Ingeniería Matemática
FACULTAD DE CIENCIAS
FÍSICAS Y MATEMÁTICAS
UNIVERSIDAD DE CHILE

Tarea 1 Optimización Matemática

Modelamiento

José Aguirre
Matías Meneses

1 Problema 1

a) Para este problema necesitamos minimizar los costes qué realiza cada robot j al mover un objeto i. La función de costes que depende del esfuerzo d de cada robot es:

$$f(d) = 50000 \tanh(D-5) - 50000 \tag{1}$$

Luego tenemos la matriz de esfuerzo D, donde las filas representan los objetos y las columnas, luego D_{ij} representa el esfuerzo que debe realizar el robot j para mover el objeto i. Dado que queremos obtener el menor costo total, es decir, minimizar el costo considerando cada robot y cada objeto, podemos plantear el problema de optimización de la siguiente manera:

Definimos la siguiente función:

$$\begin{aligned} &F(D_{i,j}, x_{i,j}) = f(D_{i,j})x_{i,j} \\ &\text{donde:} \quad x_{i,j} \in [0,1] \end{aligned}$$

Por lo tanto, el problema a optimizar es el siguiente:

$$\begin{aligned} \min_{x} \quad & \sum_{j=1}^{6}\sum_{i=1}^{6} F(D_{ij}, x_{ij}) \\ \text{sujeto a:} \quad & \sum_{i=1}^{6} x_{ij} = 1, \quad \forall j = 1,\dots,6 \\ & \sum_{j=1}^{6} x_{ij} = 1, \quad \forall i = 1,\dots,6 \end{aligned} \tag{2}$$

En la siguiente tabla, se encuentra el resultado del problema al optimizar con *Gurobi*:

Robot	**Objeto**	**Caja**	**Cinta**	**Costo**
1	1	1	Directo	11920.29
6	2	2	Directo	50000.00
3	3	3	Directo	247.26
2	4	4	Directo	99966.46
5	5	5	Directo	99752.74
4	6	6	Directo	11920.29
Costo Total				**273807.05**

Table 1: Asignación de robots a objetos con sus respectivos costos.

b) Utilizando parte del código de la a), ahora el problema agrega una variable nueva y, asignada para los caminos recorribles por la cinta. de esta manera, el nuevo problema de optimización consiste en cómo obtener el menor costo (en función de los esfuerzos) procurando que solo haya un objeto por caja, y un objeto por robot. así, hay una nueva restricción para la cual definimos la nueva función:

$$G(D_{i,j}, y_{i,j})(i,j) = y_{ij} \cdot \min_{k} \Big(f(D^{(1)}_{ki}) + f(D^{(2)}_{kj}) + \text{tarifa} \Big)$$

Quedando el problema a Optimizar como:

$$
\begin{aligned}
\min_{x,y} \quad & \sum_{i=1}^{6}\sum_{j=1}^{6}\left(x_{ij}\cdot F(D_{i,j},x_{i,j})+y_{ij}\cdot G(D_{i,j},y_{i,j})\right)\\
\text{sujeto a} \quad & \sum_{j=1}^{6}(x_{ij}+y_{ij})=1 \quad \forall i\in\{1,\ldots,6\}\\
& \sum_{i=1}^{6}(x_{ij}+y_{ij})=1 \quad \forall j\in\{1,\ldots,6\}
\end{aligned}
$$

Luego variando las tarifas se obtienes los siguientes optimos:

Robot	**Objeto**	**Caja**	**Cinta**	**Costo**
5	5	1	1	10067.07
2	2	2	1	10067.07
3	3	3	Directo	247.26
1	1	4	1	10067.07
4	4	5	2	10067.07
6	6	6	2	10067.07
Costo Total				**50582.61**

Table 2: Asignación de robots a objetos y cajas con sus respectivos costos para tarifa de $10.000

Robot	**Objeto**	**Caja**	**Cinta**	**Costo**
4	4	1	Directo	50000.00
5	5	2	1	50067.07
3	3	3	Directo	247.26
6	6	4	Directo	11920.29
2	2	5	2	50067.07
1	1	6	Directo	11920.29
Costo Total				**174221.99**

Table 3: Asignación de robots a objetos y cajas con sus respectivos costos para tarifa $50.000

Robot	**Objeto**	**Caja**	**Cinta**	**Costo**
1	1	1	Directo	11920.29
4	4	2	Directo	99966.46
3	3	3	Directo	247.26
6	6	4	Directo	11920.29
5	5	5	Directo	99752.74
2	2	6	Directo	50000.00
Costo Total				**273807.05**

Table 4: Asignación de robots a objetos y cajas con sus respectivos costos para tarifa de $100.000

Es claro que la primera tarifa de 10.000$ reduce considerablemente el costo total de la distribución; y la implementación de las cintas sigue siendo rentable hasta la tarifa de 50.000$. Pero al pasar a una tarifa de 100.000$ notamos que las asignaciones óptimas vuelven a las de distribución directa de los objetos a las cajas, pues el costo de usar una cinta pasa a ser mayor que el de ir directo.

c) Este es un caso similar a la b); pero a diferencia de esta pregunta vemos que nos dan un costo adicional por uso, según cuántos robots se hayan subido a la cinta. además, impone una nueva restricción en que no pueden haber más de 5 robots entre las 2 cintas, tal que definimos la nueva variable impuesto = \$2.000, tal que el nuevo problema a resolver es:

$$
\begin{aligned}
\min_{x,y} \quad & \sum_{i=1}^{6}\sum_{j=1}^{6} (x_{ij} \cdot F(D_{ij}, x_{ij}) + y_{ij} \cdot (G(D_{ij}, y_{ij}) + \text{impuesto})) \\
\text{sujeto a:} \quad & \sum_{j=1}^{6}(x_{ij} + y_{ij}) = 1 \quad \forall i \in \{1, \ldots, 6\} \\
& \sum_{i=1}^{6}(x_{ij} + y_{ij}) = 1 \quad \forall j \in \{1, \ldots, 6\} \\
& \sum_{i=1}^{6}\sum_{j=1}^{6} y_{ij} \leq 5
\end{aligned}
$$

resolviendo el problema, obtuvimos los resultados mostrados en la tabla:

Robot	**objeto**	**caja**	**Medio**	**Costo**
1	1	6	directo	11920.29
2	2	2	cinta 1	12067.07
3	3	3	directo	247.26
4	4	5	cinta 2	12067.07
5	5	1	cinta 1	12067.07
6	6	4	directo	11920.29

Table 5: Asignaciones de los objetos considerando el robot, caja y camino que utiliza.

Podemos ver de estos resultados podemos observar cómo esta distribución es un punto medio en el uso tanto de las cintas como el modo original de distribuir objetos, donde la mitad de los robots utilizan el medio original mientras que la otra utiliza las nuevas cintas. El costo total es considerablemente más barato que en a) (con una diferencia de 213,305 pesos) a la vez que reduce el esfuerzo total de los robots. Pero es levemente mayor que en b) (con una diferencia de 9,920 pesos), a costa de entregarle un impuesto por mantención y más restricciones al uso de las cintas.

2 Problema 2: Support Vector Machine

a) Dado que tenemos dos conjuntos de puntos disjuntos y queremos lograr clasificarlos mediante el uso de un hiperplano. Entonces supongamos que tenemos el gráfico de la Figura 1. Luego supongamos que buscamos para el hiperplano H, maximizar la distancia entre los vectores soportes (encerrado en un círculo), así, dado que aquellos puntos que se encuentren en el hiperplano positivo (H^+) serán clasificados como un grupo y los que estén en el hiperplano negativo (H^-) serán clasificados como el otro grupo.

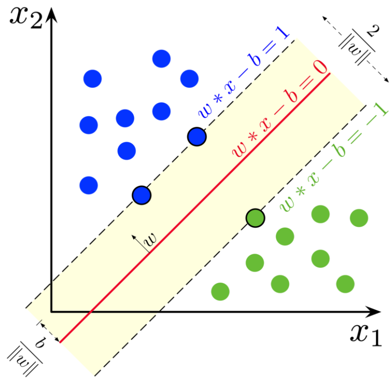

Figure 1: Modelo SVM

Por lo tanto, el problema Optimización se planteará de la siguiente manera:

$$\begin{aligned} \min_{w,b} \quad & \frac{\|w\|^2}{2} \\ \text{sujeto a} \quad & y_i(w^T x + b) \geq 1 \end{aligned} \tag{3}$$

donde:

$$y_n \begin{Bmatrix} +1 & w^T x + b \geq 1 \\ -1 & w^T x + b \leq -1 \end{Bmatrix}$$

Se utiliza la *min* de $\frac{\|w\|^2}{2}$ debido a que así tenemos un problema convexo. La restricción utilizada el valor de la clasificación del punto multiplicada por su valor en el hiperplano. Esto permitirá determinar el mejor hiperplano bajo los datos que existan.

b) El modelo propuesto anteriormente es la versión *'fuerte'* de SVM, en el cual estamos asumiendo que los grupos que ingresamos están en su mayoría (por no decir totalmente) disjuntos. Esto significa que es posible encontrar un hiperplano en la dimensión $d - 1$ de los datos. Luego, utilizando la función *SVM-Hard (X,y)* sobre los datos generados aleatoriamente, se obtiene el siguiente hiperplano:

Se obtuvo el siguiente resultado:

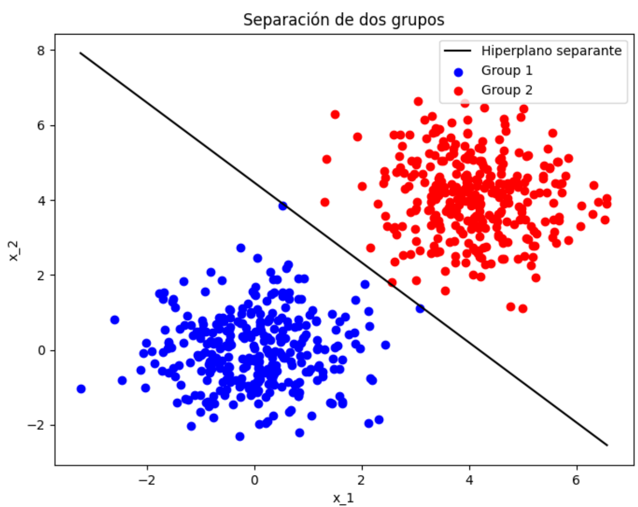

Figure 2: SVM Fuerte considerando 320 puntos aleatorios en dos grupos

Generando así, una perfecta clasificación de ambos grupos.

c) Ahora tenemos dos conjuntos no disjuntos. Es decir que al definir un plano, los puntos estarán mezclados a en este; causando un especial problema a la hora de definir un hiperplano que diferencie a ambos conjuntos.
De esta manera, el nuevo problema de optimización se plantea como sigue:

$$\begin{aligned} \min_{w,b} \quad & \frac{\|w\|^2}{2} + C\sum_{i=1}^{n} \xi_i \\ \text{sujeto a} \quad & y_i(w^T x + b) \geq 1 \end{aligned} \tag{4}$$

donde:

$$y_i \begin{Bmatrix} +1 & w^T x + b \geq 1 \\ -1 & w^T x + b \leq -1 \end{Bmatrix}$$

Al no poder generar un hiperplano que separe perfectamente ambos conjuntos (pues están mezclados), el hiperplano resultante será el que "minimice" la separación de los puntos con su conjunto original, definidos por el hiperplano.

Es por esto que implementamos las nuevas variables ξ_i y C, siendo ξ_i la variable que cuantifica la ruptura del margen (generado por el hiperplano) con respecto a cada punto, y C la variable que amplifica o disminuye la incidencia de los errores con respecto al margen y en cómo se genera el hiperplano (Mientras mayor sea C menos errores se admiten, y viceversa).

d) El modelo propuesto anteriormente es la versión 'suave' de SVM, considerando que los grupos ingresados están mezclados, donde el hiperplano resultante es el plano que minimiza los errores del margen. Así, utilizando la función SVM-Soft(X,Y) sobre los datos generados, se obtiene el hiperplano:

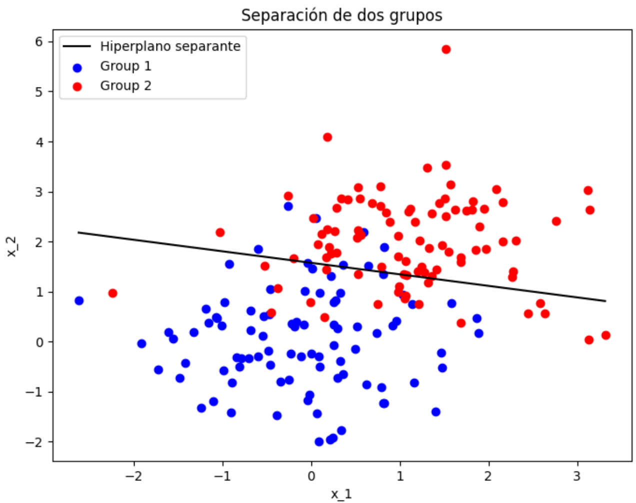

Figure 3: Gráfico resultante de los datos generados aleatoriamente.