Doble Grado en Ingeniería Informática y Matemáticas

APRENDIZAJE AUTOMÁTICO

Cuestionario 2

UNIVERSIDAD DE GRANADA

Alberto Estepa Fernández - *albertoestep@correo.ugr.es*

Mayo de 2020

Índice

1. Ejercicio 1 **2**

2. Ejercicio 2 **3**

3. Ejercicio 3 **4**

4. Ejercicio 4 **5**

5. Ejercicio 5 **6**

6. Ejercicio 6 **8**

7. Ejercicio 7 **9**

8. Ejercicio 8 **10**

9. Ejercicio 9 **11**

10.Ejercicio 10 **12**

11.Ejercicio 11 **14**

12.Ejercicio 12 **15**

1. Ejercicio 1

Considere un problema binario de clasificación en un espacio 3D. Le dan un conjunto de 1000 datos y le dicen que creen que son separables, Para corroborarlo ajusta un modelo perceptron y obtiene un error $E_{in} = 0$. Ahora le piden una cota del error de generalización de dicho clasificador ¿Qué cota sería? Justificar las decisiones que tome.

Este problema es muy parecido al planteado en el ejercicio 5.4 del libro Learning from Data proporcionado en la bibliografía. Usaremos pues las deducciones que se utilizan en dicha sección del libro.

No podemos tener una cota que generalice el error del clasificador. Como se nos comenta en dicha sección, para evitar la trampa en el ejercicio, es extremadamente importante que elijamos nuestro modelo de aprendizaje antes de ver cualquiera de los datos. La elección puede estar basada en información general sobre el problema de aprendizaje, como la cantidad de puntos de datos y el conocimiento previo sobre el espacio de entrada y la función objetivo, pero no en el conjunto de datos real $\mathcal{D}$.

Si ésto no se cumple, no podremos usar la cota de VC y ni concluir nada sobre una posible generalización. Es lo que se conoce como 'data snooping'.

2. Ejercicio 2

Suponga una clase de funciones $\mathcal{H}$ con función de partición $m_{\mathcal{H}}(N) = 1 + N + \binom{N}{2}$. Calcule una cota para E_{out} con el $95\,\%$ de confianza a partir de una muestra de entrenamiento de tamaño 1000.

Vamos a usar la cota de VC generalizada, que nos dice que con probabilidad $0.95 = 1 - \delta (\rightarrow \delta = 0.05)$, tenemos que:

$$E_{out}(h) \leq E_{in}(h) + \sqrt{\frac{8}{N} \log(\frac{4m_{\mathcal{H}}(2N)}{\delta})} \tag{1}$$

Como nos dicen que $N = 1000$ y $\delta = 0.05$:

$$E_{out}(h) \leq E_{in}(h) + \sqrt{\frac{8}{1000} \log(\frac{4m_{\mathcal{H}}(2000)}{0.05})} \tag{2}$$

Calculamos la función de partición según la fórmula dada en el enunciado:

$$m_{\mathcal{H}}(2000) = 1 + 2000 + \frac{2000!}{1998!2!} = 2001001 \tag{3}$$

Así sustituyendo 3 y simplificando en la ecuación 2, tenemos:

$$E_{out}(h) \leq E_{in}(h) + \sqrt{\frac{8}{1000} \log(\frac{4x2001001}{0.05})} = E_{in}(h) + 0.388 \tag{4}$$

3. Ejercicio 3

Considere un espacio muestral $\mathcal{X}$ de dimension d. Considere ahora la clase de funciones $\mathcal{H} = \cup_{k=1}^{K} \mathcal{H}^k$, donde $\mathcal{H}^k$ representa la clase de funciones lineales sobre la potencia k-ésima de cada uno de los elementos de la muestra. Calcular una cota para $d_{vc}(\mathcal{H})$, la dimensión de Vapnik-Cherbonenkis de $\mathcal{H}$, sabiendo que $m_{\mathcal{H}_1 \cup \mathcal{H}_2} < 2^N$ para todo N tal que $d_{vc}(\mathcal{H}_1) + 1 \leq N - d_{vc}(\mathcal{H}_2) - 1$.

Tenemos que $m_{\mathcal{H}_1 \cup \mathcal{H}_2} < 2^N$ para todo N tal que $d_{vc}(\mathcal{H}_1) + d_{vc}(\mathcal{H}_2) + 2 \leq N$

Usando ésto y la definición que se indica en el punto 2.1.3 del libro de Learning from data, tenemos que:

$$d_{vc}(\mathcal{H}_1 \cup \mathcal{H}_2) < d_{vc}(\mathcal{H}_1) + d_{vc}(\mathcal{H}_2) + 2$$

Como las dimensiones de Vapnik&Chervonenkis son el mayor valor de N que hace $m_{\mathcal{H}}(N) = 2^N$, y $N \in \mathbb{N}$, podemos afinar nuestra desigualdad, para poder afinar la cota posteriormente:

$$d_{vc}(\mathcal{H}_1 \cup \mathcal{H}_2) \leq d_{vc}(\mathcal{H}_1) + d_{vc}(\mathcal{H}_2) + 1$$

Aplicando dicha regla a de forma iterativa entre conjuntos de unión, tenemos que:

$$d_{vc}(\mathcal{H}) = d_{vc}(\cup_{k=1}^{K} \mathcal{H}^k) \leq \sum_{k=1}^{K} d_{vc}(\mathcal{H}_k) + (K-1)$$

.

Por otro lado, sabemos que $d_{vc}(\mathcal{H}^k) = d + 1$ por ser $\mathcal{H}^k$ la clase de funciones lineales sobre la potencia k-ésima de cada uno de los elementos de la muestra. Así pues:

$$d_{vc}(\mathcal{H}) \leq \sum_{k=1}^{K} d_{vc}(\mathcal{H}_k) + (K-1) = \sum_{k=1}^{K} (d+1) + (K-1) = K(d+1) + K - 1$$

Ejercicio 2.13 del libro de Learning from data.

4. Ejercicio 4

Probar la siguiente desigualdad de la función de crecimiento:

$$m_{\mathcal{H}}(2N) \leq m_{\mathcal{H}}(N)^2$$

Este problema es el 2.10 del libro Learning from Data proporcionado en la bibliografía. Usaremos pues la definicion 2.1 de dicho libro.

Podemos dividir el conjunto original, que denotamos O, de $2N$ muestras en 2 conjuntos $(I, J \subset O)$ de N muestras cada uno. Notamos $m_{\mathcal{H}}(N) = x$, entonces cada uno de los conjuntos de N muestras, a lo sumo producirán x dicotomías cada uno.

Así, si $m_{\mathcal{H}}(N) = x$ entonces $m_{\mathcal{H}}(N)^2 = x^2$.

Introduciendo la notación vista en teoría, la función de crecimiento para un conjunto de hipótesis $\mathcal{H}$ y un conjunto de muestras X se define como:

$$m_{\mathcal{H}}(N) = \max_{x_1,\ldots,x_N \in X} |\mathcal{H}(x_1, \ \ldots, x_N)|$$

Así aplicado a nuestro conjunto O:

$$m_{\mathcal{H}}(2N) = \max_{x_1,\ldots,x_{2N} \in O} |\mathcal{H}(x_1, \ldots, x_{2N})| \leq$$

$$\leq \max_{x_1,\ldots,x_N \in I} |\mathcal{H}(x_1, \ldots, x_N)| \max_{x_1,\ldots,x_N \in J} |\mathcal{H}(x_1, \ldots, x_N)| =$$

$$= x^2 = m_{\mathcal{H}}(N)^2$$

5. Ejercicio 5

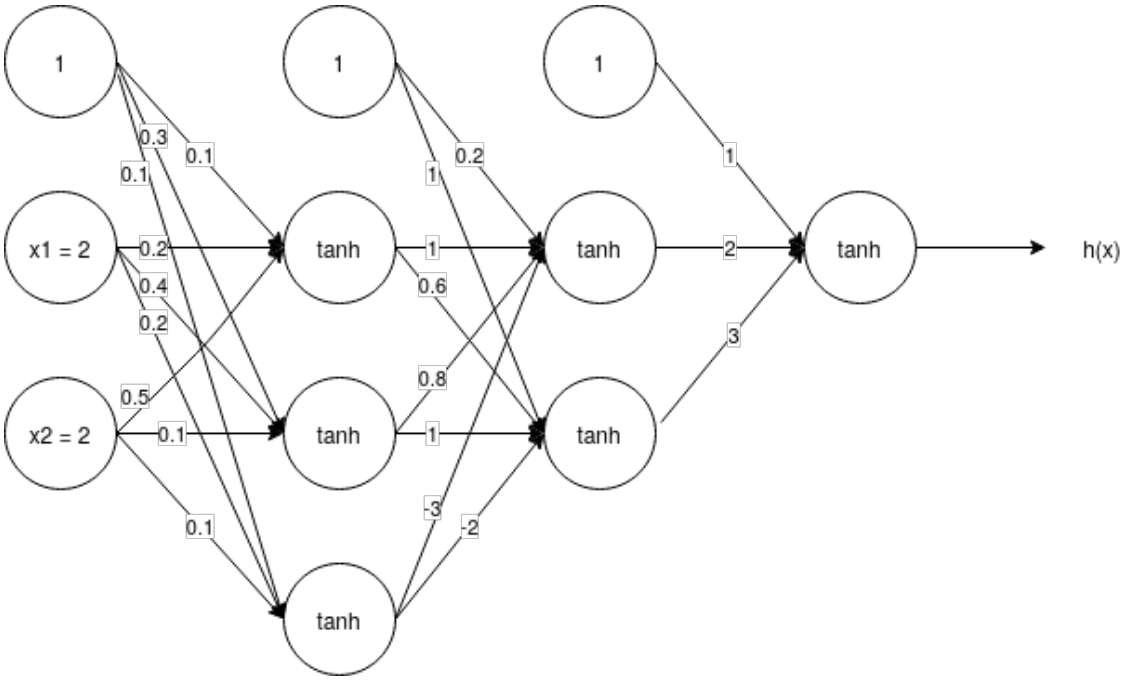
Considere las siguientes matrices de pesos,

$$W_1 = \begin{vmatrix} 0.1 & 0.3 & 0.1 \\ 0.2 & 0.4 & 0.2 \\ 0.5 & 0.1 & 0.1 \end{vmatrix},\ W_2 = \begin{vmatrix} 0.2 & 1.0 \\ 1.0 & 0.6 \\ 0.8 & 1.0 \\ -3.0 & -2.0 \end{vmatrix},\ W_3 = \begin{vmatrix} 1 \\ 2 \\ 3 \end{vmatrix}$$

como las correspondientes a un modelo MLP de clasificación de tres capas cuya entrada esta definida por vectores $[1, x1, x2]^T$ y $tanh$ como función no lineal en todas las capas.

Escriba el grafo del modelo y propage el vector de entrada $[1, 2, 2]^T$ de etiqueta 1. Muestre los cálculos específicos que dan lugar a cada uno de los siguientes valores: vectores de entrada s, salida x y sensibilidades δ de cada capa. Como consecuencia calcule los valores numéricos de las derivadas del error respecto de cada uno de los elementos de $W = \{W_1, W_2, W_3\}$.

Usando la información de las matrices de pesos, el grafo del modelo es el siguiente:



Vamos a propagar el vector de entrada dado por la red calculando los vectores de salida x y de entrada s. Sabemos que los vectores de entrada s se calculan tal que así:

$$s^{(l)} = (W_l)^T x^{(l-1)},\ \textit{con } l \in [1, 3]\ \textit{nivel de capa}$$

y el vector de salida x es aplicar la función no lineal tanh() a los valores de entrada de la anterior capa y concatenarlo con el valor 1.

$$x^{(0)} = \begin{vmatrix} 1 \\ 2 \\ 2 \end{vmatrix} \qquad s^{(1)} = \underbrace{\begin{vmatrix} 0.1 & 0.2 & 0.5 \\ 0.3 & 0.4 & 0.1 \\ 0.1 & 0.2 & 0.1 \end{vmatrix}}_{W_1^T} \underbrace{\begin{vmatrix} 1 \\ 2 \\ 2 \end{vmatrix}}_{x^{(0)}} = \begin{vmatrix} 1.5 \\ 1.3 \\ 0.7 \end{vmatrix}$$

$$x^{(1)} = \begin{vmatrix} 1 \\ \tanh(1.5) \\ \tanh(1.3) \\ \tanh(0.7) \end{vmatrix} = \begin{vmatrix} 1 \\ 0.905 \\ 0.862 \\ 0.604 \end{vmatrix} \qquad s^{(2)} = W_2^T x^{(1)} = \begin{vmatrix} -0.017 \\ 1.197 \end{vmatrix}$$

$$x^{(2)} = \begin{vmatrix} 1 \\ \tanh(-0.017) \\ \tanh(1.197) \end{vmatrix} = \begin{vmatrix} 1 \\ -0.017 \\ 0.833 \end{vmatrix} \qquad s^{(3)} = W_3^T x^{(2)} = 3.465$$

$$x^{(3)} = \tanh(3.465) = 0.998$$

Ahora calculamos las sensibilidades de cada capa. Aunque la última salida se podría aproximar a la identidad y casi terminar el ejercicio, he decidido calcular los valores de las sensibilidades sin aproximar dicho valor, así pues, sabiendo que $\tanh'(x) = sech^2(x)$:

$$\delta^{(3)} = 2(x^{(3)} - y)\tanh'(s^{(3)}) = 2(0.998 - 1)sech^2(3.465) = -0.0000156175$$

$$\delta^{(2)} = \tanh'(s^{(2)}) \otimes [W_3\delta^{(3)}]_1^{d(2)} = \begin{vmatrix} sech^2(-0.017) \\ sech^2(1.197) \end{vmatrix} \otimes (-0.0000156175) \begin{vmatrix} 2 \\ 3 \end{vmatrix} = \begin{vmatrix} -0.000031226 \\ -0.0000143626 \end{vmatrix}$$

$$\delta^{(1)} = \tanh'(s^{(1)}) \otimes [W_2\delta^{(2)}]_1^{d(1)} = \begin{vmatrix} sech^2(1.5) \\ sech^2(1.3) \\ sech^2(0.7) \end{vmatrix} \otimes \begin{vmatrix} -0.00003984356 \\ -0.0000393434 \\ 0.0001224032 \end{vmatrix} = \begin{vmatrix} -7.2x10^{-6} \\ -0.00001 \\ 0.000078 \end{vmatrix} \approx 0$$

Por último calculamos las derivadas del error respecto de cada uno de los elementos, ésta vez si consideramos el valor aproximado a cero ya que si no las cuentas serían muy engorrosas:

$$\frac{\partial e}{\partial W^{(1)}} = x^{(0)}(\delta^{(1)})^T \approx \begin{vmatrix} 0 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \end{vmatrix}, \quad \frac{\partial e}{\partial W^{(2)}} = x^{(1)}(\delta^{(2)})^T \approx \begin{vmatrix} 0 & 0 \\ 0 & 0 \\ 0 & 0 \\ 0 & 0 \end{vmatrix},$$

$$\frac{\partial e}{\partial W^{(3)}} = x^{(2)}(\delta^{(3)})^T \approx \begin{vmatrix} 0 \\ 0 \\ 0 \end{vmatrix}$$

Notamos que el operador $\otimes$ es la multiplicación componente a componente.

Información extraida del pdf de la bibliografía: *Learning_from_data_NN.pdf*

6. Ejercicio 6

Analice con detalle el algoritmo ID3 de construcción de un clasificador en árbol para muestras generales en términos del Compromiso Sesgo-Varianza. Describa la clase de funciones que ajusta y argumente la contribución de los distintos pasos del algoritmo en el crecimiento o decrecimiento de estos errores y como alcanza una buena solución.

Analizando el algoritmo ID3 con respecto a su comportamiento en el tema sesgo-varianza, podemos afirmar que:

- Al igual que otros árboles de decisión tienden a crecer muy rápido y con muchos puntos de división. Esto ocasiona una varianza muy alta que sobreajusta los datos de entrenamiento. A mayor profundidad mayor varianza y menor sesgo. El objetivo es buscar un equilibrio en dichas variables.
- El algoritmo crea buenas divisiones en la parte superior del árbol. Lo que hace que en pocas separaciones podamos tener un buen modelo y funciona mucho mejor en árboles poco profundos.

Se modifican dichos errores al aumentar la profundidad del árbol, produciendo que baje el sesgo y suba la varianza. Dicho algoritmo permite podar ciertas ramas del árbol para evitar el sobreajuste de los datos (sube el sesgo y baja un poco la varianza ante éstas podas) Si ésto ocurre durante la construcción del árbol estamos ante stopping early y si ocurre después ante post running.

Si tenemos una partición en M regiones $R_1, R_2, \ldots, R_i$ y modelamos la respuesta de la muestra como una constante c_i en cada región, la clase de funciones que ajusta es de la forma:

$$f(x) = \sum_{i=1}^{M} c_i I[x \in R_i]$$

donde $I[x \in R_i]$ es 1 si x pertenece a R_i.

Información extraída de https://www.thelearningmachine.ai/tree-id3 y de Quinlan, J. R. 1986. Induction of Decision Trees. Mach. Learn. 1 y de las transparencias del tema 8.

7. Ejercicio 7

Analice los algoritmos de los modelos SVM-Hard e identifique con precisión dos mayores ventajas y una gran desventaja frente a otros algoritmos. ¿En qué contribuye el modelo SVM-Soft al problema de clasificación binaria? Justifique las respuestas sin usar argumentos de Núcleo.

Ventajas de los modelos SVM-Hard:

- Es capaz de encontrar la separación con márgen óptimo para el problema, obteniendo mayor robustez y consistencia frente a nuevos datos y ruido que otros algoritmos.
- El coste computacional y el tiempo de ejecución es mucho menor que para otros algoritmos de clasificación.

Desventaja: Los datos tienen que ser separables.

SVM-Soft incluye una variable que suaviza las restricciones del margen, permitiendo que algunos puntos no cumplan dicho margen o incluso que ésten mal clasificados si la variable es suficientemente poco restrictivo. Así con SVM-Soft se permite que los datos no sean separables.

8. Ejercicio 8

1. **Analice la construcción del clasificador de Random Forest y discuta bajo el criterio Sesgo-Varianza las razones y las características de los problemas en los que es esperable una alta eficacia.**

   Como la dimensión VC de la clase de hipótesis que genera Random Forest es infinita, dichos clasificadores tienen un sesgo bajo, es decir puede ajustarse con precisión a los datos con los que se entrena al tener una clase de funciones de gran tamaño. Así, lo lógico es tener una alta varianza y es posible que sobreajuste a los datos. Sin embargo ésto ocurre para la construcción de cualquier árbol. Random Forest trata de elegir un número menor de atributos para minimizar la correlación entre los árboles.

   Así, Random forest tendrá más posibilidades de ser eficaz al enfrentarnos a problemas de pocas características y ante aquellos problemas que tienen variables categóricas (ésto se explicará en el apartado 2).

2. **¿Cuáles son las mejoras que introduce frente a clasificadores como SVM?**

   Random Forest emplea clasificadores simples intentando que la correlación entre ellos sea nula. Ésto permite tener mejores valores de varianza con respecto a otros clasificadores de árboles.

   Por otro lado, como hemos comentado, Random Forest es mejor que otros clasificadores, como por ejemplo SVM, cuando tenemos variables categóricas entre manos, ya que Random Forest introduce mejoras para tratar éstos tipos de datos.

   Por otro lado, Random Forest clasifica de forma multiclase directamente, sin tener que preprocesar los datos de forma concreta para dicho problema. Al contrario que los clasificadores como SVM que usan técnicas OneVsRest o modificaciones de ésta.

3. **¿Es Random Forest óptimo en algún sentido?**

   Podemos basarnos en la misma respuesta que dimos en el Cuestionario anterior, no podemos decir que una técnica es óptima en algún sentido para todos los problemas. Ésto es lo que recoge el Teorema de No-Free-Lunch. Concretamente sabemos que existe al menos una distribución de probabilidad para la cuál nuestro algoritmo falla. Por tanto, si nos encontramos con tal distribución el algoritmo no dará buenos resultados.

Justifique con solidez y precisión las contestaciones.

9. Ejercicio 9

Trabaja para una empresa pesquera que explota una piscifactoria en la costa. Una área grande que está acotada y cerrada. La empresa desea tener un modelo que permita estimar los kilos de pescado presentes y futuros a partir de muestras extraidas. Disponen de información detallada de la distribución de los bancos de peces en el área a lo largo de las horas del día. Las muestras las obtienen usando los barcos y aparejos de pesca que se usan en la explotación comercial. ¿Cómo organizaría el experimento para garantizar un resultado correcto?

El objetivo es tomar una muestra representativa de toda la población de peces. Para ello lo ideal es que los datos de la muestra se intenten en la medida de lo posible a la uniformidad de individuos de cada clase de la población de peces y que todas las clases tengan representación en la muestra.

Para lograr ésto, como tenemos información detallada de los bancos de peces en el área a lo largo de las horas del día, cada muestra debería componerse como mínimo de peces pescados durante todas las horas del día y los barcos de pesca deberían repartirse por todo el área, y con todas las posibles técnicas de pesca disponibles. Además, hay que atender a las temporadas de los peces: habrá épocas que algunos peces estarán en época de ciclo reproductivo y no será aconsejable pescarlos, o no existe dicha posibilidad; deberíamos tomar muestras a lo largo de un año como mínimo, para atender dicha variable representativa. Así podremos pensar que la muestra sigue una distrubución de probabilidad y que las muestras sean independientes e idénticamente distribuidas. Sin embargo hay que tener en cuenta otros factores importantes, como pescar peces de todas las edades, aunque sabemos que algunos peces pequeños no se pueden pescar por ley. Si podemos realizar los experiementos con todas los factores y variables descritos, lograremos un conjunto de muestras que siguen una misma distrubución de probabilidad y que son independientes e idénticamente distribuidas, lo que nos permitiría estimar modelos de aprendizaje para nuestro problema de forma satisfactoria.

Como es lógico el conjunto de muestras (el conjunto de experimentos consistente en pescar los peces como hemos descrito anteriormente: pescados durante todas las horas del día, los barcos de pesca deberían repartirse por todo el área, y con todas las posibles técnicas de pesca disponibles; debe ser considerablemente grande, para representar a la distribución de probabilidad de forma efectiva.

10. Ejercicio 10

Considere que dispone de un conjunto de datos linealmente separable. Recuerde que una vez establecido un orden sobre los datos, el algoritmo Perceptrón encuentra un hiperplano separador interando sobre los datos y adaptando los pesos de acuerdo a su regla de adaptación. Considere ahora que en lugar de seguir el orden de adaptación establecido para el Perceptron, decide cambiarlo adaptando en cada iteración el caso peor clasificado. Analice el nuevo algoritmo iterativo y diga: a) ¿Existe solución? ; b) ¿En caso afirmativo diga que tipo de solución es y si, en general, es mejor, peor o equivalente a la del algoritmo Perceptron? Justificar adecuadamente/matematicamente el resultado.

Sí existe solución ya que los datos son linealmente separables. Sabemos que la regla de adaptación del algoritmo Perceptrón estándar es la siguiente:

$$\begin{cases} w_{updated} = w_{current} + y_i x_i & si \quad sign(w^T x_i) \neq y_i \\ \\ w_{updated} = w_{current} & si \quad sign(w^T x_i) = y_i \end{cases}$$

Así pues vemos que se irán actualizando los pesos mientras se encuentre un elemento mal clasificado.

El nuevo método propuesto para el algoritmo del Perceptrón consiste es buscar el elemento de la muestra que éste peor clasificado, es decir, ya no buscamos un índice de un elemento que éste mal clasificado ($sign(w^T x_i) \neq y_i$) si no que sea el que peor clasificado éste, valor que se puede medir con el producto $w^T x_i y_i$, concretamente buscamos el valor de i que hace que dicho producto sea el mínimo (si el punto está mal clasificado este producto será menor que 0 ya que el signo de la etiqueta y_i tendrá signo opuesto al valor predicho $w^T x_i$ en ese momento, y cuanto menor sea dicho valor, más lejos de la recta de clasificación estará; pero si el punto está bien clasificado el producto será un valor positivo, pero más pequeño contra más cercano esté de la recta de clasificación).

Así el cambio con respecto al algoritmo éstandar es buscar el elemento que hace dicho producto mínimo y a partir de aquí realizar el mismo procedimiento, aunque esta vez no hace falta que el punto esté mal clasicado, es decir:

Sea N el tamaño de la muestra:

$$Escoger\ i \in \{0, \dots, N\} \text{ tal que}$$
$$w^T x_i y_i < w^T x_j y_j\ \forall j \in \{0, \dots, N\} \rightarrow w_{updated} = w_{current} + y_i x_i$$

Como los datos son linealmente separables, habrá algún momento en que no exista dicho i y ya que habremos encontrao al menos dos elementos diferentes que minimicen dicho producto y la desigualdad no sea estricta, entonces habremos encontrado nuestra solución: todos los datos estarán clasificados correctamente y existen

varios puntos con mínima distancia a la recta de clasificación, son lo que llamamos puntos de soporte. El algoritmo obtiene los mismos resultados que el algoritmo de SVM.

La solución obtenida construye un hiperplano óptimo de modo que el margen de separación entre las clases en los datos se amplía al máximo. Por ésto consideramos que es mejor con respecto al Perceptrón original, ya que es más compacta y robusta frente a nuevos datos o frente al ruido.

11. Ejercicio 11

Analice el algoritmo SVD-Hard.

1. **¿Qué información precisa podemos extraer de la solución del problema dual respecto del problema primal?.**

   Existe un método alternativo al SVM que se denominaforma dual de SVM que utiliza el multiplicador de Lagrange para resolver el problema de optimización de restricciones.

   El objetivo es:

$$\underset{\alpha}{maximizar} \sum_{i=1}^{n} \alpha_i - \frac{1}{2} \sum_{i=1}^{n} \sum_{j=1}^{n} \alpha_i \alpha_j y_i y_j (X_i^T X_j)$$

$$sujeto\ a\ que\ \alpha_i \geq 0\ \forall i \in \{1, 2, \ldots, n\}\ y\ \sum_{i=1}^{n} \alpha_i y_i = 0$$

2. **¿Qué conclusiones podemos sacar de dicha información sobre la posición de las muestras respecto de la solución?**

Argumentar los resultados.

12. Ejercicio 12

Suponga un corredor de apuestas que durante 7 semanas seguidas, recibe un e-correo que predice el resultado de una carrera del próximo fin de semana, donde siempre hay apuestas substanciosas a ganar. Cada lunes revisa la predicción y observa que la predicción es correcta en todas las ocasiones. El lunes después de la séptima carrera recibe un e-correo diciendole que si desea conocer la predicción de la próxima carrera debe pagar 5000€. Identifique el problema de aprendizaje, diga cual es la función de crecimiento y si merece la pena pagar.

Aclaración: Entendemos que el predictor estudia el ganador de la carrera y no cada una de las posiciones del la carrera. Si ésta suposición no fuese cierta llegaremos al mismo resultado, pero los números serían distintos (cambiaría en que tendríamos que estudiar las posibles permutaciones entre los participantes en cada carrera).

Estamos ante un problema de aprendizaje supervisado en el que el objetivo es predicir el resultado de la carrera del fin de semana. Disponemos de 7 muestras (cada una de las predicciones que le ha llegado por correo al corredor de apuestas).

Si la cada carrera i tiene un número p_i de participantes, sabemos que para la carrera 1, existen p_1 posibles ganadores, para la carrera 2, existen p_2 posibles ganadores y generalizando a la carrera i, existen p_i posibles ganadores.

Así para las $N = 7$ muestras que tenemos, existen $\prod_{i=1}^{7} p_i$ posibles predicciones (vectores de tamaño 7 donde se indica el ganador de cada carrera) de resultados.

De ésta forma, generalizando a un cierto N, la función de crecimiento de nuestro problema es $m_{\mathcal{H}}(N) = \prod_{i=1}^{N} p_i$

Sabiendo ésto NO merece la pena pagar, puesto que nada asegura que el predictor acierte con ninguna probabilidad. Podemos ser, por ejemplo, parte de una estafa en la que se envía un e-mail a $\prod_{i=1}^{7} p_i$ personas donde a cada uno le llega una predicción distinta, y como hemos visto, existirá una y solo una persona (tal y como hemos expuesto nuestro ejemplo) a la que le llegarán los resultados correctos y el siguiente resultado no tiene ningún fundamento matemático que lo avale. Además la muestra puede no seguir una distribución de probabilidad y no ser independiente ni idénticamente distribuída.