

CLASIFICACIÓN DE OBSERVACIONES SEPARABILIDAD DE CONJUNTOS

Trabajo de Grado

AUTORA : ANA LEVATO

DIRECTORA: Lic. MARÍA TERESA GUARDARUCCI

Universidad Nacional de La Plata - Facultad de Ciencias Exactas

1997

TES 97/17 DIF-01988 SALA	 <p>UNIVERSIDAD NACIONAL DE LA PLATA FACULTAD DE INFORMATICA Biblioteca 50 y 120 La Plata catalogo.info.unlp.edu.ar biblioteca@info.unlp.edu.ar</p>  <p>DIF-01988</p>
---	---

ÍNDICE

Agradecimientos	3
Introducción	4
Método Multisurface de Separación de Patrones	5
Clasificación de Observaciones	6
Problema	6
Procedimientos de Clasificación	8
Reconocimiento de Patrones Vía Programación Lineal	15
Conjuntos de patrones separables linealmente	15
Conjuntos de patrones linealmente inseparables	23
Un algoritmo polinomial para el problema de separación	29
Algoritmo	32
Procedimiento de separación	33
Degeneración	34
Extensión al caso de más de dos conjuntos	35
Etapas de Implementación	36
Experiencias Computacionales	45
Uso del Software	56
Apéndice	57
Bibliografía	58

Agradecimientos

Este trabajo quiero dedicárselo especialmente a mi familia, que a pesar de estar lejos físicamente, estuvieron siempre muy cerca mío apoyándome y alentándome durante toda mi carrera.

Quiero agradecerle a Marité, mi directora, que supo tenerme paciencia, me dedicó todo el tiempo que necesité para desarrollar este trabajo. Juntas probamos el programa, para que no fallara, cada vez que nos reuníamos yo iba con miedo, porque Marité encontraba algún caso que no habíamos considerado antes, y había que corregir el programa, volverlo a probar y tener la ilusión de que ese era el último cambio. Pero esa exigencia de ella hizo que el trabajo saliera lo mejor posible.

También quiero decirle muchas gracias a una persona muy especial, a Fer, mi amor, que se sentaba conmigo esas tardes en que no me salían las cosas y pensaba que nunca iba a terminar con el programa, lo revisábamos y como él es un conocedor a fondo del lenguaje Pascal, encontraba el error, o en algunos casos, se daba cuenta de algunas distracciones mías. Siempre estuvo al lado mío, a pesar de sus obligaciones.

Personalmente estoy muy contenta con el trabajo, y realizarlo me hizo ver cuantos problemas de la vida diaria podemos resolver aplicando los conocimientos que se brindan en Investigación Operativa.

Ana Levato

Introducción

El problema de clasificación o separación de patrones ocurre cuando después de realizar una serie de medidas en un individuo u objeto, se lo desea asignar a alguna categoría, basando la decisión en las mediciones efectuadas.

Este problema ocurre en muchísimas instancias diarias, tales como :

- Un persona es sometida a un test que involucra una serie de preguntas y de acuerdo a sus respuestas debe ser aceptada o no para determinada tarea.
- Restos fósiles son hallados, diferentes mediciones son efectuadas sobre ellos, y de acuerdo a los datos obtenidos, debemos decidir si pertenecen a determinada era o raza.
- Un análisis clínico es efectuado, y de acuerdo al nivel de los parámetros evaluados, se debe decidir si el paciente padece o no determinada enfermedad.
- Se realiza una evaluación sobre un objeto para determinar a que clase pertenece, ya sea un árbol, casa, auto, etc.

En algunos casos se puede suponer que hay un número finito de poblaciones a las que el resultado puede ser asignado y que cada población está caracterizada por una distribución de probabilidad de las medidas. En esa situación, pensamos en los datos obtenidos como una observación al azar proveniente de la población, y la asignación a uno de los posibles grupos, se hace basada en técnicas estadísticas.

En otras situaciones, uno ya tiene el antecedente de las mediciones efectuadas a distintos elementos y el grupo al cual pertenecen, y ante la presencia de una nueva medición, quiere tomar la decisión de a que población asignarla. En el caso particular de dos poblaciones A y B los datos se van separando en forma sucesiva, mediante la construcción de funciones lineales, hasta conseguir una función discriminante f , que cumple que para los datos provenientes de un conjunto A , $f(A) > 0$ y para los datos provenientes del conjunto B , $f(B) < 0$. Un enfoque para resolver esta situación puede basarse en técnicas de Programación Lineal.

El presente trabajo hace una breve presentación del enfoque estadístico y un detallado análisis del trabajo "Patterns Recognition Via Linear Programming : Theory and Applications to Medical Diagnosis", O.L. Mangasarian, R. Setions, W.H. Wolberg.

Todas las aseveraciones son demostradas y se presenta un programa implementando la propuesta y el algoritmo de decisión.

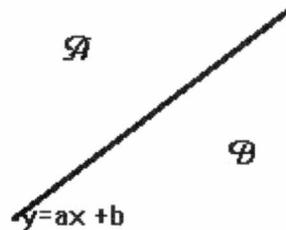
Distintos ejemplos son testeados y analizados, y para el caso particular de degeneración (en una etapa intermedia ningún dato puede ser separado), una propuesta es sugerida y testada.

MÉTODO MULTISURFACE DE SEPARACIÓN DE PATRONES

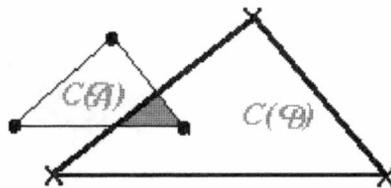
Un patrón es un punto en un espacio n -dimensional \mathbb{R}^n . Un conjunto \mathcal{A} de patrones es un conjunto de m puntos en \mathbb{R}^n y será representado por una matriz A de $m \times n$, donde cada fila A_i representa un patrón.

Problema: Dados dos conjuntos disjuntos de patrones en \mathbb{R}^n , representados por las matrices A de $m \times n$ y B de $k \times n$, conseguir un criterio para separar los elementos de cada conjunto.

Cuando las cápsulas convexas de los dos conjuntos no se intersectan, se puede encontrar un plano separador tal que ningún punto de los conjuntos se encontrará sobre el plano. En este caso decimos que los conjuntos son linealmente separables.



A menudo las cápsulas convexas $C(\mathcal{A})$ y $C(\mathcal{B})$ de los conjuntos \mathcal{A} y \mathcal{B} , representados por las matrices A y B , se intersectan y no pueden ser separados por un plano.



Cuando los datos son variables aleatorias, existe un enfoque estadístico del tema, el que resumiremos a continuación.

CLASIFICACIÓN DE OBSERVACIONES

Problema:

Se realizan mediciones de distintos parámetros sobre un mismo individuo, y en base a esas medidas tratamos de clasificarlo dentro de alguna población.

En general asumimos que hay un número finito de poblaciones de donde puede provenir el individuo y cada población se caracteriza por una distribución de probabilidades.

A un individuo lo consideramos como una observación aleatoria de la población. El problema que queremos resolver es: Dado un individuo con determinadas medidas, a qué población pertenece?

Para construir un buen procedimiento de clasificación debemos tratar de minimizar las clasificaciones erróneas.

Consideremos el caso de dos poblaciones π_1 y π_2 en el que realizamos p medidas sobre un individuo. Clasificar una observación dependerá del vector de medidas $x^t=(x_1, \dots, x_p)$ de la observación.

Una observación es un punto en un espacio p -dimensional. Dividimos al espacio en dos regiones $R_1(\pi_1)$ y $R_2(\pi_2)$ (donde $R_i(\pi_i)$ es la región de los individuos que asignaremos como pertenecientes a la población i).

Los errores que pueden ocurrir son: un individuo de π_1 clasificarlo como de π_2 y viceversa.

Consideremos que cada uno de estos errores tiene un costo asociado $C(2/1)$ ó $C(1/2)$, ($C(2/1)$ es el costo de asignarlo a R_2 cuando en realidad proviene de π_1).

Si tuviésemos una tabla de costos, tendríamos:

	π_1	π_2
π_1	0	$C(2/1)$
π_2	$C(1/2)$	0

Queremos asignar tratando de minimizar el costo de clasificación errónea esperado.

Caso 1) Conocemos a priori las probabilidades de pertenecer a π_1 y π_2 , que son q_1 y q_2 respectivamente. Tenemos la distribución de probabilidades de la población $\pi_1 \rightarrow p_1(x)$ y $\pi_2 \rightarrow p_2(x)$. R_1 es la región que clasifica a los individuos de π_1 y R_2 a los de π_2 . La probabilidad de clasificar correctamente a los de π_1 (π_2 respectivamente) es:

$$\pi_1 \rightarrow P(1/1, R) = \int_{R_1} p_1(x) dx \quad \pi_2 \rightarrow P(2/2, R) = \int_{R_2} p_2(x) dx$$

donde $P(1/1, R)$ es la probabilidad de decir que es de π_1 dado que realmente lo es, es decir la probabilidad de que los valores observados caigan en R_1 dado que son de π_1 . La probabilidad de clasificar mal es:

$$P(2/1, R) = \int_{R_2} p_1(x) dx \quad P(1/2, R) = \int_{R_1} p_2(x) dx$$

Como la probabilidad de que una observación provenga de π_1 es q_1 , la probabilidad de clasificarla correctamente (que sea de π_1 y caiga en R_1) es $q_1 \cdot P(1/1, R)$ y la de clasificarla mal es $q_1 \cdot P(2/1, R)$. Lo mismo pasa con π_2 . Luego, lo que queremos minimizar es el costo esperado:

$$E(\text{costo}) = C(2/1) P(2/1, R) q_1 + C(1/2) P(1/2, R) q_2$$

Caso 2) No conocemos a priori las probabilidades de pertenecer a π_1 y π_2 . El costo de clasificar mal lo perteneciente a π_1 será $r(1, R) = C(2/1) P(2/1, R)$; para π_2 tenemos $r(2, R) = C(1/2) P(1/2, R)$.

Consideramos un procedimiento $R = R_1 \cup R_2$. Decimos que R es al menos tan bueno como $R^* = R_1^* \cup R_2^*$ si: $r(1, R) \leq r(1, R^*)$ y $r(2, R) \leq r(2, R^*)$, es decir, si el costo de clasificar mal con el procedimiento R^* es menor o igual que el de clasificar mal usando el procedimiento R . R y R^* son procedimientos que dados q_1 y q_2 minimizan la función de costo E . Este tipo de procedimientos se conoce como procedimientos de Bayes.

R es mejor que R^* si al menos una de las desigualdades es estricta. R es admisible si no hay un procedimiento mejor que R .

Un procedimiento es minimax si la pérdida máxima esperada, $r(i, R)$ es un mínimo. Esto quiere decir:

$$\min [\max (r(i, R))]$$

lo que equivale a resolver:

$$\begin{array}{ll} \min & K \\ \text{s. a.} & r(1, R) \leq K \\ & r(2, R) \leq K \\ & r(i, R) \leq K \end{array}$$

Procedimientos de Clasificación

Caso 1) **Dos poblaciones con probabilidades conocidas a priori**

Debemos elegir regiones R_1 y R_2 de R tal que minimicen la función de costo. Dada una observación x , la probabilidad condicional de provenir de π_1 es:

$$\frac{P(x \in \pi_1 \mid x = (x_1, \dots, x_n))}{P(x = (x_1, \dots, x_n))} = \frac{q_1 \cdot p_1(x)}{q_1 \cdot p_1(x) + q_2 \cdot p_2(x)}$$

Si $C(1/2) = C(2/1) = 1$ entonces la función de costo será:

$$q_1 \int_{R_2} p_1(x) dx + q_2 \int_{R_1} p_2(x) dx$$

y esta expresión es la que queremos minimizar. Notar que al ser $C(1/2) = C(2/1) = 1$, esta expresión coincide con la del costo esperado de clasificar mal y veremos que para una observación x , podemos minimizar esa probabilidad, asignándola a la población que tiene la mayor probabilidad condicional, es decir, si

$$\frac{q_1 \cdot p_1(x)}{q_1 \cdot p_1(x) + q_2 \cdot p_2(x)} \geq \frac{q_2 \cdot p_2(x)}{q_1 \cdot p_1(x) + q_2 \cdot p_2(x)}$$

entonces decimos que x proviene de π_1 , caso contrario decimos que x proviene de π_2 .

Regla:

$$R_1 : q_1 \cdot p_1(x) \geq q_2 \cdot p_2(x)$$

$$R_2 : q_1 \cdot p_1(x) < q_2 \cdot p_2(x)$$

Si $q_1 \cdot p_1(x) + q_2 \cdot p_2(x) = 0$ entonces x puede clasificarse como de π_1 o π_2 . Notar que si $R_1 \cap R_2 = \emptyset$ entonces dado $x \in \pi_1$ $P_2(x) = 0$ y si pertenece π_2 $P_1(x) = 0$ y la separación sería inmediata

Para cualquier procedimiento $R^* = (R_1^*, R_2^*)$, la probabilidad de clasificar mal es:

$$q_1 \int_{R_2^*} p_1(x) dx + q_2 \int_{R_1^*} p_2(x) dx = \int_{R_2^*} [q_1 \cdot p_1(x) - q_2 \cdot p_2(x)] dx + q_2 \int_{R_1^* \cup R_2^*} p_2(x) dx$$

El segundo término de la parte derecha es un número dado. Así el primer término es minimizado si R_2^* incluye los puntos x tal que $q_1 \cdot p_1(x) - q_2 \cdot p_2(x) < 0$ y excluye los puntos para los cuales $q_1 \cdot p_1(x) - q_2 \cdot p_2(x) > 0$.

Lo anterior queda expresado en el siguiente teorema:

Teorema: Sean q_1 y q_2 probabilidades conocidas de una observación de una población π_1 con densidad $p_1(x)$ y de una población π_2 con densidad $p_2(x)$ respectivamente, y sea $C(2/1)$ el costo de clasificar una observación de π_1 como de π_2 y $C(1/2)$ el costo de clasificar una observación de π_2 como de π_1 . Si las regiones de clasificación R_1 y R_2 se definen, como el conjunto de las x que satisfacen

$$R_1 : \frac{p_1(x)}{p_2(x)} \geq \frac{C(1/2) q_2}{C(2/1) q_1}$$

$$R_2 : \frac{p_1(x)}{p_2(x)} < \frac{C(1/2) q_2}{C(2/1) q_1}$$

entonces la función de costo tiene un mínimo respecto de R .

Además si

$$\Pr \left\{ \frac{p_1(x)}{p_2(x)} = \frac{q_2 C(1/2)}{q_1 C(2/1)} \mid \pi_1 \right\} = 0$$

entonces el procedimiento es único excepto para los conjuntos con probabilidad cero. ■

Caso2) Dos poblaciones con probabilidades no conocidas a priori

Un procedimiento de Bayes es aquel que minimiza el error E . Veremos que todo procedimiento de Bayes es admisible.

Sea $R = (R_1, R_2)$ un procedimiento de Bayes para q_1 y q_2 . La pregunta es: Hay un procedimiento $R^* = (R_1^*, R_2^*)$ tal que $P(1/2, R^*) \leq P(1/2, R)$ y $P(2/1, R^*) \leq P(2/1, R)$ con al menos una desigualdad estricta?

Como R es un procedimiento de Bayes,

$$q_1 \cdot P(2/1, R) + q_2 \cdot P(1/2, R) \leq q_1 \cdot P(2/1, R^*) + q_2 \cdot P(1/2, R^*)$$

Esto es equivalente a:

$$q_1 \bullet [P(2/1, R) - P(2/1, R^*)] \leq q_2 \bullet [P(1/2, R^*) - P(1/2, R)]$$

Si $q_1 > 0$ y $P(1/2, R^*) \leq P(1/2, R)$ entonces $q_2 \bullet [P(1/2, R^*) - P(1/2, R)] \leq 0$ y así $P(2/1, R) \leq P(2/1, R^*)$. Así R^* no es mejor que R y R es admisible.

Teorema: Si $\Pr \{ p_2(x) = 0 \mid \pi_1 \} = 0$ y $\Pr \{ p_1(x) = 0 \mid \pi_2 \} = 0$ entonces cada procedimiento de Bayes es admisible.

Una clase de procedimientos es completa si para cada procedimiento fuera de la clase hay uno dentro de la clase que es mejor.

Una clase minimal completa es una clase completa tal que ningún subconjunto propio es una clase completa.

Teorema: Si vale

$$\Pr \left\{ \frac{p_1(x)}{p_2(x)} = k \mid \pi_i \right\} = 0 \quad 0 \leq k \leq \infty$$

entonces la clase de los procedimientos de Bayes es minimal completa. ■

Caso 3) Dos poblaciones con distribución normal conocida

Sean dos poblaciones con distribución $N(\mu^{(1)}, \Sigma)$ y $N(\mu^{(2)}, \Sigma)$ donde $\mu^{(i)t} = (\mu_1^{(i)}, \dots, \mu_p^{(i)})$ es el vector de medias de la población i y Σ es la matriz de varianzas y covarianzas de cada población. La densidad i es:

$$p_i(x) = [1 / (2\pi)^{1/2p} |\Sigma|^{1/2}] \exp [-1/2 (x - \mu^{(i)})^t \Sigma^{-1} (x - \mu^{(i)})]$$

luego

$$\frac{p_1(x)}{p_2(x)} = \exp \{ -1/2 [(x - \mu^{(1)})^t \Sigma^{-1} (x - \mu^{(1)}) - (x - \mu^{(2)})^t \Sigma^{-1} (x - \mu^{(2)})] \}$$

Consideramos R_1 al conjunto de x para los cuales $p_1(x)/p_2(x) \geq k$. Como la función logaritmo es monótona creciente, la desigualdad puede escribirse en términos del logaritmo de $p_1(x)/p_2(x)$.

Teorema: Si π_i tiene densidad:

$$\pi_i(x) = [1 / (2\pi)^{1/2p} |\Sigma|^{1/2}] \exp [-1/2 (x - \mu^{(i)})^t \Sigma^{-1} (x - \mu^{(i)})]$$

las mejores regiones de clasificación están dadas por:

- Si q_1 y q_2 son conocidas a priori y $k \neq 1$ donde

$$k = \frac{q_2 C(1/2)}{q_1 C(2/1)}$$

$$R1 : x^t \Sigma^{-1} (\mu^{(1)} - \mu^{(2)}) - 1/2 (\mu^{(1)} + \mu^{(2)})^t \Sigma^{-1} (\mu^{(1)} - \mu^{(2)}) \geq \log k$$

$$R2 : x^t \Sigma^{-1} (\mu^{(1)} - \mu^{(2)}) - 1/2 (\mu^{(1)} + \mu^{(2)})^t \Sigma^{-1} (\mu^{(1)} - \mu^{(2)}) < \log k$$

- Si $k=1$ ($\log k = 0$) entonces

$$R1 : x^t \Sigma^{-1} (\mu^{(1)} - \mu^{(2)}) \geq 1/2 (\mu^{(1)} + \mu^{(2)})^t \Sigma^{-1} (\mu^{(1)} - \mu^{(2)})$$

$$R2 : x^t \Sigma^{-1} (\mu^{(1)} - \mu^{(2)}) < 1/2 (\mu^{(1)} + \mu^{(2)})^t \Sigma^{-1} (\mu^{(1)} - \mu^{(2)})$$

Notar que para determinar R1 y R2 debemos hallar $\Sigma^{-1} (\mu^{(1)} - \mu^{(2)}) = v$ ■

Cuando las poblaciones tienen distribución normal, la matriz Σ es simétrica definida positiva entonces tenemos distintas técnicas para resolver el sistema $\Sigma v = \mu^{(1)} - \mu^{(2)}$. Algunas de estas técnicas son el Método de Choleski, que es un método directo, o el Método de Gradientes Conjugados, que es un método iterativo, que convergen a la solución en un número finito de iteraciones.

Si no conocemos las probabilidades a priori, elegimos $\log k = c$. Para hallar la probabilidad de clasificar mal deberíamos conocer la distribución de U , donde dada una observación X ,

$$U = X^t \Sigma^{-1} (\mu^{(1)} - \mu^{(2)}) - 1/2 (\mu^{(1)} + \mu^{(2)})^t \Sigma^{-1} (\mu^{(1)} - \mu^{(2)})$$

Si $X \sim N(\mu^{(1)}, \Sigma)$ entonces U tiene distribución normal con media

$$E_1(U) = 1/2 (\mu^{(1)} - \mu^{(2)})^t \Sigma^{-1} (\mu^{(1)} - \mu^{(2)}) = 1/2 \alpha$$

y varianza

$$\text{Var}_1(U) = (\mu^{(1)} - \mu^{(2)})^t \Sigma^{-1} (\mu^{(1)} - \mu^{(2)}) = \alpha$$

por lo tanto $U \sim N(1/2 \alpha, \alpha)$ si $X \sim N(\mu^{(1)}, \Sigma)$

De manera análoga, si $X \sim N(\mu^{(2)}, \Sigma)$ entonces

$$E_2(U) = 1/2 (\mu^{(2)} - \mu^{(1)})^t \Sigma^{-1} (\mu^{(1)} - \mu^{(2)}) = -1/2\alpha$$

y varianza

$$\text{Var}_2(U) = (\mu^{(1)} - \mu^{(2)})^t \Sigma^{-1} (\mu^{(1)} - \mu^{(2)}) = \alpha$$

por lo tanto $U \sim N(-1/2\alpha, \alpha)$ si $X \sim N(\mu^{(2)}, \Sigma)$

Luego, las probabilidades de clasificar mal son:

$$P(2/1) = \int_{-\infty}^{(c - 1/2\alpha)/\sqrt{\alpha}} e^{-1/2 y^2} / (2\pi)^{1/2} dy$$

$$P(1/2) = \int_{(c + 1/2\alpha)/\alpha^{1/2}}^{\infty} e^{-1/2 y^2} / (2\pi)^{1/2} dy$$

Condición 1:

Para la solución minimax, elegimos c tal que: $C(1/2) P(1/2) = C(2/1) P(2/1)$

Lo anterior queda expresado en el siguiente teorema:

Teorema: Si π_i tiene densidad $\pi_i(x)$, las regiones minimax de clasificación están dadas por:

$$R1 : x^t \Sigma^{-1} (\mu^{(1)} - \mu^{(2)}) - 1/2 (\mu^{(1)} + \mu^{(2)})^t \Sigma^{-1} (\mu^{(1)} - \mu^{(2)}) \geq \log k$$

$$R2 : x^t \Sigma^{-1} (\mu^{(1)} - \mu^{(2)}) - 1/2 (\mu^{(1)} + \mu^{(2)})^t \Sigma^{-1} (\mu^{(1)} - \mu^{(2)}) < \log k$$

donde $c = \log k$ es elegido por la condición 1, con $C(i/j)$ los costos de clasificar mal.

Si los costos son iguales, entonces $c=0$ y la probabilidad de clasificar mal es :

$$\int_{(\alpha/2)^{1/2}}^{\infty} e^{-1/2 y^2} / (2\pi)^{1/2} dy$$

Si los costos no son iguales, c puede determinarse con el método de "ensayo y error".

■

Si llamamos $\Sigma^{-1} (\mu^{(1)} - \mu^{(2)}) = \delta$, ($\Sigma\delta = (\mu^{(1)} - \mu^{(2)})$), entonces R1: $x^t \delta - \alpha \geq \log k$, y se puede mostrar que $x^t \delta$ es la función lineal que maximiza:

$$\frac{[E_1(X^t d) - E_2(X^t d)]^2}{\text{Var}(X^t d)} \quad \forall d$$

esto es decir \max_d (varianza entre muestras) / (varianza en la muestra)

Considerando que lo anterior puede escribirse como:

$$\frac{d^t [(\mu^{(1)} - \mu^{(2)}) (\mu^{(1)} - \mu^{(2)})^t] d}{d^t \Sigma d}$$

Si queremos maximizar el numerador con respecto a d , dejando $d^t \Sigma d$ constante tenemos una maximización de una función cuadrática con restricciones cuadráticas.

$$\begin{aligned} \max \quad & d^t A d \\ \text{s. a.} \quad & d^t \Sigma d = \text{cte} \end{aligned}$$

donde $A = (\mu^{(1)} - \mu^{(2)}) (\mu^{(1)} - \mu^{(2)})^t$

Para que cumpla la condición de Kun Tucker, si λ es un multiplicador de Lagrange, d debe satisfacer:

$$\begin{aligned} 2Ad &= 2\lambda \Sigma d \\ (\mu^{(1)} - \mu^{(2)}) (\mu^{(1)} - \mu^{(2)})^t d &= \lambda \Sigma d \end{aligned}$$

Como $(\mu^{(1)} - \mu^{(2)})^t d$ es un escalar v , entonces podemos escribir la fórmula anterior como $(\mu^{(1)} - \mu^{(2)}) = (\lambda / v) \Sigma d$, así vemos que la solución es proporcional a δ donde $\delta = (\lambda / v) d = \Sigma^{-1} (\mu^{(1)} - \mu^{(2)})$ con lo cual $d^* = \delta (v / \lambda)$.

Así, si tenemos una muestra de tamaño n de la población π_1 o π_2 , usamos la media de la muestra y hacemos la clasificación como $N(\mu^{(1)}, (1/n)\Sigma)$ ó $N(\mu^{(2)}, (1/n)\Sigma)$.

Clasificación en una de dos poblaciones normales, cuando los parámetros son estimados

Tenemos una muestra $x_1^{(1)}, \dots, x_n^{(1)}$ de $N(\mu^{(1)}, \Sigma)$ y una muestra $x_1^{(2)}, \dots, x_m^{(2)}$ de $N(\mu^{(2)}, \Sigma)$ donde $\mu^{(i)}$ y Σ son desconocidos. Si queremos clasificar una observación X en π_1 o π_2 , estimamos

$$\hat{\mu}^{(1)} = \bar{x}^{(1)} = \left(\sum_{\alpha=1}^n x_{\alpha}^{(1)} \right) / n \quad \hat{\mu}^{(2)} = \bar{x}^{(2)} = \left(\sum_{\alpha=1}^m x_{\alpha}^{(2)} \right) / m$$

$$\hat{E} = (N+M-2)S = \sum_{\alpha=1}^n (x_{\alpha}^{(1)} - \bar{x}^{(1)}) (x_{\alpha}^{(1)} - \bar{x}^{(1)})^t + \sum_{\alpha=1}^m (x_{\alpha}^{(2)} - \bar{x}^{(2)}) (x_{\alpha}^{(2)} - \bar{x}^{(2)})^t$$

Luego, la función discriminante nos quedaría:

$$(A) \quad \bar{x}^t S^{-1} (\bar{x}^{(1)} - \bar{x}^{(2)}) - 1/2 (\bar{x}^{(1)} + \bar{x}^{(2)})^t S^{-1} (\bar{x}^{(1)} - \bar{x}^{(2)})$$

Vemos que cuando las poblaciones son conocidas, el criterio de clasificación es el mejor, en el sentido que minimiza la pérdida esperada en el caso de que las probabilidades se conozcan a priori, y genera los procedimientos admisibles cuando los probabilidades no se conocen. Es difícil conocer la distribución de (A), pero se puede mostrar que cuanto mayor sea n , menores serán las probabilidades de clasificar mal.

Ejemplo:

Tenemos una muestra x_1, \dots, x_n de π_1 o π_2 , y queremos clasificar a la muestra como un todo. Definimos S así:

$$(n_1+n_2+n-3) S = \sum_{\alpha=1}^{n_1} (x_{\alpha}^{(1)} - \bar{x}^{(1)}) (x_{\alpha}^{(1)} - \bar{x}^{(1)})^t + \sum_{\alpha=1}^{n_2} (x_{\alpha}^{(2)} - \bar{x}^{(2)}) (x_{\alpha}^{(2)} - \bar{x}^{(2)})^t + \sum_{\alpha=1}^n (x_{\alpha} - \bar{x}) (x_{\alpha} - \bar{x})^t$$

donde

$$\bar{x} = 1/n \sum_{\alpha=1}^n x_{\alpha}$$

El criterio es :

$$R1: \quad [\bar{x} - 1/2(\bar{x}^{(1)} + \bar{x}^{(2)})]^t S^{-1} (\bar{x}^{(1)} - \bar{x}^{(2)}) \geq \log k$$

$$R2: \quad [\bar{x} - 1/2(\bar{x}^{(1)} + \bar{x}^{(2)})]^t S^{-1} (\bar{x}^{(1)} - \bar{x}^{(2)}) \leq \log k$$

Todo lo visto anteriormente puede extenderse al caso de más de dos poblaciones.

RECONOCIMIENTO DE PATRONES VÍA PROGRAMACIÓN LINEAL

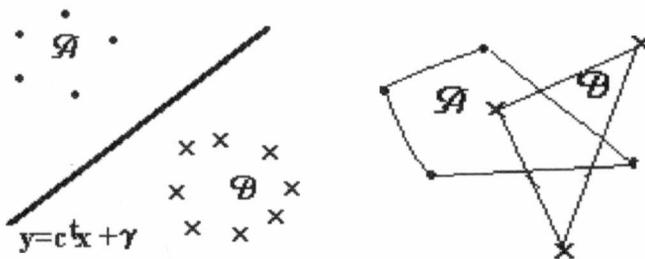
Hemos mostrado cual es la técnica estadística a seguir cuando se desean reconocer patrones, para obtener una función lineal que separa los elementos de dos, o más, poblaciones y minimiza el costo del error de clasificación errónea esperado.

Veremos ahora como tratar el tema cuando no hay información acerca de la distribución de los datos o cuando se los desea separar estrictamente en aquellos casos en que no son linealmente separables.

Queremos distinguir entre elementos de dos conjuntos de patrones distintos. Matemáticamente: Dados dos conjuntos finitos y disjuntos \mathcal{A} y \mathcal{B} de puntos en \mathbb{R}^n , queremos construir una función discriminante $f: \mathbb{R}^n \rightarrow \mathbb{R}$ tal que $f(\mathcal{A}) > 0$ y $f(\mathcal{B}) \leq 0$.

Cuando las cápsulas convexas de \mathcal{A} y \mathcal{B} no se intersectan podemos tener una función del tipo: $f(x) = c^T x + \gamma$ $c \in \mathbb{R}^n$ y $\gamma \in \mathbb{R}$. Es decir que tendríamos una función de separabilidad lineal.

En los casos de aplicaciones reales esto generalmente no pasa; las cápsulas convexas de \mathcal{A} y \mathcal{B} se intersectan y hay que resolver una función discriminante más compleja, como una función lineal a trozos que generalmente es noconvexa.



Conjuntos de patrones separables linealmente

Tenemos que ver como discriminar entre dos conjuntos de puntos en \mathbb{R}^n . Vamos a escribir a los dos conjuntos \mathcal{A} y \mathcal{B} como matrices. Si se tienen m observaciones pertenecientes al conjunto \mathcal{A} y k observaciones del conjunto \mathcal{B} , entonces la matriz $A \in \mathbb{R}^{m \times n}$ y la matriz $B \in \mathbb{R}^{k \times n}$, donde cada fila almacena un patrón, serán las representaciones matriciales de \mathcal{A} y \mathcal{B} .

Denotaremos con $C(\mathcal{A})$ y $C(\mathcal{B})$ a las cápsulas convexas de los conjuntos \mathcal{A} y \mathcal{B} .

Lema 1: Las cápsulas convexas de los conjuntos \mathcal{A} y \mathcal{B} son disjuntas si y solo si no existe $u \in \mathbb{R}^m$ y $v \in \mathbb{R}^k$ tales que:

$$(1) \quad \begin{cases} u^t A - v^t B = 0 \\ -u^t e + v^t e = 0 \\ u \geq 0, v \geq 0 \end{cases} \quad e = (1, 1, \dots, 1)$$

Dem)

Supongamos que existen u y v tales que $u^t A - v^t B = 0$ y $-u^t e + v^t e = 0$ con $u \geq 0$ y $v \geq 0$. Luego

$$\begin{aligned} (i) \quad & u^t e = v^t e \\ (ii) \quad & u^t A = v^t B = p \end{aligned}$$

es decir

$$p = \sum_{i=1}^m u_i a_i \quad p = \sum_{j=1}^k v_j b_j \quad \text{donde } a_i \text{ es una fila de } A \text{ y } b_j \text{ una fila de } B$$

Si $w = \sum u_i e$ y $y = \sum v_j e$, entonces por (i) $w = y$. Además $p^* = p/w$ pertenece a la cápsula convexa de \mathcal{A} y $p^* = p/y$ pertenece a la cápsula convexa de \mathcal{B} .

Así $p^* \in a C(\mathcal{A})$ y $p^* \in a C(\mathcal{B})$, por lo tanto la cápsula convexa de \mathcal{A} y la cápsula convexa de \mathcal{B} no son disjuntas.

Recíprocamente:

Si la cápsula convexa de \mathcal{A} y la cápsula convexa de \mathcal{B} no son disjuntas, entonces existe p tal que p pertenece a la cápsula convexa de \mathcal{A} y pertenece a la cápsula convexa de \mathcal{B} . Podemos decir entonces que:

$$p = \sum_{i=1}^m u_i a_i \quad p = \sum_{j=1}^k v_j b_j \quad \text{donde } a_i \text{ es una fila de } A \text{ y } b_j \text{ una fila de } B$$

y además $\sum u_i = \sum v_j = 1$

Así podemos decir que $p = u^t A = v^t B$ con $u^t e = v^t e = 1$. Por lo tanto si las cápsulas convexas de \mathcal{A} y \mathcal{B} no son disjuntas, existen $u \neq 0 \in \mathbb{R}^m$ y $v \neq 0 \in \mathbb{R}^k$ tales que :

$$(1) \quad \begin{cases} u^t A - v^t B = 0 \\ -u^t e + v^t e = 0 \\ u \geq 0, v \geq 0 \end{cases} \quad e = (1, 1, \dots, 1)$$

Veremos ahora que el dual de un programa lineal que trate de resolver el sistema (1) genera un plano que separa los conjuntos \mathcal{A} y \mathcal{B} cuando sus cápsulas convexas no se intersectan.

Encontrar si el sistema (1) tiene una solución, equivale a determinar si el problema de programación lineal:

$$(2) \quad \max_{u,v,r,s} \quad -(r+s)^t e$$

$$\text{s.a} \quad \begin{aligned} u^t A - v^t B + r - s &= 0 \\ -u^t e &= -1 \\ v^t e &= 1 \\ (u,v,r,s) &\geq 0 \end{aligned}$$

donde r y s son variables artificiales, tiene solución con máximo cero, ya que:

a) Si el problema (2) tiene solución (u,v,r,s) con máximo en cero, entonces:

$$\begin{aligned} -(r+s)^t e &= 0 \\ r &\geq 0 \\ s &\geq 0 \end{aligned}$$

luego $r = s = 0$. Así del problema (2) nos queda:

$$\begin{aligned} u^t A - v^t B &= 0 \\ -u^t e &= -1 \\ v^t e &= 1 \\ (u,v,r,s) &\geq 0 \end{aligned}$$

y esto es solución del problema (1)

b) Si el problema (1) tiene solución (u,v) $u \geq 0$ y $v \geq 0$, entonces existe solución admisible para el problema (2)

$$\begin{aligned} u^t A - v^t B &= 0 \\ -u^t e + v^t e &= 0 \\ (u,v) &\geq 0 \end{aligned}$$

Considerando que $u^t e = v^t e$

(i) si $u^t e = 1$ entonces $(u,v,0,0)$ es solución admisible del problema (2) con máximo en cero.

(ii) si $u^t e = w \neq 1$ entonces $u^* = (1/w) u$ es tal que $u^{*t} e = 1$
 $v^* = (1/w) v$ es tal que $v^{*t} e = 1$

luego $u^{*t} A - v^{*t} B = (1/w) (u^t A - v^t B) = 0$

Así $(u^*, v^*, 0, 0)$ es solución admisible de (2) con máximo en cero. ■

Si pensamos a (2) como el problema primal, y hallamos su problema dual tenemos

Problema Primal:

$$\begin{aligned} \max \quad & -(r+s)^t e \\ \text{s.a} \quad & u^t A - v^t B + r - s = 0 \\ & -u^t e = -1 \\ & v^t e = 1 \\ & (u, v, r, s) \geq 0 \end{aligned}$$

Problema Dual:

$$\begin{aligned} \min \quad & -\alpha + \beta \\ \text{s.a} \quad & Ac - e\alpha \geq 0 \\ & -Bc + e\beta \geq 0 \\ & e \geq c \geq -e \end{aligned} \quad (3)$$

donde el vector de costos $\text{costo} = (0, 0, -e, -e)$
el vector de términos independientes $b = (0, -1, 1)$

Recordando que si (2) tiene solución óptima acotada entonces (3) tiene solución acotada y los óptimos coinciden, el problema dual (3) nos da el siguiente criterio de separación lineal para los conjuntos \mathcal{A} y \mathcal{B} :

Teorema 1: Las cápsulas convexas de los conjuntos \mathcal{A} y \mathcal{B} son disjuntas si y sólo si el programa lineal (3) tiene un mínimo negativo ($-\alpha + \beta < 0$).

En este caso el plano $x^t c = (\alpha + \beta) / 2$ separa los conjuntos \mathcal{A} y \mathcal{B} , donde (c, α, β) es una solución de (3).

Demostración: Si (3) tiene solución con $-\alpha + \beta < 0$ entonces (2) tiene solución con $-(r+s)^t e < 0$ con $r \geq 0$ y $s \geq 0 \Rightarrow (r+s)^t e > 0$ con lo cual (1) no tiene solución admisible y $C(\mathcal{A}) \cap C(\mathcal{B}) = \emptyset$

Así, si a_i es una fila de A y b_j una fila de B,

$$\text{si } x = a_i \quad \Rightarrow x^t c = a_i^t c \geq \alpha \geq \alpha + \beta$$

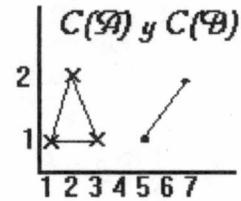
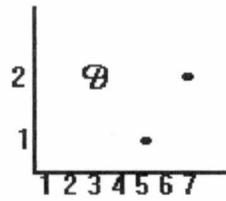
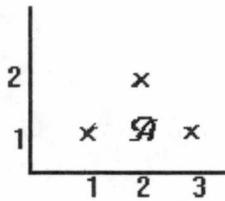
$$\text{si } x = b_j \quad \Rightarrow x^t c = b_j^t c \leq \beta \leq \beta/2 + \alpha/2$$

luego,

$$(4) \quad \begin{cases} Ac \geq e\alpha > e(\alpha + \beta)/2 \\ Bc \leq e\beta < e(\alpha + \beta)/2 \end{cases}$$

y por lo tanto $x^t c = (\alpha + \beta) / 2$ separa los conjuntos \mathcal{A} y \mathcal{B} . ■

Ejemplo:



$$A = \begin{bmatrix} 1 & 1 \\ 2 & 2 \\ 3 & 1 \end{bmatrix}$$

$$B = \begin{bmatrix} 5 & 1 \\ 7 & 2 \end{bmatrix}$$

$$\begin{aligned} \min & \quad -\alpha + \beta \\ \text{s.a.} & \quad c_1 + c_2 \geq \alpha \\ & \quad 2c_1 + 2c_2 \geq \alpha \\ & \quad 3c_1 + c_2 \geq \alpha \\ & \quad -5c_1 - 1c_2 \geq -\beta \\ & \quad -7c_1 - 2c_2 \geq -\beta \\ & \quad -1 \leq c_1 \leq 1 \\ & \quad -1 \leq c_2 \leq 1 \end{aligned}$$

En particular, si $c_0 = (-1, 0)$, $\alpha_1 = -3$, $\beta_1 = -5$ y $(\alpha + \beta) / 2 = -4$, así vemos que

$$\begin{aligned} c_0^t a_1 &= -1 \\ c_0^t a_2 &= -2 \geq -3 \geq -4 \\ c_0^t a_3 &= -3 \end{aligned}$$

$$\begin{aligned} c_0^t b_1 &= -5 \\ c_0^t b_2 &= -7 \leq -5 \leq -4 \end{aligned}$$

$$-\alpha_0 + \beta_0 = 3 - 5 = -2 < 0 \quad (c^*, \alpha^*, \beta^*) = ((-1, 0), -3, -5)$$

Por lo tanto $c_0^t x = (\alpha + \beta) / 2 = -4$ separa los conjuntos \mathcal{A} y \mathcal{B} , es decir si

$-x_1 < -4$ entonces $(x_1, x_2) \in \mathcal{B}$

$-x_1 > -4$ entonces $(x_1, x_2) \in \mathcal{A}$

$x_1 = 4$ (x_1, x_2) puede ser asignado a \mathcal{A} o a \mathcal{B}

Mostramos otro programa lineal, cuya solución permita resolver (1)

Sea el problema de programación lineal

$$(5) \quad \max_{u,v} \quad u^t e + v^t e$$

$$\text{s.a.} \quad \begin{aligned} u^t A - v^t B &= 0 \\ -u^t e + v^t e &= 0 \\ e \geq u &\geq 0 \\ e \geq v &\geq 0 \end{aligned}$$

Notar que el máximo de este problema será positivo.

Veremos que resolver este problema es equivalente a resolver el problema (1)

a) Claramente se ve que si el problema (5) tiene solución, el problema (1) también, solo cambia en que u y v están acotados superiormente por e .

b) Si el problema (1) tiene solución entonces existe solución admisible al problema (5)

$$\text{sea } u_k = \max u_i \qquad v_j = \max v_i$$

$$w = \max (u_k, v_j)$$

Si $w \leq 1$ entonces (u, v) es solución de (5).

Si $w > 1$, sean $u^* = (1/w)u$ $v^* = (1/w)v$, luego

$$0 \leq u^* \leq e \qquad 0 \leq v^* \leq e$$

$$\begin{aligned} -u^{*t} e + v^{*t} e &= (1/w) (-u^t e + v^t e) = 0 \\ u^{*t} A - v^{*t} B &= (1/w) (u^t A - v^t B) = 0 \end{aligned}$$

así (u^*, v^*) es solución admisible de (5)

Como $S = \{ (u, v) : u^t A - v^t B = 0 \quad -u^t e + v^t e = 0 \quad 0 \leq u \leq e \quad 0 \leq v \leq e \}$ es acotado, el $\max (u^t e + v^t e)$ es acotado. Además $u^{*t} e + v^{*t} e \geq 0$ entonces $\max (u^t e + v^t e) \geq 0$.

Si escribimos el dual de (5) tenemos:

Problema Primal:

$$(5) \quad \begin{aligned} & \max_{u,v} \quad u^t e + v^t e \\ & \text{s.a.} \quad u^t A - v^t B = 0 \\ & \quad \quad -u^t e + v^t e = 0 \\ & \quad \quad 0 \leq u \leq e \\ & \quad \quad 0 \leq v \leq e \end{aligned}$$

Problema Dual:

$$(6) \quad \begin{aligned} & \min_{c,y,z} \quad e^t y + e^t z \\ & \text{s.a.} \quad Ac - e^t \gamma + y \geq e \\ & \quad \quad -Bc + e^t \gamma + z \geq e \\ & \quad \quad y \geq 0 \quad z \geq 0 \end{aligned}$$

vector de costos costo = (e, e)
términos independientes (0, 0, e, e)

Si consideramos la equivalencia del lema 1 y el dual de (5) con mínimo cero obtenemos:

Teorema 2: Las cápsulas convexas de los conjuntos \mathcal{A} y \mathcal{B} son disjuntas si y sólo si el programa lineal (6) tiene mínimo cero en cuyo caso el plano $x^t c = \gamma$ separa los conjuntos \mathcal{A} y \mathcal{B} donde (c, γ, y, z) es una solución de (6). ■

Finalmente, mostramos otro programa lineal que trata de resolver (1)

Sea $\eta \in \mathbb{R}$ y consideremos el problema de programación lineal.

$$(7) \quad \begin{aligned} & \max_{u, v, \eta} \quad \eta (m + k) \\ & \text{s.a.} \quad u^t A - v^t B + \eta (e^t A - e^t B) = 0 \\ & \quad \quad -u^t e + v^t e + \eta (m + k) = 0 \\ & \quad \quad -u^t e - v^t e = -1 \\ & \quad \quad (u, v) \geq 0 \end{aligned}$$

a) Si el problema (1) tiene solución entonces existe solución admisible para (7).

Si (u, v) es solución de (1) entonces $u^t e = v^t e \neq 0$

Si $u^t e = v^t e = w^*$ puede ocurrir que $w^* = 1/2$ o que $w^* \neq 1/2$.

(i) Si $w^* = 1/2$ entonces $u^* = u$, $v^* = v$ y $\eta^* = 0$ es solución de (7).

(ii) Si $w^* \neq 1/2$ entonces $u^* = u/2w^*$ y $v^* = v/2w^*$ satisface que $u^{*t} e = v^{*t} e = 1/2$ luego $(u^*, v^*, 0)$ es admisible y $\eta^* (m + k) = 0$
Así, (7) tiene solución óptima con máximo positivo.

b) Si $\max (7)$ es positivo entonces el problema (1) tiene solución

Cuando las cápsulas convexas de \mathcal{A} y \mathcal{B} se intersectan y $(u, v, \eta = 0)$ es factible para (7) para algún $u \geq 0, v \geq 0$ tales que $e^t u = e^t v = 1/2$, y el máximo de (7) es no negativo. Así,

$$\begin{aligned} u^t A - v^t B + \eta (e^t A - e^t B) = 0 & \Rightarrow u^t A + \eta e^t A = v^t B + \eta e^t B \\ -u^t e + v^t e + \eta (-m + k) = 0 & \Rightarrow v^t e + \eta k = u^t e + \eta m = \delta \end{aligned}$$

luego,

$$\frac{(u + \eta e)^t A}{e^t u + m\eta} = \frac{(v + \eta e)^t B}{e^t v + k\eta}$$

es un punto perteneciente a $C(\mathcal{A}) \cap C(\mathcal{B})$.

Notar que los coeficientes en esta combinación suman 1:

Ej:

$$\left(\begin{array}{ccc} u_1 + \eta & u_2 + \eta & u_3 + \eta \\ \hline \delta & \delta & \delta \end{array} \right) A \quad \text{donde } \delta = e^t u + \eta m = v^t e + \eta k$$

$$\frac{u_1 + \eta}{\delta} + \frac{u_2 + \eta}{\delta} + \frac{u_3 + \eta}{\delta} = \frac{u^t e + m\eta}{\delta} = \frac{\delta}{\delta} = 1$$

■

Enunciaremos ahora otro problema lineal equivalente a decidir si (1) tiene solución, que es el dual de (7):

Teorema 3: Las cápsulas convexas de los conjuntos \mathcal{A} y \mathcal{B} son disjuntas si y sólo si el programa lineal:

$$\begin{aligned} (8) \quad \min \quad & -\rho \\ \text{s. a.} \quad & Ac - e\gamma - e\rho \geq 0 \\ & -Bc + e\gamma - e\rho \geq 0 \\ & (e^t A - e^t B)c + (-m + k)\gamma = m + k \end{aligned}$$

tiene un mínimo negativo, en cuyo caso el plano $x^t c = \gamma$ separa los conjuntos \mathcal{A} y \mathcal{B} , donde (c, γ, ρ) es una solución de (8).

■

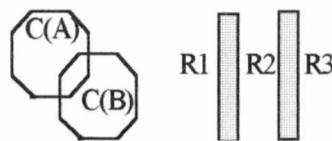
Ni (3), ni (6), ni (8) garantizan generar un plano útil para el caso en que las cápsulas convexas de \mathcal{A} y \mathcal{B} se intersecten. Esto se debe a que en los tres casos $c = 0$, y por lo tanto no obtenemos información útil para un plano que pueda ser utilizado para separar subconjuntos de \mathcal{A} y \mathcal{B} entre sí.

Es importante tener un esquema capaz de generar un plano que al menos provea una separación parcial para el caso en que sean linealmente inseparables. Debemos introducir algunas condiciones que aseguren que $c \neq 0$. Con tal fin agregaremos una restricción noconvexa. Ahora la noconvexidad aparecerá como el precio que hay que pagar por un método que garantice manejar el caso de conjuntos linealmente inseparables.

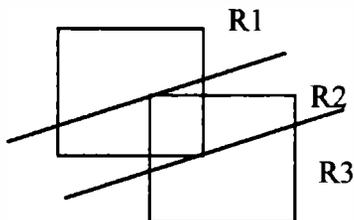
Conjuntos de patrones linealmente inseparables

Como dijimos, los programas lineales anteriores no garantizan generar un plano que separe parcialmente a \mathcal{A} de \mathcal{B} cuando $C(\mathcal{A})$ y $C(\mathcal{B})$ se intersectan.

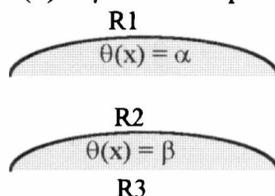
Posible Solución: Considerando que dos planos paralelos cualesquiera en \mathbb{R}^n determinan tres regiones:



Elegimos dos planos paralelos lo más cercanos posible de manera tal que R_1 contenga puntos de \mathcal{A} solamente, R_3 contenga puntos de \mathcal{B} solamente y R_2 contenga los puntos de la intersección. Si este proceso lo vamos haciendo iterativamente, descartando los puntos de afuera y quedándonos solo con los que están entre los dos planos, obtenemos una serie de planos paralelos que se usan para separar los conjuntos \mathcal{A} y \mathcal{B} .



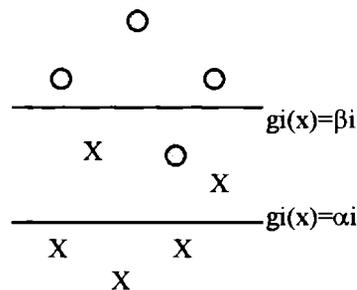
Esta alternativa puede ser extendida usando superficies paralelas. Definimos una superficie en \mathbb{R}^n como una variedad de dimensión $n-1$ y clase C^1 , determinado por la ecuación $\theta(x) = \alpha$, donde $\theta(x)$ es una función de clase C^1 sobre \mathbb{R}^n y α es una constante. Las superficies $\theta(x) = \alpha$ y $\theta(x) = \beta$ se dicen paralelas.



Procedimiento de separación: Construir una secuencia de q pares de superficies paralelas en \mathbb{R}^n : $g_i(x) = \alpha_i$, $g_i(x) = \beta_i$, $i=1, \dots, q$ tal que para cada i , $\beta_i - \alpha_i$ es minimizada:

$$\begin{aligned} \min \quad & \beta_i - \alpha_i \\ \text{s. a.} \quad & g_i(A) \geq \alpha \quad \text{donde por } g_i(A) \text{ se entiende } g_i(x) \forall x \in A \\ & g_i(B) \leq \beta \end{aligned}$$

Cuando la separación se completó, $\alpha_i \leq \beta_i$ para $i=1, \dots, q-1$ y $\alpha_q > \beta_q$
Si la separación no se completó, $\alpha_i \leq \beta_i$ para $i=1, \dots, q$



Puede mostrarse que este método separa completamente dos conjuntos disjuntos cualesquiera, si el número de superficies paralelas q es suficientemente grande. En la práctica es probable que el método no sea bueno cuando q crece.

Vamos a poner énfasis en los casos en que la función discriminante es un plano, es decir $g_i(x) = c_i^T x + \gamma_i$

Cómo determinar si dos conjuntos de patrones son linealmente separables?

El algoritmo se basa en el siguiente criterio:

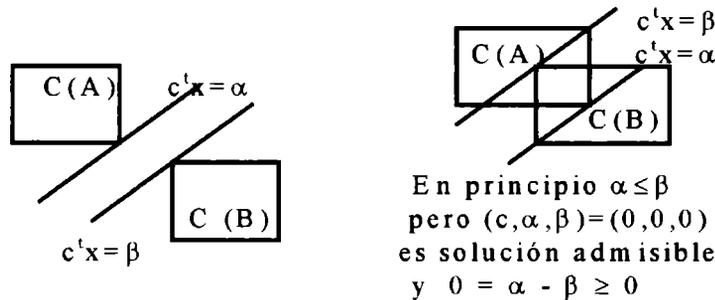
Teorema 1: Una condición necesaria y suficiente para la separabilidad lineal de dos conjuntos de patrones \mathcal{A} y \mathcal{B} es:

$$\theta(A, B) > 0$$

donde $\theta(A, B)$ es solución del programa lineal:

$$\begin{aligned} \max \quad & \alpha - \beta \\ \text{s. a.} \quad & Ac - e\alpha \geq 0 \quad \text{donde } e = (1, \dots, 1) \in \mathbb{R}^m \\ & -Bc + 1\beta \geq 0 \quad 1 = (1, \dots, 1) \in \mathbb{R}^k \\ & f \geq c \geq -f \quad f = (1, \dots, 1) \in \mathbb{R}^n \end{aligned}$$

los planos separadores son $c^t x = \alpha$ y $c^t x = \beta$



Si los conjuntos son linealmente inseparables, como $(c, \alpha, \beta) = (0, 0, 0)$ es una solución admisible, entonces $\theta(A, B) = 0$, así $c = 0$ y esto no es una solución útil.

Para asegurar la generación de un plano separador vamos a poner la condición de que el vector c sea distinto de 0, donde c es el normal al plano separador. En el método Multisurface, esto se hace junto con el programa (3) imponiendo la condición $\|c\|^2 \geq \delta$, $\delta \in [0, n]$. Así se generan dos planos, $x^t c = \alpha$ y $x^t c = \beta$ donde la intersección de \mathcal{A} y \mathcal{B} está contenida en el conjunto cerrado entre los dos planos paralelos. Repitiendo este procedimiento para los subconjuntos de \mathcal{A} y \mathcal{B} que están entre los dos planos, se obtiene una función discriminante lineal a trozos que separa \mathcal{A} y \mathcal{B} .

Con el fin de evitar que $c = 0$ sea solución, nos apoyamos en el siguiente teorema:

Teorema2: Una condición necesaria y suficiente para la separabilidad lineal de dos conjuntos de patrones \mathcal{A} y \mathcal{B} es:

$$\phi(A, B) > 0$$

donde $\phi(A, B)$ es solución del programa lineal:

$$\begin{aligned} & \max \quad \alpha - \beta \\ & c, \alpha, \beta \\ \text{s.a.} \quad & Ac - e\alpha \geq 0 \\ & -Bc + 1\beta \geq 0 \\ & f \geq c \geq -f \\ & \|c\|^2 \geq \delta \quad \quad \quad (\text{restricción no lineal } (0 < \delta \leq n)) \end{aligned}$$

Existen dos posibles métodos para este problema:

- 1) Encontrar un punto c^*, α^*, β^* que satisfaga las condiciones necesarias de optimalidad de Kuhn- Tucker.

2) Encontrar una solución δ aproximada reemplazando la restricción no lineal por su linealización sobre $c = p$, donde p es un vector n -dimensional no nulo, es decir pidiendo

$$p^t c \geq \frac{1}{2} (\delta + p^t p)$$

pues

$$h(p) = \|p\|^2 - \delta = p^t p - \delta$$

entonces

$$\nabla h(p) = 2p$$

y

$$h(c) \cong h(p) + \nabla h(p) (c-p) = p^t p - \delta + 2p^t (c-p) = -p^t p - \delta + 2p^t c$$

luego la restricción $h(c) \geq 0$ puede ser reemplazada

$$-p^t p - \delta + 2p^t c \geq 0 \text{ es decir } p^t c \geq \frac{1}{2} (\delta + p^t p)$$

Lo anterior vale para $0 < \delta \leq n$. En particular si $\delta = n$, pedir $\|c\|^2 = n$ y $-e \leq c \leq e$, equivale a pedir que c sea uno de los vértices del cubo $\{c : \|c\|_\infty \leq 1\}$ donde $\|c\|_\infty = \max |c_i|$. Nos apoyaremos en esto para mostrar que el problema es NP-Completo si $\delta = n$:

$$\begin{array}{ll} \min & -\alpha + \beta \\ (c, \alpha, \beta) & \\ \text{s. a.} & Ac - e\alpha \geq 0 \\ (9) & -Bc + e\beta \geq 0 \\ & e \geq c \geq -e \quad \|c\|^2 \geq n \quad (\text{notar que esto equivale a } \|c\|^2 = n) \end{array}$$

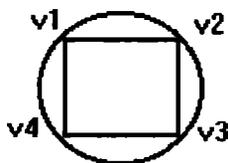
Si $x \in \mathbb{R}^n$, el problema tiene restricciones lineales del tipo $a_i^t c \geq \alpha$, $b_i^t c \leq \alpha$, $1 \geq c_i \geq -1$ y una restricción no lineal $c_1^2 + \dots + c_n^2 \geq \delta$

Teorema 4: El programa (9) con $\delta = n$ es NP-completo:

(10) Es el mínimo de $-\alpha + \beta$, sujeto a las mismas restricciones que (9), negativo?

Dem)

Vemos que $\|c\|_\infty \leq 1$ y $\|c\|^2 \geq n$ son equivalentes a pedir que c sea uno de los vértices del cubo $\{c \mid \|c\|_\infty \leq 1\}$.



El problema (10) está en NP, porque encontrar al vértice del cubo que da el mínimo valor a $(-\alpha + \beta)$ llevará tiempo polinomial al evaluar $\alpha - \beta$, donde

$$\alpha = \min_{1 \leq i \leq m} A_i c \qquad \beta = \max_{1 \leq i \leq k} B_i c$$

Esto es: conocida una instancia c_0 para hallar una solución óptima de (9) con $c = c_0$, conviene elegir α lo más grande posible y β lo más chico posible para encontrar el $\max(\alpha - \beta)$ (equivale al $\min(\beta - \alpha)$), pues

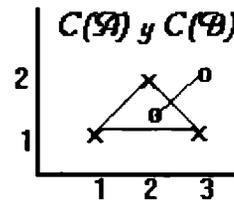
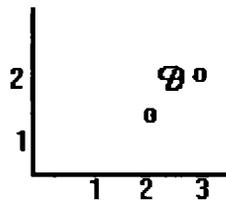
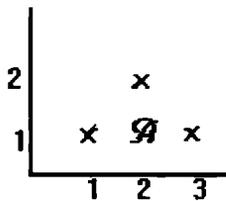
$$\left. \begin{aligned} Ac - e\alpha \geq 0 &\Rightarrow \alpha \leq \min_{1 \leq i \leq m} a_i^t c_0 \\ -Bc + e\beta \geq 0 &\Rightarrow \beta \geq \max_{1 \leq i \leq k} b_i^t c_0 \end{aligned} \right\} (Q)$$

Luego dado $c = c_0$, elegimos $\alpha_0 = \min a_i^t c_0$ $\beta_0 = \max b_i^t c_0$ y tenemos una solución factible para (9)

Si hallamos α_0 y β_0 para cada c_0 factible, entonces recorremos todos los vértices (2^n posibilidades) y podemos hallar la mejor solución.

Pero si tengo el vértice c^* correcto, verificar si $(\alpha - \beta) \leq 0$ es resolver (Q) para c^* , y eso puede hacerse en tiempo polinomial.

Ejemplo:



$$A = \begin{bmatrix} 1 & 1 \\ 2 & 2 \\ 3 & 1 \end{bmatrix}$$

$$B = \begin{bmatrix} 2 & 3/2 \\ 3 & 2 \end{bmatrix}$$

Deseamos resolver

$$\begin{aligned}
 \min \quad & -\alpha + \beta \\
 \text{s. a.} \quad & c_1 + c_2 \geq \alpha \\
 & 2c_1 + 2c_2 \geq \alpha \\
 & 3c_1 + c_2 \geq \alpha \\
 & -2c_1 - 3/2 c_2 \geq -\beta \\
 & -3c_1 - 2c_2 \geq -\beta \\
 & -1 \leq c_1 \leq 1 \\
 & -1 \leq c_2 \leq 1 \\
 & c_1^2 + c_2^2 = 2
 \end{aligned}$$

En particular, si $c_0 = (-1, -1)$ $c_0^t a_1 = -2$
 $c_0^t a_2 = -4 \Rightarrow \alpha_0 = -4$
 $c_0^t a_3 = -4$

$$\begin{aligned}
 c_0^t b_1 &= -7/2 \\
 c_0^t b_2 &= -5 \Rightarrow \beta_0 = -7/2
 \end{aligned}$$

Luego

$$\begin{aligned}
 c_0^t x &= \alpha_0 & x_1 + x_2 &= 4 \\
 c_0^t x &= \beta_0 & x_1 + x_2 &= 7/2
 \end{aligned}$$

$$-\alpha_0 + \beta_0 = 4 - 7/2 = 1/2 > 0 \quad (c^*, \alpha^*, \beta^*) = ((-1, -1), -4, -7/2)$$

Analizando las restantes soluciones se verifica que $-\alpha^* + \beta^* > 0$, y que los conjuntos no son separables. Así, cuando c es el vértice óptimo, comprobar si (9) es menor o igual que 0 implica hallar α^* y β^* , lo que se hace en tiempo polinomial, y verificar $-\alpha^* + \beta^* \leq 0$.

Vamos a ver que el problema de determinar si el mínimo de (9) es negativo es NP-hard. Para ello lo reduciremos al siguiente problema de partición:

(11) Es $d^t x = 0$ para algún $x \in \mathbb{R}^n$ tal que $e \geq x \geq -e$, $\|x\|^2 \geq n$? donde (d_1, \dots, d_n) las componentes de d son enteros positivos.

Veamos que (11) es una instancia de (10)

$$\text{Si } A = \begin{pmatrix} d \\ -d \end{pmatrix} \quad \text{y } B = \begin{pmatrix} -d \\ d \end{pmatrix}$$

el problema (9) sería:

$$\begin{aligned} \min & \quad -\alpha + \beta \\ \alpha, \beta, x & \\ & d^t x \geq \alpha \\ & -d^t x \geq \alpha \\ & -d^t x \leq \beta \\ & d^t x \leq \beta \\ & -e \leq x \leq e \\ & \|x\|^2 \geq n \end{aligned}$$

El problema (10) sería el siguiente:

$$\begin{aligned} \text{Es } \min & \quad -\alpha + \beta \\ \alpha, \beta, x & \\ \text{s.a. } & \begin{pmatrix} d \\ -d \end{pmatrix} x \geq \begin{pmatrix} \alpha \\ \alpha \end{pmatrix} \\ & \begin{pmatrix} -d \\ d \end{pmatrix} x \leq \begin{pmatrix} \beta \\ \beta \end{pmatrix} \\ & e \geq x \geq -e \quad \|x\|^2 \geq n \end{aligned}$$

menor o igual que 0 ?

Notar que $\alpha \leq d^t x \leq -\alpha \Rightarrow \alpha \leq 0$ | Así $-\alpha + \beta \geq 0$ y si existe solución con $\min \leq 0$
 $-\beta \leq d^t x \leq \beta \Rightarrow \beta \geq 0$ | entonces $\alpha = \beta = 0$ y $d^t x = 0$

En este caso resolver (10) es equivalente a resolver (11).

Podemos decir que el problema (10) es NP-hard, como también es NP entonces por definición es NP-completo.

Un algoritmo polinomial para el problema de separación

Apoyándonos en que $\|c\|^2 = n$ y $-e \leq c \leq e$ es equivalente a $\|c\|_\infty = 1$, veremos que si $\|c\|_\infty = 1$ es la condición para el programa (3) en lugar de $\|c\|^2 = n$, el problema

$$\begin{aligned} (12) \quad \min & \quad -\alpha + \beta \\ c, \alpha, \beta & \\ \text{s.a. } & Ac - e\alpha \geq 0 \\ & -Bc + e\beta \geq 0 \\ & \|c\|_\infty = 1 \end{aligned}$$

puede resolverse en tiempo polinomial.

Teorema 5: El programa noconvexo (12) puede resolverse en tiempo polinomial, resolviendo $2n$ programas lineales para $i = 1, \dots, n$

$$(13) \quad \begin{array}{ll} \min & -\alpha + \beta \\ & c, \alpha, \beta \\ \text{s.a.} & Ac - e\alpha \geq 0 \\ P_i & -Bc + e\beta \geq 0 \\ & e \geq c \geq -e \\ & c_i = \pm 1 \end{array}$$

y tomando la solución $(\alpha^*, \beta^*, c^*) = \operatorname{argmin} P_i$, es decir aquella que minimiza $(-\alpha + \beta)$ entre las $2n$ soluciones de (13) con $i = 1, \dots, n$.

Dem)

Sea $(c^i, \alpha^i, \beta^i) \in$

$$(14) \quad \begin{array}{ll} \arg \min & -\alpha + \beta \\ & c, \alpha, \beta \\ \text{s.a.} & Ac - e\alpha \geq 0 \\ & -Bc + e\beta \geq 0 \\ & e \geq c \geq -e \\ & c_i = \pm 1 \end{array}$$

Para un i fijo, $(c^{i*}, \alpha^{i*}, \beta^{i*})$ puede obtenerse resolviendo dos programas lineales del tipo (13), uno con $c^{i+} = (c_1, c_2, \dots, c_{i-1}, 1, c_{i+1}, \dots, c_n)$ y otro con $c^{i-} = (c_1, c_2, \dots, c_{i-1}, -1, c_{i+1}, \dots, c_n)$ y tomar la solución que de el menor valor a $(-\alpha + \beta)$. Sea (c^*, α^*, β^*) una solución de (12). Como $\|c^{i*}\|_\infty = 1$ entonces:

$$(15) \quad -\alpha^* + \beta^* \leq \min_{1 \leq i \leq n} (-\alpha^{i*} + \beta^{i*});$$

$|c^{*j}| = 1$, para algún j y $-1 \leq c^{*i} \neq j \leq 1$ entonces:

$$(16) \quad -\alpha^{j*} + \beta^{j*} \leq -\alpha^* + \beta^* \text{ porque } -e \leq c^{*j} \leq e, (c^{j*}, \alpha^{j*}, \beta^{j*}) \text{ es solución óptima de } P_j \text{ y } \|c^{*j}\| \text{ es admisible para } P_j.$$

Combinando (15) y (16) tenemos:

$$(17) \quad -\alpha^* + \beta^* \leq \min_{1 \leq i \leq n} -\alpha^i + \beta^i \leq -\alpha^j + \beta^j \leq -\alpha^* + \beta^*$$

Por lo tanto $-\alpha^* + \beta^* = \min(-\alpha_i^* + \beta_i^*) = -\alpha_j^* + \beta_j^*$.

Así $(c_j^*, \alpha_j^*, \beta_j^*)$ resuelve (12). Como se necesitan $2n$ programas lineales para computar (c, α, β) y cada programa lineal se resuelve en tiempo polinomial, entonces (12) se resuelve en tiempo polinomial, resolviendo $2n$ programas lineales de (13).

Puede suceder que en una iteración ningún punto sea separado. Ante esa situación resolvemos el caso degenerado:

Se fija un elemento, A_r , del conjunto \mathcal{A}^j y se deja al conjunto \mathcal{B}^j con todos sus elementos, los separa y encuentra un c^* , α^* y un β^* .

Para la iteración j redefine:

$$c^{i(j)} = c^*$$

$$\alpha^{i(j)} = -\infty$$

$$\beta^{i(j)} = \beta^*$$

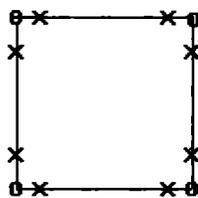
con lo que

- 1) $c^{i(j)}A_r = c^*A_r \geq \alpha^{i(j)} \geq -\infty$
- 2) $c^{i(j)}B_j = c^*B_j \geq \beta^{i(j)} = \beta^*$
- 3) $-\alpha^{i(j)} + \beta^{i(j)} = \beta^* + \infty = \infty \geq 0$

1 y 2 nos dicen que $(c^{i(j)}, \alpha^{i(j)}, \beta^{i(j)})$ es admisible, y 3 nos dice que no separa a los conjuntos.

Además $\{B_s \in \mathcal{B}^j / B_s c^{i(j)} \geq \alpha^{i(j)}\} = \mathcal{B}^j$
 $\mathcal{B}^{j+1} = \mathcal{B}^j$

Como $-\alpha^* + \beta^* \leq 0$ y como $c^{i(j)}A_r \geq \alpha^* \geq \beta^*$ entonces $A_r \notin \mathcal{A}^{j+1}$ con lo cual $\mathcal{A}^{j+1} \neq \mathcal{A}^j$



Describimos ahora el algoritmo que separa dos conjuntos de puntos disjuntos \mathcal{A} y \mathcal{B}

Algoritmo I

Discrimina entre dos conjuntos \mathcal{A} y \mathcal{B} representados por las matrices $A \in \mathbb{R}^{m \times n}$ y $B \in \mathbb{R}^{k \times n}$.

Paso 0) $j = 0$ $\mathcal{A}^0 = \mathcal{A}$ $A^0 = A$ $\mathcal{B}^0 = \mathcal{B}$ $B^0 = B$ e ingresa un entero j_{\max} . Resolver el programa lineal (3). Si el mínimo de (3) es negativo, parar, el plano:

$$x^t c = (\alpha + \beta) / 2 \text{ separa } \mathcal{A} \text{ y } \mathcal{B}.$$

Paso i) Resolver los $2n$ programas lineales (13) con $A=A^j$ y $B=B^j$. Sea $(c^{\pm i}, \alpha^{\pm i}, \beta^{\pm i})$ una solución de cada programa lineal correspondiente a $c_i = \pm 1$ y define $i(j) = \arg \min(\text{cardinalidad}\{r \mid A_r c^{\pm i} \leq \beta^{\pm i}\} + \text{cardinalidad}\{s \mid B_s c^{\pm i} \geq \alpha^{\pm i}\})$ y sea $(c^{i(j)}, \alpha^{i(j)}, \beta^{i(j)})$ una solución de uno de los programas lineales de (13) correspondiente a $i(j)$.

En este paso el conjunto cerrado entre los planos paralelos $x^t c^{i(j)} = \alpha^{i(j)}$ y $x^t c^{i(j)} = \beta^{i(j)}$ contiene el número menor de puntos de \mathcal{A}^j y \mathcal{B}^j , mientras que los semiespacios restantes contienen el resto de las partes separadas de \mathcal{A}^j y \mathcal{B}^j .

Paso ii) Sea $\mathcal{A}^{j+1} = \{A_r \in \mathcal{A}^j \mid A_r c^{i(j)} \leq \beta^{i(j)}\}$
 $\mathcal{B}^{j+1} = \{B_s \in \mathcal{B}^j \mid B_s c^{i(j)} \geq \alpha^{i(j)}\}$

Si $\mathcal{A}^{j+1} \neq \mathcal{A}^j$ o $\mathcal{B}^{j+1} \neq \mathcal{B}^j$ ir al paso iv)

Paso iii) Encontrar una fila A_r de A^j (Caso a) o fila B_s de B^j (Caso b) tal que cuando el programa lineal (3) se resuelve con $A=A_r$ y $B=B^j$ (Caso a), o $A=A^j$ y $B=B_s$ (Caso b), el mínimo del programa lineal es negativo. En cualquier caso la, solución es (c^*, α^*, β^*) .

$$\text{Caso a : } c^{i(j)} = c^* \quad \alpha^{i(j)} = -\infty \quad \beta^{i(j)} = \beta^* \\ \mathcal{A}^{j+1} = \{A_i \mid A_i \in \mathcal{A}^j \mid A_i c^* \leq \beta^*\} \quad \mathcal{B}^{j+1} = \mathcal{B}^j$$

$$\text{Caso b : } c^{i(j)} = c^* \quad \alpha^{i(j)} = \alpha^* \quad \beta^{i(j)} = \infty \\ \mathcal{A}^{j+1} = \mathcal{A}^j, \quad \mathcal{B}^{j+1} = \{B_i \mid B_i \in \mathcal{B}^j \mid B_i c^* \geq \alpha^*\}$$

Elimina al menos un punto de \mathcal{A}^j o \mathcal{B}^j y así asegura que $\mathcal{A}^{j+1} \neq \mathcal{A}^j$ o $\mathcal{B}^{j+1} \neq \mathcal{B}^j$.

Paso iv) Guardamos los planos $x^t c^{i(j)} = \alpha^{i(j)}$ y $x^t c^{i(j)} = \beta^{i(j)}$

Paso v) Si $\mathcal{A}^{j+1} = \mathcal{B}^{j+1} = \emptyset$, reemplazamos j_{\max} por j y paramos, sino incrementamos j en uno y volvemos al paso (i)

Cuando \mathcal{A} y \mathcal{B} no son linealmente separables, este algoritmo construye una secuencia de planos paralelos: $x^i c^{i(j)} = \alpha^{i(j)}$ $x^i c^{i(j)} = \beta^{i(j)}$ $j = 0, \dots, j_{\max}$. tales que si j_{\max} es suficientemente grande, los conjuntos \mathcal{A} y \mathcal{B} se separan por el procedimiento:

Procedimiento II

$j = 0$ ingresamos j_{\max} y $x \in \mathbb{R}^n$

- i) Si $j = j_{\max}$ ir a iv).
- ii) Si $x^i c^{i(j)} > \beta^{i(j)}$ entonces $x \in \mathcal{A}$, parar.
Si $x^i c^{i(j)} < \alpha^{i(j)}$ entonces $x \in \mathcal{B}$, parar.
- iii) Incrementar j e ir a (i).
- iv) Si $x^i c^{i(j)} \geq (\alpha^{i(j)} + \beta^{i(j)})/2$ entonces $x \in \mathcal{A}$, parar.
Si $x^i c^{i(j)} < (\alpha^{i(j)} + \beta^{i(j)})/2$ entonces $x \in \mathcal{B}$, parar.

Si se hace la siguiente suposición, nunca ocurrirá degeneración.

Suposición:

_Si los conjuntos disjuntos \mathcal{A} y \mathcal{B} son linealmente inseparables, para al menos uno de los $2n$ programas lineales de (13), $i = 1, \dots, n$ cuya solución es (c^i, α^i, β^i) tenemos:

$$\{A_i \mid A_i c^i > \beta^i\} \cup \{B_i \mid B_i c^i < \alpha^i\} \neq \emptyset$$

Degeneración

Cuando la degeneración ocurre, ningún punto es separado. Es decir la región R_2 comprendida entre los hiperplanos $x^t c \geq \alpha$ y $x^t c \leq \beta$ contiene a todos los puntos de \mathcal{A} y \mathcal{B} .

El procedimiento a seguir en esa situación es elegir un punto de uno de los conjuntos, dejando el otro conjunto igual y buscar un hiperplano que los separe. Alcanzar tal objetivo puede implicar recorrer todos los puntos del conjunto, de ahí que en la implementación se deja fijo el conjunto de mayor cardinalidad y la separabilidad es testada eligiendo un punto sobre el conjunto de menor cardinalidad.

Para elegir dicho punto el siguiente criterio se propone:

Sea \mathcal{A}_j y \mathcal{B}_j los conjuntos alcanzados en la iteración j en que la degeneración ocurre. Supongamos que la cardinalidad de \mathcal{A}_j es menor a la de \mathcal{B}_j , por tanto iniciamos un procedimiento interno fijando $\mathcal{B}_j = \mathcal{B}_j$ y $\mathcal{A}_j = \{ a_i \}$, si existe $a_i \in \mathcal{A}_j$ tal que $c^t a_i = \beta$ entonces a_i es el primero en ser elegido; caso contrario se elige a_i tal que $\min |c^t a_i - \beta|$, es decir la restricción cuya variable dual tiene menor valor absoluto.

El objetivo de tal procedimiento es poder encontrar un hiperplano separador en la menor cantidad de iteraciones posibles, tal como se muestra en los resultados computacionales.

Extensión al caso de más de dos conjuntos

En el caso en que los datos provienen de tres conjuntos A_1 , A_2 , y A_3 y se quiere construir un función que los separe, se propone el siguiente procedimiento:

Supongamos $k_1 = \# A_1$, $k_2 = \# A_2$, $k_3 = \# A_3$ y $k_1 \leq k_2 \leq k_3$.

En una primera etapa fijamos $\mathcal{A} = A_1 \cup A_2$ y $\mathcal{B} = A_3$ y construimos una función que separe $A_1 \cup A_2$ de A_3 .

En una segunda etapa fijamos $\mathcal{A} = A_1$, $\mathcal{B} = A_2$ y construimos una función que separe A_1 de A_2 .

Finalmente el procedimiento **reconstruye una función lineal** a trozos que separa a los tres conjuntos, la que es utilizada cada vez que se quieren testear a que conjunto pertenece un dato nuevo.

Notar que la elección de $\mathcal{A} = A_1 \cup A_2$ y $\mathcal{B} = A_3$ no es arbitraria, ya que en el peor caso, el número de iteraciones mayores necesarios para la primera etapa es $(k_1 + k_2) + k_3$ y en la segunda etapa $(k_1 + k_2)$ de donde el trabajo total será del orden de $2(k_1 + k_2) + k_3$ iteraciones, de ahí el criterio para la elección de \mathcal{A} y \mathcal{B} .

El procedimiento puede ser extendido para cuatro o más conjuntos.

Etapa de Implementación

El lenguaje de programación con el que trabajé fue Turbo Pascal, que como todo lenguaje tiene sus ventajas y desventajas. Por un lado es un lenguaje claro, permite trabajar con distintas estructuras de datos y programas bien modularizados. Es un lenguaje tipado, pero el manejo de tipos no es muy eficiente, en determinadas situaciones nos encontramos con algunos problemas como por ejemplo tener dos procedimientos similares, debido a que hacen lo mismo pero reciben parámetros de distintos tipos. Tal es el caso de asignación de vectores con distintos elementos, la solución fue tener dos procedimientos de asignación de vectores, esto no es lo más deseado pero es la forma de resolverlo.

En esta aplicación trabajo con dos conjuntos de puntos donde podemos tener más de 100 puntos en cada conjunto y trabajar en \mathbb{R}^{100} o más. La mejor forma de trabajar con estos conjuntos fue guardándolos en una matriz, hay una matriz por conjunto donde cada fila representa un punto del mismo. Los problemas al usar Pascal, fueron relacionados con el manejo de memoria, como no permitía tener más de una matriz de tamaño grande, tuve que definir punteros a las matrices para poder trabajar.

El algoritmo para encontrar los pares de hiperplanos paralelos que separan a los conjuntos, va resolviendo en cada iteración un problema de minimización ($-\alpha + \beta$). Este problema lo resuelve el software LINDO (Linear Interactive Discrete Optimizer), que resuelve problemas de programación lineal.

Los datos, en este caso los pares de hiperplanos, o el hiperplano si se trata de conjuntos separables, son guardados en un archivo, para que luego sean usados por el programa que calcula a qué conjunto pertenecen distintos puntos nuevos.

La desventaja de usar LINDO es que permite tener una determinada cantidad de variables y de restricciones, por esta razón es que el programa está preparado para resolver problemas relativamente chicos. Cada vez que se arma el archivo para ejecutar LINDO, éste pasa por un filtro que le hace las modificaciones necesarias para que no de error sobre formato del problema, como por ejemplo que las restricciones no tengan un largo mayor a 80 caracteres. LINDO solo permite que los problemas a resolver tengan como máximo 59 filas, entre restricciones y la función objetivo, con lo cual tenemos otra restricción en cuanto a la cantidad de puntos que puede tener cada conjunto.

El programa que calcula los hiperplanos separadores puede modificarse para usar cualquier otro paquete de software que resuelva problemas de minimización, esto es un paso simple ya que solo cambiarán los procedimientos de armado del problema y la ejecución del nuevo software. Esta es una de las ventajas que se obtiene al tener una programación modularizada.

Para resolver problemas con tres conjuntos, no hubo que hacer demasiadas modificaciones, ya que se resuelve simulando dos conjuntos, donde uno será la unión de dos de los conjuntos, y luego se vuelven a separar esos dos conjuntos.

Se realizaron varios ejemplos de prueba, con puntos de coordenadas negativas y positivas y se obtuvieron los resultados esperados.

Como en la mayoría de los ejemplos, se debían ingresar demasiados puntos, el programa fue modificado para que la entrada pueda ser por pantalla o por archivo. El formato que debe tener el archivo se explica en Uso del Software(pag 54).

Estructuras de Datos

Para almacenar los puntos de cada conjuntos se usaron matrices, donde cada fila representa un punto del conjunto, en el programa lo que se manejan son punteros a esas matrices:

```
type mat = array [1..100,1..100] of real
matriz = ^mat
```

Se trabaja con punteros a vectores de dos tipos, uno contiene reales, que es el vector que se obtiene al buscar los planos separadores:

```
vec = array [1..50] of real
vector = ^vec
```

El otro tipo de vector es el que en cada posición guarda un registro que es la solución de la iteración correspondiente, es decir el plano alfa, el beta y el vector c:

```
datos = record
    pa : real
    pb : real
    c : vector
end
ve = array [1..50] of datos
vect = ^ve
```

Procedimientos más importantes

A continuación se describen los procedimientos más importantes que fueron implementados para la solución del problema de separación.

El procedimiento **Resolver** es el procedimiento principal, dadas las dos matrices que representan los conjuntos de puntos, arma el problema en un archivo que luego se filtra para que pueda ser entendido por LINDO(debido a las restricciones de este software), devuelve la solución en un archivo para que pueda ser luego leído y comprobar si la solución dio negativa y parar ó si hay que seguir iterando:

```

Procedure Resolver ( var A, B : matriz);

begin

Asignar_Matriz(A,Aj,m,n);
Asignar_Matriz(B,Bj,r,n);
Armar_Archi(Aj,Bj,m,r);
Filtro('problema.mps','problema.mp',m+r,n*2+2);
EjecutarLindo('/c lindo.bat');
Assign ( Salida, 'solucion.mps');
Leer(Salida,min);
j:=1;
if ( min < 0) then
  begin
    plano := (alfa + beta)/2;
    Escribir(Solución, plano separador)
  end
else
  begin
    While (j<jmax) do
      begin
        Resolver_Pi(Aj, Bj, plapos, planeg);
        Definir_ij(Aj, Bj,pla,plapos,planeg, ij);

        { Puntos solución de la iteración j}
        alfaij := pla.pa;
        betaij := pla.pb;
        Asignar_Vector(cij, pla.c,n);

        If ( (- alfaij + betaij ) >= 0 ) then
          begin
            Armar_Nuevas(Aj,Bj,A,B,mn,rn);

            if (Iguales(A,Aj,mn,m,n) and Iguales(B,Bj,rn,r,n)) then
              begin
                ResolverA o ResolverB      { Caso Degenerado}
              end;
          end;
        end;
      end;
    end;
  end;
end;

```

```
{ Guardamos la solución de la iteración j }
```

```
planos^[j].pa:= alfaij;
planos^[j].pb:= betaij;
Asignar_Vector(planos^[j].c,cij,n);
```

```
If (IguarCero(A,mn,n) and IguarCero(B,rn,n)) then
```

```
begin
```

```
  j := jmax;          { Se acabaron los puntos de A o B, termina }
```

```
end
```

```
else
```

```
begin          { Si quedan puntos por separar sigue iterando }
```

```
  j:=j+1;
```

```
  Asignar_Matriz(A,Aj,mn,n);
```

```
  Asignar_Matriz(B,Bj,rn,n);
```

```
  m := mn;
```

```
  r := rn;
```

```
end;
```

```
end
```

```
else
```

```
  { Si  $-alfaij + betaij < 0$ , guarda la solución y termina }
```

```
begin
```

```
  planos^[j].pa:= alfaij;
```

```
  planos^[j].pb:= betaij;
```

```
  new(planos^[j].c);
```

```
  Asignar_Vector(planos^[j].c,cij,n);
```

```
  j := jmax;
```

```
end;
```

```
end;
```

```
{ Cuando termina de iterar escribe en un archivo los planos separadores }
```

```
end;
```

En cada iteración, si la función a minimizar no dio negativa, se deben resolver $2n$ programas lineales, que se diferencian en que cada uno tiene fija una componente del vector c en 1 y luego en -1, es para que no obtengamos como resultado al vector $c = 0$, el algoritmo que realiza este paso el es **Resolver_Pi** que guardará en el vector $plapos$ las n soluciones obtenidas con las distintas componentes con el calor 1 y $planeg$ con las componentes en -1:

```
Procedure Resolver_Pi(Aj,Bj : matriz; var plapos,planeg : vect);
```

```
begin
```

```
for i := 1 to n do
```

```
begin
```

```
  Armar_Problema ( Aj, Bj, 1,i);
```

```
  Filtro('mejor.mps', 'mejor.mp',m+r, n*2+2);
```

```
  EjecutarLindo ( '/c prueba.bat');
```

```
  { Guardar en plapos }
```

```
  Assign ( Sol, 'solu.mps');
```

```
  LeerDatos(Sol,plapos,i);
```

```
  Armar_Problema ( Aj, Bj, -1,i );
```

```
  Filtro('mejor.mps', 'mejor.mp', m+r, n*2+2);
```

```
  EjecutarLindo ('/c prueba2.bat');
```

```
  { Guardar en planeg }
```

```
  Assign ( Sol, 'soluneg.mps');
```

```
  LeerDatos ( Sol,planeg,i);
```

```
end;
```

```
end;
```

Una vez resueltos los $2n$ problemas lineales, encontramos un ij tal que la cantidad de puntos que queden sin clasificar, es decir entre los dos hiperplanos, sea mínima, ese procedimiento es **Definir_ij**:

```
Procedure Definir_ij ( A,B : matriz ; var pla: datos; plapos,planeg : vect; var ij : integer);
```

```
begin
```

```
  BuscarPlano(A,B,cant,pla,plapos,ij);
```

```
  BuscarPlano(A,B,cant,pla,planeg,ij);
```

```
end;
```

BuscarPlano es el procedimiento que calcula cuales son los puntos que están entre los dos planos y retorna el ij donde la cantidad es mínima, trabaja con los vectores $plapos$ y $planeg$ que eran los que tenían las $2n$ soluciones:

```
Procedure BuscarPlano( A,B : matriz; cant : real; pla : datos; plano : vect; ij : integer);
```

```
begin
for i := 1 to n do
begin
cont := 1;
conj[0] := 0;
alf := plano^[i].pa;
bet := plano^[i].pb;
Asignar_Vector ( c, plano^[i].c, n );
for j := 1 to m do
begin
sum := 0;
for k := 1 to n do
sum := sum + ( A^[j,k]*c^[k] );
if ( sum <= bet ) then
begin
conj[cont] := j;
cont := cont + 1;
conj[0] := conj[0] + 1;
end;
end;
for j := 1 to r do
begin
sum := 0;
for k := 1 to n do
sum := sum + ( B^[j,k]*c^[k] );
if ( sum >= alf ) then
begin
conj[cont] := j;
cont := cont + 1;
conj[0] := conj[0] + 1;
end;
end;
total := conj[0];
if ( total <= cant ) then
begin
cant := total;
ij := i;
Asignar_Puntos(pla,plano,ij);
end;
end;
end;
```

Armar_Nuevas es el procedimiento que arma las matrices nuevas, es decir las que tienen los puntos que no pudieron ser clasificados de cada conjunto:

Procedure Armar_Nuevas(Aj,Bj:matriz; var An,Bn: matriz; var mn, rn : integer);

```

begin
mn:=1;
rn:=1;
for i:= 1 to m do
begin
s:=0;
for j:= 1 to n do
begin
s:= s + Aj^[i,j] * cij^[j];
end;
if (s<= betaij) then
begin
for j:=1 to n do
An^[mn,j]:= Aj^[i,j];
mn:= mn+1;
end;
end;
mn:= mn-1;
for i:= 1 to r do
begin
s:=0;
for j:= 1 to n do
s:= s + Bj^[i,j]*cij^[j];
if (s>= alfaij) then
begin
for j:=1 to n do
Bn^[rn,j]:= Bj^[i,j];
rn:= rn+1;
end;
end;
rn:= rn-1;
end;

```

Para los casos en que ocurre degeneración se procede de la forma como se describió en Degeneración (Pág. 33), **ResolverA** y **ResolverB** son similares, solo que cada uno trabaja con la matriz A y B respectivamente:

Procedure ResolverA (Aj,Bj : matriz ; var alfa, beta : real; var c : vector);

```

begin
fin := false;
k := 1;
While (not fin) and ( k <= m) do
begin
s:=0;
for j:= 1 to n do
s:= s + Aj^[k,j] * cij^[j];
if (s = betaij) then { Elijo este punto para trabajar}
begin
for j:=1 to n do
Ar^[1,j]:= Aj^[k,j];
Armar_Archi(Ar,Bj,l,r);
Filtro('problema.mps','problema.mp', 1+r, n*2+2);
EjecutarLindo('/c lindo.bat');
Assign ( Auxi, 'solucion.mps');
Leer(Auxi, min);
if ( min<0 ) then
begin
fin := true;
alfaij := -1.7e38;
betaij := beta;
Asignar_Vector(cij,c,n);
end;
end
else { Sino busco otro punto }
begin
k := k + 1;
end;
end;
end;
end;

```

En el caso que trabajemos con tres conjuntos el algoritmo principal sería de la forma:

Principal

```
begin
Ingresar(A1,A2,A3);
Dividir(A1,A2,A3,A,B);
Resolver(A,B);
Case uni of
1 : begin
    Asignar_Matriz(A2,A,q,n);
    Asignar_Matriz(A3,B,t,n);
    end;

2 : begin
    Asignar_Matriz(A1,A,x,n);
    Asignar_Matriz(A3,B,t,n);
    end;

3 : begin
    Asignar_Matriz(A1,A,x,n);
    Asignar_Matriz(A2,B,q,n);
    end;
end;
Resolver(A,B);
```

Experiencias Computacionales

El programa desarrollado fue probado con distintos ejemplos, aquí se describen algunos de ellos con sus resultados.

Caso 1) Dos conjuntos separables

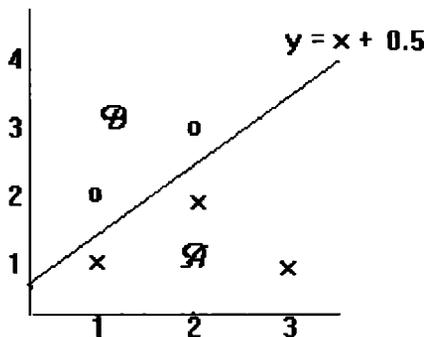
$$\mathcal{A} = \{ (1,1); (2,2); (3,1) \} \quad \mathcal{B} = \{ (1,2); (2,3) \}$$

El resultado que se obtuvo al ejecutar el programa fue el siguiente:

Los conjuntos son linealmente separables, y el hiperplano que los separa es:

$$y = x + 0.5$$

Gráficamente:



Así para todo otro punto (x,y) , si $y \geq x + 0.5$ entonces diremos que $(x,y) \in \mathcal{B}$, caso contrario, $(x,y) \in \mathcal{A}$.

Caso 2) Dos conjuntos no separables

$$\mathcal{A} = \{ (1,1); (2,2); (3,1) \}$$

$$\mathcal{B} = \{ (2, 3/2); (3,2) \}$$

El resultado que se obtuvo al ejecutar el programa fue el siguiente:

Los conjuntos no eran separables; fueron necesarias dos iteraciones para separarlos. En la primer iteración los resultados que se obtuvieron fueron:

$$x^1 c = -4 \quad x^1 c = -3.5 \quad c = (-1, -1)$$

es decir que los hiperplanos separadores fueron:

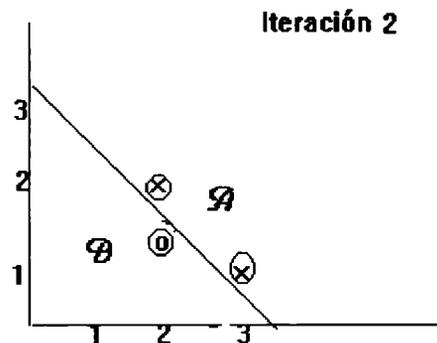
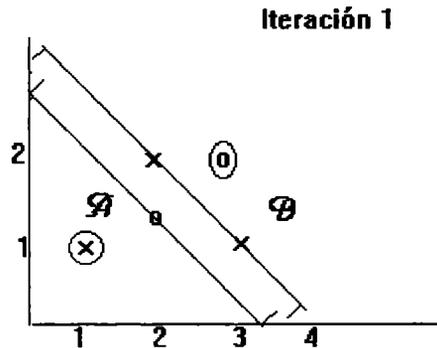
$$y = 4 - x \quad y = 3.5 - x$$

En esta iteración quedan clasificados los puntos $(1,1)$, perteneciente a \mathcal{A} y el $(3,2)$, perteneciente a \mathcal{B} .

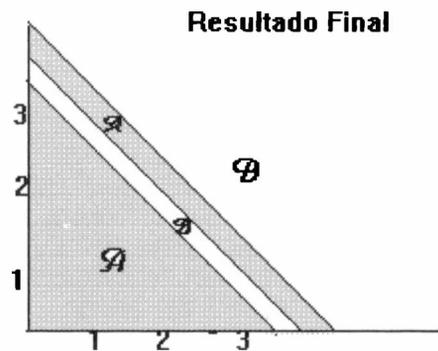
En la segunda iteración, los conjuntos quedaban separables y el hiperplano separador es:

$$y = 3.75 - x$$

Gráficamente:



En esta iteración vemos que quedan clasificados el resto de los puntos.



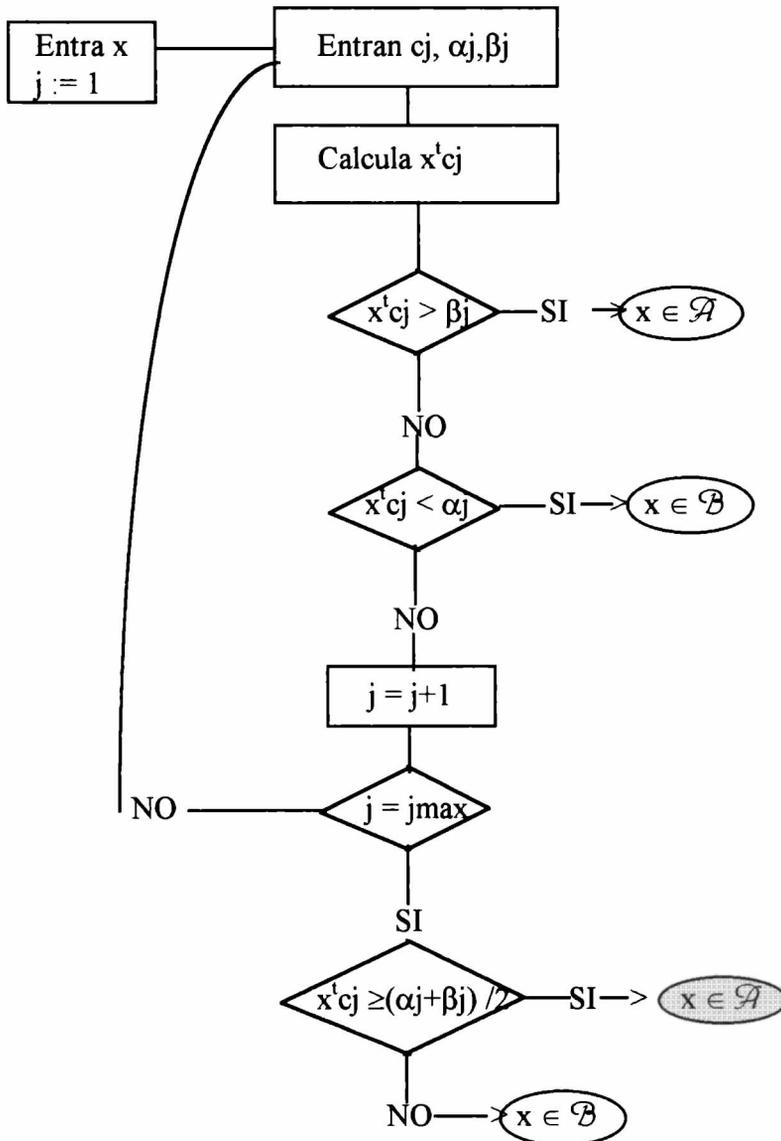
Al finalizar el procedimiento, los hiperplanos separadores obtenidos son:

$$\begin{aligned} \text{Primera iteración : } x^t c &= -4 & c &= (-1, -1) \\ & x^t c &= -3.5 \end{aligned}$$

$$\text{Segunda iteración: } x^t c = 3.75 \quad c = (1, 1)$$

La información que ellos dan permite asignar datos nuevos. Por ejemplo si queremos clasificar al punto $x = (2, 1.8)$ el algoritmo toma los planos de la primer iteración, $x^t c = -3.8$, como esto es menor que β y mayor que α , pasa a la segunda iteración, que en este caso es la última, entonces vemos que $x^t c > (\alpha + \beta)/2$, de esta manera $x \in \mathcal{A}$.

Los pasos que sigue el algoritmo de separación en este ejemplo podemos observarlos en el siguiente gráfico:

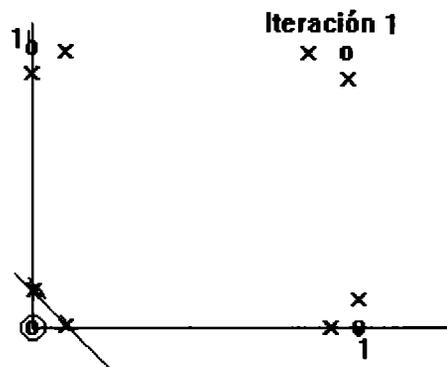


Caso 3) Caso de degeneración con dos conjuntos

$$\mathcal{A} = \{ (0,0); (0,1); (1,0); (1,1) \}$$

$$\mathcal{B} = \{ (0,0.1); (0.1,0); (0.9,0); (1,0.1); (0,0.9); (0.1,1); (0.9,1); (1,0.9) \}$$

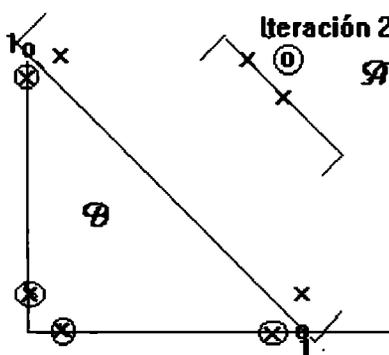
Como \mathcal{A} tiene menor cardinalidad que \mathcal{B} , se elige ResolverA. Los hiperplanos se encuentran en tres iteraciones, en la primer iteración, alfa tiene valor negativo muy chico, y el otro hiperplano es: $y = 0.1 - x$



En esta iteración queda clasificado un solo punto, $(0,0)$, perteneciente a \mathcal{A} . En la segunda iteración los planos son:

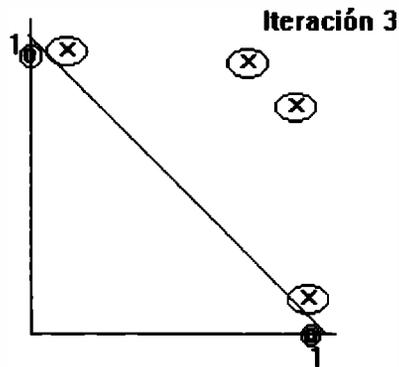
$$y = 1 - x \quad y = 1.9 - x$$

y quedan clasificados los puntos $(0.1,0); (0,0.1); (0.9,0); (0,0.9); (1,1)$.



En la tercer iteración clasificamos el resto de los puntos, que quedaban separables, y el plano es:

$$y = 1.05 - x$$

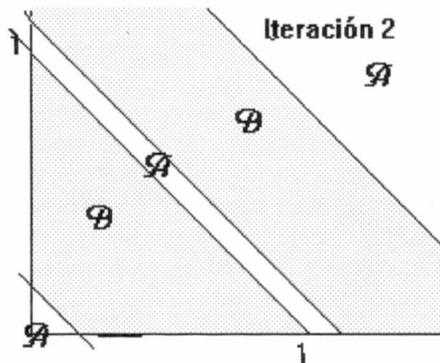


Es decir, que luego de ejecutar el programa, la salida que se obtuvo fue:

Primera iteración: $x^t c = -1.7e38$ $c = (-1, -1)$
 $x^t c = -0.1$

Segunda iteración: $x^t c = 1$ $c = (1, 1)$
 $x^t c = 1.9$

Tercer iteración: $x^t c = 1.05$ $c = (1, 1)$



Caso 4) Tres conjuntos no separables

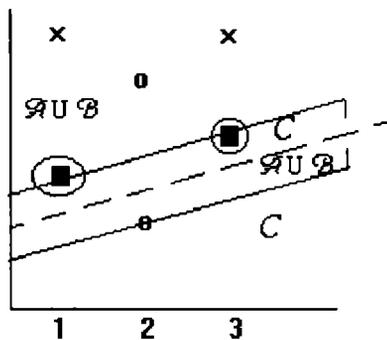
$$\mathcal{A} = \{ (1,6); (3,6) \} \quad \mathcal{B} = \{ (2,2); (2,5) \} \quad \mathcal{C} = \{ (1,3); (3,4) \}$$

Como los tres conjuntos tienen la misma cardinalidad, da lo mismo cualquier combinación de conjuntos. En este caso se elige $A = (\mathcal{A} \cup \mathcal{B})$ y $B = \mathcal{C}$. En el primer paso los planos se encuentran en dos iteraciones, en la primera:

$$y = x/2 + 2.5 \quad y = x/2 + 1$$

En la segunda iteración, los conjuntos quedan separables y el plano separador es $y = x/2 + 1.75$

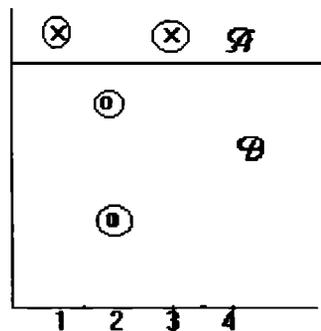
Pasada 1 (2 iteraciones)



Después de la primera pasada quedan clasificados todos los puntos de \mathcal{C} , en la segunda pasada, los conjuntos \mathcal{A} y \mathcal{B} son separables, y el plano separador es:

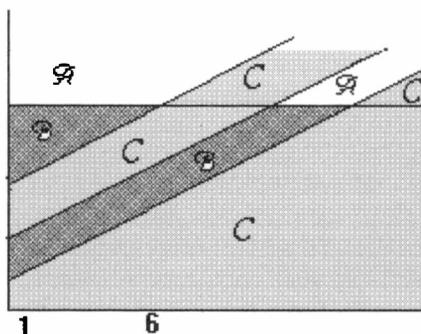
$$y = 5.5$$

Pasada 2 (1 iteración)



Así quedan clasificados los puntos de \mathcal{A} y \mathcal{B}

Resultado Final



La salida que se obtuvo fue:

Pasada Uno:

Primer iteración: $x^t c = -2.5$ $c = (\frac{1}{2}, -1)$
 $x^t c = -1$

Segunda iteración: $x^t c = 1$ $c = (1, 1)$
 $x^t c = 1.9$

Tercer iteración: $x^t c = 1.05$ $c = (1, 1)$

Ejemplo:

Para asignar un nuevo punto , por ejemplo $X = (6,5.6)$, cercano a la intersección de las tres regiones, el algoritmo de separación decide que $X \in \mathcal{A}$.

Potencialidad

Para medir el número de iteraciones necesarios para hallar los hiperplanos de separación, se generaron 15 puntos al azar para \mathcal{A} , generados por la función random de Pascal, y 15 puntos para \mathcal{B} , en \mathbb{R}^5 .

Se ejecutó el programa Calcular de separación y se obtuvieron los siguientes resultados:

Dimensión en el espacio	5
Cantidad de puntos de \mathcal{A}	15
Cantidad de puntos de \mathcal{B}	15
Cantidad de pasadas	20
Cantidad mínima de iteraciones	2
Número de veces que ocurrieron	3
Cantidad máxima de iteraciones	4
Número de veces que ocurrieron	5
Cantidad promedio de iteraciones	3.1 ~ 3

La dimensión y cantidad de puntos en cada conjuntos está restringida por la potencialidad del soft para resolver problemas lineales empleado. El cambio a un soft más potente es de fácil implementación. (Para más detalles ver capítulo Etapa de Implementación).

Aplicación Real

Los siguientes datos provienen del problema recibido por el Laboratorio de Matemática Aplicada de la Facultad de Ciencias Exactas de la Universidad Nacional de La Plata. El problema a resolver es el siguiente:

Para cada paciente conectado a un respirador artificial, se tienen datos basales, a los cinco minutos y a los treinta minutos, y de acuerdo a su valor se quiere decidir si debe sacarse o no el respirador.

La muestra clasifica en:

\mathcal{A} = éxito = conjunto de personas que después de evaluar el valor de los parámetros se decidió sacar el respirador y no fue necesario volver a conectar.

\mathcal{B} = fracaso

Se desea determinar una función que separe a dichos conjuntos, con el objetivo de poder tomar una determinación sobre los nuevos pacientes, basado en la experiencia acumulada con los datos tomados.

Se realizó un test estadístico para determinar si cada uno de los parámetros era significativo o no. Para restringir el número de variables a usar, se procesaron solo los datos significativos y se halló que \mathcal{A} y \mathcal{B} eran separables, es decir que un **análisis conjunto** de todos los parámetros de un mismo individuo podía realizarse y determinar la conveniencia o no de sacar el respirador.

Se consideraron trece parámetros, los datos fueron los siguientes:

Éxitos:

	Edad	Sexo	fc	fr	vme	vmecalc	vtti5	fr5	vt5	vt5c	p01vtti	pimax5	p01pimax5
Pac1	58	1	96	14	11.2	11.2	250	25	300	0	625	40	6.25
Pac2	44	1	90	12	9.6	9.6	278	24	300	0	556	30	6.67
Pac3	60	0	75	10	5	5	221	18.7	300	0	994.5	50	9
Pac4	21	1	105	14.7	11	11.03	435	27.7	400	1	1305	60	5
Pac5	59	1	104	14	9.8	9.8	417	28	450	1	1459	40	8.75
Pac6	24	0	90	12	0	8.4	417	21	400	1	3336	80	10

Fracasos:

Pac7	56	0	84	19.5	12.6	12.68	208.3	21.1	250	0	999.84	50	9.6
Pac8	21	1	140	22	18.71	18.7	583.3	46.8	350	1	5249.7	60	15
Pac9	63	1	120	21.5	16.1	16.12	548	30.9	460	1	3101.68	50	11.32
Pac10	57	1	116	10	6.75	6.75	297	24	356	1	742.5	30	8.33
Pac11	74	0	110	14	9.25	9.52	223	27	250	0	892	38	10.53
Pac12	64	1	104	12	8.4	8.4	772	27	525	1	9264	18	66.67

Lo que equivale a :

$$\mathcal{A} = \{ (58,1,96,14,11.2,11.2,250,25,300,0,625,40,6.25) \\ (44,1,90,12,9.6,9.6,278,24,300,0,556,30,6.67) \\ (60,0,75,10,5,5,221,18.7,300,0,994.5,50,9) \\ (21,1,105,14.7,11,11.03,435,27.7,400,1,1305,60,5) \\ (59,1,104,14,9.8,9.8,417,28,450,1,1459,40,8.75) \\ (24,0,90,12,0,8.4,417,21,400,1,3336,80,10) \}$$

$$\mathcal{B} = \{ (56,0,84,19.5,12.6,12.68,208.3,21.1,250,0,999.84,50,9.6) \\ (21,1,140,22,18.71,18.7,583.3,46.8,350,1,5249.7,60,15) \\ (63,1,120,21.5,16.1,16.12,548,30.9,460,1,3101.68,50,11.32) \\ (57,1,116,10,6.75,6.75,297,24,356.1,742.5,30,8.33) \\ (74,0,110,14,9.25,9.52,223,27,250,0,892,38,10.53) \\ (64,1,104,12,8.4,8.4,772,27,525,1,9264,18,66.67) \}$$

La ecuación del hiperplano separador es:

$$-x_1 + x_2 - x_3 - x_4 - x_5 + x_6 + 0.208261 x_7 + x_8 + 0.086414 x_9 - 0.045766 x_{10} + x_{11} - x_{12} - x_{13} = -68.779225$$

Es decir, basados en la experiencia anterior, podemos determinar sobre los valores de los parámetros de un nuevo individuo, la conveniencia o no de sacar el respirador con la siguiente regla:

Si $x = (x_1, x_2, \dots, x_{13})$ y la ecuación evaluada en el punto x es mayor o igual a -68.779225 , entonces x pertenece al conjunto \mathcal{A} (éxitos) y se espera que el respirador pueda ser retirado sin inconvenientes, caso contrario pertenece al conjunto \mathcal{B} (fracasos) y se recomienda dejar el respirador.

Explicación del Uso del Software

Para ejecutar el software, una vez posicionado sobre el directorio (DIR) que lo contiene, hay que tipear:

```
DIR> calcular
```

Lo primero que se deberá indicar es si los datos se ingresarán por archivo o por pantalla, y la cantidad de conjuntos a separar. Si los datos se ingresan por archivo, éste deberá tener el formato que se indica en el ítem siguiente. Si los datos se ingresan por pantalla, el programa pedirá los datos de cada uno de los conjuntos. Una vez que halla finalizado su ejecución, habrá guardado en el archivo `planos.sep` los pares de hiperplanos encontrados en cada iteración.

Para clasificar nuevos puntos, sobre el mismo directorio (DIR), ingresar:

```
DIR> separar
```

El programa pedirá la cantidad de conjuntos que fueron separados y luego los datos del punto a clasificar y dará como respuesta el conjunto al que pertenece.

Formato del Archivo de Datos

El archivo de datos deberá tener cada dato en una línea, primero se deberá indicar la dimensión en la que se trabajará (ej: si es \mathbb{R}^2 , se pondrá 2), luego se debe indicar la cantidad de puntos del conjunto \mathcal{A} , e ingresar cada punto de \mathcal{A} , colocando el valor de cada coordenada en una línea. Luego se ingresa la cantidad de puntos de \mathcal{B} y los puntos. Si fuesen tres conjuntos, se haría lo mismo con el tercer conjunto. Por ejemplo:

$$\mathcal{A} = \{ (1,1) ; (2,2) ; (3,1) \} \quad \mathcal{B} = \{ (2,1.5); (3,2) \}$$

El archivo de datos será:

```
2    dimensión en el espacio
3    cantidad de puntos de  $\mathcal{A}$ 
1    punto 1
1
2    punto 2
2
3    punto 3
1
2    cantidad de puntos de  $\mathcal{B}$ 
2    punto 1
1.5
3    puntos 2
2
```

APÉNDICE

Instancia: Cada caso particular de un problema

Tamaño de la instancia: Número de caracteres requeridos para representar la instancia.

Problemas P: Clase de problemas que pueden ser resueltos polinomialmente. (Si el tamaño de la instancia es l , existe un polinomio $p(l)$ tal que dicha instancia puede ser resuelta por el algoritmo, en a lo sumo $p(l)$ pasos elementales).

Problemas NP: Clase de problemas para los cuales una respuesta "SI" puede ser verificada en tiempo polinomial, cuando una información adicional ha sido especificada. (Si tenemos un $c_0 \in \mathbb{R}^n$, la determinación de α y β , y la verificación del signo de $\alpha - \beta$ se puede hacer en tiempo polinomial). La verificación se hace con algoritmos nodeterminísticos en tiempo polinomial. $P \subseteq NP$.

Problemas NP-Completos: Clase de problemas para los que están en NP y mostrar que están en P implicaría que $P = NP$. (Un algoritmo polinomialmente acotado para resolverlo puede ser usado como una subrutina para obtener un algoritmo polinomialmente acotado para cada problema en NP) .

Problemas NP-Hard: Un problema p_0 pertenece a NP-Hard si y solo si para todo problema p perteneciente a NP, $p \alpha p_0$. $NP\text{-Completo} \subseteq NP\text{-Hard}$.

α : Sean p_1 y p_2 problemas, $p_1 \alpha p_2$ si y solo si existe una función total computable 'f' tal que w pertenece a p_1 si y solo si $f(w)$ pertenece a p_2 . (De un problema desconocido pasamos a uno conocido, puede ser que sea más fácil ver si $f(w)$ pertenece a p_2 que ver que w pertenece a p_1).

Problema Dual: Asociado a cada problema de Programación Lineal (Primal), tenemos el correspondiente problema Dual de Programación Lineal. La relación de dualidad es simétrica en problemas expresados en términos de desigualdades, si el problema primal es de minimizar, el dual será de maximizar.

Ej: Primal

$$\begin{array}{ll} \min & c^t x \\ \text{st} & Ax \geq b \\ & x \geq 0 \end{array}$$

Dual

$$\begin{array}{ll} \max & y^t b \\ \text{st} & y^t A \geq c^t \\ & y \geq 0 \end{array}$$

Teorema de Dualidad: Si alguno de los problemas tiene solución óptima finita, el otro también, y los valores son los mismos. Si en alguno la solución no es acotada, no hay solución factible.

Bibliografía

- _ O.L. Mangasarian, Kristin Bennett, *Bilinear Separation of two Sets in n-Space*.
- _ O.L. Mangasarian, R. Setiono, W.H. Wolberg, *Pattern Recognition Via Linear Programming: Theory and Application to Medical Diagnosis*
- _ O.L. Mangasarian, *Multisurface Method of Pattern Separation*.
- _ G.L. Newhauser, A.H.G. Rinnooy Kan, M.J. Todd (Editors) *OPTIMIZATION (Vol 1)*.
- _ Manual de LINDO
- _ Manual de Turbo Pascal 6.0