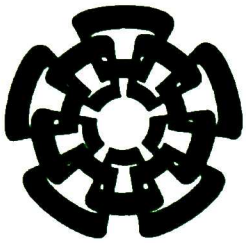
xx(113537.1)



CINVESTAV

Centro de Investigación y de Estudios Avanzados del I.P.N.
Unidad Guadalajara

Diseño de Kernels para Máquinas de Multivectores de Soporte usando Álgebra Geométrica.

Tesis que presenta:
Nancy Guadalupe Arana Daniel

para obtener el grado de:
Maestro en Ciencias

en la especialidad de:
Ingeniería Eléctrica

Directores de Tesis
Dr. Eduardo José Bayro Corrochano

Guadalajara, Jal., Noviembre del 2003.

Diseño de Kernels para Máquinas de Multivectores de Soporte usando Álgebra Geométrica

Tesis de Maestría en Ciencias
Ingeniería Eléctrica

Por:
Nancy Guadalupe Arana Daniel
Licenciada en Informática
Universidad de Guadalajara 1992-1996

Becario del CONACyT, expediente no. **165089**

Directores de Tesis
Dr. Eduardo José Bayro Corrochano

CINVESTAV del IPN Unidad Guadalajara, Noviembre del 2003.

Índice general

Índice de figuras

Índice de cuadros

Agradecimientos

Agradezco a Dios por permitirme finalizar mis estudios. A mi esposo, por ser el gran apoyo con el que siempre cuento, por brindarme el ejemplo de inteligencia y superación que siempre quise emular, por todo el amor, gracias. A mis hijas por ser mi gran inspiración en la vida. A mi madre, por que todo lo bueno que pueda existir en mi, fue su enseñanza, por ser una madre para mis hijas y mi más grande ejemplo de bondad y sabiduría. A mi padre por todo su apoyo y testimonio de fuerza de voluntad. A mis hermanos y hermanas que siempre están a mi lado cuando les necesito. A mis suegros por el apoyo incondicional y por hacerme sentir como su hija. A mis amigos y compañeros de estudio, por las múltiples asesorías y las largas, pero muy amenas horas de trabajo conjunto. A mi asesor, Prof. Dr. Eduardo Bayro, por la oportunidad de trabajar en un proyecto de mi interés. Un agradecimiento al Conacyt.

Notación.

N	Dimensión del espacio de características (*feature space*)
$y \in Y$	Vector de salida esperada y espacio de salida
$x \in X$	Vector de entrada y espacio de entrada
F	Espacio de características (*feature space*)
$< x.z >$	Producto punto entre x y z
$\Phi : X \rightarrow F$	Mapeo del espacio X hacia el *feature space* F
$K(x, z)$	Kernel $< \Phi(z).\Phi(x) >$
$f(x)$	Función real
n	Dimensión del espacio de entrada
R	Radio de la esfera contenedora de datos
w	Vector de pesos
b	Umbral o *bias*
α	Variables duales o multiplicadores de Lagrange
L	Lagrangiano primal
W	Lagrangiano dual
$\|\|.\|\|_p$	p-norma
$x^{'}, X^{'}$	Vector o matriz transpuesta
$\aleph, \Re$	Números reales o naturales
S	Conjunto de vectores de entrenamiento
ℓ	Tamaño del conjunto de entrenamiento
γ	Margen
ξ	Variables flojas
h	Dimensión VC
e	Base del espacio vectorial
e, e_0	Punto al infinito y origen del sistema

Capítulo 1

INTRODUCCIÓN

La construcción de máquinas capaces de aprender a partir de la experiencia ha sido uno de los principales objetivos de la inteligencia artificial. En los últimos años los intentos para alcanzar dicho objetivo han demostrado que los sistemas computacionales pueden lograr un nivel de aprendizaje muy significativo, al desarrollar sistemas y algoritmos que tratan de imitar el proceso de adquisición de conocimientos humano. Para ello se hace uso de lo que se conoce como *metodología del aprendizaje,* cuya tarea es lograr que, dados pares de entrada y salida que deseamos que *el sistema aprendiz* relacione, éste adquiera experiencia de ellos y por tanto, al presentarle entradas nunca antes vistas por él, de acuerdo a su conocimiento adquirido, relacione estos nuevos datos con una salida determinada. El objetivo es que el sistema *aprenda* de dichas duplas de datos de entrada/salida deseada, *clasificándolas* según el conocimiento adquirido.

La estadística tradicional y los sistemas conocidos como redes neuronales han desarrollado muchos métodos para discriminar entre dos clases de instancias (clasificar datos binarios) usando combinaciones lineales de los datos de entrada, esto es, funciones lineales, con el objeto de explotar el poder de generalización que dichas funciones brindan a los sistemas[1] Pero éstas máquinas lineales de aprendizaje tienen un limitado poder computacional,

[1] Conocidos como máquinas lineales por el uso que hacen de funciones de éste tipo para la tarea de clasificación.

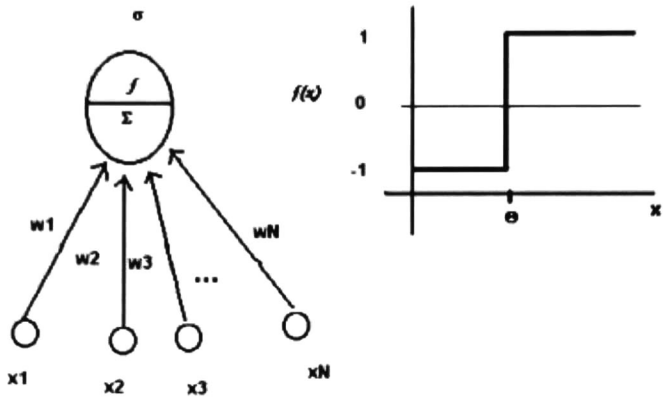


Figura 1.1: Modelo de perceptrón de Rosenblat

lo que fue resaltado por Minsky y Papert por los años sesentas (ver fig. 1.1).

En general, aplicaciones complejas del mundo real requieren más poder expresivo en los espacios de hipótesis que funciones lineales. Otra forma de ver este problema es que frecuentemente el concepto de función objetivo[2] no puede ser expresado como una simple combinación lineal de los atributos dados, ya que en general se requiere que características (*features*) más abstractas de los datos sean explotadas. El perceptrón de capas múltiples (MLP por sus siglas en inglés) fue propuesto como una solución a este problema, y esta proposición permitió el desarrollo de algoritmos de aprendizaje tales como la propagación hacia atrás (*back-propagation*) para entrenar estos sistemas. Una solución mucho más poderosa fue presentada mediante el uso de *la teoría de optimización y los kernels*, que son funciones que nos permiten llevar los datos de entrada a a un tipo especial de espacios vectoriales conocidos como espacios de Hilbert, éstos tienen una dimensión mucho más elevada que la dimensión del espacio de entrada, para de este modo, aumentar la ca-

[2]Función que al ser evaluada con un dato de entrada resulta en una cantidad que nos permite clasificar dicho dato. Función desconocida que queremos que la máquina aprenda para ser capaz de clasificar nuevos datos.

pacidad computacional de las máquinas lineales explotando características *(features)* más abstractos de los datos. Esta solución se lleva a cabo en los algoritmos de los sistemas computacionales conocidos como *Support Vector Machines -SVM-* (Máquinas con Vectores de Soporte)[3], las cuales se dicen más poderosas que las MLP's, entre otras cosas por que los resultados que nos otorgan en la tarea de clasificación son más aproximados a los óptimos y por que al mantener la linearidad del sistema los problemas se mantienen en la categoría de *tratables computacionalmente.*

De esta manera, con la motivación de contribuir al avance de los sistemas inteligentes por medio del desarrollo del algoritmo de las Support Vector Machines y con ayuda de la rama de las matemáticas conocida como Álgebra Geométrica, intentamos con este trabajo de tesis elaborar nuevos kernels que nos otorgaran resultados de clasificación con características de generalización diferentes a los ya existentes. Por lo que, apoyados en los fundamentos teóricos de dichas álgebras, explotamos la posibilidad que nos brindan de obtener dimensionalidades altas de los espacios de características (*feature spaces*), dados ciertos espacios vectoriales de entrada, por medio de la operación denominada *producto Clifford,* y del uso de mapeos de los datos de entrada a multivectores. Es esta la razón por la cual se le otorgó el nombre de la derivación de las SVM's a *Support Multivector Machines SMVM,* que obtuvimos en este trabajo.

Otro de los objetivos concernientes a la realización de esta tesis es la de demostrar que el poder geométrico descriptivo de objetos del mundo real que nos ofrecen las Álgebras Geométricas, podía aumentar de alguna forma las capacidades de aprendizaje de las SVM's. En este trabajo de investigación, explotamos dicho poder descriptivo para tratar de realizar una importante compresión de datos al representar las entradas a nuestro sistema como entidades geométricas (esferas), las cuales engloban una gran cantidad de puntos, para evitar presentar uno a uno todo el conjunto de datos de entrenamiento a la máquina de aprendizaje.

[3]En el presente documento se hará uso del término en inglés *Support Vector Machines (SVM)* para referirnos a dichas máquinas dado el uso estandarizado del término en el argot computacional.

Este documento se divide en cinco capítulos y tres apéndices. En el Capítulo 2 se brindan las bases teóricas de las Álgebras Geométricas, las cuales son el fundamento matemático en el que nos apoyamos para la realización de la derivación de las SVM hacia las SMVM.

El Capítulo 3 trata de la teoría que da pie al surgimiento de las Support Vector Machines, teoría que también es fundamento de las Support Multivector Machines. En este capítulo tambien se muestran resultados de clasificación en 2D (tanto binaria como multiclase) obtenidos con una SVM, para ello se utiliza el programa que desarrollé en lenguaje de programación C++ para plataforma LINUX. En dicho programa se implementaron los tres tipos de kernels más comúnmente empleados (identidad, polinomial y gaussiano). La motivación que nos impulsó a programar nuestra propia SVM fue la de comprender a fondo su funcionamiento y posteriormente emplear los módulos desarrollados como base para la elaboración y prueba de los kernels geométricos productos de esta tesis. Cabe mencionar que el rendimiento de la SVM implementada se comparó, tanto en tiempo como en precisión con la elaborada por el Departamento de Ciencias Computacionales e Ingeniería de la Universidad Nacional de Taiwan[4], obteniendo resultados muy similares en tiempos e idénticos en cuanto a resultado de clasificaciones.

El producto del trabajo de investigación de este trabajo se incluye en el Capítulo 4, en él se muestra el desarrollo matemático de los mapeos a los espacios de características, dados determinados espacios de entrada, el uso que se le dio al producto Clifford y a los multivectores en dichos mapeos, así como el kernel encontrado en cada caso y los resultados de clasificación que otorgan estos kernels geométricos. Se anexa en este capítulo el experimento de compresión de datos usando multivectores y el resultado de uno de los casos de estudio. También se incluye una aplicación de clasificación real, en la que se emplea uno de los kernels geométricos cuyas características de generalización son, para este problema en especial, más adecuadas que las del kernel gaussiano[5], ya que el primero de los kernels disminuye error de

[4]LIBSVM: *A library for Support Vector Machines (Version 2.33)*. Código, programa ejecutable y documentación disponible en http://www.csie.ntu.edu.tw/~cjlin

[5]Conocido por ser un aproximador universal, por lo que, en teoría todo conjunto de

clasificación a cero en los ejemplos de prueba del experimento, mientras que el gaussiano no puede lograrlo.

Las conclusiones obtenidas al realizar este trabajo están contenidas en el Capítulo 5, así como el trabajo futuro que se pretende realizar como continuación de éste. La bibliografía fundamental en la que nos apoyamos durante el trabajo de investigación antecede al Apéndice A, el cual trata de las bases del álgebra lineal necesarias para adentrarse en este documento, tales como espacios vectoriales, espacios vectoriales con producto punto y espacios de Hilbert. El Apéndice B contiene las definiciones básicas del Álgebra Geométrica y el C algunos cálculos matemáticos fundamentales de dichas álgebras.

entrenamiento puede ser clasificado con él.

Capítulo 2

ÁLGEBRA GEOMÉTRICA.

2.1. ¿Que es el Álgebra geométrica?

El Álgebra geométrica se puede definir como un lenguaje matemático poderoso para interpretar y expresar coherentemente ideas físicas . Este lenguaje permite unificar formalismos matemáticos estimulando la intuición geométrica [1]. En las siguientes líneas se intenta dar las bases que nos ayuden a comprender este sistema matemático, pero sin detenernos en conceptos básicos que el lector puede consultar en la amplia literatura de la matemática actual.

Tradicionalmente estamos acostumbrados a manejar dos tipos de cantidades en las aplicaciones de geometría e ingeniería: escalares y vectores. Si definimos un espacio vectorial V, tendremos las siguientes propiedades:

- $a + b = b + a$ (conmutatividad)
- $(a + b) + c = a + (b + c)$ (asociatividad)
- $a + 0 = a$ (elemento neutro: 0)

  $a + (-a) = 0$ (elemento opuesto: $-a$)
- $\lambda(a+b) = \lambda a + \lambda b$ (distribución del producto de un escalar con la suma de vectores)

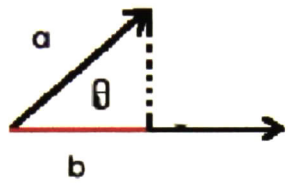


Figura 2.1: Producto punto.

- $(\lambda + \mu)a = \lambda a + \mu a$ (distribución del producto de la suma de escalares con un vector)
- $(\lambda\mu)a = \lambda(\mu a)$ (asociatividad del producto de vector con escalares)

Además tenemos definido el *producto punto* de dos vectores a y b, siendo éste un escalar con magnitud $|a|\,|b|\cos\theta$, donde $|a|$ y $|b|$ son las longitudes de a y b, y θ es el ángulo entre estos vectores.

La interpretación geométrica de este producto es que se realiza la proyección de un vector sobre otro; es decir, la componente de un vector en la dirección del otro. Este producto nos da una idea de la dirección de cada vector, puesto que si tenemos como resultado el escalar cero, significa que son perpendiculares (ver fig. 2.1).

Otro producto que se utiliza es el producto cruz ($a \times b$) el cual se define como el vector perpendicular a a y b con magnitud $|a|\,|b|\sin\theta$. La interpretación geométrica de este producto cruz, de hecho, esta "ligado" al espacio de 3D porque en 2D simplemente no existe una dirección perpendicular a dos vectores a y b en el plano (ver fig. 2.2).

Así surge un concepto más general: el producto exterior. El producto exterior o producto wedge tiene magnitud $|a|\,|b|\sin\theta$ pero no es ni un escalar ni un vector, es lo que llamaremos *bivector* o segmento de plano orientado. El producto exterior tiene la misma magnitud que el producto cruz y comparte

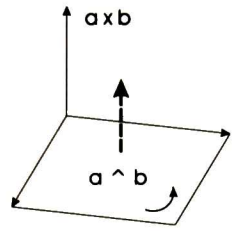


Figura 2.2: Interpretación geométrica del producto cruz y el producto wedge

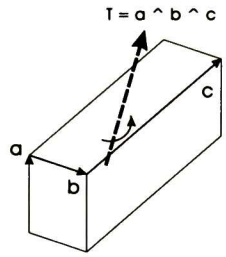


Figura 2.3: Interpretación geométrica de los trivectores

su propiedad anticonmutativa: $a \wedge b = -(b \wedge a)$.

Una forma de visualizar el producto exterior es imaginarlo como el segmento de plano resultante de deslizar el vector a a lo largo del vector b, como se muestra en la figura 2.2.

Esto nos lleva a una generalización del producto de objetos de dimensiones mayores. Por tanto, si el bivector $a \wedge b$ es deslizado a lo largo de otro vector c, entonces obtenemos un *trivector* o elemento de volumen orientado (ver fig. 2.3).

2.1.1. Producto Clifford o producto geométrico

Puesto que el producto punto es un escalar y el producto exterior es un bivector o área, estos disminuyen o aumentan, respectivamente, el "grado" de un vector. Además tienen propiedades de conmutación opuestas ya que

$$a \cdot b = b \cdot a \tag{2.1}$$

$$a \wedge b = -(b \wedge a) \tag{2.2}$$

Por tanto, podríamos pensar el estos dos productos como aquellos que forman la parte simétrica y antisimétrica de un nuevo producto que llamaremos *producto Clifford* o *producto geométrico:*

$$ab = a \cdot b + a \wedge b \tag{2.3}$$

A primera vista pareciera un tanto extraño el hecho de sumar dos cantidades diferentes: escalar y bivector; sin embargo, recordando los números complejos vemos que estamos haciendo algo muy parecido cuando representamos un objeto con una parte real y una imaginaria al cual llamamos "número complejo" Así que en este caso estamos haciendo exactamente lo mismo y llamamos a este nuevo objeto (formado por la suma de un escalar y un bivector) *"multivector"*

2.1.2. Definición general del álgebra geométrica $G_{p,q,r}$

En general, el álgebra geométrica $G_{p,q,r}$ es un espacio lineal de dimensión 2^n, donde $n = p+q+r$, con una estructura subespacial (elementos llamados *blades*) como entidades algebraicas de grado mayor en comparación con los vectores los cuales permiten representar los multivectores. Los subíndices p, q, r indican la cantidad de elementos de grado 1 de la base vectorial que elevados al cuadrado dan como resultado 1,-1 y 0, respectivamente. El álgebra geométrica $G_{p,q,r}$ se construye a partir de un espacio vectorial $\Re^{p,q,r}$ mediante la aplicación del producto Clifford definido en la sección anterior, de forma que el producto geométrico de dos vectores e_i , e_j que pertenecen a la base del espacio vectorial esta dado por

$$e_i e_j = \begin{cases} 1 & i = j \ \in \{1, ..., p\} \\ -1 & i = j \ \in \{p+1, ..., p+q\} \\ 0 & i = j \ \in \{p+q+1, ..., n = p+q+r\} \\ e_{ij} = e_i \wedge e_j = -e_j \wedge e_i & i \neq j \end{cases} \tag{2.4}$$

El espacio vectorial con $q \neq 0$ y $r \neq 0$ se llama pseudo-euclideano; si $r \neq 0$ su métrica es degenerada y el álgebra geométrica correspondiente se llama degenerada; pero debido a que en la práctica no tenemos elementos que directamente cuadreen a cero, sólo utilizaremos álgebras no degeneradas $G_{p,q}$ y, si se requiere, combinando elementos de p y q podemos formar vectores que cuadreen a cero[1].

El álgebra geométrica $G_{p,q}$ es entonces expandida por:

Elementos de grado 0 (*escalares*)	1
Elementos de grado 1 (*vectores*)	e_i
Elementos de grado 2 (*bivectores*)	$e_{ij} = e_i \wedge e_j$
Elementos de grado 3 (*trivectores*)	$e_{ijk} = e_i \wedge e_j \wedge e_k$
...	...
Elementos de grado n (n − *vectores*)	$e_{ijk,...,n} = e_i \wedge e_j \wedge e_k \wedge, ..., \wedge e_n$

y así sucesivamente hasta llegar al elemento (blade) de grado n, que es el elemento de mayor grado, el cual es llamado *pseudoescalar* y se denota mediante la letra I. El n-blade unitario es llamado pseudoescalar unitario y como en cada álgebra es un pseudoescalar diferente, utilizaremos un subíndice para indicar el álgebra geométrica a la que pertenece el pseudoescalar.

De esta manera, un elemento arbitrario X del álgebra geométrica $G_{p,q}$ está dado por

$$X = a_0 + \sum a_i e_i + \sum a_{ij} e_{ij} + \sum a_{ijk} e_{ijk} + ... + \sum a_I I \qquad (2.5)$$

donde $a \in \Re$ y los subíndices colocados en las letras a $(ijk...)$ sólo son para indicar que es el coeficiente del correspondiente blade formado por el producto exterior de los vectores $e_i, e_j, e_k, ...$; esto es, X se construye con la combinación lineal de blades de grado $0, ..., n$ (escalar, vectores, bivectores, ..., pseudoescalar).

Un concepto muy interesante es el de *dualidad*. El dual de un r-blade se

[1] En secciones posteriores se verán más detalles al respecto.

define por

$$X^* = XI \tag{2.6}$$

es decir, el dual lo obtenemos mediante el producto con el pseudoescalar unitario correspondiente al álgebra geométrica en la que se esté trabajando. Se puede observar que el dual de un r-blade es un $(n-r)$-blade.

Como una extensión podemos realizar el producto interior o producto punto de dos blades de forma que el producto de un r-blade con un s-blade se define recursivamente mediante

$$(a_1 \wedge ... \wedge a_r) \cdot (b_1 \wedge ... \wedge b_s) = \begin{cases} ((a_1 \wedge ... \wedge a_r) \cdot b_1) \cdot (b_2 \wedge ... \wedge b_s) & si\ r \geq s \\ (a_1 \wedge ... \wedge a_{r-1}) \cdot (a_r \cdot (b_1 \wedge ... \wedge b_s)) & si\ r < s \end{cases} \tag{2.7}$$

y

$$(a_1 \wedge ... \wedge a_r) \cdot b_1 = \sum_{i=1}^{r} (-1)^{r-i} a_1 \wedge ... \wedge a_{i-1} \wedge (a_i \cdot b_1) \wedge a_{i+1} \wedge ... \wedge a_r \tag{2.8}$$

$$a_1 \cdot (b_1 \wedge ... \wedge b_s) = \sum_{i=1}^{s} (-1)^{i-1} b_1 \wedge ... \wedge b_{i-1} \wedge (a_r \cdot b_i) \wedge b_{i+1} \wedge ... \wedge b_s \tag{2.9}$$

El lector interesado en más detalles puede referirse a [21].

2.2. Álgebra geométrica Euclideana tridimensional ($G_{3,0}$)

El álgebra geométrica $G_{3,0}$ (o simplemente G_3) se deriva de $\Re^3$, en la cual $n = p = 3$, y es adecuada para representar entidades y operaciones del espacio Euclideano 3D. Ahora bien, sabemos que en $\Re^3$ tenemos 3 vectores ortonormales en la base $\{e_1, e_2, e_3\}$ y que el álgebra geométrica G_3 se deriva de este espacio mediante la aplicación del producto geométrico de éstos vectores base (tal como se explicó en la sección anterior), por lo que G_3 tendrá $2^3 = 8$ elementos en su base:

$$\begin{aligned} G_3 &= span\{1,\ e_1, e_2, e_3,\ e_{12}, e_{23}, e_{31},\ e_{123} = I_3\} \\ &\quad\ escalar\ vectores\ bivectores\ pseudoescalar \end{aligned}$$

I_3 denota el pseudoescalar del espacio Euclideano tridimensional, el cual cuadrea a -1 (para detalles del cálculo del cuadrado de este pseudoescalar, ver el apéndice B).

En esta álgebra, podemos representar puntos, líneas y planos del espacio 3D de la siguente manera:

Punto: representa una posición en el espacio 3D y se puede expresar como la combinación lineal de los tres vectores de la base

$$u = u_1 e_1 + u_2 e_2 + u_3 e_3 \tag{2.10}$$

donde $u_i \in \Re$ (ver fig. 2.4).

Línea: se puede representar como un multivector (no homogeneo) usando el vector n para indicar la dirección de la línea y un bivector m representando el momento ($m = x \wedge n$ donde x es un punto perteneciente a la línea):

$$l = n + m \tag{2.11}$$

(ver fig. 2.4)

Plano: puede representarse como una entidad de un grado mayor que la línea en términos de la distancia Hesse del origen al plano (d) y el bivector unitario de dirección del origen al plano (n):

$$p = n + I_3 d \tag{2.12}$$

(ver fig. 2.4)

El lector puede consultar el apéndice A para verificar que una rotación en álgebra geométrica es dada por un bivector; es decir, un ***rotor*** R es un elemento de grado par[2] del álgebra G_3 que satisface $R\tilde{R} = 1$, donde $\tilde{R}$ es el

[2]La definición de grado de un elemento aparece en el apéndice B

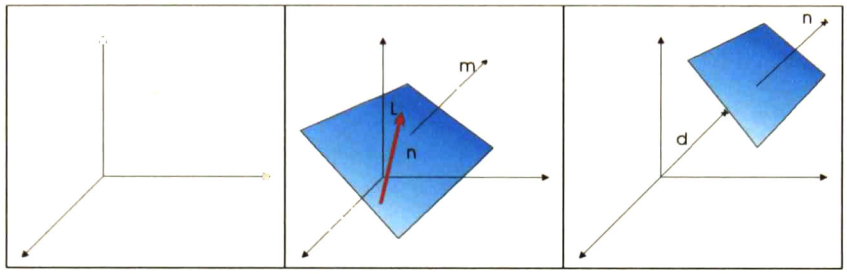


Figura 2.4: Punto (izquierda), linea (centro) y plano (derecha).

inverso[3] de R que se construye mediante la reversión[4] de sus blades:

$$R = u_0 + u_1 e_{12} + u_2 e_{23} + u_3 e_{31} \tag{2.13}$$

$$\widetilde{R} = u_0 - u_1 e_{12} - u_2 e_{23} - u_3 e_{31} \tag{2.14}$$

donde $u_i \in \Re$.

Así, la rotación de un punto representado por un vector x se realiza multiplicando el rotor R con el punto x por la izquierda y con su inverso $\widetilde{R}$ por la derecha:

$$x' = Rx\widetilde{R} \tag{2.15}$$

Una ventaja grande de los motores expresados en esta forma con respecto a las matrices de rotación de $\Re^3$ es que éstos trabajan no solo sobre puntos, sino para todos los tipos de objetos geométricos y se definen independientemente del grado y dimensión del espacio en que se trabaje. Además, el aplicar más de una rotación (composición de rotaciones) nos lleva a un nuevo rotor

$$x'' = R_2 x' \widetilde{R}_2 = R_2 R_1 x \widetilde{R_1} \widetilde{R_2} = Rx\widetilde{R} \tag{2.16}$$

Desgraciadamente, para el caso de las traslaciones no existe una forma multiplicativa de formalizarlas, a diferencia de los rotores, así que la traslación

[3]Puede consultar el concepto de inversión en el apéndice B

[4]La reversión se explica en el apéndice B.

en el espacio Euclideano se expresa en forma de suma

$$x' = x + t$$

Por lo que la composición de traslaciones es $t = t_1 + t_2$ lo cual nos lleva a observar que la traslación no es una operación lineal ya que para dos vectores x e y se tiene $T(x+y) = (x+y+t) \neq T(x) + T(y) = (x+y+2t)$.

Debido a esta no linearidad de la traslación en $G_{3,0}$ se utilizan otras álgebras de mayor dimensión para permitir la linearidad de esta operación con puntos u otras entidades.

2.3. Álgebra geométrica proyectiva ($G_{3,1}$)

Para incrementar la dimensión del espacio vectorial utilizamos las *coordenadas homogeneas*, con lo cual aumentamos en 1 la dimensión obteniendo un álgebra geométrica cuya base tiene $2^4 = 16$ elementos

$$G_{3,1} = span\{1,\ e_1, e_2, e_3, e_{\_},\ e_{12}, e_{23}, e_{31},\ e_{\_1}, e_{\_2}, e_{\_3},\ e_{123}, e_{\_12}, e_{\_23}, e_{\_31},\ e_{\_123} = I_P\} \tag{2.17}$$

Nota: $e_{\_123} = e_{\_} \wedge e_1 \wedge e_2 \wedge e_3$ y $e_{\_123}^2 = -1$

El vector de base adicional $e_{\_}$ denota el componente homogeneo que se añade a un punto. A diferencia de lo que se vió en la sección anterior donde la representación de entidades era un poco elaborada, la representación de lineas y planos en $G_{3,1}$ se da por un multivector resultado del producto exterior (wedge) de puntos homogéneos como se muestra enseguida:

Punto: un punto x en $G_{3,0}$ se representa en $G_{3,1}$ mediante un 1-vector dado por

$$X = x + e_{\_} \tag{2.18}$$

Línea: se representa como el producto exterior de dos puntos (homogeneos), lo cual nos lleva a un 2-vector

$$L = X_1 \wedge X_2 \tag{2.19}$$

$$\begin{aligned} &= (x_1 + e\_) \wedge (x_2 + e\_) \\ &= (x_1 \wedge x_2) + (x_1 \wedge e\_) + (e\_ \wedge x_2) + (e\_ \wedge e\_) \\ &= (x_1 \wedge x_2) + (x_1 - x_2) \wedge e\_ \\ &= m + ne\_ \end{aligned}$$

Se observa que la línea contiene elmomento $m = x_1 \wedge x_2$ y la dirección $r = x_1 - x_2$ (codificada en los 2-blades que contienen a $e\_$). En algunos casos es conveniente usar la representación de la línea en orden opuesto, es decir: $L = n + e\_m$ (ver fig. 2.4).

Plano: se representa como el producto exterior de tres puntos (homogeneos), dando como resultado un 3-vector

$$\begin{aligned} P &= X_1 \wedge X_2 \wedge X_3 \qquad (2.20) \\ &= (x_1 + e\_) \wedge (x_2 + e\_) \wedge (x_3 + e\_) \\ &= (x_1 \wedge x_2 \wedge x_3) + (x_1 - x_2) \wedge (x_1 \wedge x_3) \wedge e\_ \\ &= dI_3 + ne\_ \end{aligned}$$

Esta descripción formaliza al plano por la normal n a éste y la distancia Hesse d del origen al plano (ver fig. 2.4).

Como se observa, el generar entidades de mayor orden (como líneas y planos) es más natural en esta álgebra que en el álgebra geométrica euclideana $G_{3,0}$ ya que resulta del álgebra de incidencia de puntos. Así tenemos una ventaja más con respecto a $G_{3,0}$, pero aún no contamos con una forma adecuada para representar otro tipo de entidades como son círculos, esferas o pares de puntos y es ésto precisamente lo que nos lleva a analizar el álgebra geométrica conformal que se verá más adelante. Pero antes, para introducir el álgebra geométrica conformal, analizamos el plano de Minkowsky $G_{1,1}$ en la siguiente sección.

2.4. Álgebra de Minskowski ($G_{1,1}$)

Como se mencionó en la sección 2.1.2, el álgebra $G_{p,q}$ (o pseudo Euclideana) se genera a partir del espacio vectorial $\Re^{p,q}$ mediante la aplicación del producto geométrico. Si $q = 0$ se dice que el espacio tiene *signatura* (signature) *Euclideana* y si $q = 1$ se dice que es *signatura Minkowsky.*

El álgebra $G_{1,1}$ proviene del espacio vectorial $\Re^{1,1}$(también llamado el *plano de Minkowsky*) cuya base es $\{e_+, e_-\}$ cuyas propiedades son

$$e_+^2 = 1 \quad e_-^2 = -1 \quad e_+ \cdot e_- = 0 \tag{2.21}$$

En este punto podemos introducir una base nula (la llamamos nula porque estos vectores al elevarlos al cuadrado dan cero)

$$e_0 = \frac{1}{2}\left(e_+ - e_-\right) \tag{2.22}$$

$$e = e_+ + e_- \tag{2.23}$$

Estos vectores se pueden interpretar como el origen (e_0) y el punto en el infinito (e_∞).

El pseudoescalar unitario para $G_{1,1}$ se define por

$$E = e_0 \wedge e = e_+ \wedge e_- = e_+ e_- \tag{2.24}$$

y se puede mostrar que cumple con las siguientes propiedades:

$$\begin{aligned} e_0^2 = e^2 = 0 \qquad & e \cdot e_0 = -1 \qquad & E = e_+ e_- \\ E^2 = 1 \qquad & Ee = -e \qquad & Ee_0 = e_0 \\ e_+ E = e_- \qquad & e_- E = e_+ \qquad & (Absorcion) \\ e_- e = -(E+1) \qquad & e \wedge e_- = E \qquad & e_+ \cdot e = 1 \end{aligned} \tag{2.25}$$

$$e_+ e = E + 1 \tag{2.26}$$

Los vectores base y los vectores nulos se ilustran en la figura 2.5:

Estos conceptos servirán para extender el espacio $\Re^3$ (espacio Euclideano)

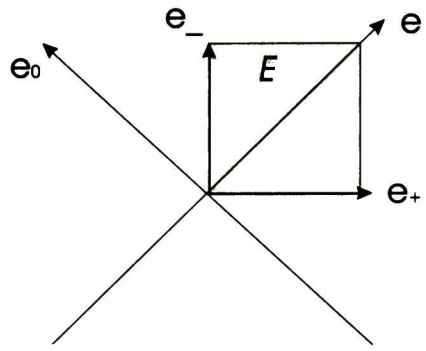


Figura 2.5: Vectores base y vectores nulos en el plano de Minkowsky (E)

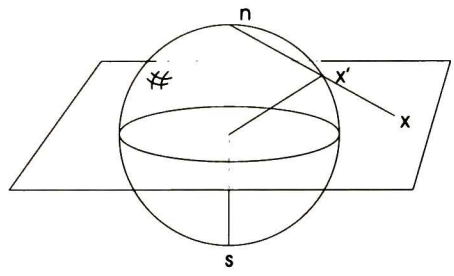


Figura 2.6: Proyección estereográfica

a $\Re^{4,1} = \Re^3 \oplus \Re^{1,1}$ dando como resultado el álgebra geométrica conformal que se ve en la siguiente sección.

2.5. El álgebra geométrica conformal ($G_{4,1}$)

La idea detrás de la geometría conformal es interpretar los puntos como puntos proyectados estereográficamente[5] Consideremos la figura 2.6:

Imaginemos que colocamos una fuente de luz en el polo norte (marcado como **n** en la figura 2.6); entonces cada punto en la esfera proyecta una sombra en el papel (plano en la figura). Para simplificar los cálculos analicemos cómo se proyectan puntos de la esfera al plano y viceversa para el caso 1D,

[5]Proyección estereográfica es una forma de hacer un mapa plano de la tierra

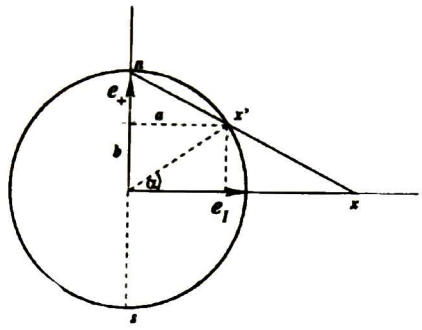


Figura 2.7: Proyección de puntos del círculo al plano y viceversa

como se muestra en la figura 2.7 (la esfera ahora es un círculo; asumimos radio 1 por simplicidad):

Además del vector del espacio 1D e_1 se añade el vector e_+perpendicular a éste. La proyección de un punto $x' = ae_1 + be_+$ del círculo sobre el eje e_1 esta dada por[6]:

$$x = \left(\frac{a}{1-b}\right) e_1 \tag{2.27}$$

y para proyectar un punto ce_1 ($c \in \Re$) hay que calcular los factores apropiados a, b para el vector $x' = ae_1 + be_+$ lo cual nos lleva a:

$$x' = \frac{2c}{c^2+1} e_1 + \frac{c^2-1}{c^2+1} e_+ \tag{2.28}$$

y si utilizamos coordenadas homogeneas, la representación homogenea del punto en el círculo es

$$x' = ce_1 + \frac{1}{2}\left(c^2-1\right) e_+ + \frac{1}{2}\left(c^2+1\right) e_- \tag{2.29}$$

Observe que $e_1^2 = e_+^2 = 1$ y $e_-^2 = -1$ con lo que además de tener la ventaja de la representación homogénea de puntos también estamos trabajando en

[6]Es fácil deducir la expresión si utilizamos semejanza de triángulos.

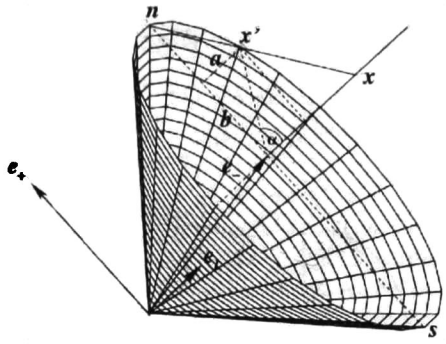


Figura 2.8: Cono nulo para el caso 1D.

un espacio de Minkowsky. Así, al elevar x' al cuadrado tenemos:

$$\begin{aligned}(x')^2 &= c^2 + \left(\frac{1}{2}\left(c^2-1\right)\right)^2 - \left(\frac{1}{2}\left(c^2+1\right)\right)^2 \\ &= c^2 + \frac{1}{4}\left[c^4 - 2c^2 + 1 - c^4 - 2c^2 - 1\right] \\ &= 0\end{aligned}$$

Así, los punto Euclideanos x, proyectados estereográficamente sobre el círculo son representados por el conjunto de vectores nulos de este nuevo espacio. Esta representación homogénea de puntos se usa como representación de puntos en el Álgebra geométrica conformal. La figura 2.8 visualiza este modelo homogeneo para proyecciones estereográficas para el caso 1D.

De la representación anterior y teniendo en cuenta lo analizado en la sección (2.4) se observa que estamos involucrando el espacio de Minkowsky. El papel del plano de Minkowsky es generar vectores nulos para extender el espacio Euclideano $\Re^n$ a $\Re^{n+1,1} = \Re^n \oplus \Re^{1,1}$ que genera el álgebra geométrica conformal $G_{n+1,1}$. Para el caso de $\Re^3$ se genera el espacio conformal $\Re^{4,1}$ cuya base es $\left\{e_1, e_2, e_3, e_+, e_-\right\}$ generando el álgebra geométrica $G_{4,1}$ de dimensión $2^5 = 32$, expandida por escalar, bivectores, 3-vectores, 4-vectores y el pseudoescalar I_C Note que el pseudoescalar $I_C = e_{123+-} = e_{+-123} = EI_3$ (E es el plano de Minkowsky e I_3 es el pseudoescalar Euclideano tridimen-

sional).

Tomemos la ecuación 2.29 y veamos que

$$\begin{aligned} x' &= ce_1 + \frac{1}{2}\left(c^2 - 1\right)e_+ + \frac{1}{2}\left(c^2 + 1\right)e_- \\ &= ce_1 + \frac{1}{2}c^2\left(e_- + e_+\right) + \frac{1}{2}\left(e_- - e_+\right) \\ &= ce_1 + \frac{1}{2}c^2 e + e_0 \end{aligned}$$

Por lo que extendiendo para un vector $x \in \Re^n$ tendríamos su transformación al espacio $\Re^{n+1,1}$ expresada por

$$\underline{x} = x + \frac{1}{2}x^2 e + e_0 \tag{2.30}$$

Los vectores base $\{e, e_0\}$ solamente nos permiten una representación más compacta.

Entonces, en esta álgebra estamos considerando puntos del llamado cono nulo que satisfacen las propiedades[7]

$$\left\{\underline{x} \in \Re^{4,1} \mid \underline{x}^2 = 0\,,\, \underline{x} \cdot e = -1\right\}$$

Así como en las secciones 2.2 y 2.3 vimos la manera de representar entidades de acuerdo a lo que cada una de las álgebras tratadas nos facilitan, también lo haremos aquí. Las entidades básicas en el espacio conformal de 3D son las esferas $\underline{s}$ que tienen el centro p y radio ρ:

$$\underline{s} = p + \frac{1}{2}\left(p^2 - \rho^2\right)e + e_0 \tag{2.31}$$

por lo que podemos interpretar el punto $\underline{x}$ (ver 2.30) como una esfera degenerada de radio $\rho = 0$. La forma dual[8] de una esfera tiene la ventaja de que

[7]Estas propiedades se vieron con anterioridad en esta sección o bien son fáciles de comprobar

[8]Recuerde que el dual de una entidad se calcula al multiplicarlo por el pseudoescalar correspondiente al álgebra en que se trabaja

Cuadro 2.1: Representación y representación dual de entidades en álgebra geométrica conformal

Entidad	Representación	Gdo.	Representación dual	Gdo.
Esfera	$\underline{s} = p + \frac{1}{2}\left(p^2 - \rho^2\right) e + e_0$	1	$\underline{s}^* = \underline{a} \wedge \underline{b} \wedge \underline{c} \wedge \underline{d}$	4
Punto	$\underline{s} = x + \frac{1}{2}x^2 e + e_0$	1	$\underline{x}^* = \left(-Ex - \frac{1}{2}x^2 e + e_0\right) I_E$	4
Línea	$\underline{L} = nI_E + emI_E$	2	$\underline{L}^* = e \wedge a \wedge b$	3
	$n = a - b$			
	$m = a \wedge b$			
Plano	$\underline{P} = nI_E - de$	1	$\underline{P}^* = e \wedge a \wedge b \wedge c$	4
	$n = (a - b) \wedge (a - c)$			
	$d = (a \wedge b \wedge c) I_E$			
Círculo	$\underline{z} = \underline{s}_1 \wedge \underline{s}_2$	2	$\underline{z}^* = \underline{a} \wedge \underline{b} \wedge \underline{c}$	3
	$\underline{P}_z = \underline{z} \cdot e \ , \ \underline{L}_z = \underline{z} \wedge e$		$\underline{p}_z = \underline{P}_z^* \wedge \underline{L}_z^*$	
	$\rho = \frac{\underline{z}}{(e \wedge \underline{z})^2}$			
Par de puntos	$\underline{PP} = \underline{s}_1 \wedge \underline{s}_2 \wedge \underline{s}_3$	3	$\underline{PP}^* = \underline{a} \wedge \underline{b} \ , \ \underline{X}^* = \underline{e} \wedge \underline{x}$	2

se puede calcular directamente de puntos sobre la esfera

$$\underline{s}^* = \underline{a} \wedge \underline{b} \wedge \underline{c} \wedge \underline{d}$$

Note que un punto está en la esfera si y solo si $\underline{x} \wedge \underline{s}^* = 0$ o bien si $\underline{x} \cdot \underline{s} = 0$, dependiendo si trabajamos en la representación dual o normal de la entidad. A partir de esta entidad básica (esfera) podemos definir otras entidades; esto se resume en la tabla 2.1.

Para mayores detalles sobre entidades en álgebra geometrica conformal se puede consultar [23].

Ahora nos enfocaremos en la forma de realizar transformaciones de movimiento rígido en esta álgebra. Se recordará que en la sección 2.2 se concluyó que las rotaciones en el espacio $\Re^3$ son tranformaciones lineales en el álgebra $G_{3,0}$, mientras que las traslaciones se comportan como no lineales.

Considerando la figura 2.6 que ejemplifica el caso 2D, se observa que la rotación de un punto x sobre el plano nos lleva a x' y considerando la proyección de estos puntos en la esfera ($\underline{x}$ y $\underline{x'}$) se observa que la rotación es exactamente la misma; sin embargo, una traslación en el plano corresponde, para los puntos proyectados en la esfera, a una rotación espacial sobre un eje

del plano. De hecho, una rotación se puede estimar en la misma forma como en G_2 o G_3 , pero una trasslación es un caso especial de rotación en $G_{3,1}$ o $G_{4,1}$.

Así como en $G_{3,0}$, las rotaciones en $G_{4,1}$ se expresan por medio de los rotores

$$R = \exp\left(\frac{\theta}{2}l\right) \tag{2.32}$$

donde l representa el bivector unitario dual al eje de rotación y el ángulo θ representa la cantidad de rotación. La rotación de una entidad (punto, línea, plano, círculo, esfera o par de puntos) se realiza simplemente multiplicando la entidad por la izquierda con el rotor R y por la derecha con el inverso $\widetilde{R}$.

Si deseamos trasladar una entidad con respecto a un vector de traslación t usamos el llamado *traslador*

$$T = \left(1 + \frac{et}{2}\right) = \exp\left(\frac{e}{2}t\right) \tag{2.33}$$

De manera similar a las rotaciones, las entidades se trasladan muliplicándolas por la izquierda con el traslador T y por la derecha con su inverso $\widetilde{T}$.

Se recordará que en la sección 2.2 se mencionó que la multiplicación de dos rotores da como resultado un nuevo rotor ($R = R_2R_1$), por tanto, considerando el traslador como un rotor podemos multiplicar T y R con lo que obtenemos lo que se llama ***motor*** y que sirve para expresar movimiento rígido (rotación y traslación):

$$M = TR \tag{2.34}$$

El nombre de "motor" es una abreviación de "moment and vector" (para mayor información referirse a [22]) y es un multivector de grado par especial. Con este nuevo operador, la rotación y traslación de un punto esta dada por $\underline{x}' = M\underline{x}\widetilde{M}$ donde $\widetilde{M}$ es el inverso de M y esta dado por $\widetilde{M} = \widetilde{R}\widetilde{T}$.

Nótese que este operador no solo puede ser aplicado a puntos, sino a cualquier entidad.

2.5.1. $G_{3,1}$ como subalgebra de $G_{4,1}$

Como se mencionó en la sección 2.3, si añadimos un componente homogeneo a los puntos tenemos varias ventajas como una representación más natural de entidades como líneas y planos. Si en lugar de añadir el vector $e_{\_}$ utilizamos el vector nulo e para el componente homogeneo, podemos ver que mantenemos las representaciones:

Punto: un punto homogéneo estará dado por

$$X = x + e \tag{2.35}$$

Línea: se representa como el producto exterior de dos puntos (homogeneos):

$$\begin{aligned} L &= X_1 \wedge X_2 \\ &= (x_1 + e) \wedge (x_2 + e) \\ &= (x_1 \wedge x_2) + (x_1 \wedge e) + (e \wedge x_2) + (e \wedge e) \\ &= (x_1 \wedge x_2) + (x_1 - x_2) \wedge e \\ &= m + ne \end{aligned} \tag{2.36}$$

(La línea contiene elmomento $m = x_1 \wedge x_2$ y la dirección $n = x_1 - x_2$ (codificada en los 2-blades que contienen a $e_{\_}$ y e_{+} que son los elementos con que se forma e).

Plano: se representa como el producto exterior de tres puntos (homogeneos)

$$\begin{aligned} P &= X_1 \wedge X_2 \wedge X_3 \\ &= (x_1 + e) \wedge (x_2 + e) \wedge (x_3 + e) \\ &= (x_1 \wedge x_2 \wedge x_3) + (x_1 - x_2) \wedge (x_1 - x_3) \wedge e \\ &= dI_E + ne \end{aligned} \tag{2.37}$$

Esto es muy parecido a los conceptos de *plano afino* que se pueden encontrar en [22] y además nos habilita para utilizar los conceptos

de distancia dirigida y relaciones de incidencia.

Capítulo 3

SUPPORT VECTOR MACHINES (SVM)

3.1. Aprendizaje Supervisado.

D.O.HEBB en su publicación "The Organization of Behavior, A Neuropsychological Theory" define el aprendizaje biológico en el postulado neuropsicológico, en el que afirma que "Cuando el axón de una célula A está lo suficientemente cerca para excitar una célula B y repetida o persistentemente dispara hacia ésta, algún proceso de crecimiento o cambio metabólico toma lugar en una o ambas células, tal que la eficiencia de A para disparar hacia B se incrementa".

El procedimiento utilizado para lograr que un sistema computacional adquiera cierto comportamiento a partir de ejemplos determinados es conocido como la *metodología del aprendizaje,* cuando se da el caso particular que dichos ejemplos son pares de información de entrada/salida deseada es llamado *aprendizaje supervisado.* Esto es, intentar que un sistema computacional aprenda que dada una entrada, debe de otorgar una salida deseada de un conjunto de datos en el mismo modo en que un niño aprende cuáles frutas son manzanas con mostrarle un gran número de diferentes tipos de éstas en lugar de darle una especificación precisa de lo que es una manzana, es conocido como la *metodología de aprendizaje supervisado.*

Los pares de entrada/salida deseada típicamente reflejan relaciones funcionales o mapeos de las entradas hacia las salidas. Cuando existe una función subyacente para mapear de las entradas a las salidas deseadas se le conoce como *función objetivo*. En el caso de problemas de clasificación esta función es llamada *función de decisión*.

3.2. Support Vector Machines para el aprendizaje.

Las Support Vector Machines (SVM) son sistemas de neurocomputación que aprenden, usando la hipótesis de espacios de funciones lineales llevadas a un espacio dimensional más alto, conocido como el espacios de características (*feature spaces*) usando kernels; son entrenadas con algoritmos de aprendizaje obtenidos de la teoría de optimización, su principal característica es que, a diferencia de otros sistemas neuronales, no paran su aprendizaje hasta que encuentran soluciones óptimas. Son utilizadas para tareas de clasificación y regresión principalmente. En este trabajo de tesis se explota su uso sólo para clasificación.

3.2.1. Teoría de la Generalización de Vapnik y Chervonenkis (VC)

La teoría de Vapnik y Chervonenkis es la más apropiada para describir las SVMs e históricamente ha motivado el estudio de estas últimas [3]. Esta subsección trata de revisar los principales resultados de la teoría VC que hacen posible limitar la complejidad de las funciones lineales en espacios kernel, en los que se lleva a cabo la clasificación

La tarea de clasificación consiste en encontrar una regla, la cual, basada en observaciones externas, asigne a un objeto una clase determinada de entre varias clases más. En el caso más simple existen sólo dos clases diferentes. Una posible formalización de esta tarea es estimar una función $f : \Re^n \rightarrow \{-1, +1\}$, usando pares de datos de entrenamiento entrada/salida deseada generados de acuerdo a una distribución de probabilidad $\boldsymbol{P(x,y)}$ desconocida, tal que f clasificaría correctamente ejemplos $(\boldsymbol{x},y)$ nunca antes vistos por la máquina.

La mejor función f que podemos obtener es aquella que minimice el *riesgo, error esperado o de generalización:*

$$R[f] = \int l(f(\mathbf{x}), y) dP(\mathbf{x}, y) \tag{3.1}$$

donde l denota una *función de pérdida* escogida de entre varias, por ejemplo $l(f(\mathbf{x}), y) = \theta(-yf(x))$, donde $\theta(z) = 0$ para $z<0$ y $\theta(z) = 1$ en otro caso (la llamada pérdida 0/1).

Desafortunadamente el *riesgo o error esperado* no puede ser minimizado directamente ya que la distribución de probabilidad $P(x,y)$ de los datos de entrenamiento no es conocida de antemano. Sin embargo, se trata de estimar una función que esté 'cercana' a la función óptima basados en la información disponible, por ejemplo en los datos de entrenamiento y las propiedades de la *clase de funciones* **F** (por ejemplo: planos en 2D), de esta manera se podría escoger la función f. Una forma particular consiste en aproximarse al mínimo del *riesgo o error esperado (error de generalización)* por medio del *riesgo empírico o error de entrenamiento:*

$$R_{emp} = \frac{1}{n} \sum_{i=1}^{n} l(f(\mathbf{x}_i), y_i) \tag{3.2}$$

Es posible dar las condiciones para las que la máquina de aprendizaje asegure que asintóticamente (cuando n$\rightarrow \infty$), el *riesgo empírico* converja hacia el *riesgo esperado.* Sin embargo, para un número pequeño de datos de entrenamiento son posibles grandes desviaciones y el problema del *sobre entrenamiento (overfitting)* puede ocurrir. En la figura 3.1 se ilustra este dilema. Dado un conjunto de entrenamiento pequeño (fig. 3.1 izquierda), tanto la hipótesis de la línea sólida como de la línea punteada pueden ser ciertas, la punteada es más compleja, pero también presenta un error de entrenamiento (o error empírico) menor (ver ecuación 3.2). Sólo con un conjunto de entrenamiento grande somos capaces de observar cuál decisión representa la distribución de los datos de una manera más certera. Si la hipótesis punteada es correcta la sólida puede presentar falta de entrenamiento (imágen central); si la hipótesis sólida es correcta, la punteada presentará sobreentrenamiento

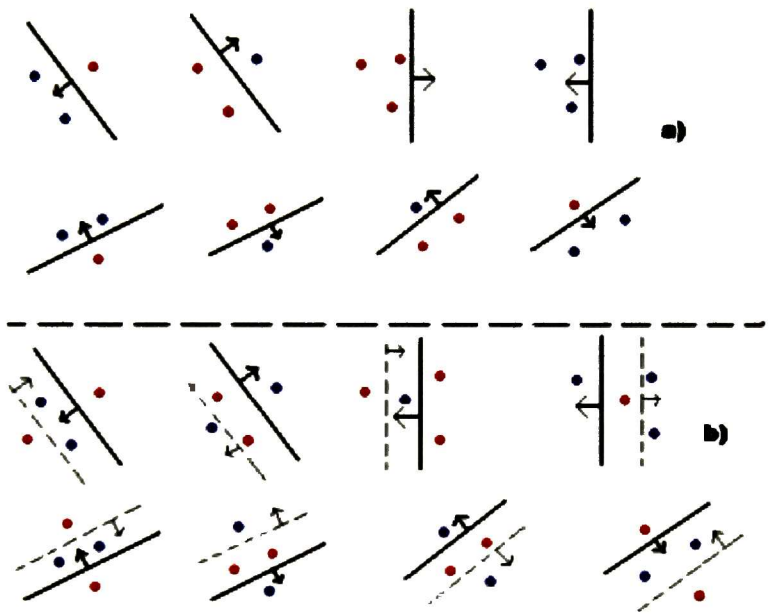


Figura 3.2: a) Tres puntos en $\Re^2$, clasificados por lineas orientadas, b) para cuatro puntos en $\Re^2$, son necesarias dos líneas.

de posición de los puntos restantes son linealmente independientes.

Corolario. La dimensión de un conjunto de hiperplanos orientados en $\Re^n$ es n+1 *ya que siempre podemos escoger n+1 puntos, y después escoger uno de estos puntos como origen, de tal manera que los vectores de posición de los restantes n puntos son linealmente independientes, pero nunca podemos escoger n+2 puntos (ya que no existen n+1 vectores en* $\Re^n$ *que puedan ser linealmente independientes).*

Si construimos una familia de clases de funciones $F_1 \subset ... \subset F_k$ con una dimensión VC creciente, el *principio de minimización de riesgo estructural* (SRM)[1] procede como sigue: Sean $f_1, ..., f_k$ las soluciones para la minimización del riesgo empírico dadas las clases de funciones F_i. El SRM escoge la clase de funciones F_i (y la función solución f_i) de tal manera que el límite superior en el error de entrenamiento o generalización es minimizado, lo cual puede ser calculado haciendo uso del siguiente teorema, presentado por Vladimir Vapnik [3]:

Teorema Sea h la dimensión VC de la clase de funciones F y R_{emp} *el riesgo definido en ecuación 3.2 usando la pérdida 0/1, n el total de muestras de entrenamiento. Para toda* $\delta > 0$ *y* $f \epsilon F$ *la desigualdad que limita el riesgo o error de generalización es:*

$$R[f] \leq R_{emp}[f] + \sqrt{\frac{h(ln\frac{2n}{h} + 1) - ln(\delta/4)}{n}} \tag{3.3}$$

en la que el término de la parte derecha del signo de suma se conoce como *término de confianza.*

El objetivo es minimizar el error de generalización o riesgo $R[f]$ lo cual puede ser obtenido por medio de un pequeño error de entrenamiento o riesgo empírico $R_{emp}[f]$ mientras mantengamos la VC dimensión (h) de la clase de función lo más pequeño que podamos, es decir mantener la complejidad de la clase de funciones lo más simple que se pueda.

Dos extremos se pueden observar de la desigualdad anterior (3.3): (i) una clase de función muy simple (con baja dimensión VC) produce que el térmi-

[1] Para un análisis más profundo de la Teoría VC y el SRM el lector puede consultar las referencias[2, 3, 13, 11]

no de la raíz cuadrada de la desigualdad (confianza) casi se desvanezca, pero aún así un error empírico o de entrenamiento puede presentarse, mientras que (ii) una clase de funciones muy complicada (con alta dimensión VC) puede desaparecer casi por completo el error empírico, pero eleva considerablemente el término de la raíz cuadrada de la desigualdad. La mejor clase de clase de funciones es generalmente aquella que nos mantiene en el punto medio de estos dos extremos, que pueda separar los datos y que mantenga un riesgo empírico bajo. En la figura 3.3 la línea verde representa el error de entrenamiento o riesgo empírico (3.2), la línea roja representa el límite superior en el término de complejidad (confianza, parte derecha de la ecuación 3.3). Con complejidad alta el error empírico decrece, pero el límite superior en el riesgo de la confianza se vuelve peor. Para una complejidad certera, la clase de funciones con el mejor riesgo esperado es obtenida. Entonces, en la práctica el objetivo es encontrar el mejor promedio entre el riesgo empírico y la complejidad.

Las siguientes desigualdades limitan la dimensión VC y la ligan con la magnitud del vector de peso w :

$$h \leq \Lambda^2 R^2 + 1 \quad y \quad ||w|| \leq \Lambda \tag{3.4}$$

donde R es el radio del círculo más pequeño que encierre datos. Por lo tanto si maximizamos el margen de una clase de función, en $\frac{2}{\Lambda}$, se puede controlar la dimensión VC.

3.2.2. Teoría de la optimización.

La teoría de la optimización es la rama de las matemáticas que se encarga de caracterizar problemas en los cuales funciones deben ser minimizadas (o maximizadas) , y desarrollar algoritmos para solucionarlos, dichos problemas pueden incluir restricciones de igualdad y desigualdad. En esta subsección se describen las características de los llamados problemas de optimización en los cuales las funciones objetivo son del tipo cuadráticas convexas, mientras que las restricciones son lineales. Este tipo de problemas de optimización

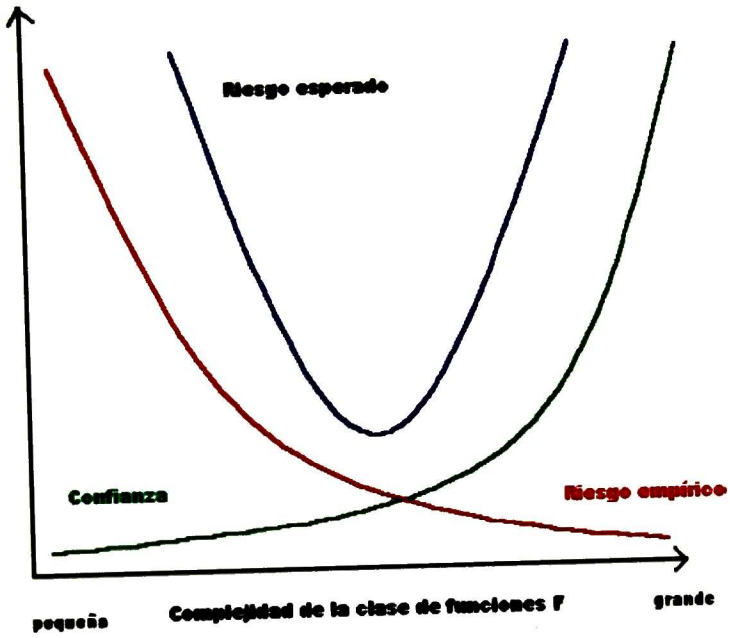


Figura 3.3: Ilustración esquemática de la ecuación 3.3.

son llamados *programas convexos cuadráticos* y son los adecuados para el entrenamiento de las Support Vector Machines [2].

El problema general de optimización se le llama *problema de optimización primal* en el cual dadas las funciones f, g_i , y h_j (con $i = 1, ..., k$ y $j = 1, ..., m$ respectivamente) definidas en un dominio $\Omega \in \Re^n$.

$$Minimizar \quad f(w) \quad w \in \Omega \tag{3.5}$$

$$Sujeto \quad g_i(w) \leq 0, \quad i = 1, ..., k \tag{3.6}$$

$$h_j(w) = 0 \quad j = 1, ..., m \tag{3.7}$$

donde $f(w)$ es llamada la *función objetivo*, y las siguientes relaciones son llamadas respectivamente *restricciones de desigualdad e igualdad.*

Un problema de optimización en el cual la función objetivo, las restricciones de igualdad y desigualdad son todas funciones lineales es llamado un *programa lineal.* Si la función objetivo es cuadrática mientras que las restricciones son todas lineales, el problema de optimización es llamado un *programa cuadrático.*

Una constante de desigualdad $g_i(w) \leq 0$ se dice que es *activa* si la solución $\mathbf{w}^*$ satisface que $g(\mathbf{w}^*) = 0$, de otro modo se denomina *inactiva.* En este sentido las restricciones de igualdad son siempre activas. Algunas veces, cantidades llamadas *variables flojas* y denotadas por ξ son introducidas para transformar una restricción de desigualdad en una de igualdad como sigue:

$$g_i(w) \leq 0 \iff g_i(w) + \xi_i = 0 \quad con \quad \xi_i \geq 0$$

Las variables *flojas* son asociadas con restricciones activas iguales a cero, mientras que para aquellas restricciones inactivas indican cierto monto de *pérdida* en la restricción.

Los problemas de optimización tratados por las Support Vector Machines siempre tienen solución, esto debido a que pertenecen a cierta clase conocida como *programas cuadráticos convexos.*

Una función real $f(w)$ es llamada *convexa* para $\mathbf{w} \in \Re^n$ si, $\forall \mathbf{w}, u \in \Re^n$

[2]Un estudio profundo de la teoría de optimización y los problemas cuadráticos convexos se encuentra en la referencia [17]

y para cualquier $\theta \in (0,1)$ se cumple que:

$$f(\theta\mathbf{w} + (1-\theta)\mathbf{u}) \leq \theta f(\mathbf{w}) + (1-\theta)f(\mathbf{u}) \tag{3.8}$$

Si se mantiene la desigualdad en forma estricta, la función se denomina *estrictamente convexa.*

Una función que es dos veces diferenciable será convexa si su matriz Hessesiana es *positiva semi-definida*, esto es, todos sus eigen valores son positivos.

Una función se llama *afina* si puede expresarse de la forma $f(\mathbf{w}) = Aw+b$, para alguna matriz $\mathbf{A}$ y un vector $\mathbf{b}$. Las funciones afinas son convexas ya que ellas tienen una matriz Hessesiano que es cero.

Un conjunto $\Omega \subseteq \Re^n$ es llamado *convexo* si, $\forall \mathbf{w}, u \in \Omega$ y para cualquier $\theta \in (0,1)$, $(\theta\mathbf{w} + (1-\theta)\mathbf{u}) \in \Omega$.

Si una función f es convexa, cualquier local mínimo w^* del problema de optimización con función objetivo f es también un global mínimo, ya que para cualquier $\mathbf{u} \neq w^*$, por definición de local mínimo, existe θ suficientemente cerca de 1 tal que

$$\begin{aligned} f(w^*) &\leq f(\theta w^* + (1-\theta)u) && (3.9) \\ &\leq \theta f(w^*) + (1-\theta)f(u) && (3.10) \end{aligned}$$

Se sigue que $f(w^*) < f(u)$. Esta es una propiedad de las funciones convexas que permite que los problemas de optimización sean resolubles cuando las funciones y los conjuntos son convexos.

Un problema de optimización en el cual el conjunto Ω, la función objetivo y todas las restricciones son convexas, se dice ser *convexo.*

Para el propósito de entrenamiento de una SVM se restringe al caso en el que las restricciones son lineales, la función objetivo es convexa y cuadrática y $\Omega \in \Re^n$, entonces consideramos *programas cuadráticos convexos.*

Teorema. Para funciones objetivo cuadráticas F, el Hessesiano es positivo definido si y solo si F es estrictamente convexa, lo que no se cumple si la

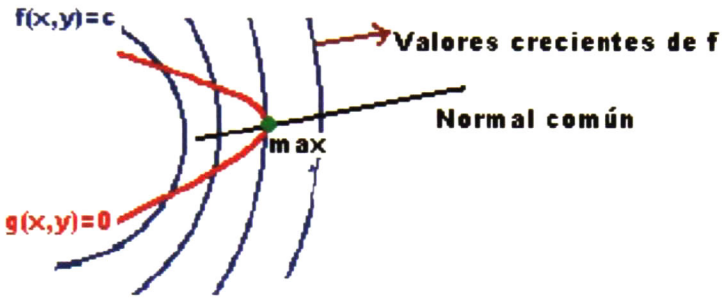


Figura 3.4: Ilustración de los multiplicadores de Lagrange.

función F no es cuadrática, en estos casos, un Hessesiano positivo definido, implica una función objetivo estrictamente convexa pero no viceversa.

3.2.2.1 Teoría de optimización de Lagrange y Kuhn-Tucker.

El propósito de la teoría de Lagrange es caracterizar la solución de un problema de optimización, inicialmente donde no hay restricciones de desigualdad. Los principales conceptos de esta teoría son los multiplicadores de Lagrange y la función de Lagrange. Este método fue desarrollado por Lagrange en 1797 para problemas mecánicos, generalizando un resultado de Fermat de 1629. En 1951 Kuhn y Tucker extendieron el método para permitir restricciones de desigualdad, lo que es conocido como teoría de Kuhn-Tucker.

Teorema (Fermat) Una condición necesaria para que $\mathbf{w}^*$ *sea un mínimo de* $f(\mathbf{w})$, *es que* $\frac{\partial f(\mathbf{w}^*)}{\partial \mathbf{w}} = 0$. *Esta condición aunada a la convexidad de* f *es una condición suficiente.*

En problemas restringidos, necesitamos definir una función conocida como Lagrangiano, que incorpora información acerca tanto de la función objetivo como de las restricciones. Precisamente el Lagrangiano es definido como la función objetivo más una combinación lineal de las restricciones, donde los coeficientes de la combinación lineal son llamados *multiplicadores de Lagrange.*

En la figura 3.4 se aprecia gráficamente el significado geométrico de los multiplicadores de Lagrange. Supóngase que deseamos encontrar extremos de la función $z = f(x,y)$ sujeta a una restricción dada por $g(x,y) = 0$. Por la figura 3.4 parece razonable esperar que para encontrar, por ejemplo, un máximo de f con restricción, necesitamos solamente encontrar la curva de nivel más alto $f(x,y) = c$ que sea tangente a la gráfica de la ecuación restrictiva $g(x,y) = 0$. Ahora bien, recordemos que, si hacemos uso de las derivadas parciales de ambas funciones, ∇f y ∇g son perpendiculares a las curvas $f(x,y) = c$ y $g(x,y) = 0$ respectivamente. Por lo tanto si $\nabla g \neq 0$ en un punto de tangencia de las curvas entonces ∇f y ∇g son paralelos en dicho punto; esto es, se hallan situados a lo largo de una normal común. Por lo tanto, para cierto escalar no nulo α, en este punto se tendrá

$$\nabla f = \alpha \nabla g \quad o\,bien \quad \begin{aligned} f_x(x,y) &= \alpha g_x(x,y) \\ f_y(x,y) &= \alpha g_y(x,y) \end{aligned}$$

en donde la variable α se llama *multiplicador de Lagrange.*

Definición. Dado un problema de optimización con dominio $\Omega \in \Re^n$.

$$\begin{aligned} Minimizar \quad & f(\mathbf{w}) \quad \mathbf{w} \in \Omega && (3.11) \\ Sujeto \quad & g_i(\mathbf{w}) \leq 0, \quad i = 1, ..., k && (3.12) \\ & h_j(\mathbf{w}) = 0 \quad j = 1, ..., m && (3.13) \end{aligned}$$

se define la función Lagrange generalizada como

$$\begin{aligned} L(w, \alpha, \beta) = \; & f(w) + \textstyle\sum_{i=1}^{k} \alpha_i g_i(w) + \sum_{j=1}^{m} \beta_i h_i(w) && (3.14) \\ & = f(w) + \alpha^t g(w) + \beta^t h(w) && (3.15) \end{aligned}$$

Definición. El problema Lagrangiano dual del primal de la definición anterior es el siguiente:

$$Maximizar \quad \theta(\alpha, \beta) \tag{3.16}$$

$$Sujeto \quad \alpha \geq 0 \tag{3.17}$$

donde $\theta(\alpha, \beta) = inf_{w \in \Omega} L(w, \alpha, \beta)$.
Teorema. (Kuhn-Tucker) Dado un problema de optimización con dominio convexo $\Omega \subseteq \Re^n$

$$Minimizar \quad f(w) \quad w \in \Omega \tag{3.18}$$

$$Sujeto \quad g_i(w) \leq 0, \quad i = 1, ..., k. \tag{3.19}$$

$$h_j(w) = 0, \quad j = 1, ..., m. \tag{3.20}$$

con f *convexa y* g_i, h_j *afinas, son condiciones suficientes y necesarias para que un punto* w^* *sea un óptimo la existencia de* α^*, β^* *las siguientes:*

$$\frac{\partial L(w^*, \alpha^*, \beta^*)}{\partial w} = 0 \tag{3.21}$$

$$\frac{\partial L(w^*, \alpha^*, \beta^*)}{\partial \beta} = 0 \tag{3.22}$$

$$\alpha_i^* g_i(w^*) = 0, \quad i = 1, ..., k \tag{3.23}$$

$$g_i(w^*) \leq 0, \quad i = 1, ..., k \tag{3.24}$$

$$\alpha_i^* \geq 0, \quad i = 1, ..., k \tag{3.25}$$

La tercera relación es conocida como la condición complementaria de Karush -Kuhn-Tucker. Ésta implica que para restricciones activas, se cumple que $\alpha_i^* \geq 0$, *mientras que para restricciones inactivas* $\alpha_i^* = 0$.

3.2.2.2 Dualidad.

El tratamiento de Lagrange para problemas convexos de optimización permite que se introduzca una descripción *dual* de dichos problemas, lo que frecuentemente se convierte en problemas más fáciles de manejar que los

primales, ya que las restricciones de desigualdad (del primal) son difíciles de manejar de manera directa. El problema dual es obtenido introduciendo los multiplicadores de Lagrange, también llamados *variables duales.*

Podemos transformar el problema primal en el dual simplemente igualando a cero las derivadas del primer problema con respecto a las variables primales y sustituyendo las relaciones obtenidas nuevamente en el problema primal, de esta manera suprimimos la dependencia con respecto a las variables primales, lo que corresponde a calcular la función:

$$\theta(\alpha, \beta) = inf_{w \in \Omega} L(w, \alpha, \beta) \tag{3.26}$$

La función resultante contiene sólo variables duales y debe ser maximizada bajo restricciones mucho más simples.

En cuanto a la condición complementaria de Karush-Kuhn-Tucker implica que sólo las restricciones activas tendrán variables duales diferentes de cero, esto es que para ciertos problemas de optimización el número de variables involucradas puede ser significantemente menor que el número completo de casos de entrenamiento. El término *support vector (vectores de soporte)* se refiere precisamente a aquellos datos de entrenamiento para los cuales las variables duales son diferentes de cero.

Cabe hacer mención que *para un programa cuadrático su dual corresponde a otro programa cuadrático pero con restricciones más simples.*

Demostración. (Programa cuadrático).

$$Minimizar \quad \tfrac{1}{2} w^t Q w - k^t w \tag{3.27}$$

$$Sujeto \qquad Xw \leq c \tag{3.28}$$

donde Q es una matriz positiva definida de tamaño $n\,x\,n$, k es un n-vector; c es un m-vector, w es desconocida, y X es una matriz de mxn. Asumiendo que el problema tiene solución, éste puede ser reescrito como:

$$max_{\alpha \geq 0}(min_w(\frac{1}{2} w^t Q w - k^t w + \alpha^t (Xw - c)) \tag{3.29}$$

El mínimo sobre w no tiene restricciones , y es sujeto a $w = Q^{-1}(k - X^t\alpha)$. Resustituyendo esto en el problema original, obtenemos el dual:

$$\begin{aligned} Maximizar \quad & \tfrac{-1}{2}\alpha^t P\alpha - \alpha^t d - \tfrac{1}{2}k^t Qk & (3.30) \\ Sujeto \quad & \alpha \geq 0 & (3.31) \end{aligned}$$

donde $P = XQ^{-1}X^t$ y $d = c - XQ^{-1}k$. Como se observa el problema dual de un problema cuadrático es otro problema cuatrático y la restricción del dual es mucho más simple que el del problema original, como se observa al comparar lass ecuaciones 3.27 y 3.30 o las ecuaciones 3.11 y 3.16.

3.2.3. Uso de kernels.

El uso de kernels ofrece una solución alternativa para proyectar los datos a un espacio de características (*feature space*) de dimensión mucho más alta que la del espacio de entrada, para incrementar el poder computacional de las máquinas lineales de aprendizaje. La ventaja de usar las máquinas en la representación dual se deriva del hecho que en esta representación el número de parámetros ajustables no depende del número de datos de entrenamiento usados [18].

En un problema de optimización los kernels se introducen al reemplazar el producto punto de los datos de entrada por la función kernel escogida, de esta manera se realiza un mapeo implícito no lineal al llamado espacio de características de dimensión mucho más elevada que el espacio de los datos de entrada sin incrementar los parámetros ajustables.

Otra ventaja del uso de kernels sobre el uso de MLP's o cualquier otro tipo de sistemas de aprendizaje es que el problema de escoger la estructura de red apropiada para cada problema es reemplazado por la opción (aparentemente un poco más fácil de solucionar) de escoger el kernel apropiado.

Con el objeto de que una máquina lineal aprenda relaciones no lineales, necesitamos seleccionar un conjunto de características no lineales y reescribir los datos en su nueva representación con respecto a dichas características. Esto es equivalente a aplicar un mapeo no lineal de los datos a un espacio

de características (*feature space*) en el cual la máquina puede ser usada y se simplifique la tarea de clasificación.

$$f(x) = \sum_{i=1}^{n} w_i \Phi_i(x) + b \tag{3.32}$$

donde $\Phi : X \rightarrow F$ es un mapeo no lineal del espacio de entrada hacia algún espacio de características determinado. Esto significa que tendremos que construir las màquinas no lineales en dos pasos: primero determinaremos un mapeo no lineal que transforme los datos hacia algún espacios de características F y posteriormente usaremos una màquina lineal para clasificar dichos datos en F.

El potencial del uso de la representación dual en los problemas de optimización que se resuelven mediante las SVM's también significa que las hipótesis pueden ser expresadas como una combinación lineal de los datos de entrenamiento, de tal manera que la regla de decisión puede ser evaluada usando solo productos punto entre los puntos de prueba y los puntos de entrenamiento:

$$f(x) = \sum_{i=1}^{n} \alpha_i y_i < \Phi(x_i).\Phi(x) > + b \tag{3.33}$$

Si tuviésemos una forma de calcular el producto punto $< \Phi(x_i).\Phi(x) >$ en el espacio de características directamente como una función de los datos de entrada originales sería posible mezclar los dos pasos necesarios para construir una máquina no lineal de aprendizaje. Llamamos a tal método de cálculo directo una función *kernel.*

Definición. Un *kernel* es una función K, tal que para todo $x, z \in X$

$$K(x, z) =< \Phi(x).\Phi(z) > \tag{3.34}$$

donde Φ es un mapeo de X hacia un espacio de características (*feature space)* con producto punto (ver figura 3.5).

Otra importante consecuencia del uso de la representación dual es que la dimensión del espacio de características no necesariamente afecta el tiem-

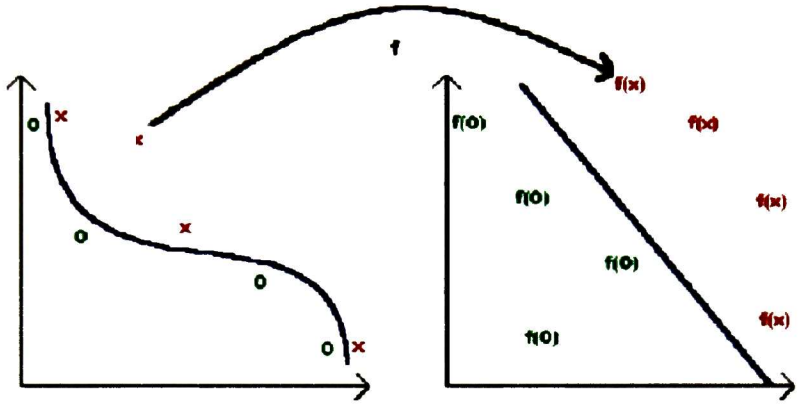


Figura 3.5: El kernel mapea los datos de entrada hacia un espacio de características en donde éstos son linealmente separables.

po de computación. Ya que no se representan los *vectores de características (feature vectors* o vectores mapeados al espacio de características*)* explícitamente, el número de operaciones requeridas para calcular el producto punto evaluando la función kernel no es necesariamente proporcional a la dimensión del espacio de características. Ya que la única información usada con respecto a los datos de entrenamiento es la llamada *Gramm matrix,* que es la matriz que contiene el valor de la función kernel para cada para de datos de entrada (ver figura 3.6)

La *Gramm Matrix* es también conocida como la *Matriz kernel* y tiene interesantes propiedades como el que debe ser una matriz simétrica y positiva definida (todos sus eigenvalores deben ser positivos)

Teorema de Mercer .Cualquier matriz simétrica positiva definida puede ser utilizada como una matriz kernel, ya que ésta representará el producto punto en algún espacio de características. Expansión en eigenvalores de Mercer [2, 4][10]:

$$K(x_1, x_2) = \sum_{i=1}^{n} \lambda_i \Phi_i(x_1) \Phi_i(x_2) \tag{3.35}$$

K=

K(1,1)	K(1,2)	K(1,3)	...	K(1,m)
K(2,1)	K(2,2)	K(2,3)	...	K(2,m)
...	...	...	...	...
K(m,1)	K(m,2)	K(m,3)	...	K(m,m)

Figura 3.6: Matriz kernel o Gramm matrix.

donde $x_1, x_2 \in \Re^n$, n es el número de datos de entrada y λ_i son los eigenvalores no negativos.

3.2.4. Caso de uso de SVM más simple: Clasificación binaria en 2D.

La clasificación binaria es llevada a cabo frecuentemente usando funciones reales $f : X \subseteq \Re^n \rightarrow \Re$ de la siguiente manera: la entrada $\mathbf{x} = (x_1, ..., x_n)^t$ es asignada a la clase positiva si $f(x) \geq 0$, y en otro caso a la clase negativa. Consideramos el caso en el que $f(x)$ es una función lineal de $\mathbf{x} \epsilon$ X, de tal manera que puede ser escrita como:

$$f(x) =< w, x > +b \tag{3.36}$$

$$= \sum_{i=1}^{n} w_i x_i + b \tag{3.37}$$

donde (w,b)$\in \Re^n \, x \, \Re$ son los parámetros que controlan la función objetivo. La regla de decisión está dada por $sgn(f(x))$. La metodología de aprendizaje implica que estos parámetros deben ser aprendidos de los datos de entrenamiento.

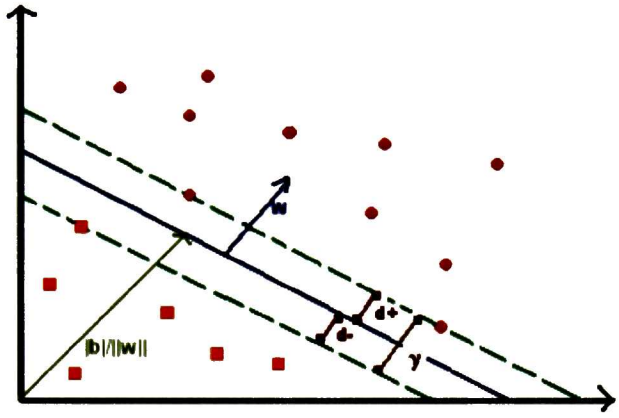


Figura 3.7: Hiperplano separador de las dos clases, <w,x> + b=0.

La interpretación geométrica de esta clase de hipótesis es que el espacio de entrada X es dividido en dos partes por el hiperplano definido por la ecuación <w,x> + b=0 (ver figura 3.7).

Por lo tanto, se desea que la máquina al aprender a clasificar encuentre un hiperplano separador óptimo con base en los patrones de entrada que se le han enseñado. En la figura 3.7 los puntos **x** que caen en el hiperplano encontrado satisfacen $\langle \mathbf{w},\mathbf{x} \rangle + b = 0$.

Donde: **w** es el vector que define una dirección perpendicular al hiperplano, lo que se conoce como normal al hiperplano.

$|b|/||w||$ es la distancia perpendicular del hiperplano al origen.

$||\mathbf{w}||$ es la norma euclideana del vector **w**.

Definamos también d_- y (d_+) como la distancia más corta del hiperplano separador al patrón de entrada negativo (o positivo) más cercano a él, a los que llamamos *vectores de soporte*. Y el *margen*(γ) de un hiperplano separador como $\gamma = d_- + d_+$

Tal que el algoritmo de la Support Vector simplemente buscará el hiper-

plano separador que entregue el máximo margen.

3.2.4.1 Formulación del problema de optimización usando el margen y dimensión VC.

Ahora bien, asumamos que el problema de clasificación es linealmente separable (separable por un plano), y se escogen funciones de la forma:

$$f(x) = (\mathbf{w}.\mathbf{x}) + b \tag{3.38}$$

Como asumimos que los datos de entrenamiento son separables, podemos reescalar el vector $\mathbf{w}$ para que los puntos más cercanos al hiperplano de las dos clases satisfagan la representación canónica de dicho hiperplano, esto es:

$$|(\mathbf{w}.x_i) + b| = 1 \tag{3.39}$$

Ahora consideremos dos ejemplos extraídos de los datos de entrenamiento, $\mathbf{x}_1$ y $\mathbf{x}_2$ de diferentes clases ambos puntos más cercanos al hiperplano, con $(\mathbf{w}.x_1) + b = 1$, por lo que la distancia de x_1 con respecto al origen es de $\frac{|1-b|}{||\mathbf{w}||}$ y $(\mathbf{w}.x_2) + b = -1$, con distancia perpendicular al origen $\frac{|-1-b|}{||\mathbf{w}||}$, entonces el *margen* se toma como la resta de las distancias perpendiculares al origen, esto es $\gamma = \frac{|1-b|}{||\mathbf{w}||} - \frac{|-1-b|}{||\mathbf{w}||} = \frac{2}{||\mathbf{w}||}$. Por lo que si queremos obtener el máximo margen y a la vez minimizar la dimensión VC de la función hiperplano de clase simplemente minimizaremos el inverso del margen esto es, el problema de optimización puede reducirse entonces a:

$$Minimizar : \frac{1}{2}(\mathbf{w}.\mathbf{w}) \tag{3.40}$$

$$Sujeto : \quad y_i[(\mathbf{w}.x_i) + b] \geq 1$$

note que $\gamma^{-1} = \frac{1}{2} < w.w >$

Dado que también la teoría VC de la generalización nos indica que de-

bemos controlar la dimensión VC de las funciones de clase limitando su margen, se procede a resolver el problema maximizando la cantidad $\frac{2}{\Lambda}$, en la que $||w|| \leq \Lambda$, lo que reafirma nuestra formulación anterior del problema de optimización.

De acuerdo a la teoría de Lagrange y Karush Kuhn Tucker introducimos los llamados multiplicadores de Lagrange $\alpha_i \geq 0$, i=1,...,n al problema 3.40, uno para cada restricción y se obtiene el siguiente problema conocido como *Lagrangiano primal:*

$$L(w, b, \alpha) = \frac{1}{2}||w||^2 - \sum_{i=1}^{n} \alpha_i(y_i((w.\Phi(x_i)) + b) - 1) \tag{3.41}$$

La tarea es minimizar 3.40 con respecto a w, b y maximizar con respecto a los multiplicadores de Lagrange α_i. En el punto óptimo, tenemos las siguientes ecuaciones del conocido "punto de silla" (*saddle point*)

$$\frac{\partial L}{\partial b} = 0 \quad y \quad \frac{\partial L}{\partial w} = 0 \tag{3.42}$$

lo que se traduce en

$$\sum_{i=1}^{n} \alpha_i y_i = 0 \quad y \quad w = \sum_{i=1}^{l} \alpha_i y_i \Phi(x_i) \tag{3.43}$$

de la ecuación derecha de 3.43, encontramos que w está contenido en el subespacio expandido por $\Phi(x_i)$. Sustituyendo los resultados de 3.43 en 3.41 y reemplazando $(\Phi(x_i).\Phi(x_j))$ con las funciones kernel $k(x_i, x_j)$, obtenemos el problema cuadrático de optimización:

$$\textit{Maximizar} \quad \sum_{i=1}^{n} \alpha_i - \frac{1}{2} \sum_{i,j=1}^{n} \alpha_i \alpha_j y_i y_j k(x_i x_j) \tag{3.44}$$

$$\textit{Sujeto} \quad \alpha_i \geq 0, \qquad i = 1, ..., n \tag{3.45}$$

$$\sum_{i=1}^{n} \alpha_i y_i = 0 \tag{3.46}$$

Así, resolviendo el problema de optimización dual, se obtienen los coeficientes α_i, para $i = 1, ..., n$, los cuales se necesitan para expresar el w que resuelve el problema 3.40, con la función de decisión no lineal:

$$\begin{aligned} f(x) &= sgn(\sum_{i=1}^{n} y_i \alpha_i (\Phi(x).\Phi(x_i)) + b) \\ &= sgn(\sum_{i=1}^{n} y_i \alpha_i k(x, x_i) + b) \end{aligned} \tag{3.47}$$

Ahora bien, para obtener un "buen" equilibrio entre el riesgo empírico 3.2 y el término de complejidad VC en 3.3 usamos una técnica que fue propuesta en [3] y posteriormente usada en [4] en la que se introducen las llamadas *variables flojas* para suavizar las restricciones respecto al margen se introducen nuevas restricciones:

$$y_i((w.\Phi(x_i)) + b) \geq 1 - \xi_i, \quad \xi_i \geq 0, \quad i = 1, ..., n$$

La solución puede ser encontrada manteniendo el límite superior de la dimensión VC pequeño o minimizando un límite superior para $\sum_{i=1}^{n} \xi_i$ en el riesgo empírico, entonces minimizamos:

$$Minimizar \quad \tfrac{1}{2}||w||^2 + C \sum_{i=1}^{n} \xi_i \tag{3.48}$$

donde la constante de regularización $C > 0$ determina el equilibrio entre el riesgo empírico y el término de complejidad. Añadimos los multiplicadores de Lagrange para obtener el *problema primal:*

$$L(w, b, \xi, \alpha) = \frac{1}{2}||w||^2 + \frac{C}{2} \sum_{i=1}^{n} \xi_i^2 - \sum_{i=1}^{n} \alpha_i (y_i (< w.x_i > + b) - 1 + \xi_i) \tag{3.49}$$

Derivando para cada variable respecto de las cuales se pretende minimizar, esto es w, b y ahora con la nueva restricción también con respecto de ξ_i y maximizar con respecto a cada uno de los multiplicadores de Lagrange α_i, para obtener las ecuaciones de punto de silla:

$$\frac{\partial L}{\partial w} = 0 \quad \frac{\partial L}{\partial b} = 0 \quad \frac{\partial L}{\partial \xi} = 0 \tag{3.50}$$

lo que se traduce en:

$$w = \sum_{i=1}^{n} \alpha_i y_i x_i, \quad \sum_{i=1}^{n} \alpha_i y_i = 0, \quad C\sum_{i=1}^{n} \xi_i - \sum_{i=1}^{n} \alpha_i = 0 \tag{3.51}$$

nuevamente sustituímos los resultados encontrados en el problema primal y obtenemos el problema dual:

$$\begin{aligned} \textit{Maximizar} \quad & \sum_{i=1}^{n} \alpha_i - \tfrac{1}{2}\sum_{i,j=1}^{n} \alpha_i\alpha_j y_i y_j K(x_i x_j) - \tfrac{1}{2C}\sum_{i,j=1}^{n} \alpha_i\alpha_j && (3.52) \\ \textit{Sujeto} \quad & 0 \leq \alpha_i \leq C, && i = 1, \ldots, n \quad (3.53) \\ & \sum_{i=1}^{n} \alpha_i y_i = 0 && (3.54) \end{aligned}$$

3.2.4.2 Propiedad de esparcimiento de las SVM.

La mayoría de los métodos de optimización están basados en condiciones de optimización de segundo orden, las llamadas condiciones Karush-Kuhn-Tucker, las cuales establecen condiciones necesarias y en algunos casos suficientes para que un conjunto de variables sean las soluciones óptimas para problemas dados. Estas condiciones son particularmente simples para el problema de optimización dual:

$$\begin{aligned} \alpha_i = 0 \quad &\Rightarrow \quad y_i f(x_i) \geq 1 \quad y \quad \xi_i = 0 \\ 0 < \alpha_i < C \quad &\Rightarrow \quad y_i f(x_i) = 1 \quad y \quad \xi_i = 0 \\ \alpha_i = C \quad &\Rightarrow \quad y_i f(x_i) \leq 1 \quad y \quad \xi_i \geq 0 \end{aligned} \tag{3.55}$$

Estas revelan una de las más importantes propiedades de las SVMs: la solución está esparcida en α, por ejemplo, muchos patrones están fuera del área del margen y las α_i's correspondientes son cero. Especificamente las condiciones KKT muestran que solo aquellas $\alpha_i's$ conectadas a patrones de entrenamiento $x_i's$ que estén en el margen ($0 < \alpha_i < C$ y $y_i f(x_i) = 1$) o dentro del área del margen ($\alpha_i = C$ y $y_i f(x_i) < 1$) son diferentes de cero. Si no fuera por la propiedad de esparcimiento de las SVM el aprendizaje sería muy difícil para conjuntos de datos grandes.

Sabemos que la solución al problema de optimización será única y global ya que el problema de optimización es cuadrático y si derivamos dos veces

el problema dual con respecto a los multiplicadores de Lagrange α, el Hessesiano correspondiente se reduce a $H_{ij} = y_i y_j K(x_i x_j)$, lo cual nos conduce a asegurar que el Hessesiano es positivo definido, ya que la Gramm matriz $K(x_i x_j)$ lo debe ser por el teorema de Mercer (ver subsección 3.2.3).

3.2.4.3 Resultados de entrenamiento usando diferentes kernels.

Las figuras 3.8, 3.9 y 3.10 son resultados de entrenamiento de SVM's binarias resolviendo el problema de optimización en el que se incluyen las condiciones Karush-Kuhn-Tucker, con los kernels más comunes[3] . Para estos experimentos de clasificación se hizo uso del programa que se implementó (en lenguaje C++ para plataforma LINUX) durante el desarrollo de esta tesis. Los resultados de clasificación de las figuras 3.8 y 3.9 se pueden interpretar de la siguiente manera: todos aquellos vectores con fondo blanco pertenecen a la clase 1 y los datos de entrenamiento de dicha clase se representan como círculos rojos, mientras que todos los vectores cuyo fondo es rojo pertenecen a la clase 2 y los datos de entrenamiento de dicha clase son círculos blancos.

La interpretación de la figura 3.10 es que todos aquellos vectores con fondo verde pertenecen a la clase 1 y los vectores de entrenamiento de la clase son círculos de perímetro negro. Los puntos cuyo fondo es azul pertenecen a la clase 2 y los vectores de entrenamiento de dicha clase son círculos de perímetro rojo.

Los clasificadores de las figuras 3.8, 3.9 y 3.10 son binarios, pero éstos pueden ser fácilmente combinados para resolver problemas multiclase. Un simple, pero efectivo enfoque entrena las SVM's es conocido como *uno contra el resto*[4], en él entrenamos una clase positiva contra el resto negativas para las N clases, con resultados como el visto en la figuras 3.12 y 3.11 , utilizando el kernel gaussiano durante el entrenamiento. Los colores de las figuras se interpretan de la siguiente manera: los vectores de la clase 1 tienen un fondo verde y vectores de entrenamiento de color azul. La clase 2 está representada por un color azul de fondo y vectores de entrenamiento son verdes. El color

[3]Diversos algoritmos de entrenamiento de SVM pueden encontrarse en [7, 6, 8, 5]

[4]El lector encontrará varios enfoques más para resolver problemas multiclase con SVM en las referencias [20, 15].

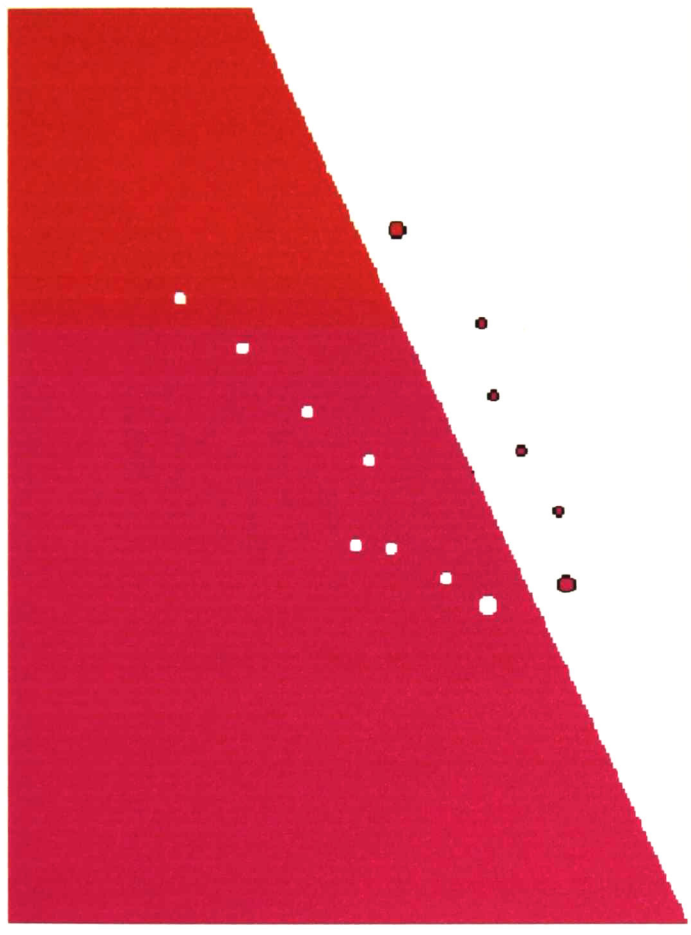


Figura 3.8: Resultado de clasificación binaria usando SVM con kernel identidad (<x.y>), los vectores de soporte de cada clase aparecen como círculos de radio mayor con respecto a los otros. Resultado obtenido en 1000 iteraciones.

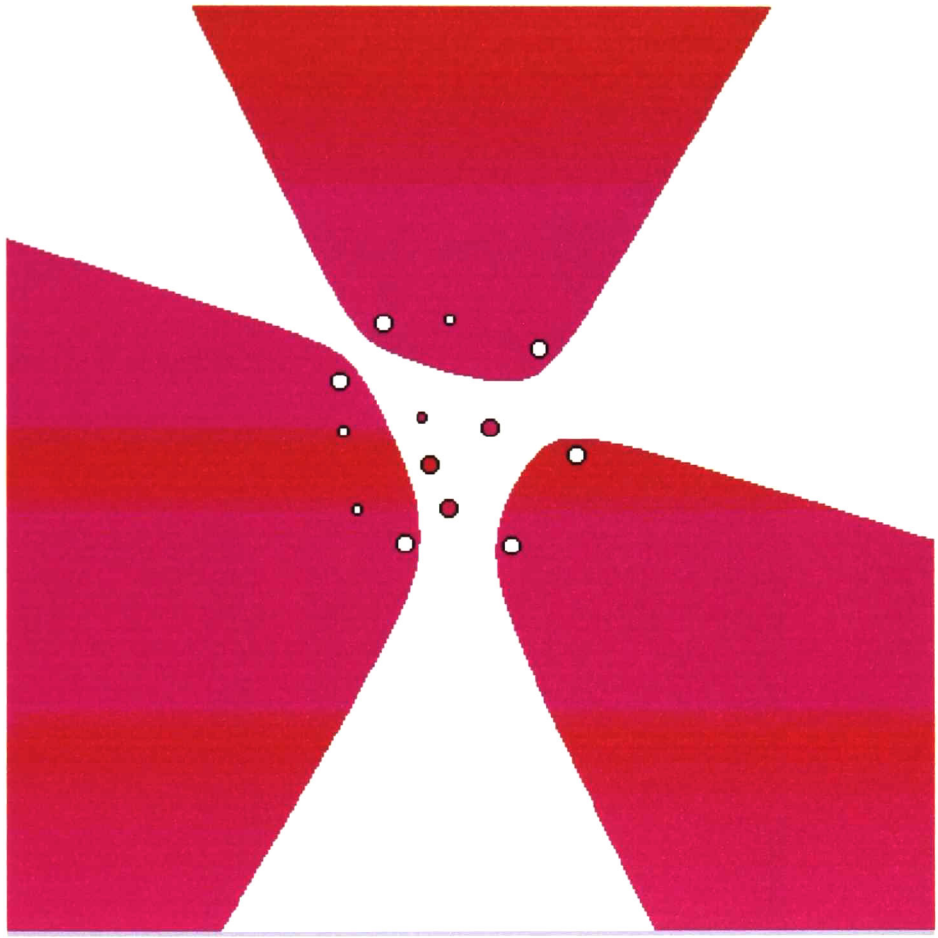


Figura 3.9: Resultado de clasificación binaria usando SVM con kernel polinomial ($(< x.y > +1)^d$), grado 5 ($d = 5$), los vectores de soporte de cada clase aparecen como círculos de radio mayor con respecto a los otros. Resultado obtenido en 1000 iteraciones.

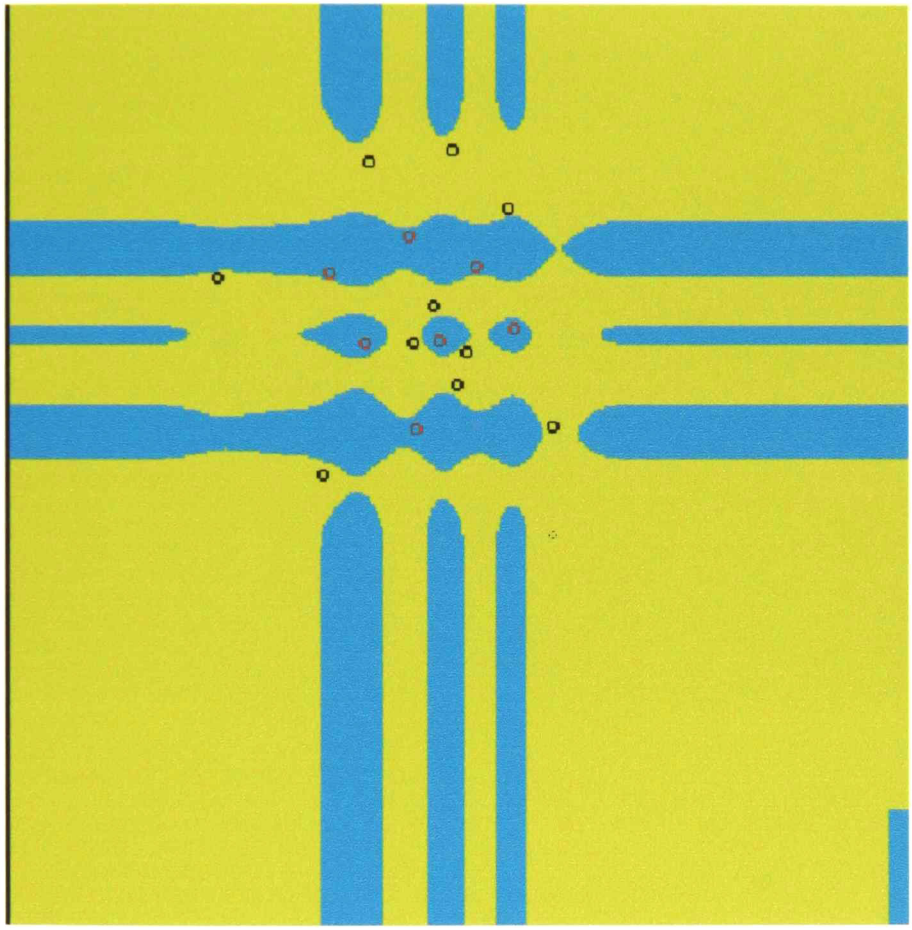


Figura 3.10: Resultado de clasificación binaria usando SVM con kernel gaussiano ($e^{\frac{\|x-y\|^2}{2\sigma}}$), los vectores de soporte de cada clase aparecen como círculos de radio mayor con respecto a los otros. Resultado obtenido en 1000 iteraciones.

de fondo de la clase 3 es rojo (o naranja) y los vectores de entrenamiento son negros; mientras que la clase 4 tiene fondo blanco y vectores de entrenamiento rojos.

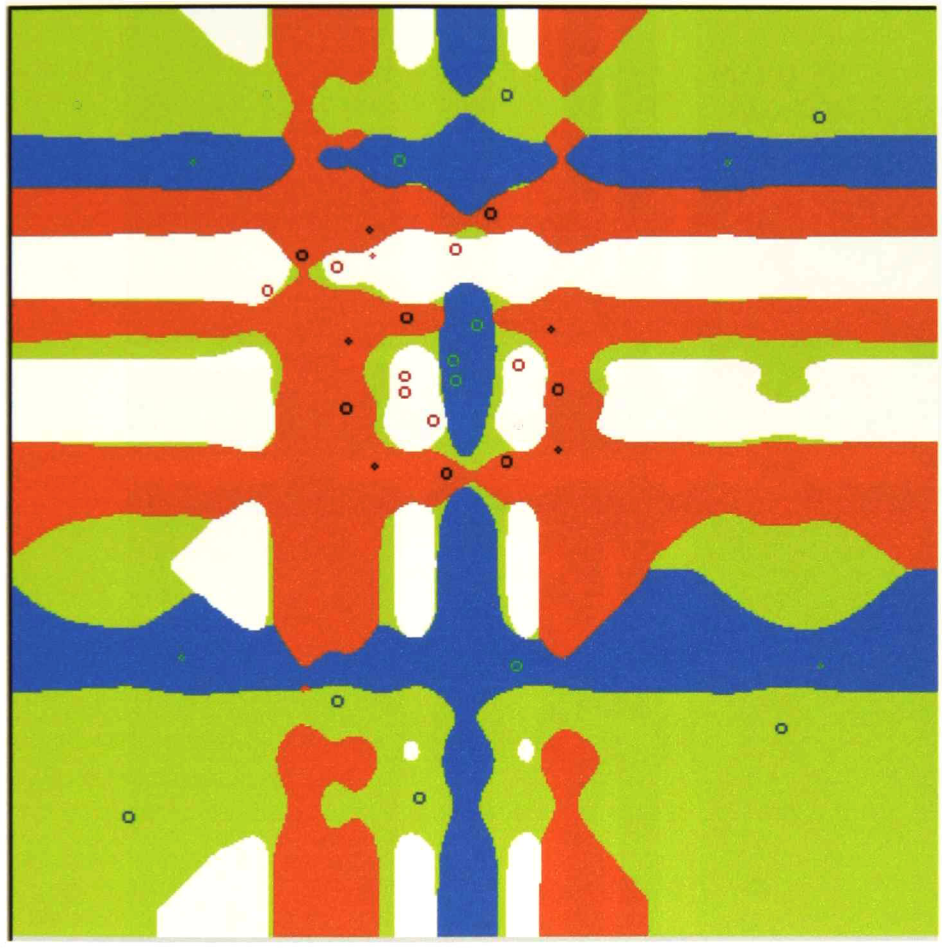


Figura 3.11: Resultado de clasificación multiclase (4 clases) usando SVM con kernel gaussiano, los vectores de soporte de cada clase aparecen como círculos de radio mayor con respecto a los otros. Resultados obtenidos en la iteración número 500.

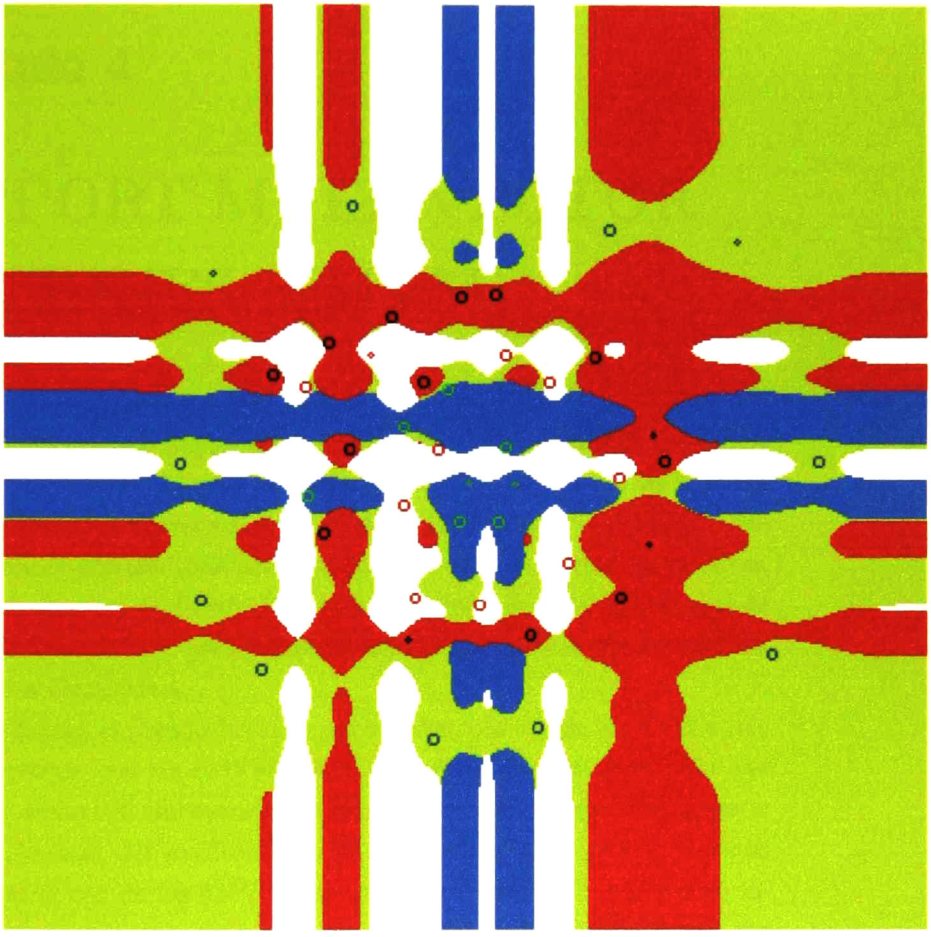


Figura 3.12: Resultado de clasificación multiclase (4 clases) usando SVM con kernel gaussiano, los vectores de soporte de cada clase aparecen como círculos de radio mayor con respecto a los otros. Resultados obtenidos en la iteración 1000.

Capítulo 4

SUPPORT MULTIVECTOR MACHINES (SMVM)

Hemos denominado Support Multivector Machines (SMVM) a aquellos sistemas neurocomputacionales de aprendizaje que se derivan de las Support Vector Machines. La diferencia entre estos sistemas es que las SMVM trabajan con datos de entrada codificados como multivectores, lo que enriquece en gran medida el poder de descripción geométrico de objetos del mundo real de estos sistemas, aunado a que los kernels con los que trabaja son formulados en el Álgebra Geométrica.

Ya que hemos explicado las nociones teóricas básicas de las SVM's (las cuales comparten con las SMVM), en este capítulo explicaremos bajo qué filosofía de desarrollo elaboramos los kernels geométricos que distinguen a nuestras máquinas con multivectores de soporte de las originales, además de presentar el uso de las SMVM con un preprocesamiento geométrico de los datos de entrada que nos permitirá una importante compresión en los ejemplos de entrenamiento dados a nuestro sistema para su aprendizaje.

4.1. Elaboración de kernels.

El uso de funciones kernel añade poder de clasificación no lineal a máquinas lineales. Si deseamos hacer uso de ese poder y elaborar nuestros propios

kernels primero determinaremos qué propiedades deberá tener una función K(x,z) para ser considerada un kernel en algún espacio de características.

Claramente la función deberá ser simétrica:

$$K(x,z) =< \Phi(x).\Phi(z) >=< \Phi(z).\Phi(x) >= K(z,x) \tag{4.1}$$

Condición de Mercer. Sea X un espacio de entrada finito con una función simétrica $K(x,z)$ en X. Entonces $K(x,z)$ es una función kernel si y sólo si la matriz

$$K = (K(x_i,x_j))_{i,j=1,\ldots,n}^{n} \tag{4.2}$$

es positiva semi-definida (no tiene eigenvalores negativos).

Esta condición es la clave para verificar que cualquier función simétrica es un kernel. La siguiente proposición muestra que los kernels satisfacen cierto número de propiedades de cerradura y nos permiten crear kernels más sofisticados partiendo de unos simples.

Proposición. Sean K_1 y K_2 kernels sobre $X \star X$, en el que $X \subseteq \Re^n$, $a \in \Re^+$, y $f(.)$ una función real en X :

$$\Phi : X \to \Re^m$$

con K_3 un kernel sobre $\Re^m \, x \, \Re^m$ y K una matriz simétrica positiva semi definida de tamaño $n \, x \, n$. 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. $K(x,z) = K_3(\Phi(x),\Phi(z))$.
6. $K(x,z) = x'Bz$

4.1.1. Elaboración de kernels a partir de características

Una filosofía de elaboración de kernels consiste en comenzar a trabajar desde las características, es decir, comenzar con los vectores de entrada ya

mapeados al espacio de características, en el que tenemos la certeza que se encontrarán muchas relaciones lineales si partimos de determinado espacio de entrada y así obtenemos el kernel respectivo trabajando con su producto punto. En este caso no es necesario comprobar características que debe cumplir todo kernel, ya que se tiene la seguridad que la función encontrada representa el producto punto de los datos de entrada en el espacio de características conocido. Los mapeos se hacen explícitos en este tipo de kernel de acuerdo a la igualdad que aparece en la ecuación 3.34, respetando la condición de que el espacio de características al que nos conducen dichos mapeos es de dimensión tratable computacionalmente, condición establecida por V.Vapnik en [3]. Ejemplos claros producto de esta filosofía son los kernels polinomiales.

Explotando la potencialidad del producto Clifford para obtener una dimensionalidad más alta que la que se obtiene usando los kernels polinomiales obtenemos los siguientes kernels:

2D $\Rightarrow$ 4D.

Dado el espacio de entrada $\Re^2$ con bases generadoras {[1,0]',[0,1]'}, los mapeos encontrados conducirán a $G_{2,0,0} = gen\{1, e_1, e_2, e_{12} = I\}$ con dimensión 4.

Sea $x_i, x_j \in \Re^2$, donde $x_i = [x, y]'$ y $x_j = [z, w]'$

$K(x_i, x_j) =< \Phi(x_i), \Phi(x_j) >$ Donde:

$\Phi(x_i) = \Phi_1(x_i) * \Phi_2(x_i)$,en el que Φ_1, Φ_2 conducen de $\Re^2$ a $G_{2,0,0}$. Donde * denota la operación del producto Clifford.

$\Phi_1(x_i) = xe_1 + yI.$

$\Phi_2(x_i) = xe_1 - ye_2.$

Por lo tanto:

$$\begin{aligned}\Phi(x_i) &= \Phi_1(x_i) * \Phi_2(x_i)\\ &= (xe_1 + yI) * (xe_1 - ye_2)\\ &= x^2e_1^2 - xye_{12} + xyIe_1 - y^2Ie_2\\ &= x^2 - xye_{12} - xye_2 - y^2e_1\\ &= \Phi([x, y])' = [x^2, -xy, -xy, -y^2]'\end{aligned}$$

Y el kernel resultante, cuya clasificación obtenida se ilustra en 4.1 y 4.2. Las imágenes de todos los resultados de clasificación de estos kernels

geométricos se pueden interpretar de la siguiente forma: los vectores que resultaron pertenecientes a la clase -1 se ilustran con fondo azul agua, los vectores del conjunto de entrenamiento de ésta clase son cuadrados rojos; mientras que los de clase 1 se colorean con fondo en tono verde olivo y los datos de entrenamiento son cuadrados negros. Los resultados de este kernel para problemas multiclase en la figura 4.3:

$$K(x_i, x_j) =< \Phi(x_i), \Phi(x_j) >=< \Phi([x,y]'), \Phi([z,w]') >$$
$$=< [x^2, xy, xy, -y^2]'.[z^2, zw, zw, -w^2]' >$$

La interpretación para ambas figuras de resultados multiclase geométricos (figuras 4.3 y 4.6) es la siguiente: el color de fondo de los vectores de la clase 1 es blanco, los vectores de entrenamiento de dicha clase son círculos naranjas. La clase 2 tiene fondo azul y vectores de entrenamiento verdes. El color de fondo de la clase 3 es naranja y los puntos de entrenamiento azules, mientras que el color de fondo de la clase 4 es el verde y de los vectores de entrenamiento de dicha clase el blanco.

2D $\Rightarrow$ 8D.

Dado el espacio de entrada $\Re^2$ con bases generadoras {[1,0]',[0,1]'}, los mapeos encontrados conducirán a $G_{3,0,0} = gen\{1, e_1, e_2, e_3, e_{12}, e_{23}, e_{31}, e_{123} = I\}$ con dimensión 8.

Sea $x_i, x_j \in \Re^2$, donde $x_i = [x,y]'$ y $x_j = [z,w]$'

$K(x_i, x_j) =< \Phi(x_i), \Phi(x_j) >$ Donde:

$\Phi(x_i) = \Phi_1(x_i) * \Phi_2(x_i)$, en el que Φ_1, Φ_2 conducen de $\Re^2$ a $G_{3,0,0}$. Donde * representa la operación del producto Clifford.

$\Phi_1(x_i) = (xe_1 + yI) * (xe_1 - ye_2)$.

$\Phi_2(x_i) = xe_1 - ye_{23}$.

Por lo tanto:

$$\Phi(x_i) = \Phi_1(x_i) * \Phi_2(x_i)$$
$$= [(xe_1 + yI) * (xe_1 - ye_2)] * (xe_1 - ye_{23}.)$$
$$= (x^2 - xye_{12} + xye_{23} + y^2e_{13}) * (xe_1 - ye_{23})$$
$$= x^3e_1 - x^2ye_{23} + x^2ye_2 + xy^2e_{13} + x^2ye_{123} + xy^2 - xy^2e_3 + y^3e_{12}$$
$$= \Phi([x,y])' = [x^3, -x^2y, x^2y, xy^2, x^2y, xy^2, -xy^2, y^3]'$$

Y el kernel resultante (cuya clasificación obtenida se ilustra en 4.4 y 4.5Los resultados de este kernel para problemas multiclase en la figura 4.6 :

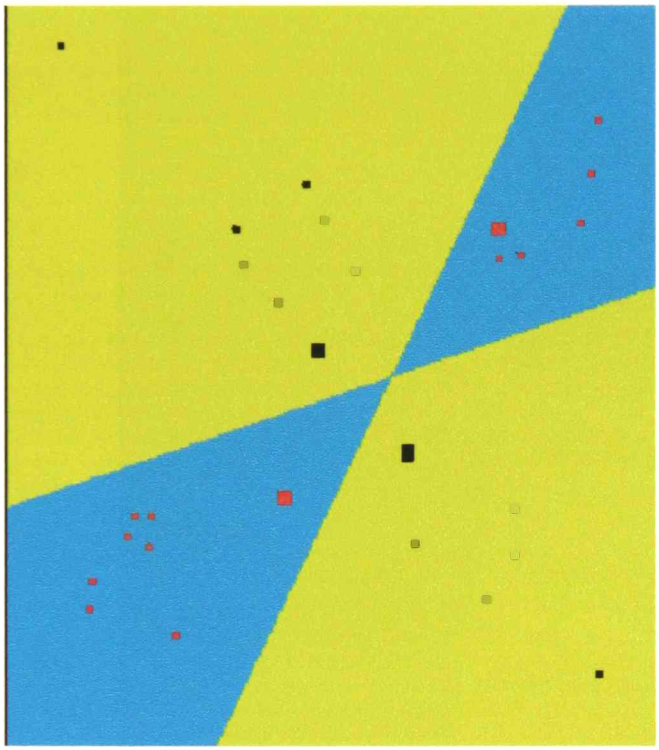


Figura 4.1: Resultado de clasificación usando kernel geométrico 2D ->4D. Los vectores de soporte aparecen como cuadrados cuya longitud de lado es mayor con respecto a los demás. Resultados obtenidos en la iteración número 1000.

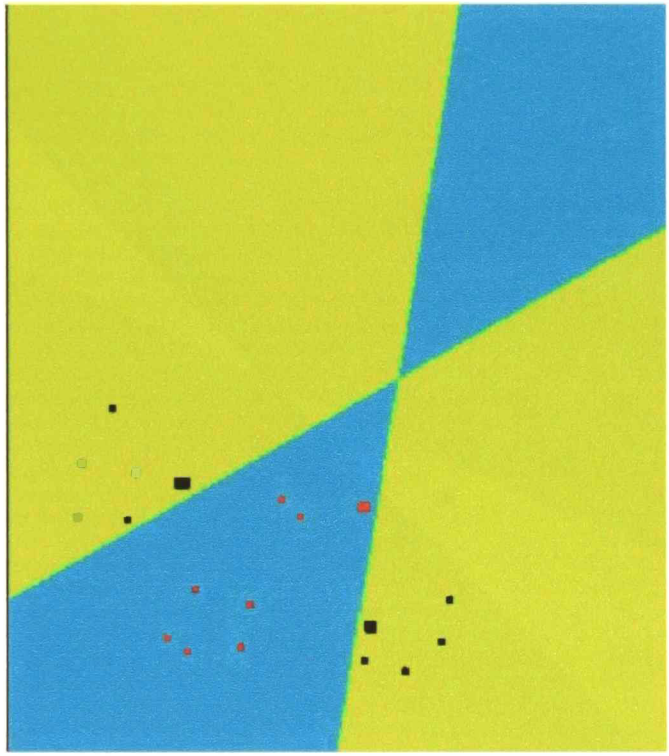


Figura 4.2: Resultados de clasificación usando kernel geométrico 2D ->4D. Los vectores de soporte aparecen como cuadrados cuya longitud de lado es mayor con respecto a los demás. Resultados obtenidos en la iteración número 1000.

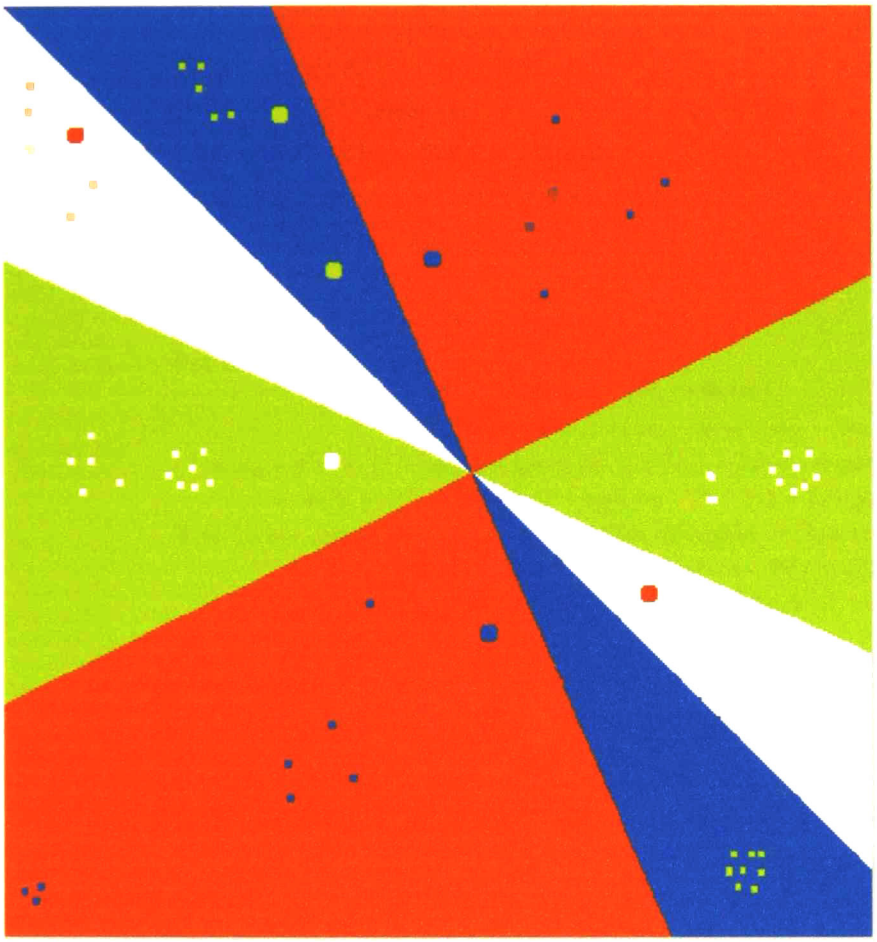


Figura 4.3: Resultados de clasificación para problema multiclase usando kernel geométrico 2D ->4D. Los vectores de soporte aparecen como círculos cuya circunferencia es mayor con respecto a los demás. Resultados obtenidos en la iteración número 1000.

$$K(x_i, x_j) =< \Phi(x_i), \Phi(x_j) >=< \Phi([x,y]'), \Phi([z,w]') >$$
$$=< [x^3, -x^2y, x^2y, xy^2, x^2y, xy^2, -xy^2, y^3]'.[z^3, -z^2w, z^2w, zw^2 z^2w, zw^2, -zw^2, z^3]' >$$

3D $\Rightarrow$ 8D.

Dado el espacio de entrada $\Re^3$ con bases generadoras {[1,0,0]',[0,1,0]',[0,0,1]'}, los mapeos encontrados conducirán a $G_{3,0,0} = gen\{1, e_1, e_2, e_3, e_{12}, e_{23}, e_{31}, e_{123} = I\}$ con dimensión 8.

Sea $x_i, x_j \in \Re^3$, donde $x_i = [x,y,z]'$ y $x_j = [p,q,r]'$

$K(x_i, x_j) =< \Phi(x_i), \Phi(x_j) >$. Donde:

$\Phi(x_i) = \Phi_1(x_i) * \Phi_2(x_i)$, en el que Φ_1, Φ_2 conducen de $\Re^2$ a $G_{3,0,0}$. Donde * representa la operación del producto Clifford.

$\Phi_1(x_i) = xe_1 + ye_2 + ze_3$.

$\Phi_2(x_i) = xe_2 - ye_{12} - e_{23} - ze_{123}$

Por lo tanto:

$$\Phi(x_i) = \Phi_1(x_i) * \Phi_2(x_i)$$
$$= (xe_1 + ye_2 + ze_3) * (xe_2 - ye_{12} - e_{23} - ze_{123})$$
$$= xy + y^2e_1 + (z - xy)e_2 - ye_3 + (x^2 - z^2)e_{12} - 2xze_{23} + yze_{13} + (-x - yz)e_{123}$$
$$= \Phi([x,y,z])' = [xy, y^2, (z - xy), -y, (x^2 - z^2), -2xz, yz, (-x - yz)]'$$

Y el kernel resultante (cuya clasificación obtenida se ilustra en 4.7):

$$K(x_i, x_j) =< \Phi(x_i), \Phi(x_j) >=< \Phi([x,y,z]'), \Phi([p,q,r]') >$$
$$=< [xy, y^2, (z - xy), -y, (x^2 - z^2), -2xz, yz, (-x - yz)]'.[pq, q^2, (r - pq), -q, (p^2 - r^2), -2pr, qr, (-p - qr$$

4.1.2. Comparación de características de generalización de kernels geométricos contra gaussianos.

En la subsección 3.2.3 se menciona como una ventaja de las SVM's sobre las MLP's que el problema de escoger la estructura de red apropiada para cada aplicación es reemplazado por la opción (aparentemente un poco más fácil de solucionar) de escoger el kernel apropiado. Esto es, se sabe que para cada problema en específico la arquitectura de una MLP debe ser diseñada antes que implementada; ahora bien, el símil de este paso de diseño al usar SVM's se convierte en escoger el kernel que nos otorgue las características de generalización del aprendizaje que deseamos para nuestro problema.

Mientras que el error empírico 3.2 puede ser minimizado usando cualquiera de los kernels existentes, el error en la etapa de prueba (riesgo esperado o error de generalización 3.1), al clasificar inadecuadamente los datos nunca

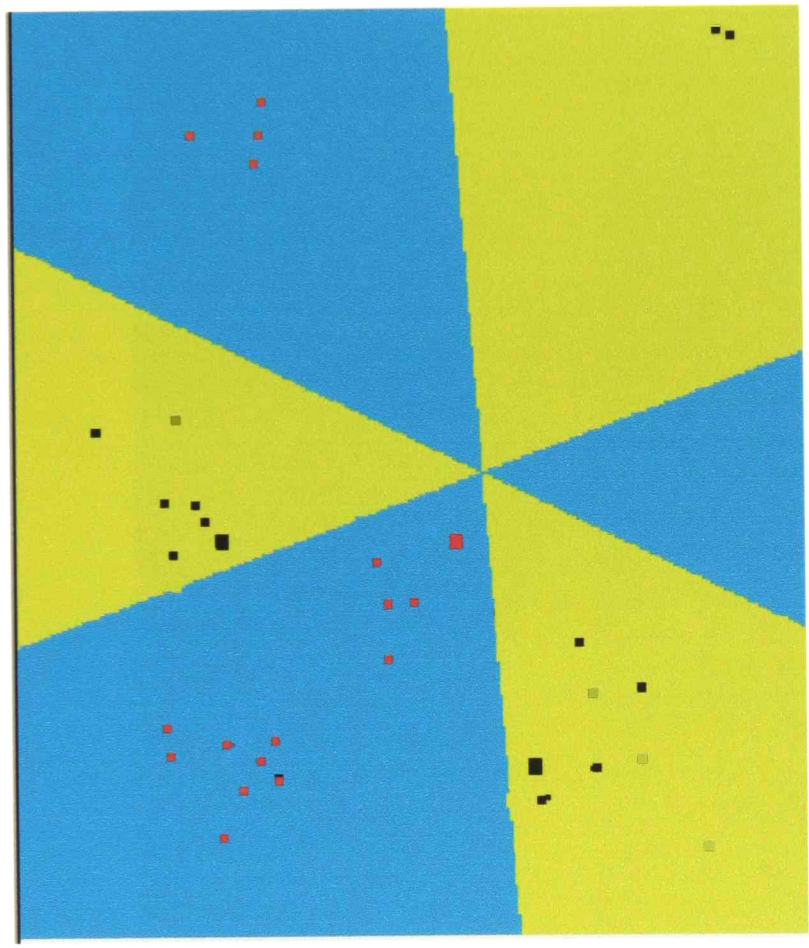


Figura 4.4: Resultado del uso del kernel geométrico 2D ->8D, los vectores de soporte aparecen como cuadrados cuya longitud de lado es mayor con respecto a los demás. Resultados obtenidos en la iteración número 1000.

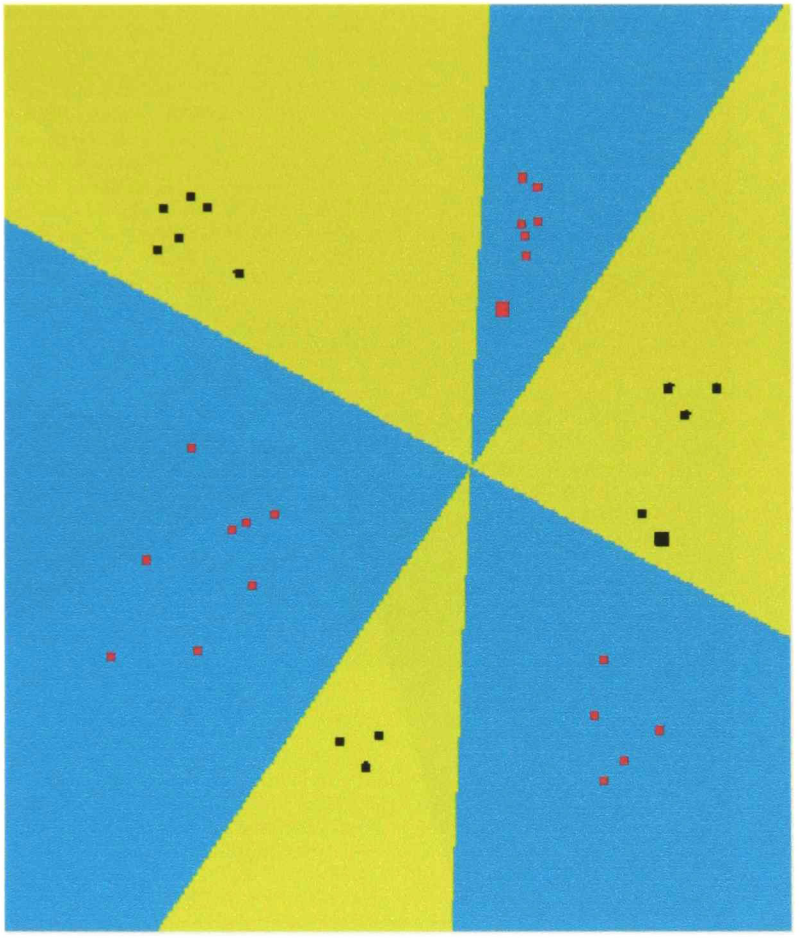


Figura 4.5: Resultado del uso del kernel geométrico 2D ->8D, los vectores de soporte aparecen como cuadrados cuya longitud de lado es mayor con respecto a los demás. Resultados obtenidos en la iteración número 1000.

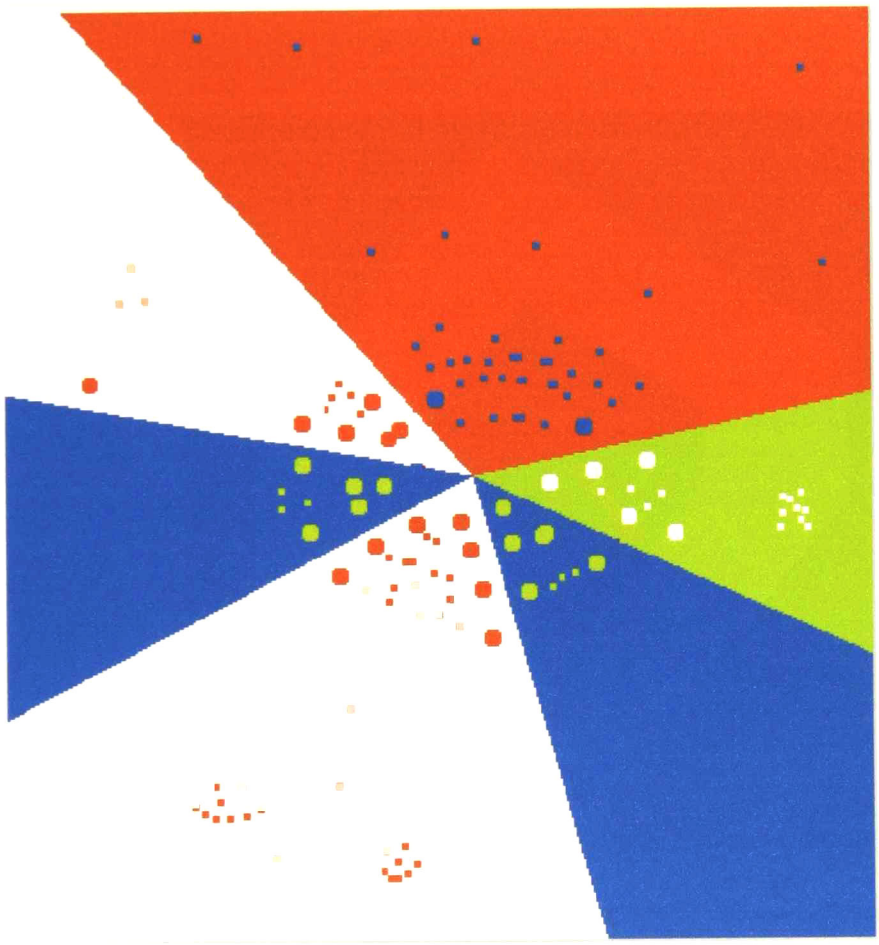


Figura 4.6: Resultados de clasificación para problema multiclase usando kernel geométrico 2D ->8D. Los vectores de soporte aparecen como círculos cuya circunferencia es mayor con respecto a los demás. Resultados obtenidos en la iteración número 1000.

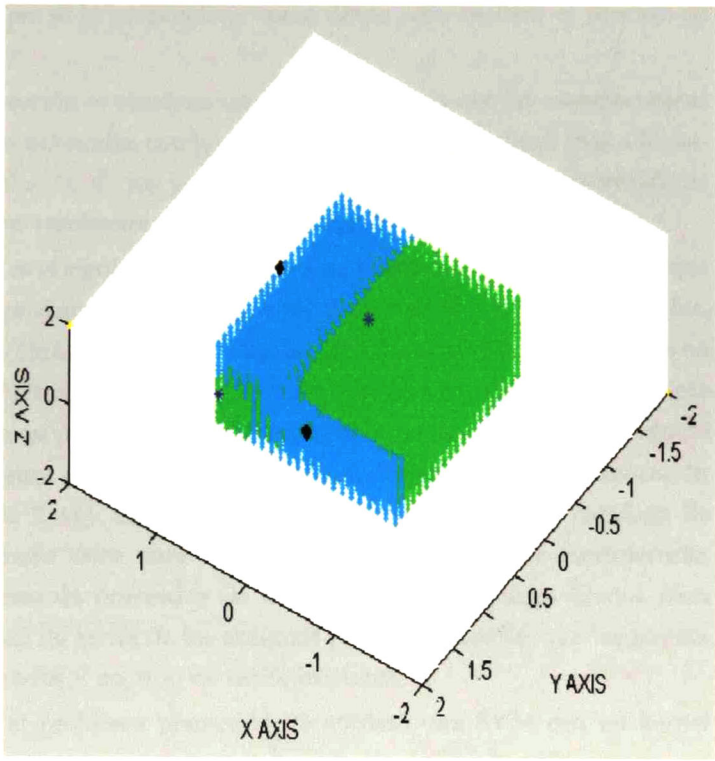


Figura 4.7: Resultado de clasificación un conjunto de datos de entrenamiento representando el problema conocido como "or exclusiva" en 3D, usando el kernel 3D ->8D. Resultados obtenidos en la iteración número 1000.

antes vistos en la etapa de entrenamiento, puede aumentar o disminuir dependiendo del kernel escogido. Ahora bien, se debe tomar en cuenta que las características del aprendizaje pueden variar dependiendo de los datos que se le presenten a la máquina como parte del *conjunto de datos de entrenamiento* y los datos que posteriormente se consideren dentro del *conjunto de datos de prueba* ya que la máquina obviamente aprenderá a generalizar según la información que se le proporcione como datos para realizar el proceso de entrenamiento.

En esta subsección se plantean una aplicación en la que las características de generalización obtenidas con los kernels geométricos resultan más adecuadas para la resolución de los problemas comparadas con las características de la clasificación resultante del uso de un kernel gaussiano[1].

El problema es el siguiente: Una fábrica de llaves (de cerradura) desea que sus 24 tipos de productos sean clasificados dentro de cuatro tipos de tarifas, de acuerdo a dos características principales, el número de llaves contenidas en 60 kilos de productos y los gramos de metal utilizados para fabricar un lote (el número de llaves por lote cambia según el tipo de llave) de dicho producto (tomando en cuenta el metal excedente que se desperdicia en la elaboración de cada clase de llave). En esta empresa planean ampliar su catálogo de productos al añadir doce nuevos tipos de llaves (los que se considerarán datos del conjunto de prueba) y de acuerdo a los estándares fijados para deteminar la clase de tarifa de los antiguos productos desean que las nuevas llaves sean asignados a un tipo de tarifa existente.

De acuerdo al problema planteado, se entrenó una SVM con un kernel gaussiano y, con los mismos datos de entrenamiento, se hizo lo mismo con una SMVM usando el kernel geométrico 2D $\Rightarrow$ 8D.

Cada llave queda codificada como un vector de dos elementos. El conjunto de datos de entrenamiento, antes de someterse a la normalización[2], se puede observar en la tabla 4.1, así como la clase que se desea que la máquina

[1]Se hizo uso del kernel gaussiano en la comparación, ya que éste es conocido por ser un aproximador universal y por tanto,en teoría, todo problema de clasificación se puede resolver con él.

[2]Sólo se dividio por mil el primer dato del vector de descripción de la llave.

aprenda para cada vector de entrada.

En la tabla 4.2 se observa el conjunto de datos de prueba y para cada vector la tarifa en la cual se desea que se clasifique, así como la clase obtenida al introducir a la red el dato y evaluar la función aprendida con un kernel gaussiano y en la columna siguiente la clase obtenida con el kernel geométrico 2D $\Rightarrow$ 8D. Las filas de esta tabla que aparecen en itálica representan los errores de prueba o generalización que se presentaron al usar el kernel gaussiano y que fueron clasificadas correctamente con el kernel geométrico.

Los datos (tanto de entrenamiento como de prueba) se normalizaron de acuerdo a las fórmulas

$$x_n = \begin{cases} -(-\frac{x}{20} + 10) & para\, x \leq 200\, y\, x \geq 0 \\ \frac{x}{20} - 10 & para\, x > 200\, y\, x \leq 400 \end{cases}$$

y

$$y_n = \begin{cases} -\frac{y}{20} + 10 & para\, y \leq 200\, y\, y \geq 0 \\ -(\frac{y}{20} - 10) & para\, y > 200\, y\, y \leq 400 \end{cases}$$

donde x e y son el primer dato del vector de entrada y el segundo dato del vector de entrada respectivamente , x_n y y_n son el primer y segundo dato del vector de entrada normalizados. El proceso de normalización se hizo de esta manera para hacer coincidir los datos de entrada con el plano cartesiano con los ejes $x, y \in [10, 10]$.

En las figuras 4.8 y 4.9 se muestra gráficamente los resultados de la clasificación para este problema. Los datos del conjunto de entrenamiento se ilustran como círculos y cuadrados, los vectores de soporte están representados como círculos cuya circunferencia es mayor con respecto a los demás puntos. Los datos del conjunto de prueba se observan como con círculos cruzados por una estrella, el color de estas figuras representativas de dichos datos de prueba, corresponde a la clase de tarifa a la que se deseaba que pertenecieran. Los colores de la clasificación se interpretan como sigue: la clase 1 tiene un fondo naranja y vectores de entrenamiento azules. El color

de fondo de la clase 2 es el blanco y los vectores de entrenamiento son naranjas. La clase 3 se ilustra con color de fondo azul y vectores de entrenamiento verdes, mientras que la clase 4 está representada con color de fondo verde y vectores de entrenamiento blancos.

Como se puede observar las características kernel gaussiano hacen que éste otorgue una clasificación cuya generalización no es la que se desea para la resolución de este problema. Por otro lado, los resultados obtenidos con la aplicación del kernel geométrico 2D $\Rightarrow$ 8D permiten, que *en este caso,* (como puede haber muchos otros) el error en la clasificación de los datos de prueba se reduzca a cero.

4.1.3. Support Multivector Machines (SMVM) con entradas codificadas como multivectores.

Otra de las mencionadas ventajas de las SMVM es aquella que se obtiene al hacer uso de la codificación de los datos de entrenamiento en forma de multivectores. En la aplicación desarrollada en esta tesis se considera que el espacio en el que se pretende realizar la clasificación es $\Re^3$ y con la finalidad de mostrar el poder de descripción geométrico del álgebra conformal con el objeto de obtener una compresión considerable de los datos de entrada para el entrenamiento de la SMVM se codifican dichos datos haciendo uso de un preprocesamiento resultado de la aplicación de una red neuronal conocida como *Spherical k-means* [24, 25] a la que se le proporcionaron en su entrenamiento, cada uno de los datos del conjunto de entrada y como resultado de su aprendizaje esta red nos entregó los 5-vectores (ver tabla 2.1) que representan a las esferas que engloban los vectores de entrada por regiones, por lo que las entradas de entrenamiento de la SMVM se convirtieron en las esferas conformales (las cuales representan el subconjunto de vectores de entrada que cada una de ellas contiene). Los resultados obtenidos en todos los experimentos de clasificación fueron satisfactorios, ya que la SMVM clasifica correctamente las esferas de entrada originales durante su aprendizaje y en la fase de prueba se le proporcionaron tanto esferas como puntos contenidos en las esferas de entrenamiento, los cuales clasificó correctamente.

Vector de entrada (Descripción de llave)	Clase deseada (tarifa)
[201,184]	1
[201,172]	1
[209,180]	1
[215,168]	1
[223,144]	1
[227,146]	1
[167,214]	2
[178,212]	2
[175,221]	2
[156,256]	2
[87,238]	2
[90,251]	2
[210,219]	3
[209,236]	3
[220,237]	3
[225,225]	3
[232,237]	3
[219,248]	3
[163,159]	4
[169,152]	4
[171,163]	4
[161,147]	4
[165,143]	4
[153,140]	4

Cuadro 4.1: Conjunto de datos de entrenamiento para problema cuyo espacio de entrada es de dimensión 2.

Figura 4.8: Resultados de clasificación usando una SVM con kernel gaussiano para la apliación del problema de la fábrica de llaves. Iteración 1000.

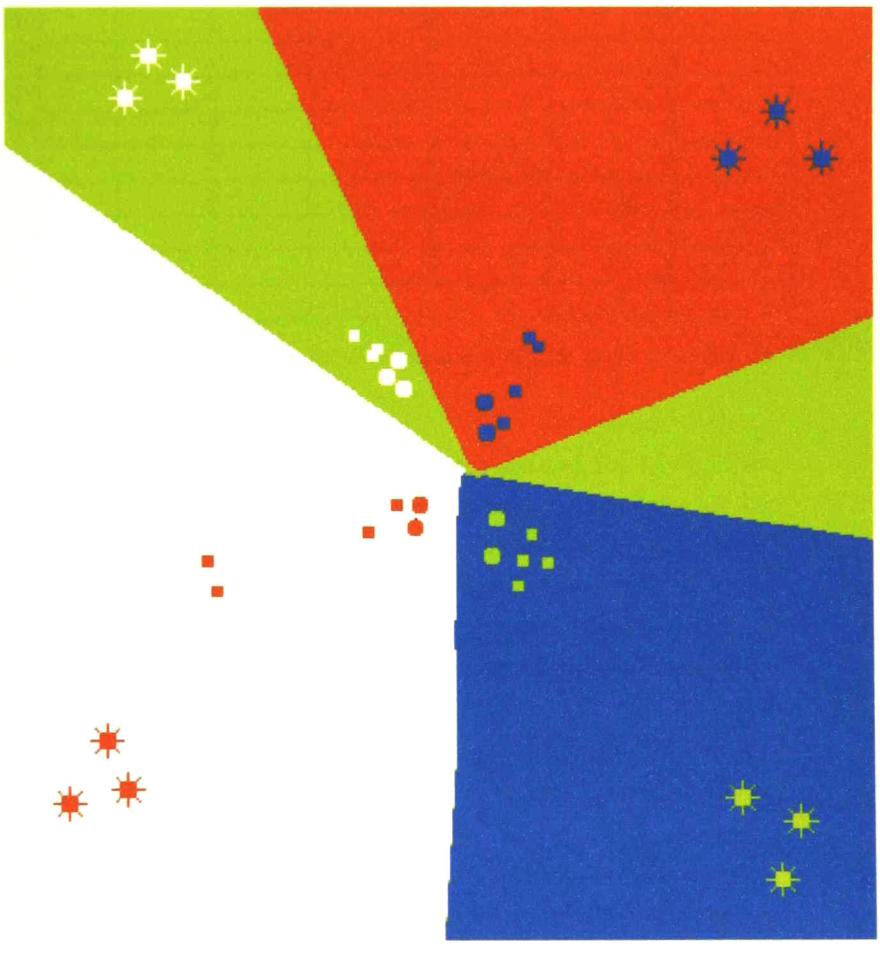


Figura 4.9: Resultados de clasificación usando una SMVM con kernel 2D ->8D para la aplicación del problema de la fábrica de llaves. Iteración 1000.

Vector de prueba	Clase deseada	Clase obtenida (K. Gaussiano)	Clase obtenida (K. Geométrico)
[299,42]	1	1	1
[317,38]	1	1	1
[331,50]	1	1	1
[44,332]	2	2	2
[34,347]	2	2	2
[48,349]	2	2	2
[313,326]	*3*	*1*	*3*
[331,331]	*3*	*1*	*3*
[321,353]	*3*	*1*	*3*
[54,19]	*4*	*1*	*4*
[51,28]	*4*	*1*	*4*
[61,30]	*4*	*1*	*4*

Cuadro 4.2: Conjunto de datos de prueba para problema cuyo espacio de entrada es de dimensión 2.

La última prueba a la que sometimos a nuestro sistema fue la de introducirle, en la fase de prueba, vectores que representan puntos fuera de las esferas de entrenamiento (a los que llamamos ***puntos difusos***), pero que se encontraban suficientemente cerca de esferas de determinada clase X como para que la SMVM la considerara de dicha clase (por su cercanía en el espacio y por las propiedades de generalización de estos sistemas), obteniendo resultados también satisfactorios.

Las figuras 4.10, 4.11 muestran un experimento de los explicados en párrafos anteriores. En la figura 4.10 se observan las esferas contenedoras de la totalidad de los datos de entrenamiento, además de una esfera contenida en otra de las llamadas de entrenamiento y la clase a la que cada una representa según lo obtenido por el preprocesamiento que se realiza al entrenar el algoritmo Spherical k-means, en la figura 4.11 se muestran los vectores encapsulados por las esferas conformales clasificados correctamente por la SMVM, además de los puntos difusos que fueron clasificados según la cercanía con cada esfera. La tabla 4.3 muestra el resultado del entrenamiento de los vectores usados en esta prueba.

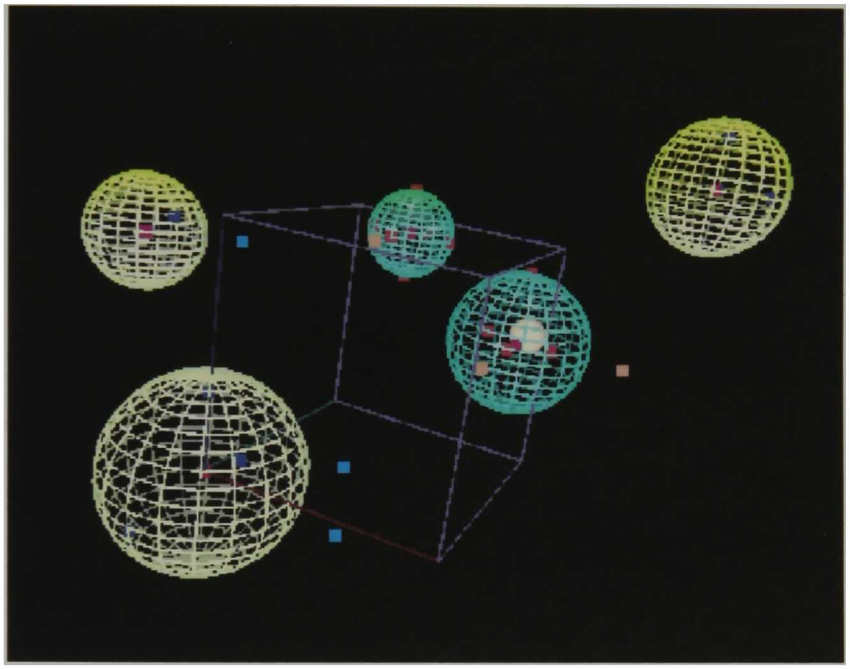


Figura 4.10: Esferas contenedoras de los datos de entrenamiento.

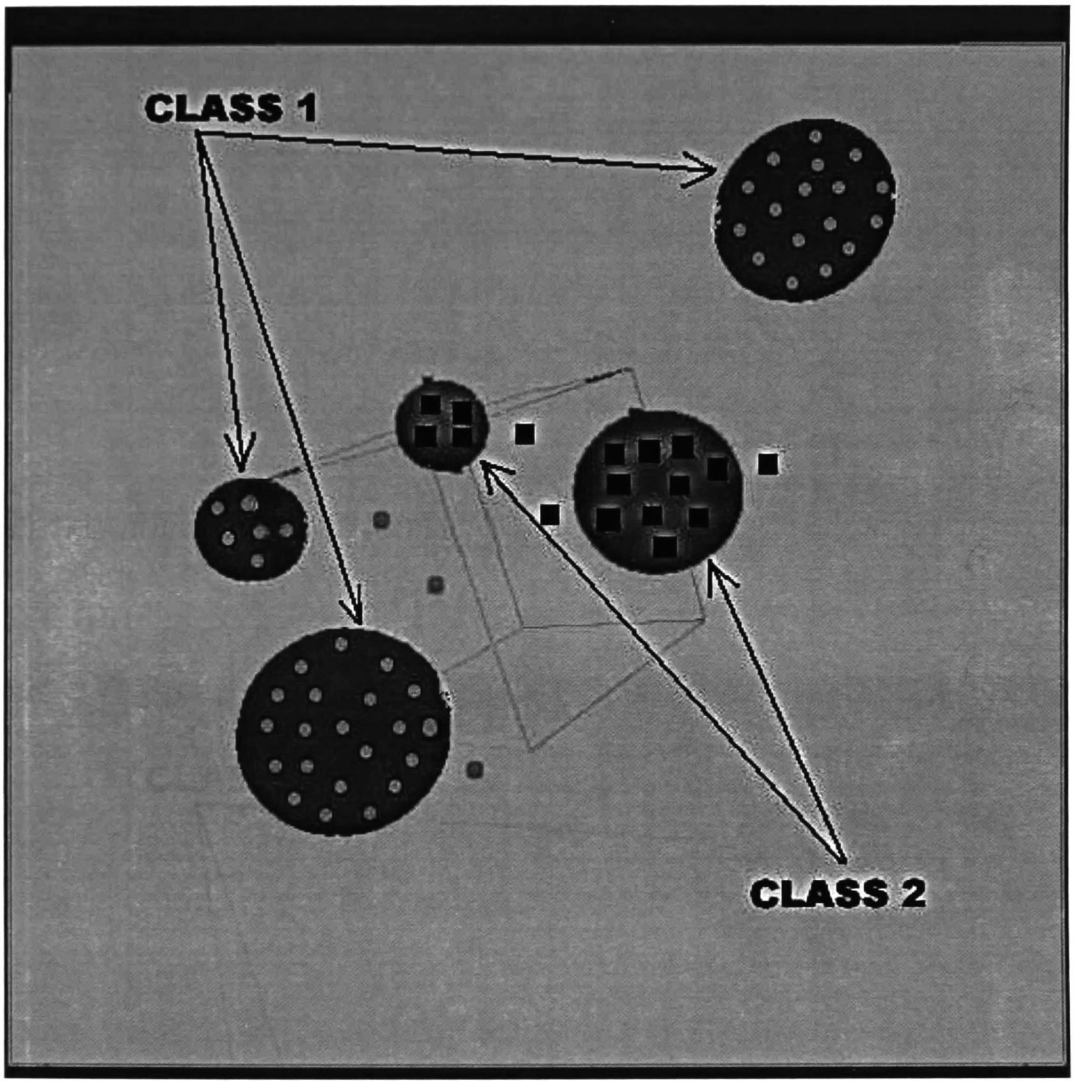


Figura 4.11: Totalidad de datos de entrenamiento como puntos en el espacio 3D.

Objeto	Salida Esperada	Salida Obtenida
s1=[1, 2, 0, 1.5, 2.5]	1	1
s2=[0, 0, 0, 1.5, 1.5]	1	1
s3=[-1.5, -1.5, -1.5, 2.155, 3.155]	-1	-1
s4=[-3, -1, 1, 4.755, 5.755]	-1	-1
s5=[3, 3, 2.5, 11.22, 12.22]		
Esfera dentro de s1		
s6=[1, 2.5, 3.09375, 4.09375]	-1	-1
Puntos dentro de s1		
ps11=[1, 2, 1, 1.5, 3.5]	1	1
ps12=[1, 2.5, 0, 3.125, 4.125]	1	1
ps13=[1, 1.5, 0, 1.125, 2.125]	1	1
ps14=[1.5, 2, 0, 2.625,3.625]	1	1
ps15=[0.5, 2.2, 0, 2.045, 3.045]	1	1
Puntos dentro de s2		
ps21=[0, 0, 1.5, 0.625, 1.625]	1	1
ps22=[0, 0.5, 1.5, 0.75, 1.75]	1	1
ps23=[0, -0.5, 1.5, 0.75, 1.75]	1	1
ps24=[0.5, 0, 1.5, 0.75, 1.75]	1	1
Puntos dentro de s3		
ps31=[-1.5, -1.5, -1.5, 2.875, 3.875]	-1	-1
ps32=[-2, -2.1, -2.2, 6.125, 7.125]	-1	-1
ps33=[-1, -1.5, -1.2, 1.845, 2.845]	-1	-1
ps34=[-1.5, -1.5, -0.5, 1.875, 2.875]	-1	-1
Puntos dentro de s4		
ps41=[-3, -1, 1, 5, 6]	-1	-1
ps42=[-2.4, -1.2, 1.3, 3.945, 4.945]	-1	-1
ps43=[-3, -0.5, 0.5, 4.25, 5.25]	-1	-1
ps44=[-3, -0.4, 1, 4.58, 5.58]	-1	-1
Puntos dentro de s5		
ps51=[3, 3, 2.5, 11.625, 12.625]	-1	-1
ps52=[3.7, 3, 2.5, 13.97, 14.97]	-1	-1
ps53=[3, 3.5, 2.5, 13.25, 14.25]	-1	-1
ps54=[3, 3, 3.2, 13.62, 14.62]	-1	-1
Puntos fuera de las esferas		
po1=[2.5, 2, 0, 4.625, 5.625]	1	1
po2=[1, 0.5, 0, 0.125, 1.125]	1	1
po3=[0, -0.9, 1.5, 1.03, 2.03]	1	1
po4=[-0.3, -0.3, -1.5, 0.715, 1.715]	-1	-1
po5=[-0.3, -0.3, -2.5, 2.715, 3.715]	-1	-1
po6=[-2.1, -0.3, 1, 2.25, 3.25]	-1	-1

Cuadro 4.3: Clasificación de esferas y puntos codificados en Àlgebra Geométrica Conformal.

Capítulo 5

CONCLUSIONES.

El uso de SVM en aplicaciones tales como reconocimiento de texto, reconocimiento de imágenes [10, 9, 15, 12], reconocimiento de rostros [16] y muchas otras más, ponen de manifiesto la importancia que estos sistemas han tomado en nuestros días. La implementación de una SVM destinada a resolver un problema determinado requiere que se realice una elección crucial: la decisión de qué kernel emplear; así como en el uso de MLP's la arquitectura de la red se define con respecto a las características del problema tratado, en SVM es necesario saber qué características de generalización deseamos que la clasificación aprendida presente, conforme a la solución que pretendemos ofrecer. Son estas razones las que hacen surgir la necesidad de ampliar la plantilla de kernels existentes.

El primer resultado importante de este trabajo se obtuvo al implementar nuestro propio programa del algoritmo de las SVM's, ya que de esta manera, adquirimos el conocimiento necesario para entender a profundidad el funcionamiento de estas máquinas de aprendizaje para posteriormente dedicar nuestros esfuerzos a mejorar el algoritmo por medio del desarrollo de los kernels geométricos obtenidos.

En este trabajo de tesis se pone de manifiesto que las grandes capacidades del Álgebra Geométrica son de gran ayuda en busca de realizar mapeos a espacios de características de dimensiones mayores con respecto a los espacios de entrada, esto gracias al uso del producto Clifford y de los multivectores

(se da el nombre a esta derivación de SVM a Support Multivector Machines -SMVM-). Con ello obtuvimos mapeos a dimensiones en las que encontramos muchas relaciones lineales que nos ayudaron a resolver problemas que se presentaban como no lineales en sus respectivas dimensiones de entrada (2D y 3D). Nos aseguramos que los kernels obtenidos nos entregan clasificaciones con características de generalización diferentes a las resultantes cuando se hace uso de los kernels ya existentes (espacialmente con los kernels polinomiales, ya que las funciones kernel obtenidas con el Álgebra Geométrica se elaboraron con la misma filosofía que dichos kernels polinomiales -ver Sección 4.1.1) por lo que los problemas a solucionar con estos nuevos kernels pueden ser diferentes a los tratados con los ya existentes.

Además se comprobó que el gran poder de descripción y codificación de entidades geométricas de las Álgebras Geométricas, en el trabajo desarrollado en esta tesis específicamente hablamos del Álgebra Conformal (AGC), añade capacidades extras a las SVM. El preprocesamiento del conjunto de datos de entrenamiento, englobando la totalidad de dichos datos en esferas nos permitió comprimirlos de una manera bastante considerable (no la podemos cuantizar debido a que el número de puntos 3D que engloba el volumen de una esfera puede ser infinito). Este preprocesamiento nos hace capaces de representar un subconjunto de datos de entrenamiento por medio de una esfera, codificada como un 5-vector según el álgebra geométrica conformal, por lo que el aprendizaje de la SMVM es también considerablemente más rápido. Durante la etapa de prueba (*test*) del sistema los vectores que introducimos pueden ser tanto puntos en 3D o volúmenes esféricos (también debidamente codificados dentro del AGC), por lo que nos permite probar también subconjuntos de datos o datos individuales (esto es, la compresión también es aplicable durante la fase de prueba).

Trabajo futuro.

A manera de continuación del desarrollo de la combinación entre las Álgebras Geométricas y las Support Vector Machines se pretende que, como objetivo de un trabajo doctoral, se implementen SMVM que trate de com-

probar la teoría de Vladimir Vapnik que intenta maximizar el margen de los multivectores de soporte de una manera más óptima al mapear los datos de entrada a superficies esféricas y en éstas encontrar los hiperplanos separadores. Lo anterior con objeto de hacer corresponder los resultados de clasificación obtenidos de una SVM (conocido como *centro de Tchebycheff* en el espacio version[1]) con la solución más óptima de cada problema (o punto de Bayes). La ayuda del Álgebra Geométrica para el desarrollo de esta teoría es fundamental ya que se encargará de encontrar los mapeos a las superficies esféricas y obtener kernels que separen los conjuntos de entrenamiento, dado que las esferas son entidades básicas de estas álgebras.

Otra vertiente del trabajo doctoral es la de encontrar una regla la cual dado un espacio de entrada determinado encuentre el mapeo a multivectores adecuado para lo suficiente la dimensionalidad de entrada y ejecute los productos Clifford y punto entre estos multivectores de manera automática; es decir una regla que encuentre el kernel adecuado dado un espacio vectorial de entrada siempre haciendo uso de las Álgebras Geométricas adecuadas.

[1]El espacio version es el subespacio vectorial, contenido en el espacio de características, que se forma con todos los hiperplanos consistentes con los datos de entrenamiento.

Bibliografía

[1] Bayro-Corrochano E. *Geometric Computing for Perception Action Systems.* Springer Verlag, New York, 2001.

[2] Cristianini N. and Shawe-Taylor J. *Support Vector Machines and other kernel-based learning methods*, pp 189, Cambridge, United Kingdom, Cambridge University Press 2000.

[3] Vapnik V.N. *Statistical Learning Theory,* Wiley, New York, 1998.

[4] Burges C.J.C., A tutorial on Support Vector Machines for Pattern Recognition, *Knowledge Discovery and Data Mining*, Vol. 2, No. 2, pp 121-167, 1998.

[5] Boser B.E., Guyon I.M. and Vapnik V.N. A training algorithm for optimal margin classifiers, in *Proceedings of the 5th Annual ACM Workshop on Computational Learning Theory,* D.Haussler, pp. 144-152, Ed., 1992.

[6] Campbell, C. and Cristianini N. Simple training algorithms for support vector machines. *Technical Report CIG-TR-KA,* University of Bristol, Engineering Mathematics, Computational Intelligence Group, 1999.

[7] Friess T-T., Cristianini N., and Campbell C. The Kernel-Adatron Algorithm: a Fast and Simple Learning Procedure for Support Vector Machines, in *Proceedings 15th International Conference On Machine Learning, pp. 188-196, 1998.*

[8] Friess T-T. and Harrison R. Support Vector Neural Networks: The kernel Adatron with Bias and Soft Margin, in *Tech Report*, Dept. of Automatic Control And Systems Engineering, University of Sheffield, 1998.

[9] Blanz V., Schölkopf B., Bülthoff H., Burges C.J.C., Vapnik V.N. and Vetter T. Comparison of view-based object recognition algorithms using realistic 3D models, in *Artificial Neural Networks ICANN'96,* C. Von Der Malsburg, W. Von Seelen, J.C. Vorbrüggen, and B. Sendhoff, Eds., Berlin, Springer Lecture Notes in Computer Science, Vol 1112, pp 251-256, 1996,

[10] Roobaert D. and Van Hulle M.M., View-based 3D object recognition with Support Vector Machines, in *Proceedings IEEE International Workshop on Neural Networks for Signal Processing (NNSP99),* Madison, Wisconsin, USA, pp. 201-212, Agosto 1999.

[11] Shawe-Taylor J., Barlett P.L., Williamson R.C., anda Anthony M. A framework for structural risk minimization, in *Proceedings COLT.*, Morgan Kaufmann, pp 175-192, 1996.

[12] Pontil M. and Verri A. Support Vector Machines for 3-D Object Recognition, *IEEE Trans. on PAMI,* 20:637-646, 1998.

[13] Platt J.C. Sequential minimal optimization: A fast algorithm for training support vector machines, *Technical Report MSR-TR-98-14,* Microsoft Research, 1998

[14] Schölkopf B., Smola A. anda Müller K-R. Kernel principal components analysis *in W. Gerstner, A. Germond, M. Hasler, and J-D. Nicoud editors, Artificial Neural Networks - ICANN'97,* Springer Lecture Notes in Computer Science, Volume 1327, pp. 583-588, 1997.

[15] Weston J. and Watkins C., Support vector machines for multi-class pattern recognition, in *Proceedings of the 6th European Symposium on Artificial Neural Networks (ESANN), pp 185-201,* 1999.

[16] Huang J., Ii D., Shao X. and Wechsler H. Pose Discrimination and Eye Detection Using Support Vector Machines (SVM), in *Proceeding of NATO-ASI on Face Recognition: From Theory to Applications,* pp. 188-197, 1998.

[17] Fletcher R. *Practical Methods of Optimization,* John Wiley and Sons, Inc., 2nd edition, volume 1, pp. 23-63, 1987.

[18] Müller K-R., Mika S., Rätsch G., Tsuda K and Schölkopf B. An Introduction to Kernel-Based Learning Algorithms, *IEEE Trans. on Neural Networks,* Vol. 12, No. 2, pp. 181-202, Marzo 2001.

[19] Chinea C.S., Una aproximación a la teoría de los Espacios de Hilbert, *Divulgacion de la Matemática en la Red,* diciembre 2000.

[20] Lee Y., Lin Y. and Wahba G. Multicategory Support Vector Machines, *Technical Report No. 1043,* University of Wisconsin, Departament of Statistics, Septiembre 29, pp.10-35, 2001.

[21] Hestenes D., Li H. and Rockwood A. New algebraic tools for classical geometry. In *Geometric Computing with Clifford Algebras.* Sommer G. (editor) Chapter 1, Springer, Heidelberg, pp 3-23, 2001.

[22] Lounesto P. *Clifford Algebras and Spinors.* Cambridge University Press, Second edition. Chapters 1-3, 2001.

[23] Rosenhahn B. and Sommer G. *Pose estimation in conformal geometric algebra.* Report number 0206. Christian-Albrechts-Universität, Kiel, Germany, pp. 13-36, November, 2002.

[24] Inderjit S.D., Yuqiang G. and Kogan J. Iterative Clustering of High Dimensional Text Data Augmented by Local Search, in *Proceedings IEEE International Conference on Data Mining (ICDM'02),* Maebashi, Japan, pp. 131-142, December 2002.

[25] Dhillon I.S. and Modha D.S. A data-clustering algorithm on distributed memory multiprocessors. *Large Scale Parallel Data Mining, Lecture Notes in Artificial Intelligence,* Volume 1759, pp. 245-260, 2000.

Apéndice A

A.1 Espacios vectoriales.

Definición. Un conjunto X es un espacio vectorial (EV) si dos operaciones (digamos suma y multiplicación por un escalar) son definidas en X de tal manera que para todo $x, y \in X$ y $\alpha \in \Re$.

$$\begin{aligned} x + y &\in X \\ \alpha x &\in X \\ 1x &= x \\ 0x &= 0 \end{aligned}$$

aunado al hecho que X es un grupo conmutativo con elemento identidad 0 bajo la operación de adición y satisface las leyes distributivas para multiplicación escalar

$$\alpha(x + y) = \alpha x + \alpha y$$

y

$$(\alpha + \beta)x = \alpha x + \beta x$$

Los elementos de X son llamados *vectores,* mientras que los números

reales se les conoce como *escalares.*

Definición. Un espacio lineal con norma es un EV X junto con una función real que mapea cada elemento $x \in X$ a un número real ||x|| llamado *la norma de x* y que satisface las siguientes propiedades:

1. Positividad. $||x|| \geq 0$, $\forall x \in X$, la igualdad con cero se mantiene sólo si x=0;

2. Desigualdad triangular. $||x+y|| \leq ||x|| + ||y||$, $\forall x, y \in X$;

3. Homogeneidad. $||\alpha x|| = |\alpha| ||x||$, $\forall \alpha \in \Re$ y $\forall x \in X$.

Definición. En un espacio lineal con norma una secuencia infinita de vectores x_n se dice que *converge* a un vector x si la secuencia $||x-x_n||$ de números reales converge a cero.

A.2 Espacios con producto punto.

Definición. Una función f de un espacio vectorial X a un espacio vectorial Y se dice que es *lineal* si para todo $\alpha, \beta \in \Re$ y $x, y \in X$

$$f(\alpha x + \beta y) = \alpha f(x) + \beta f(y)$$

Ejemplo. Sean $X=\Re^n$ y $Y=\Re^m$. Una función lineal de X a Y puede ser representada como una matriz A de tamaño mxn con entradas A_{ij} tales que el vector $\mathbf{x} = (x_1, ..., x_n)'$ son mapeadas al vector $\mathbf{y} = (y_1, ..., y_m)'$ donde y

$$y_i = \sum_{j=1}^{n} A_{ij} x_j, i = 1, ..., m$$

Una matriz con entradas $A_{ij} = 0$, para $i \neq j$ es llamada *matriz diagonal.*

Definición. Un espacio vectorial X es llamado un *espacio con producto punto* si existe un mapeo bilineal (lineal en cada argumento) que para cada dos elementos $x,y \in X$ otorga un número real denotado por <x.y>que satisface las siguientes propiedades:

<x.y>= <y.x>

<x.x>$\geq$ 0 y <x.x>=0 $\Leftrightarrow$ x=0.

La cantidad <x.y>es llamado el *producto punto* de x y de y también conocido como *producto interno* o *producto escalar*.

Definición. Una métrica en un espacio vectorial es una distancia o medida, esto es, una correspondencia tal que a cada para de vectores le corresponde una *medida*, el producto punto induce una métrica en un espacio vectorial. Una métrica se dice *euclidiana,* si el producto interior cumple las siguientes condiciones:

1. Conmutatividad

$$\forall \text{ x,y } \in \text{X, } <\text{x.y}>=<\text{y.x}>;$$

2. Distributividad con respecto a la suma vectorial

$$\forall x, y, z \in X, \ <\text{x.(y+z)}>=<\text{x.y}>+ \ <\text{x.z}>;$$

3. Asociatividad mixta

$$\forall x, y \in X, \ \alpha \in \Re^n, \ \alpha < x.y >=< (\alpha x).y >=< x.(\alpha y) >;$$

4. Definición positiva

$$\forall x \in X, \ <\text{x.x}> \geq 0;$$

5. No degeneración

$$<\text{x.x}>= 0 \Rightarrow \text{x}=0.$$

Se llama *métrica euclidiana ordinaria* a aquella métrica definida por la condición de ortonormalidad de la base.

Ejemplo. Sea $X = \Re^n$, $\mathbf{x} = (x_1, ..., x_n)'$, $\mathbf{y} = (y_1, ..., y_n)'$. Sean λ_i números positivos. La siguiente es una definición de un producto punto válido:

$$< x.y >= \sum_{i=1}^{n} \lambda_i x_i y_i = \mathbf{x}'\mathbf{A}\mathbf{y}$$

donde $\mathbf{A}$ es una matriz diagonal de tamaño $n \, x \, n$ con entradas diferentes de cero para $\mathbf{A_{ii}} = \lambda_i$.

Ejemplo. Sea $X = C[a, b]$ el espacio vectorial de funciones continuas en el intérvalo [a,b] con las definiciones obvias de adición y multiplicación por un escalar. Para $f, g \in X$ se define

$$< f.g >= \int_a^b f(t)g(t)dt$$

De la definición de producto punto se siguen las dos siguientes propieda-

des:

<0.y>=0

X es automáticamente un espacio con norma definida como

$$||x|| = \sqrt{< x.x >}$$

Definición. Dos elementos x y y son llamados *ortogonales* si <x.y>=0. Un conjunto $S = \{x_1, ..., x_n\}$ de vectores de X es llamado *ortonormal* si $<x_i.x_j>=\delta_{ij}$, donde $\delta_{ij} = 1$ para i=j, y $\delta_{ij} = 0$ en otro caso. Para un conjunto S ortonormal, y un vector y$\in X$, la expresión

$$\sum_{i=1}^{n} < x_i.y > x_i$$

se dice ser una *serie de Fourier* para y.

Si S forma una base ortonormal para X, cada vector y es igual a su serie de Fourier.

Teorema. (Desigualdad de Schwarz) para un espacio con producto punto

$$| < x.y > |^2 \leq < x.x >< y.y >$$

donde la igualdad se presenta si y solo si x y y son linealmente dependientes.

Teorema. Para los vectores x y y que pertenecen al espacio con producto punto X

$$||x + y||^2 = ||x||^2 + ||y||^2 + 2 < x.y >$$
$$||x - y||^2 = ||x||^2 + ||y||^2 - 2 < x.y >$$

Definición. El ángulo θ entre dos vectores x y y de un espacio con producto punto es defiido por

$$cos\theta = \frac{< x.y >}{||x||||y||}$$

Si $| < x.y > | = ||x||||y||$, el coseno es 1, para $\theta = 0$, y es entonces cuando x y y son llamados *vectores paralelos.* Si <x.y>=0, el coseno es 0, $\theta = \frac{\pi}{2}$ y los vectores son llamados *vectores ortogonales.*

Definición. Dado un conjunto $S = \{x_1, ..., x_n\}$ de vectores pertenecientes a un espacio vectorial con producto punto X, entonces la matriz de tamaño $n\,x\,n$ G con entradas $G_{ij} =< x_i.x_j >$ es llamada la *Gramm matriz* de S.

A.3 Espacios de Hilbert.

Definición. Un *espacio prehilbertiano* es un espacio vectorial con métrica Euclidiana. si el espacio prehilbertiano es finitodimensional, se dirá entonces que es un espacio *Euclidiano de n dimensiones.*

Definición. Una sucesión de vectores $\{x_n\}$=$\{x_1, ..., x_n, ...\}$ del espacio métrico se dice convergente hacia ℓ si se verifica que, dado un escalar positivo, por muy pequeño que fuera, ε, existe un número natual N tal que para todo n$\geq$ N, es $||\ell - x_n||<\varepsilon$.

El vector ℓ se dice que es el límite de la sucesión convergente.

Definición. Una secuencia x_n en un espacio lineal con norma se dice ser una *secuencia de Cauchy* si $||x_n - x_m|| \to 0$ como n,m$\to \infty$. Más precisamente, dado $\varepsilon > 0$, existe un entero N tal que $||x_n - x_m|| < \varepsilon$ para todo n,m>N. Un espacio se dice ser *completo* cuando cada secuencia de Cauchy del espacio converge a un elemento del espacio.

En un espacio con norma cada secuencia convergente es una secuencia de Cauchy, pero el complementario no siempre es verdad. Los espacios en los que cada secuencia de Cauchy tiene un límite se dicen ser completos. Los espacios lineales completos con norma son llamados Espacios de Banach.

Definición. Un espacio vectorial H es *separable* si existe un subconjunto contable $D \subseteq H$, tal que cada elemento de H es el límite de una secuencia

de elementos de D. Un *espacio de Hilbert* es un espacio vectorial separable completo con producto punto, o en otras palabras a un espacio prehilbertiano completo.

Espacios vectoriales de dimensión finita como $\Re^n$ son espacios de Hilbert.

Apéndice B

Definiciones básicas en álgebra geométrica

En esta sección se introducen algunos conceptos básicos que son útiles en algunas secciones del documento, sobre todo aquellas donde se explica cómo realizar transformaciones (rotaciones, traslaciones) en álgebras geométricas.

Inversión: se dice que un elemento M en $G_{p,q,r}$ es invertible si existe un elemento N en $G_{p,q,r}$ tal que $MN = NM = 1$. El elemento N es único en caso de existir y es llamado el *inverso* de M.

r-vector: es la combinación lineal de r-blades (recuerde que un r-blade, también llamado blade de grado r, es el producto exterior (wedge) de r vectores diferentes y es denotado por $\langle M \rangle_r$).

Reversión: se define como

$$\left\langle M^{\dagger} \right\rangle_i = (-1)^{\frac{i(i-1)}{2}} \langle M \rangle_i \tag{B.1}$$

es decir, la reversión es un anti-automorfismo (un anti-automorfismo es un mapeo lineal que inverte el orden de los productos geométricos).

Involución: es un mapeo lineal invertible cuya composición con él mismo es la transformación identidad.

Involución de grado: se define como

$$\left\langle M^{\ddagger}\right\rangle_i = (-1)^i \langle M\rangle_i \tag{B.2}$$

Multivector par: un multivector M es par (o tiene paridad par) si $M^{\ddagger} = M$

Multivector impar: un multivector M es impar (o tiene paridad impar) si $M^{\ddagger} = -M$

Magnitud de un multivector: se define como

$$|M| = \sqrt{\sum_{i=0}^{n} |\langle M\rangle_i|^2} \tag{B.3}$$

donde

$$|\langle M\rangle_i| = \sqrt{|\langle M\rangle_i \cdot \langle M\rangle_i|} \tag{B.4}$$

Componentes de un vector respecto a otro: consideremos la siguiente figura insertar figura de vectores a y b
La componente paralela de a es un múltiplo escalar del vector unitario $\frac{b}{|b|^2}$:

$$a_{\|} = (a \cdot b)\frac{b}{|b|^2} = (a \cdot b)\, b^{-1} \tag{B.5}$$

La componente perpendicular de a esta dada por

$$a_{\perp} = a - a_{\|} = a - (a \cdot b)\, b^{-1} = (ab - a \cdot b)\, b^{-1} = (a \wedge b)\, b^{-1} \tag{B.6}$$

Reflexión: la reflexión de un vector x a través de una línea a se obtiene al enviar $x = x_{\|} + x_{\perp}$a $x' = x_{\|} - x_{\perp}$, por tanto

$$\begin{aligned} x' &= (x \cdot a)\, a^{-1} - (x \wedge a)\, a^{-1} \\ &= (x \cdot a - x \wedge a)\, a^{-1} \\ &= (a \cdot x + a \wedge x)\, a^{-1} \\ &= axa^{-1} \end{aligned} \tag{B.7}$$

Nota: Esta fórmula se obtiene solamente utilizando las propiedades de conmutación del producto interior y exterior, las definiciones de componentes de un vector y la definición de producto Clifford.

Apéndice C

Cálculos matemáticos

En este apéndice se colocan cálculos que por su extensión o naturaleza no se consideraron conveniente colocarlos en el desarrollo de los temas expuestos en los capítulos ?? a ??.

Cálculo del cuadrado del pseudoescalar del álgebra geométrica $G_{3,0,0}$

Para el cálculo de I^2 hacemos uso de la definición de producto geométrico, de la ecuación 2.7 que define el producto punto entre 2 blades y de las propiedades enunciadas en 2.4.

$$
\begin{aligned}
I^2 = I\ I &= (e_1 \wedge e_2 \wedge e_3) \cdot (e_1 \wedge e_2 \wedge e_3) \qquad \text{(C.1)} \\
&= ((e_1 \wedge e_2 \wedge e_3) \cdot e_1) \cdot (e_2 \wedge e_3) \\
&= (e_1 \wedge e_2 \wedge (e_3 \cdot e_1) - e_1 \wedge (e_2 \cdot e_1) \wedge e_3 + (e_1 \cdot e_1) \wedge e_2 \wedge e_3) \cdot (e_2 \wedge e_3) \\
&= (e_2 \wedge e_3) \cdot (e_2 \wedge e_3) \\
&= ((e_2 \wedge e_3) \cdot e_2) \cdot e_3 \\
&= (e_2 \wedge (e_3 \cdot e_2) - (e_2 \cdot e_2) \wedge e_3) \cdot e_3 \\
&= (-e_3) \cdot e_3 \\
&= -1
\end{aligned}
$$

Elementos de la base del álgebra $G_{4,1}$

$$G_{4,1} = span \left\{ \begin{array}{cc} 1 & escalar \\ e_1, e_2, e_3, e_+, e_- & vectores \\ e_{12}, e_{13}, e_{1+}, e_{1-} & \\ e_{23}, e_{2+}, e_{2-} & bivectores \\ e_{3+}, e_{3-}, e_{+-} & \\ e_{123}, e_{12+}, e_{12-}, e_{13+} & \\ e_{13-}, e_{1+-}, e_{23+}, e_{23-} & trivectores \\ e_{2+-}, e_{3+-} & \\ e_{123+}, e_{123-}, e_{12+-}, e_{13+-}, e_{23+-} & 4-vectores \\ e_{123+-} & pseudoescalar \end{array} \right\} \quad \text{(C.2)}$$

Centro de Investigación y de Estudios Avanzados del IPN
Unidad Guadalajara

El Jurado designado por la Unidad Guadalajara del Centro de Investigación y de Estudios Avanzados del Instituto Politécnico Nacional, aprobó la tesis:

DISEÑO DE KERNELS PARA MÁQUINAS DE MULTIVECTORES DE SOPORTE USANDO ÁLGEBRA GEOMÉTRICA

del (la) C.

Nancy Guadalupe ARANA DANIEL

el día 5 de Diciembre de 2003.

Dr. Edgar Nelson SANCHEZ CAMPEROS
Investigador Cinvestav 3B
CINVESTAV GDL
Jalisco

Dr. Eduardo Jose BAYRO CORROCHANO
Investigador Cinvestav 3B
CINVESTAV GDL
Jalisco

Dr. Félix Francisco RAMOS CORCHADO
Investigador Cinvestav 2B
CINVESTAV GDL
Jalisco