

Segundo parcial IA

1. Componentes de un juego

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

Un juego se compone de las siguientes características:

- Número de jugadores.
- Juegos de información perfecta vs. Juegos de información imperfecta: Ajedrez vs. Póquer.
- Existencia de movimientos al azar.
- Orden de actuación de los jugadores.
- Existencia o no de pagos colaterales (equilibrio de Nash).
- Juegos de suma nula vs. Juegos de suma no nula.

En general, 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 qué resultado puede esperar y cómo alcanzarlo.

Por tanto, un juego puede plantearse 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 factor de ramificación es el número medio de sucesores que puede tener un nodo de un grafo y es una de las tres componentes que expresan la complejidad de un grafo, y eleva la complejidad en espacio de un juego. 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.

-Los algoritmo minimax y poda alfa-beta realizan una búsqueda en grafos de juegos bipersonales, de información perfecta y de suma nula para elaborar una estrategia óptima considerando que se juega contra un oponente infalible. Los pasos a seguir:

1. Se genera todo el árbol. Se generarán todos los nodos hasta llegar a un estado terminal.
2. Una vez llegado a los nodos hoja, se calculan los valores minimax.
3. Los nodos padres reciben el valor minimax de sus nodos hijos. Según si es un nodo MAX o MIN se elegirán los valores mínimos y máximos representando los movimientos del jugador y del oponente.
4. Elegir la jugada valorando los valores que han llegado al nivel superior.

-El algoritmo poda alfa-beta se basa en el uso recursivo del valor minimax como el algoritmo anterior pero se apoya en dos parámetros: alfa y beta para reducir la complejidad espacial del grafo podando las ramas que son irrelevantes para la solución.

- α es el valor de la mejor opción hasta el momento a lo largo del camino para MAX, esto implica por lo tanto la elección del valor más alto
- β es el valor de la mejor opción hasta el momento a lo largo del camino para MIN, esto implica por lo tanto la elección del valor más bajo.

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

1.- Problemas semánticos

- 1.1. Es difícil de expresar todo en fórmulas heurísticas (ordenando los predicados o reglas frecuentes), metaconocimiento, jerarquía (subconjunto) y herencia (cumple las propiedades básicas), igualdad (ligado con el razonamiento acerca del predicado) y sentido común (relacionado con las heurísticas, metaconocimiento).
- 1.2. Razonamiento temporal: Es un problema de las variables de tiempo, se maneja de forma poco satisfactoria.
- 1.3. Razonamiento acerca de predicados
- 1.4. Información Incompleta (Faltan datos o ES VERDAD o ES FALSA) y/o imprecisa (vaguedad o imprecisión, probabilidad o desconocimiento o priori, Aplicando el principio de Laplace: Da la misma probabilidad a todo). Por ejemplo hay diferentes tonos de colores por lo tanto ni es cierto ni falso.
- 1.5. Excepción: Relacionado con la herencia y la jerarquía.
- 1.6. Monotonía: impide la revisión de las demostraciones al actualizar las Bases de Conocimiento.

2.- Problemas computacionales

- 2.1. Consistencia: Solidez (que tenga la garantía de la verdad absoluta). Lo que se demuestra como verdadera lo es realmente. Se suele demostrar por refutación por resolución.
- 2.2. Completitud: (Solo se da en la refutación por resolución), si una cosa es verdadera puede demostrarse. Semidecidible.
- 2.3. Complejidad computacional: (tratabilidad) .No tienen un algoritmo en tiempo polinomial.

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

Los modelos de conocimiento heredable son estructuras jerárquicas que permiten representar el conocimiento de manera incremental de manera que facilitan la representación del conocimiento humano. Se obtiene relacionando clases con otras clases de carácter más general, lo que permite reutilizar el conocimiento general en las nuevas clases y heredar de manera total o parcial el conjunto de atributos que tienen definidos. Dentro de los modelos de conocimiento heredable se encuentran las redes asociativas y los marcos (frames).

REDES ASOCIATIVAS:

Cada nodo representa un concepto (o una proposición) y los enlaces corresponden a relaciones (inclusión, pertenencia, causalidad) o a categorías gramaticales (verbo principal, sujeto, objeto, complementos, etc). Entre ellas se conocen:

- Redes semánticas: son las destinadas a representar o a comprender el lenguaje natural.
- Redes de clasificación: es exactamente lo que su nombre indica, una clasificación de objetos o conceptos con sus características propias (herencia y demás).
- Redes causales: son las que llevan asociadas, junto a sus nodos que representan variables, una relación de influencia, representada por los enlaces.

Las redes semánticas son una forma de lógica para representar el lenguaje natural y la notación que proporcionan para cierta clase de sentencias es a menudo más conveniente, pero si se deja de lado la <<interfaz humana>>, los conceptos base (objetos, relaciones, cuantificación, etc) son los mismos.

Existen diversas variantes pero todas son capaces de representar objetos individuales, categorías de objetos y relaciones entre objetos. Se representa como un grafo dirigido etiquetado constituido por:

- Nodos: representan conceptos (un objeto individual o una clase de objetos).
- Arcos: representan relaciones binarias entre los conceptos.

MARCOS:

Un marco es una estructura de datos para representar una situación estereotipada, como encontrarse en un cierto tipo de sala de estar o asistir a un cumpleaños infantil.

Cada marco se caracteriza por un conjunto de campos o slots que se asocian en general a atributos, 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.

La información (propiedades) específica al concepto representado por un marco es representado mediante atributos o slots. Los atributos ofrecen un medio de representar las propiedades de objetos individuales o clases de objetos.

Una faceta es considerada como una propiedad asociada a un atributo:

- faceta valor: es la más común y referencia el valor real del atributo.
- faceta valor por defecto: denota el valor inicial del atributo en caso de que no se especifique lo contrario.
- faceta tipo valor: especifica el tipo de datos del valor del atributo.
- faceta cardinalidad: especifica si se trata de un atributo uni o multi-valuado.
- faceta máxima cardinalidad: solo es válida para atributos multivaluados y especifica el máximo número de valores asociados al atributo.
- facetas demonio: permiten la integración de conocimiento declarativo y procedural. Un demonio o valor activo es un procedimiento que es invocado en un momento determinado durante la manipulación del atributo donde ha sido especificado.
- faceta tipo atributo: especifica si se trata de un atributo heredable o no heredable.
- faceta herencia: especifica el tipo de herencia del atributo.

5. Estructuras y componentes de un sistema experto

Un sistema Basado en Conocimiento (SBC) es aquel programa, en general, sistema hardware-software que emplea masivamente conocimiento para resolver un problema dentro de un dominio determinado.

Los SBC se aplican a una gran diversidad de campos y/o áreas. A continuación se listan algunas de las principales: Milita, Informática, Geología, etc.

Dentro de esta categoría destacan los Sistemas Expertos (SE) que pueden definirse como aquellos SBC capaces de comportarse como un experto (humano) en un determinado dominio de actividad:

- Resuelven un problema.
- Pueden ser consultados y justifican su Razonamiento.

Las componentes de estos sistemas son:

- Una Base de Conocimiento (BC), que contenga el conocimiento necesario sobre el dominio/Universo del problema a resolver.

La BC contendrá una descripción del mundo: sus objetos y las relaciones entre ellos.

- Un Motor de Inferencia, que permite razonar sobre el conocimiento de la BC y los datos proporcionados por un usuario.
- Una interfaz de usuario para entrada/salida de datos de cada problema. A veces los datos de un problema se conocen como Base de Hechos.
- Módulo de explicaciones/justificación.

6.Paradigmas de Aprendizaje Automático

-Aprendizaje memorístico

El aprendizaje memorístico o repetitivo se basa en retener mentalmente datos sin procesarlos detenidamente. Los contenidos memorizados no son comprendidos y tampoco se intenta analizar su significado. Se repiten las suficientes veces hasta que se recuerdan.

-Aprendizaje deductivo

Es una forma razonar y explicar la realidad partiendo de leyes o teorías generales a casos particulares.

-Aprendizaje analítico

Basado 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

Es un tipo de razonamiento de carácter inductivo en el cual se relacionan dos situaciones distintas pero que responden a la misma lógica amparados en un proceso determinado. Aplicando el razonamiento analógico, es posible llegar a una conclusión, basándose en premisas establecidas con anterioridad y estableciendo una comparación entre ambas, para así obtener un resultado producto de una analogía.

-Aprendizaje inductivo

Es el método de aprendizaje en el que el aprendiz busca de manera autónoma el conocimiento, en un proceso que parte de la observación y el análisis de las características del concepto, habilidad o competencia a aprender mediante el uso de ejemplos y contraejemplos.

-Tipos de aprendizaje según el conocimiento utilizado

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

1. Aprendizaje Supervisado: Para cada entrada, se dispone de un profesor/supervisor que proporciona una salida deseada, ya sea una clase o un valor a aproximar(clasificación vs regresión).
2. Aprendizaje no supervisado: No se dispone de una salida deseada cada entrada, sino que se busca agrupar/clasificar los datos en función de ciertas características(medida de distancia)
3. 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.

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

A la hora de encontrar una hipótesis que sea consistente con un conjunto de ejemplos, es decir, que sea capaz de predecir ejemplos que no se conoce, podemos utilizar

APRENDIZAJE-ÁRBOL-DECISIÓN. En este modelo de aprendizaje construimos un árbol de decisión que toma como entrada una situación descrita a través de un conjunto de atributos y devuelve una

“decisión”(deseada), en función de esos atributos. Este árbol se obtiene gracias al análisis un conjunto de entrenamiento que ajustan el árbol. Cada ruta de este árbol es una “combinación” del posible estado de los atributos. La idea fundamental es elegir buenos atributos que sean capaz de minimizar este árbol, ya que este árbol se va ramificando con cada testeo de un atributo en un nivel del árbol. Un buen atributo debería dividir el conjunto de ejemplos en subconjuntos que sean o “todos positivos” o “todos negativos”, será mejor si divide los ejemplos en conjuntos que contienen solo ejemplos positivos o negativos . (La idea básica del algoritmo APRENDIZAJE-ÁRBOL-DECISIÓN es realizar primero el test sobre el atributo más importante. Se considera como «atributo más importante» aquel que discrimina más claramente los ejemplos.

De esta forma, esperamos obtener la clasificación correcta con un número de test pequeño, es decir, que todos los caminos en el árbol sean cortos y así el árbol completo será pequeño.)

Se presenta por lo tanto dos problemas el ruido y sobreajuste:

Ruido: Si hay dos o más ejemplos con la misma descripción (en términos de los atributos), pero diferentes clasificaciones, el algoritmo APRENDIZAJE-ÁRBOL-DECISIÓN fallará en la búsqueda de un árbol de decisión que sea consistente con todos los ejemplos. Esto ocurre a la hora de elegir atributos para test en la toma de decisiones Si no quedan atributos, pero sí ejemplos positivos y negativos, tenemos un problema, ya que los ejemplos que quedan tienen exactamente la misma descripción, pero clasificaciones diferentes. Esto sucede cuando alguno de los datos es incorrecto; diremos que hay ruido en los datos. También ocurre cuando los atributos no proporcionan suficiente información para describir completamente la situación, o cuando el dominio no es determinista. Una forma sencilla de resolver el problema es utilizar el voto de la mayoría. Si no quedan atributos, pero sí ejemplos positivos y negativos, tenemos un problema, ya que los ejemplos que quedan tienen exactamente la misma descripción, pero clasificaciones diferentes. Esto sucede cuando alguno de los datos es incorrecto; diremos que hay ruido en los datos. También ocurre cuando los atributos no proporcionan suficiente información para describir completamente la situación, o cuando el dominio no es determinista. Una forma sencilla de resolver el problema es utilizar el voto de la mayoría. La solución consiste en tener en cada nodo hoja la clasificación de la mayoría de su conjunto de ejemplos, si se requiere una hipótesis determinista.

Sobreajuste: Encontrar “regularidades” poco significativas en los datos. Se dice que una hipótesis h se sobreajusta al conjunto de entrenamiento si existe alguna otra hipótesis h' tal que el error de h es menor que el de h' sobre el conjunto de entrenamiento, pero es mayor sobre la distribución completa de ejemplos del problema (entrenamiento + test). El sobreajuste es un fenómeno muy generalizado que ocurre cuando la función principal no es del todo aleatoria, es decir que funcionara bien para un conjunto de entrenamiento y estará preparada para predecir decisiones dentro de ese conjunto pero una vez que le proponemos un caso distinto es probable que falle debido a ese sobre ajuste que solo refleja ese conjunto.

Para afrontar este problema se usa una la técnica poda del árbol de decisión. La poda funciona impidiendo divisiones recursivas sobre atributos que no son claramente relevantes, incluso cuando los datos en ese nodo del árbol no estén clasificados de forma uniforme.

¿cómo se detecta un atributo irrelevante? Esta pregunta se puede responder utilizando un test de significancia estadístico.

Definición más formal: Dado un espacio de hipótesis H , se dice que la hipótesis $h \in H$ sobrepasa los datos de entrenamiento si existe alguna hipótesis alternativa $h' \in H$, tal que h tiene un error menor que h' sobre los ejemplos de entrenamiento, pero h' tiene un error menor que h en toda la distribución de instancias.

8. ¿Qué son y cómo se construyen los árboles de decisión?

Un árbol de decisión es un modelo de predicción muy similar a los sistemas de predicción basados en reglas, que sirve para representar y categorizar una serie de condiciones que ocurren de forma sucesiva para la resolución de un problema. Toma como entrada un objeto o una situación descrita a través de un conjunto de atributos y devuelve una “decisión”, el valor previsto de la salida dada la entrada. Los atributos de entrada pueden ser continuos o discretos, a su vez el valor de salida puede ser continuo o discreto.

Un árbol de decisiones desarrolla una secuencia de test para poder alcanzar una decisión. Cada nodo interno del árbol corresponde con un test sobre el valor de una de las propiedades, y las ramas que salen del nodo están etiquetadas con los posibles valores de dicha propiedad. Cada nodo hoja del árbol representa el valor que ha de ser devuelto si dicho nodo hoja es alcanzado.

Para construir los árboles de decisiones se deben coger ejemplos positivos donde un predicado meta es verdadero y ejemplos negativos en los que el mismo predicado meta es falso. El conjunto de ejemplos completo se denomina conjunto de entrenamiento.

El problema de encontrar un árbol de decisión que sea consistente con el conjunto de entrenamiento puede parecer difícil pero, de hecho, existe una solución trivial. El problema del árbol trivial es que memoriza exactamente las observaciones y no extrae ningún patrón a partir de los ejemplos, por lo tanto no se puede esperar que sea capaz de extrapolar a ejemplos que no le han sido proporcionados. La manera de óptima de inferir el árbol compatible con todas las instancias es la navaja de Ockham pero es inviable computacionalmente. Se recurre a una solución pseudo-óptima, seleccionando el atributo en cada nivel del árbol en función de la calidad de la división que produce. Para la elección de los atributos del test se debe elegir primero los atributos que proporcionen una clasificación lo más exacta posible de los ejemplos y así minimizar la profundidad del árbol final. Un atributo perfecto divide los ejemplos en conjuntos que contienen solo ejemplos positivos o negativos. Para saber si un atributo es bastante adecuado, la medida debe tener su valor máximo cuando el atributo es perfecto, y su valor mínimo cuando el atributo es inadecuado. Una medida adecuada es la ganancia de información proporcionada por el atributo, donde se utiliza el término de información en su sentido matemático.