Introducción al RA - Modelos Probabilistas

Características del conocimiento aproximado:

- No es del todo confiable, incierto, impreciso, incompleto, contradictorio, no monótono
- La lógica clásica no alcanza para representarlo (solo sirve para razonamientos basados en información certera T o F)
- El conocimiento cuenta con **predicados y cuantificadores vagos** (no precisos)

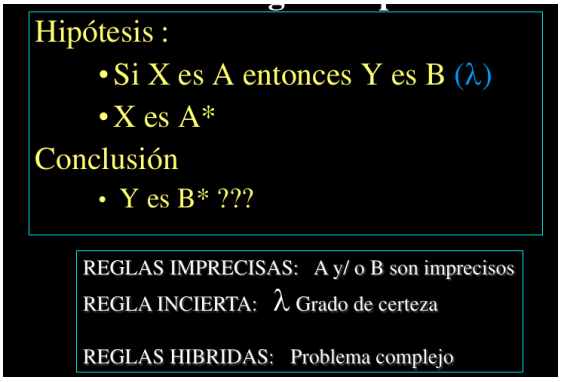
Conocimiento incierto: el conocimiento se expresa mediante predicados precisos pero no podemos establecer el valor de verdad de la expresión. En consecuencia se evalua la probabilidad, posibilidad, necesidad/plausabilidad o grado de certeza.
el ejemplo de conocimiento incierto mas gral: cuan cierto es que pase ... dado que ...
Conocimiento incompleto: se debe tomar decisiones a partir de información incompleta o parcial. Se suele manejar a través de supuestos o valores por defecto.
Conocimiento no monótono: la información es recibida a partir de distintas fuentes o en diferentes momentos es conflictiva y cambiante.

Proposiciones precisas (X toma valores en S): {p : “X es s” / s \in S}
Proposiciones imprecisas: {p : “X es r” / r \contains S}

Imprecisa - no borrosa => si r es un conjunto clásico
Imprecisa - borrosa (fuzzy) => si r es un conjunto borroso (fuzzy)

Razonamiento aproximado: trata como representar, combinar y realizar inferencias con conocimiento impreciso y/o incierto.

Esquema general en sistemas basados en reglas de producción:



Modelos de razonamiento aproximado: **modelos probabilísticos (RB)**, modelos evidenciales, modelos posibilísticos (fuzzy logic).

Regla de Bayes: P(B/A) = P(A/B) * P(B) / P(A) ----> A -> B … P(B/A) == inferencia
Distribución conjunta: P(x1, …, xn) = \prod P(xi / Padres(xi)) for i=1,n

Redes Bayesianas:

- Para representar la dependencia que existe entre determinadas variables, en aplicaciones complejas.
- sirve para especificar de manera concisa la distribución de probabilidad conjunta.

- Independencia: se hace explícita mediante la separación de grafos.
- Se construye incrementalmente por el experto agregando objetos y relaciones.
- Los arcos NO son estáticos (cuantifica la certeza de los nodos que une, B => A)

Además de especificar la topología de la red: base de conocimientos abstractos, válida en una gran cantidad de escenarios diversos. Representa la estructura general de los procesos causales del dominio.

Belief revision: consiste en encontrar la asignación global que maximice cierta probabilidad
Si quiero maximizar x, quiero encontrar y,z,w,v… / P(x,y,z,wv) es máxima

Belief update: es la actualización de probabilidades de un nodo dadas un conjunto de evidencias
Actualizo R dado: P(R/E)

Sistemas Borrosos

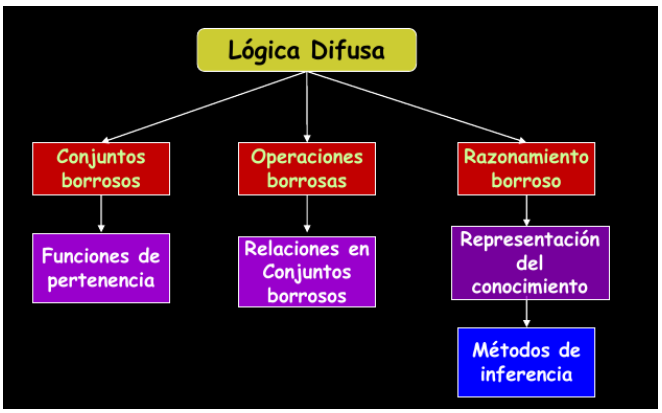
En este caso, nos estaremos manejando mediante incertidumbre/pertenencia: si tengo 0,1 de pertenencia de que haya veneno en 10 vasos de agua, entonces HAY 0,1 de veneno. En cambio, en los modelos probabilísticos con las probabilidades está la posibilidad de que esté o no. Puede no estar.

Una medida borrosa g en X es una función de conjunto g : 2^x -> [0,1] que satisface los siguientes axiomas:

1. g(\emptyset) = 0
2. g(X) = 1
3. A \contain B \contain => g(A) <= g(B)

La aditividad de las probabilidades es cambiada por una propiedad más débil (3), llamada monotonía respecto a la inclusión, lo que permite una mayor potencialidad descriptiva en las relaciones entre elementos de toma de decisión.

Al contrario de los conjuntos clásicos donde los límites están bien definidos, un objeto no pertenece simplemente sí o no a un conjunto: cada valor pertenece “en grados” a un conjunto borroso.



Conjunto borroso A = {x, f_A(x)} donde f_A representa el grado de pertenencia de x en A (puede ser gaussiana, sigmoide, lineal por partes, etc). Las funciones de pertenencia son asignadas por expertxs y deben ser lo más sencillas posible.
ALTX = (0/1.65, 0.8/1.75, 0.9/1.85, 1/1.95)

Propiedades en conjuntos borrosos:

- Normalidad: A es normal si su supremo es uno: sup f_A(x) = 1 \forall x \in X
- \alpha-cut: los elementos que pertenecen a A al menos en grado \alpha es llamado el conjunto \alpha-level: A\alpha = {x \in X / f_A(x) >= a}}
- Igualdad: A = B ⇔ f_A(x) = f_B(x) \forall x \in X
- Inclusión: A \containq B ⇔ f_A(x) <= f_B(x) \forall x \in X

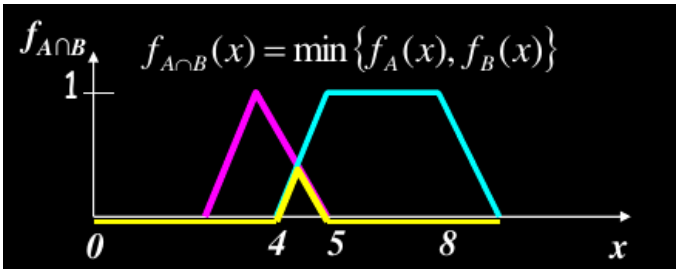
Operaciones en conjuntos borrosos:

Complemento
Dado un conjunto borroso A = {x, f_A(x)}, el N(A) se interpreta como el grado en que x NO pertenece a A. C = N: [0,1] -> [0,1] / f_C(A) (x) = 1 - f_A (x)
Props: Frontera N(0)=1 y N(1)=0, Monotonía N(a) ≥ N(b) if a ≤ b, Involución N[N(a)] = a

Intersecciones (T-normas)
f_ A \cap B (x) = T(f_A(x), f_B(x))
Props: Conmutativa, asociativa, Límite: T(a, 0) = 0, T(a, 1) = a,
Monotonía: T(a, b) ≤ T(a, c) if b ≤ c

Tipos de T-normas:

- Intersección estándar: Min(a,b) = min(a,b)
- Producto algebraico: Prod(a,b) = a * b
- Diferencia acotada (Lukasiewicz): W(a,b) = max(0, a+b-1)
- Intersección drástica: Z(a,b) = a si b=1, =b si a=1 y =0 en otro caso.

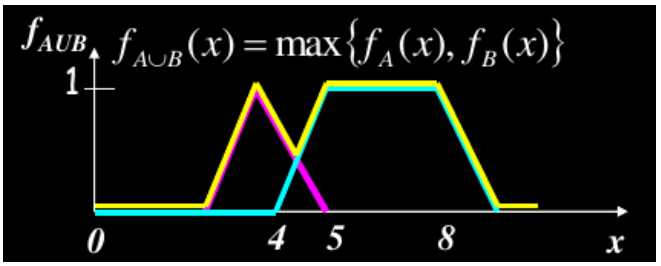


Siempre se cumple que Z <= W <= Prod <= Min

Uniones (T-conormas) borrosas
f_ A \cup B (x) = T(f_A(x), f_B(x))
Props: Conmutativa, asociativa, Límite: T(a, 0) = a, T(a, 1) = 1,
Monotonía: T(a, b) ≤ T(a, c) if b ≤ c

Tipos de T-conormas:

- Unión estándar: Max(a,b) = max(a,b)
- Suma algebraica: Prod*(a,b) = a + b - a*b
- Suma acotada (dual de Lukasiewicz): W*(a,b) = min(1, a+b)
- Unión drástica: Z*(a,b) = a si b=0, =b si a =0, =1 en otro caso

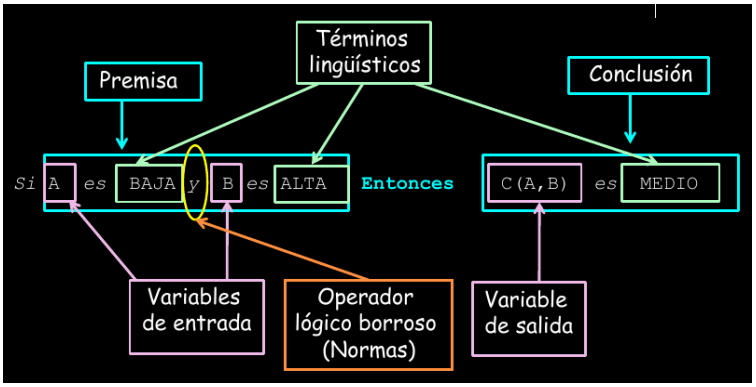


Siempre se cumple que Max <= Prod* <= W* <= Z*

Propiedades de operaciones borrosas:

- A \cap C(A) =/= \emptyset
- A \cup C(A) =/= U

Variable lingüística: es caracterizada por una quíntupla (x, T(x), X, G, M) donde
x: variable base (nombre de la variable)
T(x): conjunto de términos lingüísticos de x que refieren a la variable base
X: conjunto universo
G: regla sintáctica (gramática) para generar términos lingüísticos
M: regla semántica que asigna a cada término un significado.



Implicaciones usuales:

- Zadeh: Max(1-p, Min(p,q)) → compatible con la lógica clásica
- Min (de Mamdani): Min(p,q) → NO compatible con la lógica clásica
- Luckasiewicz: Min(1, 1-p+q) → compatible con la lógica clásica
- Larsen: p * q → NO compatible con la lógica clásica

p	q	Zadeh	Mamdani	Luckas.	Larsen
1	1	1	1	1	1
1	0	0	0	0	0
0	1	1	0	1	0
0	0	1	0	1	0

Mamdani y Larsen (no compatibles con la lógica clásica) se usan para modelos causales donde las consecuencias SOLO se dan por la aparición de las causas => es falsa la implicación antecedente F y consecuente T. De esto se obtiene un modelo “implicador” como relación causa-efecto. implicancia a -> b como min(a,b) == pesimista, causa y efecto bien reflejadas

Inferencias:

1) Inferencia Mamdani max-min (valores crisp, una regla, forma gráfica)
REGLA: IF A THEN B

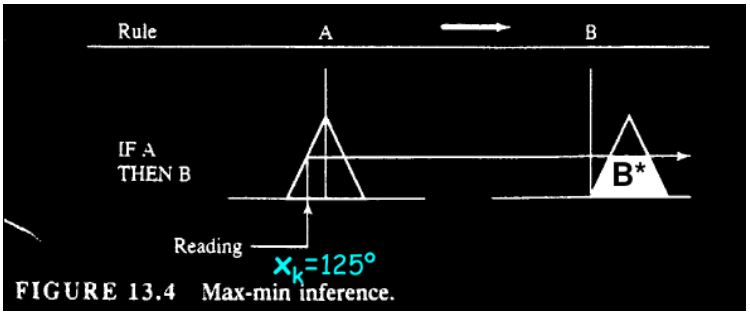
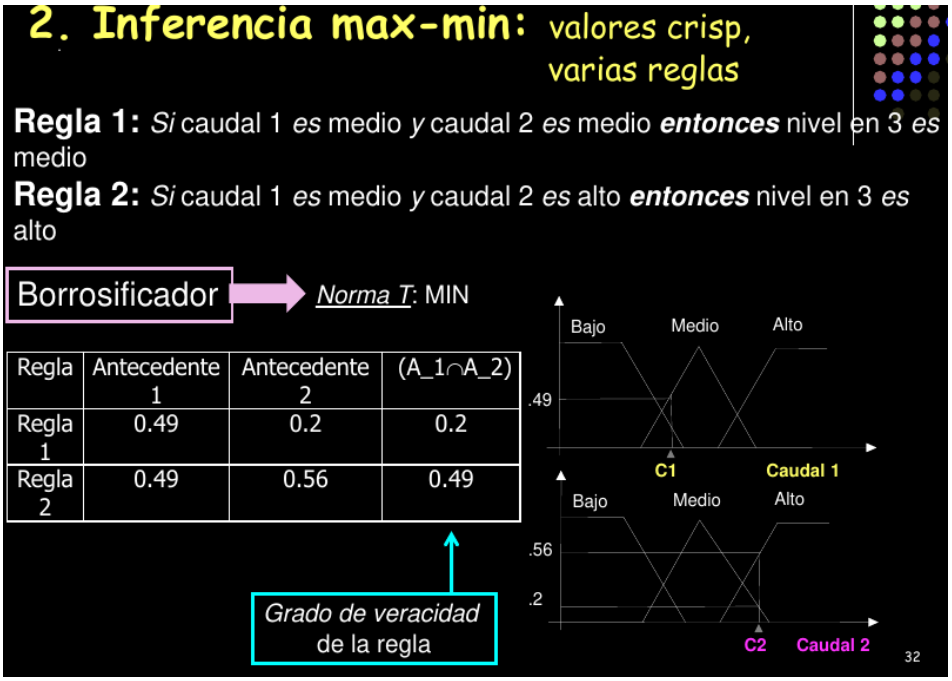
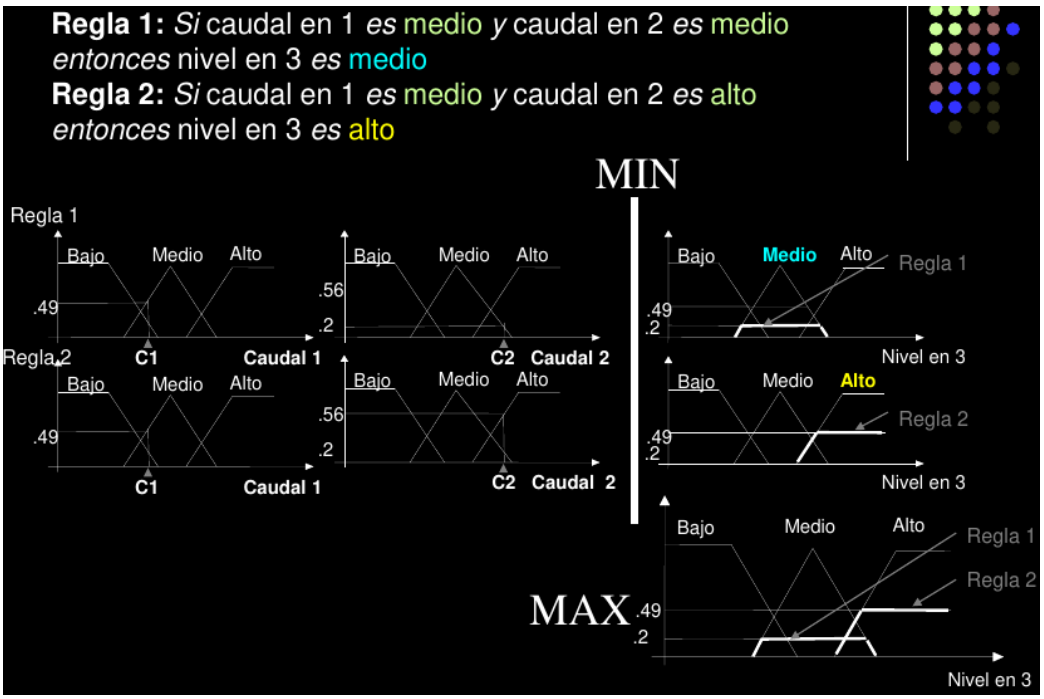
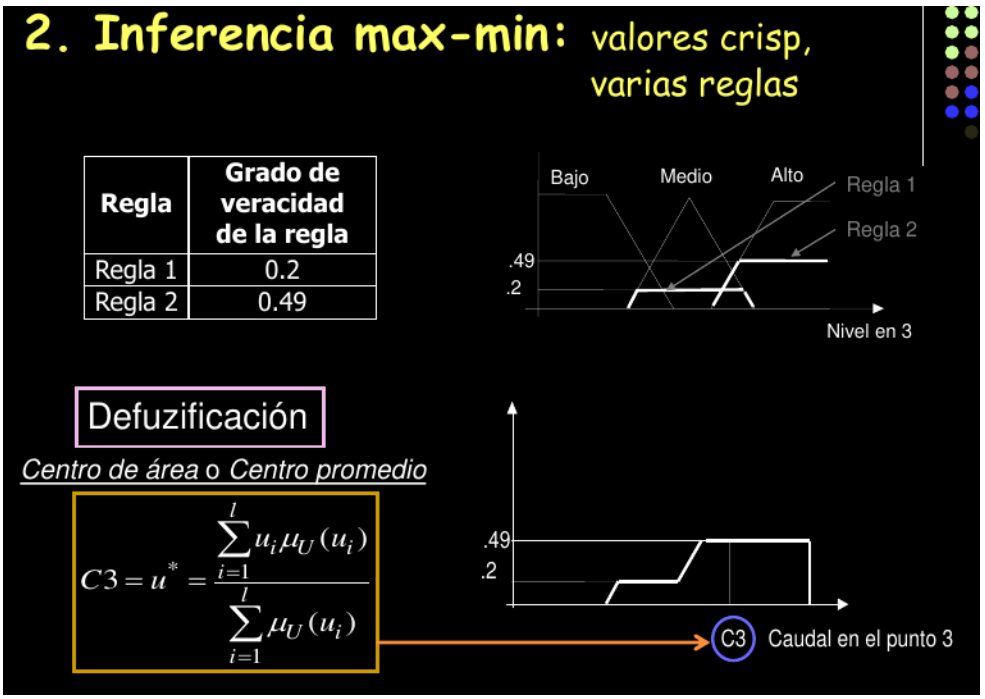


FIGURE 13.4 Max-min inference.

2) Inferencia max-min: valores crisp, varias reglas. Se aplica cuando las reglas tienen mismo consecuente.







(ver ejercicio entregado)

3) Inferencia borrosa: maxT con entrada borrosa
Si la entrada a la regla es una lectura difusa, nosotros podemos considerar la intersección de A y A*, es decir, min(a_i,a_i*) para inducir el B*

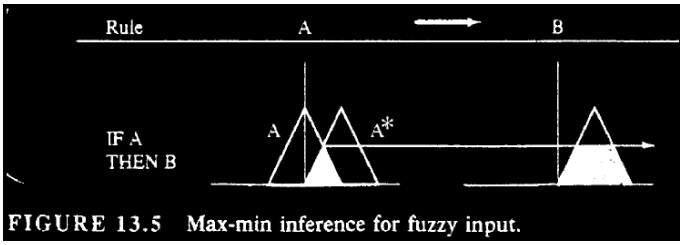
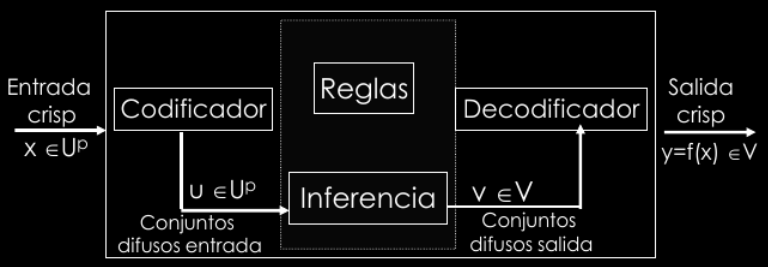


FIGURE 13.5 Max-min inference for fuzzy input.

Métodos de Defuzificación

- La salida de un proceso de inferencia es un conjunto difuso, en procesos en línea se requieren valores crisp



Métodos de Defuzificación

- Por ej.:

Centro de Gravedad

$$y = \frac{\sum_i b_i \int \mu(i)}{\sum_i \int \mu(i)}$$

Centros Promediados

$$y = \frac{\sum_i b_i \mu_{premisa}(i)}{\sum_i \mu_{premisa}(i)}$$

42

¿Cuándo usar lógica borrosa?
- En procesos complejos, si no existe un modelo de solución sencillo
-En procesos no lineales
- Cuando haya que introducir la experiencia de un operador “experto” que se base en conceptos imprecisos obtenidos de su experiencia
- Cuando ciertas partes del sistema a controlar son desconocidas y no pueden medirse de forma fiable
- Cuando el ajuste de una variable puede producir el desajuste de otras
- En general cuando se desea representar y operar con conceptos que tengan imprecisión o incertidumbre

Desventajas:
- Estabilidad: No hay garantía teórica que un sistema difuso no tenga un comportamiento caótico y no siga siendo estable, aunque tal posibilidad parece ser baja debido a los resultados obtenidos hasta ahora

- La determinación de las funciones de pertenencia y las reglas no siempre son sencillas
- La verificación de los modelos y sistemas borrosos expertos requiere de gran cantidad de pruebas

Notas:
se desfuzzifica sobre el eje de las abcisas (el rango dijo xd)
"si es algo riesgoso, yo tomo el primer máximo para no arriesgarme"
"no se nada del sistema, voy tomando los valores medios"

Aprendizaje automatizado

El Aprendizaje Automatizado introduce métodos que pueden resolver algunos problemas "aprendiendo" la solución a partir de ejemplos de cómo se realizan.

Problemas en AA:

- Clasificación:Dado un objeto (descrito por un conjunto de características medidas de alguna forma) asignarle una (o varias) etiqueta/s de un conjunto finito.
- Regresión: Dado un objeto asignarle un número real.(ejemplo: predecir el dólar de mañana)
- Ranking-Retrieval: Dado un objeto (necesidad de información/query), asignarle y ordenar las respuestas más adecuadas dentro de una base de datos (ejemplo buscador de internet)
- Detección de novedades: Detectar "outliers", objetos que son diferentes a los demás. (ej: Alarmas de comportamiento en compras con tarjeta)
- Clustering

¿Cómo se generaliza? Para generalizar incorporamos un "bias" a los datos. En general usamos la "navaja de Occam": La respuesta más simple que explica las observaciones es la válida. Diferentes métodos de machine learning usan diferente bias, pero en igualdad de condiciones la explicación más sencilla suele ser la más probable.

Un programa adquiere experiencia mediante de dos maneras:

- Aprendizaje supervisado: A partir de ejemplos suministrados por un usuario (un conjunto de ejemplos clasificados o etiquetados).
- Aprendizaje no supervisado: Mediante exploración autónoma (ej. software que aprende a jugar al ajedrez mediante la realización de miles de partidas contra sí mismo).

Tipos de aprendizaje:

- Aprendizaje inductivo: este se puede hacer con datos de entrada específicos (modelo general, el usuario provee un subconjunto con todas las posibles situaciones, llamado conjunto de entrenamiento) o con datos de salida generales (regla o modelo que puede ser usada en una situación.
  Tiene cuatro elementos fundamentales: modelo resultante (hipótesis), instancias, atributos y clases.
- Aprendizaje por refuerzo: son sistemas de aprendizaje no supervisado que aprenden sin datos de entrada, mediante prueba y error y que realizan exploración autónoma

(modelos) para inferir reglas de comportamiento. Se requiere un número de repeticiones muy elevado.
- Otros: Aprendizaje deductivo (EBL), Razonamiento basado en casos (CBR).

Resultado: modelo que se infiere a partir de los ejemplos.
Instancia: cada uno de los ejemplos.
Atributo: cada una de las propiedades que se miden (observan) de un ejemplo. Estos pueden ser reales, discretos o categóricos (o discretos no ordenados)
Clase: el atributo que debe ser deducido a partir de los demás.

Los modelos pueden ser de diversas formas:
–Árboles de decisión.
–Listas de reglas.
–Redes neuronales.
–Modelos bayesianos o probabilísticos.

¿Cómo resolver un problema de ML?
1) Identificar el problema
2) Conseguir datos, muchos datos.
3) Elegir un método adecuado (o varios)
4) Entrenar varios modelos con el conjunto de entrenamiento, evaluarlos con el conjunto de validación → decisiones fundamentales: el tipo de modelo, el algoritmo utilizado para construir o ajustar el modelo a partir de las instancias de entrenamiento.
5) Estimar el error con el conjunto de testeo

Árboles de decisión

- Compuestos de nodos y ramas.
- Representan reglas lógicas(if -then).
- Nodos internos= atributos(atributo=valor).
- Nodos hoja= clases.
- Nodo raíz= nodo superior del árbol.
- Objetivo en AA: Obtener un árbol de decisión(resultado) a partir de un conjunto de instancias o ejemplos.
- **Bias: árbol mínimo**

Selección del modelo y/o algoritmo
- Capacidad de representación: no muy elevada
- Legibilidad: muy alta. Uno de los mejores modelos en este sentido.
- Tiempo de cómputo on-line: muy rápido. Clasificar un nuevo ejemplo es recorrer el árbol hasta alcanzar un nodo hoja.
- Tiempo de cómputo off-line: rápido. Los algoritmos son simples.
- Dificultad de ajuste de parámetros
- Robustez ante el ruido: robusto.
- Sobreajuste: se controla a través de una poda.
- Minimización del error

Entrada: conjunto de entrenamiento, objetos caracterizables mediante propiedades (pares atributos-valor). La función objetivo toma valores discretos.
Salida: un árbol de clasificación (nodos hojas en una clase) o en árboles de decisión, una decisión.
-> Nos interesa tener un conjunto de reglas lógicas de clasificación (aunque posiblemente existan errores en los datos de entrenamiento robustos al ruido y/o falte información en algunos de los datos de entrenamiento)
-> La elección de nodos más altos se basa en la ganancia de información (o usando el método de Gini). Hay ganancia de información cuando la división envía instancias con clases distintas a los distintos nodos.

El espacio de hipótesis (para el algoritmo ID3) es el conjunto de todos los árboles posibles. ID3 realiza una búsqueda hill-climbing de lo simple a complejo (sin backtracking).

Entropía: Es la medida de la incertidumbre que hay en un sistema. Es decir, ante una determinada situación, la Probabilidad de que ocurra cada uno de los posibles resultados.
E(S) = - p+ * log_2(p+) - p- * log_2(p-)

donde siempre p+ y p- se cuentan a partir del total de elementos con un cierto atributo, por ejemplo: S1 es el subconjunto de S en el cual Humidity = High. De esos, cuántos dan positivos y cuántos negativos.

La ganancia de información mide la reducción esperada de entropía sabiendo el valor del atributo A

$$Gain(S,A) \equiv Entropía(S) - \sum_{v \in Valores(A)} (|Sv|/|S|)Entropía(Sv)$$

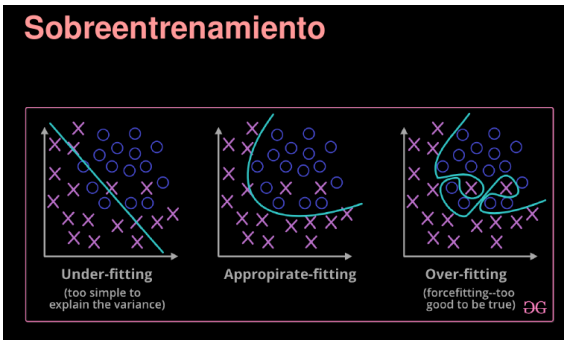
Valores(A) : conjunto de posibles valores del atributo A
Sv: subconjunto de S en el cual el atributo A tiene el valor v

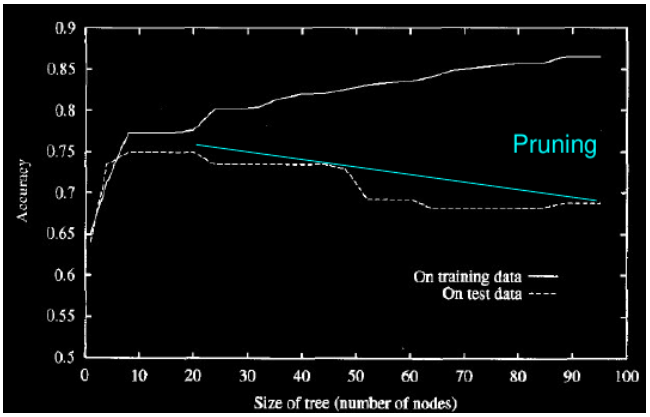
Otras medidas para decidir:
- Impureza de Gini (métodos estadísticos): se puede calcular sumando la probabilidad de cada elemento siendo elegido multiplicado por la probabilidad de un error en la categorización de ese elemento. Alcanza su mínimo (cero) cuando todos los casos del nodo corresponden a una sola categoría de destino.
- Reducción de la varianza

- ID3 nunca produce árboles demasiado grandes
- C4.5 ya que aunque discretice información continua, puede repetir atributos (ej: temp < 26, temp > 24, temp < 25, etc)
- Un árbol demasiado grande puede producir sobreajuste (overfitting) => es necesario podar los árboles (pruning)

Overfitting
Se da cuando un modelo es más complejo de lo que la función objetivo (generalización) debe ser, cuando trata de satisfacer datos ruidosos (lectura ruidosa, muestra chica).
En una hipótesis (modelo) se dice que existe sobreentrenamiento si existe alguna

otra hipótesis que tiene mayor error sobre los datos de entrenamiento pero menor error sobre todos los datos.





Se debe evitar el sobreentrenamiento, parando el crecimiento del árbol y realizando un post-procesamiento del árbol (poda). Otras maneras pueden ser usar un conjunto de ejemplos de validación y estadísticas.