CAPÍTULO 5.
DETECCIÓN DE MICROCALCIFICACIONES.

5.1 MÁQUINAS DE VECTOR SOPORTE (SVM).

Finalmente, para decidir si los píxeles candidatos son microcalcificaciones o no, vamos a usar un **descriptor** que está basado en una máquina de vectores soporte. Un descriptor no es más que una característica de un objeto, el cual puede, o no, agruparse con otros descriptores, formando un patrón que sirva para clasificar el objeto dentro de una clase. Un buen descriptor es aquella característica por la que es más fácil discriminar entre las distintas clases a las que puede pertenecer el objeto. Por tanto, en nuestro caso parece lógico elegir como descriptores aquellas características que midan la forma del candidato y su intensidad, ya que son las características que diferencian a una microcalcificación de cualquier otro elemento de la imagen.

Los fundamentos teóricos de las máquinas de vector soporte (*SVM, del inglés Support Vector Machines*) se encuentran en los trabajos de Vapnik (véase referencia 15) y otros autores sobre la teoría del aprendizaje estadístico, desarrollados a finales de los años 70 y durante los 80. El modelo de SVM, tal como se entiende actualmente, fue presentado en la conferencia COLT (*COmputacional Learning Theory*) del año 1992 y a partir de ese momento el interés por este modelo de aprendizaje no ha parado de crecer hasta el presente. Como consecuencia, tanto su desarrollo teórico como el campo de las aplicaciones reales han experimentado un avance muy importante, hasta el punto que, hoy en día, las SVM constituyen en referente completamente establecido para las disciplinas del aprendizaje automático y de la minería de datos. Los campos donde las SVM han sido aplicadas con éxito incluyen, entre otros, la visión por computador, la bioinformática, la recuperación de información o el procesamiento del lenguaje natural. Es interesante comentar que la comunidad científica que trabaja actualmente en el estudio y desarrollo de SVM es marcadamente interdisciplinaria puesto que incluye aspectos y técnicas del aprendizaje automático, optimización, estadística, análisis funcional…

Para una tarea de aprendizaje, con una cantidad finita de datos de entrenamiento, el mejor rendimiento de generalización se conseguirá si se equilibra la balanza entre la exactitud obtenida sobre ese conjunto de entrenamiento y la capacidad de la máquina, esto es, la habilidad de la máquina de aprender cualquier conjunto de entrenamiento sin error.

5.1.1 Separación de puntos con hiperplanos en espacios D-dimensionales.

Las máquinas vectores soporte pertenecen a la familia de los *clasificadores lineales* puesto que introducen separadores lineales o hiperplanos en espacios de características de muy alta dimensionalidad (introducidas por funciones núcleo o kernel) con un sesgo inductivo muy particular (maximización del margen).

En primer lugar, recordemos que todo hiperplano en un espacio D-dimensional se puede expresar como:

$$h(x) =< w, x > +b \tag{5.1}$$

donde $w \in \mathbf{R}D$ es el vector ortogonal al hiperplano $b \in \mathbf{R}$ y $<\cdot,>$ expresa el producto escalar habitual en **R**D. Visto como un clasificador binario, la regla de clasificación se puede expresar como:

$$f(x) = signo(h(x)) \tag{5.2}$$

donde la función signo se define como:

$$signo(x) = \begin{Bmatrix} 1 \rightarrow x \geq 0 \\ -1 \rightarrow x < 0 \end{Bmatrix} \tag{5.3}$$

En la terminología de la clasificación, las $x \in \mathbf{R}D$ son representaciones vectoriales de los ejemplos, con una componente real por cada atributo, y el vector w se suele denominar *vector de pesos*. Este vector contiene un peso para cada atributo indicando su importancia o contribución en la regla de clasificación. Finalmente, b suele denominarse sesgo (*bias*) y define el umbral de decisión.

Dado un conjunto binario (es decir, con dos clases) de datos linealmente separables (como puede ser un vector, siendo éste el caso que nos ocupa), existen diversos algoritmos incrementales para construir hiperplanos *(w,b)* que los clasifiquen correctamente. A pesar de que esté garantizada la convergencia de todos ellos hacia un hiperplano solución, las particularidades de cada algoritmo de aprendizaje pueden conducirnos a soluciones ligeramente distintas, puesto que puede haber varios, infinitos de hecho, hiperplanos que separen correctamente el conjunto de ejemplos.

Suponiendo que el conjunto de ejemplos el linealmente separable, ¿cuál es el *mejor* hiperplano separador en términos de generalización? La idea que subyace tras las SVM de *margen máximo* consiste en seleccionar el hiperplano separador que está en la misma distancia de los ejemplos más cercanos de cada

clase. De manera equivalente, es el hiperplano que maximiza la distancia mínima (o *margen geométrico*) entre los ejemplos del conjunto de datos y el hiperplano. Intuitivamente, este hiperplano está situado en la posición más neutra posible con respecto a las clases representadas por el conjunto de datos, sin estar sesgado, por ejemplo, hacia la clase más numerosa. Además, sólo considera los puntos que están en las fronteras de la región de decisión, que es la zona donde puede haber dudas sobre a qué clase pertenece un ejemplo (son los denominados vectores soporte). En la figura que se muestra a continuación se presenta geométricamente este hiperplano equidistante para el caso bidimensional.

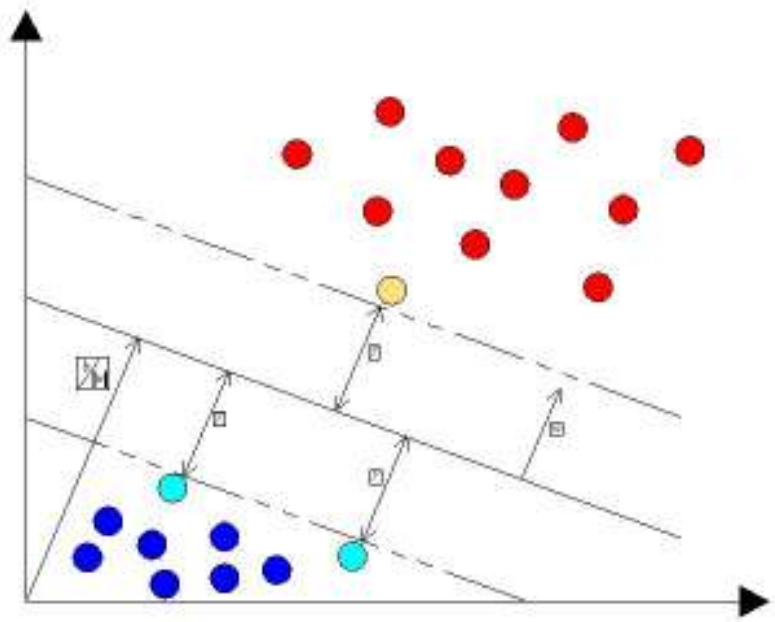

Figura 5.1. Hiperplano *(w,b)* equidistante a dos clases, margen geométrico y vectores soporte.

5.1.2 Minimización estructural del riesgo.

Este sesgo inductivo de aprendizaje consistente en maximizar el margen se justifica dentro de la teoría del aprendizaje estadístico y se enmarca en el principio de *minimización del riesgo estructural* (SRM). En esta teoría, maximizar el margen geométrico se ve como una buena heurística para minimizar la complejidad de la clase de hiperplanos separadores que, a su vez, interviene directamente en las expresiones que acotan superiormente el error de generalización. A nivel práctico, el hiperplano separador de margen máximo ha demostrado una muy buena capacidad de generalización en numerosos problemas reales, así como una robustez notable frente al sobreajuste u *overfitting*.

A nivel algorítmico, el aprendizaje de las SVM representa un problema de optimización con restricciones que se puede resolver usando técnicas de programación cuadrática (QP). La convexidad garantiza una solución única (esto supone una ventaja con respecto al modelo clásico de redes neuronales) y las implementaciones actuales permiten una eficiencia razonable para problemas reales con miles de ejemplos y atributos.

5.1.3 Funciones núcleo (*kernel functions*).

El aprendizaje de separadores no lineales con SVM se consigue mediante una transformación no lineal del espacio de atributos de entrada (*input space*) en un espacio de características (*feature space*) de dimensionalidad mucho mayor y donde sí es posible separar linealmente los ejemplos. El uso de las denominadas funciones núcleo, que calculan el producto escalar de dos vectores en el espacio de características, permite trabajar de una manera eficiente en el espacio de características sin necesidad de calcular explícitamente las transformaciones de los ejemplos de aprendizaje.

El aprendizaje en espacios características vía transformaciones no lineales por medio de funciones núcleo no es exclusiva del paradigma SVM. Aunque se suele asociar los métodos basados en funciones núcleo con las SVM al ser su ejemplo más paradigmático y avanzado, hay muchos otros algoritmos que se puede *kernelizar* para permitir el aprendizaje funciones no lineales.

Un requisito básico para aplicar con éxito las SVM a un problema real es la elección de una función núcleo adecuada, que debe reflejar el conocimiento a priori sobre el problema. El desarrollo de funciones núcleo para estructuras no vectoriales (por ejemplo, estructuras secuenciales, árboles o grafos) es actualmente una importante área de investigación con aplicación en dominios como el procesamiento del lenguaje natural y la bioinformática.

5.1.4 Modelo SVM con margen blando.

A veces, los ejemplos de aprendizaje no son linealmente separables ni tan siquiera en el espacio característica. Otras veces no es deseable conseguir un separador perfecto del conjunto de aprendizaje, puesto que los datos de aprendizaje no están libres de errores (ejemplos mal etiquetados, valores de atributos mal calculados, inconsistencias…).

Centrarse demasiado en todos los ejemplos de aprendizaje puede comprometer seriamente la generalización del clasificador aprendido por culpa del sobreajuste. En estos casos es preferible ser más conservador y admitir algunos ejemplos de aprendizaje mal clasificados a cambio de tener separadores más generales y prometedores. Este comportamiento se consigue mediante la introducción del modelo de SVM con margen blando (soft margin). En este caso, la función objetivo que deseamos minimizar está compuesta por la suma de dos términos:

1. **El margen geométrico.**

2. **Término de regularización** que tiene en cuenta los ejemplos mal clasificados.

La importancia relativa de los dos términos se regula mediante un parámetro, normalmente llamado C. Este modelo, aparecido en 1995, es el que realmente abrió la puerta a un uso real y práctico de las SVM, aportando robustez frente al ruido.

No obstante, dado que este tipo de modelo no va a ser utilizado para la implementación de nuestra herramienta, no profundizaremos más en este asunto.

5.2 MÁQUINAS DE VECTOR SOPORTE PARA CLASIFICACIÓN BINARIA.

En esta sección veremos los principales ingredientes de las SVM para clasificación binaria: la maximización del margen y el uso de funciones núcleo. La resolución del problema de optimización planteado siguiendo la idea de la maximización del margen tendrá como consecuencia la aparición de los vectores soporte que dan nombre a las SVM.

Para fijar la notación, consideremos el problema de clasificación binaria dado por un conjunto de N datos $S = \{(x_1, y1),(x_2, y_2),...,(x_N, y_N)\}$, donde cada x_i pertenece a un cierto espacio de entrada X, y cada y_i pertenece a {-1,1} e indica la clase a la que pertenece x_i. A los elementos de X los denominaremos ejemplos o vectores. Para simplificar, supondremos que X es un conjunto del espacio D-dimensional.

En los casos en que los datos se presenten de manera diferente (habitualmente estructuras discretas de tipo secuencia, árbol o grafo), también es posible trabajar con SVM diseñando funciones núcleo específicas para el problema.

En primer lugar, se describirán las SVM lineales con margen máximo que servirán como introducción a modelos más complejos y robustos. Asumimos conocimientos básicos sobre problemas de optimización.

5.2.1 SVM lineal con margen máximo.

La SVM lineal con margen máximo (*maximal margin linear SVM*) es el modelo más sencillo e intuitivo de SVM, aunque también el que tiene condiciones de aplicabilidad más restringidas puesto que parte de la hipótesis de que el conjunto de datos es linealmente separable en el espacio de entrada. Aun así, contiene muchas de las ideas subyacentes en la teoría de las SVM y es la base de todas las demás.

Supongamos que el conjunto de datos es linealmente separable en el espacio entrada. Es decir, sin hacer ninguna transformación de los datos, los ejemplos pueden ser separados por un hiperplano de manera que cada en lado del mismo sólo hay ejemplos de una clase. En términos matemáticos es equivalente a decir que existe un hiperplano $h : X \to \Re$ tal que h(x)>0 para los ejemplos de la clase +1 y $h(x)<0$ para los ejemplos de la clase -1. De manera más

concisa, *h* cumple que $y_i \cdot h(x_i)>0$ para todo i entre 1 y N, es decir, para todos los ejemplos.

5.2.1.1 Formulación original de las SVM.

Recordemos que la idea que hay detrás de las SVM consiste en seleccionar el hiperplano separador que está a la misma distancia de los ejemplos más cercanos de cada clase. Es muy fácil ver que la distancia de un vector *x* a un hiperplano *h*, definido por *(w,b)* como $h(x) =< w,b > +b$, viene dada por la fórmula:

$$dist(h,x) = \frac{|h(x)|}{\|w\|} \qquad \textbf{(5.4)}$$

donde $\|w\|$ es la norma en **R**D asociada al producto escalar, es decir:

$$\|w\|^2 =< w,w > \qquad \textbf{(5.5)}$$

Así pues, el hiperplano equidistante a dos clases es el que maximiza el valor mínimo de *dist(h,x)* en el conjunto de datos. Además, dados dos puntos z_1 y z_2 equidistantes a un hiperplano, se cumple que:

$$b = -\frac{(< w,z_1 > + < w,z_2 >)}{2} \qquad \textbf{(5.6)}$$

Como el conjunto es linealmente separable, podemos reescalar w y b de manera que la distancia de los vectores más cercanos al hiperplano sea $1/\|w\|$ (al multiplicar w y b por una constante la distancia no varía). Como consecuencia, los vectores z más cercanos tendrán $|h(z)| = 1$, mientras que para el resto $|h(z)| \geq 1$. De manera que el problema de encontrar el hiperplano equidistante a dos clases se reduce a encontrar la solución del siguiente problema de optimización con restricciones:

1. Maximizar $1/\|w\|$.
2. Sujeto a: $y_i(< w,x_i > +b) \geq 1$ con i$\in$[1,N].

El *margen funcional* de un ejemplo (x,y) con respecto a una función f se define como y·f(x), mientras que el margen geométrico (o normalizado) de un hiperplano se define como $1/\|w\|$. Con estas definiciones, la solución de la SVM

lineal con margen máximo es el hiperplano que maximiza el *margen geométrico*, restringido a que el margen funcional de cada ejemplo sea mayor o igual que 1.

La formulación de la SVM lineal con margen máximo en su *formulación original*, equivalente a la anterior, es la siguiente:

1. Minimizar $\frac{1}{2} < w, w >$.
2. Sujeto a: $y_i(< w, x_i > + b) \geq 1$ con $i \in [1,N]$.

Como puede observarse, se trata de un problema de optimización convexa, consistente en minimizar la función cuadrática bajo restricciones en forma de desigualdad lineal. En esta formulación original, el hiperplano queda caracterizado por un vector de pesos w con una componente por cada atributo que indica la importancia relativa de cada atributo en el hiperplano solución.

5.2.1.2 Formulación dual de las SMV.

Es posible obtener una *formulación dual* equivalente a la de la SVM de margen máximo en la que la solución se expresa como una combinación lineal de los vectores de aprendizaje. Su desarrollo se explica a continuación. Tenemos:

$$L(w,b,\alpha) = \frac{1}{2} < w, w > + \sum_{i=1}^{N} \alpha_i (1 - y_i(< w, x_i > + b)) \qquad \textbf{(5.7)}$$

Derivando con respecto a las variables originales:

$$\begin{aligned} &\frac{\partial L(w,b,\alpha)}{\partial w} = 0 \Rightarrow w - \sum_{i=i}^{N} y_i x_i \alpha_i = 0 \Rightarrow w = \sum_{i=i}^{N} y_i x_i \alpha_i \\ &\frac{\partial L(w,b,\alpha)}{\partial b} = 0 \Rightarrow \sum_{i=i}^{N} y_i \alpha_i = 0 \end{aligned} \qquad \textbf{(5.8)}$$

Sustituyendo estas dos últimas relaciones en la función de Lagrange generalizada, se obtiene:

$$\begin{aligned} &L(w,b,\alpha) = \frac{1}{2} < w, w > + \sum_{i=1}^{N} \alpha_i (1 - y_i(< w, x_i > + b)) = \\ &= \frac{1}{2} \sum_{i,j=1}^{N} y_i y_j \alpha_i \alpha_j < x_i, x_j > - \sum_{i,j=1}^{N} y_i y_j \alpha_i \alpha_j < x_i, x_j > + \sum_{i=1}^{N} \alpha_i = \\ &= \sum_{i=1}^{N} \alpha_i - \frac{1}{2} \sum_{i,j=1}^{N} y_i y_j \alpha_i \alpha_j < x_i, x_j > \end{aligned} \qquad \textbf{(5.9)}$$

TEOREMA (SVM LINEAL CON MARGEN MÁXIMO): Supongamos que el conjunto de datos es linealmente separable en el espacio de entrada. Sea a^* una solución del problema dual:

1. *Maximizar* $\sum_{i=1}^{N}\alpha_i - \frac{1}{2}\sum_{i,j=1}^{N} y_i y_j \alpha_i \alpha_j < x_i, x_j >$.
2. *Sujeto a* $\sum_{i=1}^{N} y_i \alpha_i = 0$, *con* $i \in [1,N]$. *Entonces el vector* $w^* = \sum_{i=1}^{N} y_i \alpha_i^* x_i$ *es el vector ortogonal al hiperplano con margen geométrico máximo. La SVM lineal con margen máximo es:*

$$h(x) = < w^*, x > + b^* = \sum_{i=1}^{N} y_i \alpha_i^* < x_i, x > + b^*;$$

donde $b^* = -\frac{1}{2}\left(\max_{y_j=-1}\{< w^*, x_j >\} + \min_{y_j=-1}\{< w^*, x_j >\}\right)$.

Una de las consecuencias importantes del teorema, expresada en la relación $w^* = \sum_{i=1}^{N} y_i \alpha_i^* x_i$ indica que el vector ortogonal al hiperplano se puede representar como una combinación lineal de N ejemplos del conjunto de datos.

5.2.1.3 Las condiciones de Karush-Kuhn-Tucker (KKT).

Las condiciones KKT juegan un papel central en la teoría y la práctica de la optimización obligada. Para el problema primario presentado en el punto anterior, las condiciones KKT quedan como:

$$\frac{\partial}{\partial w_v} L_P = w_v - \sum_i \alpha_i y_i x_{iv} = 0 \quad v = 1,\ldots,d \tag{5.10}$$

$$\frac{\partial}{\partial b} L_P = -\sum_i \alpha_i y_i = 0 \tag{5.11}$$

$$y_i\left(< x_i, w > + b\right) - 1 \geq 0 \quad i = 1,\ldots,N \tag{5.12}$$

$$\alpha_i \geq 0 \quad \forall i \tag{5.13}$$

$$\alpha_i\left(y_i\left(< w, x_i > + b\right) - 1\right) = 0 \quad \forall i \tag{5.14}$$

Las condiciones KKT se satisfacen en la solución de cualquier problema de optimización obligada, con cualquier clase de restricciones. El problema para las SVMs es convexo y para ese caso las condiciones KKT son necesarias y

suficientes para que w, b y α sean una solución. Por tanto resolver un problema de SVM es equivalente a encontrar una solución de las condiciones KKT.

Mientras que w está explícitamente determinado por el proceso de entrenamiento, el umbral b no lo está, aunque si está implícitamente determinada. Pero b se encuentra fácilmente usando la condición KKT complementaria (5.13), eligiendo cualquier i para el que $\alpha_i \neq 0$ y calcular b.

Lo que hemos conseguido es pasar el problema a un problema de optimización con restricciones más manejables. Encontrar la solución en un problema del mundo real requiere normalmente el empleo de métodos numéricos.

Por otro lado, la condición de Karush-Kuhn-Tucker, tras aplicar el teorema citado en el apartado anterior, se convierte en:

$$\alpha_i^*(1 - y_i(< w^*, x_i > + b^*)) = 0 \text{, para i} \in [1,N]. \qquad \textbf{(5.15)}$$

De esta relación se desprende que sólo algunas de las a^*_i son diferentes de cero. Por tanto, en realidad sólo algunos de los ejemplos del conjunto de datos forman parte de la combinación lineal que da lugar a w^*: son los llamados **vectores soporte.** Obsérvese que si todos los demás vectores fueran eliminados del conjunto de aprendizaje, la SVM lineal de margen máximo resultante sería exactamente la misma. Los vectores soporte son aquellos para los cuales $\alpha_i^* \neq 0$ y se cumple que están en la frontera de la región de decisión. Como consecuencia directa de la condición KKT, los vectores soporte tienen un margen funcional 1 con respecto a la SVM lineal con margen máximo. El resto de ejemplos tiene margen funcional mayor que 1, puesto que cumplen las condiciones del problema original.

Otra de las ventajas del teorema anteriormente comentado con respecto a la formulación original, tiene que ver con su aplicación práctica: aunque en teoría son equivalentes, el problema de optimización dual es más fácil de resolver, con las técnicas actualmente conocidas, que el del problema original. Las implementaciones actuales basadas en este teorema son muy eficientes.

La expresión de la solución final (como suma de los productos escalares con los vectores soporte) es una consecuencia de la manera en la que se resuelve el problema asociado, a diferencia de otros modelos cuyas soluciones también

se expresan como sumas de funciones simples (modelos clásicos de redes neuronales), en los que la expresión de la solución está prefijada a priori.

Una buena solución, especialmente para conjuntos de ejemplos voluminosos, es aquella en que la proporción de vectores soporte con respecto al número total de vectores *N* es pequeña. Sólo así se puede conseguir un factor de compresión elevado en la solución de la SVM (solución dispersa con respecto al número de vectores de aprendizaje). Nótese, sin embargo, que en la formulación original de las SVM no es posible controlar la dispersión de la solución de manera explícita.

5.2.1.4 Restricciones.

1. El clasificador resultante es lineal. Es bien conocido que la mejor manera de representar muchos problemas no es un modelo lineal.

2. En segundo lugar, necesita que el conjunto de datos sea linealmente separable, cosa que no tiene por qué ser cierta o fácil de conseguir. De hecho, aunque el problema sea lineal, la existencia de ruido puede hacer que no sea separable linealmente o que no sea conveniente separarlo por el hiperplano de margen máximo.

Las secciones que dedicamos a continuación sirven para explicar cómo eliminar estas restricciones manteniendo la filosofía subyacente a la SVM lineal con margen máximo.

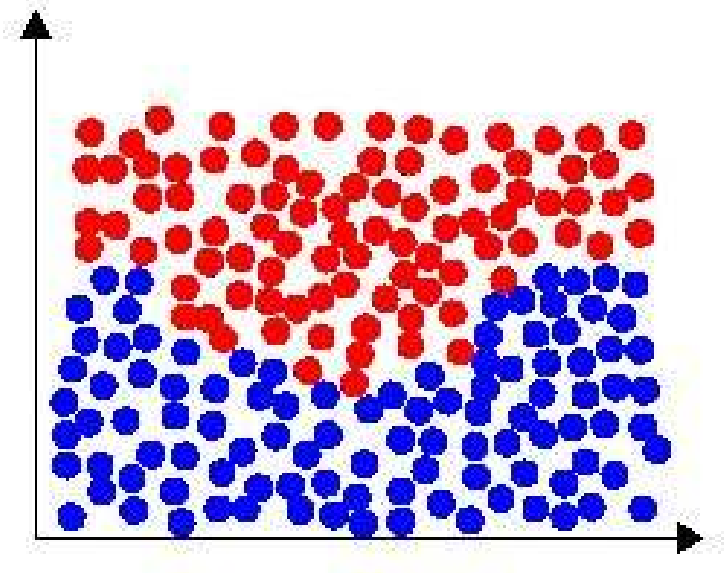

Figura 5.2. Conjunto no linealmente separable.

5.2.2 SVM con margen máximo en el espacio de características.

En la figura anterior se puede ver un conjunto de datos que no es linealmente separable en el que la SVM lineal con margen máximo no es la mejor solución. Para evitar la restricción de linealidad de la solución resultante, observemos que todo lo desarrollado en la sección anterior para la SVM lineal con margen máximo sólo depende de la existencia de un producto escalar en el espacio de entrada: hiperplanos, distancias, derivadas del producto escalar, etc.

La SVM no lineal con margen máximo en el espacio de características se basa en la idea de hacer una transformación no lineal del espacio de entrada a un espacio dotado de un producto escalar, fenómeno conocido en álgebra como espacio de Hilbert. En este espacio se pueden aplicar los mismos razonamientos que para la SVM lineal con margen máximo. Dicho de otro modo, supongamos que existe una transformación no lineal del espacio de entrada de un cierto *espacio de características J*:

$$\Phi : \Re D \rightarrow J$$
$$x \rightarrow \Phi(x)$$

dotado de un producto escalar $< \Phi(x), \Phi(y) >$. Si el conjunto de datos es linealmente separable en J (con los hiperplanos definidos a partir del producto escalar correspondiente), entonces la SVM con margen máximo en el espacio de características se puede obtener sustituyendo en la SVM lineal con margen máximo <*x*,*y*> por $< \Phi(x), \Phi(y) >$.

La dimensión del espacio de características necesaria para poder separar el conjunto de datos puede ser arbitrariamente grande. Pero al aumentar la dimensión de *J* también se incrementa el coste computacional de cualquier algoritmo que calcule el producto escalar operando directamente con las componentes $\Phi(x)$.

Afortunadamente, para ciertos espacios de características y ciertas transformaciones existe una forma muy efectiva de calcular el producto escalar usando las denominadas *funciones núcleo*. Una función núcleo es una función definida como $K : XxX \rightarrow \Re$ tal que $K(x, y) = < \Phi(x), \Phi(y) >$, donde Φ es una transformación de X en un cierto espacio de Hilbert J. Es decir, el producto escalar se puede calcular usando la función núcleo, quedando implícita la transformación del espacio de entrada en el espacio de características.

Supongamos que tenemos definido un producto escalar por medio de una función núcleo en su correspondiente espacio de características. Adaptando los mismos razonamientos de la SVM lineal con margen máximo, tenemos el siguiente resultado:

TEOREMA (SVM CON MARGEN MÁXIMO EN EL ESPACIO DE CARACTERÍSTICAS). Supongamos que el conjunto de datos es linealmente separable en el espacio de las características definido implícitamente por una función núcleo K(x,y). Sea a^* una solución del problema dual.

1. *Maximizar* $\sum_{i=1}^{N}\alpha_i - \frac{1}{2}\sum_{i,j=1}^{N} y_i y_j \alpha_i \alpha_j K(x_i, x_j)$.
2. *Sujeto a:* $\sum_{i=1}^{N} y_i \alpha_i = 0$, *con* $i \in [1,N]$. *Entonces el vector* $w^* = \sum_{i=1}^{N} y_i \alpha_i^* \Phi(x_i)$ *es el vector ortogonal al hiperplano con margen geométrico máximo en el espacio de características. La SVM lineal con margen máximo es:*

$$h(x) = < w^*, \Phi(x) > + b^* = \sum_{i=1}^{N} y_i \alpha_i^* K(x_i, x) + b^*;$$

donde $b^* = -\frac{1}{2}\left(\max_{y_j=-1}\{< w^*, \Phi(x_j) >\} + \min_{y_j=-1}\{< w^*, \Phi(x_j) >\}\right)$.

Observemos que todo lo comentado para la SVM lineal de margen máximo acerca de los vectores soporte sigue siendo válido ahora. En este caso, además, el número de vectores soporte afecta a la eficiencia en el cálculo de la solución resultante.

Como vemos, uno de los *parámetros* del modelo anterior es la función núcleo *K(x,y)*. A continuación enumeramos los *kernels* más usuales.

5.2.2.1 Algunas funciones núcleo.

Las funciones núcleo de propósito general más comúnmente utilizadas en **R**D son:

1. **Polinómica:** $K(x,y) = (< x, y > + c)^d$, con c∈**R** y d∈**N.**
2. **Gaussiana**: $K(x,y) = e^{-\|x-y\|^2/\gamma}$, con γ>0.
3. **Sigmoidal:** $K(x,y) = \tanh(s < x, y > + r)$, con s,r∈**R.**

4. **Multicuadrática inversa:** $K(x,y)=\frac{1}{\sqrt{\|x-y\|^2+c^2}}$, con c≥0

5. **Función de base radial (RBF):** $K(x,y)=(<x,y>+1)^p$

Para el caso del *RBF* (que es el núcleo que nosotros utilizamos en nuestro programa), el número de centros (N_S), los centros (s_i), los pesos (α_i), y el umbral (b) son producidos automáticamente por el entrenamiento de la SVM, dando excelentes resultados en comparación a las *RBF* clásicas.

Una de las grandes ventajas de las funciones núcleo es que su aplicación no está limitada a ejemplos de tipo vectorial sino que son aplicables a prácticamente cualquier tipo de representación. Dado que los datos de los problemas reales presentan habitualmente estructuras discretas, es posible y conveniente diseñar funciones núcleo específicas para estos casos concretos con el fin de capturar mejor las propiedades del problema. Ésta es, actualmente, una de las áreas activas de investigación.

5.2.2.2 Caracterización de las funciones núcleo.

Este apartado puede resultar algo técnico. El motivo de incluirlo aquí es el de hacer más autocontenido este proyecto. Pero omitir su lectura no impedirá la comprensión del resto de la teoría.

En espacios finitos, las funciones núcleo quedan caracterizadas por los valores propios de la matriz de productos escalares.

TEOREMA. Sea $\Delta=\{x_1,x_2,...,x_N\}$ *un conjunto finito. Una función* $K:\Delta x\Delta\to\Re$ *simétrica es una función núcleo en* Δ *si y sólo si la matriz* $K=(K(x_i,x_j))_{i,j=1}^N$ *es semidefinida positiva, es decir, tiene valores propios no negativos.*

El resultado en el que se basan las SVM con respecto a las funciones núcleo es debido a Mercer, desarrollado dentro del área del análisis funcional. El siguiente teorema es una versión simplificada del teorema de Mercer.

TEOREMA DE MERCER. Sea Δ *un subconjunto compacto de* $\mathbf{R}D$ *y* $K:\Delta x\Delta\to\Re$ *una función simétrica y continua tal que* $\forall f\in L_2(\Delta)\ \int K(x,y)f(x)f(y)dxdy\geq 0$. *Entonces*

K se puede expandir en una serie uniformemente convergente: $K(x,y)=\sum_j \lambda_j \Phi(x)_j \Phi(y)_j$ *con* $\lambda_j > 0$ *y* $\Phi_j \in L_2(\Delta)$.

Como consecuencia del teorema de Mercer, dada una función *K* que cumpla las condiciones de dicho teorema, existe un espacio de Hilbert *J* y una transformación $\Phi : \Re D \to J$ tal que $K(x,y) = <\Phi(x), \Phi(y)>$. La transformación del espacio de entrada en el espacio de características es $\Phi(x) = (\phi_1(x), \phi_2(x), ...)$, y el producto escalar *K* en *J* el expresado en el teorema.

Lo interesante de usar funciones núcleo con SVM es que el producto escalar se puede calcular implícitamente, sin conocer de manera explícita *J* ni la transformación Φ y evitando el coste computacional derivado de la posible alta dimensionalidad de *J*.

5.2.3 Métodos de solución

Un problema de optimización de vectores soporte puede resolverse analíticamente solo cuando el número de datos de entrenamiento es muy pequeño, o en el caso de que los datos sean separables y se conozcan de antemano cuales de los datos de entrenamiento son los vectores soporte.

Por ello en los casos del mundo real las ecuaciones suelen resolverse numéricamente. Para problemas pequeños, cualquier paquete de propósito general que resuelva programas lineales, cuadráticos, convexos y sujetos nos sirve.

Para problemas mayores existe un gran abanico de técnicas a usar. La receta básica es en primer lugar anotar las condiciones KKT que la solución óptima debe satisfacer, lo segundo es definir una estrategia para aproximarse a la solución óptima incrementado uniformemente la función objetivo dual sujeta a sus restricciones, y en tercer lugar decidir una descomposición del algoritmo de forma que solo una parte de los datos se maneje en cada momento. Algunos algoritmos ven el problema como una secuencia de igualdades, y usan el método de Newton para resolverlo. Otros algoritmos mueven un hiperplano dado hasta encontrar una nueva restricción, en ese momento añaden la restricción y reinician el proceso, por lo que solo se añade una restricción en cada paso. Otra posibilidad son los métodos de proyección, donde un punto fuera de la región factible es calculado, y entonces las búsquedas y las

proyecciones se hacen de forma que el movimiento se mantenga en la región factible, de esta forma pueden añadirse varias restricciones en un solo paso. En todos los algoritmos solo es necesario el cálculo de la parte esencial del hessiano, las columnas que se corresponden con $\alpha_i \neq 0$. Por último existen los métodos de puntos interiores, en los cuales las variables se van reescalando de forma que siempre se mantengan en la región factible.

5.3 CARACTERÍSTICAS Y LIMITACIONES DE LAS SVM.

A pesar de la aparente simplicidad de las ideas subyacentes a las SVM, los resultados teóricos que soportan la idea de las SVM y la maximización del margen son bastante complejos. Recordemos que el principal problema que se quiere resolver en un problema de aprendizaje supervisado es, a grandes rasgos, el de aproximar la función objetivo de manera que se obtenga el menor error posible en todo el espacio de entrada. Para ello sólo disponemos del valor de la función objetivo en un conjunto finito de puntos. La elección de la familia de funciones con la que queremos aproximar la función objetivo es uno de los puntos clave que diferencia las diversas familias de métodos de aprendizaje.

Uno de los resultados fundamentales de la teoría del aprendizaje estadístico acota el error de generalización por el error en el conjunto de datos más un término que depende de la complejidad de la familia de funciones con la que queremos aproximar la función objetivo. Dicha complejidad está relacionada con la capacidad de aproximar un conjunto aleatorio de puntos. Otro resultado importante acota la complejidad de la familia de los hiperplanos por el margen geométrico invertido. La heurística resultante de combinar los dos resultados anteriormente comentados intenta maximizar el margen para acotar la complejidad de la familia de funciones, que a su vez acota el error de generalización.

5.3.1 Limitaciones de las SVM.

1. La elección de un núcleo adecuado es todavía un área abierta de investigación. Una vez elegido el núcleo, los clasificadores basados en SVM tienen como único parámetro a ajustar por el usuario: la penalización del error *C*.

2. La complejidad temporal y espacial, tanto en el entrenamiento como en la evaluación, son también una limitación. Es un problema sin resolver el entrenamiento con grandes conjuntos de datos (del orden de millones de vectores soporte). Los algoritmos existentes para resolver esta familia de problemas tardan un tiempo que depende cuadráticamente del número de puntos.

3. Fue inicialmente creado para clasificación binaria. Aunque hay ya algunos trabajos que estudian el entrenamiento de SVM multiclase en

una sola pasada, aun se está lejos de diseñar un clasificador multiclase óptimo basado en SVM.

La SVM siempre soluciona un problema bloque: un cambio en los patrones de entrenamiento supone obtener una nueva SVM pues pueden surgir distintos vectores soporte (aunque existen ya algunas alternativas para el entrenamiento *on-line* de SVM capaces de encajar modificaciones del conjunto de datos sin necesidad de reentrenar el sistema).

5.3.2 Características.

1. El entrenamiento de una SVM es básicamente un problema de programación cuadrática (QP) convexa, que es atractivo por dos motivos:

   • Su eficiente computación (existen paquetes software que permiten su resolución eficientemente).

   • La garantía de encontrar un extremo global de la superficie de error (nunca alcanzará mínimos locales). La solución obtenida es única y la más óptima para los datos de entrenamiento dados.

2. A la vez que minimiza el error de clasificación en el entrenamiento.

3. La solución no depende de la estructura del planteamiento del problema.

4. Permite trabajar con relaciones no lineales entre los datos (genera funciones no lineales, mediante *kernel*). El producto escalar de los vectores transformados se puede sustituir por el *kernel* por lo que no es necesario trabajar en el espacio extendido.

5. Generaliza muy bien con pocas muestras de entrenamiento.

5.4 IMPLEMENTACIÓN DE UNA APLICACIÓN EN MATLAB.

Vamos a desarrollar el clasificador basado en SVM en el entorno MatLAB. Para ello haremos uso de un paquete de funciones de MatLAB ya desarrollado por Fernando Pérez-Cruz y usado en anterioridad por Sergio Palencia para la clasificación de quemaduras. Del mismo modo, estas funciones han sido revisadas y adaptadas a las necesidades específicas de este proyecto por B. Acha y C. Serrano.

Nuestra **situación de partida** es la que sigue. Ya hemos terminado nuestro análisis de la mamografía con el programa en C y recordemos que se generó un vector con las coordenadas de ciertos píxeles a los que llamamos *candidatos.* Se trata ahora, pues, de decidir si, finalmente, esos puntos van a ser considerados como *microcalcificaciones* o no. Para ello, extraemos unas determinadas características de dichos píxeles y sus alrededores y mediante una *red neuronal*, que entrenamos durante la ejecución del programa, analizamos dichas características y los clasificamos como microcalcificaciones (identificador *1*) o no (identificador *-1*).

5.4.1 Extracción de características.

Para cada candidato a microcalcificación extraemos características estadísticas relacionadas con la textura y éstas, tras un proceso de selección, serán pasadas a un clasificador. Más específicamente, los descriptores escogidos para cada candidato son:

1. Tamaño de los píxeles circundantes (k) donde el TR superaba al umbral.

2. Valor del TR.

3. Parámetro **ID** (del inglés *inter-distance*). Este parámetro se apoya en el hecho de que aquellos píxeles que tienen un elevado valor de intensidad (superior al 98%) pertenecerán a una microcalcificación. En tal caso, los píxeles deben estar juntos. Definimos, pues, ID como:

$$ID = \frac{1}{N}\sum_{i=1}^{N}\sqrt{(x_i - x_c)^2 + (y_i - y_c)^2} \qquad \textbf{(5.16)}$$

4. Parámetro **MS** (del inglés *mean slopes*). Se calcula el descenso medio de la pendiente desde el píxel de máximo valor (en los píxeles *kxk* de alrededor) en las cuatro posibles direcciones (norte, sur, este y oeste) y MS se obtiene como la media de acuerdo con la ecuación:

$$MS = \frac{1}{4}\cdot\frac{1}{(k/6)+1}\left(\sum_{n=n\max}^{n\max-k/6}\left[x(n-1,m)-x(n,m)\right]+\sum_{n=n\max}^{n\max+k/6}\left[x(n+1,m)-x(n,m)\right]+\right.$$
$$\left.+\sum_{m=m\max}^{m\max-k/6}\left[x(n,m-1)-x(n,m)\right]+\sum_{m=m\max}^{m\max+k/6}\left[x(n,m+1)-x(n,m)\right]\right) \quad \textbf{(5.17)}$$

donde *n*max y *m*max son las coordenadas del píxel con el valor máximo de aquéllos que se encuentran en el cuadrado *kxk*. El intervalo que hemos escogido para promediar la pendiente es *k/6* ya que el diámetro estimado de la microcalcificación es *k/3* en el cuadrado *kxk*.

5. Media de las pendientes máximas, **MaxS.** Esta característica representa la media de las máximas pendientes en las direcciones norte, sur, este y oeste, según:

$$MS = \frac{1}{4}\left(\max_{n\max\geq n\geq n\max-k/6}\left[x(n-1,m)-x(n,m)\right]+\max_{n\max\leq n\leq n\max+k/6}\left[x(n+1,m)-x(n,m)\right]+\right.$$
$$\left.+\max_{m\max\geq m\geq m\max-k/6}\left[x(n,m-1)-x(n,m)\right]+\max_{m\max\leq m\leq m\max+k/6}\left[x(n,m+1)-x(n,m)\right]\right) \quad \textbf{(5.18)}$$

6. Entropía, **Ent.** Se define como:

$$Ent = -\sum_{j=0}^{N-1} P(j)\log_2(P(j)) \quad \textbf{(5.19)}$$

donde *N* es el número de grises en la imagen y P(j) la probabilidad de ocurrencia del gris de valor j.

7. Altura media del histograma dentro del cuadrado *kxk*, **AH** (del inglés, *average height*). Dicho parámetro se define como:

$$AH = \frac{\sum_{j=0}^{N-1} h(j)}{\left(\max(X)-\min(X)\right)} \quad \textbf{(5.20)}$$

donde *h* representa el histograma de la distribución de datos *X* dentro del área *kxk*.

8. Correlación con la distribución gaussiana de desviación típica $\sigma=k/6$, ***CG*.** Aunque las microcalcificaciones varían en su forma, se considera como una suposición plausible que tienen una distribución gaussiana con simetría circular. Por consiguiente, si una microcalcificación está presente en la región cuadrada analizada, la correlación evaluada entre la gaussiana y la imagen centrada en el máximo dentro del cuadrado

estará cercana a 1 siempre que ambas señales tengan energía normalizada a 1.

9. Contraste, **C.**

$$C = \frac{mean_k - m}{mean_k + m} \quad \textbf{(5.21)}$$

donde $mean_k$ es el valor medio de los píxeles pertenecientes al cuadrado $k \times k$ y m representa el valor medio de los píxeles pertenecientes al borde del cuadrado (de anchura 2).

10. Rango dinámico, DR. Siendo su expresión:

$$DR = \max(X) - \min(X) \quad \textbf{(5.22)}$$

y X representa los valores de la imagen en el cuadrado kxk.

5.4.2 Selección de características.

No por usar más descriptores se obtiene una mejor clasificación. Normalmente el uso de muchos descriptores lo único que consigue es aumentar la complejidad del clasificador disminuyendo sus prestaciones. Por este motivo es importante realizar una buena selección de descriptores, de forma que se intentará obtener las mejores prestaciones del clasificador.

Tras analizar las características descritas anteriormente, resulta necesario aplicar un método de selección para obtener el conjunto óptimo de características para los sucesivos pasos de clasificación. Utilizando los métodos SFS y SBS (*Sequential Fordward Selection* y *Sequential Backward Selection*, respectivamente) vía una SVM, descrito en el apartado 5.1, ha sido como hemos discriminado y escogido las más importantes.

El SFS es un procedimiento de búsqueda de abajo a arriba donde una característica es añadida al conjunto. En cada etapa, la característica que va a ser incluida en el conjunto es seleccionada de las que aún quedan disponibles, es decir, aquéllas que no hayan sido incluidas en el conjunto todavía, siempre que el nuevo conjunto proporcione un error menor en comparación con el anterior conjunto. El algoritmo finaliza cuando añadir otra característica implica un incremento en el error de clasificación.

Por otro lado, el SBS es la versión homóloga del SFS pero, esta vez, de arriba abajo. Comienza con todas las características y, a cada paso, la característica que muestra el menor poder de discriminación es descartada. El algoritmo se para cuando eliminar cualquier otra característica implica incrementar un error en la clasificación.

El análisis de ambos métodos se llevó a cabo en un trabajo de investigación realizado por B. Acha, C. Serrano, R. M. Rangayyan y J. E. Desautels (véase referencia 14), donde se usaron cinco mamografías de las cuales se seleccionaron 630 puntos (230 microcalcificaciones y 400 puntos que no lo eran). De dichos puntos, 184 microcalcificaciones y 320 que no lo son, fueron usados en el entrenamiento del conjunto clasificador y las restantes 46 microcalcificaciones y 80 puntos singulares fueron utilizados como conjunto de prueba. Los resultados de la selección fueron evaluados por el método de validación cruzada (XVAL), siendo el proceso repetido cinco veces cambiando cada vez tanto el conjunto de entrenamiento como el de prueba, quedando la media de los resultados computados. De esta manera, la desventaja que suponía la baja sensibilidad de los métodos SBS y SFS al conjunto de entrenamiento queda reducida. Con el fin de implementar dichos métodos, se requiere un clasificador. Con este fin, se usó un SVM, cuyos fundamentos fueron explicados en el apartado anterior.

Los resultados de aplicar los métodos SFS y SBS quedan resumidos en:

Tabla 5.1. Resultados de los métodos SFS y SBS para selección de características

Método	Conjunto de características	Error medio
SFS	*MS*, *CG*, *C*, *DR*, *AH*, *ID*	4.13%
SBS	*TR*, *AH*, *CG*, *C*, *DR*	4.92%

El error medio fue calculado contando los errores de clasificación y dividiendo por el total de microcalcificaciones. De la tabla 5.1 podemos observar que el método SFS da el mejor conjunto de características (pendiente media, MS; correlación con una distribución gaussiana, C; contraste, C; rango dinámico, DR; altura media, AH; e ID), con un error medio del 4.13%.