PONTIFICIA UNIVERSIDAD CATÓLICA DE CHILE
ESCUELA DE INGENIERÍA
DEPARTAMENTO DE CIENCIA DE LA COMPUTACIÓN
IIC2613 INTELIGENCIA ARTIFICIAL

8 de Noviembre, 2016

Interrogación 3

Tiempo: 90 minutos, SIN APUNTES

Sólo consultas de enunciado, si necesita realizar algún supuesto indíquelo en su respuesta.

Nombre:_____

1. (*16 puntos*) Comente o responda las siguientes afirmaciones o consultas, fundamentando brevemente cada respuesta.

1.1. Un vector de soporte debe siempre corresponder a una instancia del set de entrenamiento.

Solución: *Correcto, los vectores de soporte son las instancias del set de entrenamiento que definen la posición de la superficie de decisión.*

1.2. Según la visión genérica de un algoritmo de aprendizaje de máquina, el uso de una métrica de rendimiento permite encontrar la hipótesis que presenta la mejor capacidad de generalización medida en un set de test.

Solución: *Falso, la exploración del espacio de hipótesis se realiza sobre el set de entrenamiento, no el set de test.*

1.3. El problema de sobreajuste (overfitting) consiste en que el rendimiento medido en el set de test es menor al error observado en el set de entrenamiento.

Solución: *Falso, el observar un menor error en set de test no es necesariamente una indicación de sobreajuste, pues este error puede ser ocasionado por otras razones, incluyendo la natural aleatoriedad con que usualmente se generan los conjuntos de entrenamiento y test. El problema en la afirmación es indicar que el sobreasjuste "consiste" de lo indicado, pues en realidad el sobreajuste consiste en la pérdida de generalidad del algoritmo de aprendizaje de máquina debida a una memorización (sobreasjuste) del set de entrenamiento.*

1.4. El algoritmo visto en clases para explorar el espacio de hipótesis de un árbol de decisión garantiza encontrar el árbol que presenta el mejor rendimiento posible en el set de entrenamiento.

Solución: *Falso, como discutimos en clases, la estrategia de exploración del algoritmo visto en clases es de tipo codicioso (greedy). En cada paso, se elige independientemente el atributo que mejor separa*

los datos según el valor de las clases, no se considera la discriminatividad de atributos en conjunto, por tanto, esta estrategia sólo garantiza encontrar óptimos locales.

1.5. Como discutimos en clases, la técnica de random forest consiste en contruir una serie de árboles de decisión (ensamble), en que cada árbol contiene un atributo diferente para el nodo raíz.

Solución: *Falso, dada la aleatoriedad con que se seleccionan los atributos usados en la construcción de cada árbol, no es posible garantizar que cada árbol tenga un atributo diferente en el nodo raíz.*

1.6. En general, en problemas de aprendizaje de máquina se puede asegurar que mientras más datos de entrenamiento sean usados, mejor será el rendimiento del modelo resultante en términos de sus capacidades de generalización.

Solución: *Falso, si bien en muchos casos más datos ayudan, no es posible "asegurar" que al incorporar más datos el resultado mejore. Por ejemplo, los nuevos datos incorporados al entrenamiento pueden ser redundantes o no informativos, aún peor, los nuevos datos pueden ser ruidosos y por tanto degradar la solución. Para asegurar que los nuevos datos van a ayudar al rendimiento, se debe verificar que estos nuevos datos aporten diversidad al set de entrenamiento y que no sean ruidosos. .*

1.7. Después de ejecutar una red neuronal usando el algoritmo de backpropagation y el método batch, los resultados arrojados no son satisfactorios, una posibilidad para intentar mejorar el rendimiento obtenido es volver a entrenar la red pero utilizando un método incremental de entrenamiento para ajustar los pesos.

Solución: *Falso, el método incremental es una aproximación del método batch por tanto no entregará una mejor solución.*

1.8. Después de ejecutar una red neuronal usando el algoritmo de backpropagation y el método batch, los resultados arrojados no son satisfactorios, una posibilidad para intentar mejorar el rendimiento obtenido es volver a entrenar la red pero con un valor inicial distinto para los pesos de la red.

Solución: *Correcto, el bajo rendimiento puede ser ocasionado por la convergencia a un óptimo local, por lo cual, cambiar el punto de inicio puede permitir que el descenso de gradiente converga a una mejor solución.*

1.9. En un árbol de decisión, el nodo donde se produce el mayor valor de ganancia de información durante la construcción del árbol es siempre el nodo raíz.

Solución: *Falso, la ganancia de información está asociada a cuan homogéneos sean los grupos resultantes de un split en un nodo, por tanto, no se puede garantizar lo indicado. De hecho, dado que en general a lo largo de árbol los grupos se van paulatinamente homogeneizando, es esperable que los valores más altos de ganancia de información sean cerca de los nodos hoja en lugar del nodo raíz.*

1.10. Las redes neuronales tipo feedforward son una alternativa a considerar para problemas de clasificación donde la salida debe considerar un factor de confianza en la clasificación.

Solución: *Correcto, como discutimos en clases, con una transformación apropiada, los valores de salida de la red neuronal pueden ser convertidos a un valor de confianza o incluso una probabilidad.*

1.11. Sin contar los nodos hoja, un árbol de decisión que es entrenado con un set de N atributos puede tener como máximo N nodos en el árbol resultante.

Solución: *Falso, cada atributo puede aparecer en varias ramas del árbol, por tanto, el número de nodos puede ser mayor a N.*

1.12. El espacio de hipótesis de una red neuronal esta dado por el conjunto de posibles valores que toman los pesos de la red.

Solución: *Correcto, la exploración que realiza backpropagation busca asignar valores a los pesos de la red.*

1.13. El espacio de hipótesis de una máquina de vectores de soporte esta dado por el set de hiperplanos que potencialmente puedan separar los datos de entrenamiento.

Solución: *Correcto, la optimización en un SVM busca posible hiperplanos, ya sea en el espacio original o el dado por la función de kernels. Adicionalmente, la óptimización busca valores apropiados para las variables slack.*

1.14. Sólo si las atributos involucrados son binarios, el espacio de hipótesis de un árbol de decisión esta dado por las disyunciones de conjunciones posibles con estos atributos.

Solución: *Falso, los árboles de decisión pueden operar con atributos que toman más de 2 valores. Las disyunciones de conjunciones son sobre el resultado del test (consulta) en cada nodo, no sobre el espacio de valores de cada atributo.*

1.15. Si en un problema de clasificación se observa que el rendimiento en el set de validación es sustancialmente menor que en el set de entrenamiento, esto puede indicar problemas de sobreajuste.

Solución: *Verdadero, una baja sustancial en el error del set de validación indica la posibilidad de problemas de sobreajuste. Algo similar es válido cuando uno observa una baja sustancial del error en el set de test.*

1.16. En una máquina de vectores soporte se puede evitar, o aminorar, problemas de sobreajuste aumentando el valor de la constante asociada a la penalización dada a las variables de slack ξ_k en la función de pérdida.

Solución: *Falso, cada variable slack regula la violación de la condición de margen de la instancia de entrenamiento asociada. Mientras más grande sea la penalización asociada a estas variables slack, el modelo tenderá a buscar soluciones que minimizen violaciones a la condición de mínimo margen (slack=0), incluso si esto significa modelar un punto ruidoso (sobreajustar). A medida que la penalización a los slack es menor, el modelo será menos estricto en su afán de encontrar soluciones que no violen el mínimo margen, siendo más propenso a permitir violaciones a través de compensación con valores de slack distintos a cero.*

2. (*16 puntos*) Redes Neuronales

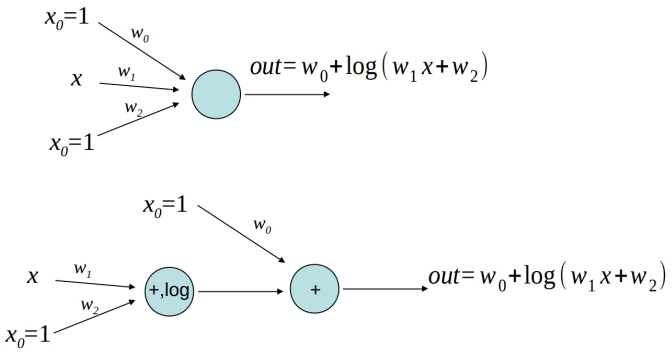
a. (*8 pts*) Se desea entrenar una neurona con la siguiente estructura:

$$f(x) = w_0 + log(w1\ x + w_2)$$

(a) Haga un esquema de la estructura de la red (en este caso 1 sola neurona) [2 puntos].

Solución:

En el enunciado se mencionaba 1 neurona, por tanto, la primera figura es la correcta. Al hacer la pauta me di cuenta que también es correcto pensar en 2 unidades (neuronas), por tanto, el esquema en la segunda figura también se considerará correcto.



(b) Indique las ecuaciones para actualizar los pesos w_0, w_1 y w_2, asumiendo que utiliza el método incremental de descenso de gradiente y error medio cuadrático como función de error (i.e., la misma función de error vista en clases) [6 puntos].

Solución:

La regla de actualización está dada por:

$$w_i^{new} = w_i^{old} - \eta \frac{\partial E}{\partial w_i}$$

donde la función de error se calcula sobre los ejemplos (x_i, t_i) del set de entrenamiento D. Las salidas de la red están dadas por $o_i = w_0 + log(w1\ x_i + w_2)$, por tanto tenemos:

$$E(w) = \sum_D \frac{1}{2}(t_i - o_i)^2$$

En el modo incremental, la sumatoria que resulta al derivar el gradiente de la función de error se omite, por tanto, las reglas de actualización de pesos son:

$$w_0^{new} = w_0^{old} + \eta(t_i - o_i)$$

$$w_1^{new} = w_1^{old} + \eta(t_i - o_i)\frac{x_i}{w_1\ x_i + w_2}$$

$$w_2^{new} = w_2^{old} + \eta(t_i - o_i)\frac{1}{w_1\ x_i + w_2}$$

b. (*8 pts***)** Considere las siguientes posibles configuraciones para una red neuronal:

(i) Perceptron lineal.

(ii) Perceptron con activación sigmoidal.

(iii) Perceptron multicapa (i.e., red de unidades sigmoidales con capa oculta).

(a) ¿Cuál(es) de estas estructuras puede aprender un OR lógico?, justifique brevemente [4 puntos].

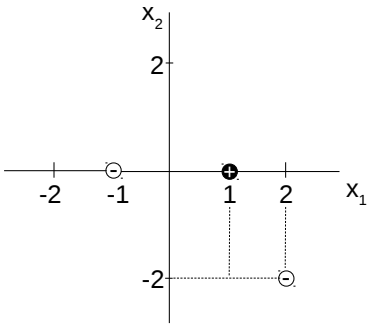
Solución: *La superficie de decisión es linealmente separable, por tanto, de las alternativas, el Perceptron con activación sigmoidal y el Perceptron multicapa pueden resolver el problema. El perceptron con salida lineal no puede represesentar o aproximar salidas binarias, por tanto, no es una herramienta adecuada para este problema.*

(b) ¿Cuál(es) de estas estructuras puede aprender un OR exclusivo (XOR)?, justifique brevemente [4 puntos].

Solución: *Este problema lo discutimos en clases, dado que la superficie de decisión no es linealmente separable, de las alternativas, sólo el Perceptron multicapa puede resolver este caso.*

3. (*16 puntos*) SVM

a. (*8 pts*) Se tienen 3 datos para entrenamiento: 2 de clase negativa (-1,0) y (2,-2); y 1 de clase positiva (1,0), tal como muestra la figura.



Se desea determinar una solución usando un kernel lineal (hyperplano) que tenga como salida -1 y 1 para las clases negativa y positiva, respectivamente.
¿Cuál(es) de los siguientes separadores lineales cumple con las condiciones de optimalidad de la formulación de SVM vista en clases?. En cada caso justifique brevemente su respuesta.

(a) $x_1 + x_2 = 0$

(b) $x_1 + 1.5x_2 = 0$

(c) $2x_1 + 3x_2 = 0$

Solución:

(a) En este caso, el plano no cumple las condiciones de optimalidad, basta chequear que el plano pasa por el punto (2,-2).

(b) En este caso, el plano cumple las condiciones de optimalidad, i.e, clasifica los 3 puntos correctamente y proporciona un margen igual a 1:

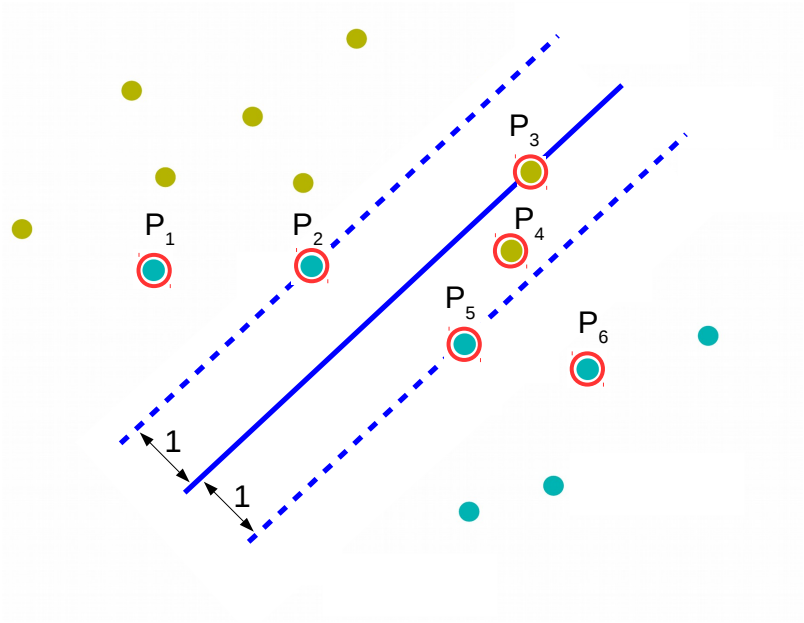
- $f(-1, 0)$: -1 + 0 = -1, clase negativa. Slack es 0, margen=1.
- $f(2, -2)$: 2 -3= -1, clase negativa. Slack es 0, margen=1.
- $f(1, 0)$: 1 + 0 = 1, clase positiva. Slack es 0, margen=1.

(c) En este caso, el plano NO cumple todas las condiciones de optimalidad, i.e, si bien clasifica los 3 puntos correctamente, el margen es igual a 2:

- $f(-1, 0)$: -2 + 0 = -2, clase negativa. Slack es 0, margen=2.
- $f(2, -2)$: 4 - 6= -2, clase negativa. Slack es 0, margen=2.
- $f(1, 0)$: 2 + 0 = 2, clase positiva. Slack es 0, margen=2.

Al analizar los planos en (b) y en (c), si bien son equivalentes, sólo uno de ellos cumple la condición de normalización que realizamos al derivar las ecuaciones de optimización para SVMs.

b. (*8 pts*) En la siguiente figura indique el valor exacto o un rango posible de valores para las variables de slack asociadas a los 6 puntos P_1 a P_6, destacados con un doble círculo.

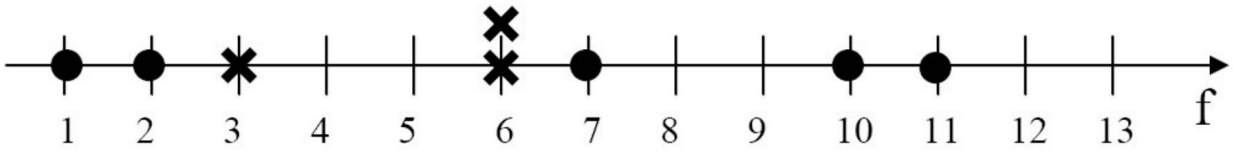


Solución: El comportamiento de las variables de slack en las distintas partes del espacio de características (feature space) aparece en las slides del curso. Específicamente, para este caso tenemos:

1. $P_1 \rightarrow \xi_k > 2$
2. $P_2 \rightarrow \xi_k = 2$
3. $P_3 \rightarrow \xi_k = 1$
4. $P_4 \rightarrow 1 < \xi_k < 2$
5. $P_5 \rightarrow \xi_k = 0$
6. $P_6 \rightarrow \xi_k = 0$

4. (*16 puntos*) Árboles de Decisión

Considere los siguientes datos en una dimensión denominada **f**, los cuales pertenecen a las clases cruz o círculo, tal como muestra la siguiente figura:



Usando estos datos como entrenamiento y la métrica de Ganancia de Información, construya un árbol de decisión que permita separar los registros de la figura. Para esto considere como posibles atributos para los nodos del árbol los siguientes valores de **f**:

Atributo 1: $\mathbf{f} \leq 12.5$

Atributo 2: $\mathbf{f} \leq 6.5$

Atributo 3: $\mathbf{f} \leq 2.5$

Atributo 4: $\mathbf{f} \leq 1.0$

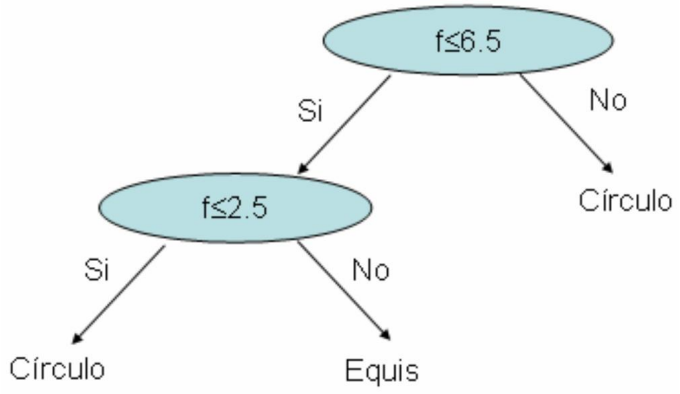
Solución:

- Para el atributo 1 los subgrupos formados son:(5-o,3-x), (0-o,0-x)
- Para el atributo 2 los subgrupos formados son:(2-o,3-x), (3-o,0-x)
- Para el atributo 3 los subgrupos formados son:(2-o,0-x), (3-o,3-x)
- Para el atributo 4 los subgrupos formados son:(1-o,0-x), (4-o,3-x)

Por simple inspección es posible observar que los subgrupos formados son más homogéneos para el atributo 2. Podemos verificar esta observación al realizar los cálculos de Ganancia de Información (GI):

- (5-o,3-x) ; (0-o,0-x) $\rightarrow$ GI=0
- (2-o,3-x) ; (3-o,0-x) $\rightarrow$ GI=0.9544-0.6068=0.3476
- (2-o,0-x) ; (3-o,3-x) $\rightarrow$ GI=0.9544-0.75=0.2044
- (1-o,0-x) ; (4-o,3-x) $\rightarrow$ GI=0.9544-0.8621=0.0923

Por tanto el nodo raíz es el atributo 2. Al usar este atributo en la raíz, el set de datos se divide en los subgrupos (2-o,3-x), (3-o,0-x). La rama derecha esta lista pues es homogénea, por tanto, sólo nos queda separar la rama izquierda, vale decir, (2-o,3-x). En este caso, por inspección es fácil verificar que el atributo 3 permite separar en forma perfecta estos subgrupos. En este caso la ganancia de información es igual a la entropía del subgrupo, vale decir, 0.9710. Por tanto, el árbol final queda:


f≤6.5
Si
No
Círculo
f≤2.5
Si
No
Círculo
Equis