

**UNIVERSIDAD CENTROCCIDENTAL  
“LISANDRO ALVARADO”  
Decanato de Ciencias y Tecnología  
Coordinación de postgrado.**



***Núcleos basados en polinomios ortogonales  
clásicos y máquinas de vectores de soporte***

**Trabajo Especial de Grado presentado por  
Ifigenia Yeniree Romero Arenas**

**Como requisito final para obtener el título de  
Maestría en Ciencias-Mención Optimización**

**Área de Conocimiento: Matemática aplicada.  
Tutor: Dr. Javier Hernández Benítez.**

Barquisimeto - Venezuela

Octubre 2016

# **Núcleos basados en polinomios ortogonales clásicos y máquinas de vectores de soporte**

Ifigenia Y. Romero A.

## **RESUMEN**

En este trabajo se implementará el uso de núcleos polinómicos basados en familia de polinomios ortogonales clásicos (Jacobi, Laguerre, Hermite y Bessel) en las máquinas de vectores de soporte para transformar datos no separables por un hiperplano en un espacio que sí lo son y se estudiará su influencia en distintos datos experimentales con miras a determinar los beneficios y dificultades de usar estos tipos de núcleos en las SVM

*A Dios ser supremo que le debo lo que soy y seré, es mi guía espiritual y a quien le pido ilumine mi camino siempre. A mi abuela María que desde niña con mucho amor y cariño me guió por el camino correcto y hay desde el cielo comparte este triunfo. A mi angel hermanita Mariangela que desde el cielo sigue llenando mi vida con su dulce sonrisa. A mis Padres, Eugenio y Ketty, quienes me dieron el ser y me ayudaron a realizar cada uno de mis sueños y me dieron la enseñanza para hoy honrarles. A mi segunda madre Maribel, por su incondicional apoyo en todo lo que hago, por su comprensión, paciencia y su amor, eres mi mejor ejemplo de constancia y superación.*

## Índice general

|   |           |
|---|-----------|
| <b>Índice de figuras</b>                        | <b>IV</b> |
| <b>Agradecimientos</b>                          | <b>VI</b> |
| <b>Introducción</b>                             | <b>2</b>  |
| <b>1. Máquinas de vectores de soporte (SVM)</b> | <b>6</b>  |
| <b>2. Función núcleo</b>                        | <b>19</b> |
| <b>3. Polinomios ortogonales clásicos</b>       | <b>29</b> |

---

|  |           |
|--|-----------|
| 3.1. Polinomios ortogonales . . . . .                                | 29        |
| 3.1.1. Fórmula de recurrencia . . . . .                              | 32        |
| 3.1.2. Ceros de los polinomios ortogonales . . . . .                 | 33        |
| 3.2. Polinomios ortogonales clásicos . . . . .                       | 34        |
| 3.2.1. Polinomios de Jacobi . . . . .                                | 35        |
| 3.2.2. Polinomios de Laguerre . . . . .                              | 39        |
| 3.2.3. Polinomios de Hermite . . . . .                               | 41        |
| 3.2.4. Polinomios de Bessel . . . . .                                | 43        |
| 3.3. Función núcleo de los polinomios ortogonales clásicos . . . . . | 45        |
| <b>4. Resultados numéricos</b>                                       | <b>48</b> |
| <b>Bibliografía</b>  | <b>80</b> |
| <b>A. Códigos en MATLAB</b>  | <b>84</b> |
| A.1. graficadepuntos.m . . . . .                                     | 84        |
| A.2. svmgeneral.m . . . . .  | 85        |

A.3. nucleomenu.m . . . . . 89

## Índice de figuras

|   |    |
|---|----|
| 1.1. Hiperplanos separadores . . . . .            | 7  |
| 1.2. Hiperplano separador óptimo . . . . .        | 8  |
| 1.3. SVM estandar . . . . .                       | 10 |
| 1.4. Margen máximo . . . . .                      | 11 |
| 1.5. SVM no separable por un hiperplano . . . . . | 14 |
| 1.6. Nucleo . . . . .                             | 18 |
| 4.1. TABLERO2 . . . . .                           | 49 |
| 4.2. TABLERO4 . . . . .                           | 55 |

**ÍNDICE DE FIGURAS**

---

v

|                         |    |
|-------------------------|----|
| 4.3. CIRCULO2 . . . . . | 61 |
| 4.4. CIRCULO4 . . . . . | 67 |
| 4.5. NUEVE . . . . .    | 73 |



## Agradecimientos

A Dios Por no permitir que los obstáculos que se me presentaron en el transcurso de esta maestría me vencieran.

A mis padres Eugenio y Ketty que con mucho cariño comprensión y apoyo guiaron mi camino por el bien. En especial a mi madre te admiro porque nunca te has dejado vencer, eres mi mejor ejemplo a seguir Te amo Madre.

A mi mami Maribel por su ayuda y sus enseñanzas y por no perder la fe en mí.

A mis Hermanos por acompañarme en cada uno de mis sueños, por comprenderme, ayudarme y apoyarme. Los amo hermanitos.

A mis tíos ejemplo de superación que estuvieron siempre ayudándome y animándome a no decaer.

A mis primos y sobrino que este triunfo sea un ejemplo para ellos.

A mi amiga Yesenia por estar conmigo en este largo camino, dándome sus consejos, animándome y ayudándome en momentos difíciles, como también a su familia por permitirme entrar a su hogar y brindarme su apoyo.

A mi tutor Javier Hernández quien con su valiosa enseñanza y paciencia contribuyo con la culminación de este trabajo, sin su ayuda y conocimiento no estaría donde me encuentro ahora.

A la Profesora Jurancy Ereú por su apoyo incondicional por su valiosa ayuda en la elaboración de este trabajo, por sus enseñanzas y sus buenos consejos.

A mis amigos Carlos, Emily, Francis, Marcos y Hector por sus buenos consejos, por apoyarme y ayudarme en momentos difíciles durante esta maestría y por compartir momentos únicos e inolvidables.

A todos los que colocaron su granito de arena para lograr este triunfo, y si olvide a alguien disculpen.

## Introducción

Las máquinas de vectores de soporte (SVM por sus siglas en inglés) son un conjunto de algoritmos de aprendizaje supervisado desarrollado originalmente por Vladimir Vapnik [17], estos métodos están propiamente relacionados con problemas de clasificación y regresión (ver [22] y [23]); la técnica de las SVM consiste en, dado un conjunto de puntos en un espacio, en el que cada uno de ellos pertenece a una de dos posibles categorías, un algoritmo basado en SVM construye un modelo capaz de predecir si un punto nuevo, cuya categoría desconocemos, pertenece a una categoría o a la otra. La SVM busca un hiperplano o conjuntos de hiperplanos que separen de forma óptima a los puntos de una clase de la de otra, que eventualmente han podido ser previamente proyectados a un espacio de dimensión superior o incluso infinita. Este tipo de algoritmos buscan el hiperplano que tenga la máxima distancia (margen) con los puntos que estén más cerca de él mismo [11]. Por eso las SVM se conocen como clasificadores de margen máximo.

Existen problemas donde los datos no pueden ser separados a través de un hiperplano óptimo, y esto ocurre muchas veces debido a que la dimensión del problema en el espacio de entrada es baja, sin embargo, existe la posibilidad de separar los puntos pero en un espacio de mayor dimensión y esto se logra mediante el uso de la función núcleo. Los núcleos son funciones que se emplean en las máquinas de soporte vectorial [7]; que consisten en un producto interno en el espacio de características que tiene su equivalente en el espacio de entrada [15], esta función está dada por:

$$K(x; x^T) = \langle \phi(x), \phi(x') \rangle,$$

donde  $k$  es una función simétrica definida positiva que cumple las condiciones de Mercer. Entre los núcleos más conocidos podemos mencionar el núcleo Gaussiano, núcleo Perceptron, núcleos sigmoide, núcleo polinómicos, entre otros.

En este trabajo se implementará el uso de núcleos polinómicos basados en polinomios ortogonales clásico en máquinas de vectores de soporte. La ventaja de usar los polinomios ortogonales es que nos permiten expresar funciones de un cierto espacio vectorial como combinaciones lineales de polinomios ortogonales. Aquí se trabajó específicamente con polinomios ortogonales clásicos pues tienen una gran cantidad de propiedades útiles tales como: satisfacen una relación de recurrencia de tres términos (TTRR), son la solución de una ecuación diferencial lineal de segundo orden, sus derivadas (o diferencias finitas) también constituyen una familia ortogonal, entre otros (ver [1] y [3]). Se han realizado estudios donde se observa la aplicación de los polinomios ortogonales con SVM entre estos podemos mencionar a Ye, N. y Sun, R. en el año 2006 estudiaron las SVM con el

núcleo ortogonal de Chebyshev (ver [20]), Ozer, Sedat y Chen en el 2011 estudiaron clasificación de patrones usando SVM con núcleo ortogonal de Chebyshev [12], además Zhao, J. y Yan, G. en el 2013 trabajaron con vectores soportes de regresión Basados en polinomios ortogonales clásicos específicamente de chebyshev [23], entre otras.

La estructura de este trabajo está diseñada de la forma siguiente: en el capítulo 1 se exponen la definición de las máquinas de vectores de soporte, cuando los datos son linealmente separables y cuando no lo son. Se introduce en la sección de caso linealmente separable la resolución analítica del problema y los pasos a llevar a cabo para su solución; mientras que la sección de caso no separable linealmente se expone la estrategia a seguir para poder separar los datos mediante el uso de funciones núcleo.

En el capítulo 2 estudiamos la función núcleo y se muestra en detalle conceptos básicos acerca de la función núcleo, núcleo reproductor y sus propiedades. También introducimos los Espacios de Hilbert con Núcleo Reproductor. Este concepto es fundamental en las Máquinas de vectores de soporte, debido a que estos espacios permitirán la clasificación de los datos, cuando éstos no son linealmente separables. Al final de este capítulo se incluye la condición de Mercer que nos permite establecer que condición debe cumplirse para que un producto interno sea un núcleo.

En el capítulo 3 abordamos la definición de polinomios ortogonales, además se describe los polinomios ortogonales clásicos tales como Jacobi, Laguerre, Hermite y a estas tres familias se le incluye el polinomio de Bessel, para cada uno de

estos polinomios se describe la relación de ortogonalidad, fórmula de Rodrigues, coeficiente principal, solución de una EDO y su fórmula de recurrencia. Estudiamos también una herramienta útil para obtener en forma explícita la función núcleo para cada una de estas familias de polinomios ortogonales esta es la Fórmula de sumación de Christoffel–Darboux ver [9] y como queda expresada esta fórmula para cada polinomio ortogonal clásico.

Finalmente el capítulo 4 muestra algunos ejemplos implementando estos núcleos para las máquinas de soporte vectorial, su influencia en diversos datos experimentales y los beneficios y dificultades de usar estos tipos de núcleos en las SVM, además de la estrategia empleada para llegar a esos resultados.

## Máquinas de vectores de soporte (SVM)

Las máquinas de vectores de soporte han sido introducidas como una poderosa herramienta para resolver problemas de clasificación, reconocimiento de una gran cantidad de puntos y problemas de regresión. Una SVM envía los puntos del espacio de entrada a un espacio de características de mayor dimensión y encuentra un hiperplano que separa los datos, además maximiza el margen entre las clases. Intuitivamente una SVM es un modelo que representa los puntos de una muestra en un espacio, separando estos puntos en clases y separándolos entre sí con la mayor distancia posible, es decir con un margen máximo. Maximizar el margen consiste en maximizar la distancia entre el hiperplano separador y el valor de entrada más cercano, que se conoce como el vector soporte (punto donde se apoya el margen máximo). Una SVM, constituye un modelo capaz de predecir si un nuevo punto (cuya categoría es desconocida) pertenece a una categoría o a la otra.

Es importante resaltar que la formulación de las máquinas de vectores soportes varía dependiendo de la naturaleza de los datos. Supongamos que tenemos un

conjunto de datos  $(x_1, y_1), (x_2, y_2) \dots (x_m, y_m) \in X \times \{\pm 1\}$ , donde  $X \subset \mathbb{R}^n$  es un conjunto no vacío llamado espacio de entrada. Los datos  $x_i$  son extraídos del conjunto  $X$  y los  $y_i$  son las denominadas etiquetas; acá tendremos una  $x \in X$  y queremos predecir un  $y \in \{\pm 1\}$  y esto se logra encontrando un hiperplano que separe ambas clases de puntos; es importante destacar que pueden existir infinitos hiperplanos que separen los datos.

En medio de todos los posibles hiperplanos que separan las clases, existe solo un hiperplano de separación óptima  $X \subset \mathbb{R}^n$ . Los vectores  $x_i$  asociados a los datos de una clase estarán a un lado del hiperplano, y los asociados a otra clase, en otro lado del hiperplano, ó lo que es lo mismo, existe un par  $(\omega, \gamma) \in \mathbb{R}^{n+1}$  tal que  $x_i^T \omega + \gamma > 0$  si el punto  $x_i$  esta en la clase positiva, y  $x_i^T \omega + \gamma < 0$  si el punto  $x_i$  esta en la clase negativa.(ver la figura 1.1 y 1.2)

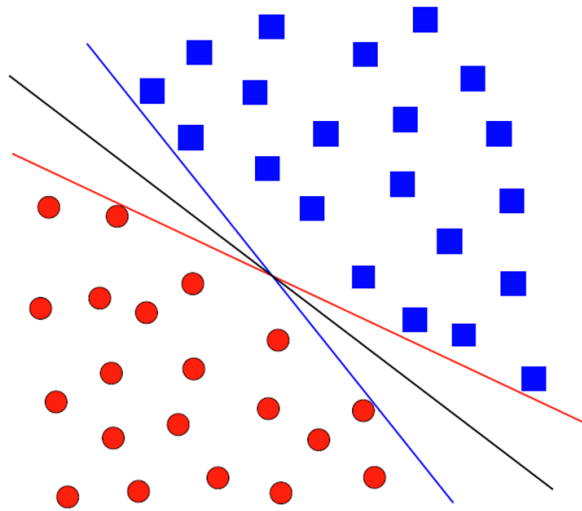


Figura 1.1: Hiperplanos separadores



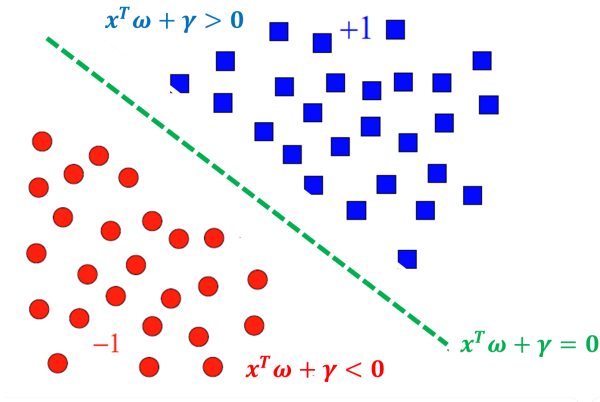


Figura 1.2: Hiperplano separador óptimo

**Definición 1.0.1 (Hiperplano Separador [13])** *Dados dos conjuntos  $C_1$  y  $C_2$ , subconjuntos de  $\mathbb{R}^n$  no vacíos, un hiperplano*

$$H = \{x \in \mathbb{R}^n / c^T x = \alpha; \text{ con } \alpha \in \mathbb{R}, c \in \mathbb{R}^n\}$$

*Separa los dos conjuntos  $C_1$  y  $C_2$  cuando se verifica que:*

$$x_1 \in C_1 \Rightarrow c^T x_1 \geq \alpha$$

$$x_2 \in C_2 \Rightarrow c^T x_2 \leq \alpha$$

**Teorema 1.0.2 [13]** *Sean  $C_1$  y  $C_2$  conjuntos no vacíos en  $\mathbb{R}^n$ . Entonces existe un hiperplano  $H$  que separa  $C_1$  y  $C_2$  propiamente si y sólo si existe un vector  $b$  tal*

que

$$\begin{aligned}\inf\{\langle x, b \rangle / x \in C_1\} &\geq \sup\{\langle x, b \rangle / x \in C_2\} \\ \sup\{\langle x, b \rangle / x \in C_1\} &> \inf\{\langle x, b \rangle / x \in C_2\}\end{aligned}$$

El hiperplano  $H$  separa a  $C_1$  y  $C_2$  fuertemente si y sólo si existe un vector  $b$  tal que

$$\inf\{\langle x, b \rangle / x \in C_1\} > \sup\{\langle x, b \rangle / x \in C_2\}$$

Como lo que se quiere es que la separación de los hiperplanos sea la mayor distancia posible, consideremos la distancia  $d_+$  (la distancia euclídea mas corta entre el hiperplano y la clase positiva mas cercana) y  $d_-$  (la distancia euclídea mas corta entre el hiperplano y la clase negativa mas cercana). Definiremos el margen de separación del hiperplano como  $d_+ + d_-$  (Observe la figura 1.2).

Si un hiperplano esta definido por un par  $(\omega, \gamma)$ , entonces cualquier par de la forma  $(\lambda\omega, \lambda\gamma)$  con  $\lambda > 0$ , define el mismo hiperplano, además, si  $x_i^T \omega + \gamma \geq 0$  para cierto par  $(\omega, \gamma)$  entonces va a existir  $(\omega_i, \gamma_i)$  que define el mismo hiperplano tal que  $x_i^T \omega_i + \gamma_i \geq 1$ . De manera similar se procede con la clase negativa.

Por tanto las condiciones pueden ser agregadas en el problema de optimización de la siguiente manera:

$$\begin{aligned}x_i^T \omega + \gamma &\geq 1 \quad \text{para } y_i = +1 \\ x_i^T \omega + \gamma &\leq 1 \quad \text{para } y_i = -1\end{aligned}\tag{1.0.1}$$

Esto es

$$y_i(x_i^T \omega + \gamma) - 1 \geq 0 \quad \forall i\tag{1.0.2}$$

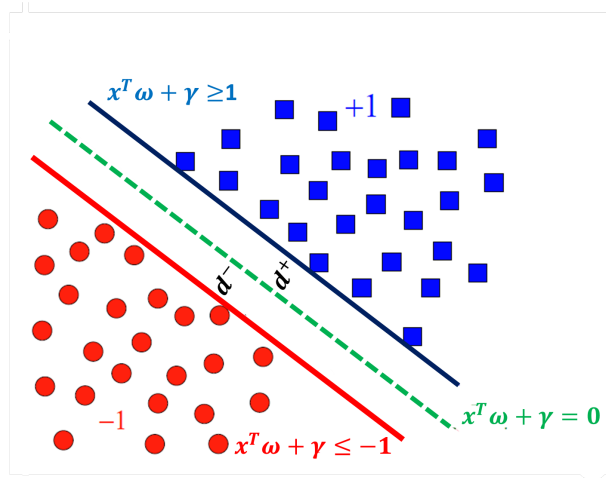


Figura 1.3: SVM estándar

Consideremos los puntos para los que se cumple la igualdad  $H_1 : x_i^T \omega + \gamma = 1$ , estos puntos están sobre el hiperplano  $H_1$  con vector normal  $\omega$  y la distancia del hiperplano al origen esta dada por  $\frac{|\gamma|}{\|\omega\|}$  donde  $\|\omega\|$  es la norma euclídea de  $\omega$ , análogamente se cumple para el hiperplano  $H_2 : x_i^T \omega + \gamma = -1$ , por tanto;  $d_+ = d_- = \frac{1}{\|\omega\|}$ .

Nótese que  $H_1$  y  $H_2$  son paralelos (poseen el mismo vector normal) y que no hay puntos de entrenamientos entre ellos (ver la figura 1.4).

Podemos en definitiva encontrar el par que da el margen máximo, y lo podemos hacer mediante la resolución de dicho problema de la forma

$$\begin{aligned} \text{máx}_{\omega, \gamma} \quad & \frac{2}{\|\omega\|} \\ \text{s.a.} \quad & y_i(x_i^T \omega + \gamma) - 1 \geq 0, \quad i = 1, \dots, m, \end{aligned} \quad (1.0.3)$$

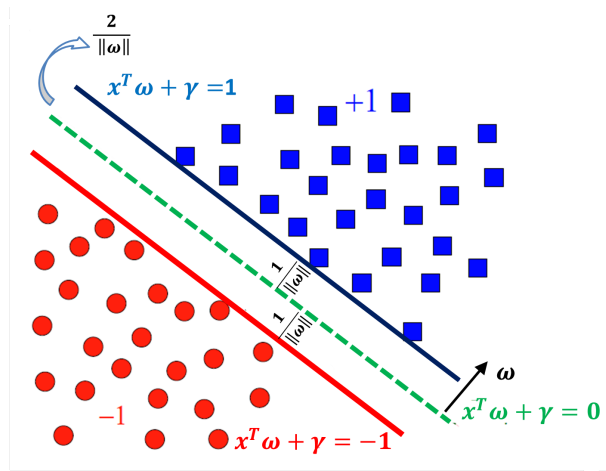


Figura 1.4: Margen máximo

Este problema es equivalente a minimizar la norma euclídea de  $\omega$ , de esta manera el problema de optimización a resolver nos queda:

$$\begin{aligned} \underset{\omega, \gamma}{\text{mín}} \quad & \frac{\|\omega\|^2}{2} \\ \text{s.a.} \quad & y_i(x_i^T \omega + \gamma) - 1 \geq 0, \quad i = 1, \dots, m \end{aligned} \quad (1.0.4)$$

Resolviendo este problema se obtiene un vector óptimo  $\omega^*$  que permite calcular el margen máximo de un hiperplano de separación.

Este problema puede ser resuelto usando el Dual de Wolfe, introduciendo los multiplicadores de Lagrange, es decir, que las restricciones de (1.0.4) se sustituirán por multiplicadores de Lagrange que serán más fáciles de manejar.[10]

El problema anterior puede tener o no solución, en el caso en que tenga solución este problema sería cuadrático convexo y tendría una única solución óptima

global. Podemos obtener la solución encontrando las condiciones de optimalidad de primer orden de **KKT** (Karush -Kuhn-Tucker),[13] para esto hacemos uso del Lagrangiano, introduzcamos  $m$  multiplicadores de Lagrange que denotaremos por  $\alpha_1, \dots, \alpha_m$  uno por cada restricción de (1.0.4). De esta manera lo anterior nos lleva a la siguiente función Lagrangiana

$$L_p = \frac{\|w\|^2}{2} - \sum_{i=1}^m \alpha_i (y_i (x_i^T w + \gamma) - 1), \quad (1.0.5)$$

luego,

$$L_p = \frac{\|w\|^2}{2} - \sum_{i=1}^m \alpha_i y_i (x_i^T w + \gamma) + \sum_{i=1}^m \alpha_i, \quad (1.0.6)$$

Así las condiciones de KKT para el problema (1.0.4) viene dadas por:

$$\begin{aligned} \frac{\partial L_p}{\partial w_j} &= 0, \\ w_j - \sum_{i=1}^m \alpha_i y_i x_i^T &= 0, \end{aligned} \quad (1.0.7)$$

$$\begin{aligned} \frac{\partial L_p}{\partial \gamma} &= 0, \\ - \sum_{i=1}^m \alpha_i y_i &= 0, \end{aligned} \quad (1.0.8)$$

$$y_i (x_i^T w + \gamma) - 1 \geq 0, \quad (1.0.9)$$

$$\alpha_i \geq 0, \quad (1.0.10)$$

$$\alpha_i (y_i (x_i^T w + \gamma) - 1) = 0, \quad (1.0.11)$$

Como se debe minimizar  $L_p$  con respecto a  $\omega$  y  $\gamma$  podemos resolver en su forma equivalente con su problema Dual. Aplicando dualidad lagrangiana tenemos que

$$L_D = \sum_{i=1}^m \alpha_i - \frac{1}{2} \sum_{i=1}^m \sum_{j=1}^m \alpha_i \alpha_j y_i y_j (x_i^T x_j) \quad (1.0.12)$$

Luego, su problema Dual viene dado por:

$$\begin{aligned} \max_{\alpha} \quad & \sum_{i=1}^m \alpha_i - \frac{1}{2} \sum_{i=1}^m \sum_{j=1}^m \alpha_i \alpha_j y_i y_j (x_i^T x_j) \\ \text{s.a.} \quad & \sum_{i=1}^m \alpha_i y_i = 0. \\ & \alpha_i \geq 0 \quad i = 1, \dots, m. \end{aligned} \quad (1.0.13)$$

En definitiva estamos en presencia de un problema de programación cuadrática convexa, con restricciones lineales. La solución para este problema consiste en obtener los valores óptimos para los multiplicadores de Lagrange  $\alpha_i$ , que no es mas que encontrar  $(\omega^*, \gamma^*)$ , que define el hiperplano de separación óptima.

Nótese que hay un multiplicador de Lagrange para cada muestra de entrenamientos. Además, tras obtener una solución aquellos puntos donde  $\alpha_i > 0$ , con  $i = 1, \dots, n$ , se denominan los vectores soporte y yacen sobre los hiperplanos  $H_1, H_2$ . El resto de los datos tienen  $\alpha_i = 0$  y cumplen con

$$y_i(x_i^T w + \gamma) - 1 \geq 0 \quad \forall i$$

Así  $\omega$  se escribirá como combinación lineal de los vectores soportes, los mismos son los elementos críticos del conjunto de entrenamiento, son lo mas cercano a la frontera de decisión, y si el resto de los puntos no se consideran en un nuevo entrenamiento, el algoritmo encontraría el mismo hiperplano de separación.

En el caso en que problema no tenga solución es decir, los datos no pueden ser separados a través de un hiperplano, Esto es, no se llega a una solución factible pues no existirá un hiperplano que cumpla con la condición  $y_i(x_i^T w + \gamma) - 1 \geq 0$  es por ello, que nos interesa poder relajar las restricciones, pero únicamente cuando sea necesario, para ello añadimos un nuevo coste a la función objetivo.(ver la figura 1.5)

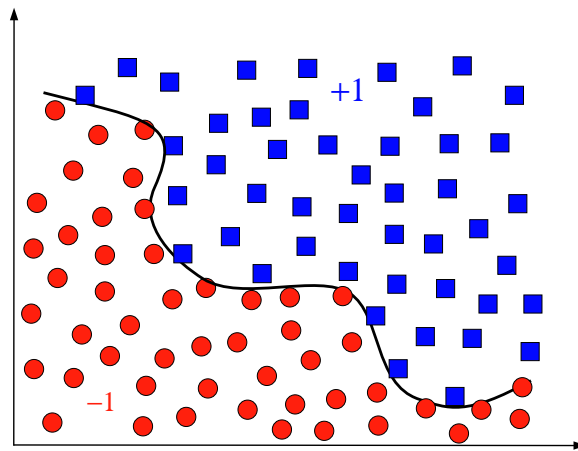


Figura 1.5: SVM no separable por un hiperplano

Se añadirá una penalización como consecuencia de los patrones mal clasificados. Una posible forma de hacerlo es introducir variables de holgura  $\xi_i$  con

$i = 1, \dots, m$  en las restricciones para convertirlas en;

$$\begin{aligned} x_i^T \omega + \gamma &\geq +1 - \xi_i & \text{para } y_i = +1 \\ x_i^T \omega + \gamma &\leq -1 + \xi_i & \text{para } y_i = -1 \\ \xi_i &\geq 0 \end{aligned} \quad (1.0.14)$$

esto es;

$$\begin{aligned} y_i(x_i^T \omega + \gamma) &\geq 1 - \xi_i \\ \xi_i &\geq 0 \end{aligned} \quad \text{para todo } i \quad (1.0.15)$$

Ahora, podemos preguntarnos como es posible aceptar puntos mal clasificados por el hiperplano; para ello penalizamos el error de clasificación a través una combinación lineal con la función objetivo, es decir, sustituir esta expresión

$$\min_{\omega, \gamma} \frac{\|\omega\|^2}{2} \quad (1.0.16)$$

por la siguiente expresión

$$\min_{\omega, \gamma, \xi} \frac{\|\omega\|^2}{2} + C \left( \sum_{i=1}^m \xi_i \right) \quad (1.0.17)$$

donde  $C$  es un parámetro ajustable; note que, si  $C$  tiene un valor muy alto equivale a una mayor penalización de los errores. Para patrones que son correctamente clasificados  $\xi_i = 0$ . Ahora bien, la noción de suavizar el margen permite tratar datos mas realistas; es por ello, que el problema anterior se reduce a un nuevo problema de programación cuadrática convexa con la agradable propiedad de que tanto los  $\xi_i$  como los multiplicadores de lagrange asociados a esta variable no



aparecen en el problema Dual.

$$\begin{aligned}
 \text{máx}_{\alpha} \quad L_D &= \sum_{i=1}^m \alpha_i - \frac{1}{2} \sum_{i=1}^m \sum_{j=1}^m \alpha_i \alpha_j y_i y_j (x_i x_j) \\
 \text{s.a} \quad &\sum_{i=1}^m \alpha_i y_i = 0, \\
 &0 \leq \alpha_i \leq C, \quad i = 1, \dots, m,
 \end{aligned} \tag{1.0.18}$$

la solución viene dada por:

$$\omega = \sum_{i=1}^{N_s} \alpha_i y_i x_i \tag{1.0.19}$$

donde  $N_s$  es el número de vectores soportes. Luego, la solución óptima puede escribirse como una combinación lineal de vectores soporte. Por lo tanto, la única diferencia con el caso separable del hiperplano óptimo es que los valores de  $\alpha_i$  están ahora acotados superiormente por  $C$ .

Necesitaremos las condiciones Karush-Kuhn-Tucker para el problema primal.

El Lagrangiano primal es,

$$L_p = \frac{\|\omega\|^2}{2} + C \sum_i \xi_i - \sum_i \alpha_i [y_i (x_i^T \omega + \gamma) - 1 + \xi_i] - \sum_i \mu_i \xi_i, \tag{1.0.20}$$

Pues los  $\mu_i$  representan los multiplicadores de Lagrange asociados a la condición  $\xi_i \geq 0$ .

Las condiciones KKT para el problema primal son por lo tanto,

$$\frac{\partial L_p}{\partial \omega_v} = \omega_v - \sum_i \alpha_i y_i x_{iv} = 0,$$

$$\begin{aligned}\frac{\partial L_p}{\partial \gamma} &= - \sum_i \alpha_i y_i = 0, \\ \frac{\partial L_p}{\partial \gamma} &= - \sum_i \alpha_i y_i = 0, \\ y_i(x_i^T \omega + \gamma) - 1 + \xi_i &\geq 0, \\ \alpha_i[y_i(x_i^T \omega + \gamma) - 1 + \xi_i] &= 0, \\ \mu_i \xi_i &= 0, \\ \xi_i \geq 0, \quad \alpha_i \geq 0, \quad \mu_i &\geq 0.\end{aligned}$$

En muchas situaciones, los datos pueden ser separados linealmente, a través de una transformación no lineal del espacio de entrada en un espacio de mayor dimensión llamado espacio de características. La transformación de los datos de un espacio inicial a otro de mayor dimensión se logra mediante, el uso de una función núcleo. En este sentido una función núcleo es un producto interno en el espacio de las características que tiene su equivalente en el espacio de entrada y esta dada por;

$$k(x, x^T) = \langle \Phi(x), \Phi(x^T) \rangle, \quad (1.0.21)$$

donde  $k$  es una función simétrica definida positiva que cumple con las condiciones de Mercer. A continuación mostramos el problema de optimización a resolver para las SVM con un núcleo general el cual esta definido por un problema de

programación cuadrática con restricciones dado por:

$$\begin{aligned}
 \max_{\alpha} \quad L_D &= \sum_{i=1}^m \alpha_i - \frac{1}{2} \sum_{i=1}^m \sum_{j=1}^m \alpha_i \alpha_j y_i y_j k(x_i, x_j) \\
 \text{s.a} \quad & \sum_{i=1}^m \alpha_i y_i = 0 \\
 & 0 \leq \alpha_i \leq C \quad i = 1, \dots, m,
 \end{aligned} \tag{1.0.22}$$

De manera gráfica se puede observar en la figura 1.5 como la función núcleo permite realizar la separación y traslado de los datos al espacio de características.

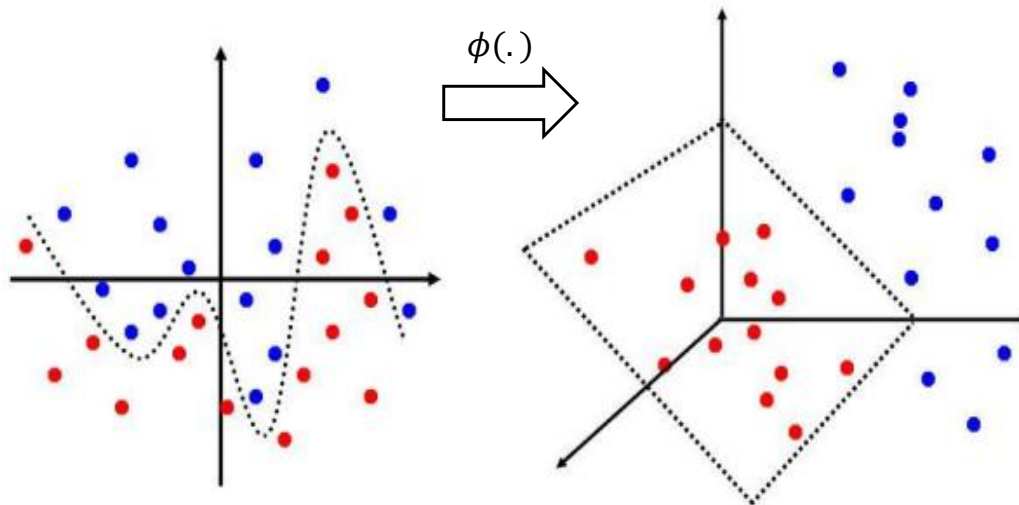


Figura 1.6: Nucleo

## Función núcleo

Las funciones núcleo, son funciones que se emplean en las máquinas de soporte vectorial (SVM). Estas funciones son las que permiten convertir la información de un espacio de entrada a un nuevo espacio de mayor dimensión llamado espacio de características. Generalmente las funciones núcleo se usan cuando el problema no puede ser separado a través de un hiperplano, y esto ocurre muchas veces debido a que la dimensión del problema en el espacio de entrada es baja, para ello es necesario proyectar los datos originales a un espacio de dimensión superior o incluso infinita a través de una función  $\phi$ , donde los datos sí pueden ser separados a través de un hiperplano. No cualquier función puede ser una función núcleo, ella debe satisfacer la condición de Mercer ver [15].

A continuación describiremos de forma detallada una función núcleo y sus propiedades y la utilidad del teorema de Mercer.

**Definición 2.0.3 (Mapa de características)** *La función  $\Phi$  que traslada los ele-*

mentos del espacio  $\mathbb{X}$  al espacio de Hilbert  $\mathbb{H}$

$$\begin{aligned}\Phi : \mathbb{X} &\longrightarrow \mathbb{H} \\ x &\longrightarrow \Phi(x)\end{aligned}\tag{2.0.1}$$

se denomina mapa de características

**Definición 2.0.4 (Función núcleo ver [15])** *El producto escalar definido en el espacio  $\mathbb{K}$  se denomina función núcleo. Por tanto podemos reescribir el producto escalar del espacio, el núcleo  $k$  en función del mapa  $\Phi$*

$$\begin{aligned}k : \mathbb{X} \times \mathbb{X} &\longrightarrow \mathbb{K} \\ (x, x') &\longrightarrow k(x, x') = \langle \Phi(x), \Phi(x') \rangle_{\mathbb{H}}\end{aligned}\tag{2.0.2}$$

donde  $\mathbb{K} = \mathbb{C}$  o  $\mathbb{K} = \mathbb{R}$

Algunas propiedades que debe cumplir la función núcleo para definir un espacio de características son las siguientes:

1. Simetría  $k(x, x') = k(x', x)$
2. Desigualdad de Cauchy-Schwarz  $k(x, x')^2 \leq k(x, x) \cdot k(x', x')$

Ahora vamos a imponer algunas condiciones de la función  $k$  y antes de introducir la definición del núcleo reproductor, daremos dos teoremas importantes en la teoría de análisis funcional como lo es el el Primer Teorema de Representación

de Riesz, este Teorema será de particular interés, debido a que es la base de la Propiedad Reproductiva; el cual nos permite tener una representación explícita y única de funcionales lineales acotados y el Segundo Teorema de Representación de Riesz, nos permite representar de forma única una función sesquilineal acotada, como un producto interno sobre un espacio de Hilbert.

**Teorema 2.0.5 (Primer Teorema de Representación de Riesz [14])** . *Todo funcional lineal acotado  $f$  en un espacio de Hilbert  $H$  puede ser representado como*

$$f(x) = \langle x, z \rangle \quad (2.0.3)$$

*de forma única donde  $z$  depende de  $f$  mas aun, la  $\|f\| = \|z\|$*

**Definición 2.0.6** *Sean  $X$  y  $Y$  dos espacios vectoriales sobre el mismo cuerpo  $K$ . Una forma sesquilineal es una aplicación  $h : X \times Y \rightarrow K$  tal que para todo  $x, x_1, x_2 \in X$ , todo  $y, y_1, y_2 \in Y$  y todo  $\alpha, \beta \in K$*

1.  $h(\alpha x_1 + \beta x_2, y) = \alpha h(x_1, y) + \beta h(x_2, y)$
2.  $h(x, \alpha y_1 + \beta y_2) = \bar{\alpha} h(x, y_1) + \bar{\beta} h(x, y_2)$

**Teorema 2.0.7 (Segundo Teorema de Representación de Riesz [14])**  $\mathbb{H}_1$  y  $\mathbb{H}$  espacios de Hilbert y  $h : \mathbb{H}_1 \times \mathbb{H} \rightarrow K$  una forma sesquilineal acotada. Entonces, existe un único operador lineal acotado  $S : \mathbb{H}_1 \rightarrow \mathbb{H}$  tal que

$$h(x, y) = \langle S_x, y \rangle_{\mathbb{H}}$$

para todo  $x \in \mathbb{H}_1$  y todo elemento  $y \in \mathbb{H}$  mas aun  $\|h\| = \|S\|$

Introduciremos ahora Espacios de Hilbert con Núcleo Reproductivo. Este concepto es fundamental para las Máquinas vectoriales de soportes, debido a que estos espacios permitirán la clasificación de los datos, cuando éstos no son linealmente separables.

**Definición 2.0.8 (Núcleo Reproductor [16])** Sea  $\mathbb{X}$  un conjunto no vacío llamado también conjunto de índices y sea  $\mathbb{H}$  un espacio de Hilbert de funciones  $f : \mathbb{X} \rightarrow \mathbb{R}$  una función  $k : \mathbb{X} \times \mathbb{X} \rightarrow \mathbb{K}$  es un núcleo reproductor de un espacio de Hilbert  $\mathbb{H}$  si y solo si se cumple:

1.  $k(x, \cdot) \in \mathbb{H}$  para todo  $x \in \mathbb{X}$
2. Propiedad reproductora:  $\langle f, k(x, \cdot) \rangle_{\mathbb{H}} = f(x)$  para todo  $f \in \mathbb{H}$  y  $x \in \mathbb{X}$

El nombre de propiedad reproductora se debe al hecho de que el valor de la función  $f$  en el punto  $x$  es reproducido por el producto interno de  $f$  y  $k$  de esta definición esta claro que:

La siguiente proposición permite definir un tipo especial de espacio de Hilbert basado en núcleo reproductor.

**Proposición 2.0.9 [16]** Si  $k$  es un núcleo reproductor cumple que:

$$\langle k(x, \cdot), k(x', \cdot) \rangle = k(x, x') \quad \text{para todo } (x, x') \in \mathbb{X} \times \mathbb{X}$$

conociendo el mapa de características  $\Phi(x) = k(x, \cdot)$ , vemos que el núcleo reproductor es un núcleo.

**Definición 2.0.10 (Espacios de Hilbert con núcleo reproductor [16])** *Un espacio de Hilbert que posee un núcleo reproductor se denomina Espacio de Hilbert con Núcleo Reproductor (RKHS). Su mapa de características se denomina mapa canónico y se define en función del núcleo  $k$ ,*

$$\Phi(x) = k(x, \cdot)$$

El siguiente teorema es muy importante en la teoría de (RKHS)

**Teorema 2.0.11 (Moore Aronszajn) [16]** *Un Espacio de Hilbert con Núcleo Reproductor determina un único núcleo reproductivo.*

El uso de la función núcleo hace posible el envío de la información de entrada  $(x, x')$  al espacio de características  $(\Phi(x), \Phi(x'))$  de forma implícita y entrena a la máquina en dicho espacio. La única información necesaria para el entrenamiento es la matriz de Gram, Dicha matriz también es conocida como la matriz núcleo la cual se denota con la letra  $\mathbf{K}$ , a continuación daremos la definición de dicha matriz:

**Definición 2.0.12 (Matriz de Gram ver[15])** *:Dada una función  $k : \mathbb{X} \times \mathbb{X} \rightarrow \mathbb{K}$  y patrones  $x_1, x_2, \dots, x_m \in \mathbb{X}$ , la matriz  $K$  de orden  $m \times m$  con elementos:*

$$K_{ij} := k(x_i, x_j)$$



Es llamada matriz de Gram o matriz núcleo de  $k$  con respecto a  $x_1, x_2, \dots, x_m$

**Definición 2.0.13 (Matriz definida positiva[16])** : Una matriz  $K$  simétrica de orden  $m \times m$  que satisface

$$\sum_{ij} c_i \bar{c}_j K_{ij} \geq 0$$

para todo  $c_i \in \mathbb{K}$  se llama definida positiva (PD)

**Nota 2.0.14** Una matriz simétrica es definida positiva si y solo si todos sus valores propios son no negativos.

**Definición 2.0.15 (Núcleo definido positivo[16])** Sea  $\mathbb{X}$  un conjunto no vacío,  $k$  en  $\mathbb{X} \times \mathbb{X}$  y para todo  $x_1, x_2, \dots, x_m \in \mathbb{X}$  con  $m \in \mathbb{N}$  que da lugar a una matriz de Gram definida positiva es llamado núcleo definido positivo.

Ahora que tenemos los conocimientos básicos la pregunta natural es cuando una función  $k$  en  $\mathbb{X} \times \mathbb{X}$  es un núcleo reproductor, una caracterización de núcleos reproductores viene dada por el siguiente teorema

**Teorema 2.0.16 (ver[15])** Un núcleo reproductor en un espacio de Hilbert es una función definida positiva.

Para que sea de utilidad un núcleo reproductor en una SVM, este debe satisfacer cierta condición conocida en la literatura como es la condición de Mercer. El

mismo ha jugado un papel crucial en la composición de las SVM y proporciona información valiosa sobre la geometría de los espacios de funciones. Así mismo, éste provee las condiciones que debe cumplir una función  $k$  para ser un núcleo.

**Teorema 2.0.17 (Mercer [16])** *Suponga  $k \in \mathcal{L}_\infty(\mathbb{X}^2)$  es una función a valores reales simétrica tal que el operador integral*

$$T_k : \mathcal{L}_2(\mathbb{X}) \longrightarrow \mathcal{L}_2(\mathbb{X})$$

$$(T_k f)(x) := \int_{\mathbb{X}} k(x, x') f(x') d\mu(x')$$

*es definido positivo, es decir, para todo  $f \in \mathcal{L}_2(\mathbb{X})$  tenemos*

$$\int_{\mathbb{X}^2} k(x, x') f(x) f(x') d\mu(x) d\mu(x') \geq 0.$$

*Sean  $\psi_j \in \mathcal{L}_2(\mathbb{X})$  funciones propias ortonormales de  $T_k$  asociadas a los autovalores  $\lambda_j \geq 0$ . ordenados en forma no creciente. entonces:*

**i**  $(\lambda_j)_j \in l_1$

**ii**  $k(x, x') = \sum_{j=1}^{N_H} \lambda_j \psi_j(x) \psi_j(x')$

La condición de Mercer permite determinar las propiedades que debe cumplir un producto interno en el espacio de características para que se pueda escribir a través de un cierto núcleo; a continuación mostramos dos resultados que nos ayudan

**Proposición 2.0.18 (Mapa de núcleo de Mercer[16])** Si  $k$  es un núcleo que satisface la condición del teorema de Mercer podemos construir  $\phi$  tal que

$$\langle \phi(x), \phi(x') \rangle = k(x, x'), \forall x, x' \in \mathbb{X} \quad (2.0.4)$$

opcionalmente, dado  $\epsilon > 0$ , existe un mapeo  $\phi_n$  en un espacio  $n$ -dimensional con producto interno, tal que:

$$|k(x, x') - \langle \phi^n(x), \phi^n(x') \rangle| < \epsilon, \quad \forall x, x' \in \mathbb{X}. \quad (2.0.5)$$

**Teorema 2.0.19 (Condición de Mercer[16])** Existe una transformación  $\phi$  y una expansión en series  $k(x_i, x_j) = \sum \phi(x_i)\phi(x_j)$  si y solo si para cualquier  $g(x)$  con  $g \in \mathcal{L}_2$  se tiene

$$\int_{\mathbb{X}^2} k(x, y)g(x)g(y)dxdy \geq 0$$

equivalentemente;

$$\begin{bmatrix} k(x_1, x_1) & k(x_1, x_2) & \dots \\ k(x_2, x_1) & \ddots & \\ \vdots & & \end{bmatrix}$$

Es definida positiva.

**Nota 2.0.20** Utilizando la función núcleo no es necesario calcular explícitamente el mapa  $\phi : X \rightarrow H$  para el entrenamiento en el espacio de las características.

Tanto el núcleo de Mercer como el núcleo definido positivo pueden ser repre-

sentados como un producto interno en un espacio de Hilbert la siguiente proposición muestra el caso donde estos núcleos coinciden.

**Proposición 2.0.21** [16] Sea  $\mathbb{X} = [a, b]$  un intervalo compacto y sea  $k : [a, b] \times [a, b] \rightarrow \mathbb{C}$  continua entonces  $k$  es un núcleo definido positivo si y solo si

$$\int_a^b \int_a^b k(x, x') f(x) f(x') dx dx' \geq 0 \quad (2.0.6)$$

para cada función continua  $f : \mathbb{X} \rightarrow \mathbb{C}$

**Proposición 2.0.22 (continuidad del mapa de características[16])** Si  $X$  es un espacio topológico y  $k$  es un núcleo continuo definido positivo en  $\mathbb{X} \times \mathbb{X}$ , entonces existe un espacio Hilbert  $\mathbb{H}$  y un mapeo  $\phi : \mathbb{X} \rightarrow \mathbb{H}$  y un mapeo  $\phi : \mathbb{X} \rightarrow \mathbb{H}$  continuo tal que  $\forall x, x' \in \mathbb{X}$ , tenemos:

$$k(x, x') = \langle \phi(x), \phi(x') \rangle \quad (2.0.7)$$

Por otro lado, veamos a continuación una caracterización de los núcleos definidos positivos. En el caso que no se obtiene un núcleo definido positivo se puede relajar la condición de definido positivo y obtener otras clases de núcleos como son los condicionalmente definido positivo;

**Definición 2.0.23 (Matriz condicionalmente definida positiva[16])** Una matriz

$K$  simétrica de orden  $m \times m$  ( $m \geq 2$ ) tomando valores en  $\mathbb{K}$  que satisfice;

$$\sum_{i,j=1}^m C_i \overline{C_j} K_{ij} \geq 0 \quad \forall C_i \in \mathbb{K}, \quad \text{con } \sum_{i=1}^m C_i = 0.$$

es llamada condicionalmente definida positiva (CDP)

**Definición 2.0.24 (Núcleo Condicionalmente definido positivo[16])** Diremos que un núcleo  $k$  es núcleo condicionalmente definido positivo si la matriz de Gram es condicionalmente definida positiva para todo  $m \geq 2$ ,  $x_1, x_2, \dots, x_m \in \mathbb{X}$ .

**Proposición 2.0.25 (Construcción de Núcleo DP desde un núcleo CDP[16])** Sea  $x_0 \in \mathbb{X}$ , y sea  $k$  un núcleo simétrico en  $\mathbb{X} \times \mathbb{X}$  entonces

$$\tilde{k}(x, x') := \frac{1}{2}(k(x, x') - k(x, x_0) - k(x_0, x') + k(x_0, x_0)) \quad (2.0.8)$$

es definido positivo si y solo si  $k$  es condicionalmente definido positivo.

## Polinomios ortogonales clásicos

Uno de los motivos por los que los polinomios ortogonales son interesantes es que son un análogo infinito dimensional de las bases ortonormales de espacios de dimension infinita con producto escalar; es decir, nos permiten expresar funciones de un cierto espacio vectorial como combinaciones lineales de polinomios ortogonales, además se pueden usar funciones núcleos basadas en polinomios ortogonales y las ventajas de usar este tipo de núcleo en las SVM es que estos permiten construir un hiperplano de separación con menos vectores soportes posibles

### 3.1. Polinomios ortogonales

Los polinomios ortogonales son una clase de polinomios que forman una base ortogonal en el espacio de Hilbert. Estos polinomios aparecen sobretodo en la teoría de ecuaciones diferenciales y en la mecánica cuántica. Veamos la definición de estos polinomios.

Sea  $[a, b] \subseteq \mathbb{R}$ , donde  $-\infty \leq a < b \leq +\infty$ , contemplando, por tanto, también los intervalos no acotados. Sea  $\omega(x)$  una función peso en  $[a, b]$ , que verifica las siguientes propiedades:

- $\omega(x) > 0$  en  $(a, b)$ .
- $\omega(x)$  es integrable en el  $(a, b)$ , es decir  $\omega(x) \in \mathcal{L}_1(x)$ , esto es,  $\int_a^b \omega(x) dx < +\infty$ .

A partir de esta función peso se define un producto escalar para el espacio vectorial de los polinomios  $\mathbb{P}$ . Sean  $p, q \in \mathbb{P}$ ,

$$\langle p, q \rangle_\omega := \int_a^b p(x)q(x)\omega(x)dx, \quad (3.1.1)$$

y su correspondiente norma

$$\|p\|_\omega = \sqrt{\int_a^b p(x)^2\omega(x)dx}, \quad (3.1.2)$$

que es clave para definir los polinomios ortogonales.

**Definición 3.1.1 (Polinomios Ortogonales, ver [3])**  $\{p_n(x)\}_{n=0}^\infty$  es una sucesión de polinomios ortogonales con respecto del peso  $\omega$  si verifica:

1.  $\forall n \in \mathbb{Z}_+$ , se cumple que  $gr(p_n(x)) = n$ .

2.

$$\langle p_n, p_m \rangle_\omega = \int_a^b p_n(x)p_m(x)\omega(x)dx = \begin{cases} 0, & n \neq m; \\ \kappa_n, & n = m, \kappa_n \neq 0. \end{cases} \quad (3.1.3)$$

donde  $gr : \mathbb{P} \rightarrow \mathbb{N}$  y  $gr(p)$  es el grado del polinomio  $P$ . Además si se impone la restricción de que para todo  $n \in \mathbb{Z}_+$ , la norma de  $p_n(x)$  sea igual a uno ( $\|p_n\|_\omega = 1$ ) entonces  $\{p_n(x)\}_{n=0}^\infty$  se llama sucesión de polinomios ortonormales con respecto a  $\omega$ .

Esta definición es equivalente a la siguiente condición, debido a la estructura de espacio vectorial de  $\mathbb{P}_n$ , que admite como base  $\{p_n(x)\}_{k=0}^n$  :

$$\langle p_n, x^k \rangle_\omega = \int_a^b p_n(x)x^k\omega(x)dx = C_n\delta_{nk} = \begin{cases} C_n, & n = k; \\ 0, & n \neq k. \end{cases}, k = 0, 1, 2, \dots, n, \quad (3.1.4)$$

donde  $C_n \neq 0$  es una constante arbitraria. Esta última determina un sistema de ecuaciones con  $n$  ecuaciones y  $n + 1$  incógnitas que son los coeficientes del polinomio. Esto indica que estos últimos están determinados salvo una constante multiplicativa.

Ahora veamos el siguiente resultado, que garantiza la existencia y unicidad de sucesiones de polinomios ortogonales asociados a una medida.

**Teorema 3.1.2 (Existencia y unicidad de polinomios ortogonales [3])** *Para cada  $\omega$  función peso, existe una única sucesión de polinomios ortonormales  $p_0, p_1, \dots, p_n, \dots$ , con coeficiente principal positivo. Es decir:*



1.  $\deg(p_n(x)) = n; \forall n \in \mathbb{Z}_+$ .
2.  $p_n(x) = \gamma_n x^n + \dots + \gamma_0$ , donde  $\gamma_n > 0$ .
3.  $\langle p_n, p_m \rangle_\omega = \int_a^b p_n(x)p_m(x)\omega(x)dx = \delta_{n,m}$ .

A continuación veremos algunas propiedades importantes que satisfacen los polinomios ortogonales

### 3.1.1. Fórmula de recurrencia

Una de las características importantes de los polinomios ortogonales es el hecho de que cualquiera de los tres polinomios consecutivos están conectados por una relación muy simple llamada formulas recursivas o recurrencia que ligan términos de distintos grados y es una forma de construir los polinomios. Esta formula viene dada por:

**Teorema 3.1.3 (Fórmula de recurrencia a tres términos[3])** *Sea la función peso  $\omega$  y  $\{p_n\}_{n=0}^\infty$  su sucesión de polinomios ortonormales asociados. Entonces, se verifica:*

$$a_{n+1}p_{n+1}(x) = (x - b_n)p_n(x) - a_n p_{n-1}(x), \forall n \geq 1, \quad (3.1.5)$$

siendo  $p_n(x) = \gamma_n x^n + \dots$ ,  $a_n = \frac{\gamma_{n-1}}{\gamma_n} \geq 0$ ,  $b_n = \int_a^b x p_n^2(x) p_m(x) \omega(x) dx$ .

Además, existe un resultado de tipo inverso que incluimos seguidamente sin demostración. Si cierta sucesión de polinomios satisface esta relación de recurrencia sera ortogonal, resultado que se recoge en el siguiente teorema.

**Teorema 3.1.4 (Teorema de Favard, ver [6])** Sea  $\{p_n\}_{n=0}^{\infty}$  una sucesión de polinomios que satisface la relación de recurrencia a tres términos, con  $a_0 > 0$ , con las condiciones iniciales  $p_0(x) = \frac{2}{a_0}$ ;  $p_1(x) = \frac{1}{a_1 a_0}(x - b_0)$ . Entonces, existe una función  $\omega(x)$  tal que la sucesión es ortonormal respecto de dicha función peso.

### 3.1.2. Ceros de los polinomios ortogonales

Los ceros de los polinomios ortogonales desempeñan un papel importante en teoría de interpolación, formulas de cuadratura, teoría espectral de algunos operadores lineales, esquemas de filtros digitales, por lo que su estudio posee gran interés. Los ceros de los polinomios ortogonales estándar tienen buenas propiedades, son simples y tienen la propiedad de entrelazamiento.

**Teorema 3.1.5** Dos polinomios ortogonales consecutivos no tienen ceros comunes. Es decir,  $\forall n \in \mathbb{N}; C_{p_n} \cap C_{p_{n+1}} = \emptyset$ , donde  $C_q$  denota el conjunto de ceros de un polinomio  $q$ .

**Proposición 3.1.6** Sea  $p_n$ , el  $n$ -ésimo polinomio ortogonal respecto de la función peso  $\omega$ ,  $p_n$  tiene exactamente  $n$  ceros en el intervalo  $(a, b)$ .

Por último, otro teorema nos asegura que los ceros de dos polinomios ortogonales están entrelazados". Para ello, denotemos por  $x_{m,k}$  el  $k$ -ésimo cero de  $p_m(x)$ , y supongamos que ordenamos en forma decreciente los ceros de los polinomios  $p_n$ ,  $p_{n+1}$  esto es:

$$x_{n,n} < x_{n,n+1} < \cdots < x_{n,1}, \quad (3.1.6)$$

$$x_{n+1,n+1} < x_{n+1,n} < \cdots < x_{n+1,1}. \quad (3.1.7)$$

**Proposición 3.1.7 (Separación de ceros, ver [3])** Sean  $p_n, p_{n+1}$  dos polinomios ortogonales consecutivos respecto de la función peso  $\omega$ . Entonces, sus ceros están ordenados de modo que verifican las iniciaciones:

$$x_{n+1,k+1} < x_{n,k} < x_{n+1,k}, \quad k = 1, \dots, n. \quad (3.1.8)$$

Ahora estudiaremos los polinomios ortogonales clásicos definidos sobre la recta real.

## 3.2. Polinomios ortogonales clásicos

Las familias de polinomios ortogonales sobre la recta más estudiadas y con un extenso campo de aplicaciones son los llamados polinomios ortogonales clásicos, que tuvieron una importancia no solo en el desarrollo de esta teoría sino también en las aplicaciones a otras ciencias.

Se clasifican en tres grandes familias: Jacobi, Laguerre, Hermite y a estas tres familias se le incluye el polinomio de Bessel. En función de las características del intervalo de definición  $[a, b]$ , según se trate de un intervalo acotado  $[a, b]$ , semi-infinito  $[a, \infty)$ , infinito  $\mathbb{R}$  o la circulo unitario  $\mathbb{T}$ .

Las características fundamentales que distinguen a estas familias de las restan-

tes familias de polinomios ortogonales son sus propiedades diferenciales, de ellas las dos mas relevantes son:

1. Constituyen soluciones de una ecuación diferencial de segundo orden del tipo

$$\alpha(x)y''(x) + \beta(x)y'(x) + \alpha_n y(x) = 0, \quad (3.2.1)$$

donde  $\alpha(x), \beta(x)$  son polinomios fijos para cada familia, independiente de  $n$  y con grados respectivos dos y uno. el enésimo polinomio es solución de la ecuación con el correspondiente valor.

2. Pueden ser generado mediante una formula que contiene derivadas de orden  $n$ , conocida como **formula de Rodrigues** (ver [5]). A saber,

$$P_n(x) = \frac{1}{\kappa_n \gamma(x)} \frac{d^n}{dx^n} [\gamma(x) \rho^n(x)], \quad (3.2.2)$$

donde  $\kappa_n$  es independiente de  $x$ ,  $\gamma(x)$  es no negativa e integrable en ciertos intervalo de la recta real y  $\rho(x)$  es un polinomio independiente de  $n$ , de grado a los sumo dos. Tanto  $\kappa_n$ ,  $\gamma(x)$  como  $\rho(x)$  son fijos para cada familia

### 3.2.1. Polinomios de Jacobi

En este caso el intervalo de soporte es  $[-1, 1]$ . Dependen de dos parámetros  $\alpha$  y  $\beta$ , a partir de los cuales se construye la función peso  $\omega(x)$ ,  $x \in [-1, 1]$ . Esta familia de polinomios cumple las siguientes propiedades:

### 1. Relación de ortogonalidad

$$\begin{aligned} \langle P_n^{(\alpha,\beta)}(x), P_m^{(\alpha,\beta)}(x) \rangle_\omega &= \int_{-1}^1 P_n^{(\alpha,\beta)}(x) P_m^{(\alpha,\beta)}(x) \omega(x) dx \\ &= \begin{cases} \frac{2^{\alpha\beta+1}}{n!(2n+\alpha+\beta+1)} \frac{\Gamma(n+\alpha+1)\Gamma(n+\beta+1)}{\Gamma(n+\beta+1)}, & n = m \\ 0, & n \neq m \end{cases} \end{aligned} \quad (3.2.3)$$

donde  $\omega(x) = (1-x)^\alpha(1+x)^\beta$ ,  $x \in [-1, 1]$   $\alpha > -1$   $\beta > -1$ .

### 2. Fórmula de Rodrigues

$$P_n^{(\alpha,\beta)}(x) = \frac{1}{(-2)^n n! (1-x)^\alpha (1+x)^\beta} \frac{d^n}{dx^n} [(1-x)^{\alpha+n} (1+x)^{\beta+n}], \quad (3.2.4)$$

qué, mediante la regla de Leibniz permite obtener una expresión explícita:

$$P_n^{(\alpha,\beta)}(x) = \frac{1}{2^n} \sum_{k=0}^n \binom{n+\alpha}{n-k} \binom{n+\beta}{k} (x-1)^k (x+1)^{n-k}. \quad (3.2.5)$$

### 3. Coeficiente principal o director

Si  $\kappa_n^{(\alpha,\beta)}$  es el coeficiente principal de  $P_n^{(\alpha,\beta)}(x)$  se tiene que

$$\kappa_n^{(\alpha,\beta)} = \frac{1}{2^n} \binom{2n+\alpha+\beta}{n}, \quad (3.2.6)$$

$$P_n^{(\alpha,\beta)}(-x) = (-1)^n P_n^{(\alpha,\beta)}(x) \quad P_n^{(\alpha,\beta)}(1) = \binom{2n+\alpha}{n}. \quad (3.2.7)$$

### 4. Solución de una E.D.O

El polinomio de Jacobi de orden  $n$  es solución de la ecuación diferencial de

segundo orden con coeficientes polinómicos:

$$(x^2 - 1)y''(x) + [(2 + \alpha + \beta)x + \alpha - \beta]y'(x) - n[n + 1 + \alpha + \beta]y(x) = 0. \quad (3.2.8)$$

### 5. Fórmula de recurrencia

La fórmula de recurrencia que verifican los polinomios de Jacobi es la siguiente:

$$p_0^{(\alpha, \beta)}(x) = 1, \quad p_1^{(\alpha, \beta)}(x) = \frac{(\alpha + \beta + 2)x}{2} + \frac{\alpha - \beta}{2}.$$

Para  $n \geq 2$

$$\begin{aligned} (2n + 2)(n + \alpha + \beta + 1)(2n + \alpha + \beta)p_{n+1}^{(\alpha, \beta)}(x) = \\ (2n + \alpha + \beta + 1)[(2n + \alpha + \beta + 2)(2n + \alpha + \beta)x + \alpha^2 - \beta^2]p_n^{(\alpha, \beta)}(x) \\ - 2(n + \alpha)(n + \beta)(2n + \alpha + \beta + 1)p_{n-1}^{(\alpha, \beta)}(x). \end{aligned} \quad (3.2.9)$$

Dentro de los polinomios de Jacobi hay subfamilias de polinomios ortogonales de gran interés, una de ellas son **Polinomios de Gengenbauer**, también llamado polinomio ultraesférico, corresponden al caso simétrico en que  $\alpha = \beta$ , se denota por  $C_n(x)$ , el peso  $\omega(x) = (1 - x^2)^\alpha$ ,  $\alpha > -1$  y el coeficiente principal esta dado por  $\kappa_n^{(\alpha)} = \frac{1}{2^n} \binom{2n+2}{n}$ . Entre ellos se encuentran las siguientes subfamilias que son casos particulares de estos polinomios.

- Polinomios de Chebyshev de primer tipo:** Cuando  $\alpha = \beta = -\frac{1}{2}$ , se denota por  $T_n(x)$ , el peso  $\omega(x) = \frac{1}{\sqrt{1-x^2}}$ , y el coeficiente principal esta dado por  $\kappa_n = \frac{1}{2^n} \binom{2n-1}{n}$ .

Formalmente los vamos a definir mediante la siguiente formula y a partir de ella iremos deduciendo sus propiedades.

$$T_n(x) = \sum_{k=0}^{\lfloor \frac{n}{2} \rfloor} \binom{n}{2k} x^{n-2k} (x^2 - 1)^k. \quad (3.2.10)$$

El coeficiente principal  $\kappa_n$  de  $T_n(x) = \kappa_n x^n + \dots$ ,

$$\kappa_n = \sum_{k=0}^{\lfloor \frac{n}{2} \rfloor} \binom{n}{2k} = 2^{n-1} \neq 0. \quad (3.2.11)$$

Por lo tanto,  $T_n(x)$  es de grado exactamente  $n$  y el correspondiente polinomio de Chebyshev mónico de grado  $n$ ,  $\tau_n(x)$  esta dado por

$$\tau_n(x) = \frac{T_n(x)}{2^{n-1}}, \quad n \geq 1. \quad (3.2.12)$$

Veamos ahora algunas de las propiedades características de estos polinomios.

**Proposición 3.2.1** *Tenemos que:*

- $T_n(x) = \cos(n \arccos(x))$ .
- $T_n(x) = \frac{(x + \sqrt{x^2 - 1})^n + (x - \sqrt{x^2 - 1})^n}{2} \quad x \in \mathbb{C}$ .

**Proposición 3.2.2** *(Localización de ceros) Si  $n \geq 1$  el polinomio  $T_n(x)$  tiene  $n$  ceros simples en los puntos*

$$x_k = \cos\left(\frac{(2k + 1)\pi}{2n}\right), \quad k = 0, 1, 2, \dots, n - 1. \quad (3.2.13)$$

**Proposición 3.2.3** (*Ortogonalidad*) para todo  $m, n \in \mathbb{Z}_+$  los polinomios de Chebyshev cumplen que:

$$\begin{aligned} \langle T_n(x), T_m(x) \rangle &= \int_{-1}^1 T_n(x) T_m(x) \frac{dx}{\sqrt{1-x^2}} \\ &= \begin{cases} \pi, & n = m = 0, \\ \frac{\pi}{2}, & n > m, \\ 0, & n \neq m. \end{cases} \end{aligned} \quad (3.2.14)$$

**Proposición 3.2.4** (*Formula de recurrencia*) para todo  $x \in \mathbb{C}$  y  $n > 1$

- $T_{n+1}(x) = 2xT_n(x) - T_{n-1}(x)$ ,  $T_0(x) = 1$  y  $T_1(x) = x$ .
  - $\tau_{n+1}(x) = x\tau_n(x) - \frac{1}{4}\tau_{n-1}(x)$ ,  $\tau_0(x) = 1$  y  $\tau_1(x) = x$ .
- **Polinomios de Chebyshev de segundo tipo:** Cuando  $\alpha = \beta = \frac{1}{2}$ , se denota por  $U_n(x)$ , el peso  $\omega(x) = \sqrt{1+x^2}$ , y el coeficiente principal esta dado por  $\kappa_n = \frac{1}{2^n} \binom{2n+1}{n}$ .
  - **Polinomios de Legendre:** Cuando  $\alpha = \beta = 0$ , se denota por  $P_n(x)$ , el peso  $\omega(x) = 1$ , y el coeficiente principal esta dado por  $\kappa_n = \frac{1}{2^n} \binom{2n}{n}$ .

### 3.2.2. Polinomios de Laguerre

El intervalo de soporte en este caso es  $(0, \infty)$ , es decir, semi- infinito. La función peso depende de un parámetro,  $\alpha$ , el  $n$ -ésimo polinomio de Laguerre se denota por  $L_n^\alpha(x)$ . Esta familia de polinomios cumple las siguientes propiedades:



### 1. Relación de ortogonalidad

$$\begin{aligned} \langle L_n^\alpha(x), L_m^\alpha(x) \rangle_\omega &= \int_0^\infty L_n^\alpha(x) L_m^\alpha(x) \omega(x) dx \\ &= \begin{cases} \frac{\Gamma(n+\alpha+1)}{n!}, & n = m \\ 0, & n \neq m \end{cases} \end{aligned} \quad (3.2.15)$$

donde  $\omega(x) = x^\alpha \exp(-x)$ ,  $x \in (0, \infty)$ ,  $\alpha > -1$ .

### 2. Fórmula de Rodrigues

Puede generarse a partir de la fórmula de Rodrigues

$$L_n^\alpha(x) = \frac{\exp(x)}{n! x^\alpha} \frac{d^n}{dx^n} [\exp(-x) x^{n+\alpha}], \quad (3.2.16)$$

donde se deduce, mediante la regla de Leibniz, la siguiente expresión explícita:

$$L_n^\alpha(x) = \sum_{k=0}^n \binom{n+\alpha}{n-k} \frac{(-x)^k}{k!}. \quad (3.2.17)$$

### 3. Coeficiente principal o director

Se tiene que  $\kappa_n^\alpha$  es el coeficiente principal de  $L_n^\alpha(x)$  y está dado por

$$\kappa_n^\alpha = \frac{(-1)^n}{n!}. \quad (3.2.18)$$

### 4. Solución de una E.D.O

El  $n$ -ésimo polinomio de Laguerre es solución de la ecuación diferencial:

$$xy''(x) + (\alpha + 1 - x)y'(x) + ny(x) = 0. \quad (3.2.19)$$

### 5. Fórmula de recurrencia

La fórmula de recurrencia que verifican los polinomios de Laguerre es la siguiente:

$$L_0^{(\alpha)}(x) = 1, \quad L_1^{(\alpha)}(x) = -x + \alpha + \beta.$$

Para  $n = 1, 2, 3, \dots$

$$(n+1)L_{n+1}^{(\alpha)}(x) = (2n + \alpha - 1 - x)L_n^{(\alpha)}(x) - (n + \alpha)L_{n-1}^{(\alpha)}(x). \quad (3.2.20)$$

### 3.2.3. Polinomios de Hermite

El intervalo de soporte en este caso es  $(-\infty, \infty)$ . La notación habitual para estos polinomios es,  $H_n(x)$ , para cada  $n \in \mathbb{Z}_+$ . Esta familia de polinomios cumple las siguientes propiedades:

#### 1. Relación de ortogonalidad

$$\begin{aligned} \langle H_n(x), H_m(x) \rangle &= \int_{-\infty}^{\infty} H_n(x)H_m(x)\omega(x)dx \\ &= \begin{cases} \sqrt{\pi}2^n n!, & n = m \\ 0, & n \neq m \end{cases} \end{aligned} \quad (3.2.21)$$

donde  $\omega(x) = \exp(-x^2)$ ,  $x \in \mathbb{R}$ .

#### 2. Fórmula de Rodrigues

Puede generarse a partir de la fórmula de Rodrigues

$$H_n(x) = (-1)^n \exp(x^2) \frac{d^n}{dx^n} [\exp(-x^2)], \quad (3.2.22)$$

donde se deduce, mediante la regla de Leibniz, la siguiente expresión explícita:

$$H_n(x) = n! \sum_{k=0}^{\lfloor \frac{n}{2} \rfloor} \frac{(-1)^k (2x)^{n-2k}}{(n-2k)! k!}. \quad (3.2.23)$$

### 3. Coeficiente principal o director

Se tiene que  $\kappa_n$  es el coeficiente principal de  $H_n(x)$  y está dado por

$$\kappa_n = 2^n. \quad (3.2.24)$$

### 4. Solución de una E.D.O

El  $n$ -ésimo polinomio de Hermite es solución de la ecuación diferencial:

$$y''(x) - 2xy'(x) + 2ny(x) = 0. \quad (3.2.25)$$

### 5. Fórmula de recurrencia

La fórmula de recurrencia que verifican los polinomios de Hermite es la siguiente:

$$H_0(x) = 1, \quad H_1(x) = 2x.$$

$$H_{n+1}(x) = 2xH_n(x) - 2nH_{n-1}(x). \quad (3.2.26)$$

### 3.2.4. Polinomios de Bessel

Los polinomios de Bessel, a diferencia de las tres familias anteriores no corresponden a un caso definido positivo, es decir la medida de ortogonalidad no es positiva. Aunque estos polinomios habían sido considerados por muchos matemáticos, fueron H. L. Krall y O. Frink [8] quienes los presentaron formalmente y les dieron el nombre por su relación con las funciones de Bessel. El intervalo de soporte en este caso es  $\mathbb{T} = \{z \in \mathbb{C} : |z| = 1\}$ . La notación habitual para estos polinomios es,  $B_n^{(\alpha)}(x)$ , para cada  $n \in \mathbb{Z}_+$ [3]. Esta familia de polinomios cumple las siguientes propiedades:

#### 1. Relación de ortogonalidad

$$\begin{aligned} \langle B_n^\alpha(x), B_m^\alpha(x) \rangle_\omega &= \int_{\mathbb{T}} B_n^\alpha(x) B_m^\alpha(x) \omega(x) dx \\ &= \begin{cases} 2\pi i \frac{2(-1)^{n+1} n!}{(2n+\alpha+1)(\alpha+1)_n}, & n = m \\ 0, & n \neq m \end{cases} \end{aligned} \quad (3.2.27)$$

donde  $\omega(x) = x^\alpha e^{\frac{2}{x}}$ ,  $x \in [-1, 1]$ ,  $\alpha \geq 0$ .

#### 2. Fórmula de Rodrigues

Puede generarse a partir de la formula de rodrigues

$$B_n^\alpha(x) = \frac{\exp \frac{2}{x}}{2^n x^\alpha} \frac{d^n}{dx^n} \left[ \frac{x^{2n+\alpha}}{\exp \frac{2}{x}} \right], \quad (3.2.28)$$

donde se deduce, la expresión explícita:

$$B_n^\alpha(x) = \sum_{k=0}^n \binom{n}{k} (n + \alpha + 1)_k \left(\frac{x}{2}\right)^k. \quad (3.2.29)$$

### 3. Coeficiente principal o director

Se tiene que  $\kappa_n^\alpha$  es el coeficiente principal de  $B_n^\alpha(x)$  y esta dado por

$$\kappa_n^\alpha = \frac{(n + \alpha + 1)_n}{2^n}. \quad (3.2.30)$$

### 4. Solución de una E.D.O

El enésimo polinomio de Bessel es solución de la ecuación diferencial:

$$x^2 y''(x) + 2(x + 1)y'(x) - n(n + 1)y(x) = 0. \quad (3.2.31)$$

### 5. Fórmula de recurrencia

La fórmula de recurrencia que verifican los polinomios de Bessel es la siguiente:

$$B_0^{(\alpha)}(x) = 1, \quad B_1^{(\alpha)}(x) = \frac{(\alpha+2)}{2}x + 1.$$

Para  $n \geq 1$

$$\begin{aligned} & 2(n + \alpha + \beta)(2n + \alpha)B_{n+1}^{(\alpha)}(x) = \\ & (2n + \alpha + 1)[(2n + \alpha)(2n + \alpha + 2)x + 2\alpha]B_n^{(\alpha)}(x) \\ & + 2n(2n + \alpha + 2)B_{n-1}^{(\alpha)}(x). \end{aligned} \quad (3.2.32)$$

### 3.3. Función núcleo de los polinomios ortogonales clásicos

Una función que juega un papel importante en la teoría de polinomios ortogonales es la función núcleo. Esta suma puede ser calculada en forma cerrada. El resultado, que se conoce como fórmula de Christoffel–Darboux.

**Teorema 3.3.1 (Fórmula de sumación de Christoffel–Darboux ver [9])** *Consideremos la sucesión  $\{p_n\}_{n=0}^{\infty}$  de polinomios ortogonales correspondiente entonces se cumple:*

$$K_n(x, y) = \sum_{k=0}^n p_k(x)p_k(y) = a_{n+1} \frac{p_{n+1}(x)p_n(y) - p_n(x)p_{n+1}(y)}{x - y}. \quad (3.3.1)$$

Si hacemos tender  $y \rightarrow x$ , obtenemos la fórmula confluyente de Christoffel–Darboux siguiente:

**Corolario 3.3.2 (Fórmula confluyente de sumación de Christoffel–Darboux ver [9])**

*Consideremos la sucesión  $\{p_n\}_{n=0}^{\infty}$  de polinomios ortogonales correspondiente entonces se cumple:*

$$K_n(x, x) = \sum_{k=0}^n (p_k(x))^2 = a_{n+1} (p'_{n+1}(x)p_n(x) - p'_n(x)p_{n+1}(x)). \quad (3.3.2)$$

**Proposición 3.3.3 (ver [9])** *Los polinomios núcleos satisfacen la siguiente pro-*

*propiedad reproductora*

$$\int_a^b p(x)K_n(x, y)dx = p(y), \quad \forall p(x) \in \mathbb{P}_n. \quad (3.3.3)$$

La fórmula de Christoffel-Darboux para los polinomios mencionados anteriormente son:

■ Polinomios de Jacobi

$$K_n(x, y) = (2n+2)(n+\alpha+\beta+1)(2n+\alpha+\beta) \frac{P_{n+1}^{(\alpha, \beta)}(x)P_n^{(\alpha, \beta)}(y) - P_n^{(\alpha, \beta)}(x)P_{n+1}^{(\alpha, \beta)}(y)}{x - y}. \quad (3.3.4)$$

● Polinomios de Chebyshev de primer tipo

$$\begin{aligned} K_n(x, y) &= \frac{T_{n+1}(x)T_n(y) - T_n(x)T_{n+1}(y)}{x - y} \\ &= \frac{\cos((n+1)\cos^{-1}(x))\cos(n\cos^{-1}(y)) - \cos(n\cos^{-1}(x))\cos((n+1)\cos^{-1}(y))}{x - y}. \end{aligned} \quad (3.3.5)$$

■ Polinomios de Laguerre

$$K_n(x, y) = (n + 1) \frac{L_{n+1}^\alpha(x)L_n^\alpha(y) - L_n^\alpha(x)L_{n+1}^\alpha(y)}{x - y}. \quad (3.3.6)$$

■ Polinomios de Hermite

$$K_n(x, y) = \frac{H_{n+1}(x)H_n(y) - H_n(x)H_{n+1}(y)}{x - y}. \quad (3.3.7)$$

- Polinomios de Bessel

$$K_n(x, y) = 2(n + \alpha + \beta)(2n + \alpha) \frac{B_{n+1}^\alpha(x)B_n^\alpha(y) - B_n^\alpha(x)B_{n+1}^\alpha(y)}{x - y}. \quad (3.3.8)$$



## Resultados numéricos

En este capítulo presentamos los resultados del proceso de clasificación de varios conjuntos de datos. La implementación algorítmica se realizó con el software MATLAB version 7.12.0.635 (R2011a) de 32-bit, march 18, 2011, en un computador lenovo G485, modelo 20136, procesador AMD C-70 APU, disco duro de 500 GB, memoria RAM de 2,00 GB y sistema operativo de 32-bit.

Para resolver el problema de programación cuadrática del capítulo 1 se usó el programa Quadprog del toolbox optimización de MATLAB.

Las pruebas consisten en aplicar la clasificación de los datos con los diferentes núcleos tales como: núcleo generado por el polinomio ortogonal de Chebyshev, núcleo generado por el polinomio ortogonal de Bessel, núcleo generado por el polinomio ortogonal de Hermite, núcleo polinomial homogéneo y núcleo polinomial no homogéneo y así comparar el comportamiento de estos en la data.

En los siguientes ejemplos se muestran cuatro tablas para cada conjunto de da-

tos a clasificar las cuales describen los resultados de las corridas con los diferentes núcleos empleados, entre ellos resaltan, el numero de vectores soportes, el tiempo que tarda en ejecutarse cada corrida, si clasifica o no el núcleo correspondiente y finalmente se anexan las imágenes para cada caso.

**Ejemplo 4.0.4** *Tablero con una cantidad de 234 puntos a clasificar, donde 125 pertenecen a una clase y 109 pertenecen a la otra clase, con tolerancia de  $10^{-3}$*

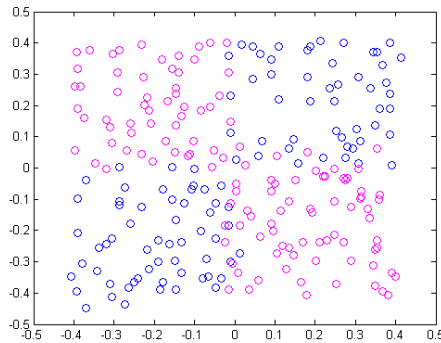


Figura 4.1: TABLERO2

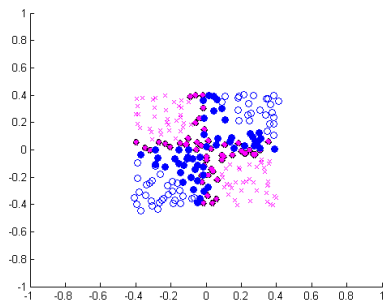
*El núcleo que se usó para mostrar los resultados de en esta tabla fue el núcleo basado en un polinomio de de chebyshev, es decir*

$$K_n(x, y) = \frac{\langle T, T' \rangle}{\sqrt{1 - xy}}$$

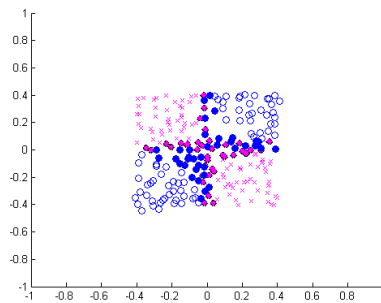
*específicamente se utilizaron los tres primeros grados [ver [21]]*

| Núcleo                  | Tiempo en seg | Vec.Sopt. | Clasifica |
|-------------------------|---------------|-----------|-----------|
| Chebyshev Primer grado  | 7.8846        | 104       | si        |
| Chebyshev Segundo grado | 8.2016        | 77        | si        |
| Chebyshev Tercer grado  | 8.9783        | 85        | si        |

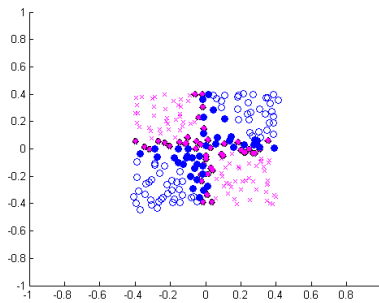
Cuadro 4.1: Núcleo basado en un polinomio de Chebyshev



Núcleo basado en un polinomio de Chebyshev primer grado



Núcleo basado en un polinomio de Chebyshev segundo grado



Núcleo basado en un polinomio de Chebyshev tercer grado

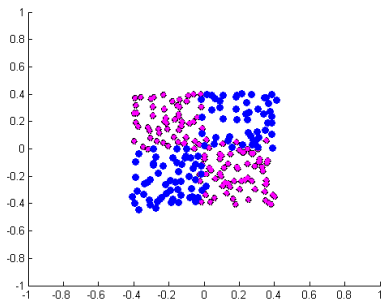
*Se puede observar que si hubo clasificación en los tres casos estudiados, pero en el núcleo basado en un polinomio de chebyshev de segundo grado fue donde se obtuvo una menor cantidad de vectores soportes; es decir; esta fue la mejor clasificación.*

En este caso el núcleo usado para clasificar es el núcleo basado en un polinomio de Bessel que viene dado por

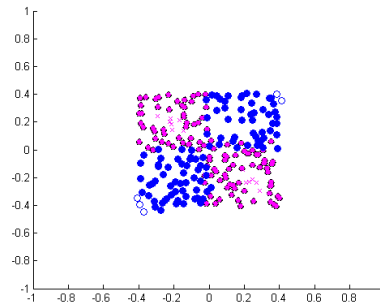
$$K(x, y) = -Bessel_{\nu+1}^n(\sigma(|x - y|^2))$$

| Núcleo | parámetro    | Tiempo en seg | Vec.Sopt. | Clasifica |
|--------|--------------|---------------|-----------|-----------|
| Bessel | $\nu = -5$   | 26.6683       | 234       | no        |
| Bessel | $\nu = 0$    | 23.0430       | 216       | no        |
| Bessel | $\nu = 0, 1$ | 25.7249       | 87        | si        |
| Bessel | $\nu = 0, 5$ | 29.3824       | 54        | si        |
| Bessel | $\nu = 1$    | 30.4242       | 217       | no        |
| Bessel | $\nu = 10$   | 21.6660       | 234       | no        |

Cuadro 4.2: Núcleo basado en un polinomio de Bessel



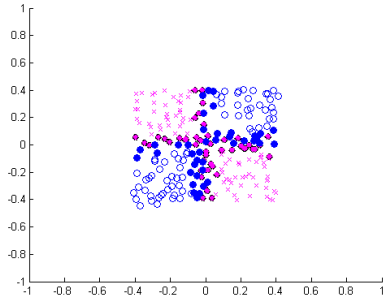
Núcleo basado en un polinomio de Bessel  $\nu = -5$



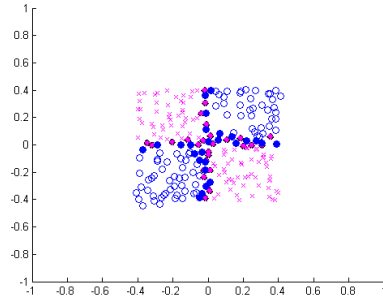
Núcleo basado en un polinomio de Bessel  $\nu = 0$

Se observa que para el caso en que el parámetro  $\nu$  está muy próximo a cero es donde si hubo clasificación, y la mejor fue para  $\nu = 0, 5$ , sin embargo en el caso en que  $\nu$  tiene valores muy altos o muy bajo el programa no clasifica.

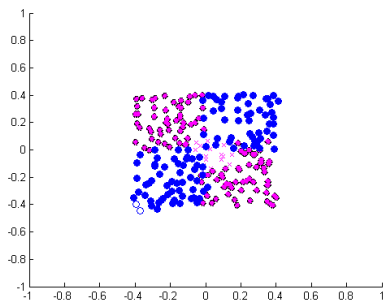
El núcleo usado para clasificar en la tabla siguiente es el núcleo basado en



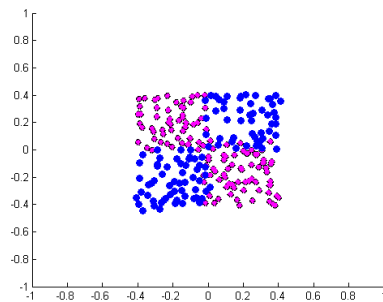
Núcleo basado en un polinomio de Bessel  $\nu = 0,1$



Núcleo basado en un polinomio de Bessel  $\nu = 0,5$



Núcleo basado en un polinomio de Bessel  $\nu = 1$



Núcleo basado en un polinomio de Bessel  $\nu = 10$

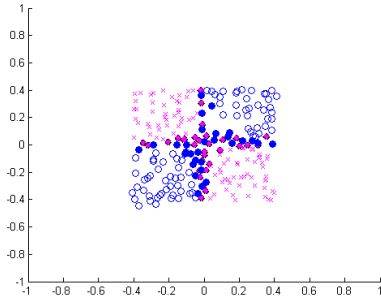
un polinomio de Hermite viene dado por:

$$K_n(x, y) = \frac{H_{n+1}(x)H_n(y) - H_n(x)H_{n+1}(y)}{e^{-xy}}$$

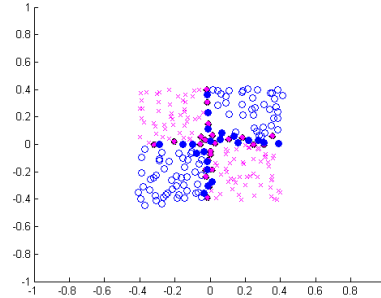
donde  $H_{n+1}(x)$  se obtiene por la fórmula de recurrencia para el polinomio de Hermite (3.3.7)

| Núcleo                | Tiempo en seg | Vec.Sopt. | Clasifica |
|-----------------------|---------------|-----------|-----------|
| Hermite primer grado  | 16.9691       | 62        | si        |
| Hermite segundo grado | 18.5438       | 46        | si        |
| Hermite tercer grado  | 19.4974       | 46        | si        |
| Hermite cuarto grado  | 27.4958       | 44        | si        |

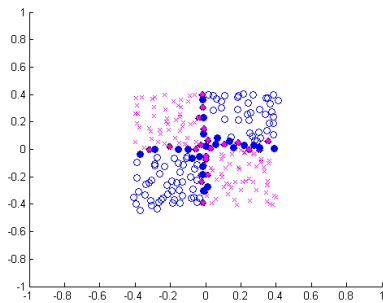
Cuadro 4.3: Núcleo basado en un polinomio de Hermite



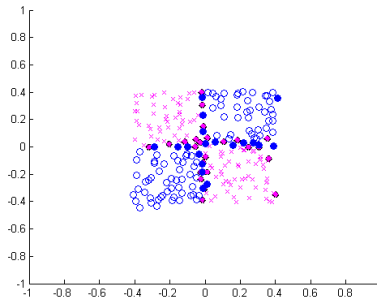
Núcleo basado en un polinomio de Hermite primer grado



Núcleo basado en un polinomio de segundo grado



Núcleo basado en un polinomio de tercer grado



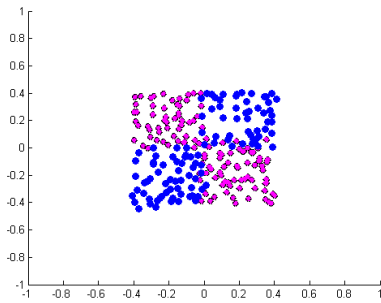
Núcleo basado en un polinomio de Hermite cuarto grado

*Se puede ver que si hubo clasificación en los cuatro casos estudiados y se observa que a medida que se aumenta el grado del núcleo se tiene una mejor clasificación pues la cantidad de vectores soportes disminuye.*

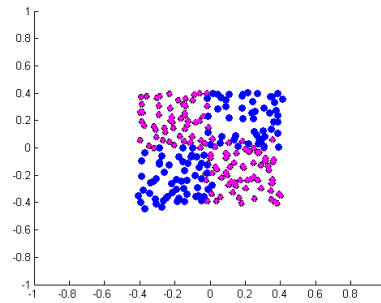
*Los resultados mostrados en la siguiente tabla son realizados con el núcleo polinomial homogéneo dado por:  $K(x, y) = (x * y)^n$  y núcleo polinomial no homogéneo dado por  $K(x, y) = ((x * y) + c)^n$  donde  $n$  es el grado polinomio y  $c > 0$*

| Núcleo                  | Grado P | Parámetro | Tiempo  | Vec.Sopt. | Clasifica |
|-------------------------|---------|-----------|---------|-----------|-----------|
| Polinomial Homogéneo    | 10      |           | 11.3749 | 234       | no        |
| Polinomial Homogéneo    | 21      |           | 17.3173 | 234       | no        |
| Polinomial No Homogéneo | 3       | C=1       | 15.3438 | 55        | si        |
| Polinomial No Homogéneo | 18      | C=2       | 20.8948 | 34        | si        |

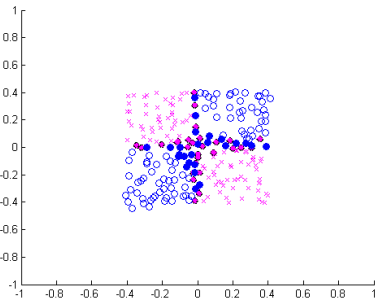
Cuadro 4.4: Núcleo polinomial homogéneo y no homogéneo



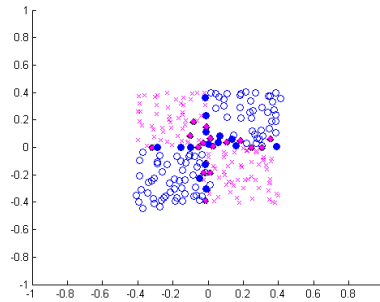
Núcleo polinomial homogéneo de grado 10



Núcleo polinomial homogéneo de grado 21



Núcleo polinomial no homogéneo de grado 3 c=1



Núcleo polinomial no homogéneo de grado 18 c=2

*Se observa que para el núcleo polinomial homogéneo no hubo clasificación, mientras que el núcleo polinomial no homogéneo si clasificó y a medida que el grado del polinomio aumenta se obtiene una mejor clasificación pues el número de vectores soportes disminuye*

**Ejemplo 4.0.5** En este ejemplo se muestra un tablero pero ahora con una cantidad de 402 puntos a clasificar, donde 200 pertenecen a una clase y 202 pertenecen a la otra clase, y una tolerancia de  $10^{-3}$

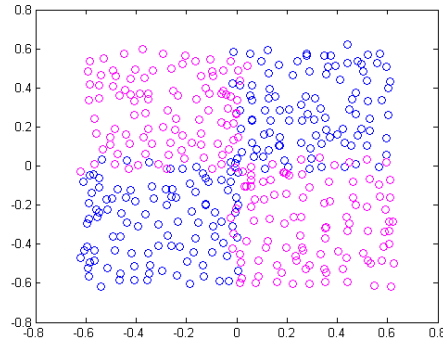


Figura 4.2: TABLERO4

En la siguiente tabla se muestran los resultados obtenidos usando el núcleo basado en un polinomio de chebyshev, es decir

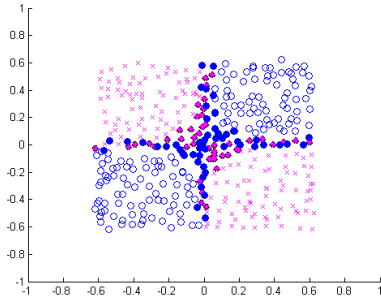
$$K_n(x, y) = \frac{\langle T, T' \rangle}{\sqrt{1 - xy}}$$

específicamente se utilizaron los tres primeros grados [ver [21]]

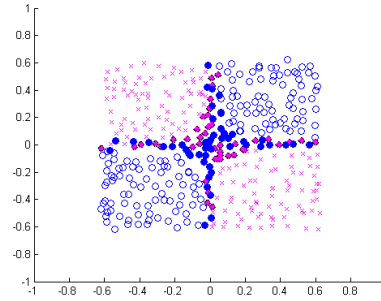
| Núcleo                  | Tiempo en seg | Vec.Sopt. | Clasifica |
|-------------------------|---------------|-----------|-----------|
| Chebyshev Primer grado  | 7.1604        | 101       | si        |
| Chebyshev Segundo grado | 30.8002       | 95        | si        |
| Chebyshev Tercer grado  | 37.2737       | 101       | si        |

Cuadro 4.5: Núcleo basado en un polinomio de Chebyshev

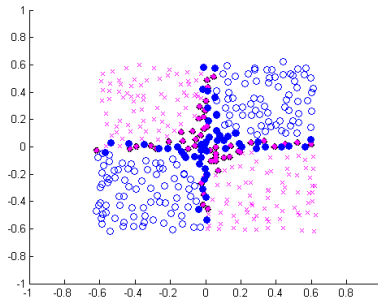




Núcleo basado en un polinomio de Chebyshev primer grado



Núcleo basado en un polinomio de Chebyshev segundo grado



Núcleo basado en un polinomio de Chebyshev tercer grado

*Las figuras nos muestran que si hubo clasificación en los tres casos estudiados, pero en el núcleo ortogonal de chebyshev de segundo grado fue donde se obtuvo la mejor clasificación pues los resultados arrojaron una menor cantidad de vectores soportes.*

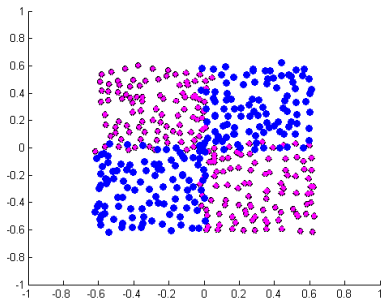
*En este caso el núcleo usado para clasificar es el núcleo basado en un polinomio de Bessel que viene dado por*

$$K(x, y) = -Bessel_{\nu+1}^n(\sigma(|x - y|^2))$$

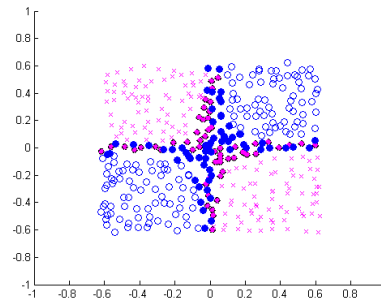
*donde  $\nu$  es un numero real*

| Núcleo | parámetro   | Tiempo en seg | Vec.Sopt. | Clasifica |
|--------|-------------|---------------|-----------|-----------|
| Bessel | $\nu = -50$ | 86.1418       | 402       | no        |
| Bessel | $\nu = 0$   | 54.5672       | 120       | si        |
| Bessel | $\nu = 50$  | 95.0748       | 402       | no        |
| Bessel | $\nu = 100$ | 70.7439       | 402       | no        |

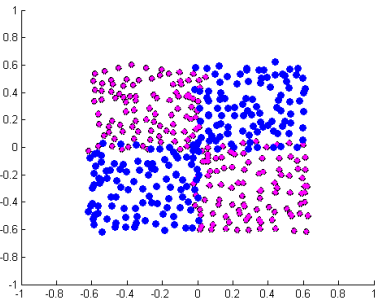
Cuadro 4.6: Núcleo basado en un polinomio de Bessel



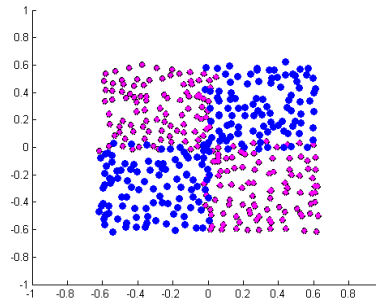
Núcleo basado en un polinomio de Bessel  $\nu = -50$



Núcleo basado en un polinomio de Bessel  $\nu = 0$



Núcleo basado en un polinomio de Bessel  $\nu = 50$



Núcleo basado en un polinomio de Bessel  $\nu = 100$

En las figuras y la tabla se observa que para el caso en que el parámetro  $\nu = 0$  es donde se obtiene clasificación, mientras que para valores de  $\nu$  muy altos o muy bajo el programa no clasifica, además en el caso en que  $\nu$  esta proximo a cero el programa no entra.

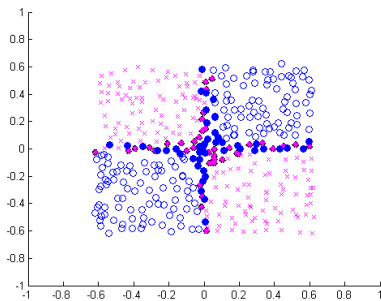
El núcleo usado para clasificar en la tabla siguiente es el núcleo basado en un polinomio de Hermite viene dado por:

$$K_n(x, y) = \frac{H_{n+1}(x)H_n(y) - H_n(x)H_{n+1}(y)}{e^{-xy}}$$

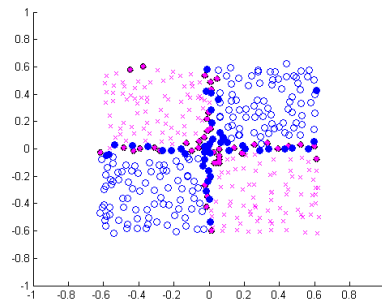
donde  $H_{n+1}(x)$  se obtiene por la fórmula de recurrencia para el polinomio de Hermite (3.3.7)

| Núcleo                | Tiempo en seg | Vec.Sopt. | Clasifica |
|-----------------------|---------------|-----------|-----------|
| Hermite primer grado  |               |           | no entra  |
| Hermite segundo grado | 49.6853       | 87        | si        |
| Hermite tercer grado  | 48.5247       | 88        | si        |
| Hermite cuarto grado  | 93.5368       | 89        | si        |

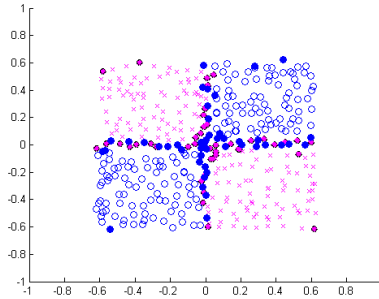
Cuadro 4.7: Núcleo basado en un polinomio de Hermite



Núcleo basado en un polinomio de Hermite segundo grado



Núcleo basado en un polinomio de Hermite tercer grado



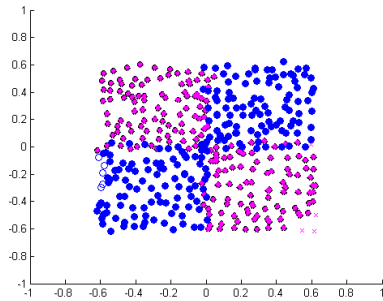
Núcleo basado en un polinomio de Hermite cuarto grado

Los resultados obtenidos en la tabla y en las figuras muestran que para el núcleo de primer grado los datos no entraron al programa, mientras que a partir del núcleo de segundo grado si hubo clasificación, sin embargo a medida que se aumenta el grado del núcleo se puede ver que el número de vectores soportes aumenta de manera mínima, por ello para esta data el que mejor clasificó fue el núcleo ortogonal de Hermite de segundo grado.

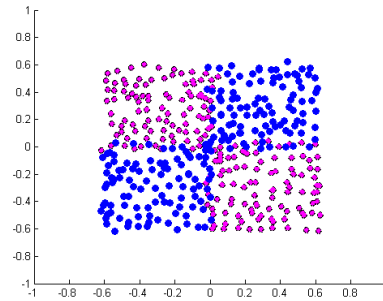
Los resultados mostrados en la siguiente tabla son realizados con el núcleo polinomial homogéneo dado por:  $K(x, y) = (x * y)^n$  y núcleo polinomial no homogéneo dado por  $K(x, y) = ((x * y) + c)^n$  donde  $n$  es el grado polinomio y  $c > 0$

| Núcleo                  | Grado P | Parámetro | Tiempo  | Vec.Sopt. | Clasifica |
|-------------------------|---------|-----------|---------|-----------|-----------|
| Polinomial Homogéneo    | 3       |           | 28.2011 | 393       | no        |
| Polinomial Homogéneo    | 20      |           | 28.4368 | 402       | no        |
| Polinomial No Homogéneo | 2       | C=2       | 37.3667 | 94        | si        |
| Polinomial No Homogéneo | 15      | C=3       | 79.2961 | 54        | si        |

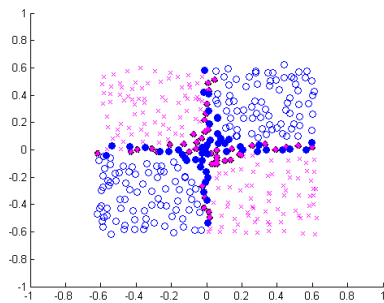
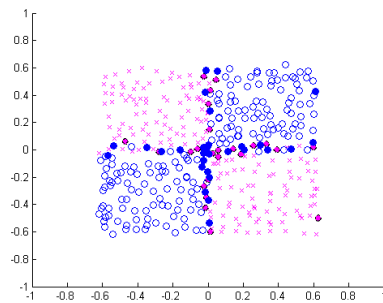
Cuadro 4.8: Núcleo polinomial homogéneo y no homogéneo



Núcleo polinomial homogéneo de grado 3



Núcleo polinomial homogéneo de grado 20

Núcleo polinomial no homogéneo de grado 2  $c=2$ Núcleo polinomial no homogéneo de grado 15  $c=3$ 

*Se observa que para el núcleo polinomial homogéneo no hubo clasificación, mientras que el núcleo polinomial no homogéneo si clasificó y a medida que el grado del polinomio aumenta se obtiene una mejor clasificación pues el número de vectores soportes disminuye*

**Ejemplo 4.0.6** *El siguiente ejemplo muestra una data mas suave es un círculo con una cantidad de 238 puntos a clasificar, donde 161 pertenecen a una clase y 77 pertenecen a la otra clase, y una tolerancia de  $10^{-3}$*

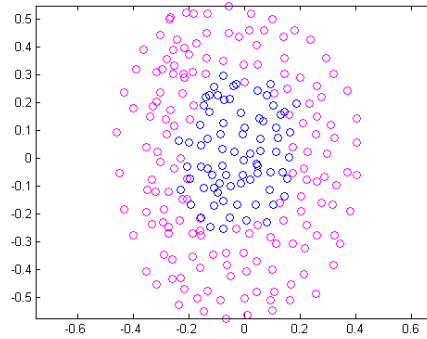


Figura 4.3: CIRCULO2

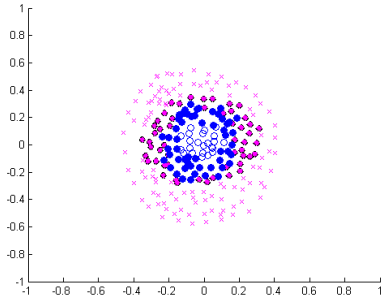
*La tabla que se muestra a continuación se ven los resultados obtenidos usando el núcleo basado en un polinomio de chebyshev, es decir*

$$K_n(x, y) = \frac{\langle T, T' \rangle}{\sqrt{1 - xy}}$$

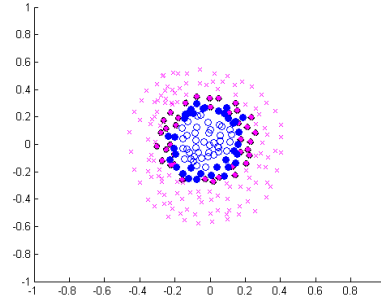
*específicamente se utilizaron los tres primeros grados [ver [21]]*

| Núcleo                  | Tiempo en seg | Vec.Sopt. | Clasifica |
|-------------------------|---------------|-----------|-----------|
| Chebyshev Primer grado  | 19.1119       | 107       | si        |
| Chebyshev Segundo grado | 10.1339       | 74        | si        |
| Chebyshev Tercer grado  | 10.7170       | 42        | si        |

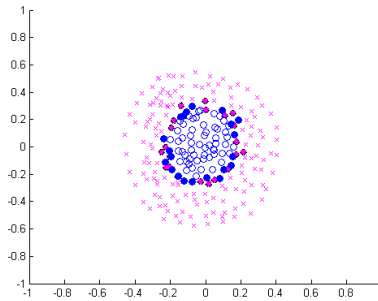
Cuadro 4.9: Núcleo basado en un polinomio de Chebyshev



Núcleo basado en un polinomio de Chebyshev primer grado



Núcleo basado en un polinomio de segundo grado



Núcleo basado en un polinomio de Chebyshev tercer grado

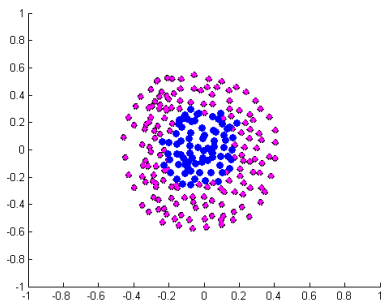
*Se puede observar que si hubo clasificación en los tres casos estudiados, y a medida que se aumenta el grado de núcleo se obtiene una mejor calcificación, en este caso el mejor fue el núcleo ortogonal de chebyshev de tercer grado, pues allí se obtuvo una menor cantidad de vectores soportes.*

*En este caso el núcleo usado para clasificar es el núcleo basado en un polinomio de Bessel que viene dado por*

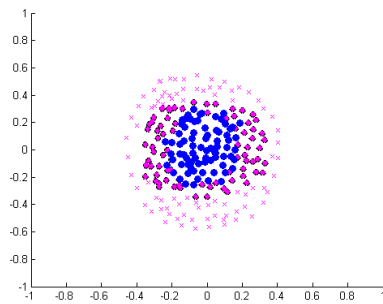
$$K(x, y) = -Bessel_{\nu+1}^n(\sigma(|x - y|^2))$$

| Núcleo | parámetro   | Tiempo en seg | Vec.Sopt. | Clasifica |
|--------|-------------|---------------|-----------|-----------|
| Bessel | $\nu = -3$  | 29.4606       | 238       | no        |
| Bessel | $\nu = 0$   | 24.3960       | 155       | si        |
| Bessel | $\nu = 0,5$ | 40.4603       | 42        | si        |
| Bessel | $\nu = 1,5$ | 32.7268       | 101       | si        |
| Bessel | $\nu = 20$  | 27.6100       | 238       | no        |

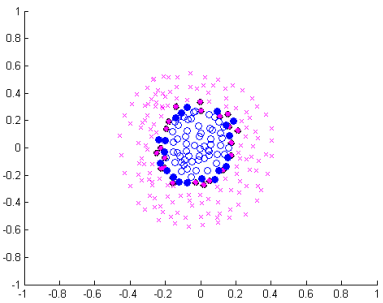
Cuadro 4.10: Núcleo basado en un polinomio de Bessel



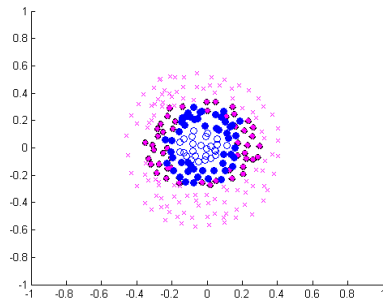
Núcleo basado en un polinomio de Bessel  $\nu = -3$



Núcleo basado en un polinomio de Bessel  $\nu = 0$

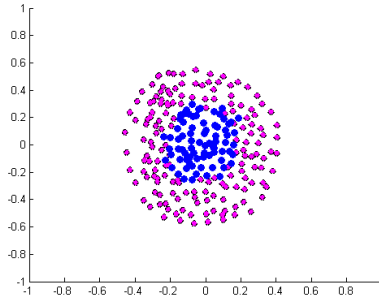


Núcleo basado en un polinomio de Bessel  $\nu = 0,5$



Núcleo basado en un polinomio de Bessel  $\nu = 1,5$



Núcleo basado en un polinomio de Bessel  $\nu = 20$ 

*Se observa que para el caso en que el parámetro  $\nu$  está muy próximo a cero e incluso igual a cero si hubo clasificación, y la mejor fue para  $\nu = 0,5$ , sin embargo en el caso en que  $\nu$  tiene valores muy altos o muy bajos el programa no clasifica.*

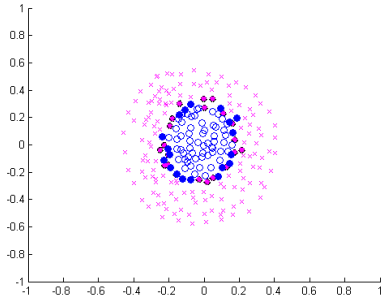
*El núcleo usado para clasificar en la tabla siguiente es el núcleo basado en un polinomio de Hermite viene dado por:*

$$K_n(x, y) = \frac{H_{n+1}(x)H_n(y) - H_n(x)H_{n+1}(y)}{e^{-xy}}$$

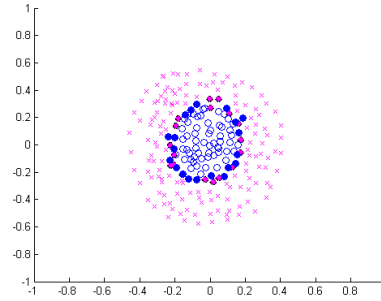
*donde  $H_{n+1}(x)$  se obtiene por la fórmula de recurrencia para el polinomio de Hermite (3.3.7)*

| Núcleo                | Tiempo en seg | Vec.Sopt. | Clasifica |
|-----------------------|---------------|-----------|-----------|
| Hermite primer grado  | 18.1352       | 40        | si        |
| Hermite segundo grado | 16.8108       | 38        | si        |
| Hermite tercer grado  | 18.6192       | 39        | si        |
| Hermite cuarto grado  | 34.3039       | 39        | si        |

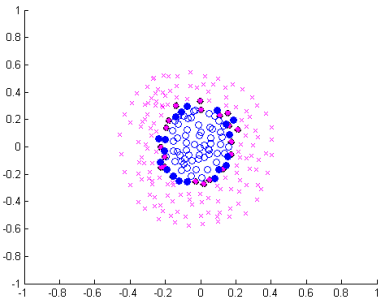
Cuadro 4.11: Núcleo basado en un polinomio de Hermite



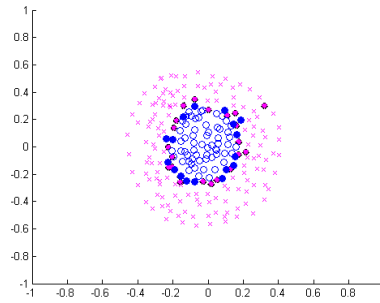
Núcleo basado en un polinomio de Hermite primer grado



Núcleo basado en un polinomio de Hermite segundo grado



Núcleo basado en un polinomio de Hermite tercer grado



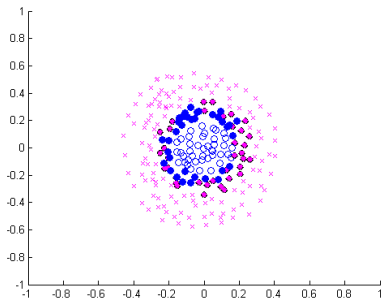
Núcleo basado en un polinomio de Hermite cuarto grado

*Se puede ver que si hubo clasificación en los cuatro casos estudiados y se observa que a medida que se aumenta el grado del núcleo se tiene una mejor clasificación pues la cantidad de vectores soportes disminuye.*

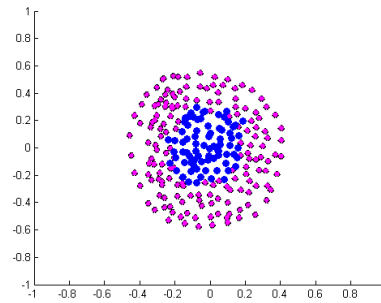
*Los resultados mostrados en la siguiente tabla son realizados con el núcleo polinomial homogéneo dado por:  $K(x, y) = (x * y)^n$  y núcleo polinomial no homogéneo dado por  $K(x, y) = ((x * y) + c)^n$  donde  $n$  es el grado polinomio y  $c > 0$*

| Núcleo                  | Grado P | Parámetro | Tiempo  | Vec.Sopt. | Clasifica |
|-------------------------|---------|-----------|---------|-----------|-----------|
| Polinomial Homogéneo    | 2       |           | 12.2494 | 71        | si        |
| Polinomial Homogéneo    | 9       |           | 11.7184 | 238       | no        |
| Polinomial No Homogéneo | 4       | C=4       | 27.1424 | 38        | si        |
| Polinomial No Homogéneo | 20      | C=3       | 27.2293 | 10        | si        |

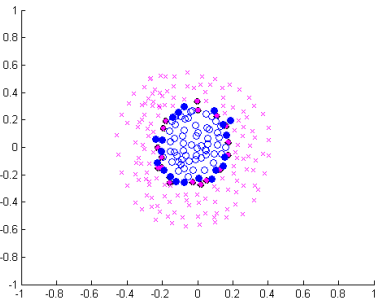
Cuadro 4.12: Núcleo polinomial homogéneo y no homogéneo



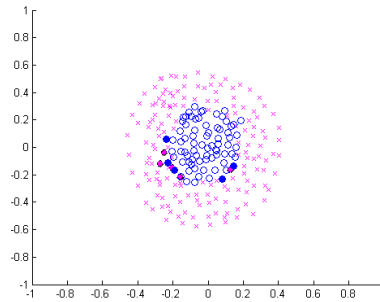
Núcleo polinomial homogéneo de grado 2



Núcleo polinomial homogéneo de grado 9



Núcleo polinomial no homogéneo de grado 4 c=4



Núcleo polinomial no homogéneo de grado 20 c=3

*Se observa que para el núcleo polinomial homogéneo si hubo clasificación, en el caso en que el grado del polinomio es bajo, mientras que en el caso en que el grado es alto no clasifica, para el núcleo polinomial no homogéneo si clasificó y a medida que el grado del polinomio aumenta se obtiene una mejor clasificación pues el número de vectores soportes disminuye*

**Ejemplo 4.0.7** En este ejemplo se muestra el círculo pero ahora con una cantidad de 445 puntos a clasificar, donde 256 pertenecen a una clase y 189 pertenecen a la otra clase, y una tolerancia de  $10^{-3}$ .

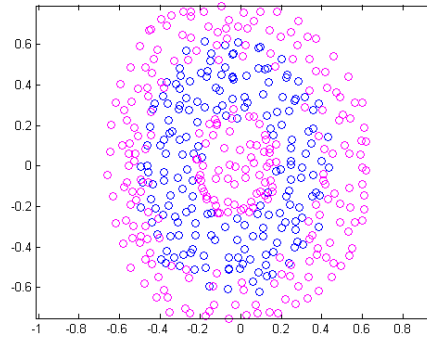


Figura 4.4: CIRCULO4

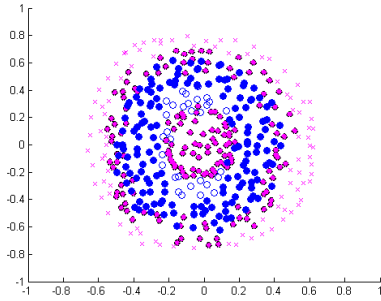
La tabla que se muestra a continuación se ven los resultados obtenidos usando el Núcleo basado en un polinomio de chebyshev, es decir

$$K_n(x, y) = \frac{\langle T, T' \rangle}{\sqrt{1 - xy}}$$

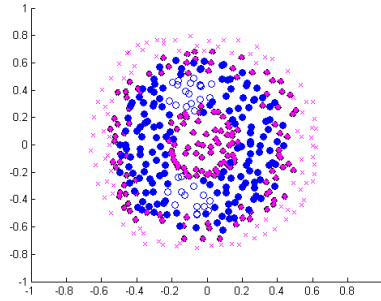
específicamente se utilizaron los tres primeros grados [ver [21]]

| Núcleo                  | Tiempo en seg | Vec.Sopt. | Clasifica |
|-------------------------|---------------|-----------|-----------|
| Chebyshev Primer grado  | 47.6508       | 319       | si        |
| Chebyshev Segundo grado | 38.5353       | 314       | si        |
| Chebyshev Tercer grado  | 38.8197       | 290       | si        |

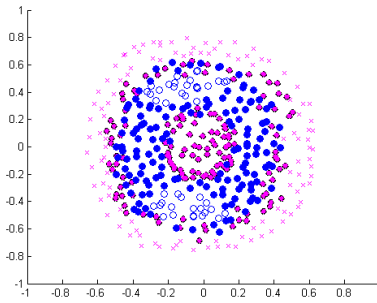
Cuadro 4.13: Núcleo basado en un polinomio de Chebyshev



Núcleo basado en un polinomio de Chebyshev primer grado



Núcleo basado en un polinomio de Chebyshev segundo grado



Núcleo basado en un polinomio de Chebyshev tercer grado

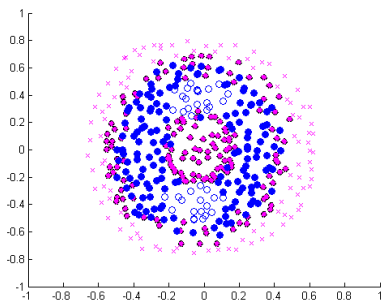
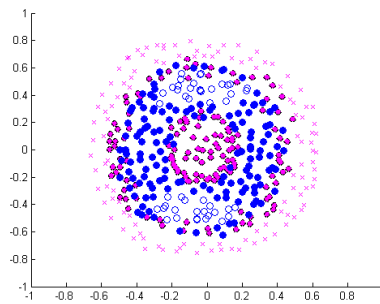
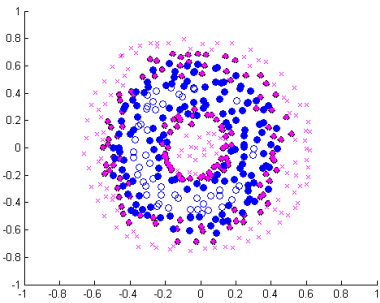
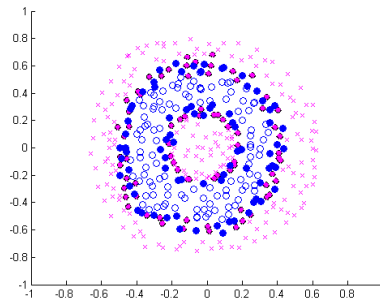
*Se puede observar que si hubo clasificación en los tres casos estudiados y la mejor fue en el caso de tercer grado con la menor cantidad de vectores soportes.*

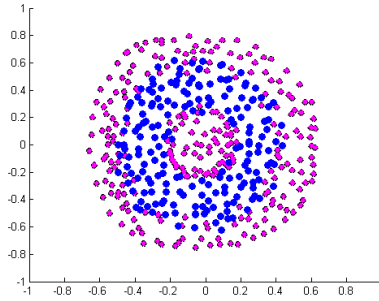
*En este caso el núcleo usado para clasificar es el núcleo basado en un polinomio de de Bessel que viene dado por*

$$K(x, y) = -Bessel_{\nu+1}^m(\sigma(|x - y|^2))$$

| Núcleo | parámetro   | Tiempo en seg | Vec.Sopt. | Clasifica |
|--------|-------------|---------------|-----------|-----------|
| Bessel | $\nu = -2$  | 67.0560       | 303       | si        |
| Bessel | $\nu = 0$   | 78.1375       | 284       | si        |
| Bessel | $\nu = 0,5$ | 82.5441       | 157       | si        |
| Bessel | $\nu = 0,1$ | 73.6818       | 270       | si        |
| Bessel | $\nu = 5$   | 81.5275       | 445       | no        |

Cuadro 4.14: Núcleo basado en un polinomio de Bessel

Núcleo basado en un polinomio de Bessel  $\nu = -2$ Núcleo basado en un polinomio de Bessel  $\nu = 0$ Núcleo basado en un polinomio de Bessel  $\nu = 0,1$ Núcleo basado en un polinomio de Bessel  $\nu = 0,5$

Núcleo basado en un polinomio de Bessel  $\nu = 5$ 

Se observa que para el caso en que el parámetro  $\nu$  está muy próximo a cero e incluso igual a cero es donde hubo clasificación, y la mejor fue para  $\nu = 0, 5$ , sin embargo en el caso en que  $\nu$  tiene valores muy altos o muy bajos el programa no clasifica.

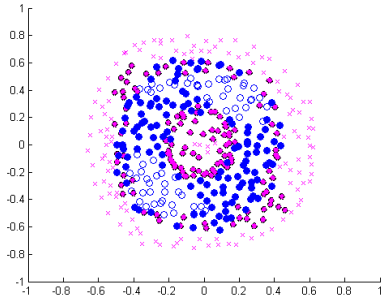
El núcleo usado para clasificar en la tabla siguiente es el núcleo basado en un polinomio de Hermite viene dado por:

$$K_n(x, y) = \frac{H_{n+1}(x)H_n(y) - H_n(x)H_{n+1}(y)}{e^{-xy}}$$

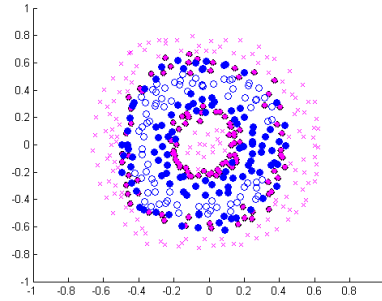
donde  $H_{n+1}(x)$  se obtiene por la fórmula de recurrencia para el polinomio de Hermite (3.3.7)

| Núcleo                | Tiempo en seg | Vec.Sopt. | Clasifica |
|-----------------------|---------------|-----------|-----------|
| Hermite primer grado  | 44.4823       | 254       | si        |
| Hermite segundo grado | 71.8480       | 216       | si        |
| Hermite tercer grado  | 58.2724       | 144       | si        |
| Hermite cuarto grado  | 86.5238       | 144       | si        |

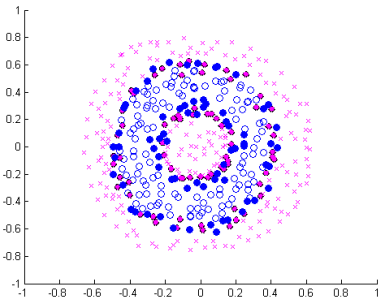
Cuadro 4.15: Núcleo basado en un polinomio de Hermite



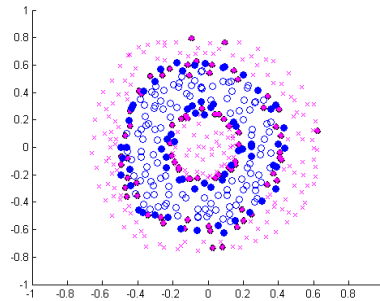
Núcleo basado en un polinomio de Hermite primer grado



Núcleo basado en un polinomio de Hermite segundo grado



Núcleo basado en un polinomio de Hermite tercer grado



Núcleo basado en un polinomio de cuarto grado

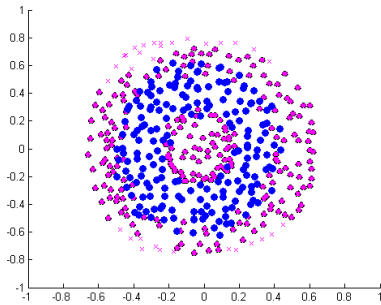
*Se puede ver que para el núcleo ortogonal de Hermite si hubo clasificación en los cuatro casos estudiados y se observa que a medida que se aumenta el grado del núcleo se tiene una mejor clasificación pues la cantidad de vectores soporte disminuye; sin embargo en este caso el mejor fue el núcleo de tercer grado, porque a pesar de tener la misma cantidad de vectores soporte que el de cuarto grado, el tiempo fue menor.*

*Los resultados mostrados en la siguiente tabla son realizados con el núcleo polinomial homogéneo dado por:  $K(x, y) = (x * y)^n$  y núcleo polinomial no homogéneo dado por  $K(x, y) = ((x * y) + c)^n$  donde  $n$  es el grado polinomio y  $c > 0$*

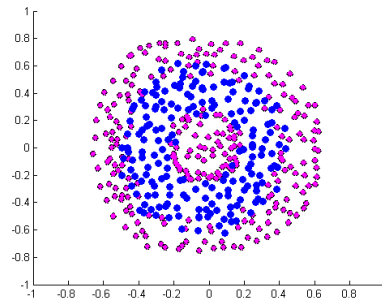


| Núcleo                  | Grado P | Parámetro | Tiempo  | Vec.Sopt. | Clasifica |
|-------------------------|---------|-----------|---------|-----------|-----------|
| Polinomial Homogéneo    | 10      |           | 35.9858 | 413       | no        |
| Polinomial Homogéneo    | 21      |           | 24.8312 | 445       | no        |
| Polinomial No Homogéneo | 5       | C=2       | 48.2221 | 202       | si        |
| Polinomial No Homogéneo | 23      | C=1       | 46.3973 | 146       | si        |

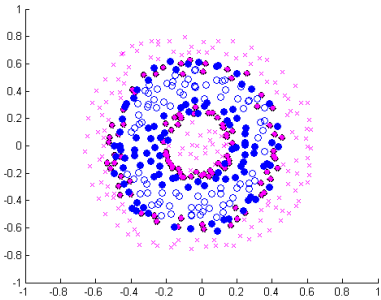
Cuadro 4.16: Núcleo polinomial homogéneo y no homogéneo



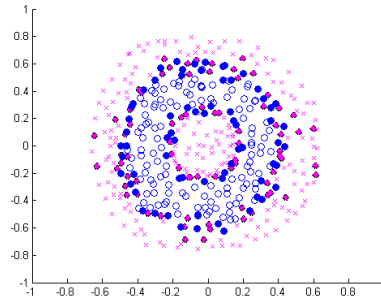
Núcleo polinomial homogéneo de grado 10



Núcleo polinomial homogéneo de grado 21



Núcleo polinomial no homogéneo de grado 5 c=2



Núcleo polinomial no homogéneo de grado 23 c=1

*Se observa que para el núcleo polinomial homogéneo no hubo clasificación, mientras que el núcleo polinomial no homogéneo si clasificó y a medida que el grado del polinomio aumenta se obtiene una mejor clasificación pues el número de vectores soportes disminuye.*

**Ejemplo 4.0.8** En este ejemplo se muestra una figura con forma de número nueve con una cantidad de 436 puntos a clasificar, donde 255 pertenecen a una clase y 181 pertenecen a la otra clase y tolerancia de  $10^{-3}$

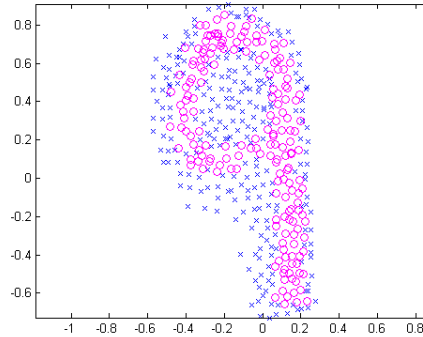


Figura 4.5: NUEVE

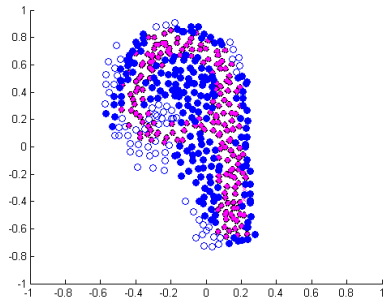
En tabla y las figuras que se muestra a continuación se ven los resultados obtenidos usando el núcleo basado en un polinomio de chebyshev, es decir

$$K_n(x, y) = \frac{\langle T, T' \rangle}{\sqrt{1 - xy}}$$

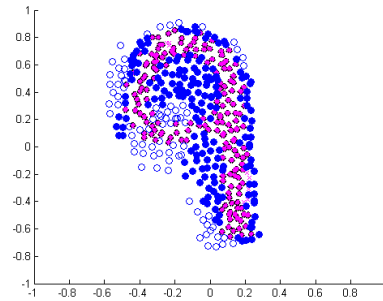
específicamente se utilizaron los tres primeros grados [ver [21]]

| Nucleo                  | Tiempo en seg | Vec.Sopt. | Clasifica |
|-------------------------|---------------|-----------|-----------|
| Chebyshev Primer grado  | 34.8317       | 365       | no        |
| Chebyshev Segundo grado | 34.7819       | 349       | no        |
| Chebyshev Tercer grado  | 44.6121       | 341       | no        |

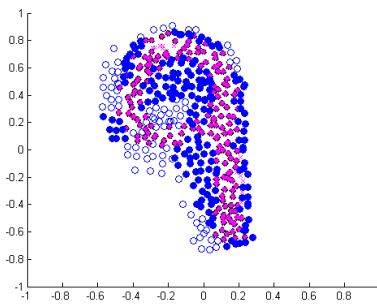
Cuadro 4.17: Núcleo basado en un polinomio de Chebyshev



Núcleo basado en un polinomio de Chebyshev primer grado



Núcleo basado en un polinomio de Chebyshev segundo grado



Núcleo basado en un polinomio de Chebyshev tercer grado

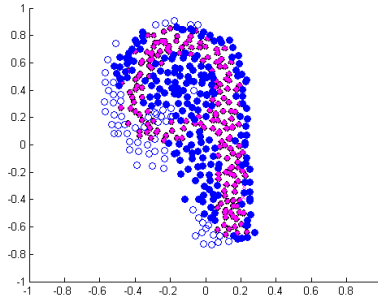
*Se puede observar que no hubo clasificación en ninguno de los tres casos estudiados.*

*En este caso el núcleo usado para clasificar es el núcleo de Bessel que viene dado por*

$$K(x, y) = -Bessel_{\nu+1}^n(\sigma(|x - y|^2))$$

| Núcleo | parámetro | Tiempo en seg | Vec.Sopt. | Clasifica |
|--------|-----------|---------------|-----------|-----------|
| Bessel | $\nu = 0$ | 71.7559       | 369       | no        |

Cuadro 4.18: Núcleo Ortogonal Bessel

Núcleo ortogonal Bessel  $\nu = 0$ 

Los resultados de la tabla arrojan que en el único caso en que el programa entra es cuando el parámetro  $\nu = 0$ , sin embargo este no clasifica. En el resto de los caso que se intento correr el programa no entro.

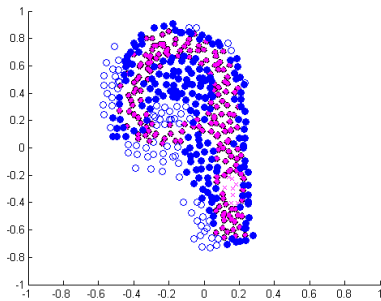
El núcleo usado para clasificar en la tabla siguiente es el núcleo basado en un polinomio de Hermite viene dado por:

$$K_n(x, y) = \frac{H_{n+1}(x)H_n(y) - H_n(x)H_{n+1}(y)}{e^{-xy}}$$

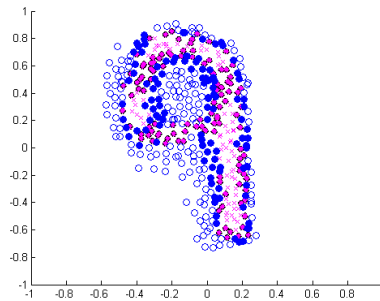
donde  $H_{n+1}(x)$  se obtiene por la fórmula de recurrencia para el polinomio de Hermite (3.3.7)

| Núcleo                | Tiempo en seg | Vec.Sopt. | Clasifica |
|-----------------------|---------------|-----------|-----------|
| Hermite segundo grado | 59.8881       | 352       | no        |
| Hermite tercer grado  | 57.6301       | 221       | si        |
| Hermite cuarto grado  | 205.5572      | 171       | si        |

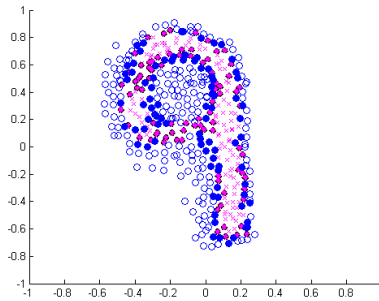
Cuadro 4.19: Núcleo basado en un polinomio de Hermite



Núcleo basado en un polinomio de Hermite segundo grado



Núcleo basado en un polinomio de Hermite tercer grado



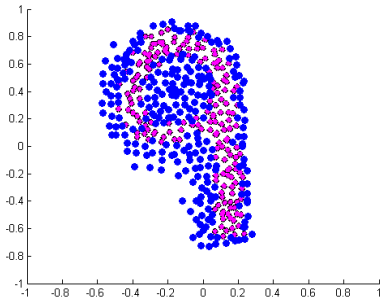
Núcleo basado en un polinomio de Hermite cuarto grado

*Los resultados obtenidos en la tabla y en las figuras muestran que para el núcleo de primer grado los datos no entraron al programa, mientras que para el de segundo grado no clasifica, a partir del núcleo de tercer grado si hubo clasificación donde el que mejor clasificó fue el núcleo ortogonal de Hermite de cuarto grado con menor cantidad de vectores soportes.*

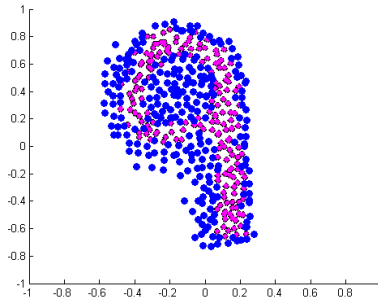
Los resultados mostrados en la siguiente tabla son realizados con el núcleo polinomial homogéneo dado por:  $K(x, y) = (x * y)^n$  y núcleo polinomial no homogéneo dado por  $K(x, y) = ((x * y) + c)^n$  donde  $n$  es el grado polinomio y  $c > 0$

| Núcleo                  | Grado P | Parámetro | Tiempo  | Vec.Sopt. | Clasifica |
|-------------------------|---------|-----------|---------|-----------|-----------|
| Polinomial Homogéneo    | 10      |           | 30.5176 | 436       | no        |
| Polinomial Homogéneo    | 21      |           | 53.2673 | 445       | no        |
| Polinomial No Homogéneo | 5       | C=2       | 43.0075 | 349       | no        |
| Polinomial No Homogéneo | 23      | C=1       | 52.4661 | 139       | si        |

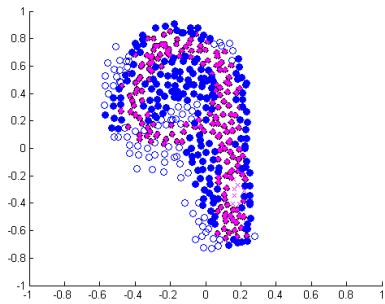
Cuadro 4.20: Núcleo polinomial homogéneo y no homogéneo



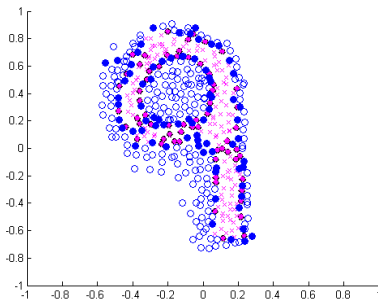
Núcleo polinomial homogéneo de grado 10



Núcleo polinomial homogéneo de grado 21



Núcleo polinomial no homogéneo de grado 5 c=2



Núcleo polinomial no homogéneo de grado 23 c=1

Se observa que para el núcleo polinomial homogéneo no hubo clasificación, mientras que el núcleo polinomial no homogéneo cuando el grado del polinomio es bajo no clasifica y si el grado del polinomio es alto si clasificó.

## Conclusión

Una vez analizados los datos a los cuales se les realizaron pruebas con diferentes funciones núcleos entre ellas resaltan: núcleo ortogonal de Chebyshev, núcleo ortogonal de Bessel, núcleo ortogonal de Hermite, núcleo polinomial homogéneo y núcleo polinomial no homogéneo. Para el análisis de los datos con cada función núcleo se consideró la comparación gráfica entre los resultados se pudo observar que para el núcleo de Bessel en los diferentes datos la mejor clasificación fue en el caso en que el parámetro  $\nu$  estaba muy próximo a cero, así mismo en el caso del núcleo de Chebyshev se pudo apreciar que dependiendo de la cantidad de puntos y de la forma en que se encuentren distribuidos los datos este núcleo se comportó de una manera favorable es decir si clasificó en algunos casos y la mejor clasificación fue el núcleo ortogonal de Chebyshev de segundo grado, del mismo modo en el núcleo ortogonal de Hermite se observó que fue uno de los mejores clasificadores pues en la mayoría de los diferentes datos se obtuvo excelente clasificación además se puede ver que a medida que se aumenta el grado del polinomio se obtenía

una mejor clasificación.

En el análisis de los problemas estudiados a lo largo de este trabajo, surgen de manera natural una serie de asuntos que pueden ser abordadas en el futuro tales como: la implementación de núcleos polinómicos pero esta vez basados en otros tipos de polinomios ortogonales no estandar entre estos podemos mencionar los polinomios para-ortogonales [18], los polinomios de Sobolev [4], los polinomios de variable discreta [2], y los  $q$ -polinomios hipergeométricos [19], todos ellos aplicados a maquinas de vectores de soporte, también se pueden hacer estudios de vectores soportes de regresión basados en polinomios ortogonales clásicos incluso en otros tipos de polinomios, así como el estudio de algunas de sus aplicaciones.



## Bibliografía

- [1] R Álvarez-Nodarse. On characterizations of classical polynomials. *Journal of computational and applied Mathematics*, 196(1):320–337, 2006.
- [2] Iván Carlos Area Carracedo. *Polinomios ortogonales de variable discreta: pares coherentes. Problemas de conexión*. PhD thesis, Universidade de Vigo, 1999.
- [3] T. S. Chihara. *An Introduction to Orthogonal Polynomials*. Gordon and Breach, New York, 1978.
- [4] Teresa Encarnacion Pérez Fernández. *Polinomios ortogonales respecto a productos de Sobolev: el caso continuo*. PhD thesis, Universidad de Granada, 1994.
- [5] W. Gautschi. *Orthogonal Polynomials: Computation and Approximation*. Oxford University Press, Oxford, 2004.

- 
- [6] Ya. L. Geronimus. *Orthogonal Polynomials: Estimates, Asymptotic Formulas, and Series of Polynomials Orthogonal on the Unit Circle and on an Interval*. Consultants Bureau, New York, 1961.
- [7] Thomas Hofmann, Bernhard Schölkopf, and Alexander J Smola. Kernel methods in machine learning. *The annals of statistics*, pages 1171–1220, 2008.
- [8] Hi L Krall and Orrin Frink. A new class of orthogonal polynomials: The bessel polynomials. *Transactions of the American Mathematical Society*, 65(1):100–115, 1949.
- [9] G. Lopez and H. Pijeira. *Polinomios ortogonales*. Decimocuarta escuela venezolana de matematica, Merida, 2001.
- [10] O. L. Mangasarian. Data mining via support vector machines. In E. W. Sachs and R. Tichatschke, editors, *System Modeling and Optimization XX*, pages 91–112. Kluwer Academic Publishers, Boston, 2003.
- [11] O. L. Mangasarian and D. R. Musicant. Lagrangian support vector machines. *J. Machines Learning Research*, 2000.
- [12] Sedat Ozer, Chi H Chen, and Hakan A Cirpan. A set of new chebyshev kernel functions for support vector machine pattern classification. *Pattern Recognition*, 44(7):1435–1447, 2011.
- [13] R. T. Rockafellar. *Convex Analysis*. Princeton landmarks in mathematics and physics. Princeton University Press, 1997.

- 
- [14] Laura Sánchez Gómez. Teoría de núcleos reproductivos en espacios de hilbert y aplicaciones a máquinas de soporte vectorial. 2012.
- [15] B. Schölkopf and A. Smola. *Learning with kernels*. The MIT press, Cambridge, Massachusetts, 2002.
- [16] B. Schölkopf and A. J. Smola. *Learning with Kernels: Support Vector Machines, Regularization, Optimization, and Beyond*. Adaptive computation and machine learning. MIT Press, 2002.
- [17] V. N. Vapnik. *The nature of Statistical Learning Theory*. Springer, New York, 1995.
- [18] Pablo González Vera. Polinomios para-ortogonales y fórmulas de cuadratura para integrandos periódicos. *Números*, (43):323–328, 2000.
- [19] R. Álvarez Nodarse y J. Hernández. Los q-polinomios hipergeométricos.
- [20] N. Ye, R. Sun, Y. Liu, and L. Cao. Support vector machine with orthogonal chebyshev kernel. In *18th International Conference on Pattern Recognition (ICPR'06)*, volume 2, pages 752–755. IEEE, 2006.
- [21] Ning Ye and Ruixiang sun. Support vector machine with ortogonal chebyshev. *Optimization Online*, page 2, 2006.
- [22] Xiu-ling Zhang, Shao-yu ZHANG, Guang-zhong TAN, and Wen-bao ZHAO. A novel method for flatness pattern recognition via least squares support vector regression. *Journal of Iron and Steel Research, International*, 19(3):25–30, 2012.

- [23] J. Zhao, G. Yan, B. Feng, W. Mao, and J. Bai. An adaptive support vector regression based on a new sequence of unified orthogonal polynomials. *Pattern Recognition*, 46(3):899–913, 2013.



## Códigos en MATLAB

### A.1. graficadepuntos.m

La función `graficadepuntos` permite crear el conjunto de datos; representado en matrices de las dos clases de puntos, donde  $A0$  y  $A1$ . Los puntos son ingresados a gusto del usuario mediante el uso del cursor. Además, luego esos puntos son graficados en el plano.

```
1 function [A0,A1,A]=graficadepuntos(p,Datos)
2 % p=ejes permite el ajuste de los ejes p=[-a b -c d].
3 axis(p)
4 sw=0;
5 A0=[];A1=[];
6 while sw==0
7     grid on
8     [x0 y0]=ginput(); %grafica de una clase puntos
```

```
9     A0=[A0;x0 y0];
10    plot(A0(:,1),A0(:,2),'o','MarkerSize',6,'...
        MarkerFaceColor','m','LineWidth',2,'MarkerEdgeColor'...
        , 'k')
11    axis(p)
12    grid on
13    [x1 y1]=ginput(); %grafica de una otra clase puntos
14    A1=[A1;x1 y1];
15    hold on
16    plot(A1(:,1),A1(:,2),'o','MarkerSize',6,'...
        MarkerFaceColor','b','LineWidth',2,'MarkerEdgeColor'...
        , 'k')
17    nt=size(A0)+size(A1);
18    NT=nt(1)
19    sw=input('satisfecho?')
20    %%si estas satisfecho con el tamano de la data.
21    end
22    A=[A0;A1];
23    save(Datos,'A0','A1','A') % guardar la data
```

## A.2. svmgeneral.m

Este algoritmo consigue la solución a un problema de programación cuadrática convexa que describe una SVM estándar planteado en [10]. El algoritmo realiza

una clasificación de los datos a través de una superficie no necesariamente lineal en el espacio de entradas, maximizando la distancia entre los planos separadores en un espacio de mayor dimensión, dentro del mismo se hace un llamado a la función núcleo  $KA$  de nombre `nucleomenu.m`. que se encarga de separar el conjunto de datos, hace uso del toolbox de optimización de MATLAB llamado `quadprog`, por ende todos los cálculos realizados dentro del algoritmo son hechos de manera tal, que el programa que describe esta dado por (1.0.22)

```
1 function [u,gamma,y,aux,FVAL,EXITFLAG,OUTPUT,LAMBDA,AA0,...
    AA1,tiempo]=svmgeneral(A0,A1,nu,p)
2 tic;
3 A=[A0;A1]; %data generada por generadoradedatos
4 n0=size(A0)
5 n1=size(A1)
6 d=[ones(n0(1),1);-ones(n1(1),1)];
7 D=diag(d);
8 KA=nucleomenu(A); %KA funcion nucleo menu
9 DAD=D*KA*D;
10 m=n0(1,1) + n1(1,1);
11 E=ones(m,1);
12 DE=D*E;
13 Id=eye(m,m);
14 O1=zeros(m,m);
15 O2=zeros(m,1);
16 %E1=ones(m,1);
17 B1=[-DAD DE -Id];
```

```
18 B2=[01 02 -Id];
19 AB=[B1;B2];
20 C=[-E;02];
21 O3=zeros(m,1);
22 H= eye(m,m); %H ES SIMETRICA DEFINIDA POSITIVA
23 H1=[H 02 01];
24 H2=[03' 0 03'];
25 H3=[01 02 01];
26 HH=[H1; H2; H3];
27 f= nu*[03; 0; E];
28 options=optimset('maxIter',10000,'Algorithm','interior-...
    point-convex');
29 [X,FVAL,EXITFLAG,OUTPUT,LAMBDA]=quadprog(HH,f,AB,C...
   ,[],[],[],[],[],options);
30 u=X(1:m);gamma=X(m+1);y=X(m+2:length(X));
31 %XX=[X(1:m+1);02];
32 aux=B1*X+E
33 %abs(aux(k))
34 %BUSQUEDA DE VECTORES SOPORTES
35 %para A0
36 cont0=0;
37 hold on
38 for k=1:n0(1)
39     if abs(aux(k))<1.e-2
40         cont0=cont0 + 1;
```



```
41     AA0(cont0,:)=A0(k,:);
42     plot(AA0(:,1),AA0(:,2),'x','MarkerSize',4,'...
         MarkerFaceColor','m','LineWidth',4,'...
         MarkerEdgeColor','m')
43     axis(p)
44     % pause
45     end
46 end
47 plot(AA0(:,1),AA0(:,2),'kx','LineWidth',4)
48 %BUSQUEDA DE VECTORES SOPORTES
49 %para A1
50 cont1=0;
51 for j=1:n1(1)
52     if abs(aux(n0(1)+j))<1.e-2
53 % j
54 % aaa=aux(j)
55         cont1=cont1 + 1;
56         AA1(cont1,:)=A1(j,:);
57         plot(AA1(:,1),AA1(:,2),'o','MarkerSize',4,'...
             LineWidth',4)
58         axis(p)
59         % pause
60     end
61 end
62 %cont1
```

```
63 % AA0
64 % AA1
65 plot(A0(:,1),A0(:,2),'xm')
66 %'MarkerFaceColor','m'
67 hold on %grafica de ambos vectores
68 plot(A1(:,1),A1(:,2),'ob')
69 plot(AA0(:,1),AA0(:,2),'x','MarkerSize',4,'...
    MarkerFaceColor','m','LineWidth',4,'MarkerEdgeColor',...
    'm')
70 plot(AA1(:,1),AA1(:,2),'o','MarkerSize',4,'LineWidth',4)...
    %,'MarkerEdgeColor','b','MarkerFaceColor','b')
71 tiempo=toc
72 %para guardar la data
73 Datos=input('ingrese el nombre de la data: ','s')
74 save(Datos,'u','gamma','y','aux','FVAL','EXITFLAG','...
    OUTPUT','LAMBDA','AA0','AA1','tiempo')
```

### A.3. nucleomenu.m

Este algoritmo describe las diferentes funciones núcleos empleadas en los datos, permitiendo escoger el núcleo que el usuario desee usar.

```
1 function KA=nucleomenu(A)
2
```

```
3 choice = menu('menu', 'Chebyshev', 'Bessel', 'Hermite', '...
    Phomogeneo', 'Pno-homogeneo')
4 n=size(A);
5 switch choice
6
7 % Nucle Chebyshev
8     case 1
9         for i=1:n(1)
10            for j=1:n(1)
11                KA(i,j)=1/(sqrt(1-(A(i,:)*A(j,:)'))); % primero ...
                    elemento
12                KA(i,j)=(1+(A(i,:)*A(j,:)'))/(sqrt(1-(A(i,:)*A(j...
                    ,:)'))); %segundo elemento
13                KA(i,j)=(1+(A(i,:)*A(j,:)'+(2*(A(i,:)).^2-1)...
                    *(2*(A(j,:)').^2)))/(sqrt(1-(A(i,:)*A(j,:)')));...
                    % tercer elemento
14            end
15        end
16
17 %nucleo Bessel
18     case 2
19         nnu=input('ingrese el valor del parametro nnu:') %nnu es ...
                    un numero real
20         sigma=2*norm(std(A))^2;
21         for i=1:n(1)
```

```

22     for j=1:n(1)
23         aa=sigma*(norm(A(i,:)-A(j,:))^2);
24         KA(i,j)=-besselj(nnu,aa);
25     end
26 end
27
28 %nucleo Hermite
29     case 3
30         for i=1:n(1)
31             for j=1:n(1)
32                 KA(i,j)=(((4*(A(i,:)).^2-2)*(2*A(j,:)'))-((2*A(i,:))...
                    *(4*(A(j,:)).^2-2)))/(exp(-A(i,:)*A(j,:))); % ...
                    primer element
33                 KA(i,j)=((8*(A(i,:)).^3-(12*A(i,:)))*(2*(A(j,:)).^2-4)...
                    )-((2*(A(i,:)).^2-4)*(8*(A(j,:)).^3-(12*A(j,:))))...
                    /(exp(-A(i,:)*A(j,:))); % segundo termino
34                 KA(i,j)=((16*(A(i,:)).^4-(48*(A(i,:)).^2+12)*(8*(A(j...
                    ,:)).^3-(12*A(j,:))))-((8*(A(i,:)).^3-(12*A(i,:))...
                    *(16*(A(j,:)).^4-(48*(A(j,:)).^2+12)))/(exp(-A(i...
                    ,:)*A(j,:))); % Tercer termino
35                 Hij1=(32*A(i,:).^5 - 160*A(i,:).^3 + 120*A(i,:))*(16*A(...
                    j,:).^4 - 48*(A(j,:)).^2 + 12);
36                 Hij2=(16*A(i,:).^4 - 48*A(i,:).^2 + 12)*(32*(A(j,:))'...
                    .^5 - 160*(A(j,:))'.^3 + 120*A(j,:))');
37                 KA(i,j)=(Hij1-Hij2)/(exp(-(A(i,:))*(A(j,:))'));

```

```
38     end
39     end
40
41 %nucleo polinomial homogeneod
42     case 4
43 d=input('ingrese el grado del polinomio:') %d es un numero...
    natural que es el grado del polinomio
44 for i=1:n(1)
45     for j=1:n(1)
46         KA(i,j)=(A(i,:)*(A(j,:))')^(d);
47     end
48 end
49
50 %nucleo polinomial no-homogeneo
51     case 5
52 d=input('ingrese el grado del polinomio:') %d es un numero...
    natural que es el grado del polinomio
53 c=input('ingrese el valor del parametro libre c:') % c es ...
    un parametro libre mayor que 0
54 KA=(A*A'+c).^d;
55
56 end
```