

Preguntas parcial 2- IA

1.- Componentes de un juego.

Un juego es cualquier situación de decisión con varios agentes (jugadores), gobernada o regida por un conjunto de reglas y con un resultado bien definido, caracterizada porque ninguno de los jugadores con su sola actuación puede determinar el resultado (interdependencia estratégica).

Los componentes de un juego son:

- Número de jugadores
- Juegos de información perfecta vs juegos de información imperfecta (ajedrez vs póker)
- Existencia de movimientos de azar
- Orden de actuación de los jugadores
- Existencia o no de pagos colaterales (equilibrio Nash)
- Y juegos de suma nula vs juegos de suma no nula

En un juego, cada jugador intenta conseguir el mayor beneficio para sus intereses, la solución de un juego es la determinación de una sucesión de actuaciones que indican a cada jugador que resultado puede esperar y cómo alcanzarlo. Por tanto, un juego puede emplearse como un problema de maximización, aunque finalmente y en muchos casos solo pueda llegarse a una satisfacción.

2.- Qué es el factor de ramificación y cómo afecta a la complejidad de un juego? Describe en líneas generales el algoritmo minimax y el de la poda alfa-beta

El algoritmo minimax es un método de decisión para minimizar la pérdida máxima esperada en juegos con adversario y con información perfecta, siendo un algoritmo recursivo. Es decir, una estrategia óptima según el criterio minimax es una que minimiza el daño máximo que puede hacer el contrincante. Este algoritmo sirve para juegos bipersonales de suma nula con información perfecta.

Principios fundamentales de la teoría de juegos:

- Cada jugador hace la acción mejor posible.
- Cada jugador sabe que su contrincante está también haciendo la acción mejor posible.

Notación min-max:

- MAX: el primer jugador (nodos max): busca el máximo beneficio.
- MIN: el segundo jugador (nodos min): busca el mínimo beneficio.

Siendo en cada nivel del árbol min o max, con el objetivo de encontrar un conjunto de movimientos accesible que dé como ganador max. Una estrategia ganadora para max es un subárbol en el que todos los nodos terminales son ganadores.

Reglas del minimax:

- El valor de un nodo de la frontera de búsqueda es igual al de su evaluación estática.
- En caso contrario:
 - Si el nodo es un nodo MAX, entonces el valor del nodo es igual al máximo de los valores de sus nodos sucesores.
 - Si el nodo es un nodo MIN, entonces el valor del nodo es igual al mínimo de los valores de sus nodos sucesores.

En la poda alfa-beta hay 2 parámetros alfa y beta, que describen los límites sobre los valores que aparecen a lo largo del camino:

- Alfa: el valor de la mejor opción (el más alto) que se ha encontrado hasta el momento en cualquier punto del camino, para MAX.
- Beta: el valor de la mejor opción (el más bajo) que se ha encontrado hasta el momento en cualquier punto del camino, para MIN.

La búsqueda alfa-beta actualiza los valores de alfa y beta según se va recorriendo el árbol y termina la recursión cuando encuentra un nodo peor que el actual valor alfa o beta.

Al principio en la poda alfa-beta toma los valores -infinito para alfa y +infinito para beta. Desde un nodo MAX, iremos actualizando el valor de alfa a medida que vayamos explorando cada arista de ese nodo.

Para actualizar el valor de alfa comparamos el valor obtenido para cada arista con el valor de alfa, entonces si el valor obtenido es mayor al de alfa, se actualizará con el nuevo valor en alfa, si estamos en un nodo MAX. Solo nos interesa proseguir la búsqueda desde ese nodo si podemos mejorar alfa.

En caso de estar en un nodo MIN lo mismo, pero obteniendo el mínimo para beta, por tanto, beta almacenara el peor valor (para MAX) encontrado por MIN hasta el momento. Solo nos interesa proseguir la búsqueda desde ese nodo si podemos mejorar beta.

¿Cuándo se realiza poda en la búsqueda alfa-beta?

- Cuando el valor de alfa sea mayor a beta ($\alpha > \beta$)
- O cuando alfa y beta valgan lo mismo ($\alpha == \beta$)

Y cuando se realiza poda no se recorre ese subárbol podado. De tal forma que si $\beta \leq \alpha$ se realizara poda, en caso contrario proseguirá la búsqueda. Puede que el valor no sea el mismo que haciéndolo con minimax completo, pero conduce a la selección del mismo movimiento.

Caso en el que se puede representar el árbol completo del juego:

Cada nodo hoja se etiqueta 1, -1, 0, si es ganador para MAX, para MIN o empate. Con el objetivo de encontrar un conjunto de movimientos accesible que de como ganador a MAX. Se propagan los valores de las jugadas terminales de las hojas hasta la raíz.

Por ejemplo, el juego simple de tic-tac-toe es demasiado complejo para dibujar el árbol de juegos entero, pero por ejemplo el 3 en raya de un 3x3 si es posible de dibujar el árbol completo.

Hoy en día es imposible resolver los juegos complejos, ya que es imposible la exploración total hasta la terminación. Una alternativa es encontrar una buena jugada inmediata.

Con juegos en los que intervienen el azar:

Por ejemplo, MIN y MAX van a realizar un movimiento, primero tienen que lanzar un dado. En este caso, podemos imaginar que cada vez que se lance el dado, un jugador ficticio, al que se llamara DADO, realiza un movimiento, que además esta determinado por el azar. En el caso de 1 dado, todos los posibles resultados son equiprobables, aunque la componente de azar puede estar sujeta a la probabilidad.

3.- ¿Que problemas plantea el cálculo de predicados en la resolución de problemas de IA?

En el cálculo de predicados puede presentar 2 tipos de problemas:

- **Problemas Semánticos**

- Es difícil expresar todo en formulas, como por ejemplo 2 averías simultaneas es menos corriente que una avería grave (metaconocimiento), igualdad de objetos, heurística, sentido común, etc.
- El razonamiento temporal también es difícil en la lógica de predicados.
- No se puede razonar acerca de predicados.
- No podemos aportar información incompleta y/o imprecisa (eso incluye la probabilidad)
- Tampoco se puede usar excepciones
- Y por último la monotonía, es decir si se deduce una cosa de un conjunto de acciones esa cosa ya queda deducida para siempre, incluso si se amplía el conjunto de acciones, de tal forma que impide revisar razonamientos
- **Problemas computacionales**
 - Consistencia (solidez): lo que se demuestra como verdadero lo es realmente. Por ejemplo, si tengo un conjunto de axiomas pequeños puede, pero si tengo un conjunto de axiomas grandes no se podría demostrar. Esto se hace por refutación por solución, pero no es seguro que sea consistente.
 - Tampoco es completo (completitud): si una cosa es verdadera puede demostrarse, no es seguro saber con certeza. Ejemplo si se demuestra que una cosa es verdadera lo es, pero si no puedo demostrarlo no puedo decir que sea falsa, en este sentido la lógica de predicados es semidecidible.
 - Y también tiene una complejidad computacional, de tal forma que la complejidad en tiempo y espacio del razonamiento aumenta exponencialmente

4.- Modelos de conocimiento heredable ¿Qué tipo de conocimiento organizan las redes semánticas? Describir en líneas generales el concepto de “frame”.

Este modelo de conocimiento heredable tiene almacenamiento jerarquizado, redes semánticas, frames guiones, representaciones orientadas a objetos, etc.

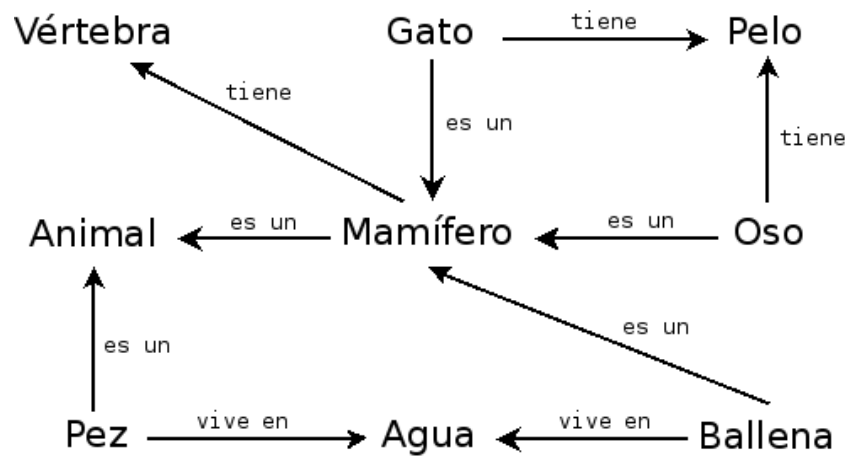
Hay muchos modelos, pero los 2 modelos de conocimiento heredable más conocidos son:

- Las redes semánticas
- Y los frames o marcos

Las **redes semánticas** están destinadas a representar o comprender el lenguaje natural, formando parte de un concepto más general, teniendo un formalismo muy limitado para dominios

más complejos, también está limitado para tratar con formas de inferencia sofisticada. Este modelo ha tenido muchas variantes, pero las que se usa hoy en día son los modelos isa. Las redes semánticas se utilizaron para hacer diccionarios electrónicos y es precursor de los frames.

Ejemplo:



Un **frame o marco** es una estructura de datos que representa una situación estereotipada. Cada marco se caracteriza por un conjunto de campos que se asocian en general a los atributos y que en conjunto sirven para identificar los marcos. Estos marcos son concebidos para tareas de reconocimiento, haciendo que la información recibida active ciertos marcos y esta a su vez otros marcos. También los marcos se pueden organizar jerárquicamente, muy parecidas a las redes semánticas, permitiendo la herencia.

Ejemplo:

El frame Pájaro, tiene los slots:

- Es_un: animal
- Forma_moverse: volar
- Activo_durante: día

El frame pingüino es una subclase de pájaro, con los slots:

- Es_un: Pájaro
- Color: blanco_y_negro



participa
aquí

- Forma_moverse: andar
- Tamaño: mediano

5.- Estructura y componentes de un sistema experto

Un sistema experto, se pueden definir como sistemas expertos basados en el conocimiento, es decir es un programa que emplea conocimiento para resolver un problema concreto en un dominio determinado, capaces de comportarse como un experto (humano) en un determinado dominio de actividad:

- Resuelven un problema
- Pueden ser consultados y justifican su razonamiento

En resumen son sistemas que imitan o intentan imitar las actividades de un humano para resolver problemas.

Un sistema basado en el conocimiento necesita 3 componentes:

1. Una base de conocimiento, que contenga el conocimiento necesario sobre el universo del problema a resolver, conteniendo una descripción del mundo (sus objetos y las relaciones entre ellos).
2. Un motor de inferencia, que permita razonar sobre el conocimiento de la base de conocimiento y los datos obtenidos del usuario, que se incorpora a algo que se llama base de hechos, pero no forma parte del sistema experto.
3. Una interfaz de usuario, para la entrada/salida de datos de cada problema.

Los sistemas expertos incorporan además un modelo de explicaciones / justificación.

El proceso de construcción de un SE:

- Extraer el conocimiento experto y como representarlo de una forma abstracta efectiva (bibliografía, entrevistas a expertos reales, etc.).
- Se modela y se adquiere el conocimiento, representando el conocimiento en términos de formalismo utilizando un lenguaje adecuado, para que una máquina pueda procesar.

- Se crea la Base de Conocimiento con el conocimiento adquirido, es decir creando el motor de inferencias y como usar estas estructuras abstractas, para generar información útil en cada caso.

Los principales formalismos de representación tienen como pieza básica el triple objeto-atributo-valor, es la pieza clave para los sistemas expertos.

Ejemplo: procesador-frecuencia-4.5GH

Para los sistemas expertos podemos usar distintos formalismos de representación usando un método de inferencia específico:

- Resolución en Lógica
- Razonamiento hacia adelante y hacia atrás en sistemas de reglas (razonamiento abductivo)
- Herencia en sistemas de frames

Los sistemas expertos más comunes son los basados en reglas, basados en sistemas basados en el conocimiento, donde el conocimiento se añade en formas de reglas y hechos, siendo los más clásicos y comunes. Estos sistemas trabajan mediante la aplicación de reglas, comparación de resultados y la aplicación de nuevas reglas basadas en situación modificada.

Aunque estas reglas o hechos pueden implementarse, como por ejemplo en el cálculo de predicados, no obstante, es más adecuado emplear reglas de producción para las reglas. Siendo la condición de la causa al efecto y del efecto a la causa.

Proceso de construcción:

- Se **extrae el conocimiento** experto (bibliografía, entrevistas a expertos, etc.)
- Se **modela y se adquiere el conocimiento**, utilizando un lenguaje.
- Y se **crea la Base del Conocimiento** con el conocimiento adquirido.

Problemas:

- Es posible que una regla no sea aplicable siempre. Este hecho no permite que la regla sea admitida en un sistema de cálculo proposicional o de predicados, dando lugar a sistemas inconsistentes.

6.- Paradigmas de Aprendizaje Automático.

Aprendizaje automático: Es un programa de computadora que aprende de la experiencia E con respecto a alguna tarea T y alguna medida el rendimiento P, si su desempeño en T, medida por P, mejora con la experiencia E. Es decir, son programas que mejoran su comportamiento con la experiencia.

Aprendizaje de damas:

- T: jugar a las damas.
- P: % de juegos ganados.
- E: partidas jugadas contra una copia de sí mismo

El aprendizaje automático tiene 3 etapas:

1. Definir del problema y su dominio, es decir donde aplicar las técnicas de aprendizaje.
2. Adquirir el conocimiento.
3. Y refinar del conocimiento obtenido, dando una forma útil.

Desde el punto de vista informático el objetivo es siempre determinar una estructura computacional como:

- Una o varias funciones
- Conjunto de reglas
- Programas Lógicos
- Máquinas de estado finito
- Gramáticas formales
- Sistemas para la resolución de problemas

En los paradigmas de aprendizaje tenemos:

- **Aprendizaje memorístico:** Esto en el aprendizaje automático se utiliza en la creación de base de datos, creación de sistemas orientados a objetos, etc.



Invita a otros estudiantes, crea contenido y gana los premios que te alegrarán el verano

- **Aprendizaje deductivo:** Se basa en una teoría y se intenta aplicar a un objeto concreto.
 - Inferencia y deducción empleando modelos Lógicos
 - Razonamiento artificial, es decir sistemas basados en lógica que es capaz de deducir a partir de hechos.
- **Aprendizaje analítico, basado en explicaciones:** Se trata de construir una explicación para cada ejemplo en relación con un concepto dado y generalizar la explicación de modo que pueda emplearse en el futuro.
- **Aprendizaje analógico:** Entender y resolver una situación, con un parecido a otras anteriormente resueltas.
- **Aprendizaje inductivo:** Es el paradigma más amplio estudiado dentro del aprendizaje, hablando en términos muy generales se trata de aprender un concepto o clasificación a partir de ejemplos y contraejemplos. El objetivo es aprender la función f , un ejemplo es un par $(x, f(x))$. El problema es encontrar una hipótesis h tal que $h = f$ sobre los conjuntos de los ejemplos de entrenamiento, una hipótesis estará bien generalizada si puede predecir ejemplos que no conoce.

Uno de los puntos clave del aprendizaje es el tipo de realimentación disponible en el proceso:

- **Aprendizaje Supervisado:** Para cada entrada, se dispone de un supervisor que proporciona una salida deseada, ya sea una clase o un valor a aproximar (clasificación vs regresión).
- **Aprendizaje no supervisado:** No se dispone de una salida deseada en cada entrada, sino que se busca agrupar o clasificar los datos en función de ciertas características. Es decir, construye su propia teoría, para explicar lo que ve.
- **Aprendizaje por refuerzo:** Aprende a partir de la información obtenida al realizar procesos de ensayo-error en los que se obtienen señales de beneficio o coste.

Las herramientas notables en aprendizaje automático son:

- **Técnicas estadísticas**
- **Técnicas de árboles**
- **Análisis cluster:** Es agrupar objetos, son técnicas de aprendizaje no supervisado.
- **Modelos bioinspirados:**
 - Algoritmos genéticos
 - Programación evolutiva



**participa
aquí**



Hstas el
15/06/2023

WUOLAH

- Modelos conexionistas (redes neuronales)

7.- Describir el problema del ruido y el del sobreajuste en aprendizaje automático.

8.- ¿Qué son y como se construyen los árboles de decisión

La inducción de árboles de decisión es uno de los métodos más sencillos y con más éxito para construir algoritmos de aprendizaje.

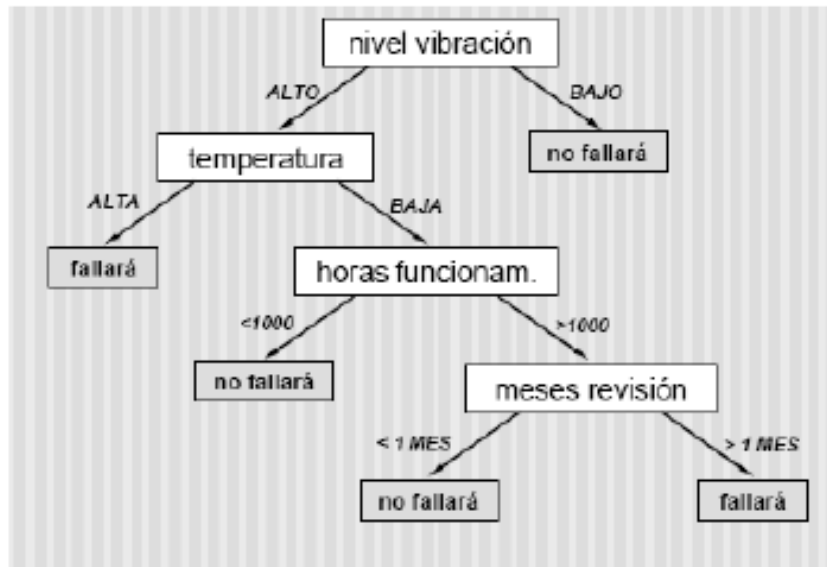
Un árbol de decisión toma como entrada un objeto o una situación descrita a través de un conjunto de atributos y devuelve una decisión del valor previsto de la salida dada la entrada. Es decir, es un árbol cuyo nodo raíz es un conjunto de objetos caracterizados por unos atributos y por un valor de clasificación o de función, y cada arista constituye una pregunta o una decisión sobre el valor de un atributo, de tal forma que los nodos hoja son conjunto de objetos perfectamente clasificados.

Siendo una de las herramientas más potentes que tiene en el aprendizaje inductivo.

Los atributos pueden ser **discretos** o **continuos** y las salidas discretas (clasificación) o continuas (regresión).

Ejemplo: Modelado de la probabilidad de fallo de una máquina

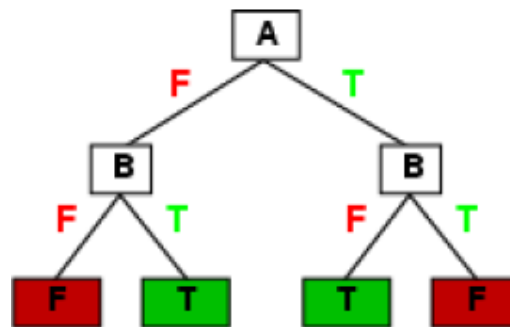
Temperatura	Nivel de vibraciones	Horas de funcionamiento	Meses desde revisión	Probabilidad de fallo
ALTA	ALTO	< 1000	> 1 MES	fallará
BAJA	BAJO	< 1000	< 1 MES	no fallará
ALTA	BAJO	>1000	> 1 MES	no fallará
ALTA	BAJO	< 1000	> 1 MES	no fallará
BAJA	ALTO	< 1000	> 1 MES	no fallará
BAJA	ALTO	>1000	> 1 MES	fallará
ALTA	ALTO	< 1000	< 1 MES	fallará



Los árboles de decisión pueden expresar cualquier función y cualquier clasificación a partir de los atributos de entrada.

Por ejemplo, para funciones booleanas, cada fila de la tabla de verdad se traslada a un camino del árbol:

A	B	A xor B
F	F	F
F	T	T
T	F	T
T	T	F



De forma trivial, hay un árbol de decisión consistente para cualquier conjunto de entrenamiento con un camino asociado a cada ejemplo, pero seguramente no será bueno para generalizar nuevos ejemplos. Es preferible encontrar árboles de decisión más compactos.

Hay múltiples formas de inferir en el árbol, pero las 3 más básicas son:

- **Trivial:** Se crea un camino del árbol por cada instancia de entrenamiento, es decir explora el árbol y se crea una ruta para encontrar un nodo hoja donde tendrá una clasificación del objeto. Problemas, son árboles muy grandes y no funcionan bien con instancias nuevas, porque no recoge bien la compactación de nuevos atributos.
- **Óptimo:** Encuentra el árbol más pequeño posible compatible con todas las instancias (usando el principio de la navaja de Ockham). Problema, obligaría a crear todos los árboles posibles, siendo inviable computacionalmente.
- **Pseudo-óptimo (heurístico):** Selecciona un atributo en cada nivel del árbol en función de la calidad de división que produce, es decir el árbol se va construyendo a partir del conjunto de entrenamiento, preguntando por valores de atributos y llegando a los nodos hoja que estén bien clasificados. Los principales programas de generación de árboles utilizan procedimientos similares (ID3, CART, etc.). Problema que hace una búsqueda

local y no global, obteniendo una óptima, pero podría haber otra aun más óptima.

Para la elección de buenos atributos la idea es un buen atributo que sea candidato a ser utilizado debería dividir el conjunto de ejemplos en subconjuntos que sean o todos positivos o todos negativos, es decir que estén bien clasificados.