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

13 de Noviembre, 2018

Interrogación 3

Tiempo: 90 minutos, SIN APUNTES

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

Nombre:_____

1. (*16 puntos*) Responda las siguientes afirmaciones o consultas, fundamentando cada respuesta con una breve frase. En lo posible utilice el espacio asignado.

1.1. Para problemas de clasificación binaria y atributos binarios, el algoritmo visto en clases para el entrenamiento de un árbol de decisión, conocido originalmente como ID3, garantiza encontrar el árbol que alcanza la menor tasa de error de clasificación en el set de entrenamiento.

Solución: *F, la estrategia codiciosa (greedy) de entrenamiento no garantiza encontrar un óptimo global.*

1.2. En general, la estimación del error de un clasificador en el set de entrenamiento entrega una estimación optimista del error real de este clasificador.

Solución: *V, es esperable que el error en los datos que el algoritmo procesa durante entrenamiento sea menor que el que obtendrá en datos nuevos, ejemplo, un set de test.*

1.3. En un set de datos S, los atributos A, B y C son usados para construir un árbol de decisión que permite predecir la clase de un atributo binario D. Después de calcular la ganancia de información, G(S,Atributo), se obtiene que G(S,A)=0.1, G(S,B)=0.3 y G(S,C)=0.25, por tanto el atributo B es usado como nodo raíz del árbol.

Solución: *V, se elige el atributo con mayor ganancia de información, por tanto, atributo B.*

1.4. Si un algoritmo alcanza un 100% de efectividad en el set de entrenamiento, entonces se garantiza que la hipótesis respectiva tendrá un alto grado de generalidad para clasificar nuevas instancias.

Solución: *F, puede que exista un problema de sobreajuste.*

1.5. Mientras mayor sea la capacidad de representación de un algoritmo de aprendizaje de máquina (mayor espacio de hipótesis), menor será la probabilidad de sufrir problemas de sobreajuste.

Solución: *F, un mayor espacio de hipótesis tiene mayor capacidad de modelación, por ende, mayor proabilidad de modelar ruido o sobreajustar los datos de entrenamiento.*

1.6. 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: *F, 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.*

1.7. Después de entrenar un SVM, los resultados arrojados no son satisfactorios. Una posible alternativa para mejorar el rendimiento es modificar el valor inicial de los coeficientes del hiperplano con que se inicializa el clasificador SVM.

Solución: *F, no hay un valor inicial de los coeficientes del plano separador.*

1.8. Después de entrenar un SVM, los resultados arrojados no son satisfactorios. Una posible alternativa para mejorar el rendimiento es aumentar el valor de la constante que pondera las variables Slack del clasificador.

Solución: *V, ver respuesta a la pregunta 1.6.*

1.9. Si un modelo lineal y otro cuadrático modelan igualmente bien los datos, uno debería preferir el cuadrático.

Solución: *F, según Occam's razor siempre preferir modelo más simple, en este caso el lineal pues tiene menos parámetros.*

1.10. En cierto problema de aprendizaje de máquina con árboles de decisión, se decide extender una de las ramas de un árbol D1 agregando un nodo adicional que aplica un nuevo test sobre uno de los atributos originales del problema (obs: se extiende sólo una rama), con lo cual se obtiene un árbol D2. De acuerdo a lo anterior, podemos decir que el árbol de decisión D2 tiene un espacio de hipótesis mayor al árbol D1.

Solución: *F, no se agregan nuevos atributos o algún test que no fuera considerado en el espacio original.*

1.11. En el mismo problema anterior, podemos inferir que el árbol de decisión D2 tendrá al menos el mismo número de reglas lógicas que el árbol D1, en otras palabras, en la representación equivalente de árbol de decisión utilizando disjunciones de conjunciones, el número total de conjunciones de D1 y D2 satisfacen: $conjunciones(D1) \leq conjunciones(D2)$.

Solución: *V, la bifurcación que se agrega en el árbol D2 crea una nueva conjunción.*

1.12. En términos de técnicas para implementar un clasificador de múltiples clases en base a un clasificador binario, en general, la estrategia one-vs-one implica una complejidad computacional mayor a la estrategia one-vs-rest.

Solución: *V, en este caso la complejidad computacional está dominada por el número de modelos que se necesita entrenar, en este caso $O(n^2)$ vs $O(n)$.*

1.13. En términos de técnicas para implementar un clasificador de múltiples clases en base a un clasificador binario, en general, la estrategia one-vs-one puede ser resultar más difícil de implementar que la estrategia one-vs-rest debido a problemas de datos no balanceados durante el entrenamiento (hint: balanceados respecto a los rótulos de las clases).

Solución: *F, justo lo opuesto, la estrategia one-vs-rest puede ser difícil de implementar por problemas de balance de datos.*

1.14. La extensión de un clasificador SVM binario al caso de múltiples clases se basa en una formulación de entrenamiento conjunto de clasificadores siguiendo una estrategia del tipo one-vs-rest.

Solución: *V, se implementa asignando un hiperplano a cada clase según una estrategia one-vs-rest.*

1.15. Al aumentar significativamente el número de ejemplos del set de entrenamiento generalmente el error en el set de validación disminuye y en el set de entrenamiento aumenta.

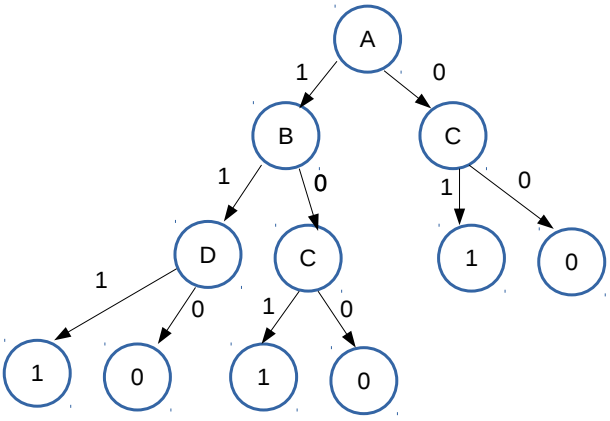
Solución: *V, en general, un número significativo de datos de entrenamiento implica un modelo con mejor capacidad de generalización, por ende menor error en set de validación, y también un set más difícil de modelar, por tanto mayor error en set de entrenamiento.*

1.16. Un algoritmo de aprendizaje de máquina puede ser útil en una situación en que se requiera actualizar automáticamente la lista de operarios del departamento de ventas de cierta compañía.

Solución: *F, éste es un problema determinístico, donde no hay necesidad de predicciones, sino una consulta directa a la base de datos correspondiente.*

2. (*16 puntos*) Árbol de decisión

a. (*8 pts*) Bosqueje un árbol de decisión que represente la siguiente función lógica para el caso de clasificación binaria con atributos binarios : $(A == 1 \wedge B == 1 \wedge D == 1) \vee \neg(A == 1 \wedge B == 1 \wedge D == 0) \vee (A == 1 \wedge B == 0 \wedge C == 1) \vee \neg(A == 1 \wedge B == 0 \wedge C == 0) \vee (A == 0 \wedge C == 1) \vee \neg(A == 0 \wedge C == 0)$.
Obs: $\neg X$ simboliza negación de X.
Ayuda: recuerde que en el contexto de un árbol de decisión: $X == 0$ **no** representa el valor lógico 0, sino el cumplimiento de la condición indicada, en este caso, que la variable X toma el valor 0.



b. (*8 pts*) Como vimos en clases, una mejora a los árboles de decisión es construir lo que se denomina un *random forest*. En forma esquemática, ya sea mediante pseudocódigo o un diagrama de flujo, describa las extensiones que es necesario aplicar al algoritmo ID3 de entrenamiento de un árbol de decisión, para realizar el entrenamiento de un *random forest*. Para facilitar la implementación de su pseudocódigo o diagrama de flujo, considere la función o bloque **tree=ID3(X,Y)**, que implementa el algoritmo original ID3 recibiendo como entrada los atributos **X** y rótulos **Y**, y entregando como salida el modelo entrenado **tree**.
Hay muchas formas de implementar el pseudocódigo, las cuales deberían ser variantes de los siguiente:

```
#Pseudocódigo para entrenar random forest
function randomForest(X,Y,nTrees)

nTotalInstancias=sizeRows(X)
nTotalAtributos=sizeCols(X)

#Asume entrenar cada árbol con 80% de instancias y 80% de atributos.
nInstancias2Train=0.8*nTotalInstances
nAtributos2Train=0.8*nTotalAtributos

for i=1:nTrees
    #Randomly select instancias y atributos del set de datos original.
    [x,y]=selectRows(X,Y,nInstancias2Train)
    [x,y]=selectCols(x,y,nAtributos2Train)
    tree = ID3(x,y)
    randomForestTrees[i]=tree
end
return(randomForestTrees)
```

El pseudocódigo anterior, considera seleccionar conjunta y aleatoriamente tuplas (rows) y atributos (cols) del set de datos original, sin embargo, como vimos en clases, es posible implementar un random forest seleccionando sólo tuplas o atributos.

3. (*16 puntos*) SVMs

a. (*8 pts*) Según lo visto en clases para el caso base de clasificación binaria y datos linealmente separables, la formulación de SVM conlleva al siguiente problema de optimización:

$$\underset{w,c}{\operatorname{argmax}} \left\{ \frac{1}{||w||} \min_k \{z_k g(x_k)\} \right\} \tag{1}$$
$$\text{sujeto a: } z_k(w\ x_k + c) > 0, \quad k = 1 \ldots n$$

o equivalentemente al escalar el mínimo margen según $\min_k \{z_k g(x_k)\} = 1$:

$$\underset{w,c}{\operatorname{argmax}} \frac{1}{||w||} \tag{2}$$
$$\text{sujeto a: } z_k(wx_k + c) > 1, \quad \forall k \in TS \qquad (TS : \text{trainint set})$$

- Indique el problema de optimización equivalente, si en lugar de escalar el mínimo margen a 1, es decir, $\min_k \{z_k g(x_k)\} = 1$, éste se escala a una constante arbitraria $\gamma > 0$.

$$\underset{w,c}{\operatorname{argmax}} \frac{\gamma}{||w||}$$
$$\text{sujeto a: } z_k(wx_k + c) > \gamma, \quad \forall k \in TS \qquad (TS : \text{trainint set})$$

- Demuestre que bajo este problema de optimización modificado se obtiene como resultado el mismo plano separador del problema original en la ecuación (1).

El plano $wx_k + c = 0$ es equivalente al plano $\frac{1}{\gamma}(wx_k + c) = 0$. Reemplazando en la ecuación (2):

$$\underset{w,c}{\operatorname{argmax}} \frac{1}{||\frac{1}{\gamma} * w||}$$
$$\text{sujeto a: } z_k \frac{1}{\gamma}(wx_k + c) > 1, \quad \forall k \in TS \qquad (TS : \text{trainint set})$$

Utilizando el hecho que $\gamma > 0$ y que para una constante α la norma l_2 cumple que: $||\alpha * w|| = \alpha||w||$

$$\underset{w,c}{\operatorname{argmax}} \frac{\gamma}{||w||}$$
$$\text{sujeto a: } z_k(wx_k + c) > \gamma, \quad \forall k \in TS \qquad (TS : \text{trainint set})$$

b. (*8 pts*) Se tiene el siguiente set de datos:

a_1	a_2	Clase
1	3	-
1	1	-
3	2	+
3	4	+

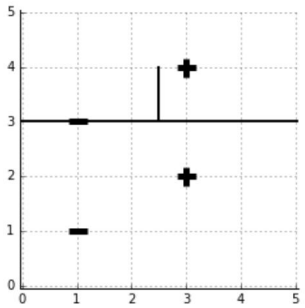
Considerando el hiperplano de solución $w_2a_2 + w_1a_1 + w_0 = 0$, responda las siguientes consultas:

- Si $(w2, w1, w_0) = (1, 0, -3)$, ¿Cuál es el valor de la pérdida para cada instancia de entrenamiento, en otras palabras, cuál es el valor de la variable de slack ξ_k asociada a cada instancia de entrenamiento?. Para su respuesta, considere el caso de clasificación binaria con variables de slack, mínimo margen de 1 y sin función de kernel. En otras palabras, considere la siguiente formulación:

$$\underset{w=\{w_2,w_1\},w_0}{\operatorname{argmin}} \frac{1}{2}||w||^2 + C\sum_{k=1}^{K}\xi_k;$$
$$\text{sujeto a: } z_k(w_2x_2 + w_1x_1 + w_0) \geq 1 - \xi_k,\ \xi_k \geq 0,\ k = 1 \ldots n.$$

  donde $C > 0$ controla el trade-off entre penalización y margen.

a_1	a_2	Slack ?
1	3	1
1	1	0
3	2	2
3	4	0

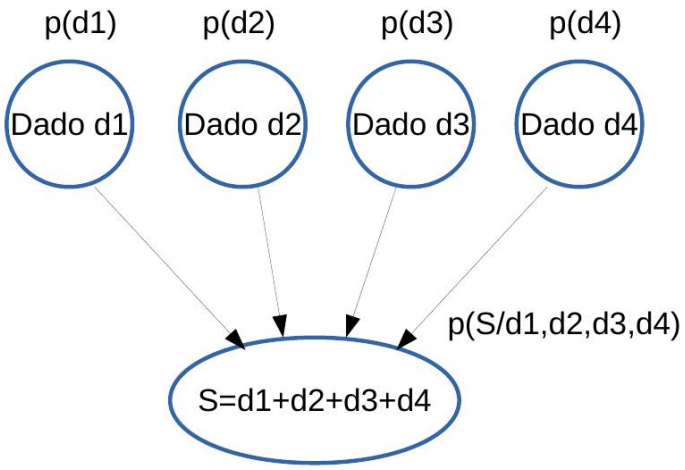


- Si el valor de la constante que múltiplica a las variables de slack ξ_k es $C = 1$, ¿Cuál es el valor total de la función de pérdida para el plano indicado?

$$\tfrac{1}{2}||(0,1)||^2 + (1 + 0 + 2 + 0) = 3.5$$

4. (*16 puntos*) Razonamiento probabilísto usando regla de Bayes

a. (*6 pts*) Cierto juego de azar consiste en lanzar 4 **dados regulares** y realizar una apuesta en torno al valor de la suma resultante. Dibuje una red de Bayes que permita modelar este problema e indique las funciones de probabilidad relevantes, en su notación utilice d_i para indicar el evento de lanzar el dado i y como s el valor de la suma resultante.



b. (*10 pts*) Se requiere implementar un clasificador que determine el grado de éxito que tendrá la apertura de un nuevo local de una cadena de restaurants. Luego de un estudio inicial se determina que para la aplicación es suficiente medir el estado de éxito E en términos de 5 niveles: *muy bajo, bajo, regular, bueno, muy bueno*. Para la realizar la predicción, la cadena de restaurant cuenta con información relevante de otros 5000 locales, la cual incluye los siguientes atributos:

- A_1 : flujo de personas en el sector, representado como una variable discreta con 3 posibles valores: *bajo, medio, alto*.
- A_2 : tamaño del local, representado como una variable discreta con 2 posibles valores: *express o mega*.
- A_3 : nivel de costo, medido como una variable que integra información de costos de construcción, transportes, mantenimiento etc. A_3 es representado como una variable continua en unidades U, es decir, $A_3 \in \mathbb{R}$ (valores negativos pueden ser explicados por desviaciones negativas respecto de un valor esperado de costos, por ejemplo, situaciones como atrasos o cierre temporal del local).
- E : nivel de éxito del local, con valores *muy bajo, bajo, regular, bueno, muy bueno*.

  Indique los pasos y ecuaciones relevantes para implementar en esta aplicación un clasificador del tipo Naive Bayes.

  ¿Cuál es la ecuación relevante para predecir el nivel de éxito de un nuevo local para el cual se tiene: A_1=*bajo*, A_2=*mega*, A_3=*430,84U*?

Solución:
La ecuación relevante de clasificación es $P(E|A_1, A_2, A_3)$, que de acuerdo a la aproximación de Naive Bayes queda: $P(E)P(A_1|E)P(A_2|E)P(A_3|E)$. Por tanto es necesario estimar las siguientes funciones de probabilidad:
$P(E)$
$P(A_1|E = muy\_bajo), P(A_1|E = bajo), P(A_1|E = regular), P(A_1|E = bueno), P(A_1|E = muy\_bueno).$
$P(A_2|E = muy\_bajo), P(A_2|E = bajo), P(A_2|E = regular), P(A_2|E = bueno), P(A_2|E = muy\_bueno).$
$P(A_3|E = muy\_bajo), P(A_3|E = bajo), P(A_3|E = regular), P(A_3|E = bueno), P(A_3|E = muy\_bueno).$

Para el caso de predicir A_1=*bajo*, A_2=*mega*, A_3=*430,84U*, se debe resolver:

$$\underset{e_i}{\operatorname{argmax}}\, P(E = e_i)P(A_1 = bajo|e_i)P(A_2 = mega|e_i)P(A_3 = 430,84U|e_i),$$

donde $e_i \in \{muy\_bajo, bajo, regular, bueno, muy\_bueno]\}$. Por tanto, las probabilidades relevantes son:

$P(E = muy\_bajo)P(A_1 = bajo/E = muy\_bajo)P(A_2 = mega|E = muy\_bajo)P(A_3 = 430,84U/E = muy\_bajo)$
$P(E = bajo)P(A_1 = bajo/E = bajo)P(A_2 = mega|E = bajo)P(A_3 = 430,84U/E = bajo)$
$P(E = regular)P(A_1 = bajo/E = regular)P(A_2 = mega|E = regular)P(A_3 = 430,84U/E = regular)$
$P(E = bueno)P(A_1 = bajo/E = bueno)P(A_2 = mega|E = bueno)P(A_3 = 430,84U/E = bueno)$
$P(E = muy\_bueno)P(A_1 = bajo/E = muy\_bueno)P(A_2 = mega|E = muy\_bueno)P(A_3 = 430,84U/E = muy\_bueno)$