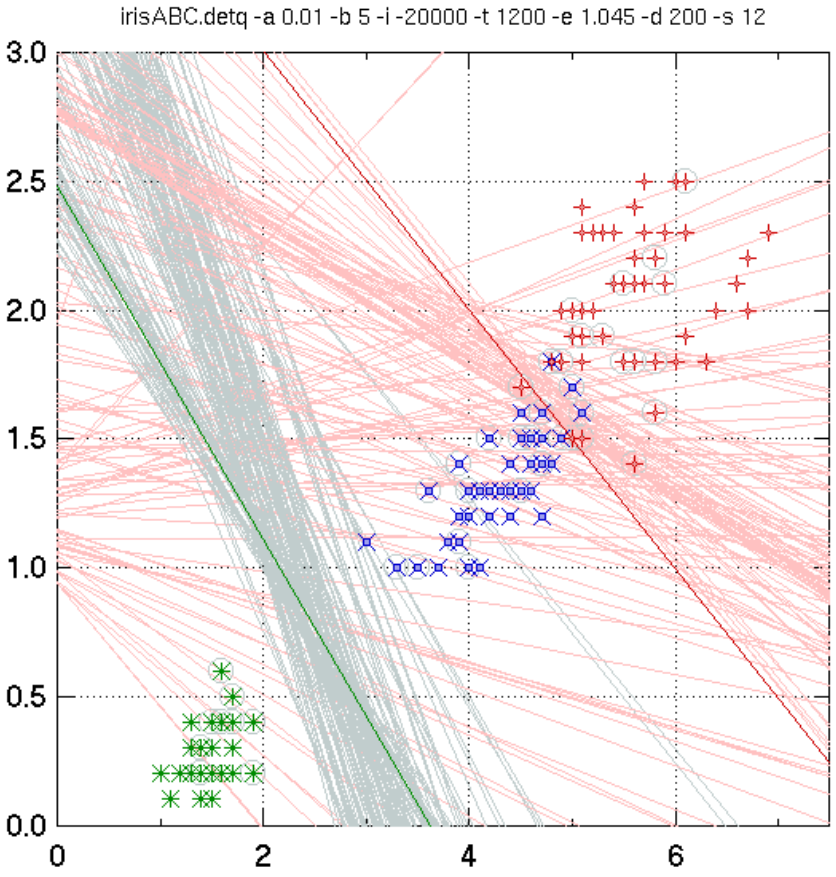


Ilustración del funcionamiento del Perceptrón: Iris2D (no separable)

margen nulo



margen positivo



Índice

Probabilidad empírica de error de decisión

Sea p la *verdadera* probabilidad de error de decisión de un sistema. Una estimación empírica ($\hat{p}$) de p puede obtenerse contabilizando el número de errores de decisión, N_e, que se producen en una *muestra de test* con N datos:

$$\hat{p} = \frac{N_e}{N}$$

Si $N >>$, podemos asumir que $\hat{p}$ se distribuye normalmente: $\hat{p} \sim \mathcal{N}\left(p,\ \frac{P(1-p)}{N}\right)$

Intervalo de confianza al $95\,\%$:

$$P(\hat{p} - \epsilon \leq p \leq \hat{p} + \epsilon) \ = \ 0.95; \qquad \epsilon \ = \ 1.96\sqrt{\frac{\hat{p}(1-\hat{p})}{N}}$$

Ejemplo: Se observan 50 decisiones erróneas de 1000 datos de test. Con una confianza del 95 % podemos afirmar que el verdadero riesgo de error es:

$$p \ = \ 0.05 \pm 1{,}96\sqrt{\frac{0{,}05 \cdot 0{,}95}{1000}} \ = \ 0{,}05 \pm 0{,}014 \ \ (5\,\% \pm 1{,}4\,\%)$$

Ejemplo: Si hay 5 errores en 100 datos de test el verdadero riesgo de error es:

$$p \ = \ \cdots \ = \ 0{,}05 \pm 0{,}043 \ \ (5\,\% \pm 4{,}3\,\%)$$

Métodos de partición de datos

Cuando un sistema se diseña mediante técnicas de *Aprendizaje Automático*, se necesitan datos no solo para estimar el error, sino para aprender los modelos de decisión. Dado un conjunto de datos etiquetados, este se puede dividir de diversas formas en subconjuntos de *entrenamiento* y de *test*:

- *Resustitución (Resubstitution):* Todos los datos disponibles se utilizan tanto para para entrenamiento como para test. Inconveniente: es *(muy) optimista*.
- *Partición (Hold Out):* Los datos se dividen en un subconjunto para entrenamiento y otro para test. Inconveniente: desaprovechamiento de datos.
- *Validación Cruzada en B bloques (B-fold Cross Validation):* Los datos se dividen aleatoriamente en B bloques. Cada bloque se utiliza como test para un sistema entrenado con el resto de bloques. Inconvenientes: Reduce el número de datos de entrenamiento (sobre todo cuando B es pequeño) y el coste computacional se incrementa con B.
- *Exclusión individual (Leaving One Out):* Cada dato individual se utiliza como dato único de test de un sistema entrenado con los $n-1$ datos restantes. Equivale a Validación Cruzada en n bloques. Inconveniente: el coste computacional.

Índice

Bibliografía

[1] R.O. Duda, D.G. Stork, P.E. Hart. Pattern Classification. Wiley, 2001.