

## **Preguntas parcial 2- IA**

### **Componentes de un juego.**

La primera de las propiedades a tener en cuenta será el número de jugadores o agentes involucrados, información de gran vitalidad a la hora de diseñar el algoritmo. Un juego puede ser sin adversario (por ejemplo un 8-puzzle), con 1 adversario (por ejemplo el 3 en raya), o con N adversarios. La siguiente propiedad a conocer es el orden de los movimientos. Saber si por ejemplo los jugadores mueven alternativamente o por azar también es muy importante. Una vez situados los jugadores y sus turnos, hay que saber qué conocimiento tienen éstos. En los juegos los jugadores pueden tener o bien conocimiento perfecto, donde no afecta el azar y todos saben en todo momento lo mismo, o bien conocimiento imperfecto, donde el azar sí puede participar y además hay ocultación de información entre jugadores. Al hecho de que el azar aparezca o no aparezca en alguna de las propiedades anteriores se le conoce como determinismo. También podemos encontrarnos juegos donde un movimiento que te beneficia siempre perjudica al mismo nivel al adversario (equilibrio Nash), o juegos donde ésto no ocurre. Según sus características encontraremos juegos simétricos y asimétricos, de suma cero y de suma no cero, cooperativos, simultáneos y secuenciales, de información perfecta, de longitud infinita, o juegos bayesianos entre otros.

Suma cero: uno gana y el otro pierde

Suma no cero: puede ser que todos ganen o todos pierdan

### **Describir en líneas generales el algoritmo minimax.**

MiniMax es el algoritmo más conocido (y utilizado) para problemas con exactamente 2 adversarios, movimientos alternos ("ahora tú ahora yo"), e información perfecta. Identificaremos a cada jugador como el jugador MAX y el jugador MIN.

MAX será el jugador que inicia el juego, el cual supondremos que somos nosotros, y nos marcaremos como objetivo encontrar el conjunto de movimientos que proporcionen la victoria a MAX (nosotros) independientemente de lo que haga el jugador MIN. Etiquetar los jugadores como máximos y mínimos plasma perfectamente la dualidad de los objetivos de cada uno, además de que así se puede contabilizar perfectamente cómo una situación que mejora para un jugador hace que empeore para el otro. Matizar que

no importa qué jugador deseemos ser, pues bajo nuestro punto de vista siempre calcularemos jugadas positivas a nuestro favor y negativas para el oponente. Y todo ésto nos lleva a que por convenio una jugada vencedora del jugador MAX (lo que vendría a ser un “jaque mate”) tendrá valor infinito, y una jugada del jugador MIN tendrá valor menos infinito. Deberá existir para el juego a resolver una función de evaluación heurística que devuelva valores elevados para indicar buenas situaciones, y valores negativos para indicar situaciones favorables al oponente. Ahora ya podemos ver como para cada movimiento conoceremos su valor numérico en función de su “grado de bondad”, y podremos ser capaces de identificar nuestro mejor movimiento. Además de utilizar la anterior función heurística, usaremos una estrategia DFS de profundidad limitada para explorar el conjunto de jugadas. Como es imposible hacer una exploración exhaustiva de todas las jugadas, se hace una búsqueda limitada en profundidad. Significa que en lugar de estudiar todas las posibles situaciones hasta el fin de la partida, se buscarán por ejemplo todas las situaciones hasta de aquí 3 turnos. Aunque ésto limitará lo que veamos del espacio de búsqueda, y quizás nos incitará a realizar unas jugadas que a la larga sean nefastas. La calidad de nuestras jugadas vendrá determinada por la profundidad a la que lleguemos en cada exploración. Cuanto más profunda sea, mejor jugaremos, pero mayor coste tendrá el algoritmo.

## **Paradigmas de Aprendizaje Automático.**

Es una rama de la Inteligencia Artificial que tiene como objetivo el desarrollar técnicas que permitan que las computadoras puedan aprender.

Es decir, crear programas que sean capaces de generalizar los comportamientos partiendo de una información que no esté estructurada suministrada en forma de ejemplos.

Una máquina puede considerarse inteligente cuando cumple con los siguientes requisitos:

- ❑ Si es capaz de percibir visualmente los objetos que la rodean y reconocer sus formas.
- ❑ Si es capaz de “entender” el lenguaje natural, hablado o escrito, así como de producir respuestas en dicho lenguaje.
- ❑ Elaborar actuaciones de acuerdo con las condiciones cambiantes del entorno y llevarlas a cabo mediante los correspondientes elementos físicos.
- ❑ Cuando puede almacenar información y conocimientos a los que manipula mediante reglas y algoritmos para alcanzar soluciones a los problemas que plantea su funcionamiento.

En el aprendizaje automático entra el término *reconocimiento de formas* en la cual se hace distinción sobre dos aproximaciones:

- Aprendizaje memorístico .
- Aprendizaje deductivo.
- Inferencia y deducción empleando modelos Lógicos
- Razonamiento Artificial

- Aprendizaje analítico, Basado en explicaciones

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 por su parecido con otras anteriormente resueltas.

- Aprendizaje inductivo

- Es el paradigma más ampliamente estudiado dentro del aprendizaje.

- Hablando en términos muy generales se trata de aprender un concepto o una clasificación a partir de ejemplos y contraejemplos.

#### 1. Clasificación no supervisada

Enfoca la clasificación como el descubrimiento de las clases del problema. También se la denomina “clustering”.

Su objetivo es el descubrimiento de grupos de objetos, que con características afines, separen las diferentes clases del problema clasificatorio.

#### 2. Clasificación supervisada

Basándose en un conjunto de entrenamiento, construye un modelo o regla general que se utilizará para clasificar objetos nuevos de los cuales no sepamos su clase.

Este tipo de clasificación ha sido utilizada en áreas como: diagnóstico de enfermedades, concesión o rechazo de créditos de banca, predicción de bancarrota en empresas, etc.

*Aproximaciones Paramétricas:* Se asume un conocimiento a priori acerca de la forma funcional de las distribuciones de probabilidad de cada clase sobre el espacio de representación, las cuales vendrán determinadas por un conjunto finito de parámetros.

*Aproximaciones no Paramétricas:* No supone ninguna forma de las distribuciones de probabilidad sobre el espacio de representación, de modo que el único conocimiento a priori será el correspondiente a la información inducida a partir de un conjunto de muestras controladas.

Aprendizaje por refuerzo:

Se aprende a partir de la información obtenida al realizar procesos de ensayo-error en los que se obtienen “señales” de beneficio/coste.

## **Describir el problema del ruido y el del sobreajuste en aprendizaje automático.**

Para un conjunto con  $m$  puntos, sabemos con toda seguridad que existe un polinomio de grado  $(m-1)$  que se ajusta perfectamente a esos puntos, es decir, que pasa por todos ellos. Pero, aunque siempre podremos realizar este tipo de ajuste, independientemente de la disposición de los puntos, no nos servirá para establecer algún tipo de relación en el conjunto de puntos utilizado. En estos casos diremos que hemos sobreajustado dichos datos. Si, además, los datos se ven afectados por algún tipo de ruido (casos en los que el ajuste a una recta puede ser suficiente), el polinomio de grado  $(m-1)$  se habría ajustado de manera exacta a los datos con ruido. Para evitar este tipo de problemas, a la hora de realizar un ajuste, el número de puntos a utilizar tiene que ser sensiblemente mayor que el grado del polinomio que vamos a utilizar. Y, dado un conjunto suficientemente grande, el principio de la Navaja de Occam nos dice que de entre todos los posibles polinomios con los que podemos realizar el ajuste, el de grado menor será la mejor opción.

El ruido se da cuando hay dos o más ejemplos con la misma descripción en términos de atributos, pero entre ellos se encuentran en diferentes clasificaciones.

El sobreajuste es el efecto de sobreentrenar un algoritmo de aprendizaje con unos datos para los que se conoce el resultado deseado. El algoritmo de aprendizaje debe alcanzar un estado en el que será capaz de predecir el resultado en otros casos a partir de lo aprendido con los datos de entrenamiento, generalizando para poder resolver situaciones distintas. Sin embargo, cuando un sistema se entrena demasiado (se sobreentrena) o se entrena con datos extraños, el algoritmo de aprendizaje puede quedar ajustado a unas características muy específicas de los datos de entrenamiento que no tienen relación causal con la función objetivo. Durante la fase de sobreajuste el éxito al responder las muestras de entrenamiento sigue incrementándose mientras que su actuación con muestras nuevas va empeorando.

## **¿Que problemas plantea el cálculo de predicados en la resolución de problemas de IA?**

- 1.- Problemas semánticos
  - 1.1 Es difícil expresar todo en fórmulas lógicas: heurísticas, metaconocimiento, jerarquía y herencia, igualdad, sentido común, etc.
- 2.- Razonamiento temporal.
- 3.- Razonamiento acerca de predicados
- 4.- Información incompleta y/o imprecisa (vaguedad, probabilidad).
- 5.- Excepciones.

6.- Monotonía

2.- Problemas computacionales

2.1.- Consistencia (solidez, soundness): lo que se demuestra como verdadero

lo es realmente. Refutación por resolución.

2.2.- Completitud: si una cosa es verdadera puede demostrarse.

Refutación

por resolución. Semidecible.

2.3.- Complejidad Computacional (tratabilidad)

## **¿Qué tipo de conocimiento organizan las redes semánticas?**

Una red semántica o esquema de representación en Red es una forma de representación de conocimiento lingüístico en la que los conceptos y sus interrelaciones se representan mediante un grafo. En caso de que no existan ciclos, estas redes pueden ser visualizadas como árboles. Las redes semánticas son usadas, entre otras cosas, para representar mapas conceptuales (Esta técnica sirve para representar el conocimiento de forma gráfica a través de una red de conceptos simbolizados por nodos enlazados entre sí para indicar sus relaciones.) y mentales (diagrama usado para representar palabras, ideas, tareas, lecturas, dibujos, u otros conceptos ligados y dispuestos radicalmente alrededor de una palabra clave o de una idea central.). En un grafo o red semántica los elementos semánticos se representan por nodos. Dos elementos semánticos entre los que se admite se da la relación semántica que representa la red, estarán unidos mediante una línea.

El objetivo de las redes semánticas es desarrollar una infraestructura para generar datos que las computadoras puedan entender, de tal forma que puedan ser compartidos y procesados no sólo por personas sino también por herramientas automatizadas

## **Describir en líneas generales el concepto de “frame”**

Los marcos son una estructura para organizar el conocimiento con énfasis en el conocimiento por omisión. Cada marco representa una clase de elementos de la misma manera que un nodo de clase se emplea para representar tales elementos en una red semántica. Los marcos se emplean para organizar nuestra comprensión básica de las cosas que típicamente son ciertas para algunas clases generales de elementos.

Un marco se divide en campos, llamados Slots. Un slot también se define como un atributo. Un slot define una cualidad de una unidad de conocimiento y que en conjunto sirven para identificar los marcos. Los marcos están especialmente concebidos para tareas de

reconocimiento, la información recibida hace que se activen unos marcos y esta a su vez provoca la activación de otros marcos conectados con los primeros, dando lugar así a una red de activación, cuyo objetivo es predecir y explicar la información que se va a encontrar en esa situación. Este reconocimiento suele denominarse herencia o más generalmente reconocimiento descendente.

## **Estructura y componentes de un sistema experto**

Los sistemas expertos utilizados en inteligencia artificial son *software* que emula el comportamiento de un experto humano en la solución de un problema. Los sistemas expertos funcionan de manera que almacenan conocimientos concretos para un campo determinado y solucionan los problemas, utilizando esos conocimientos, mediante deducción lógica de conclusiones. Con ellos se busca una mejora en calidad y rapidez de respuestas dando así lugar a una mejora de la productividad del experto.

Los sistemas expertos pueden estar *basados en reglas*, es decir, disponen de unos conocimientos predefinidos que se utilizan para tomar todas las decisiones (aplicando *heurística*), o *basados en casos* (*CBR, Case Based Reasoning*), aplicando el razonamiento basado en casos, donde la solución a un problema similar planteado con anterioridad se adapta a un nuevo problema.

Estas reglas y hechos pueden implementarse, por ejemplo, mediante el cálculo de predicados, no obstante es más “adecuado” emplear reglas de producción para las reglas. El proceso de construcción de un SEBR (estas ideas se extienden a SBC’s con diferente estructura) es el siguiente:

Se extrae el conocimiento experto (bibliografía, entrevistas a expertos reales, etc.).

Se modela y se adquiere el conocimiento, utilizando un lenguaje adecuado,

Se crea la Base de Conocimiento con el conocimiento adquirido.

Además, normalmente, la percepción del mundo por parte de un agente no es perfecta.

Del mismo modo, es posible que una regla no sea aplicable siempre (aunque sí en un gran número de casos). Este hecho no permite que la regla sea admitida en un sistema de cálculo proposicional o de predicados, dado que daría lugar a sistemas inconsistentes.

Se hace necesario establecer mecanismos para tratar con incertidumbre.

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

En computación, árboles (Estructura de datos) y teoría de juegos, el Factor de ramificación es el número de nodos hijos en cada nodo. Si este valor no es uniforme, se puede calcular el factor de ramificación medio.

Por ejemplo, en ajedrez, si se considera un "nodo" como una posición legal, el factor de ramificación medio es aproximadamente 35. Esto significa que, de media, un jugador puede realizar alrededor de 35 movimientos legales en cada turno.

Factores de ramificación altos hacen que los algoritmos que evalúan todas las ramas de todos los nodos, como los de búsqueda por fuerza bruta, sean más costosos computacionalmente hablando por el crecimiento exponencial del número de nodos. El factor de ramificación puede ser reducido mediante algoritmos de poda.

[Explicar minimax y poda alfa-beta]

## **¿Se puede resolver de forma exacta un juego bipersonal con información perfecta? ¿De qué forma?**

## **¿Por qué podemos decir que la poda alfa-beta es un algoritmo mejor que el procedimiento minimax para la resolución de juegos?**

La poda alfa beta es una técnica de búsqueda que reduce el número de nodos evaluados en un árbol de juego por el algoritmo Minimax. El problema de la búsqueda Minimax es que el número de estados a explorar es exponencial al número de movimientos. Partiendo de este hecho, la técnica de poda alfa-beta trata de eliminar partes grandes del árbol, aplicándolo a un árbol Minimax estándar, de forma que se devuelva el mismo movimiento que devolvería este, gracias a que la poda de dichas ramas no influye en la decisión final.

La eficacia de la poda alfabeta depende del orden en el que se examinan los sucesores, es decir, el algoritmo se comportará de forma más eficiente si examinamos primero los sucesores que probablemente serán los mejores.

[Explicar minimax]

La regla minimax es la base para la propuesta de algoritmos que puedan obtener la mejor jugada en juegos bipersonales con información completa y suma nula. ¿Cómo se podría modificar dicha regla para poder proponer algoritmos que puedan obtener la mejor jugada en juegos bipersonales y de suma nula, pero en los que aparezca incertidumbre?