



Examen

Pregunta 1

Verdadero o Falso: Argumente su respuesta en cualquiera de los dos casos.

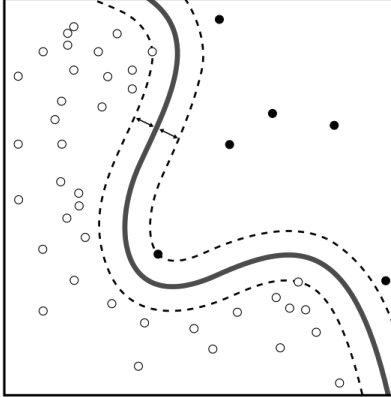
- a) Sea $\Sigma = \{\forall x (P(x) \rightarrow x = a)\}$. Existe una fórmula atómica ψ , que sólo menciona al predicado P , tal que $\Sigma \cup \{\psi\}$ es inconsistente. (Ayuda: las formulas atómicas no contienen conectivos lógicos ni cuantificadores.) **(0.75 pts.)**
- b) $\text{DFS}(h)$ es una versión del algoritmo de búsqueda en profundidad DFS que recibe una función heurística h como parámetro. Al expandir un nodo, ordena sus sucesores por h y los agrega a la frontera de búsqueda de tal forma que un sucesor de menor h queda al tope del stack. Si h^* es la heurística perfecta (es decir, predice el costo exacto para todo nodo), entonces $\text{DFS}(h^*)$ es óptimo y ejecuta en tiempo lineal en el largo de la solución cuando el problema es uno de costos uniformes (es decir, todas las acciones tienen el mismo costo asociado). **(0.75 pts.)**
- c) Considere el programa:

```
poss(soplar(V),S) :- vela(V), holds(encendida(V),S).  
poss(encender(V),S) :- vela(V), \+ holds(encendida(V),S).  
efecto_negativo(soplar(V),encendida(V)).  
efecto_positivo(encender(V),encendida(V)).  
holds(F,do(A,S)) :- holds(F,S),\+ efecto_negativo(A,F).  
holds(F,do(A,S)) :- efecto_positivo(A,F).  
legal(s0).  
legal(do(A,S)) :- legal(S),poss(A,S).  
  
vela(V) :- member(V,[v1,v2]).  
holds(encendida(v1),s0).
```

La segunda respuesta dada por Prolog a la consulta `legal(S),holds(encendida(v2),S)` es `S = do(soplar(v1),do(encender(v2), s0))`. **(0.75 pts.)**

- d) Suponga un problema de búsqueda con adversarios en donde el adversario siempre puede ejecutar una única acción. Sobre este problema, Minimax se comporta esencialmente igual que DFS (es decir, expanden los mismos nodos, en el mismo orden y ocupan recursos similares), cuando a este último se le pasa la condición objetivo “ganar el juego”. **(0.75 pts.)**
- e) El algoritmo de descenso de gradiente estocástico permite entrenar clasificadores utilizando una gran cantidad de datos, pero su convergencia puede ser lenta, debido al gradiente parcial que es utilizado para el descenso. Una manera de aliviar este problema es usar la técnica de momentum, que permite evitar las oscilaciones dadas por el ruido inherente de un minibatch. **(0.75 pts.)**
- f) Una manera de mejorar los árboles de decisión, es modificando el procedimiento de testeo realizado en cada nodo. En particular, con el fin de reducir el sobreentrenamiento, en vez de usar tests basados en umbrales para variables numéricas, es posible usar un clasificador de K-vecinos cercanos. **(0.75 pts.)**

- g) Siempre es posible transformar un problema de clasificación multiclase, en una serie de problemas de clasificación binarios. En particular, dado un problema con K clases, es necesario entrenar al menos $\frac{K \times (K-1)}{2}$ clasificadores binarios (uno por cada par de clases), para luego tomar la decisión en base a qué clasificador obtuvo más *votos*. **(0.75 ptos.)**
- h) Un *Soft-margin SVM* no puede clasificar perfectamente el problema presentado en la siguiente figura **(0.75 ptos.)**:



Pregunta 2

Algunos problemas de búsqueda parecen ser una combinación de otros dos problemas de búsqueda. Como ejemplo, considere el problema de descargar y llevar a destino containers en un puerto. En esta situación los containers se pueden ubicar unos encima de otros (similar a un problema de bloques), pero, a su vez, los containers deben moverse a diferentes destinos (similar a un problema logístico). No es sorprendente, entonces, que para estos problemas sea posible construir dos heurísticas admisibles h_1 y h_2 , cada una de las cuales se “enfoca” en una parte del problema, prácticamente ignorando a la otra.

- a) Demuestre que cuando h_1 y h_2 son admisibles, entonces $\max\{h_1, h_2\}$ también lo es. **(1 pto.)**
- b) Existen situaciones en que $\max\{h_1, h_2\}$ es una “mala heurística”. Encuentre una de estas situaciones, definiendo un problema de búsqueda en donde se aprecie que A^* no es guiado bien por tal heurística. Inspírese en el ejemplo de arriba y ejerza la claridad y la elegancia al explicar. **(3 ptos.)**
- c) Explique con precisión cómo podría usar A^* con h_1 como heurística principal y a h_2 como un “quebrador de empates”. Hágalo de tal forma el algoritmo siga siendo óptimo. (No debe demostrar que lo sigue siendo.) **(2 ptos.)**

(Dato curioso e inútil para resolver esta pregunta: En 2010, Gabi Röger y Malte Helmert estudiaron seriamente cómo combinar heurísticas para buscar. Concluyeron que desempatar no es lo mejor. Su propuesta es sencilla, pero no es el momento de explicarla... Su artículo se llama “*The More, the Merrier: Combining Heuristic Estimators for Satisficing Planning*”)

Pregunta 3

Para representar el conocimiento en un árbol genealógico se utilizan dos predicados:

- $Pr(x, y)$: expresa que x es progenitor de y .
- $An(x, y)$: expresa que x es ancestro de y .

Para definir el predicado An en términos de Pr es posible utilizar la siguiente fórmula:

$$\psi = \forall x \forall y \forall z (Pr(x, y) \wedge An(y, z) \rightarrow An(x, z)) \wedge \forall x \forall y (Pr(x, y) \rightarrow An(x, y))$$

- a) Use resolución para demostrar que “si a no es ancestro de b entonces no hay un camino de largo 2 entre a y b a través de la relación Pr ”. **(4 ptos.)**
- b) Generalice la resolución anterior. Específicamente, diga qué pasos de resolución son necesarios para demostrar que no hay un camino de largo n entre a y b si a no es ancestro de b , para cualquier n . **(2 ptos.)**

Pregunta 4

Un *Double Soft-margin SVM* es un algoritmo de clasificación similar a un *Soft-margin SVM*, pero que impone dos restricciones sobre el margen funcional: una sobre su valor mínimo (*Soft-margin SVM*) y una sobre el máximo. Tomando esto en consideración, responda las siguientes preguntas:

- a) Defina el problema de aprendizaje para un *Double Soft-margin SVM*, como un problema de minimización cuadrático con restricciones lineales. **(3 ptos.)**
- b) Modifique el problema de aprendizaje definido en el ítem anterior, planteándolo esta vez como un problema de minimización cuadrático sin restricciones. **(3 ptos.)**

Pregunta 5

Considere una red convolucional profunda de 8 capas, que es utilizada para resolver un problema binario de clasificación de imágenes, consistente en decidir la aparición o no de personas en ellas. La red tiene 5 capas convolucionales (*stride* de 1, sin *padding*) con 128 filtros de 5x5 cada una, seguida por 3 capas *fully-connected* con 256, 256 y 1 neurona, respectivamente. En base a esta información, responda las siguientes preguntas:

- a) ¿Cuál es la función de las capas convolucionales en esta red? ¿Por qué no basta con usar sólo capas *fully-connected*. **(2 ptos.)**
- b) Asumiendo que las imágenes tienen una resolución de 224x224, indique la cantidad total de parámetros que utiliza esta red. **(2 ptos.)**
- c) Después de realizar un análisis exploratorio, se verificó que en las imágenes utilizadas para entrenar la red, las cabezas de las personas tienen un diámetro aproximado de 15 píxeles. ¿En que capa esperaríamos observar entonces filtros (neuronas) que se activen fuertemente con cabezas humanas? **(2 ptos.)**