Centro de Investigación en Matemáticas A.C

CIMAT

Algunas propiedades del kernel empleado en Máquinas de Soporte Vectorial para clasificación.

TESIS
que para obtener el grado de
Maestro en Ciencias

con especialidad
Matemáticas Aplicadas

PRESENTA:
Héctor Damián Méndez Dávila

ASESORES:
Dr. Arturo Hernández Aguirre
Dr. Miguel Angel Moreles Vázquez

Agosto del año 2003 Guanajuato, Gto. México

Agradecimientos

A mis padres Héctor y Ofelia, por su cariño, confianza y por su incondicional apoyo en todas mis decisiones.

A mis hermanos Jairo y Armando, por animarme en todo momento a pesar de la distancia.

A mis compañeros durante la maestría, de entre ellos y sin dejar de lado a los demás, especialmente a Lenin, Laura, César, Cynthia, Fidel, Geiser y Manuel quienes fueron los más cercanos a mí por su amistad y confianza.

A mis asesores, el Dr. Arturo Hernández y el Dr. Miguel Angel Moreles por su apoyo y su guía a través de esta tesis.

A toda la comunidad del CIMAT pues nunca olvidaré el maravilloso ambiente que viví en esta institución.

A Dios por permitirme terminar esta tesis y ayudarme a no desfallecer durante este período de mi vida.

Resumen

En el primer capítulo se introduce el concepto de SVMs y se plantean los objetivos de la tesis.

El segundo capítulo presenta una breve introducción al método de SVMs, cómo trabajan resolviendo el caso linealmente separable y no linealmente separable y algunos de los conceptos básicos que serán de utilidad a lo largo de la tesis, como el concepto de kernel.

En el tercer capítulo se presentan algunas propiedades generales del espacio de rasgos y se analizan algunas características y propiedades de algunos de los kernels más comunmente usados en las SVMs a través de las funciones ϕ y los espacios de rasgos que definen implícitamente. Se presenta también un enfoque nuevo acerca de cómo construir kernels a partir de las funciones de separación que se requieran.

Por ultimo, en el cuarto capítulo se presenta cómo construir kernels con polinomios ortogonales y la construcción de un kernel con los polinomios de Tchevyshev además de kernels con los polinomios de Hermite y con series de Fourier a partir de algunas ideas sugeridas por Vapnik [1]. Además se presentan los resultados de algunos experimentos realizados con estos kernels, se calula el error de generalización y se compara con una de las cotas de éste error presentada en el capítulo 2.

Índice general

Índice de figuras

Índice de cuadros

Capítulo 1

Introducción

En Inteligencia Artificial, uno de los primeros pasos para lograr máquinas inteligentes es programar un método de aprendizaje. El problema de aprendizaje consiste en, dada una muestra de tamaño limitado, encontrar una descripción concisa de los datos. Cuando los datos son muestras de patrones de entrada y de salida, una descripción concisa de los datos se puede dar en forma de una función que pueda producir datos de salida a partir de datos de entrada (función de decisión). Ejemplos donde se presenta este problema incluyen clasificación de letras y dígitos escritos a mano, clasificación de noticias en una agencia de noticias o la clasificación de páginas Web a partir de su contenido.

Las SVMs (Máquinas de Soporte Vectorial) son una técnica para resolver el problema de aprendizaje motivadas por los resultados logrados por Vapnik [1] y sus colaboradores en la teoría estadística del aprendizaje, la cual motivó a muchos investigadores a enfocarse en la teoría del aprendizaje y su potencial en el diseño de nuevos algoritmos.

La técnica de SVMs buscan encontrar una región de decisión (o una aproximación a una función para el caso de regresión) a través de planos (o hiperplanos) de separación. Sin embargo, cuando el problema es no linealmente separable (esto es, que no pueda resolverse con un hiperplano de separación), se puede encontrar una región de decisión no lineal (con funciones de separación no lineales) incorporando al método una función kernel. La razón de esto es que el kernel definirá un mapeo implícito a un espacio donde el problema pueda resolverse con hiperplanos de separación (si el kernel es adecuado).

Este procedimiento de sustituir los productos punto por kernels es la más impor-

tante e interesante característica de las SVMs; ha despertado gran interés en muchas áreas y se ha buscado exportarlo a otros métodos.

En esta tesis se estudian algunas características de los kernels más comunes en problemas de clasificación, así como la construcción de nuevos kernels usando polinomios ortogonales y algunas técnicas presentadas por Vapnik.

En el capítulo 2 se presentan los preliminares y las bases para entender el problema de clasificación y cómo se resuelve con el método de SVMs así como la introducción del concepto de kernel.

En el capítulo 3 se muestran algunos resultados con respecto a los espacios de rasgos de kernels y específicamente se analizan los kernels polinomial sencillo, polinomial y gaussiano.

Por último, en el capítulo 4 se muestra la construcción de algunos nuevos kernels así como un breve análisis de ellos. Se presenta además algunos experimentos con estos kernels y se comparan los resultados con una cota del error presentada en el capítulo 2.

Uno de los objetivos de esta tesis es presentar una perspectiva diferente para el uso de kernels en SVMs orientado a la función de separación que generan, esto es, si se puede obtener información a priori del tipo de función de separación que se espera, la elección de un kernel adecuado será más sencilla. El analisis de los kernels polinomial sencillo, polinomial y gaussiano que se presentan en el capítulo 3 y en especial la última sección del mismo están orientados a este respecto.

Otro de los objetivos es presentar algunos resultados y propiedades de kernels para clarificar un poco la forma en que trabajan en el algoritmo de SVMs. Uno de los resultados en este respecto es el hecho de que la imagen de un conjunto finito bajo alguna función ϕ que defina un kernel fijo está contenida en un espacio cuya dimensión es la misma para cualquier ϕ que defina ese kernel. Esta dimensión es la que se refiere en varios libros como "dimensión del feature space", aunque de forma teórica. Este resultado se presenta en el capítulo 3.

Como última aportación, presentamos nuevos kernels para lograr una mayor variedad de kernels y en ciertos casos en el tipo de regiones de separación que se puedan obtener, aunque sólo se presenta un breve análisis de ellos.

Capítulo 2

Máquinas de soporte vectorial

En este capítulo se presenta una breve introducción al método de SVMs y algunos de los conceptos básicos que serán de utilidad a lo largo de la tesis, como el concepto de kernel. El principal objetivo es mostrar cómo el método resuelve el caso linealmente separable y cómo se incorporan los kernels para resolver el caso no linealmente separable.

2.1. Aprendizaje por medio de muestras

El uso de las computadoras ha facilitado en gran medida la resolución de problemas gracias a la rapidez y exactitud con la que realizan operaciones en la computadora.

Recientemente se ha buscado la manera de resolver por medio de métodoa computacionales problemas en donde se requiere alguna información de datos desconocidos tomando como referencia información de datos conocidos. Un ejemplo de estos problemas puede ser reconocer letras escritas a mano. Este problema no se puede abordar con programación clásica debido a la infinidad de formas diferentes en que una letra puede ser escrita. En este caso se quisiera programar un método computacional para encontrar alguna manera de distinguir entre las letras escritas por cualquier persona a partir de algunos ejemplos particulares. En otras palabras, se busca un método que sea capaz de "aprender" a través de ejemplos guardando cierta analogía a la forma en que lo hace el ser humano. A esta forma de atacar los problemas se le llama Metodología del aprendizaje.

Dentro de esta área nos interesa trabajar con *el método de aprendizaje supervi-*

sado principalmente con *el problema de clasificación.*

El método de aprendizaje supervisado consiste en encontrar una regla general que explique datos a partir de una muestra de tamaño limitado. En detalle, se da como base al método un conjunto de parejas de datos (datos de entrada y de salida) llamados *datos de entrenamiento* o *conjunto de entrenamiento*, y la tarea del método es, encontrando una relación entre los datos de entrada con sus respectivos datos de salida, tratar de dar una salida adecuada para cualquier dato de entrada que se le proporcione (no importando que no sea parte de los datos de entrenamiento).

En *el problema de clasificación* se tiene un conjunto de datos que se quiere clasificar en una de varias clases. Para usar el método de aprendizaje supervisado en este problema se usará el conjunto de datos a clasificar como datos de entrada y los datos de salida serán las clases correspondientes de entre las posibles clases.

Nos interesa el caso en que los datos de entrada pueden representarse por puntos en $\mathbb{R}^n$ (usualmente en $\mathbb{R}^2$) y además en que sólo hay 2 clases posibles (*clasificación binaria*).

2.2. Máquinas de soporte vectorial

El algoritmo de *máquinas de soporte vectorial* (support vector machine o SVM) es un método de aprendizaje supervisado para resolver varios tipos de problemas de aprendizaje. Con el fin de describir este algoritmo veremos cómo se aplica para resolver el problema de clasificación binario. Para resolverlo, este algoritmo busca una función que asigne a cada punto un valor el cual debe ser el mismo para puntos que pertenezcan a una misma clase y diferente para puntos que pertenezcan a diferente clase. Por convención (y para simplificar cálculos posteriores) se busca que estos valores sean 1 y -1.

Denotaremos:

$$\begin{aligned} X \subset \mathbb{R}^n & \quad \text{Espacio de entrada} \\ Y = \{-1, 1\} & \quad \text{Espacio de salida (clases)} \\ S = \{(x_1, y_1), \ldots, (x_i, y_i), \ldots, (x_l, y_l)\} & \quad \text{Conjunto de entrenamiento} \end{aligned}$$

La función encontrada por el método de SVM dividirá el espacio de entrada en dos (o más) regiones, cada una de las cuales le corresponderá una clase, ya sea $+1$ o

-1. Usualmente, esta función tiene la forma

$$d(x) = sign(f(x)) \tag{2.1}$$

para alguna función real f.

Llamaremos a d la *función de decisión* y a f la *función de separación*, ya que para que h clasifique correctamente los puntos de entrada, la curva definida por $f(x) = 0$ necesariamente debe separar los puntos de la clase $+1$ de los de la clase -1.

A continuación veremos cómo se aplica el método de SVM al *caso linealmente separable*, el cual será la base para resolver el *caso no linealmente separable.*

2.2.1. Caso linealmente separable

Decimos que los datos de entrada son *linealmente separables* si existen $w \in \mathbb{R}^n$ y $b \in \mathbb{R}$ tales que la función de decisión dada por la ecuación (2.1), donde $f(x) = \langle w, x \rangle + b$, es tal que clasifica correctamente todos los datos de entrenamiento $x_i \in X$, es decir $d(x_i) = y_i$. En este caso $f(x) = 0$ define un hyperplano en $\mathbb{R}^n$ el cual separa las regiones de decisión. La figura 2.1 muestra un ejemplo de f en $\mathbb{R}^2$.

Observemos que para un conjunto de entrenamiento S dado, puede haber varios w y b para los cuales la función de decisión correspondiente d clasifica correctamente los puntos de entrenamiento.

Una forma de distinguir entre las posibles funciones de desición que clasifican correctamente los datos de entrada es definiendo el margen de f respecto a los datos de entrada.

Definiremos el margen de f con respecto al mínimo de las distancias entre el hiperplano $f(x) = 0$ (llamado *hiperplano de separación*) y los x_i. Distinguiremos dos tipos de margen:

$$\begin{aligned} \text{Margen funcional} &= min\ y_i(\langle w, x_i \rangle + b) \\ \text{Margen geométrico} &= min\ y_i \frac{\langle w, x_i \rangle + b}{\|w\|} \end{aligned} \tag{2.2}$$

El margen funcional, a diferencia del margen geométrico, cambia bajo un reescalamiento del plano: $\langle \lambda w, x \rangle + \lambda b = 0$ para $\lambda \in \mathbb{R}^+$, por lo tanto podemos tomar el

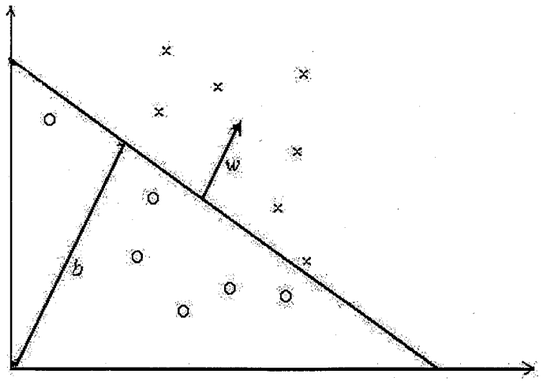

Figura 2.1: hiperplano de separación f en $\mathbb{R}^2$ (recta) que separa la clase de las $\times$ de la de las $\circ$

margen funcional igual a 1 sin afectar el margen geométrico, esto es

$$\langle w, x^+ \rangle + b = +1 \tag{2.3}$$
$$\langle w, x^- \rangle + b = -1 \tag{2.4}$$

donde x^+ y x^+ son los puntos más cercanos al hiperplano de separación $\langle w, x \rangle + b = 0$ de la clase $+1$ y -1 respectivamente. Luego, sustituyendo las ecuaciones (2.3) y (2.4) en la ecuación (2.2) obtenemos

$$\gamma = \frac{1}{\|w\|} \tag{2.5}$$

donde γ es el margen geométrico.

En la figura 2.2 se muestra un ejemplo de margen geométrico en $\mathbb{R}^2$.

La tarea del algoritmo de SVM no sólo es encontrar un hiperplano $f(x) = \langle w, x \rangle + b$ que clasifique correctamente los datos de entrada (usando d dada por la ecuación (2.1)), también se busca que tal f tenga el menor *error de generalización* posible, esto es, que sea capaz de clasificar correctamente la mayor cantidad de puntos diferentes a los puntos de entrenamiento. Formalmente, llamaremos error de generalización a la probabilidad de que un vector sea incorrectamente clasificado.

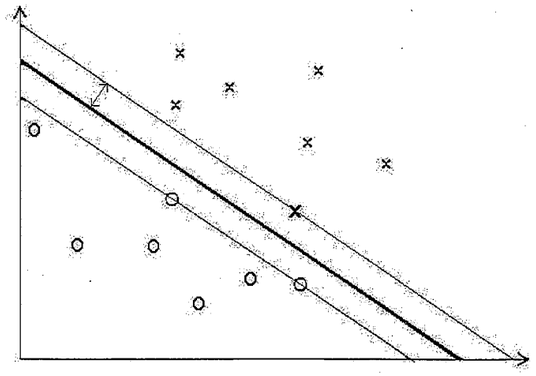

Figura 2.2: El margen geométrico, delimitado por las líneas más delgadas, queda determinado por los vectores mas cercanos a la función de separación (la recta más gruesa)

Introduciremos el parámetro h que se conoce como *dimensión VC* (Vapnik-Chervonenkis), el cual es un indicador acerca de la capacidad de separación de la función f. Entonces, para el caso cuando f es un hiperplano, Vapnik [1] establece la siguiente cota para h:

$$h \leq \text{mín}\{[\frac{D^2}{\gamma^2}], m_0\} + 1 \tag{2.6}$$

donde m_0 es la dimensión del espacio en el que se encuentran los vectores de entrada y D es el diámetro de la bola más pequeña que contiene a dichos vectores.

Ahora, Vapnik [1] establece que, con probabilidad $1 - \eta$, se tiene la siguiente cota para el error de generalización:

$$Error_{gen} = \frac{m}{l} + \frac{\varepsilon}{2}\left(1 + \sqrt{1 + \frac{4m}{l\varepsilon}}\right) \tag{2.7}$$

donde

$$\varepsilon = 4\frac{h(\ln \frac{2l}{h} + 1) - \ln \frac{\eta}{4}}{l} \tag{2.8}$$

y m es el número de vectores de entrenamiento que no fueron clasificados correctamente por la función de decisión encontrada por el método.

De (2.6) y (2.7) observamos que para minimizar el error de generalización, es necesario maximizar el margen geométrico γ. De la ecuación (2.5), esto equivale a minimizar $\|w\|$.

Luego, el problema de las SVMs se reduce a resolver el siguiente problema de programación cuadrática:

$$\begin{aligned} &minimizar_{w,b} \quad \langle w, w\rangle \\ &\quad sujeto\ a \quad y_i(\langle w, x_i\rangle + b) \geq 1 \quad i = 1, \ldots, l \end{aligned}$$

donde las restricciones vienen de combinar las ecuaciones (2.3), (2.4) y el hecho de que $d(x_i) = y_i$ cuando x_i está bien clasificado (l es el número de datos de entrada).

Para resolverlo, obtenemos su Lagrangiano:

$$L(w, b, \alpha) = \frac{1}{2}\langle w, w\rangle - \sum_{i=1}^{l} \alpha_i [y_i(\langle w_i, x_i\rangle + b) - 1] \tag{2.9}$$

donde $\alpha_i \geq 0$ son los *multiplicadores de Lagrange*. Luego, por el Teorema de Lagrange debe cumplirse que

$$\frac{\partial L(w, b, \alpha)}{\partial w} = w - \sum_{i=1}^{l} y_i \alpha_i x_i = 0$$

$$\frac{\partial L(w, b, \alpha)}{\partial b} = \sum_{i=1}^{l} y_i \alpha_i = 0$$

de donde

$$w = \sum_{i=1}^{l} y_i \alpha_i x_i \tag{2.10}$$

$$\sum_{i=1}^{l} y_i \alpha_i = 0 \tag{2.11}$$

Sustituyendo las ecuaciones (2.10) y (2.11) en la ecuación (2.9) (llamado también

problema primal), obtenemos el siguiente problema de optimización cuadrática (llamado *problema dual*), el cual es equivalente a resolver el problema original

$$\begin{aligned} &maximizar \quad W(\alpha) = \sum_{i=1}^{l} \alpha_i - \frac{1}{2} \sum_{i,j=1}^{l} y_i y_j \alpha_i \alpha_j \langle x_i, x_j\rangle \\ &sujeto\ a \quad \sum_{i=1}^{l} y_i \alpha_i = 0 \\ &\qquad\quad \alpha_i \geq 0 \qquad i = 1, \ldots, l \end{aligned} \tag{2.12}$$

Resolviendo el problema dual obtenemos las α_i optimas (α_i^*) para así, de la ecuación (2.10), obtener w^* (w óptima)

$$w^* = \sum_{i=1}^{l} y_i \alpha_i^* x_i \tag{2.13}$$

Para obtener b óptima (b^*) usamos el hecho de que la distancia del plano $\langle w, x\rangle + b$ al origen es $\frac{b}{\|w\|}$ y las ecuaciones (2.3) y (2.4) para obtener:

$$b^* = -\frac{\text{máx}_{y_i=-1}(\langle w^*, x_i\rangle) + \text{mín}_{y_i=1}(\langle w^*, x_i\rangle)}{2} \tag{2.14}$$

Entonces, el hiperplano de separación óptimo queda dado por

$$f(x) = \langle w^*, x\rangle + b^* = \sum_{i=1}^{l} y_i \alpha_i \langle x_i, x\rangle + b^* \tag{2.15}$$

Y la función de decisión queda dada por la expresión (2.1).

La solución del problema dual 2.12 satisface además las *condiciones complementarias de Karush-Kuhn-Tucker* dados en [1]:

$$\alpha_i^* [y_i(\langle w_i^*, x_i\rangle + b^*) - 1] = 0 \tag{2.16}$$

Los vectores x_i para los cuales las correspondientes α_i^* son diferentes de cero son llamados *vectores de soporte*. Por las condiciones complementarias de Karush-Huhn-Tucker (ecuación (2.16)), los vectores de soporte caen justamente sobre el margen de la función de separación. En la figura 2.2, los vectores de soporte aparecen resaltados.

2.2.2. Caso no linealmente separable

Para resolver el *caso no linealmente separable* quisieramos encontrar un mapeo $\phi : X \to F$ que mapee los puntos de entrada a un espacio F (llamado feature space o espacio de rasgos) en el cual el problema sea linealmente separable y así aplicar el procedimiento anterior.

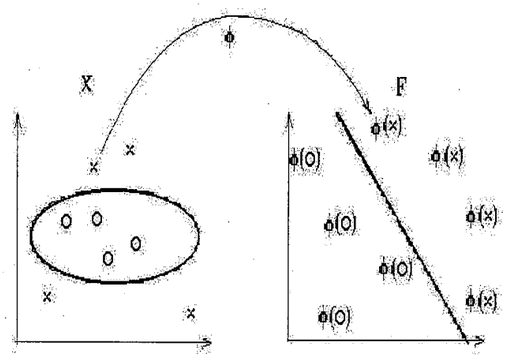

Figura 2.3: Visualización del procedimiento para resolver el caso no linealmente separable, mapeando al espacio F a través de ϕ deonde se transforma en un problema linealmente separable

Supongamos que tenemos un mapeo ϕ que cumpla las características anteriores, entonces tomamos las imágenes de los puntos de entrada originales como los nuevos puntos de entrada y aplicamos el método anterior trabajando en F, esto es, hay que resolver el problema dual:

$$\begin{array}{rl} \textit{maximizar} & W(\alpha) = \sum_{i=1}^{l} \alpha_i - \frac{1}{2} \sum_{i,j=1}^{l} y_i y_j \alpha_i \alpha_j \langle \phi(x_i), \phi(x_j) \rangle \\ \textit{sujeto a} & \sum_{i=1}^{l} y_i \alpha_i = 0 \\ & \alpha_i \geq 0 \qquad i = 1, \ldots, l \end{array}$$

Y obtener w^* y b^* a partir de α^* de manera análoga

$$w^* = \sum_{i=1}^{l} y_i \alpha_i^* \phi(x_i) \tag{2.17}$$

$$b^* = -\frac{\text{máx}_{y_i=-1}(\langle w^*, \phi(x_i) \rangle) + \text{mín}_{y_i=1}(\langle w^*, \phi(x_i) \rangle)}{2} \tag{2.18}$$

Observemos que $w^* \in F$.

Entonces obtenemos una función de separación no lineal f para $x \in X$:

$$f(x) = \langle w^*, \phi(x) \rangle + b^* = \sum_{i=0}^{l} \alpha_i^* y_i \langle \phi(x_i), \phi(x) \rangle + b^* \tag{2.19}$$

La cual define una función de decisión d dada por la ecuación (2.1).

Observemos que tanto en el planteamiento del problema dual como en la expresión de la función de separación f óptima (parte derecha de la ecuación (2.19)) sólo hacemos referencia a los datos de entrada en F, $\phi(x_i)$, dentro de productos punto. Esto da lugar a la importación del concepto de *kernel*.

2.3. El kernel trick en las SVMs

Definición 1. *Un kernel es una función real en dos variables $K : X \times X \to \mathbb{R}$, para un espacio X, tal que $\forall x, z \in X$, se tiene que $k(x, z) = \langle \phi(x), \phi(z) \rangle$ para algún mapeo $\phi : X \to F$ y F un espacio de Hilbert.*

Usando el concepto de kernel podemos sustituir los productos punto por el kernel en el problema dual:

$$\begin{array}{rl} \textit{maximizar} & W(\alpha) = \sum_{i=1}^{l} \alpha_i - \frac{1}{2} \sum_{i,j=1}^{l} y_i y_j \alpha_i \alpha_j k(x_i, x_j) \\ \textit{sujeto a} & \sum_{i=1}^{l} y_i \alpha_i = 0 \\ & \alpha_i \geq 0 \qquad i = 1, \ldots, l \end{array} \tag{2.20}$$

Y en la solución en (2.19):

$$f(x) = \sum_{i=0}^{l} \alpha_i^* y_i k(x_i, x) + b^* \tag{2.21}$$

donde $b^* = -\dfrac{\text{máx}_{y_i=-1}(\sum_{j=0}^{l} y_j \alpha_j^* k(x_j, x_i)) + \text{mín}_{y_i=1}(\sum_{j=0}^{l} y_j \alpha_j^* k(x_j, x_i))}{2}$

Y así obtenemos la función de decisión d nuevamente por la expresión (2.1).

Notamos que para resolver el caso no linealmente separable no se necesita conocer ϕ ni el espacio de rasgos F siempre que tengamos el kernel K, lo cual ahorra muchos cálculos reduciendo el problema a resolver el problema (2.20) para α y sustituir en (2.21). Esta sustitución del producto punto por la función kernel, se conoce como *kernel trick*.

Al usar el kernel trick, la función de separación f dada por la expresión (2.21) resulta usualmente no lineal. La naturaleza de la función f queda determinada por el kernel que se utilice para resolver el problema, ya que cada kernel determina un subespacio en el espacio de funciones (llamado *espacio de hipótesis*) dentro de el cual el algoritmo de SVMs elige la función de separación óptima f. Por ejemplo, el kernel $k(x, z) = \langle x, z \rangle$ toma como espacio de hipótesis el conjunto de hiperplanos en X pues el usar este kernel en el algoritmo equivale a resolver el caso linealmente separable.

Ahora, ya que puede haber muchos kernels con los que el algoritmo clasifique correctamente los datos de entrada, nos gustaría tener una manera de comparar los errores de generalización que generan. Para ello, Vapnik establece que, con una probabilidad de $1 - \eta$, se cumple la siguiente cota para el error de generalización del algoritmo de SVM con el kernel k:

$$Error_{gen} \leq \frac{1}{l - nsv}\left(nsv \log_2 \frac{e \cdot l}{nsv} + \log_2 \frac{l}{\eta}\right) \tag{2.22}$$

donde l es el número de vectores de entrada, nsv es el número de vectores de soporte y $\log_2$ es el logaritmo base 2. La expresión (2.22) se puede encontrar en [6].

Luego, de (2.22) deducimos que para disminuir el error de generalización hay que buscar disminuir el número de vectores de soporte. Es por eso que cuando se tienen varios kernels que clasifican satisfactoriamente los vectores de entrada se toma el kernel con el que se genera el menor número de vectores de soporte.

2.4. Más acerca de kernels

Aunque la definición de kernel lo relaciona con una función $\phi : X \to F$, no es necesario conocer ϕ o F para reconocer si una función k es un kernel. Una forma de verificar si una función k es un kernel es usando el Teorema de Mercer, el cual nos dice que si k cumple que

$$\int_{X \times X} k(x, z) g(x) g(z) dx dz \geq 0 \qquad \forall g \in L_2(X) \tag{2.23}$$

entonces k es un kernel, donde $L_2(X)$ es el conjunto de funciones de cuadrado integrable en X, esto es, el conjunto de las funciones g tal que

$$\|g\|_{L_2} = \int_X f(x)^2 dx < \infty$$

Además, a partir de uno o más kernels se pueden construir nuevos kernels mediante la siguiente proposición:

Proposición 1. *Sean K_1 y K_2 kernels sobre $X \times X$, $X \subseteq \mathbb{R}^n$, $a \in \mathbb{R}^+$, $f(\cdot)$ una función que tome valores reales en X, $\phi : X \to \mathbb{R}^m$, K_3 un kernel sobre $\mathbb{R}^m \times \mathbb{R}^m$ y B una matriz $n \times n$ positiva semi-definida. Entonces, las siguientes funciones son kernels:*

1. $K(x, z) = K_1(x, z) + K_2(x, z)$.
2. $K(x, z) = aK_1(x, z)$.
3. $K(x, z) = K_1(x, z)K_2(x, z)$.
4. $K(x, z) = f(x)f(z)$.
5. $K_3(\phi(x), \phi(z))$.
6. $x^T B z$.

Una demostración de esta proposición se puede encontrar en [6].

A continuación se muestran algunos de los kernels de uso más común en SVMs.

Kernel polinomial sencillo	$k(x,z) = \langle x,z\rangle^m$
Kernel polinomial	$k(x,z) = (\langle x,z\rangle + c)^m$
Kernel Gaussiano (RBF)	$k(x,z) = e^{\frac{-\|x-z\|^2}{2\sigma^2}}$
Kernel Sigmoidal	$k(x,z) = tanh(\kappa\langle x,z\rangle + c)$
Kernel inverso multicuadrático	$k(x,z) = \frac{1}{\sqrt{\|x-z\|^2+c^2}}$

Donde m, c, σ y κ son parámetros libres. En el siguiente capítulo se analizarán los primeros 3 kernels de la tabla anterior.

En los capítulos siguientes, cuando hablemos de la ϕ dada por algún kernel, nos referiremos a alguna función ϕ que puede tomar el lugar de la ϕ que aparece en la definición de kernel (Definición 1).

Capítulo 3

Espacio de rasgos para kernels

En este capítulo se analizarán algunas características y propiedades de algunos de los kernels más comunmente usados en las SVMs a través de las funciones ϕ y los espacios de rasgos que definen implícitamente. La primera sección pretende orientar un poco hacia el enfoque que se usará en el resto del capítulo y al mismo tiempo mostrar algunos resultados generales acerca del espacio de rasgos.

3.1. Sobre los mapeos ϕs y sus espacios de rasgos

Como vimos en el capítulo anterior, los kernels definen implícitamente una función ϕ que mapea a un espacio de rasgos F. Sin embargo, para que el algoritmo de las SVMs pueda generar una región de decisión que clasifique correctamente, es necesario que el problema sea linealmente separable en F. En este capítulo veremos algunas características del espacio de rasgos de algunos kernels para poder analizar qué tipo de problemas pueden ser resueltos por cada kernel.

Hay que mencionar que la función ϕ y el espacio de rasgos F no son únicos para cada kernel. Por ejemplo, si definimos las funciones ϕ_1 y ϕ_2 en $\mathbb{R}^2$ como sigue

$$\phi_1(x) = \phi_1(x_1,x_2) = \begin{pmatrix} x_1^2 \\ x_2^2 \\ x_1x_2 \end{pmatrix}$$

$$\phi_2(x) = \phi_2(x_1, x_2) = \begin{pmatrix} x_1^2 \\ x_2^2 \\ \frac{3}{5}x_1x_2 \\ \frac{4}{5}x_1x_2 \end{pmatrix}$$

aunque $\phi_1 \neq \phi_2$ y sus respectivos espacios F_1 y F_2 son de diferente dimensión (ϕ_1 mapea a $\mathbb{R}^3$ mientras que ϕ_2 mapea a $\mathbb{R}^4$), ambas definen el mismo kernel:

$$\begin{aligned} k_2(x,z) &= \langle \phi_2(x), \phi_2(z) \rangle = x_1^2 z_1^2 + x_2^2 z_2^2 + \frac{9}{25}x_1x_2z_1z_2 + \frac{16}{25}x_1x_2z_1z_2 \\ &= x_1^2 z_1^2 + x_2^2 z_2^2 + x_1x_2z_1z_2 = \langle \phi_1(x), \phi_1(z) \rangle = k_1(x,z) \end{aligned}$$

Sin embargo, dado un kernel k, la dimensión de la imagen de un conjunto J finito bajo cualquier ϕ tal que $\langle \phi(x), \phi(z) \rangle = k(x,z)$ es fija. En el caso del ejemplo anterior esto significa que $dim(\phi_1(J)) = dim(\phi_2(J))$ para J un conjunto finito de vectores.

Con el fin de demostrar esta aseveración introduciremos los siguientes lemas, y definiciones.

Definición 2. *Sean x_1, x_2, ..., x_n vectores en un espacio de Hilbert H. Llamaremos la matriz de productos punto de x_1, x_2, ..., x_n a la matriz K tal que $K_{ij} = \langle x_i, x_j \rangle$.*

Lema 1. *La matriz K es de rango completo si y sólo si x_1, x_2, ..., x_n son linealmente independientes.*

Demostración. Supongamos que $x_1, x_2, \cdots, x_n$ no son linealmente independientes. Sin perder generalidad supongamos que $x_1 = a_1x_2 + a_2x_3 + \cdots + a_{n-1}x_n$, donde

$a_j \neq 0$ para al menos un j. Entonces

$$\begin{aligned} K_1 &= \begin{pmatrix} \langle x_1, x_1 \rangle \\ \langle x_1, x_2 \rangle \\ \vdots \\ \langle x_1, x_n \rangle \end{pmatrix} = \begin{pmatrix} \langle a_1x_2 + a_2x_3 + \cdots + a_{n-1}x_n \ , \ x_1 \rangle \\ \langle a_1x_2 + a_2x_3 + \cdots + a_{n-1}x_n \ , \ x_2 \rangle \\ \vdots \\ \langle a_1x_2 + a_2x_3 + \cdots + a_{n-1}x_n \ , \ x_n \rangle \end{pmatrix} \\ &= \begin{pmatrix} a_1\langle x_2, x_1 \rangle + a_2\langle x_3, x_1 \rangle + \cdots + a_{n-1}\langle x_n, x_1 \rangle \\ a_1\langle x_2, x_2 \rangle + a_2\langle x_3, x_2 \rangle + \cdots + a_{n-1}\langle x_n, x_2 \rangle \\ \vdots \\ a_1\langle x_2, x_n \rangle + a_2\langle x_3, x_n \rangle + \cdots + a_{n-1}\langle x_n, x_n \rangle \end{pmatrix} \\ &= a_1 \begin{pmatrix} \langle x_2, x_1 \rangle \\ \langle x_2, x_2 \rangle \\ \vdots \\ \langle x_2, x_n \rangle \end{pmatrix} + a_2 \begin{pmatrix} \langle x_3, x_1 \rangle \\ \langle x_3, x_2 \rangle \\ \vdots \\ \langle x_3, x_n \rangle \end{pmatrix} + \cdots + a_{n-1} \begin{pmatrix} \langle x_n, x_1 \rangle \\ \langle x_n, x_2 \rangle \\ \vdots \\ \langle x_n, x_n \rangle \end{pmatrix} \\ &= a_1K_2 + a_2K_3 + \cdots + a_{n-1}K_n \end{aligned} \tag{3.1}$$

De la ecuación 3.1, la primer columna de K se puede poner como combinación lineal de las demás por lo que K no es de rango completo. De aquí se deduce que si K es de rango completo $\Longrightarrow$ x_1, x_2, ..., x_n son linealmente independientes.

Ahora supongamos que K no es de rango completo. Sin perder generalidad supongamos que $K_1 = a_1K_2 + a_2K_3 + \cdots + a_{n-1}K_n$ donde $a_j \neq 0$ para al menos un j.

Sea $P = a_1x_2 + a_2x_3 + \cdots + a_{n-1}x_n$. Entonces, de la ecuación 3.1 tenemos que

$$\langle x_1, x_i \rangle = \langle P, x_i \rangle \qquad \forall i \in \{1, 2, \cdots, n\} \tag{3.2}$$

Observemos que la norma de P es igual a la norma de x_1 ya que

$$\begin{aligned}
\|x_1\|^2 &= \langle x_1, x_1\rangle = \langle a_1x_2 + a_2x_3 + \cdots + a_{n-1}x_n\ ,\ x_1\rangle \\
&= a_1\langle x_2, x_1\rangle + a_2\langle x_3, x_1\rangle + \cdots + a_{n-1}\langle x_n, x_1\rangle \\
&= a_1\langle x_2\ ,\ a_1x_2 + a_2x_3 + \cdots + a_{n-1}x_n\rangle \\
&\quad + a_2\langle x_3\ ,\ a_1x_2 + a_2x_3 + \cdots + a_{n-1}x_n\rangle \\
&\quad \cdots + a_{n-1}\langle x_n\ ,\ a_1x_2 + a_2x_3 + \cdots + a_{n-1}x_n\rangle \\
&= \langle a_1x_2 + a_2x_3 + \cdots + a_{n-1}x_n\ ,\ a_1x_2 + a_2x_3 + \cdots + a_{n-1}x_n\rangle \\
&= \|a_1x_2 + a_2x_3 + \cdots + a_{n-1}x_n\|^2 = \|P\|^2 \qquad (3.3)
\end{aligned}$$

Y además la proyección ortogonal de P sobre x_1 es x_1:

$$\frac{P^Tx_1}{x_1^Tx_1}x_1 = \frac{\langle P, x_1\rangle}{\langle x_1x_1\rangle}x_1 = \frac{\langle x_1, x_1\rangle}{\langle x_1, x_1\rangle}x_1 = x_1 \qquad (3.4)$$

Por lo tanto, $P = x_1$, es decir, x_1 se puede poner como combinación lineal de x_2, x_3 ..., x_n. De aquí se deduce que si x_1, x_2, ..., x_n son linealmente independientes, las columnas de K también son linealmente independientes, esto es, K es de rango completo.

□

Antes de enunciar el siguiente lema introduciremos la siguiente definición.

Definición 3. *Sea M una matriz cuadrada $m \times m$ y $s \in \{1, 2, \ldots, m\}$. Definimos M'_s como la submatriz de M que resulta de sustraer la hilera s y la columna s.*

Lema 2. *Si $K_{n\times n}$ tiene rango $L < m$ y x_s se puede expresar como combinación lineal de los otros $n-1$ vectores x_2, x_3 ..., x_n entonces K'_s tiene rango L.*

Demostración. Supongamos que el $rango(K'_s) < L$, entonces K'_s tiene menos columnas linealmente independientes que K.

Sea K_u, $u \neq s$, una columna linealmente independiente de K (correspondiente al vector x_u) tal que su equivalente en K'_s no es linealmente independiente (su equivalente será la columna en K'_s correspondiente a x_u) a la que llamaremos $(K'_s)_u$.

Entonces, ya que $(K'_s)_u$ no es linealmente independiente de las demás columnas de K'_s podemos expresarla como $(K'_s)_u = \sum_{i\neq u, i\neq s} a_i(K'_s)_i$ (donde $(K'_s)_i$ es la columna de

K'_s correspondiente a x_i) con $a_j \neq 0$ para algún j, esto es, para todo $k \in \{1, 2, \ldots, n\}$ con $i \neq s$ se tiene que

$$\langle x_u, x_k\rangle = \sum_{i\neq u, i\neq s} a_i\langle x_i, x_k\rangle = \langle \sum_{i\neq u, i\neq s} a_ix_i, x_k\rangle \qquad (3.5)$$

Ahora definimos una nueva numeración para x_i y a_i, de tal forma que

$$x'_1 = x_u,\ x'_2 = x_{u+1},\ \ldots,\ x'_{s-u-1} = x_{s-1},\ x'_{s-u} = x_{s+1},$$
$$x'_{s-u+1} = x_{s+2},\ \ldots, x'_{n-1} = x_{u-1}$$

(esto si $u < s$), en el caso en que $u > s$ sería:

$$x'_1 = x_u,\ x'_2 = x_{u+1},\ \ldots,\ x'_{n-u+s-1} = x_{s-1},$$
$$x'_{n-u+s} = x_{s+1},\ x'_{n-u+s+1} = x_{s+2},\ \ldots, x'_{n-1} = x_{u-1}$$

y de forma análoga para a'_i.

Entonces, haciendo $P = \sum_{i=1}^{n-1} a'_ix'_i$, observamos que la ecuación (3.5) es equivalente a la ecuación (3.2) (tomando n como $n-1$). Entonces, de las ecuaciones (3.3) y (3.4) deducimos que $x'_1 = P = \sum_{i=1}^{n-1} a'_ix'_i$, que en terminos de los x_i significa que $x_u = \sum_{i\neq u, i\neq s} a_ix_i$. Luego, se tiene que:

$$\begin{aligned}
K_u &= \begin{pmatrix} \langle x_u, x_1\rangle \\ \langle x_u, x_2\rangle \\ \vdots \\ \langle x_u, x_n\rangle \end{pmatrix} = \begin{pmatrix} \langle \sum_{i\neq u, i\neq s} a_ix_i\ ,\ x_1\rangle \\ \langle \sum_{i\neq u, i\neq s} a_ix_i\ ,\ x_2\rangle \\ \vdots \\ \langle \sum_{i\neq u, i\neq s} a_ix_i\ ,\ x_n\rangle \end{pmatrix} = \sum_{i\neq u, i\neq s} a_i \begin{pmatrix} \langle x_i, x_1\rangle \\ \langle x_i, x_2\rangle \\ \vdots \\ \langle x_i, x_n\rangle \end{pmatrix} \\
&= \sum_{i\neq u, i\neq s} a_iK_i \qquad (3.6)
\end{aligned}$$

Hemos expresado K_u como combinación lineal de las demás columnas de K, lo cual contradice la hipótesis inicial. Luego, $rango(K'_s) \geq L$.

Sin embargo, al quitarle un renglón o columna a una matriz, el rango de la nueva matriz siempre es menor o igual al rango de la matriz original, de aquí que $rango(K'_s) \leq L$.

Por lo tanto $rango(K'_s) = L$.

□

Lema 3. *La matriz $K_{n\times n}$ es de rango L si y sólo si exactamente L de los vectores $x_1, x_2, \ldots, x_n$ son linealmente independientes.*

Demostración. Si $L = n$, el lema queda demostrado por el Lema 1.

Si $L < n$ entonces, por el Teorema 1, existe $s \in \{1, 2, \ldots, n\}$, tal que x_s puede ser expresado como combinación lineal de los demás vectores $x_1, x_2, \ldots, x_n$. Entonces, por el lema 2 ,K'_s tiene rango L.

Luego, si K'_s es de rango completo ($L = n - 1$), por el lema 1 los L vectores $x_1, x_2, \ldots, x_{s-1}, x_{s+1}, \ldots, x_n$ son linealmente independientes.

Si K'_s no es de rango completo, repetimos el proceso eliminando vectores que se puedan poner como combinación lineal de los demás (que el Lema 1 nos asegura que habrá) hasta obtener una matriz $L \times L$ de rango completo que, por construcción y por el lema 1 nos dará L vectores linealmente independientes.

Observamos que los $n-L$ vectores que fueron desechados de entre los n originales por el proceso podían expresarse como combinación lineal de los vectores que iban quedando. Entonces, se tiene que estos $n - L$ vectores podían ser expresados como combinación de los otros L, lo cual demuestra el lema.

□

Y como corolario obtenemos el resultado que buscabamos:

Corolario 1. *Sea k un kernel en $X \times X$, y $J = \{x_1, x_2, \ldots, x_n\}$ un conjunto de vectores en el espacio de entrada X. Entonces si $\phi_1 : X \to F_1$ y $\phi_2 : X \to F_2$ son mapeos tales que $\langle \phi_1(x), \phi_1(z) \rangle = k(x, z) = \langle \phi_2(x), \phi_2(z) \rangle$, se tiene que $dim(\phi_1(J)) = dim(\phi_2(J))$.*

Demostración. Formemos la matriz K de tal forma que $K_{ij} = k(x_i, x_j) = \langle \phi_1(x), \phi_1(z) \rangle = \langle \phi_2(x), \phi_2(z) \rangle$. Luego, por el lema 3, el rango de K es igual al número de vectores linealmente independientes del conjunto $\phi_1(J) = \{\phi_1(x_1), \phi_1(x_2), \ldots, \phi_1(x_n)\}$, y también al número de vectores linealmente independientes del conjunto $\phi_2(J) = \{\phi_2(x_1), \phi_2(x_2), \ldots, \phi_2(x_n)\}$. De aquí se sigue que $dim(\phi_1(J)) = dim(\phi_2(J))$.

□

En las siguientes secciones de este capítulo analizaremos algunos kernels a través de algunas de sus funciones ϕ asociadas. Como comentamos anteriormente, las ϕ's asociadas a los kernels no son únicas, sin embargo, el corolario 1 nos ayuda a entender porqué no es tan importante la ϕ que escojamos para analizar. Las ϕs que analizaremos no serán las ϕs cuyo espacio de rasgos F tenga dimensión más pequeña sino las que nos faciliten más las operaciones.

Buena parte del estudio de los siguientes kernels comienza en $X = \mathbb{R}^2$, pero en la mayoría de los casos se puede extender fácilmente para $X = \mathbb{R}^n$.

3.2. Kernel polinomial sencillo

El kernel polinomial sencillo está dado por

$$K(x, z) = (\langle x, z \rangle)^m \tag{3.7}$$

donde m es un parámetro libre el cual lo tomaremos en $\mathbb{N}$.

Es fácil demostrar que K es un kernel usando el hecho de que el producto punto usual es un kernel (trivial tomando ϕ como la identidad y $X = F$) y la tercera propiedad de la Proposición 1 $m - 1$ veces.

Tomando $X = \mathbb{R}^2$ y desarrollando (3.7) tenemos

$$K(x, z) = x_1^m z_1^m + m x_1^{m-1} z_1^{m-1} x_2 z_2 + \cdots + \binom{m}{j} x_1^{m-j} z_1^{m-j} x_2^j z_2^j + \cdots + m x_1 z_1 x_2^{m-1} z_2^{m-1} + x_2^m z_2^m \tag{3.8}$$

De aquí podemos observar que una ϕ para el kernel polinomial sencillo es

$$\phi(x) = \begin{pmatrix} x_1^m \\ \sqrt{m} x_1^{m-1} x_2 \\ \vdots \\ \sqrt{\binom{m}{j}} x_1^{m-j} x_2^j \\ \vdots \\ \sqrt{m} x_1 x_2^{m-1} \\ x_2^m \end{pmatrix} \tag{3.9}$$

La cual mapea a $\mathbb{R}^{m+1}$.

Ahora, nos interesa saber qué tipo de problemas se vuelven linealmente separables en el espacio de rasgos F de esta ϕ. Para ello, observaremos qué tipo de curvas en X son mandadas a planos en F.

Un plano en F tiene la forma

$$\langle w, z\rangle + b' = 0 \tag{3.10}$$

para $w \in F$ y $b' \in \mathbb{R}$. Luego, los puntos $x \in \mathbb{R}^2$ que son mapeados por ϕ en un plano deben cumplir

$$\langle w, \phi(x)\rangle = b \tag{3.11}$$

para algún $w \in F$, $w \neq 0$ y $b \in \mathbb{R}$. Sustituyendo (3.9) en la ecuación (3.11), tenemos que los puntos que son mapeados por ϕ a un plano deben satisfacer

$$w_1 x_1^m + \sqrt{m} w_2 x_1^{m-1} x_2 + \cdots + \sqrt{\binom{m}{j}} w_{j+1} x_1^{m-j} x_2^j + \cdots + w_{m+1} x_2^m = b \tag{3.12}$$

donde $b \in \mathbb{R}$ y $w_i \in \mathbb{R}$ con $w_s \neq 0$ para algún s. Ya que las variables w_i pueden tomar cualquier valor en $\mathbb{R}$, podemos expresar la ecuación 3.12 de la siguiente manera

$$a_1 x_1^m + a_2 x_1^{m-1} x_2 + \cdots + a_{j+1} x_1^{m-j} x_2^j + \cdots + a_m x_1 x_2^{m-1} + a_{m+1} x_2^m = b \tag{3.13}$$

donde $b \in \mathbb{R}$ y $a_i \in \mathbb{R}$ con $a_s \neq 0$ para algún s.

Entonces, la imagen inversa de un plano en F bajo ϕ tendrá la forma de la ecuación (3.13).

De lo anterior y de la expresión (2.19) del capítulo anterior, deducimos que el método de SVMs con el kernel polinomial sencillo separa vectores con funciones de separación f tales que $f(x) = 0$ se puede expresar como la expresión dada por (3.13). Esto es porque el método de SVMs buscará un hiperplano de separación óptimo en el espacio de rasgos F y, por lo tanto, su imágen inversa bajo ϕ tendrá necesariamente la forma dada en la expresión (3.13).

Ahora, cuando tomamos $X = \mathbb{R}^n$, el desarrollo en (3.8) se vuelve

$$K(x, z) =$$

$$\sum_{j_1=0}^{m} \sum_{j_2=0}^{m-j_1} \cdots \sum_{j_{m-1}=0}^{m-j_1-\cdots-j_{n-2}} \frac{m!(x_1 z_1)^{j_1} \cdots (x_{n-1} z_{n-1})^{j_{n-1}} (x_n z_n)^{m-j_1-\cdots-j_{n-1}}}{j_1! \cdots j_{n-1}! (m - j_1 \cdots - j_{n-1})!}$$

Y las componentes de la ϕ se vuelven los términos de la función polinomial

$$P(x) = \sum_{j_1=0}^{m} \sum_{j_2=0}^{m-j_1} \cdots \sum_{j_{n-1}=0}^{m-j_1-\cdots-j_{n-2}} \frac{m!(x_1)^{j_1} \cdots (x_{n-1})^{j_{n-1}} (x_n)^{m-j_1-\cdots-j_{n-1}}}{j_1! \cdots j_{n-1}! (m - j_1 \cdots - j_{n-1})!}$$

Y, sustituyendo ϕ en (3.11), y de manera análoga al caso para $X = \mathbb{R}^2$ tenemos que los puntos mapeados por ϕ a un plano deben cumplir

$$\sum_{j_1=0}^{m} \sum_{j_2=0}^{m-j_1} \cdots \sum_{j_{n-1}=0}^{m-j_1-\cdots-j_{n-2}} a_{j_1, j_2, \ldots, j_{n-1}} (x_1)^{j_1} \cdots (x_{n-1})^{j_{n-1}} (x_n)^{m-j_1-\cdots-j_{n-1}} = b \tag{3.14}$$

donde $b \in \mathbb{R}$ y $a_{i_1, i_2, \ldots, i_{m-1}} \in \mathbb{R}$ con $a_{s_1, s_2, \ldots, s_{m-1}} \neq 0$ para algunos $s_1, s_2, \ldots, s_{m-1}$. Observamos que la expresión del lado derecho de (3.14) es la expresión polinomial general en n variables de grado m, por lo que las funciones de separación f encontradas por el método de SVMs con este kernel serán expresiones polinomiales de grado m

Entonces, la función de separación generada por el kernel polinomial sencillo tiene la forma $f(x) = b$, donde f es una función polinomial que contiene sólo términos de grado m.

De las expresiones anteriores, se tiene que al multiplicar este kernel por una constante positiva c ($K'(x, z) = cK(x, z)$, el cual es kernel por la segunda propiedad de la proposición 1), con $X = \mathbb{R}^2$ la función ϕ se volverá

$$\phi(x) = \sqrt{c} \begin{pmatrix} x_1^m \\ \sqrt{m} x_1^{m-1} x_2 \\ \vdots \\ \sqrt{\binom{m}{j}} x_1^{m-j} x_2^j \\ \vdots \\ \sqrt{m} x_1 x_2^{m-1} \\ x_2^m \end{pmatrix} \tag{3.15}$$

Y la expresión (3.13) se transformaría en

$$\sqrt{c}a_1x_1^m + \sqrt{c}a_2x_1^{m-1}x_2 + \cdots + \sqrt{c}a_{j+1}x_1^{m-j}x_2^j + \cdots + \sqrt{c}a_mx_1x_2^{m-1} + \sqrt{c}a_{m+1}x_2^m$$

$$= a'_1x_1^m + a'_2x_1^{m-1}x_2 + \cdots + a'_{j+1}x_1^{m-j}x_2^j + \cdots + a'_mx_1x_2^{m-1} + a'_{m+1}x_2^m = b \quad (3.16)$$

Luego, teóricamente el kernel $cK(x,z)$ generará el mismo tipo de funciones de separación y tomará la misma función de decisión. Incluso si se tienen kernels cuya ϕ sea similar a la mostrada en (3.9) pero con una o más componentes multiplicadas por constantes generarán la misma función de separación. Sin embargo, en la práctica esto sólo es cierto siempre que alguna de las componentes de ϕ no se acerque demasiado a cero (o se vuelve demasiado pequeña comparada con otras), en este caso puede suceder que el algoritmo desprecie dicha componente en ϕ y por tanto desaparezca su correspondiente término en (3.13). Esto ocasiona que la función de decisión encontrada pueda ser diferente.

Esto se sigue cumpliendo para $X = \mathbb{R}^n$ y la demostración es análoga.

A continuación veremos como ejemplo el caso del kernel polinomial sencillo cuando $m = 2$, esto es, $K(x,z) = (\langle x,z\rangle)^2$.

La ϕ para este caso es

$$\phi(x) = \begin{pmatrix} x_1^2 \\ \sqrt{2}x_1x_2 \\ x_2^2 \end{pmatrix} \quad (3.17)$$

Y la expresión (3.13) nos queda

$$a_1x_1^2 + a_2x_1x_2 + a_3x_2^2 = b \quad (3.18)$$

Por lo que el kernel generará elipses e hipérbolas como funciones de separación. En la figura 3.1 se muestra un ejemplo de una función de separación encontrada con este kernel.

La imagen en F bajo ϕ de las funciones de separación encontradas por este kernel se encuentra contenida dentro de un plano, el cual es el plano de separación en F. La figura 3.2 muestra este hecho.

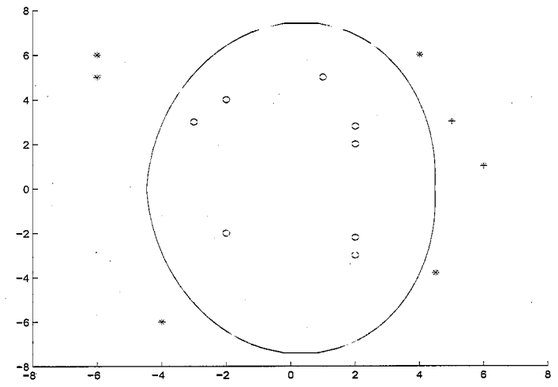

Figura 3.1: Región de separación encontrada por el método de SVMs con el kernel $k(x,z) = (\langle x,z\rangle)^2$

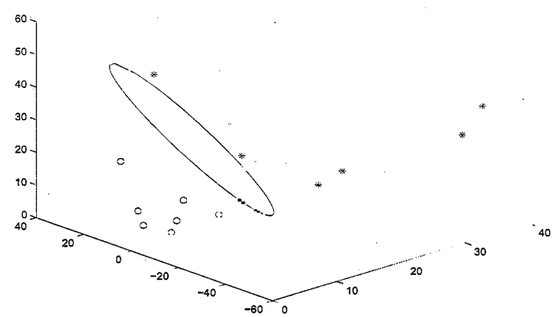

Figura 3.2: Imagen bajo ϕ de la función de separación en el espacio de rasgos. Nótese que el círculo está contenido en un plano, el cual es el plano de separación en F

3.3. Kernel polinomial

El kernel polinomial está dado por

$$K(x,z) = (\langle x,z\rangle + c)^m \tag{3.19}$$

donde m y c son parámetros libres con $m \in \mathbb{N}$ y $c > 0$.

Desarrollando la expresión (3.19) tenemos:

$$K(x,z) = (\langle x,z\rangle)^m + mc(\langle x,z\rangle)^{m-1} + \cdots + \binom{m}{j} c^j (\langle x,z\rangle)^{m-j} + \cdots mc^{m-1}\langle x,z\rangle + c^m \tag{3.20}$$

esto es, el kernel se puede expresar como una suma de kernels de la forma $aK'(x,z)$, donde K' es un kernel polinomial sencillo y a una constante positiva. De aquí es fácil ver que K es un kernel usando el hecho de que el kernel polinomial sencillo es un kernel y la primera y segunda propiedades de la proposición 1.

Para obtener una ϕ para este kernel, observamos que si tenemos un kernel k como suma de kernels k_1, k_2 cuyas respectivas ϕs son ϕ_1, ϕ_2, entonces

$$k(x,z) = k_1(x,z) + k_2(x,z) = \langle \phi_1(x), \phi_1(z)\rangle + \langle \phi_2(x), \phi_2(z)\rangle$$

Por lo que una posible ϕ para k sería una concatenación de ϕ_1 y ϕ_2:

$$\phi(x) = \begin{pmatrix} \phi_1(x) \\ \phi_2(x) \end{pmatrix} \tag{3.21}$$

De tal forma que:

$$\langle \phi(x), \phi(z)\rangle = \langle \phi_1(x), \phi_1(z)\rangle + \langle \phi_2(x), \phi_2(z)\rangle = k(x,z)$$

Luego, la ϕ del kernel polinomial es una concatenación de las ϕs de los kernels polinomiales sencillos que aparecen en la suma de la expresión (3.20). Entonces, podemos poner esta ϕ como

$$\phi(x) = \begin{pmatrix} \phi_m(x) \\ \sqrt{mc}\phi_{m-1}(x) \\ \vdots \\ \sqrt{\binom{m}{j}c^j}\phi_{m-j}(x) \\ \vdots \\ \sqrt{mc^{m-1}}\phi_1(x) \\ \sqrt{c^m} \end{pmatrix} \tag{3.22}$$

donde ϕ_i es la función ϕ del kernel polinomial sencillo $k(x,z) = (\langle x,z\rangle)^i$ la cual, para $X = \mathbb{R}^2$ está dada por la expresión (3.9) para $m = i$ y para $X = \mathbb{R}^n$, aunque no se da explícitamente, se da una idea de la forma de esta ϕ_i.

Las raíces cuadradas en la expresión (3.22) vienen de obtener la ϕ para el kernel $ck(x,z)$ cuando k es un kernel polinomial sencillo y $c > 0$ como se explicó en la sección anterior. La última componente de ϕ en (3.22), $\sqrt{c^m}$, es la ϕ para el kernel $k(x,z) = c^m$.

Para $X = \mathbb{R}^2$, la ϕ dada por (3.22) mapea de $\mathbb{R}^2$ a $F = \mathbb{R}^{\frac{(m+1)(m+2)}{2}}$. Notemos que la dimensión de la imágen de ϕ en (3.22) es de, a lo más $\frac{(m+1)(m+2)}{2} - 1$, ya que la última componente de ϕ es siempre constante.

Ahora, los puntos $x \in \mathbb{R}^2$ que son mapeados por ϕ dentro de un plano deben cumplir (3.11) para algún $w \in F$, $w \neq 0$ y $b \in \mathbb{R}$. Para hacer nuestra tarea más fácil, expresaremos w como

$$w = \begin{pmatrix} w'_{m+1} \\ w'_m \\ \vdots \\ w'_2 \\ w'_1 \end{pmatrix} \tag{3.23}$$

donde w'_{m+1} es el vector formado por las primeras $m+1$ componentes de w, w'_m es el vector formado por las siguientes m y así sucesivamente. Entonces, sustituyendo, (3.22) en (3.11), tenemos que los puntos mapeados por ϕ a un plano dado por (3.11) deben cumplir

$$\langle w'_{m+1}, \phi_m(x)\rangle + \sqrt{mc}\langle w'_m, \phi_{m-1}(x)\rangle + \cdots + \sqrt{\binom{m}{j}c^j}\langle w'_{m-j+1}, \phi_{m-j}(x)\rangle + \cdots$$

$$\cdots + \sqrt{mc^{m-1}}\langle w'_2, \phi_1(x)\rangle + w'_1\sqrt{c^m} = b' \tag{3.24}$$

Al sustituir las ϕ_i de la misma forma que en (3.12):

$$(w'_{m+1})_1 x_1^m + \cdots + \sqrt{\binom{m}{j}}(w'_{m+1})_{j+1} x_1^{m-j}x_2^j + \cdots + (w'_{m+1})_{m+1}x_2^m$$

$$+ \quad \sqrt{mc}[(w'_m)_1 x_1^{m-1} + \cdots + \sqrt{\binom{m-1}{j}}(w'_m)_{j+1}x_1^{m-j-1}x_2^j + \cdots + (w'_m)_m x_2^{m-1}]$$

$$+ \qquad\qquad \vdots$$

$$\cdots + \sqrt{mc^{m-1}}[(w_2')_1 x_1 + (w_2')_2 x_2] + w_1'\sqrt{c^m} = b' \tag{3.25}$$

Ya que las w_i (las componentes de w) son arbitrarias, podemos expresar (3.25) de la siguiente manera (análoga a lo que hicimos en (3.13)):

$$\begin{aligned} & a_{m,1}x_1^m + a_{m,2}x_1^{m-1}x_2 + \cdots + a_{m,j+1}x_1^{m-j}x_2^j + \cdots + a_{m,m+1}x_2^m \\ + \; & a_{m-1,1}x_1^{m-1} + a_{m-1,2}x_1^{m-2}x_2 + \cdots + a_{m-1,j+1}x_1^{m-j-1}x_2^j + \cdots + a_{m-1,m}x_2^{m-1} \\ + \; & \vdots \\ \cdots + \; & a_{1,1}x_1 + a_{1,2}x_2 = b \end{aligned} \tag{3.26}$$

Luego, los puntos mapeados por ϕ dentro de un plano deben cumplir (3.26) para $a_i \in \mathbb{R}$, $a_s \neq 0$ para algún s.

Análogamente al kenel polinomial sencillo, de lo anterior y de la expresión (2.19) del capítulo anterior, deducimos que el método de SVMs con el kernel polinomial separa vectores con funciones de separación f tales que $f(x) = 0$ se puede expresar como la expresión dada por (3.26).

Luego, la función de separación generada por este kernel tiene la forma $f(x) = b$, donde f es una función polinomial de grado, a lo más m.

Cuando $X = \mathbb{R}^n$, las ϕ_i en (3.22) son las correspondientes ϕ_i del kernel polinomial sencillo para $X = \mathbb{R}^n$, y la expresión (3.26) se vuelve la suma de las expresiones del lado derecho de (3.14) correspondientes a las diferentes ϕ_i. Esto es, las funciones de separación f encontradas por el método de SVMs con este kernel es nuevamente una expresión polinomial de grado m.

Notemos que no importa el valor del parámetro c en (3.19), el kernel generará el mismo tipo de funciones de separación. Sin embargo, en la práctica, al igual que para el kernel polinomial sencillo, esto deja de cumplirse cuando una o más componentes de ϕ en (3.22) se acercan demasiado a cero o se vuelven demasiado pequeñas en comparación con las demás. Es muy común que el parámetro c se fije como $c = 1$ el cual no tiene mucho riesgo de influir negativamente.

Obviamente el kernel polinomial es más completo que el kernel polinomial sencillo ya que teóricamente, el espacio de hipótesis del segundo está contenido en el espacio de hipótesis del primero ya que el primero puede generar las mismas funciones de decisión que el segundo y aún más.

Ahora, supongamos que queremos encontrar una función de separación polinomial con el método de SVMs. Si buscaramos que la función fuera una curva polinomial de grado u, entonces necesitaríamos usar un kernel polinomial con $m \geq u$. Esto se deduce del siguiente lema.

Lema 4. *Sea $\phi : X \to F$ una función, con $X \subseteq \mathbb{R}^n$ y $F = \mathbb{R}^s$. Luego, si las componentes de $\phi(x)$: $\phi_1(x), \phi_2(x), \ldots, \phi_s(x)$ generan a las expresiones polinomiales de grado u en X, entonces todas las curvas polinomiales de grado u en X, $P(x) = P(x_1, ..., x_n) = b$ donde $P(x)$ es una expresión polinomial de grado u en X y $b \in \mathbb{R}$, son llevadas a un hiperplano a través de ϕ.*

Demostración. Sea $P(x)$ una expresión polinomial de grado u en X. Tenemos que las $\phi_i(x)$ generan a las expresiones polinomiales de grado u en X si y sólo si existen $a_1, a_2, \ldots, a_s \in \mathbb{R}$ tales que

$$a_1\phi_1(x) + a_2\phi_2(x) + \cdots + a_s\phi_s(x) = P(x) \tag{3.27}$$

Definimos

$$a = \begin{pmatrix} a_1 \\ a_2 \\ \vdots \\ a_s \end{pmatrix} \tag{3.28}$$

Entonces, de (3.27) y (3.28), para todo $x \in X$ tal que $P(x) = b$, se tiene que

$$\langle a, \phi(x) \rangle = a_1\phi_1(x) + a_2\phi_2(x) + \cdots + a_s\phi_s(x) = P(x) = b \tag{3.29}$$

Por lo que la imagen de la curva $P(x) = b$ bajo ϕ queda dentro del hiperplano $\langle a, z \rangle = b$.

□

Ahora, como mencionamos anteriormente, las componentes $\phi_i(x)$ en (3.22) son los términos de la expresión polinomial general de grado m, por lo que generan a cualquier expresión polinomial de grado menor o igual a m. Aplicando el lema 4 y ya que todas las imágenes de las posibles curvas de separación para las SVMs con este kernel deben estar contenidas dentro de un hiperplano, deducimos que para obtener una curva de separación polinomial de grado u se necesita usar un kernel polinomial con $m \geq u$.

3.4. Kernel Gausiano

El kernel gaussiano está dado por

$$k(x,z) = e^{\frac{-\|x-z\|^2}{2\sigma^2}} \tag{3.30}$$

donde σ es un parámetro libre.

Podemos expresar (3.30) como

$$k(x,z) = e^{\frac{-\|x-z\|^2}{2\sigma^2}} = (e^{\frac{-\|x\|^2}{2\sigma^2}} e^{\frac{-\|z\|^2}{2\sigma^2}}) e^{\frac{\langle x,z\rangle}{\sigma^2}} \tag{3.31}$$

Analizaremos (3.31) como un producto de kernels

$$k(x,z) = e^{\frac{-\|x-z\|^2}{2\sigma^2}} = k_1'(x,z)k_2'(x,z) \tag{3.32}$$

donde

$$k_1'(x,z) = e^{\frac{-\|x\|^2}{2\sigma^2}} e^{\frac{-\|z\|^2}{2\sigma^2}} \tag{3.33}$$

$$k_2'(x,z) = e^{\frac{\langle x,z\rangle}{\sigma^2}} \tag{3.34}$$

De aquí que el kernel gaussiano es un kernel si k_1' y k_2' son kernels. Para verificarlo mostraremos las ϕs de k_1' y k_2'.

k_1' es un kernel que mapea a $\mathbb{R}$ y cuya ϕ está dada por

$$\phi_1'(x) = e^{\frac{-\|x\|^2}{2\sigma^2}} \tag{3.35}$$

Mientras que k_2' puede ser expresado de la siguiente forma por el desarrollo de Taylor de la función e^x

$$k_2'(x,z) = e^{\frac{\langle x,z\rangle}{\sigma^2}} = 1 + \frac{\langle x,z\rangle}{\sigma^2} + \frac{(\frac{\langle x,z\rangle}{\sigma^2})^2}{2!} + \cdots + \frac{(\frac{\langle x,z\rangle}{\sigma^2})^{i-1}}{(i-1)!} + \cdots \tag{3.36}$$

Observamos que los términos en (3.36) son kernels polinomiales sencillos multiplicados por una constante. Luego, la ϕ del k_2' mapea a un espacio de dimensión infinita (y por lo tanto también el kernel gaussiano). Si truncamos la serie hasta el término

j-ésimo, esta ϕ la podemos aproximar como

$$\phi_2'(x) = \begin{pmatrix} 1 \\ \frac{1}{\sigma^2}\phi_1(x) \\ \frac{1}{\sigma^4 2!}\phi_2(x) \\ \vdots \\ \frac{1}{\sigma^{2(j-1)}(j-1)!}\phi_j(x) \end{pmatrix} \tag{3.37}$$

Donde ϕ_i es la ϕ del kernel polinomial sencillo dada por (3.9) con $m = i$. Observamos que las componentes de (3.37) son multiplos de las componentes mostradas en (3.22), por lo que este kernel aproxima la función e separación que un kernel polinomial con m grande.

Para obtener una ϕ para el kernel gaussiano, observamos que

$$k_1'(x,z)k_2'(x,z) = (\phi_1'(x)\phi_2'(z))(\langle \phi_2'(x), \phi_2'(z)\rangle) = \langle \phi_1'(x)\phi_2'(x), \phi_1'(z)\phi_2'(z)\rangle \tag{3.38}$$

ya que $\phi_1'(x) \in \mathbb{R}$ para todo $x \in X$. Por lo tanto, la ϕ para el kernel gaussiano se aproxima por

$$\phi(x) = e^{\frac{-\|x\|^2}{2\sigma^2}} \begin{pmatrix} 1 \\ \frac{1}{\sigma^2}\phi_1(x) \\ \frac{1}{\sigma^4 2!}\phi_2(x) \\ \vdots \\ \frac{1}{\sigma^{2(j-1)}(j-1)!}\phi_j(x) \end{pmatrix} = \begin{pmatrix} e^{\frac{-\|x\|^2}{2\sigma^2}} \\ e^{\frac{-\|x\|^2}{2\sigma^2}}\frac{1}{\sigma^2}\phi_1(x) \\ e^{\frac{-\|x\|^2}{2\sigma^2}}\frac{1}{\sigma^4 2!}\phi_2(x) \\ \vdots \\ e^{\frac{-\|x\|^2}{2\sigma^2}}\frac{1}{\sigma^{2(j-1)}(j-1)!}\phi_j(x) \end{pmatrix} \tag{3.39}$$

Luego, los puntos que son mapeados a un plano en F a través de esta ϕ deben cumplir (3.11) para algún $w \in F$ y $b \in \mathbb{R}$, esto es

$$\langle w, \phi(x)\rangle = \langle w, \phi_1'(x)\phi_2'(x)\rangle = \phi_1'(x)\langle w, \phi_2'(x)\rangle = b \tag{3.40}$$

Entonces, como las componentes de ϕ_2' son las mismas que en (3.22) con m grande, generan las expresiones polinomiales de cualquier grado (ya que j es arbitrario en (3.39) y en (3.37)) y entonces (3.40) nos queda

$$e^{\frac{-\|x\|^2}{2\sigma^2}} P(x_1, x_2, \ldots, x_n) = b \tag{3.41}$$

donde P es una expresión polinomial en n variables (de cualquier grado) y $x_1, x_2, \ldots, x_n$ son las componentes de x. Luego, la función de separación encontrada por el método de SVMs con el kernel gaussiano tendrá la forma dada en (3.41).

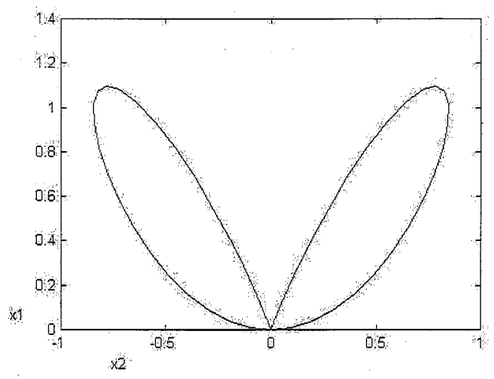

Figura 3.3: Gráfica de la función $e^{\frac{-\|x\|^2}{10}}(x_1 - x_2^2) = 0$. Nótese que cuando la norma de x es mayor a 1, la curva se va a cero exponencialmente.

Las curvas dadas en (3.41) son generalmente curvas cerradas, ya que cuando la norma de x tiende a infinito, la parte exponencial de (3.41) tiende a cero, dominando la expresión. La figura 3.3 muestra este hecho con una de estas curvas, tomando $P(x) = x_1 - x_2^2$ y $b = 0$.

Observemos también que cuando la norma de x es muy cercana a cero, la parte exponencial de (3.41) tiende a uno, por lo que la curva tiende a parecerse a $P(x) = 0$. Por ejemplo, en la figura 3.3 se puede observar que la curva, antes de comenzar a cerrarse (y a tender a cero), se asemeja a la parábola $x_1 - x_2^2 = 0$, especialmente cerca de cero. La figura 3.4 muestra otro ejemplo de curvas de la forma $e^{\frac{-\|x\|^2}{2\sigma^2}} P(x) = b$.

El parámetro σ es un factor importante para la elección de la función de separación que se encontrará usando el kernel gaussiano y ahora veremos algunas formas en las que interfiere en el funcionamiento del kernel.

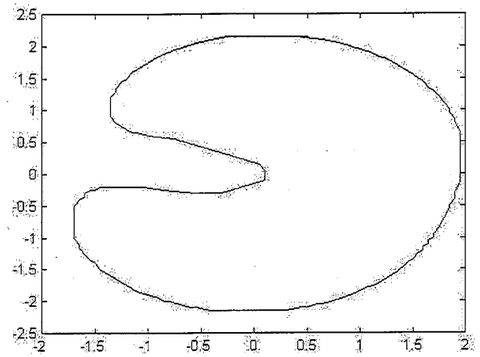

Figura 3.4: Gráfica de la función $e^{\frac{-\|x\|^2}{1}}(x_2 + 3x_1^2 + x_2x_1 + x_2^2) = \frac{1}{8}$. Este es un ejemplo más general del tipo de curvas de separación que encuentra el método de SVMs con el kernel gaussiano.

Veamos que cuando σ es muy grande, los coeficientes de los kernels polinomiales sencillos en (3.36) tienden a cero rápidamente (más rápidamente conforme el kernel es de mayor grado). Esto se traduce en que las componentes de ϕ_2' en (3.37) que estén dadas por kernels polinomiales sencillos de grado muy grande serán tomadas como sero y por tanto serán despreciadas por la computadora. Esto significa que, en este caso, P en (3.41) será una expresión de grado generalmente pequeño y por tanto el espacio de hipótesis generado por éste kernel será más pobre.

En el caso límite, cuando $\sigma \to \infty$, podemos aproximar $k_2'(x, z)$ dado en (3.36) por los primeros dos términos y $k_1'(x, z)$ dado en (3.34) lo podemos aproximar por 1. En este caso, el kernel gaussiano dado por (3.32) se aproxima a

$$K(x, z) = 1 + c\langle x, z\rangle \tag{3.42}$$

donde c está dada en el segundo término de (3.36). Luego, K encontrará funciones de separación lineales, esto se puede observar en que la ϕ de este kernel es una concatenación de las ϕs de los kernels $k_1(x, z) = 1$ y $k_2(x, z) = c\langle x, z\rangle$. La ϕ del primero es $\phi(x) = 1$ y del segundo es $\sqrt{c}\phi_1(x)$, donde $\phi_1(x)$ es la ϕ para el kernel polinomial sencillo de grado 1 (lineal). Luego, la ϕ para la aproximación del kernel

gaussiano queda dada por

$$\phi(x) = \begin{pmatrix} \sqrt{c}\phi_1(x) \\ 1 \end{pmatrix} \tag{3.43}$$

observamos que (3.43) tiene la misma forma que (3.22) (tomando $m = 1$ y $c = 1$), sólo que con algunas componentes multiplicadas por una constante (las componentes de ϕ_1). Sin embargo, como mencionamos en el caso del kernel polinomial, la función de separación encontrada con este kernel no cambia si las componentes se multiplican por constantes, por lo que la función de separación encontrada con el kernel gaussiano se aproxima a la encontrada por el kernel polinomial de grado 1.

En resumen, el kernel gaussiano se aproxima a un kernel lineal cuando $\sigma \to \infty$.

3.5. Kernels a partir de ϕs

En las secciones anteriores presentamos algunos de los kernels mas usados y vimos qué tipo de funciones de separación son capaces de generar. Sin embargo, es posible generar kernels en el caso inverso, es decir, construir un kernel que encuentre un tipo de función de separación específico. En esta sección veremos un ejemplo de cómo se puede hacer esto.

Supongamos que trabajamos con $X = \mathbb{R}^3$ como espacio de entrada y buscamos separar los vectores de entrada con una esfera, es decir, queremos un kernel que encuentre una función de separación esférica. Para la construcción de este kernel tomaremos esferas centradas en el origen como funciones de separación (se pueden trasladar los vectores de entrada más cerca del origen antes de usar este kernel con el fin de hacer más eficiente la separación). Entonces, la función de separación debe tener la forma

$$x^2 + y^2 + z^2 = r^2 \tag{3.44}$$

con $r \in \mathbb{R}$. Para hacer la tarea más fácil (además de aumentar el espacio de hpótesis del kernel que construiremos) admitiremos también elipsoides e hiperboloiddes como funciones de separación, entonces la función de separación tendrá la forma

$$ax^2 + by^2 + cz^2 = r^2 \tag{3.45}$$

Para $a, b, c \in \mathbb{R}$. Ahora, (3.45) podemos verla como producto punto

$$\left\langle \begin{pmatrix} a \\ b \\ c \end{pmatrix}, \begin{pmatrix} x^2 \\ y^2 \\ z^2 \end{pmatrix} \right\rangle = r^2 \tag{3.46}$$

Esto sugiere la siguiente función ϕ

$$\phi(x) = \begin{pmatrix} x_1^2 \\ x_2^2 \\ x_3^2 \end{pmatrix} \tag{3.47}$$

puesto que, los puntos que son mandados a hiperplanos (en este caso planos ya que estamos tomando $F = \mathbb{R}^3$) por ϕ deben cumplir (3.11) que, al sustituir (3.47), equivale a (3.46) (con $b = r^2$). Esto quiere decir que la función de separación encontrada por el kernel que define la ϕ dada en (3.47) tendrá la forma

$$w_1x_1^2 + w_2x_2^2 + w_3x_3^2 = b \tag{3.48}$$

la cual es la misma que (3.45), que era lo que buscabamos (notemos que b puede tomar valores negativos, pero en este caso siempre podemos multiplicar por -1 la ecuación (3.48)).

Luego, el kernel que buscamos es

$$K(x, z) = \langle \phi(x), \phi(z) \rangle = x_1^2z_1^2 + x_2^2z_2^2 + x_3^2z_3^2 \tag{3.49}$$

Este kernel no tiene una forma tan elegante como los otros que analizamos en este capítulo, principalmente porque su uso está restringido a los problemas en que el espacio de entrada es $X = \mathbb{R}^3$.

De la misma manera con que se fabricó este kernel, se pueden generar kernels "sugiriendo" algún tipo de función como función de separación generada por el kernel. Sólo se necesita expresar este tipo de funciones explícitamente como se sugirió para (3.45) y después expresarlo como un producto punto como se hizo en (3.46) para poder deducir una ϕ que nos de el kernel.

En la figura 3.5 se muestra un ejemplo de una función de separación encontrada con el kernel dado en (3.49), la cual es una elipsoide. En este ejemplo, los vectores marcados con círculos son los vectores $(5, 5, 5)$, $(4, 3.8, 6)$, $(3, 5, 4)$, $(3, 3, 3)$, $(5, 2, 3)$, $(0, 5, 2)$, $(3, 1, 4)$ y los marcados con asteriscos son los vectores $(1, 2, 1)$, $(2, 1, 1.3)$, $(1, 1.2, 2)$, $(1.1, 0.5, 1)$, $(2, 1, 1)$, $(0, 4, 0)$, $(1, 3, 2)$.

Ya que para este kernel $F = \mathbb{R}^3$, podemos graficar también el plano de separación. La figura 3.6 muestra el plano de separación en F encontrado para los mapeos bajo

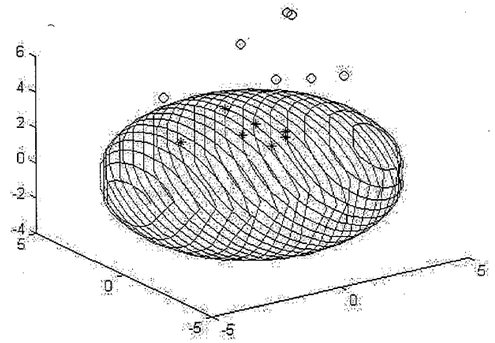

Figura 3.5: Ejemplo de una función de separación encontrada con el kernel $K(x,z) = x_1^2z_1^2 + x_2^2z_2^2 + x_3^2z_3^2$.

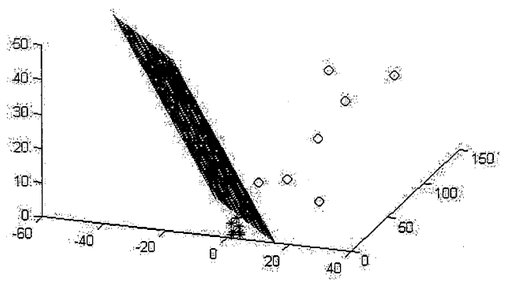

Figura 3.6: Plano de separación en F encontrado con el kernel $K(x,z) = x_1^2z_1^2 + x_2^2z_2^2 + x_3^2z_3^2$. La imagen de la función de separación de la figura 3.5 se encuentra contenida en el plano.

ϕ de los mismos vectores de entrada de la figura 3.5 .

Este kernel es un ejemplo de que no es necesario que el espacio de rasgos F definido por la ϕ de este kernel sea de dimensión mucho más grande el espacio de entrada X para que el kernel pueda encontrar una función de decisión que separe correctamente los vectores de entrada, pues en este kernel la dimensión de X y la de X son iguales (ϕ va de $\mathbb{R}^3$ a $\mathbb{R}^3$).

En general, es muy común que un kernel eficiente defina un espacio de rasgos F de dimensión muy grande, (por ejemplo, la ϕ del kernel polinomial dada por (3.22) muestra que la dimensión del espacio F al que mapea aumenta considerablemante cuando m es grande) e incluso infinita, aunque sea teóricamente, pues en la práctica la computadora puede cortar la dimensión (como en el caso del kernel gaussiano cuando σ es muy pequeña). Sin embargo, está no es una condición que garantiza que el kernel encontrará una función de decisión que clasifique los vectores correctamente y además es posible encontrar kernels (como el presentado en esta sección) que puedan clasificar los puntos que se le solicitan sin generar una dimensión muy grande.

Capítulo 4

Algunos experimentos con otros kernels

En este capítulo se presenta la construcción de 3 nuevos kernels a partir de algunas ideas sugeridas por Vapnik [1]. Además se presentan los resultados de algunos experimentos realizados con estos kernels, se calula el error de generalización y se compara con una de las cotas de éste error presentada en el capítulo 2.

4.1. Usando polinomios ortogonales para obtener kernels

Dado una serie de polinomios ortogonales $P_1(x)$, $P_2(x)$, ..., $P_{m+1}(x)$ con $x \in \mathbb{R}$, Vapnik [1] establece que la función K_m definida como sigue es un kernel

$$K_m(x,y) = \sum_{j=1}^{m} P_j(x)P_j(z) = A_m \frac{P_{m+1}(x)P_m(z) - P_m(x)P_{m+1}(z)}{x-z} \quad \text{para} \quad x \neq z \tag{4.1}$$

$$K_m(x,x) = \sum_{j=1}^{m} P_j^2(x) = A_m[P'_{m+1}(x)P_m(x) - P'_m(x)P_{m+1}(x)] \tag{4.2}$$

donde $P'_i(x)$ es la derivada del polinomio $P_i(x)$ y A_m depende de m y es tal que

$$P_{n+1}(x) = (A_n x + B_n)P_n(x) - C_n P_n(x) \quad \forall n \in \mathbb{N} \tag{4.3}$$

Así, A_n, al igual que B_n y C_n, es una variable característica de los polinomios ortogonales $P_i(x)$ y define una fórmula recurrente para los polinomios ortogonales. A_n,

B_n y C_n existen para cualquier conjunto de polinomios ortogonales.

La expresión dada por (4.1) se conoce como la fórmula de Christoffel-Darboux (la expresión (4.2) es el caso límite).

Podemos observar que las sumatorias en (4.1) y en (4.2) en efecto forman un kernel cuya ϕ está dada como

$$\phi(x) = \begin{pmatrix} P_1(x) \\ P_2(x) \\ \vdots \\ P_m(x) \end{pmatrix} \tag{4.4}$$

Observemos sin embargo, que este kernel se limita solamente a el caso en que el espacio de entrada está en $\mathbb{R}$ (a este tipo de kernels se les llama *kernels unidimensionales* o *one-dimensional kernels*), por lo que en la mayoría de los problemas este kernel no nos servirá pues usualmente se presentan como mínimo con $X \subseteq \mathbb{R}^2$.

Afortunadamente, Vapnik establece también una forma para fabricar kernels definidos para espacios de entrada en $\mathbb{R}^n$ (llamados *kernels multidimensionales*) a partir de kernels definidos para espacios de entrada en $\mathbb{R}$. Este método está dado por el siguiente lema.

Lema 5. *Si las funciones $k_1(x,z)$, $k_2(x,z)$, ..., $k_n(x,z)$ definidas para $x, z \in \mathbb{R}$ son kernels, entonces, para $u, v \in \mathbb{R}^n$, la función $k(u,v)$ definida por*

$$k(u,v) = \prod_{i=1}^{n} k_i(u_i, v_i)$$

es un kernel.

La demostración de este lema se puede encontrar en [1].

Entonces, esto nos sugiere que, dado un conjunto de polinomios ortogonales $P_1(x)$, ..., $P_{m+1}(x)$, tomemos el kernel

$$K(x,z) = \prod_{i=1}^{n} k_m(x_i, z_i) \tag{4.5}$$

con K_m dado por (4.1) y por (4.2), siendo ésta la forma más sencilla de generar un kernel con polinomios ortogonales.

4.2. Kernel con polinomios de Tchevyshev de primer orden

Ahora mostraremos un ejemplo de un kernel construido con el procedimiento que se mostró en la sección anterior. Para ello, usaremos los llamados polinomios de Tchevyshev de primer orden, definidos en $[-1,1]$. La mayor ventaja que se logra con estos polinomios es que se tiene una forma de expresar el término n-ésimo de forma explícita:

$$T_n(x) = 2^{-n+1} \cos[n \cos^{-1} x] \tag{4.6}$$

Por lo que podemos usar directamente las fórmulas de la parte derecha de (4.1) y de (4.2) con los polinomios m y $m+1$ sin tener que generar los demás. Además, para estos polinomios $A_m = \frac{1}{2}$ para todo m.

Entonces, si sustituímos (4.6) en las fórmulas (4.1) y (4.2), obtenemos el siguiente kernel unidimensional para $x, z \in [-1,1]$

$$\begin{aligned} K_m(x,z) &= A_m \frac{T_{m+1}(x)T_m(z) - T_m(x)T_{m+1}(z)}{x - z} \\ &= 2^{-2m} \frac{\cos[(m+1)\cos^{-1} x]\cos[m\cos^{-1} z] - \cos[(m+1)\cos^{-1} z]\cos[m\cos^{-1} x]}{x - z} \end{aligned}$$

Luego, usando la expresión (4.5), podemos construir el siguiente kernel multidimensional

$$K(x,z) = \prod_{i=1}^{n} 2^{-2m} \frac{\cos[(m+1)\cos^{-1} x_i]\cos[m\cos^{-1} z_i] - \cos[(m+1)\cos^{-1} z_i]\cos[m\cos^{-1} x_i]}{x_i - z_i} \tag{4.7}$$

A continuación mostramos un experimento realizado con este kernel para probar su error de generalización. Generamos 120 puntos contenidos en $[-1,1] \times [-1,1]$ y a partir de una curva predeterminada (para estos experimentos usamos la curva $50(x-0.9)(x+0.3)(x-0.5)(x+0.8)x)$) clasificamos dichos puntos, los puntos encima de la curva les asignamos la clase $+1$ y los demás la clase -1.

Ahora, de estos 120 puntos, tomaremos los primeros 90 como conjunto de entrenamiento S y usaremos el método de SVMs con el kernel K para obtener una función de decisión que clasifique correctamente todos los vectores de entrada. Luego clasificaremos con esta región de decisión los 30 puntos restantes y compararemos la clase asignada por la región de decisión con la clase asignada previamente a estos

Cuadro 4.1: Experimento para kernel de polinomios de Tchevyshev con primer orden

m	Num. de v. de s.	Error	Cota del error
3	14	10.00 %	61.65 %
4	15	6.67 %	88.94 %
5	17	10.00 %	93.63 %
6	25	13.33 %	******
7	32	13.33 %	******

puntos (que llamaremos puntos de prueba). El número de puntos de prueba clasificados incorrectamente por la región de decisión entre el número total de puntos de prueba será el error de generalización de K para este conjunto de puntos.

Los resultados de este experimento son mostrados en el cuadro 4.2, donde la primer columna se refiere al valor de m para el kernel K en (4.7), la segunda columna es el número de vectores de soporte generados por el algoritmo, la tercer columna muestra los respectivos errores de generalización y la última columna muestra la cota del error de generalización obtenida con la expresión (2.22) del capítulo 2 con una confianza de 90 %. Como se puede apreciar, la cota suele ser demasiado grande cuando se tienen muchos vectores de soporte. Los asteriscos en la última columna indican que la cota no aporta información para esos casos.

La lista comienza con $m = 3$ ya que para $m = 1$ y $m = 2$ la región de decisión encontrada no clasifica correctamente todos los puntos de entrada.

Se puede observar la conocida tendencia del error de generalización de primero disminuir hasta un mínimo y luego aumentar conforme aumenta la complejidad del kernel (en este caso el valor de m, es decir el número de polinomios que se tomarán). En este caso, el kernel que nos da el menor error de generalización se logra con $m = 4$.

La figura 3.1 nos muestra una de las regiones de decisión encontradas con este kernel a lo largo de los experimentos. En la figura se pueden alcanzar a apreciar algunos de los datos de entrada del experimento anterior (los asteriscos son puntos de la clase $+1$ y los círculos son de la clase -1)

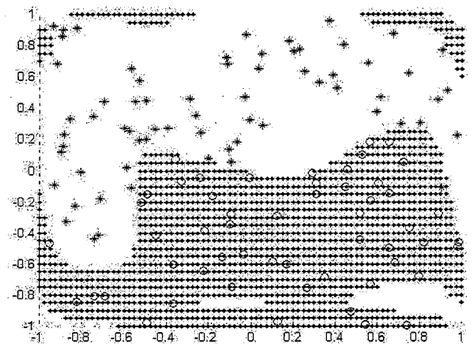

Figura 4.1: Ejemplo de una región de decisión encontrada por el kernel fabricado con los polinomios de Tchevyshev de primer orden. La región clara es la clase $+1$ y la región obscura es la clase -1.

4.3. Kernel con polinomios de Hermite

Aunque es posible usar el método descrito en la primera sección de este capítulo para construir un kernel usando los polinomios de Hermite, el kernel sería difícil de expresar y de usar ya que no hay forma de expresar el m-ésimo polinomio de Hermite sin generar los $m - 1$ anteriores (o al menos no de una forma sencilla como con los polinomios de Tchevyshev).

Sin embargo Vapnik [1] propone el siguiente kernel unidimensional construido con los polinomios de Hermite

$$\begin{aligned} K_q(x,z) &= \sum_{i=0}^{\infty} q^i H_i(x) H_i(z) \\ &= \frac{1}{\sqrt{\pi(1-q^2)}} e^{[\frac{2xzq}{1+q} - \frac{(x-z)^2 q^2}{1-q^2}]} \end{aligned} \tag{4.8}$$

donde $0 < q < 1$ y la segunda igualdad es una función generadora (generator function). Ejemplos de este tipo de funciones (incluyendo la función dada en (4.8)) y

cómo obtenerlas se pueden encontrar en [7] y en [8].

Este kernel tiene una ϕ de dimensión infinita que podemos expresar como

$$\phi_q(x) = \begin{pmatrix} H_0(x) \\ q^{\frac{1}{2}} H_1(x) \\ \vdots \\ q^{\frac{i}{2}} H_j(x) \\ \vdots \end{pmatrix} \tag{4.9}$$

Como en los casos anteriores, las componentes que se acerquen a cero de esta ϕ muy probablemente serán despreciadas por la computadora. Es por ello que el parámetro q es importante, pues define que tan cerca estará de cero la i-ésima componente de ϕ y por tanto si será o no tomada en cuenta.

Entonces, si q es pequeña se tomarán menos componentes de ϕ, y el espacio de hipótesis será más pequeño y si q es grande, se tomarán más componentes de ϕ y el espacio de hipótesis será más grande. En otras palabras, q denota la complejidad de la función de decisión encontrada por el kernel.

Entonces, truncando esta ϕ hasta la componente j, y sustituyendo en la expresión (3.11), nos queda que con este kernel encontramos funciones de separación de la forma

$$a_1 H_1(x) + a_2 H_2(x) + \cdots + a_{j+1} H_j(x) = b \tag{4.10}$$

Ahora, para obtener un kernel multidimensional usamos el lema 5 para obtener el siguiente kernel multidimensional (con $k_i = K_q$ para una q fija y para todo i)

$$\begin{aligned} K(x,z) &= \prod_{i=1}^{n} \frac{1}{\sqrt{\pi(1-q^2)}} e^{[\frac{2x_i z_i q}{1+q} - \frac{(x_i - z_i)^2 q^2}{1-q^2}]} \\ &= \frac{1}{(1-q^2)^{\frac{n}{2}}} e^{[\frac{2\langle x,z\rangle q}{1+q} - \frac{\|x-z\|^2 q^2}{1-q^2}]} \end{aligned} \tag{4.11}$$

el cual es el kernel que buscamos.

A continuación repetimos el experimento que se realizó para el kernel construido con los polinomios de Tchevyshev de primer orden pero ahora para el kernel de los

Cuadro 4.2: Experimento para kernel con polinomios de Hermite

q	Num. de v. de s.	Error	Cota del error
0.1	13	6.67 %	84.23 %
0.2	14	10.00 %	88.94 %
0.3	14	10.00 %	88.94 %
0.4	13	6.67 %	84.23 %
0.5	13	6.67 %	84.23 %
0.6	11	3.33 %	74.73 %
0.7	11	3.33 %	74.73 %
0.8	11	0.00 %	74.73 %
0.9	17	6.67 %	******

polinomios de Hermite. Para este experimento obtenemos el cuadro 4.3 (análogo a el anterior).

En este caso, tomaríamos $q = 0.8$ para el menor error de generalización tanto por el encontrado experimentalmente como el sugerido por la cota que se calculó.

4.4. Kernel con series de Fourier

Aquí presentaremos otro kernel unidimensional construido esta vez con series de Fourier. En este caso, nuestro objetivo es mapear con una ϕ, de la siguiente forma

$$\phi_N(x) = \begin{pmatrix} \frac{1}{\sqrt{2}} \\ \sin x \\ \vdots \\ \sin Nx \\ \cos x \\ \vdots \\ \cos Nx \end{pmatrix} \tag{4.12}$$

para algún $N \in \mathbb{N}$.

Cuadro 4.3: Experimento para kernel con series de Fourier

N	Num. de v. de s.	Error	Cota del error
2	14	6.67 %	88.94 %
3	14	6.67 %	88.94 %
4	14	6.67 %	88.94 %
5	25	0.00 %	******
6	20	0.00 %	******
7	25	0.00 %	******
8	32	3.33 %	******
9	35	6.67 %	******
10	38	10.00 %	******

Entonces, el kernel (unidimensional) que buscamos tiene la siguiente forma

$$\begin{aligned} K_N(x,z) &= \frac{1}{2} + \sum_{i=1}^{N} (\sin ix \sin iz + \cos ix \sin iz) \\ &= \frac{1}{2} + \sum_{i=1}^{N} \cos i(x-z) \\ &= \frac{\sin \frac{(2N+1)}{2}(x-z)}{\sin \frac{(x-z)}{2}} \end{aligned} \tag{4.13}$$

donde la ultima igualdad es llamada la fórmula de Dirichlet.

Luego, el kernel multidimensional que buscamos tendrá la forma

$$K(x,z) = \prod_{i=1}^{n} K_N(x_i, z_i) \tag{4.14}$$

para $N \in \mathbb{N}$.

Una vez mas repetimos para este kernel el experimento que se realizó para los kernels anteriores y obtenemos el cuadro 4.4.

En este caso, los kernels que nos dan el menor error de generalización experimental son los que tienen $N = 5$, $N = 6$ y $N = 7$. Sin embargo, las cotas nos indican

que sería mejor tomar el kernel con $N = 2$, $N = 3$ o $N = 4$.

La figura 3.2 nos muestra una de las regiones de decisión encontradas con este kernel (una vez mas los asteriscos son puntos de la clase $+1$ y los círculos son de la clase -1)

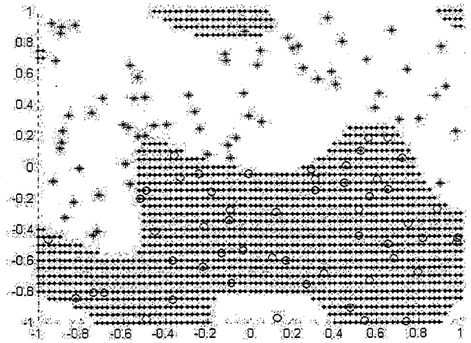

Figura 4.2: Ejemplo de una región de decisión encontrada por el kernel fabricado con la serie de Fourier. La región clara es la clase $+1$ y la región obscura es la clase -1.

Capítulo 5

Conclusiones y trabajo futuro

En esta tesis se mostró un análisis que se espera ayude a entender mejor la forma en que operan los kernels gaussiano, polinomial y polinomial sencillo en las SVMs para el problema de clasificiación.

Este análisis se puede usar como una forma de encontrar un kernel adecuado (de entre los que se analizaron) para algún problema de clasificación a travéz de información que el conjunto de entrenamiento pueda sugerir acerca de la función de separación que se busca. Por ejemplo, si se necesita que la función de separación tenga varias "curvas" (como cambios de concavidad) digamos s, y se quiere usar un kernel polinomial para el problema, lo más adecuado sería tomar valores de m en (3.19) alrededor de s para que la función de separación fuera una expresión polinomial con grado alrededor de n. Análogamente, en el caso del kernel Gaussiano, se sugeriría tomar σ suficientemente pequeña para que la componente $n+1$ de la expresión (3.39) sea suficientemente grande para que sea tomado en cuenta para encontrar la función de separación.

Además, si se sabe algo de información a priori acerca de la función de separación, se puede construir un kernel que obligue a tomar cierto tipo de funciones de separación.

Por otra parte, se presentan también otros 3 kernels, en el capítulo 3, construídos a partir técnicas diversas presentadas por Vapnik y que, aunque no presentan una forma tan elegante como los kernels mas usados, esperamos puedan presentar mayor variedad para encontrar mejores funciones de decisión o menor error de generalización para ciertos problemas.

Por ultimo, esperamos que este cambio de perspectiva de la forma convencional del estudio de kernels sirva para animar u orientar a buscar mejores maneras de identificar el mejor kernel para un problema dado.

Bibliografía

[1] Vladimir N. Vapnik, *Statistical learning theory*, 1998, John Wiley & Sons Inc. New York, Chichester, Weinheim, Brisbane, Singapore, Toronto. AT&T Labs-Research London University.

[2] Vladimir N. Vapnik, *The nature of statistical learning theory* Springer-Verlag New York Inc., 1995, AT&T Bell Laboratories 101 Crawfords Corner Road Holmdel.

[3] Christopher J. C. Burges, *A tutorial on Support Vector Machines for Pattern Recognition*, Microsoft Research (formerly Lucent Technologies), Data Mining and Knowledge Discovery 2, 121-167, 1998.

[4] Vojislav Kecman, *Learning and soft computing: Support vector machines, neural networks and fuzzy logic models*, A Bradford Book the MIT (Massachusetts Institute of Technology) Press Cambridge, Massachussets, London England, 2001.

[5] Ricardo del Angel Pérez Flores*Entrenamiento de Máquinas de Soporte Vectorial y su aplicación al reconocimiento de patrones*, Tesis de maestría, CIMAT Julio de 2002.

[6] Nello Cristianini and Jhon Shawe-Taylor, *An introduction to Support Vector Machines and other kernel-based methods*, Cambridge University Press 2000.

[7] Harry Hochstadt, *The functions of mathematical physics*, Dover Publications Inc. New York 1971, 1986, Library of Congress Cataloging-in-Publication Data.

[8] Elma B. McBride, *Obtaining generating functions*, Springer-Verlag New York, Springer Tracts in natural Philosophy, 21, 1971.

[9] Gabor Szego, *Orthogonal polynomials*, 1939, American Mathematical Society Colloquium Publications Volume XXIII

[10] Gilbert G. Walter, *Wavelets and other orthogonal systems with applications*, CRC Press Inc., 1994, Library of Congress Cataloging-in-Publication data.