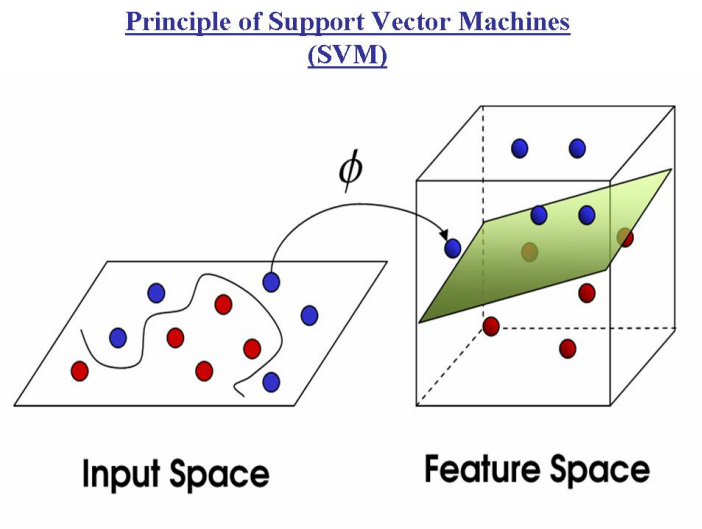
SUPPORT VECTOR MACHINES

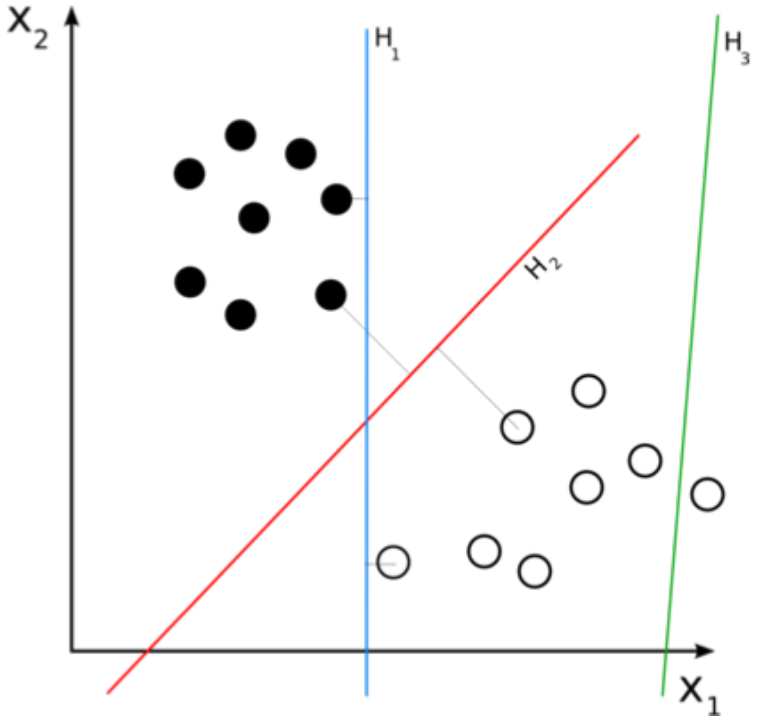
Inteligencia Artificial

MSc William Caicedo Torres



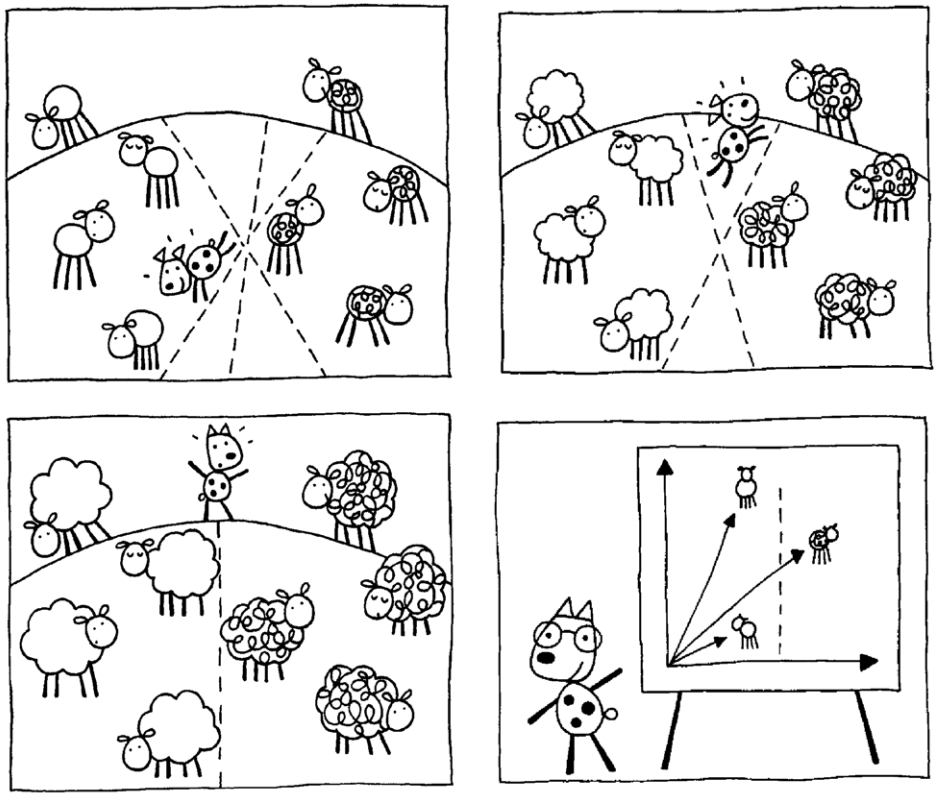
Support Vector Machines

- Para un conjunto de datos como el siguiente:



- ¿Cuantas fronteras de decisión separan las 2 clases perfectamente?

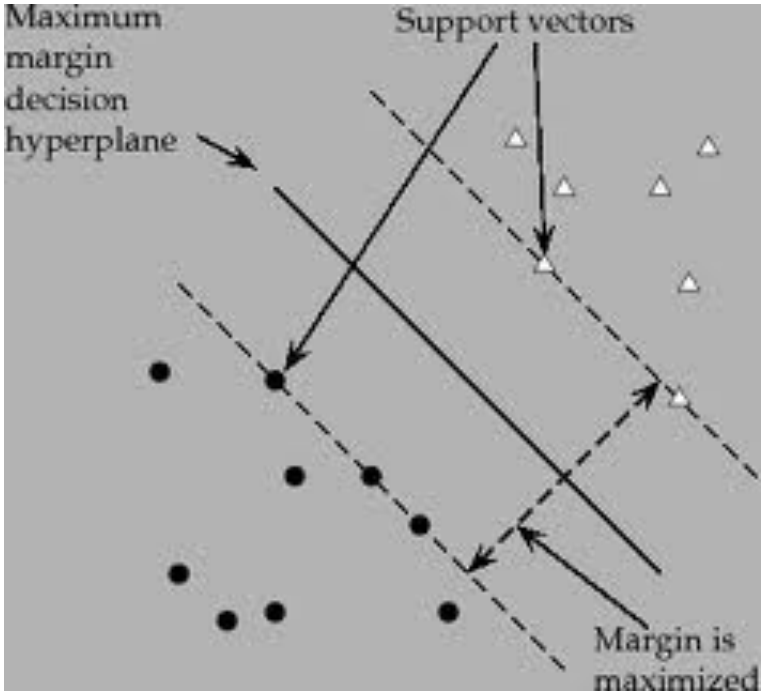
Support Vector Machines - Intuición



. . . the story of the sheep dog who was herding his sheep, and serendipitously invented both large margin classification and Sheep Vectors. . .

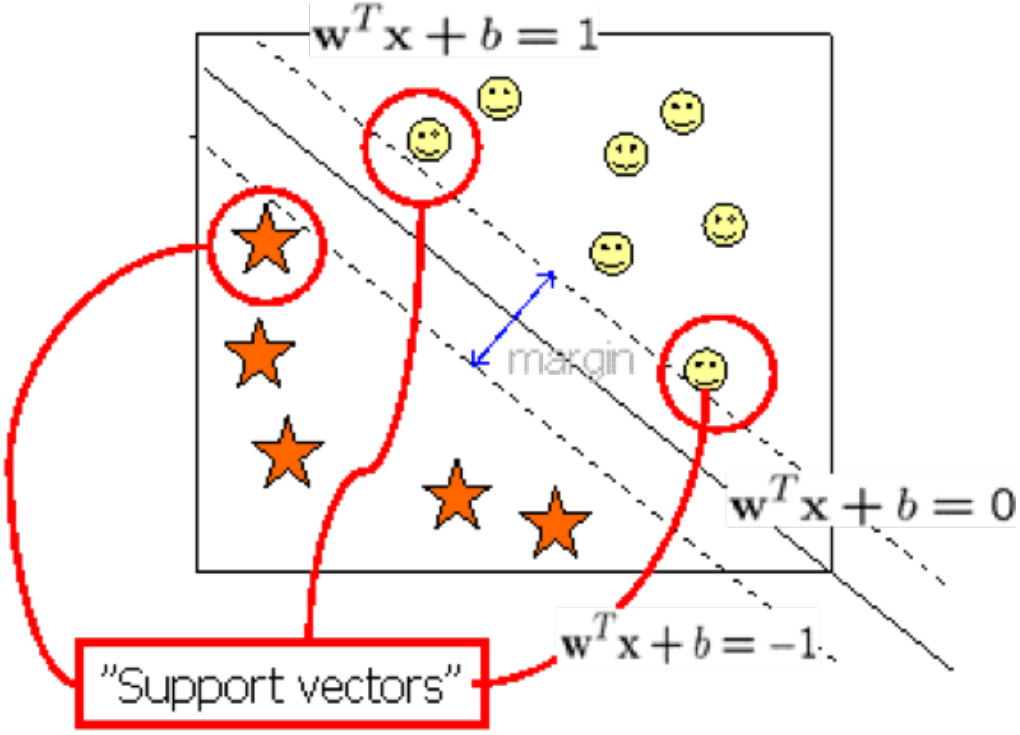
Support Vector Machines – El Hiperplano de Máximo Margen

- La función de costo de un SVM busca maximizar el margen – distancia entre la frontera de decisión y los puntos que se desea separar.
- De esta forma se puede obtener el hiperplano de máximo margen.



Support Vector Machines – Vectores de soporte

- Para encontrar el hiperplano de máximo margen, solo vamos a necesitar los puntos que yacen sobre el margen: Los vectores de soporte



Support Vector Machines – Maximizando el margen

- Cómo encontramos el hiperplano que maximice el margen M?
- El hiperplano (frontera) de decisión:

$$w^T x + b = 0$$

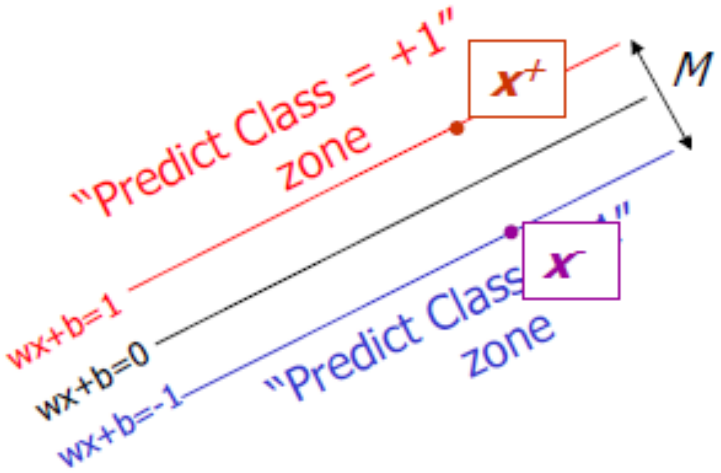
- Los hiperplanos que caen sobre el margen:

$$w^T x + b = 1$$
$$w^T x + b = -1$$

- Igual que en métodos anteriores, podemos especificar la frontera de decisión, a partir del vector w.
- Del algebra lineal, podemos recordar vector de pesos w es perpendicular al plano correspondiente.

Support Vector Machines – Maximizando el margen

- Ahora, para encontrar el hiperplano que maximice el margen, debemos encontrar una expresión que relacione w con m.



- Siendo x^+ un punto perteneciente a la clase positiva y x^- un punto perteneciente a la clase negativa, podemos decir que $x^+ = x^- + \lambda w$ (recordemos que los planos son paralelos y w es perpendicular a ellos).

Support Vector Machines – Maximizando el margen

- Con lo anterior, podemos computar el valor del margen.
- Tenemos un sistema 2 x 2:

$$x^+ = x^- + \lambda w \rightarrow |x^+ - x^-| = |\lambda w| = M$$

$$w \cdot x^+ + b = 1$$

$$w \cdot (x^- + \lambda w) + b = 1$$

$$w \cdot x^- + b + \lambda w \cdot w = 1$$

$$-1 + \lambda w \cdot w = 1$$

$$\lambda = \frac{2}{w \cdot w}$$

$$M = |\lambda w| = \lambda |w|$$

$$M = \frac{2\sqrt{w.w}}{w.w} = \frac{2}{\sqrt{w.w}}$$

$$M = \frac{2}{||w||}$$

Support Vector Machines – Maximizando el margen

- Ahora, solo queda buscar el vector w que maximice el margen M
- Cómo? Resolviendo el siguiente programa:

$$\min ||w||$$

$$s.a.$$

$$w \cdot x_i + b \geq 1, si\ y_i = 1$$

$$w \cdot x_i + b \leq -1, si\ y_i = 0$$

Support Vector Machines – Maximizando el margen

- La formulación anterior tiene un inconveniente: La norma del vector w involucra una raíz cuadrada: $||w|| = \sqrt{w.w}$. Esto hace el problema complicado de minimizar. Pero si elevamos la expresión al cuadrado obtenemos la siguiente formulación:

$$\min \frac{1}{2}||w||^2$$

$$s.a.$$

$$y_i(w \cdot x_i + b) \geq 1$$

- Esto convierte el problema en un problema de optimización cuadrática, y los algoritmos para resolver este tipo de problemas son maduros y bien conocidos. También redujimos las restricciones a una sola para facilitar los cálculos.

Support Vector Machines – Tomando la decisión

Para la predecir la clase a la que pertenece un punto de entrenamiento, procedemos a calcular el resultado de la siguiente expresión:

$$sign(w \cdot x_i + b)$$

Si el resultado es positivo, decimos que el SVM está prediciendo la clase positiva.

Support Vector Machines – Clases no linealmente separables

- La anterior formulación asume que los datos son linealmente separables. Qué sucede si no? No es suficiente con maximizar el margen, también habría que minimizar los errores de clasificación. Lo anterior convierte el problema de optimización en el siguiente:

$$min_{w,\xi} \ \frac{1}{2}||w||^2 + C\sum_{i=1}^{n}\xi_i$$

$$s.a.$$

$$y(w \cdot x_i + b) \geq 1 - \xi_i$$

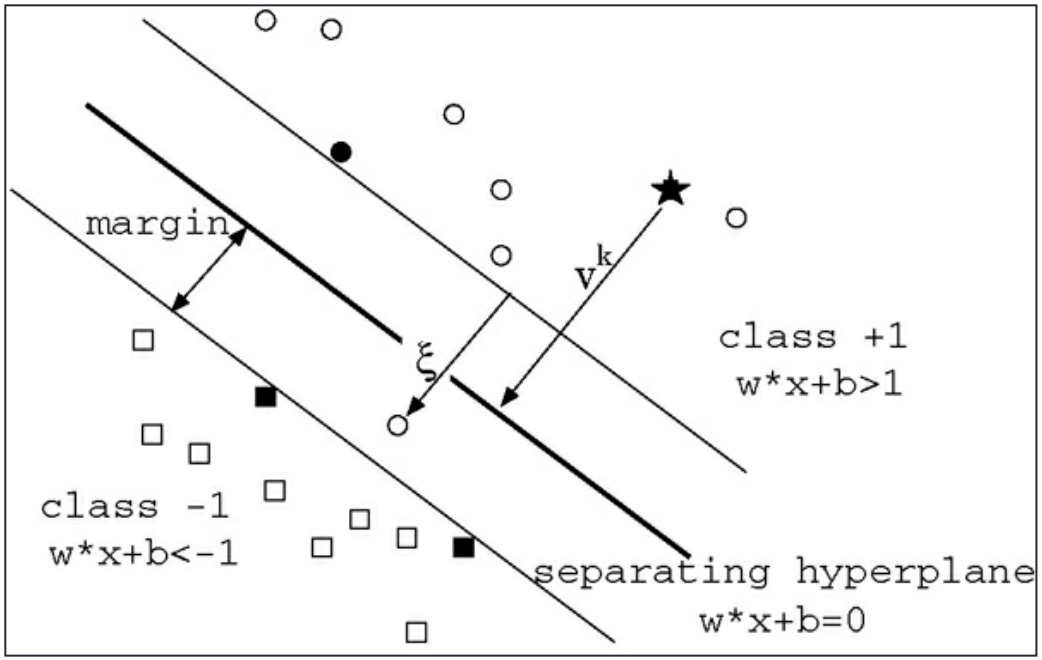
- Los ξ_i representan variables de holgura, que relajan las restricciones y aportan al objetivo.

Support Vector Machines – Clases no linealmente separables

$$min_{w,\xi} \ \frac{1}{2}||w||^2 + C\sum_{i=1}^{n}\xi_i$$

$$s.a.$$

$$y(w \cdot x_i + b) \geq 1 - \xi$$



Support Vector Machines – Dualidad Lagrangiana

- Existe un método bien conocido para resolver problemas de optimización no lineales sujetos a restricciones: El método de los multiplicadores de Lagrange.
- Consideremos un problema de optimización de la siguiente forma:

$$min_w\ f(w)$$
$$s.a.\quad h_i = 0, i = 1, \dots l$$

- Para resolverlo utilizando el método de los multiplicadores de Lagrange, definimos el Lagrangiano:

$$\Lambda(w, \beta) = f(w) + \sum_{i=1}^{l} \beta_i h_i(w)$$

Support Vector Machines – Dualidad Lagrangiana

- Hallando los puntos del Lagrangiano donde $\frac{\partial \Lambda}{\partial w_i} = 0$ y $\frac{\partial \Lambda}{\partial \beta_i} = 0$, obtendremos sus puntos críticos. La idea es que los puntos extremos de $f(w)$ corresponden a los puntos estacionarios de $\Lambda(w, \beta)$.
- El método de Lagrange se puede extender para problemas con restricciones tanto de desigualdad como de igualdad. En este caso, el problema a resolver (que en adelante llamaremos el problema primal), tendría la siguiente forma:

$$min_w \, f(w)$$
$$s.a.$$
$$g_i(w) \leq 0, i = 1, \dots k$$
$$h_i(w) = 0, i = 1, \dots l$$

Support Vector Machines – Dualidad Lagrangiana

- De acuerdo a esto, procedemos a definir el Lagrangiano generalizado (α, β son los multiplicadores de Lagrange):

$$\Lambda(w, \alpha, \beta) = f(w) + \sum_{i=1}^{l} \alpha_i g_i(w) + \sum_{i=1}^{l} \beta_i h_i(w)$$

- Definamos la siguiente cantidad:

$$\theta_{p(w)} = max_{\alpha,\beta:\alpha_i \geq 0} \Lambda(w, \alpha, \beta)$$

- El valor de esta cantidad esta dado de la siguiente manera:

$$\theta_{p(w)} = \begin{cases} f(w) \text{ si } w \text{ satisface las restricciones del primal} \\ \infty \text{ en caso contrario} \end{cases}$$

Support Vector Machines – Dualidad Lagrangiana

- Teniendo en cuenta que el máximo del Lagrangiano toma los mismos valores del primal cuando las restricciones se cumplen, podemos expresar el problema inicial de la siguiente forma:

$$min_w \theta_{p(w)} = min_w max_{\alpha,\beta:\alpha_i \geq 0} \Lambda(w, \alpha, \beta)$$

- Ahora, planteemos un problema ligeramente diferente. Definimos

$$\theta_{D(\alpha,\beta)} = min_w \Lambda(w, \alpha, \beta)$$

- Ahora estamos minimizando el Lagrangiano con respecto a w. A partir de esta expresión, podemos plantear el problema de optimización dual:

$$max_{\alpha,\beta:\alpha_i \geq 0} \theta_{D(\alpha,\beta)} = max_{\alpha,\beta:\alpha_i \geq 0} min_w \Lambda(w, \alpha, \beta)$$

Support Vector Machines – Relación entre el dual y el primal

- El problema primal y el dual están íntimamente relacionados. Se puede demostrar que:
  $$max_{\alpha,\beta:\alpha_i \geq 0} min_w \Lambda(w, \alpha, \beta) \leq min_w max_{\alpha,\beta:\alpha_i \geq 0} \Lambda(w, \alpha, \beta)$$
- Qué quiere decir esto? La solución del problema dual siempre será menor o igual que la solución del problema primal. En otras palabras, para una función determinada el máximo de los mínimos es estrictamente menor o igual al mínimo de los máximos.
- Bajo ciertas condiciones especiales, tendremos que la solución de ambos problemas es la misma. Y gracias a esto, podríamos resolver el problema dual para hallar la respuesta al problema primal.

Support Vector Machines – Condiciones Karush-Kuhn-Tucker (KKT)

- Supongamos que $f(w)$ y las $g(w)_i$ son convexas y que las $h(w)_i$ son afinas (lineales más un termino independiente). Además supongamos que las restricciones $g(w)_i$ son estrictamente factibles (es decir, existe algún w tal que $g(w)_i < 0$ para todo i).
- Asumiendo lo anterior, deben existir valores para w^*, α^*, β^* tal que w^* sea la solución al problema primal, α^*, β^* la solución al problema dual. Y además de esto la solución del problema primal y dual será igual a $\Lambda(w^*, \alpha^*, \beta^*)$. Adicionalmente w^*, α^*, β^* satisfacen las condiciones Karush-Kuhn-Tucker.

Support Vector Machines – Condiciones Karush-Kuhn-Tucker (KKT)

- Las condiciones Karush-Kuhn-Tucker son las siguientes:

$$\nabla\Lambda((w^*, \alpha^*, \beta^*) = 0, \qquad i = 1, \dots, n$$

$$\alpha^*{}_i g_i(w^*) = 0, \qquad i = 1, \dots, k$$

$$g_i(w^*) \leq 0, \qquad i = 1, \dots, k$$

$$\alpha^* \geq 0, \qquad i = 1, \dots, k$$

Si algún conjunto de valores para w^*, α^*, β^* satisfacen las condiciones KKT, también son solución para los problemas dual y primal.

Support Vector Machines – Formulación dual

- Recordemos la forma primal del problema de maximización del margen:

$$\min \frac{1}{2}||w||^2$$

$$s.a.$$

$$y_i(w \cdot x_i + b) \geq 1, i = 1, \ldots, m$$

- Podemos re-escribir las restricciones de la siguiente forma:

$$g_i(w) = -y_i(w \cdot x_i + b) + 1 \leq 0$$

- Cómo interpretamos una restricción igual a 0? Una restricción igual a cero nos habla de un punto de entrenamiento que cae exactamente sobre el margen. En ese caso, de acuerdo a una de las condiciones KKT ($\alpha^*{}_i g_i(w^*) = 0$), para su correspondiente multiplicador de Lagrange tendremos $\alpha_i > 0$. Mientras que si el punto no cae sobre el margen, su correspondiente multiplicador debe ser 0.

Support Vector Machines – Formulación dual

- Al construir el Lagrangiano para nuestro problema de optimización obtenemos:

$$\Lambda(w, b, \alpha) = \frac{1}{2}||w||^2 - \sum_{i=1}^{m} \alpha_i [y_i(w.x_i + b) - 1]$$

- Nótese que solo tenemos multiplicadores α_i, porque el problema original no tiene restricciones de igualdad.
- Ahora, para obtener la formulación dual del problema debemos encontrar las derivadas del Lagrangiano con respecto a las variables del primal e igualarlas a 0:

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

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

Support Vector Machines – Formulación dual

- Si reemplazamos w en el Lagrangiano por la expresión que acabamos de obtener para w , tendremos que:

$$\Lambda(w, b, \alpha) = \sum_{i=1}^{m} \alpha_i - \frac{1}{2} \sum_{i,j=1}^{m} y_i y_j \, \alpha_i \alpha_j \left(x_i . x_j\right) - b \sum_{i=1}^{m} \alpha_i \, y_i$$

- Además el último termino del Lagrangiano es igual a cero, en virtud de $\frac{\partial \Lambda(w,b,\alpha)}{\partial b} = \sum_{i=1}^{m} \alpha_i y_i = 0$, por lo que ahora obtenemos que

$$\Lambda(w, b, \alpha) = \sum_{i=1}^{m} \alpha_i - \frac{1}{2} \sum_{i,j=1}^{m} y_i y_j \, \alpha_i \alpha_j \left(x_i . x_j\right)$$

Support Vector Machines – Formulación dual

- De esta manera planteamos el problema dual como:

$$max_{\alpha} \sum_{i=1}^{m} \alpha_i - \frac{1}{2} \sum_{i,j=1}^{m} y_i y_j \, \alpha_i \alpha_j \langle x_i . x_j \rangle$$

$$s.a. \qquad \alpha_i \geq 0$$

$$\sum_{i=1}^{m} \alpha_i y_i = 0$$

- Supongamos que el SVM ya fue entrenado y queremos clasificar una nueva entrada x. Lo que haríamos sería calcular $w \cdot x + b$ y predecir la clase positiva si el resultado es mayor a cero. Podemos reemplazar w por el resultado que obtuvimos anteriormente y tendremos:

$$w \cdot x + b = \left(\sum_{i=1}^{m} \alpha_i y_i x_i \right) . x + b = \sum_{i=1}^{m} \alpha_i y_i \langle x_i . x \rangle + b$$

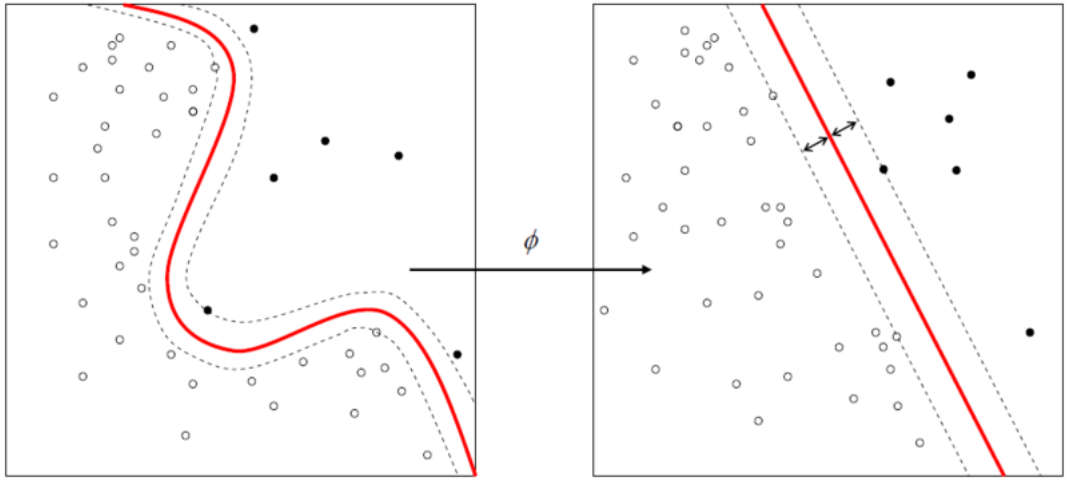
Support Vector Machines - Otras consideraciones

- La versión dual del programa formulado revela que el objetivo solo depende de los puntos que caen sobre el margen: los vectores de soporte.
- Si los datos no son linealmente separables: Introducir una penalidad por cada error cometido en la F.O. y variables de holgura en las restricciones. De esta manera el SVM encontrará la mejor frontera posible minimizando los errores de clasificación.
- Si los datos no son linealmente separables: Uso de KERNELS:

$$K(x, z) = \phi(x).\phi(z)$$

Support Vector Machines – Kernels

- Técnicamente un Kernel (Núcleo) es una función que representa una medida de similitud entre 2 vectores.
- En los SVM la introducción de un Kernel tiene como efecto un mapeo de las entradas a un espacio de alta dimensionalidad, donde los datos si serán linealmente separables (Kernel Trick).



Support Vector Machines – Ejemplos de Kernels

- "Lineal": $K(x,z) =< x.z >= x^T z$
- Polinomial: $K(x,z) = (x^T x + \tau)^d$
- Gaussiano: $K(x,z) = \exp\left(-\frac{||x-z||^2}{2\sigma^2}\right)$
- "Red Neuronal": $K(x,z) = \tanh(kx^T z + \tau)$

Preguntas??