



Aprendizaje

CI5438 - Inteligencia Artificial 2

Clase 6

Cap 18. Russel & Norvig: Inteligencia Artificial. Un enfoque estructurado



Grupo de Inteligencia Artificial



Aprendizaje: Motivación

- ◆ La lección del los Sistemas Expertos:
 - Adquisición conocimiento: El proceso de extraer la experticia de un humano para codificarla el un programa
 - Problemas:
 - ◆ Trabajo extremadamente intensivo
 - ◆ El desempeño de un sistema experto es incorrecto si la percepción no se corresponde con alguna regla codificada.
 - ◆ Nuevo conocimiento requiere re-codificación.
 - Solución: Diseñar sistemas que aprendan de su propia experiencia.



Grupo de Inteligencia Artificial

Razonamiento

Universidad Simón Bolívar



Deductivo e Inductivo

◆ Las conclusiones deductivas "están contenidas" en las premisas:

- $\forall x \text{ Hombre}(x) \rightarrow \text{Mortal}(x)$
- $\text{Mortal}(\text{Pedro})$
- Luego, $\text{Mortal}(\text{Pedro})$

◆ Las conclusiones Inductivas "van más allá" de las premisas:

- $\text{Hombre}(\text{Pedro}) \& \text{Mortal}(\text{Pedro})$
- $\text{Hombre}(\text{Luis}) \& \text{Mortal}(\text{Luis})$
- $\text{Hombre}(\text{Maria}) \& \text{Mortal}(\text{Maria})$
- Luego, $\forall x \text{ Hombre}(x) \rightarrow \text{Mortal}(x)$ [nuevo conocimiento ha sido aprendido]



Grupo de Inteligencia Artificial

Universidad Simón Bolívar



18.0 Qué es el aprendizaje?



Grupo de Inteligencia Artificial



Qué es el aprendizaje?

- ◆ “El aprendizaje denota cambios en un sistema que... capacitan al sistema a realizar la misma tarea más eficientemente la próxima vez”.
–Herbert Simon
- ◆ “Aprender es construir o modificar la representación de lo que ha sido experimentado”. –Ryszard Michalski
- ◆ “Aprender es hacer cambios útiles en nuestras mentes”. –Marvin Minsky



Por qué aprender?

- ◆ Entender y mejorar la eficiencia del aprendizaje humano
 - Para mejorar los métodos para enseñar a la gente (e.j. mejores sistemas de instrucción asistida por computadora)
- ◆ Descubrir nuevas cosas o estructuras que eran desconocidas para los humanos
 - Ejemplos: data mining



Por qué aprender?

- ◆ Rellenar el esqueleto o la especificación incompleta de un dominio
 - En general, los sistemas de IA complejos con pueden ser derivados completamente a mano, y requieren de una actualización dinámica para incorporar nueva información
 - Aprender nuevas características expande el dominio o la experticia y disminuye la “fragilidad” del sistema
- ◆ Construir agentes de software que puedan adaptarse a sus usuarios o a otros agentes de software



18.1 – Tipos de Aprendizaje





Aprendizaje por observación

- ◆ El aprendizaje en un agente involucra percibir, razonar y actuar en el ambiente que lo rodea
- ◆ El agente interactúa con el mundo, y observa sus propios procesos de toma de decisiones



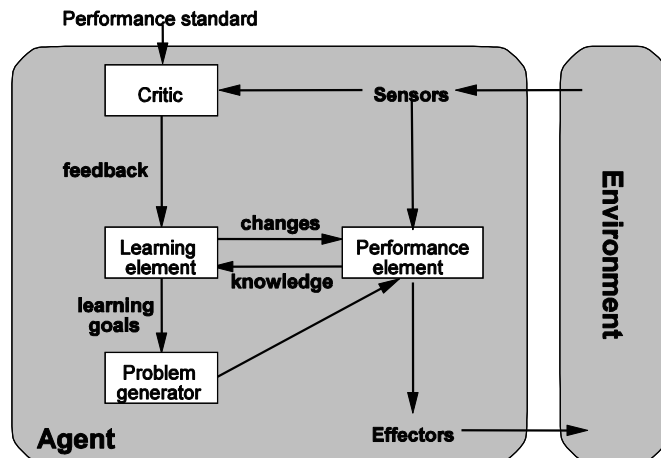
Modelo general de los agentes con aprendizaje



- ◆ All agents described so far have only used their percepts to come up with an action
- ◆ True intelligence requires using percepts to also improve future actions (learning)
- ◆ The agents described so far have been *performers*, not *learners*
- ◆ A general model of learning contains a *performance element* as just one of several elements (fig. 18.1)



Modelo general de los agentes con aprendizaje



Modelo general de los agentes con aprendizaje



- ◆ Los agentes que aprenden están conformados por 4 componentes:
 - El elemento de aprendizaje (*Learning Element*)
 - El elemento de desempeño (*Performance Element*)
 - El crítico (*Critic*)
 - El generador de problemas (*Problem Generator*)



Modelo general de los agentes con aprendizaje

Universidad Simón Bolívar



- ◆ **Elemento de Desempeño** (E.D.): Los agentes que hemos estudiado hasta el momento
- ◆ **Elemento de Aprendizaje** (E.A.): Toma el *conocimiento* del E.D y el *feedback* sobre las acciones del agente para modificar el E.D en el futuro
- ◆ **Crítico**: provee el feedback para el E.A. mediante el monitoreo del resultado de las acciones tomadas por el E.D. (debe estar fuera del E.D.)
- ◆ **Generador de problemas**: Provee las acciones que conllevarán a experiencias de aprendizaje (como experimentos científicos)



Grupo de Inteligencia Artificial

Diseño de un Agente que aprende

Universidad Simón Bolívar



- Cuatro cuestiones principales afectan el diseño del Elemento de Aprendizaje:
- ◆ Cuáles *componentes* del Elemento de Performance van a ser mejorados?
 - ◆ Qué *representación* se usará para esos componentes?
 - ◆ Qué *feedback* está disponible?
 - ◆ Qué *información previa* hay disponible?



Grupo de Inteligencia Artificial

Componentes del Elemento de Desempeño

Universidad Simón Bolívar



- ◆ **No**: "Cómo voy a poder aprender esto?"
- ◆ "Qué clase de Elemento de Desempeño necesitará mi agente para hacer esto, una vez que halla aprendido como"



Grupo de Inteligencia Artificial

6 Componentes del Elemento de Desempeño

Universidad Simón Bolívar



1. Un mapeo directo de las condiciones del estado actual en acciones
2. Una medida para inferir las propiedades relevantes del mundo a partir de la secuencia de percepciones
3. Información sobre como el mundo evoluciona y sobre el resultado de las acciones posibles que el agente puede tomar



Grupo de Inteligencia Artificial

6 Componentes del Elemento de Desempeño

Universidad Simón Bolívar



4. Información sobre la Utilidad, que indique lo deseable que son los estados del mundo
5. Información sobre el Valor de una Acción (Action-value), que indique lo deseable de una acción particular en un estado particular
6. Metas, que describan las clases de estados cuyo alcance maximice la utilidad del agente



Grupo de Inteligencia Artificial

Feedback and Prior Knowledge

Universidad Simón Bolívar



- ◆ If the Critic provides feedback in the form of the correct action, the learning is **supervised**
- ◆ If the correct action is not given, but the consequences of the chosen action are, this is reinforcement learning
- ◆ If no information at all about the action is given, then any learning is unsupervised
- ◆ Prior knowledge may or may not be available, but it is enormously helpful for learning (most M.L. research does not use prior knowledge)



Grupo de Inteligencia Artificial

Feedback Disponible

Universidad Simón Bolívar



- ◆ **Aprendizaje Supervisado** – el agente puede percibir tanto las entradas como las salidas de una componente. Generalmente las salidas son proporcionadas por un tutor amistoso. El problema del aprendizaje supervisado involucra aprender una función a partir de ejemplos de sus entradas y salidas



Grupo de Inteligencia Artificial

Feedback Disponible

Universidad Simón Bolívar



- ◆ **Aprendizaje No Supervisado** – el agente no tiene idea de las salidas correctas. Puede aprender a predecir sus percepciones futuras dadas las previas, pero no puede aprender que hacer a menos que ya posea una función de utilidad. El problema del aprendizaje no supervisado involucra aprender patrones en las entradas cuando los valores específicos de las salidas no son suministrados



Grupo de Inteligencia Artificial



Feedback Disponible

- ◆ **Aprendizaje por Reforzamiento** – cuando el agente recibe alguna evaluación de sus acciones (como una multa por chocar un carro), pero no se le indica cuál es la acción correcta (frenar antes). El agente en lugar de tener un tutor que le diga que hacer, el agente debe aprenderlo por un reforzamiento



Representación de los Componentes

- ◆ **Descripciones Determinísticas** como funciones polinomiales para programas de juegos
- ◆ **Logica Proposicional y de primer orden** para todos los componentes de un agente logico
- ◆ **Descripciones Probabilísticas** como redes de creencias, para el componente de inferencias de un agente de “decisiones teóricas”



Conocimiento Previo

- ◆ La mayor parte de la investigación en ML se centra en agentes que no tienen conocimiento previo de que es lo que trata de aprender
- ◆ Los agente sólo tienen acceso a los ejemplos que les presentan sus experiencias
- ◆ Si el agente tiene conocimiento previo, puede ayudar en el aprendizaje



18.2 – Aprendizaje Inductivo





Aprendizaje Supervisado

◆ Todo aprendizaje pretende aprender la representación de una función:

- *estado* → *estado*: Como evoluciona el mundo
- *estado* → (0,1): si el estado es una meta
- *acción* → *real*: función de utilidad, etc.

◆ *Aprendizaje Inductivo*

- *Dado: un conjunto de ejemplos $(x, f(x))$*
- *Tarea: encontrar una función h que aproxime f .*
- *Como varias funciones pueden satisfacer los ejemplos, cualquier preferencia por una es un **bias**. Todos los algoritmos de aprendizaje exhiben **bias***

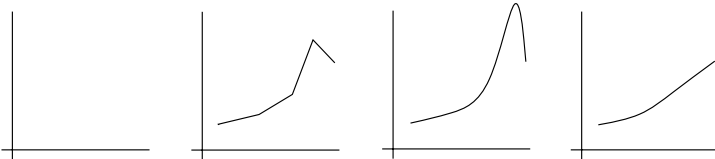


Aprendizaje Inductivo

- ◆ Un **ejemplo** es un par $(x, f(x))$ donde x es la entrada y $f(x)$ es la salida de la función aplicada a x
- ◆ **Inferencia inductiva pura (o inducción)**: dado un conjunto de ejemplos del f , retornar una función h que aproxime f . La función h se denomina **hipótesis**
- ◆ Una buena hipótesis **generalizará** bien. i.e., predecirá ejemplos desconocidos correctamente



Aprendizaje Inductivo : Ejemplo



En (a) mostramos un ejemplo de pares (entrada, salida). En (b), (c), (d) tenemos tres hipótesis de funciones desde las cuales podrían haberse obtenido estos ejemplos



Aprendizaje Inductivo

- ◆ Una hipótesis es **consistente** si concuerda con todos los datos
- ◆ Cómo escogemos entre varias hipótesis consistentes?
- ◆ **Ockham razor**: Preferir la hipótesis más simple que sea consistente con los datos
- ◆ La posibilidad de encontrar una hipótesis simple y consistente depende fuertemente del espacio de hipótesis escogido.





18.3 – Aprendizaje en árboles de Decisión



Árboles de Decisión

- ◆ Un árbol de decisión es un árbol de búsqueda que retorna una “decisión”
 - Entrada: Un **objeto** o **situación** descrito por un conjunto de **atributos**
 - Salida: Una Decisión.
- ◆ La entrada puede ser discreta o continua.
- ◆ La salida también puede ser discreta o continua
- ◆ Aprender una función con salida **discreta** se denomina aprendizaje de **clasificación**
- ◆ Aprender una función con salida **continua** se denomina **regresión**



Árboles de Decisión

- ◆ Cada **nodo** en el árbol corresponde con una prueba de uno de los atributos
- ◆ Cada **enlace** entre los nodos representa uno de los valores que puede tomar el nodo padre
- ◆ Cada hoja en el árbol es el valor booleano retornado si la búsqueda llegó a esa hoja
- ◆ La meta es aprender la definición del **predicado meta**, donde esta definición se expresa como un árbol de decisión



Árboles de Decisión

- ◆ El proceso de construir un árbol de decisión es como sigue:
 - Comenzar con el problema
 - Determinar cuál es el predicado meta para el problema
 - Construir una lista de atributos que describen el dominio del problema
 - Eliminar los atributos irrelevantes
 - Construir el árbol usando la lista filtrada de atributos



Ejemplo de Árbol de Decisión

- El problema es decidir si esperar o no por una mesa en un restaurant
- Un predicado meta para este ejemplo es Esperar. En otras palabras, si una persona esperará por una mesa para comer en el restaurant

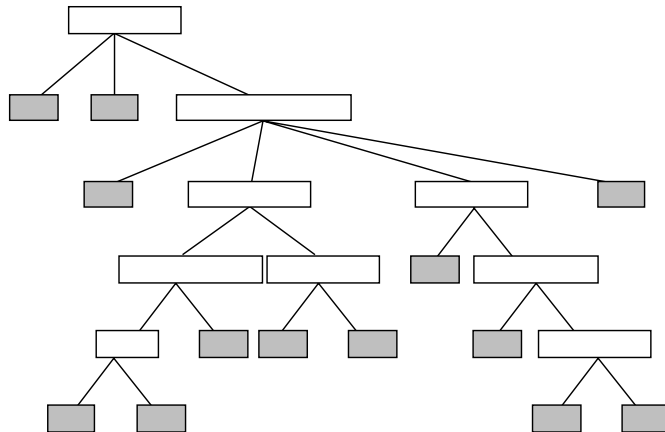


Ejemplo de Árbol de Decisión

Attribute	Description
Alternate	Whether there is a suitable alternative restaurant nearby.
Bar	Whether the restaurant has a comfortable bar area to wait in.
Fri/Sat	True on Fridays and Saturdays.
Hungry	Whether the person is hungry.
Patrons	How many people are in the restaurant? (Values are None, Some, and Full)
Price	The restaurant's price range. (Values are \$, \$\$, \$\$\$)
Raining	Whether it is raining outside.
Reservation	Whether the person made a reservation.
Type	The kind of restaurant.
WaitEstimate	The wait estimated by the host. (Values are 0-10 minutes, 10-30, 30-60 and >60)



Ejemplo de Árbol de Decisión



Expresividad de los árboles de decisión

- Cualquier árbol de decisión particular para un predicado meta PM es una expresión de la forma:
 - $\forall s \text{ PM}(s) \Leftrightarrow (P_1(s) \vee P_2(s) \vee \dots \vee P_n(s))$
 - Donde $P_i(s)$ es la conjunción de pruebas de un camino entre la raíz y una hoja con salida positiva
 - Proposicional: solo 1 variable y predicados unarios



Expresividad de los árboles de decisión

- ◆ Pregunta: Si los árboles de decisión se corresponden con un conjunto de conjunto de implicaciones, pueden representar cualquier conjunto?
- ◆ Respuesta: No!
 - Los árboles de decisión están implícitamente limitados a hablar de solo un objeto o situación
 - El lenguaje de los árboles de decisión es esencialmente proposicional, donde cada uno de los atributos que se prueban es un objeto simple
- ◆ Los árboles de decisión son completamente expresivos en el lenguaje de la lógica proposicional
- ◆ En otras palabras, cualquier función booleana puede ser representada con un árbol de decisión

Induciendo árboles de decisión a partir de ejemplos

- ◆ Meta: Encontrar el árbol más pequeño que sea consistente con los ejemplos de entrenamiento
- ◆ Un ejemplo consiste en un vector de atributos de entrada X , y un valor booleano de salida y

Induciendo árboles de decisión a partir de ejemplos

- ◆ Ejemplo – Un ejemplo consiste en un vector de atributos de entrada X , y un valor booleano de salida y
- ◆ Clasificación – Valor del predicado meta
- ◆ Ejemplo Positivo – El predicado meta es True
- ◆ Ejemplo Negativo – El predicado meta es False
- ◆ Conjunto de Entrenamiento – Un conjunto completo de ejemplos positivos y negativos

Conjunto de Entrenamiento

- ◆ Ejemplos descritos como valores de atributos (booleanos, discretos, continuos, etc.)

Example	Attributes										Goal
	Alt	Bar	Fri	Hun	Pat	Price	Rain	Res	Type	Est	WillWait
X_1	Yes	No	No	Yes	Some	\$\$\$	No	Yes	French	0-10	Yes
X_2	Yes	No	No	Yes	Full	\$	No	No	Thai	30-60	No
X_3	No	Yes	No	No	Some	\$	No	No	Burger	0-10	Yes
X_4	Yes	No	Yes	Yes	Full	\$	No	No	Thai	10-30	Yes
X_5	Yes	No	Yes	No	Full	\$\$\$	No	Yes	French	>60	No
X_6	No	Yes	No	Yes	Some	\$\$	Yes	Yes	Italian	0-10	Yes
X_7	No	Yes	No	No	None	\$	Yes	No	Burger	0-10	No
X_8	No	No	No	Yes	Some	\$\$	Yes	Yes	Thai	0-10	Yes
X_9	No	Yes	Yes	No	Full	\$	Yes	No	Burger	>60	No
X_{10}	Yes	Yes	Yes	Yes	Full	\$\$\$	No	Yes	Italian	10-30	No
X_{11}	No	No	No	No	None	\$	No	No	Thai	0-10	No
X_{12}	Yes	Yes	Yes	Yes	Full	\$	No	No	Burger	30-60	Yes

Ockham's razor

- ◆ “The most likely hypothesis is the simplest one that is consistent with all observations”
- ◆ The razor states that the most likely hypothesis is the simplest one that is consistent with all observations.



Algoritmo de Aprendizaje de un árbol de decisión

- ◆ El algoritmo DECISION-TREE-LEARNING cumple con el objetivo de construir un árbol pequeño y consistente con el conjunto de entrenamiento
- ◆ La idea central del algoritmo es probar primero el atributo más importante
- ◆ El término “más importante” significa que sea el atributo que haga una mayor diferencia en la clasificación de los ejemplos
- ◆ El resultado de esta heurística es que el árbol de decisión es más pequeño



Algoritmo de Aprendizaje de un árbol de decisión



function DECISION-TREE-LEARNING(*examples, attributes, default*) returns a decision tree

inputs: *examples*, set of examples
attributes, set of attributes
default, default value for the goal predicate

```

if examples is empty then return default
else if all examples have the same classification then return the classification
else if attributes is empty then return MAJORITY-VALUE(examples)
else

```

best ■ CHOOSE-ATTRIBUTE(*attributes, examples*)

tree ■ a new decision tree with root test *best*
for each value v_i of *best* do

examples_i ■ elements of *examples* with *best* = *v_i*

```
subtree ■ DECISION-TREE-LEARNING(examples, attributes – best,  
MAJORITY-VALUE(examples))
```

add a branch to *tree* with label v_i and subtree *subtree*

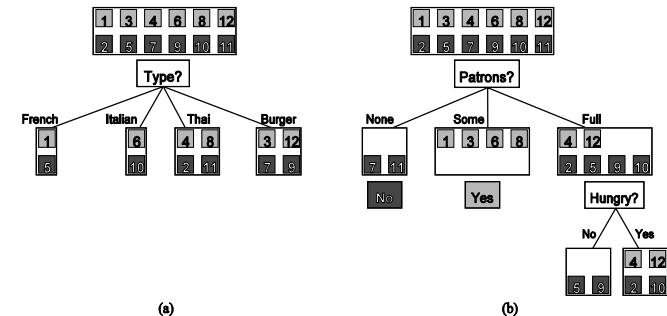
end

return tree

Escogiendo un atributo



- ◆ Idea: Un buen atributo separa los ejemplos en subconjuntos que son (idealmente) "todos positivos" o "todos negativos"





Información

- ◆ La información responde a la pregunta
- ◆ Mientras menos idea tengo sobre la respuesta inicialmente, hay más información contenida en la respuesta
- ◆ Escala: 1 bit = respuesta a una pregunta booleana con previo $<0.5, 0.5>$
- ◆ La información en una respuesta cuando el previo es $<P_1, \dots, P_n>$ es

$$I(<P_1, \dots, P_n>) = \sum -P_i \log(P_i)$$



Información (cont.)

- ◆ Supongamos que tenemos p ejemplos positivos y n negativos en la raíz
 - Se necesitan $I(<p/(p+n), n/(p+n)>)$ bits para clasificar un nuevo ejemplo
 - En el ejemplo del restaurant $p=n=6$, luego necesitamos 1 bit
- ◆ Un atributo separa los ejemplos E en subconjuntos E_i , cada uno de los cuales (esperamos) necesite menos información para completar la clasificación



Información (cont.)

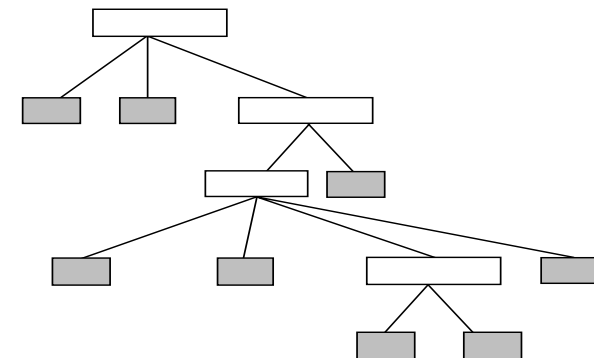
- ◆ Sea E_i con p_i ejemplos positivos y n_i negativos
 - Se necesitan $I(<p_i/(p_i + n_i), n_i/(p_i + n_i)>)$ bits para clasificar un nuevo ejemplo
 - El número esperado de bits sobre todas las ramificaciones es

$$\sum (p_i + n_i)/(p + n) (I(<p_i/(p_i + n_i), n_i/(p_i + n_i)>))$$
- ◆ Para *patrons*? Es 0.459 bits y para *type* es 1
- ◆ Escoger el atributo que minimize el la información que será requerida



Ejemplo (cont)

- ◆ Árbol de Decisión inducido por los ejemplos



Desempeño del Algoritmo de Aprendizaje



- ◆ Un algoritmo de aprendizaje es bueno si produce hipótesis que predican adecuadamente la clasificación de ejemplos desconocidos.
- ◆ Una metodología para verificar la calidad de la predicción:
 1. Coleccionar un gran conjunto de ejemplos
 2. Separarlo en dos conjuntos disjuntos: el **conjunto de entrenamiento** y el **conjunto de prueba**
 3. Usar en algoritmo en es conjunto de entrenamiento para generar la hipótesis h
 4. Medir el porcentaje de los ejemplos del conjunto de pruebas que son clasificados correctamente por h
 5. Repetir los pasos 1 al 4 variando el tamaño del conjunto de entrenamiento y seleccionando aleatoriamente los elementos para cada tamaño



Medida de Desempeño



- ◆ Cómo sabemos si $h \approx f$?
 - Usar los teoremas de la teoría de aprendizaje estadístico
 - Probar h es un nuevo conjunto de ejemplos de prueba
- ◆ Curva de aprendizaje = % de aciertos en el conjunto de prueba como función del tamaño del conjunto de entrenamiento

