Optimización II
Tarea IV

Joel Chacón Castillo

Guanajuato, México

1. Métodos de punto interior

Los métodos de punto interior pueden ser aplicados a problemas de programación cuadrática por medio de algoritmos lineales simples. En particular, el problema a tratar es de programación cuadrática convexa con desigualdades de restricción de la forma en que se indica a continuación:

$$min_x \quad q(x) = \frac{1}{2}x^T GX + x^T c \quad s.a. \quad Ax \geq b \tag{1}$$

donde G es una matriz simétrica positiva semidefinida ($x^T Gx \geq 0$), además $A\ [m \times n]$ y b corresponden a las restricciones de desigualdad. Las condiciones KKT son de la forma:

$$\begin{aligned} &Gx - A^T\lambda + c = 0 \\ &Ax - b \geq 0 \\ &(Ax - b_i)\lambda_i = 0 \quad \forall i \in 1, ..., m \\ &\lambda \geq 0 \end{aligned} \tag{2}$$

Agregando el vector slack s se tiene:

$$\begin{aligned} &Gx - A^T\lambda + c = 0 \\ &Ax - y - b \geq 0 \\ &y_i\lambda_i = 0 \quad \forall i \in 1, ..., m \\ &(y_i, \lambda) \geq 0 \end{aligned} \tag{3}$$

La medición de complementariedad μ es definido de la forma:

$$\mu = \frac{y^T\lambda}{m} \tag{4}$$

Además considerando las condiciones de KKT perturbadas:

$$F(x, y, \lambda; \sigma, \mu) = \begin{pmatrix} Gx - A^T\lambda + c \\ Ax - y - b \\ Y\Lambda e - \sigma\mu e \end{pmatrix} = \mathbf{0} \tag{5}$$

donde $Y = diag(y_1, .., y_m)$, $\Lambda = diag(\lambda_1, ..., \lambda_m)$, $e = (1, ..., 1)^T$, y $\sigma \in [0, 1]$, todos los valores positivos σ, μ definen la ruta central, que es la trayectoria en que la solución del problema cuadrático tiende a cero. Fijando el parámetro μ y aplicando el método de Newton se obtiene el sistema lineal siguiente:

$$\begin{bmatrix} G & 0 & -A^T \\ A & -I & 0 \\ 0 & \Lambda & Y \end{bmatrix} \begin{bmatrix} \Delta_x \\ \Delta_y \\ \Delta_\lambda \end{bmatrix} = \begin{bmatrix} -r_d \\ -r_p \\ -\Lambda Y e + \sigma\mu e \end{bmatrix} \tag{6}$$

donde $rd = Gx - A^T\lambda + c$ y $rp = Ax - y - b$, de forma iterativa se obtiene que $(x^+, y^+, \lambda^+) = (x, y, \lambda) + \alpha(\Delta_x, \Delta_y, \Delta_\lambda)$, donde se escoge un α la cual mantenga la desigualdad $(y^+, \lambda^+) > 0$

Algorithm 1 Predicto-Corrector Mehrotra - SQP

1: Cacular (x_0, y_0, λ_0) con $(y_0, \lambda_0) > 0$
2: **for** k=1,2,.., **do**
3: Resolver

$$\begin{bmatrix} G & 0 & -A^T \\ A & -I & 0 \\ 0 & \Lambda & Y \end{bmatrix} \begin{bmatrix} \Delta_x^{aff} \\ \Delta_y^{aff} \\ \Delta_\lambda^{aff} \end{bmatrix} = \begin{bmatrix} -r_d \\ -r_p \\ -\Lambda Y e \end{bmatrix} \tag{7}$$

donde $r_d = Gx - A^T\lambda + c$, $r_p = Ax - y - b$
4: Calcular $\mu = \frac{y^T\lambda}{m}$
5: $(x^{k+1}, \lambda^{k+1}, s^{k+1}) = (x^k, \lambda^k, s^k) + \alpha(\Delta_x, \Delta_\lambda, \Delta_s)$
6: Calcular $\alpha_{aff}^{pri}, \alpha_{aff}^{dual}$ y μ_{aff} de la siguiente forma

$$\begin{aligned} \alpha_{aff}^{pri} &= min(1, min_{i:\Delta y_i^{aff}<0} - \frac{y_i}{\Delta y_i^{aff}}) \\ \alpha_{aff}^{dual} &= min(1, min_{i:\Delta \lambda_i^{aff}<0} - \frac{\lambda_i}{\Delta \lambda_i^{aff}}) \\ \alpha &= max(\alpha_{aff}^{pri}, \alpha_{aff}^{dual}) \\ \mu_{aff} &= (y + \alpha\Delta y^{aff})^T(\lambda + \alpha\Delta\lambda^{aff})/n \\ \sigma &= (\frac{\mu_{aff}}{\mu})^3 \end{aligned} \tag{8}$$

7: Resolver

$$\begin{bmatrix} G & 0 & -A^T \\ A & -I & 0 \\ 0 & \Lambda & Y \end{bmatrix} \begin{bmatrix} \Delta_x \\ \Delta_y \\ \Delta_\lambda \end{bmatrix} = \begin{bmatrix} -r_d \\ -r_p \\ -\Lambda Y e - \Delta\Lambda^{aff}\Delta Y^{aff} e + \sigma\mu e \end{bmatrix} \tag{9}$$

donde $r_d = Gx - A^T\lambda + c$, $r_p = Ax - y - b$
8: Calcular $\alpha_{max}^{pri} = min_{i:\Delta y_i<0} - \frac{y_i - (1-\tau)y_i}{\Delta y_i}$ y $\alpha_{max}^{dual} = min_{i:\Delta\lambda_i<0} - \frac{\lambda_i - (1-\tau)\lambda_i}{\Delta\lambda_i}$
9: Asignar $\alpha^{pri} = min(1, \alpha_{max}^{pri})$ y $\alpha^{dual} = min(1, \alpha_{max}^{dual})$
10: Asignar $\alpha = min(\alpha^{pri}, \alpha^{dual})$
11: Actualizar $x = x + \alpha\Delta x$ y $(\lambda, s) = (\lambda, s) + \alpha(\Delta\lambda, \Delta s)$

Dado los puntos arbitratios (x_0, y_0, λ_0) el punto inicial es calculado como se indica a continuación:

$$y_0 = max(1, |\hat{(}y) + \Delta y^{aff}|, \quad \lambda_0 = max(1, |\hat{(}\lambda) + \Delta\lambda^{aff}|, \quad X_0 = \hat{(}x) \tag{10}$$

2. Máquina de soporte vectorial para dator linealmente separables

El algortimo de máquinas de soporte vectoria (SVM) es un problema de optimización utilizado para la clasificación de datos y es planteado como se indica a continuación:

$$min_{\omega,\omega_0} \frac{1}{2}||\omega||^2 \quad s.a. \quad y_i(\omega^T x_i + \omega_0) \geq 1 \tag{11}$$

2.1. *Planteamiento primal del problema SVM para datos linealmente separables*

Para el planteamiento del problema primal la forma del problema debe ser la indicada en la ecuación 1, esto se indica a continuación:

$$
\begin{aligned}
& min_{\omega,\omega_0} \frac{1}{2}[\omega_1, ..., \omega_n, \omega_0] \begin{bmatrix} I & 0 \\ 0 & \epsilon \end{bmatrix} [\omega_1, ..., \omega_n, \omega_0]^T - [\omega_1, ..., \omega_n, \omega_0][0, .., 0]^T \\
& s.a. \\
& \begin{bmatrix} y_1 x_1 & y_1 \\ .. & . \\ y_m x_m & y_m \end{bmatrix} [\omega_1, ..., \omega_n, \omega_0]^T \succeq 1
\end{aligned} \tag{12}
$$

donde ϵ es un valor muy pequeño con la finalidad de que G sea no singular, es decir se agrega un valor para que sea una matriz definida positiva, esto es necesario ya que al ensamblar la matriz del funcional se debe resolver el sistema. Es importante mencionar que resolver este problema de forma directa se considera como difícil debido a que las restricciones de desigualdad son muy complejas. Por lo tanto se ha optado por resolver su forma dual [1].

2.2. *Forma dual del problema SVM para datos linealmente separables*

$$
\begin{aligned}
\mathscr{L}(\lambda, \omega, \omega_0) &= \frac{1}{2}\omega^T\omega - \sum_{i=1}^{m} \lambda_i (y_i(\omega^T x_i + \omega_0) - 1) \\
\nabla_{\omega_0}\mathscr{L}(\lambda, \omega, \omega_0) &= \sum_{i=1}^{m} \lambda_i (y_i) = 0 \\
\nabla_{\omega}\mathscr{L}(\lambda, \omega, \omega_0) &= \omega^T - \sum_{i=1}^{m} \lambda_i y_i x_i = 0 \longrightarrow \omega = \sum_{i=1}^{m} \lambda_i y_i x_i
\end{aligned} \tag{13}
$$

$$
\begin{aligned}
\mathscr{L}(\lambda, \omega, \omega_0) &= \frac{1}{2}(\sum_{i=1}^{m} \lambda_i y_i x_i)^T (\sum_{j=1}^{m} \lambda_j y_j x_j) - \sum_{i=1}^{m} \lambda_i y_i \omega^T x_i - \sum_{i=1}^{m} \lambda_i y_i \omega_0 + \sum_{i=1}^{m} \lambda_i \\
&= \frac{1}{2}(\sum_{i=1}^{m} \lambda_i y_i x_i)^T (\sum_{j=1}^{m} \lambda_j y_j x_j) - \sum_{i=1}^{m} \lambda_i y_i (\sum_{j=1}^{m} \lambda_j y_j x_j)^T x_i + \sum_{i=1}^{m} \lambda_i \\
&= \frac{1}{2}(\sum_{i=1}^{m} \sum_{j=1}^{m} \lambda_i y_i \lambda_j y_j (x_i^T x_j)) - \sum_{i=1}^{m} \sum_{j=1}^{m} \lambda_i y_i \lambda_j y_j (x_j^T x_i) + \sum_{i=1}^{m} \lambda_i \\
&= -\frac{1}{2}(\sum_{i=1}^{m} \sum_{j=1}^{m} \lambda_i \lambda_j y_i y_j (x_i)^T (x_j) + \sum_{i=1}^{m} \lambda_i
\end{aligned}
\tag{14}
$$

Por lo tanto, la forma dual del problema SVM es como se indica a continuación:

$$
\begin{aligned}
& max_\alpha \sum_{i=1}^{m} \alpha_i - \frac{1}{2} \sum_{i,j=1}^{m} y_i \alpha_i y_j \alpha_j x_i^T x_j \\
& s.a. \\
& \alpha_i \geq 0 \quad i \in 1..m \\
& \sum_i y_i \alpha_i = 0
\end{aligned}
\tag{15}
$$

Este problema es computacionalmente más sencillo debido a que sus restricciones son más sencillas. La dirección ω^* del hyperplano puede ser recuperada de una solución α^* dado el problema dual de la siguiente forma:

$$
\omega^* = \sum_i \alpha_i^* y_i x_i \tag{16}
$$

Determinar el sesgo (bias) b^* se convierte en un problema unidimensional. Por lo tanto la función del discriminante lineal puede ser escrito de la siguiente forma:

$$
\begin{aligned}
\hat{y}(x) = \omega^{*T} x + b^* &= \sum_{i=1}^{n} y_i \alpha_i^* x_i^T X + b^* \\
1 = y_i(\omega^{*T} x + b^*) &= y_i(\sum_{i=1}^{n} y_i \alpha_i^* x_i^T X + b^*)
\end{aligned}
\tag{17}
$$

Aunque se puede obtener un sesgo b resolviendo esta ecuación para una X albitraria, una solución numérica y más estable es multiplicar primeramente por y_i obligando que $y_i^2 = 1$ y luego promediando las ecuaciones sobre todos los vectores de soporte y resolviendo para b [2].

$$b = \frac{1}{m}\sum_{i=1}^{m}(y_i - \sum_{j=1}^{m}\alpha_j y_j x_n^T x_j) \tag{18}$$

La adecuación estándar de la ecuación (1) del problema dual es de la siguiente forma:

$$\begin{aligned} & min_\lambda \frac{1}{2}[\alpha_1, ..., \alpha_m, S_1] \begin{bmatrix} y_1y_1(x_1^Tx_1) & .. & y_1y_m(x_1^Tx_m) & 0 \\ ..... & .... & .... & 0 \\ y_my_1(x_m^Tx_1) & .. & y_my_m(x_m^Tx_m) & 0 \\ 0 & 0 & 0 & \epsilon \end{bmatrix} [\alpha_1, ..., \alpha_m, S_1]^T \\ & - [1, 1, .., 1, 0]^T[\alpha_1, \alpha_2, ..., \alpha_m, S_1] \\ & s.a. \\ & \begin{bmatrix} y_1 & ... & y_m & 1 \\ & I & & \end{bmatrix} \begin{bmatrix} \alpha_1 \\ ... \\ \alpha_m \\ S_1 \end{bmatrix} \geq \mathbf{0} \end{aligned} \tag{19}$$

Es importante hacer mención que se debe considerar la matriz identidad en las restricciones para asegurar no negatividad en las variables.

3. Máquina de soporte vectorial para datos no linealmente separables

Los problemas de optimización y además las función discriminante consideran el cálculo de productos punto (producto interno), es decir se realiza un mapeo implítico al espacio de productos punto (identificado como espacio de Hilbert) [3]. Esto quiere decir que no es necesario tener los vectores de características X, únicamente se requiere el producto punto de los vectores. En su lugar se ha propuesto utilizar funciones de kernel (truco kernel) que representa el producto punto en algún espacio de características de alta dimensionalidad. Hasta ahora se ha asumido que los datos de entrenamiento son linealmente separables en el espacio de las características. Sin embargo

esto no sucede en la mayoría de los casos, principalmente cuando la frontera de desición no es lineal, por lo tanto se desea modificar SVM para que se puedan cometer algunos erroes en el entrenamiento de los datos. Para realizar esto se introduce una variable *slack* $\xi \geq 0$. Estos están definidos para los datos que están en el interior del margen de la frontera y $\xi = |y_i - \hat{(y)}(x_i)|$ para otros casos. Por lo tanto los datos que se encuentran en la frontera de decisión $y_i = 0$ tendrán $\xi = 1$ y los puntos con $\xi > 1$ serán mal clasificados como se puede observar en la figura 1.

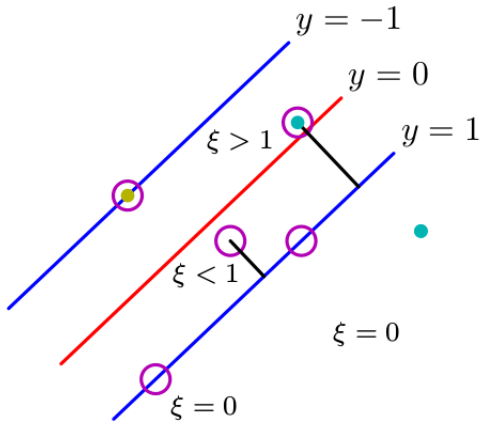

Figura 1: Ilustración de las variables *Slack* $\xi \geq 0$, los puntos dentro de los círculos son los vectores de soporte.

Por lo tanto la nueva formulación del problema primal se define a continuación:

$$
\begin{aligned}
& min_{\omega,\omega_o,\xi} J(\omega, \omega_o, \xi) = \frac{1}{2}||\omega||^2 + C\sum_{i=1}^{N}\xi \\
& s.a. \\
& y_i(\omega^T x_i + \omega_0) \geq 1 - \xi_i, \quad \forall i \in \{1, .., m\} \\
& \xi \geq 0 \quad \forall i \in \{1, .., m\}
\end{aligned}
\tag{20}
$$

donde $C > 0$ es un parámetro de penalización y controla la compensación entre la variable de penalización *slack* y el margen. Este enfoque es descrito

como una relajación y suaviza el margen, esto permite que algunos datos estén mal clasificados. Es importante notar que este enfoque es sensible a los datos atípicos ya que la clasificación errónea incrementa de forma lineal con ξ. Además, el parámetro C es análogo (el inverso)los coeficientes de regularización, y además en el límite $C \to \infty$ se recupera el problema linealmente separable de las máquinas de soporte vectorial.

El lagrangiano de la forma primal (ecuación 20) se indica a continuación:

$$\mathscr{L}(\lambda, \omega, \omega_0) = \frac{1}{2}||\omega||^2 + C\sum_{i=1}^{m}\xi_i - \sum_{i=1}^{m}\lambda_i(y_i\omega^T x_i + y_i\omega_0 - 1 + \xi_i) - \sum_{i=1}^{m}\xi_i S_i \tag{21}$$

donde

$$\begin{aligned} \lambda_i &\geq 0 \\ y_i(\omega^T x + \omega_0) - 1 + \xi_i &\geq 0 \\ \lambda_i(y_i(\omega^T x + \omega_0) - 1 + \xi) &= 0 \\ s_i &\geq 0 \\ \xi_i &\geq 0 \\ s_i\xi_i &= 0 \end{aligned} \tag{22}$$

Optimizando el lagrangiano con respecto a ω, b y ξ se tiene lo siguiente:

$$\begin{aligned} \nabla_\omega \mathscr{L}(\lambda, \omega, \omega_0, \xi) &= \omega - \sum_{i=1}^{m}\lambda_i y_i x_i \to \omega = \sum_{i=1}^{m}\lambda_i y_i x_i \\ \nabla_{\omega 0} \mathscr{L}(\lambda, \omega, \omega_0, \xi) &= \sum_{i=1}^{m}\lambda_i y_i = 0 \\ \nabla_{\xi i} \mathscr{L}(\lambda, \omega, \omega_0, \xi) &= \lambda_i - C + S_i \to \lambda_i = C - S_i \end{aligned} \tag{23}$$

Utilizando estos valores en el lagrangiano del caso de la ecuación (20) se tiene el lagrangiano de la forma

$$\begin{aligned} & max_\alpha \sum_{i=1}^{m}\alpha_i - \frac{1}{2}\sum_{i,j=1}^{m} y_i\alpha_i y_j\alpha_j x_i^T x_j \\ & s.a. \\ & C \geq \alpha_i \geq 0 \quad i \in 1..m \\ & \sum_i y_i\alpha_i = 0 \end{aligned} \tag{24}$$

Esta forma es similar al caso separable, las diferencias se pueden observar en las restricciones, además se requiere que $\alpha_i \geq 0$ ya que son los multiplicadores, además con $s_i \geq 0$ se implica que $C \geq \lambda_i$, estas restricciones también son conocidas como restricciones de caja. Si $\lambda < C$ entonces $s_i > 0$ que por la condición de complementariedad se tiene $\xi_i = 0$ y por lo tanto los puntos están en el margen. Los puntos con $\lambda_i = C$ están ubicados dentro del margen y pueden ser correctamente clasificados si $\xi_i < 1$ o mal clasificados si $\xi_i > 1$. Para determinar el sesgo b se nota que los vectores en $0 < \lambda_i < C$ tienen $\xi_i = 0$ por lo tanto $y_i(\omega^T x + \omega_0) = 1$ y cumplirán $y_i(\sum_j (\omega^T x + \omega_0)x_j + b) = 1$. Dejando a la solución que es numéricamente estable:

$$b = \frac{1}{m}\sum_{i=i}^{m}(y_i - \sum_{j=1}\lambda_j y_j x_j^t x_i) \tag{25}$$

donde m es el número de datos que están dentro del margen $0 < \lambda_i < C$

La adecuación estándar del problema primal considerando un margen se indica a continuación:

$$min_{\omega,\omega_0}\frac{1}{2}[\omega_1,...,\omega_n,\omega_0.\xi]\begin{bmatrix} I_\omega & 0 & 0 \\ 0 & \epsilon_{\omega_0} & 0 \\ 0 & 0 & \xi \end{bmatrix}[\omega_1,...,\omega_n,\omega_0,\xi]^T - [\omega_1,...,\omega_n,\omega_0,\xi][0,..,0,C]^T$$

$$s.a.$$

$$\begin{bmatrix} y_1x_1 & y_1 & \xi \\ .. & . & \\ y_mx_m & y_m & \xi \\ & I & \end{bmatrix}[\omega_1,...,\omega_n,\omega_0,\xi]^T \succeq \begin{bmatrix} 1 \\ .. \\ 1 \\ \mathbf{0} \end{bmatrix} \tag{26}$$

Es importante considerar las restricciones de no negatividad de la variable ξ. La adecuación estándar del problema dual considerando un margen se indica a continuación:

$$
\begin{aligned}
min_{\alpha} & \frac{1}{2}[\alpha_1, ..., \alpha_m, S_1] \begin{bmatrix} y_1 y_1 (x_1^T x_1) & .. & y_1 y_m (x_1^T x_m) & 0 \\ ..... & .... & .... & 0 \\ y_m y_1 (x_m^T x_1) & .. & y_m y_m (x_m^T x_m) & 0 \\ 0 & 0 & 0 & \epsilon \end{bmatrix} [\alpha_1, ..., \alpha_m, S_1]^T \\
& - [1, 1, .., 1, 0]^T [\alpha_1, \alpha_2, ..., \alpha_m, S_1] \\
& s.a. \\
& \begin{bmatrix} y_1 & ... & y_m & 1 \\ 0 & & 0 & 1 \\ & -I & 0 & 0 \\ & I & & 0 \end{bmatrix} \begin{bmatrix} \alpha_1 \\ ... \\ \alpha_m \\ S_1 \end{bmatrix} \geq \begin{bmatrix} 0 \\ 0 \\ -C \\ .. \\ 0 \end{bmatrix}
\end{aligned} \tag{27}
$$

En base a varios experimentos se concluyó que un valor apropiado es $C =$ 0,1, es importante mencionar que si no se agregan las condiciones de no negatividad $\alpha_i \geq 0$ no existe convergencia en el algoritmo de punto interior.

Una formulación alternativa de SVM propuesta por Scholkopf and Smola en [4] conocido como v-SVM se indica a continaución:

$$
\begin{aligned}
max_{\alpha} & -\frac{1}{2} \sum_{i=1}^{m} \sum_{j=1}^{m} \alpha_i \alpha_j y_i y_j (x_i^T x_j) \\
& s.a. \\
& 0 \leq \alpha_i \leq 1/m \\
& \sum_{i=1}^{m} {}_i y_i = 0 \\
& \sum_{i=1}^{m} \alpha_i \geq V
\end{aligned} \tag{28}
$$

Este enfoque reemplaza el parámetro C con V y puede ser interpretado como una cota superior en la fracción de errores que pertenecen al margen.

3.1. Resultados de la versión primal y dual para los datos linealmente separables

Se implementó el algoritmo de punto interior para resolver el problema de máquina de soporte vectorial, el número de datos generados fue de 200

	Primal	Dual
Tiempo	0.127532958984	0.201560974121
Iteraciones	24	25
Función objetivo	0.0308108612753	0.068942510487

Cuadro 1: Resultados del problema primal y dual con datos que son linealmente separables

(N=200), se utilizó la siguiente matriz de covarianza $\Sigma = \begin{bmatrix} 0{,}1 & 0{,}5 \\ 1 & 1 \end{bmatrix}$, los vectores de media utilizados son $\mu_1 = [-5, -5]$ y $\mu_2 = [5, 5]$, la tolerancia fue asignado de acuerdo a la norma del gradiente, el cual es asignado a $1e-10$

Es importante mencionar que en la práctica se prefiere utilizar la forma dual de este problema pues es mas sencillo de resolver, sin embargo existe una complejidad computacional elevada pues en algunos casos se debe calcular la matriz de Grahm compuesta por el producto punto entre los datos, sin embargo existen aplicaciones donde esta matriz es ralas.

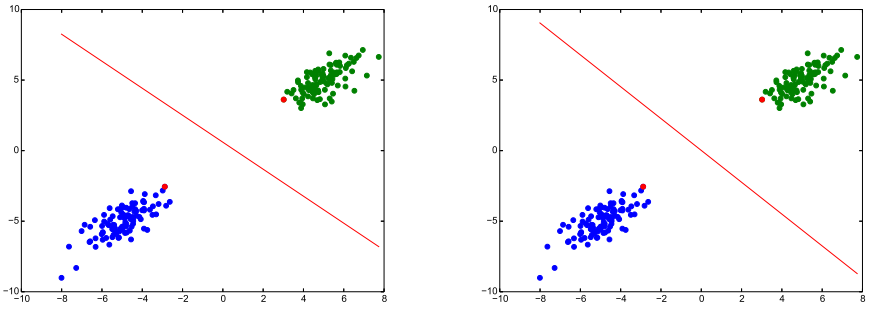

Figura 2: Ilustración primal y dual con datos linealmente separables y con la misma semilla para el generados de números.

3.2. Resultados de la versión primal y dual para los datos que no son linealmente separables, en color rojo se indican los vectores de soporte.

Se implementó el algoritmo de punto interior para resolver el problema de máquina de soporte vectorial, el número de datos generados fue de 200 (N=200), se utilizó la siguiente matriz de covarianza $\Sigma = \begin{bmatrix} 0{,}5 & 0 \\ 0 & 0{,}5 \end{bmatrix}$, los vectores de media utilizados son $\mu_1 = [-1, -1]$ y $\mu_2 = [1, 1]$, la tolerancia fue asignado de acuerdo a la norma del gradiente, el cual es asignado a $1e-10$.

	Primal	Dual
Tiempo	0.492672920227	1.96275496483
Iteraciones	30	26
Función objetivo	2.514257082982.51425708298	1.4145959812

Cuadro 2: Resultado de SVM primal y dual con datos no linealmente separables.

En la tabla 2 se puede apreciar que el método dual requiere de un menor número de iteraciones y proporciona un mejor valor de la función objetivo, es importante notar que el tiempo es mayor, esto puede suceder por la matriz G que utiliza la matriz de Grahm.

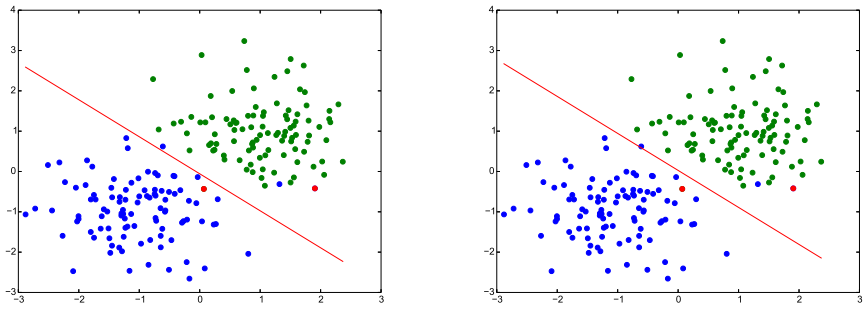

Figura 3: Ilustración primal y dual con datos linealmente no separables y con la misma semilla para el generador de números aleatorios, con el parámetro de penalización $C = 0{,}1$.

4. Mis experimentos

En este apartado se realiza una serie de experimentos para comprender a produndidad el mapeo implícito que se realiza al aplicar al truco kernel. Primeramente, se plantea el problema de clasificación con SVM y comprender mas a detalle las implicaciones de utilizar un kernel con grado d. El punto de inicio para *Suppor Vectorial Machines - SVM* es minimizar ecuación (aquí se pone la notación que se utiliza en la lectura recomendada [5]):

$$\boldsymbol{\omega}^* = min_{\boldsymbol{\omega}}\{c\sum_i(1 - y_i\boldsymbol{\omega}\cdot\boldsymbol{x_i})_+ + \frac{1}{2}||\boldsymbol{\omega}||^2\} \tag{29}$$

Minimizar esta ecuación implica ajustar un model con SVM. La principal diferencia con los kernels es que se aplica una optimización de una forma distinta que permite trabajar de forma eficiente con funciones kernel.

Como se menciona anteriormente se ha probado que si w^* es el vector de pesos que resulta de este proceso de optimización, entonces se puede calcular de forma alternativa w^* de la siguiente forma:

$$\boldsymbol{w}^* = \sum_i \alpha_i y_i \boldsymbol{x_i} \tag{30}$$

donde α_i representa la optimización de:

$$max_{0 \leq \boldsymbol{\alpha} \leq c} \sum_i \alpha_i - \frac{1}{2} \sum_i \sum_j \alpha_i \alpha_j y_i y_j \boldsymbol{x_i} \cdot \boldsymbol{x_j} \tag{31}$$

Una vez resuelto este problema de optimización es posible resolver el problema de clasificar a un punto nuevo por:

$$f(\boldsymbol{x}) = \boldsymbol{x} \cdot \boldsymbol{w}^* = \sum_i \alpha_i y_i \boldsymbol{X} \cdot \boldsymbol{x_i} \tag{32}$$

De esta forma sólo es necesario enfocarse en optimizar las variables α_i dado que son las que se están ajustando.

Al final se puede reemplazar el producto punto con kernels resultando en el problema de optimización como se indica a continuación:

$$max_{0 \leq \boldsymbol{\alpha} \leq c} \sum_i \alpha_i - \frac{1}{2} \sum_i \sum_j \alpha_i \alpha_j y_i y_j k(\boldsymbol{x_i}, \boldsymbol{x_j}) \tag{33}$$

y la regla de clasificación

$$f(\boldsymbol{x}) = \sum_i \alpha_i y_i k(\boldsymbol{x}, \boldsymbol{x_i}) \tag{34}$$

donde $k(\boldsymbol{x}, \boldsymbol{v}) = x \cdot v$

Por otra parte es importante considerar la forma generalizada de un kernel polinomial [2]:

$$K(x, y) = < \Phi(x), \Phi(y) > = (\gamma < x, y > + c)^d, \quad \gamma > 0 \tag{35}$$

donde usualmente se considera $\gamma = 1$ y $c = 1$.

En este caso es importante observar en la ecuación (34) donde es aplicado el kernel. Principalmente, un kernel de grado uno implicaría un clasificador de la forma $f(\boldsymbol{x}) = \sum_i \alpha_i y_i [< (\boldsymbol{x}, \boldsymbol{x_i} > +1]$. Esto implica el producto punto que intrínsicamente representa el ángulo entre las observaciones y por lo tanto en un kernel de grado uno no es posible realizar esta clasificación adecuadamente en el ejemplo que se muestra en la figura 4 ya que implica transformar el espacio como se indica a continuación ($\gamma = 1, c = 1, d = 1$):

$$\begin{aligned} K(x,y) &= (< x, y > +1) \\ &=< (x_1^2, 1, 1), (1, y_1^2, 1) > \\ &=< \Phi(x), \Phi(y) > \end{aligned} \tag{36}$$

Ahora bien, si se considera un kernel de grado dos, la transformación del espacio en donde están las observaciones es posible realizar la clasificación en función del ángulo entre las observaciones y el origen. Esto se puede observar en la siguiente ecuación, que por motivos didácticos se considera un kernel polinomial con $\gamma = 1, c = 0, d = 2$.

$$\begin{aligned} K(x,y) &= (< x, y >)^2 \\ &= x_1^2 y_1^2 + 2x_1 x_2 y_1 y_2 + x_2^2 y_2^2 \\ &=< (x_1^2, \sqrt{2} x_1 x_2, x_2^2), (y_1^2, \sqrt{2} y_1 y_2, y_2^2) > \\ &=< \Phi(x), \Phi(y) > \end{aligned} \tag{37}$$

La transformación al nuevo espacio, en este caso tiene una implicación importante: se considera una dimensión extra la cual es definida por el signo de las observaciones. En la figura 5 se observa el espacio implícito como resultado de aplicar el truco kernel, es importante considerar que no es necesario aplicar la transformación directamente, basta con aplicar el producto punto entre las observaciones lo cual ayuda computacionalmente.

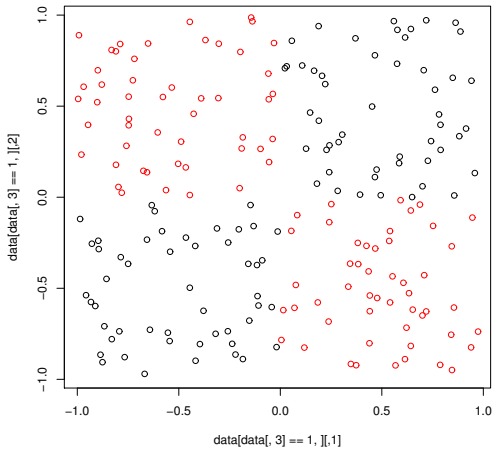

Figura 4: Observaciones cada clase indica un color distinto.

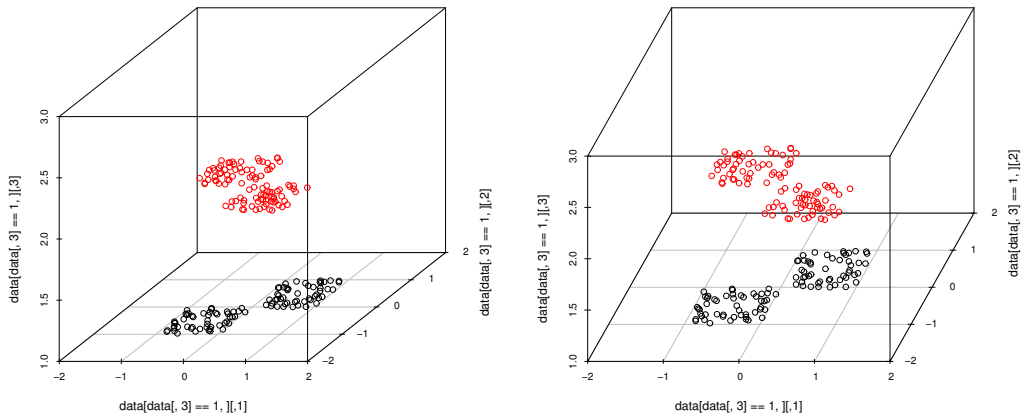

Figura 5: Observaciones en el nuevo espacio utilizando el truco kernel es decir $\Phi(x) = (x_1^2, \sqrt{2}x_1x_2, x_2^2)$.

Análisis empírico

Basado en lo anterior se realiza un procedimiento de clasificación con SVM y por medio de validación cruzada, se generan 500 puntos de donde es tomado

el 70 % para entrenamiento y el resto es de prueba. En la figura 6, se muestran los resultados para el conjunto de entrenamiento con la validación cruzada, principalmente se puede observar que el kernel de grado 1 proporciona los peores resultados en este proceso, por o tanto queda comprobado lo explicado previamente, que el grado mínimo para clasificar SVM con kernels es de grado dos. El modelo que se ajusto más en la validación cruzada es con un kernel de grado 6 y un factor $C = 1{,}75$, esta configuración fue utilizada para el conjunto de prueba. En la figura 7 se presenta la matriz de información como resultado de comparar el conjunto de prueba entre la estimación y el valor real, este modelo tuvo una precisión del 0,99 %.

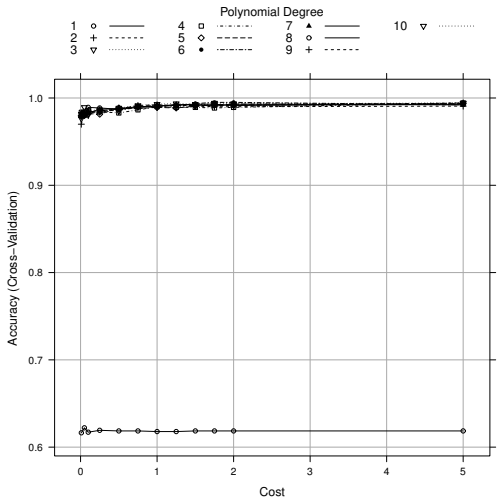

Figura 6: Validación cruzada con distintos pesos en la función objetivo.

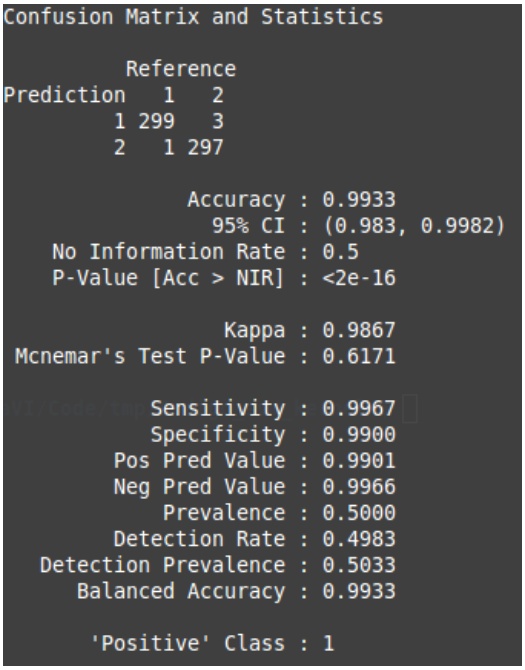

Figura 7: Matriz de información entre los resultados de entrenamiento.

FIGURA C: Analizando la ecuación (33), se puede observar que para maximizar la función objetivo es necesario que el segundo término sea mínimo, es decir minimizar el término $-\frac{1}{2}\sum_i\sum_j \alpha_i\alpha_j y_i y_j k(\boldsymbol{x_i}, \boldsymbol{x_j})$. Este término es el que define el grado del kernel, y además el grado define el número de puntos óptimos (o raíces del polinomio). El grado debe ser igual al número de reglas de decisión que comprende cada dimensión, en el primer ejemplo (figura 5) existen dos reglas , en el segundo la figura 8 existen seis reglas con $x = \{-0{,}5, 0{,}0, 0{,}5\}$, y en $y = \{-0{,}5, 0{,}0, 0{,}5\}$, esto se puede comprobar en la validación cruzada (figura 9) donde se observa que los kernels con un grado mayor a cinco ofrecen mejores resultados.

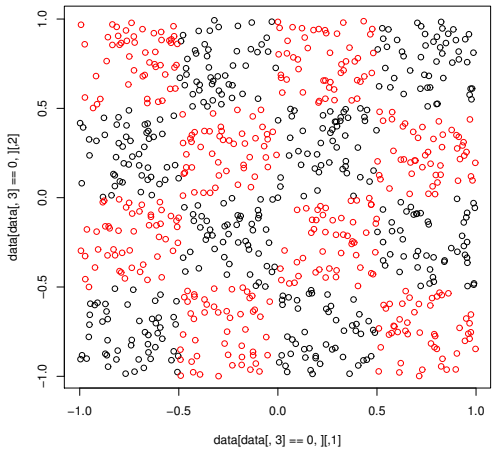

Figura 8: Observaciones cada clase indica un color distinto.

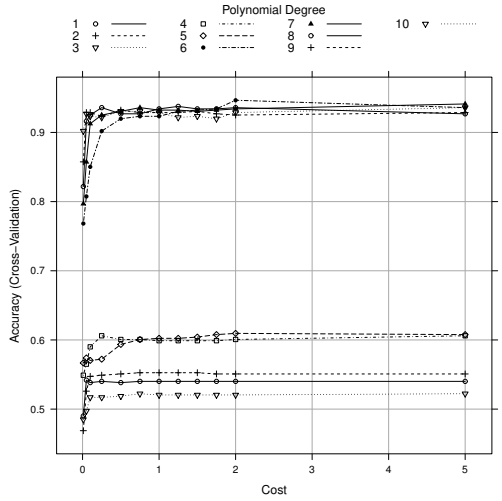

Figura 9: Validación cruzada con distintos pesos en la función objetivo.

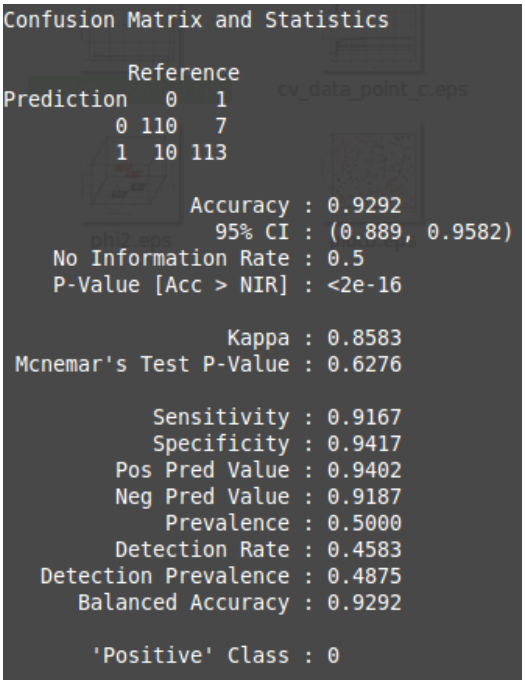

Figura 10: Matriz de información entre los resultados de entrenamiento.

Inducción

Sin saber lo anterior es posible generalizar por medio de inducción matemática, en la figura 11 se muestra otra figura con un número mayor de regiones distintas, que empíricamente en la figura 12 se puede observar que la validación cruzada se ajusta mejor con un kernel polinomial mayor a nueve. Si con un grid de 4 regiones basta un kernel de grado 2, uno de 16 regiones basta un kernel de grado 6 y con 36 baste un kernel de grado 10, entonces por inducción:

$$g > \left(2\left(\sqrt{r}\right) - 2\right) \tag{38}$$

donde r es el número de regiones y g es el grado mínimo del kernel. Esto se puede observar en la tabla 3 Además, es posible comprobar esto analizando un último caso (figura 11) existen diez reglas o raíces que pertenecen en el kernel $x = \{-0{,}75, -0{,}5, -0{,}25, 0{,}0, 0{,}25, 0{,}5, 0{,}75\}$, y en $y = \{-0{,}75, -0{,}5, -0{,}25, 0{,}0, 0{,}25, 0{,}5, 0{,}75\}$

Cuadro 3: Generalización de grado mínimo requerido en un kernel

Regiones	**Grado mínimo**
4	2
16	6
36	10
64	14
100	18
144	22
196	26
256	30
324	34
400	38

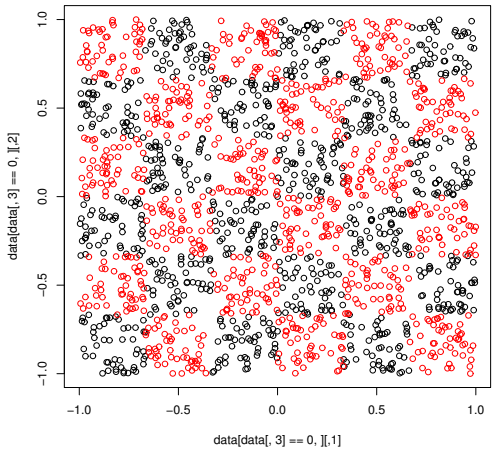

Figura 11: Observaciones cada clase indica un color distinto.

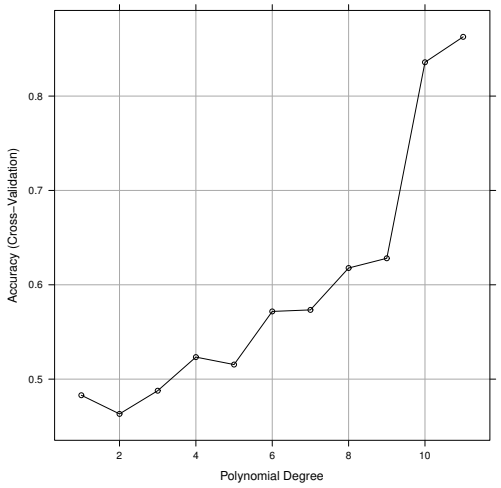

Figura 12: Validación cruzada con distintos pesos en la función objetivo.

[1] D. P. Bertsekas, J. N. Tsitsiklis, Neuro-dynamic programming: an overview, in: Proceedings of the 34th IEEE Conference on Decision and Control, volume 1, IEEE Publ. Piscataway, NJ, pp. 560–564.

[2] C. M. Bishop, Pattern Recognition and Machine Learning (Information Science and Statistics), Springer-Verlag, Berlin, Heidelberg, 2006.

[3] L. Bottou, C.-J. Lin, Support vector machine solvers, Large scale kernel machines 3 (2007) 301–320.

[4] B. Scholkopf, A. J. Smola, Learning with kernels: support vector machines, regularization, optimization, and beyond, MIT press, 2001.

[5] D. Summers-Stay, Productive vision: Methods for automated image comprehension, Ph.D. thesis, University of Maryland, College Park, 2013.