Aprendizaje Automático

La idea del aprendizaje consiste en utilizar las percepciones no sólo para actuar, sino también para mejorar la habilidad del agente para actuar en el futuro. El aprendizaje entra en juego cuando el agente observa sus interacciones con el mundo y sus procesos de toma de decisiones. Se llama hipótesis a la función que se utiliza para aproximar f(x). Entre múltiples hipótesis consistentes, la mejor es la más simple. Esto es correcto, pero es importante resaltar que la razón principal para esta elección es que la función mas simple es la que mejor generaliza y que las funciones complejas tienden a sobre ajustarse a los datos de entrenamiento.

Reconocimiento de Patrones

Reconocimiento es considerado como una función básica del ser humano así como de otros organismos vivientes.

Patrón es la descripción de un objeto.

Podemos distinguir dos **categorías de reconocimiento**:

- **Reconocimiento perceptual**, como por ejemplo una forma (patrón espacial) o una secuencia (patrón temporal). **Objetos concretos**.
- **Reconocimiento conceptual**, tal como un viejo argumento o la solución de un problema. **Objetos abstractos**.

Puede tener 2 **objetivos**:

- Predicción: La mayoría de los casos, el objetivo es la clasificación.
- Comprender la relación entre las variables.

Patrón: descripción de un objeto definido como una relación entre sus características. Conjunto mínimo de características comunes a un universo de datos que permite identificar dichos datos como pertenecientes a diferentes clases. El reconocimiento de patrones está asociado al reconocimiento perceptual. Cuando una persona percibe un patrón, realiza una inferencia inductiva y asocia esta percepción con algunos conceptos generales o pistas derivados de su experiencia pasada.

- El **problema de reconocimiento** puede ser concebido como el de discriminar, clasificar o categorizar la información de entrada, no entre patrones individuales sino entre poblaciones, por medio de la búsqueda de características o atributos invariantes entre los miembros de una población.
- El **reconocimiento humano** es la estimación del parecido relativo entre datos de entrada y poblaciones conocidas. El **reconocimiento automático** es la clasificación de datos de entrada entre poblaciones mediante la búsqueda de características o atributos invariantes entre los miembros de cada población.

El diseño de un sistema de reconocimiento automático involucra por lo general las tareas siguientes (etapas para reconocimiento de patrones):

- Sensado.
- Extracción de Características.
- Clasificación.

Sensado

Sensado se refiere a la representación de la información obtenida mediante algún tipo de sensor sobre los objetos a ser reconocidos. Cada cantidad medida describe una característica del objeto. Esto puede evidenciarse suponiendo, por ejemplo, que se obtiene información sobre caracteres alfanuméricos. Es la obtención de datos.

En este caso puede considerarse un esquema de medición de grilla como el mostrado en el lazo izquierdo de la figura.

Esta grilla puede representarse mediante un vector patrón como sigue:

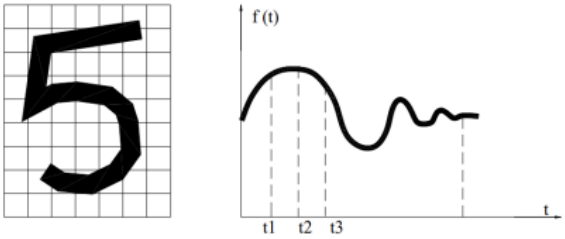


Fig. 2.1: Los objetos se representan mediante un vector patrón.

$$\mathbf{x} = \begin{pmatrix} x_1 \\ x_2 \\ \vdots \\ x_n \end{pmatrix} \quad con \quad x_i = 0 \quad o \quad x_i = 1$$

A la salida es común que la veamos como una escalar, pero también puede ser un vector. Donde se asigna a x_i el valor 1 si el elemento forma parte del carácter y 0 en caso contrario. En el caso de una imagen, x_i sería la intensidad del píxel i.

Variables cuantitativas: Se representan como números del tipo que sea necesario. Por ejemplo, peso, altura y edad son variables cuantitativas y se podrían representar con un float. En una imagen, las intensidades de los píxeles se pueden representar con enteros.

Variables categóricas o cualitativas: Pueden tomar solo un valor dentro de un conjunto limitado de valores. En el caso del nivel educativo, estos valores podrían ser "ninguno", "primario", "secundario", "terciario", "universitario o superior".

Extracción de Características

Lo que se logra con esto es reducir la dimensión de los patrones, pero con la posibilidad de perder un bajo porcentaje de la información contenida en ellos. Dentro de esta etapa podemos encontrar las siguientes tareas:

- Creación de nuevas características.
- Eliminación de características.
- Reducción de la dimensión.
- Otras transformaciones.

Clasificación

Determinar a qué clase pertenece cada vector de entrada: Para lograr esto, se requiere diseñar una mecanismo que, dado un conjunto de patrones de los cuales no se conoce a priori la pertenencia a la clase de cada uno de ellos, permita determinar a qué clase pertenece cada patrón.

Generación de límites de decisión entre regiones: Funciones de decisión. Se deben generar tantas funciones de decisión como clases haya. Tomando un dato como entrada,se evalúa en las x funciones, y el valor más alto es la clase a la que pertenece. Cuando 2 funcionesdan el mismo valor, el punto forma parte del límite entre las regiones.

Tipos de problemas reales: Linealmente separables y No linealmente separables. Si el sistema de reconocimiento puede autoajustar ciertos coeficientes internos que definen las funciones discriminantes estaremos en presencia de un sistema adaptivo o con capacidad de aprendizaje.

Definiendo a una clase como una **categoría** determinada por algunos atributos comunes y un patrón como la descripción de cualquier miembro de una categoría que representa a una clase de patrones, decimos que la teoría del Reconocimiento de Patrones se ocupa de buscar la solución al problema general de reconocer miembros de una clase dada en un conjunto que contienen elementos de muchas clases diferentes.

- Heurísticos: basados en la intuición y experiencia.
- Matemáticos:
    - Determinístico: de base estadística pero esta no está explicitada. Clasificadores entrenables.
    - Estadístico: de base estadística explícita. Ej: clasificador de Bayes.
- Sintácticos: basados en relaciones entre los objetos. Expresan una gramática. Útiles cuando los patrones no pueden ser apropiadamente descritos por mediciones (numéricamente).

Resultado de las etapas

El resultado de las etapas de sensado y extracción de características es un conjunto de vectores de características. Cada vector tiene la forma: $\boldsymbol{x} = [\boldsymbol{x_1}, \boldsymbol{x_2}, \ldots, \boldsymbol{x_n}]$ donde x_i es el valor de la característica i.

Después de obtener los vectores de características se realiza una de las siguientes tareas:

- **Clasificación**. Asignar cada vector de características a una clase. Por ejemplo, para detectar qué carácter aparece en una imagen.
- **Predicción o regresión.** Predecir un valor (contínuo) para cada vector de características. Por ejemplo, para calcular el valor futuro de ciertas acciones bursátiles.

Los métodos de aprendizaje automático que vemos son clasificadores, pero con un mínimo cambio es posible adaptarlo para hacer predicciones.

Función de Decisión

Suponiendo un problema de reconocimiento de c clases distintas denominadas $\omega_1, \ldots, \omega_c$, puede considerarse al espacio de las entradas compuesto por c regiones, donde cada una de las cuales contiene a los elementos de una clase. La solución puede interpretarse como la generación de límites de decisión entre las regiones. Estos límites pueden estar dados por funciones de decisión o funciones discriminantes $d_1(x), \ldots, d_c(x)$, que son funciones escalares de los vectores de entrada. La función que obtiene el mayor valor es la que determina la pertenencia del vector a la clase correspondiente a esa función, es decir, si $d_i(x) > d_j(x)\ para\ i, j = 1, \ldots, c$ y $i \neq j$, entonces x pertenece a la clase ω_i.

En los problemas, muy frecuentes, donde la salida solo puede pertenecer a una clase de un total de dos clases (clasificación binaria), usualmente se utiliza una única función de decisión $d(x)$. Si $d(x > 0)$, x pertenece a una clase, en caso contrario, a la otra.

Clasificador Lineal

El caso más simple de función de decisión es la función de decisión lineal. El modelo de clasificación que implementa esta función se llama clasificador lineal. Para un vector de entradas $x^T = [x_1, \ldots, x_n]$, la función de decisión lineal es $d(x) = w_1 x_1 + \cdots + w_n x_n + w_{n+1}$, donde los coeficientes $w_i \in R$ son los parámetros del clasificador. El vector formado por todos los coeficientes se llama vector de parámetros $w = [w_i, \ldots, w_{n+1}]$. Es útil calcular el valor de la función de decisión como un producto entre vectores, para lo cual el vector x se redefine como $x^T = x_1, \ldots, x_n, 1]$, lo que lleva el nombre de vector de entradas aumentado. Con el nuevo vector x, la función de decisión se puede calcular como $d(x) = w_x$.

Para el caso una clasificación binaria, x pertenecerá a la clase ω_1 si $d(x) < 0$ y pertenecerá a la clase ω_2 si $d(x) > 0$. Entonces, la frontera entre las clases en el espacio de las entradas queda definida por el hiperplano $d(x) = 0$. Los elementos de la clase ω_1 están pintados de color amarillo y los de ω_2 en color azul. La función de decisión se ve como un plano inclinado y la frontera entre las clases se indica con la línea de puntos.

El entrenamiento del modelo se realiza ajustando los valores del vector de parámetros w. En primer lugar, se crea un vector $y = [y_1, \ldots, y_L]$ con las salida esperada para cada vector de entradas, donde $y_j = -1$ para $x_j \in \omega_1$ y $y_j = 1$ para $x_j \in \omega_2$. Después, se calcula el valor óptimo de w como la combinación de parámetros que minimiza la función del error cuadrático medio $mse = \frac{1}{L} * \sum_{j=1}^{L} (y_j - w x_j)^2$. Partiendo de la derivada del error cuadrático medio con respecto a los parámetros $\partial mse / \partial w$, se obtiene la expresión del vector de parámetros óptimo $w = (X \cdot X^T)^{-1} \cdot X \cdot y^T$, donde X es la matriz formada por los L vectores de entrada.

Redes Neuronales Artificiales

Una **neurona** es una célula del cerebro cuya función principal es la recogida, procesamiento y emisión de señales eléctricas. Se piensa que la capacidad de procesamiento de información del cerebro proviene principalmente de *redes* de este tipo de neuronas. Por esta razón, algunos de los primeros trabajos en IA pretendían crear **redes neuronales** artificiales.

Las redes neuronales están compuestas de nodos o **unidades** conectadas a través de **conexiones** dirigidas. Una conexión de la unidad j a la unidad i sirve para propagar la **activación** a_j de j a i. Además, cada conexión tiene un **peso** numérico $W_{j,i}$ asociado, que determina la fuerza y el signo de la conexión.

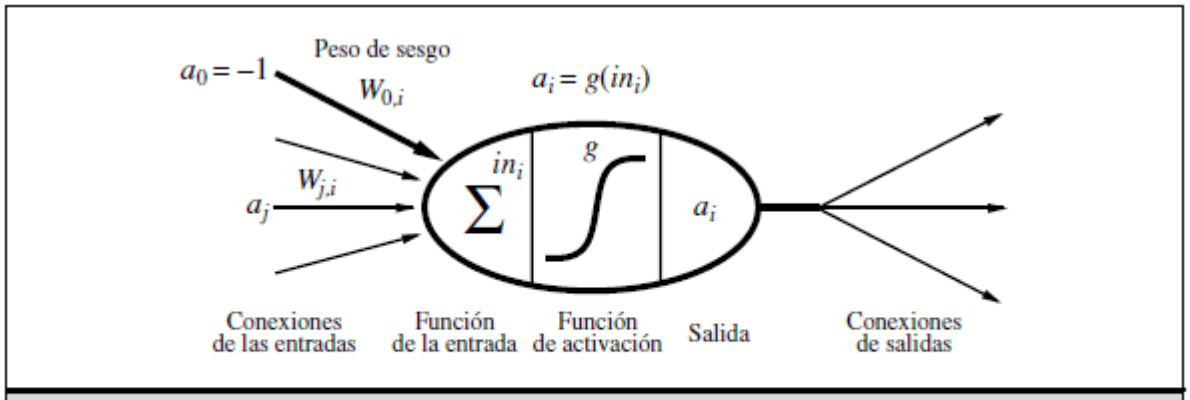


Figura 20.15 Un modelo matemático sencillo para una neurona. La activación de la salida de la unidad es $a_i = g(\sum_{j=0}^{n} W_{j,i} a_j)$, donde a_j es la activación de la salida de la unidad j y $W_{j,i}$ es el peso de la conexión de la unidad j a esta unidad.

Nótese que se ha incluido un **peso de sesgo** $W_{0,i}$ conectado a una entrada fija $a_0 = -1$.

Cada unidad *i* primero calcula una suma ponderada de sus entradas:

$$in_i = \sum_{j=0}^{n} W_{j,i}\, a_j$$

Luego aplica una **función de activación** g a esta suma para producir la salida:

$$a_i = g(in_i) = g\left(\sum_{j=0}^{n} W_{j,i}\, a_j\right)$$

La función de activación *g* se diseña con dos objetivos:

1. Queremos que la unidad esté «activa» (cercana a +1) cuando se proporcionen las entradas «correctas», e «inactiva» (cercana a 0) cuando se den las entradas «erróneas».
2. La activación tiene que ser no *lineal*, en otro caso la red neuronal en su totalidad se colapsaría con una sencilla función lineal.

Perceptron

Una red con todas las entradas conectadas directamente a las salidas se denomina red neuronal de una sola capa, o red perceptrón. Es un método de aprendizaje supervisado que realiza una clasificación binaria en base a una transformación lineal, al igual que el clasificador lineal.

El perceptron representa una neurona biológica donde las dendritas son las entradas del perceptron, el axón es la salida, las sinapsis son los coeficientes de la función de decisión y el comportamiento (muy simplificado) se replica acumulando la intensidad de los impulsos recibidos que, al superar cierto umbral, "activan" la neurona emitiendo una señal por la salida.

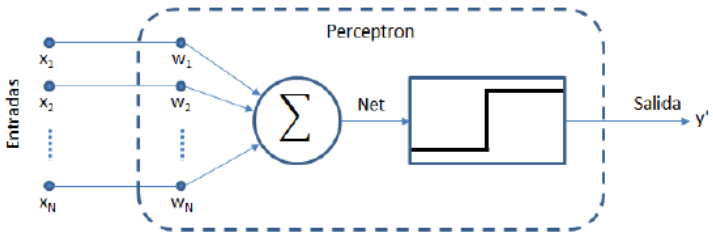


Figura 8.5: Esquema del perceptron.

El perceptrón devuelve 1 si y sólo si la suma ponderada de sus entradas (incluyendo los sesgos) es positiva:

$$\sum_{j=0}^{n} W_j x_j > 0 \quad o \quad \boldsymbol{W} \cdot \boldsymbol{x} > 0$$

La ecuación $\boldsymbol{W} \cdot \boldsymbol{x} = 0$ define un *hiperplano* en el espacio de entrada, así que el perceptrón devuelve 1 si y sólo si la entrada está en un lado de ese hiperplano. Por esta razón, el perceptrón se denomina **separador lineal**.

El perceptrón puede representar sólo funciones **linealmente separables**.

El ajuste de los pesos del perceptron se realiza con la siguiente ley de aprendizaje: $\Delta w_i = \alpha(y - y')x_i$. Donde y es la salida esperada y α es la tasa de aprendizaje. El proceso es el siguiente:

- Inicializar los pesos con valores aleatorios.
- Mientras $error > 0$ para algún vector de entradas, hacer:
  - Ejecutar ciclo completo de entrenamiento ($epoch$). Mientras existan vectores de entrenamiento hacer:
    * Tomar un vector entrada/salida del conjunto de datos de entrenamiento.
    * Calcular la salida $y' = f\left(\sum_{i=0}^{N} w_i x_i\right)$
    * Calcular el $error = y - y'$
    * Calcular $\Delta_{w_i} = \alpha(y - y')x_i$
    * Modificar los pesos haciendo $w_{i_{t+1}} = w_{i_t} + \Delta_{w_i}$

Notar que tal como está declarado el método, y a menos que se defina una cantidad máxima de ciclos, el entrenamiento termina solo cuando todos los vectores están bien clasificados.

El perceptron tiene la misma desventaja que el clasificador lineal, solo es capaz de clasificar correctamente las entradas de problemas linealmente separables. Este problema NO se puede solucionar, el perceptron NO puede resolver el problema, sin embargo, hay una alternativa **teórica** que involucra otro modelo y abre un camino para tratar este tipo de problemas. Conectando perceptrones entre sí, en lo que se llama perceptron multicapa, se podría resolver el problema **si supiéramos cómo adaptar los pesos**.

¿Por qué no se puede aplicar este modelo? No sabemos cómo entrenarlo... No sabemos cómo modificar los pesos de los perceptrones de la primera capa porque no sabemos cuánto contribuyó cada uno al error de la última salida final.

Adaline

Adaline (ADAptative LINear Element) es un modelo de red neuronal artificial que tiene dos diferencias con respecto al perceptron.

La primera diferencia es la función de activación, en Adaline se utiliza una función lineal. La salida de Adaline es directamente $y' = Net$. Si este modelo se aplica a una predicción del tipo regresión (cosa imposible para el perceptron), la salida utilizada es Net. En el caso de la clasificación, será necesario utilizar una función escalón en algún momento. La función de activación de Adaline es lineal y, fuera del modelo, se aplica una función umbral para determinar la clase.

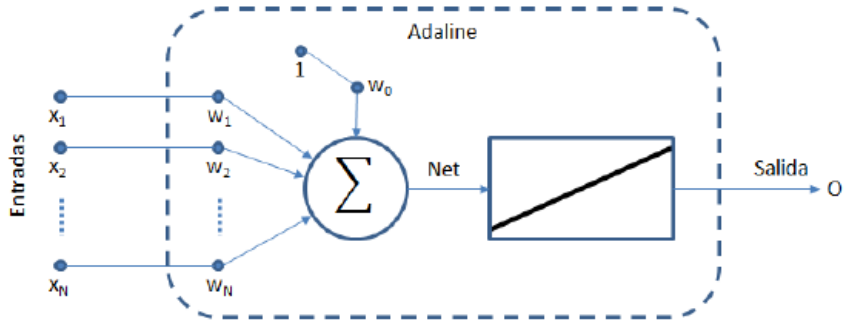


Figura 8.12: Esquema de Adaline.

La segunda diferencia es la ley de aprendizaje. En Adaline se utiliza la regla delta, basada en el mínima error cuadrático medio (Least Mean Squared o LMS error). Los pesos de la red se ajustan mediante una búsqueda guiada por el gradiente descendiente de la función del error.

- O: salida de la red (por *output*). En Adaline $O = \sum_{i=0}^{N} w_i x_i = Net$.
- k: identificador de un vector de entrenamiento.
- E: error cuadrático medio. $E = \frac{1}{2L} \sum_{k=1}^{L} (y_k - O_k)^2$, donde L es la cantidad de vectores de entrenamiento.

La adaptación de los pesos se lleva a cabo a través de una búsqueda sobre la superficie del error. Para encontrar el mínimo de la función del error los pesos se modifican en cantidades proporcionales al gradiente decreciente de la función E. La expresión de la ley de aprendizaje de Adaline es igual a la del perceptron, pero en el caso de Adaline el error puede tomar cualquier valor en R.

$$\begin{aligned} \Delta_{w_{ik}} &= -\alpha\delta\ x_i \\ &= -\alpha\left(-(y_k - O_k)\right) x_i \\ &= \alpha(y_k - O_k) x_i \end{aligned}$$

El método de entrenamiento también es similar al del perceptron, con la diferencia de que el error nunca es cero, así que se necesita otra condición de corte. Típicamente se corta el entrenamiento cuando se llega a un umbral de error previamente definido. En cuanto a las limitaciones, Adaline solo puede clasificar problemas que sean linealmente separables.

Metaheurísticas

Introducción

A menudo resulta imposible calcular soluciones óptimas para problemas de optimización con importancia industrial y/o científica. Muchas veces un agente inteligente puede estar satisfecho con soluciones buenas que no son óptimas o no se puede comprobar que lo sean. Las metaheurísticas representan una familia de técnicas de optimización aproximada que proporcionan soluciones aceptables en un tiempo razonable para resolver problemas complejos.

El sufijo “meta” es una palabra griega que significa “después” o “más allá” y que suele ser usado para hacer referencia a una abstracción de nivel superior. En el término “metaheurística” indica que hay un procedimiento heurístico que hace uso de una heurísticas o de otro método heurístico. Podemos pensar en las metaheurísticas como métodos de búsqueda de nivel superior o metodologías generales (templates) que pueden utilizarse como estrategias guía en el diseño de heurísticas subyacentes para resolver problemas de optimización específicos.

Al diseñar una metaheurística se deben tener en cuenta dos criterios contradictorios: la exploración del espacio de búsqueda (diversificación) y la explotación de las mejores soluciones encontradas (intensificación). Las regiones prometedoras se determinan a través de las soluciones buenas obtenidas. La intensificación se refiere a la exploración más a fondo de las regiones prometedoras con la esperanza de encontrar mejores soluciones. La diversificación implica la visita de regiones no exploradas para asegurarse de que todas las regiones del espacio de búsqueda se exploren de manera equitativa y de que la búsqueda no se limite solo a una zona reducida.

Se pueden utilizar diversos criterios de clasificación para las metaheurísticas:

- Inspiradas en la naturaleza vs. no inspiradas en la naturaleza. Muchas metaheurísticas están inspiradas en procesos naturales, como los algoritmos evolutivos y sistemas inmunitarios artificiales (biología); colonias de hormigas, colonias de abejas y enjambres de partículas (ciencias sociales); y el enfriamiento simulado (física).
- Con memoria vs. sin memoria. Algunos algoritmos metaheurísticos no recuerdan, es decir, no utilizan información extraída durante la búsqueda, por ejemplo la búsqueda local, GRASP y el enfriamiento simulado. El otro grupo si se apoya en esa información, por ejemplo, la búsqueda tabú.
- Deterministas vs. estocásticos. En métodos deterministas (por ejemplo, búsqueda local y búsqueda tabú) la solución utilizada como punto de partida lleva siempre a la misma solución final. Los métodos estocásticos tienen la posibilidad de tomar decisiones distintas para la misma situación, lo que les da mayor variabilidad en los caminos de búsqueda (por ejemplo, enfriamiento simulado y algoritmos evolutivos).
- Basados en población vs. basados en una solución. Los algoritmos basados en una sola solución (por ejemplo, búsqueda local y enfriamiento simulado) manipulan y transforman una sola solución durante la búsqueda, mientras que en los algoritmos basados en población (por ejemplo, enjambre de partículas y algoritmos evolutivos) se adapta toda un conjunto de soluciones. Estas dos familias tienen características complementarias: las basadas en una solución están orientadas a la explotación; tienen la capacidad de intensificar la búsqueda en regiones locales. Las metaheurísticas basadas en población están orientadas a la exploración; permiten una mejor diversificación en todo el espacio de búsqueda.
- Iterativos versus avaros. Los algoritmos iterativos comienzan con una solución completa (o población de soluciones) y la transforman en cada iteración. Los algoritmos avaros (greedy) parten de una solución vacía y, en cada paso, asignan una variable de decisión del problema hasta obtener una solución completa. La mayoría de las metaheurísticas son algoritmos iterativos.

¿Cuándo utilizar Meta Heurísticas?

Depende de la complejidad del problema y del tamaño de las instancias que se supone que el algoritmo debe resolver, es decir de O(g(n)) y de n. Incluso si un problema es NP-duro, instancias pequeñas pueden resolverse mediante un método exacto. Además, la estructura de las instancias desempeña un papel importante. Algunas instancias de tamaño mediano o incluso grande con una estructura específica pueden resolverse de manera óptima mediante métodos exactos. Por último, el tiempo requerido para resolver un problema dado es un dato muy importante en la selección del algoritmo de optimización.

No es prudente utilizar metaheurísticas para resolver problemas para los cuales existen algoritmos exactos eficientes (clase P). Si un algoritmo exacto proporciona un tiempo de búsqueda aceptable para resolver un caso particular, las metaheurísticas son una mala decisión. Por ejemplo, no se debe usar una metaheurística para encontrar un árbol de expansión mínimo o el camino más corto en un grafo porque existen algoritmos exactos conocidos de tiempo polinómico para esos problemas.

Algoritmos Heurísticos

Los algoritmos heurísticos son los más fáciles de utilizar, ya que se basan en el conocimiento de una heurística que guía el proceso de búsqueda. El conocimiento del problema usualmente ayuda a encontrar una heurística razonable que encontrará rápidamente una solución aceptable. Un algoritmo de este tipo sólo buscará dentro de un subespacio del área total a una solución buena (que no necesariamente es la mejor) que satisfaga las restricciones impuestas. La principal limitación es su incapacidad para escapar de óptimos locales encontrar soluciones parcialmente óptimas).

Algoritmos Metaheurísticos

Una metaheurística es un proceso iterativo maestro que guía y modifica las operaciones de una heurística subordinada para producir eficientemente soluciones de alta calidad. Las metaheurísticas pueden manipular una única solución completa (o incompleta) o una colección de soluciones en cada iteración. La heurística subordinada puede ser un procedimiento de alto o bajo nivel, una búsqueda local, o un método constructivo. Entre los algoritmos metaheurísticos más conocidos están, el recocido simulado y los algoritmos genéticos.

Cuatro tipos:

1. Las metaheurísticas de relajación se refieren a procedimientos de resolución de problemas que utilizan relajaciones del modelo original (es decir, modificaciones del modelo que hacen al problema más fácil de resolver), cuya solución facilita la solución del problema original.

2. Las metaheurísticas constructivas se orientan a los procedimientos que tratan de obtener una solución a partir del análisis y selección paulatina de las componentes que la forman.

3. Las metaheurísticas de búsqueda guían los procedimientos que usan transformaciones o movimientos para recorrer el espacio de soluciones alternativas y explorar las estructuras de entornos asociadas.

4. Las metaheurísticas evolutivas están enfocadas a los procedimientos basados en conjuntos de soluciones que evolucionan sobre el espacio de soluciones.

   Las metaheurísticas evolutivas establecen estrategias para conducir la evolución en el espacio de búsqueda de conjuntos de soluciones (usualmente llamados poblaciones) con la intención de acercarse a la solución óptima con sus elementos. El aspecto fundamental de las heurísticas evolutivas consiste en la interacción entre los miembros de la población frente a las búsquedas que se guían por la información de soluciones individuales.

   Las diferentes metaheurísticas evolutivas se distinguen por la forma en que combinan la información proporcionada por los elementos de la población para hacerla evolucionar mediante la obtención de nuevas soluciones.

   Los algoritmos genéticos y meméticos y los de estimación de distribuciones emplean fundamentalmente procedimientos aleatorios, mientras que las metaheurísticas de búsqueda dispersa o de reencadenamiento de caminos (Path-Relinking) emplean procedimientos sistemáticos.

Metaheurísticas – Algoritmos Genéticos

Un **algoritmo genético** (o AG) es una variante de la búsqueda de haz estocástica en la que los estados sucesores se generan combinando *dos* estados padres, más que modificar un solo estado. Replican el modelo de selección natural propuesto por Darwin, y que resume la famosa frase 'la supervivencia del más fuerte o adaptado'.

Los AGs comienzan con un conjunto de *k* estados generados aleatoriamente, llamados **población**. Cada estado, o **individuo**, está representado como una cadena sobre un alfabeto finito (el más común, una cadenas de 0s y 1s).

Este modelo básicamente dice que, dentro de una población, los individuos que sobreviven son aquellos que están más adaptados al medio, por lo tanto, las generaciones futuras de estos estarán mejor adaptadas ya que serán combinaciones de los mejores genes de sus antepasados. Además, esta teoría de la evolución introduce un concepto muy interesante que son las mutaciones. Una mutación es un pequeño cambio que se produce de manera aleatoria en ciertos individuos e introduce de esta manera versatilidad en las poblaciones. Habrá mutaciones que den lugar a cambios favorables y otros desfavorables.

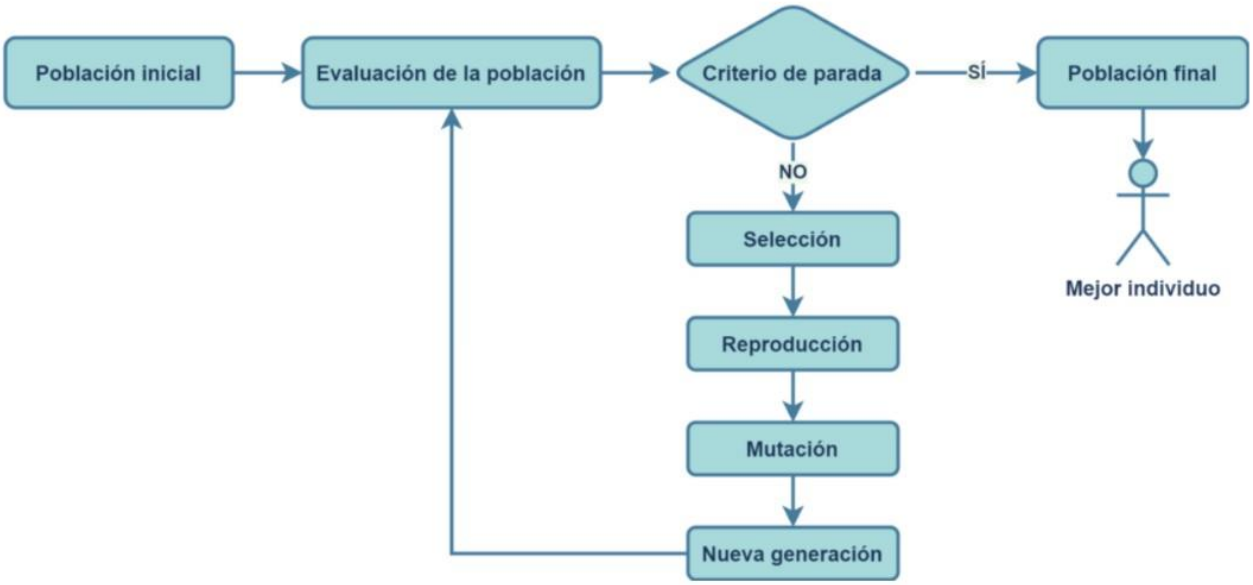
Se utilizan para resolver problemas de Búsqueda y Optimización, ya que se basan en evolucionar poblaciones de soluciones hacia valores óptimos del problema.

Estructura de un algoritmo genético

Tener en cuenta:

- Individuo: los individuos de la población son las posibles soluciones al problema que se intenta resolver.
- Población: conjunto de individuos.
- Función fitness o de adaptación: función que evalúa a los individuos y les asigna una puntuación en función de que tan buenas sean las soluciones para el problema.
- Función de cruce: función que dados dos individuos genera dos descendientes a partir de la combinación de genes de sus padres, esta función depende del problema en cuestión.

Cada estado se tasa con la función de evaluación o (en terminología AG) la **función idoneidad**. Una función de idoneidad debería devolver valores más altos para estados mejores. Se seleccionan dos pares, de manera aleatoria, para la reproducción, de acuerdo con las probabilidades. Notemos que un individuo se selecciona dos veces y uno ninguna. Para que cada par se aparee, se elige aleatoriamente un punto de **cruce** de las posiciones en la cadena. Los descendientes se crean cruzando las cadenas paternales en el punto de cruce. Finalmente, cada posición está sujeta a la **mutación** aleatoria con una pequeña probabilidad independiente.



- Fase inicial: se genera una población inicial de individuos (soluciones)
- Fase de evaluación: se evalúan los individuos de la población con la función fitness
- Fase de selección: se seleccionan los mejores individuos
- Fase de reproducción: se cruzan los individuos seleccionados mediante la función de cruce, dando lugar a una nueva generación que va a sustituir a la anterior
- Fase de mutación: se introducen mutaciones (pequeños cambios) en ciertos individuos de la nueva población de manera aleatoria o un entrecruzamiento cromosómico (también llamado crossover o recombinación)
- Se obtuvo una nueva generación, en general, con soluciones mejores que la anterior. Se vuelve al punto 2

Los algoritmos genéticos pueden finalizar cuando se alcanza un número de generaciones concreto o cuando cumplen una condición de parada.

Ventajas

Se desenvuelven bien en problemas con un paisaje adaptativo complejo: aquéllos en los que la función de aptitud es ruidosa, cambia con el tiempo, o tiene muchos óptimos locales, gracias a los cuatro componentes principales de los

algoritmos genéticos, paralelismo, selección, mutación y cruzamiento, los que trabajan juntos para conseguir su buen desempeño.

Pueden explorar el espacio de soluciones en múltiples direcciones a la vez, por lo que, los algoritmos genéticos funcionan particularmente bien resolviendo problemas cuyo espacio de soluciones potenciales es realmente grande, demasiado vasto para hacer una búsqueda exhaustiva en un tiempo razonable

Los algoritmos genéticos realizan cambios aleatorios en sus soluciones candidatas y luego utilizan la función de aptitud para determinar si esos cambios producen una mejora. Como sus decisiones están basadas en la aleatoriedad, todos los caminos de búsqueda posibles están abiertos; en contraste a cualquier otra estrategia de resolución de problemas que dependa de un conocimiento previo

Desventajas

Si se elige mal una función de aptitud o se define de manera inexacta, puede que el algoritmo genético sea incapaz de encontrar una solución al problema, o puede acabar resolviendo el problema equivocado.

Pueden tardar mucho en converger, o no converger en absoluto, dependiendo en cierta medida de los parámetros que se utilicen tamaño de la población, el ritmo de mutación y cruzamiento, el tipo y fuerza de la selección.

El lenguaje utilizado para especificar soluciones candidatas debe ser robusto; es decir, debe ser capaz de tolerar cambios aleatorios que no produzcan constantemente errores fatales o resultados sin sentido. Una de las formas mas usadas es definir a los individuos como listas de números -binarios, enteros o reales- donde cada número representa algún aspecto de la solución candidata.

Modelos Bayesianos

Comportamiento Bajo Incertidumbre

Cuando un agente conoce hechos suficientes sobre su entorno, el enfoque de la lógica le permite obtener planes con garantías para su desarrollo. Desafortunadamente, *los agentes casi nunca tienen acceso a toda la verdad sobre su entorno*. Los agentes deben, así, saber comportarse bajo **incertidumbre**.

Para un agente lógico, sería imposible construir una descripción completa y exacta de cómo se desarrollarán sus acciones. Suponga, por ejemplo, que el agente quiere conducir algo para tomar un vuelo y considera un plan, *A*90, que supone dejar la casa 90 minutos antes de que salga el vuelo y conducir a una velocidad razonable. Incluso aunque el aeropuerto esté tan sólo a unas 15 millas, el agente no inferirá con certidumbre que «El Plan *A*90 nos llevará al aeropuerto a tiempo». En lugar de eso, llega a la conclusión más débil «El Plan *A*90 nos llevará al aeropuerto a tiempo, siempre que mi coche no se averíe o se quede sin gasolina, y no tenga un accidente, y no haya accidentes en el puente, y el avión no salga anticipadamente, y…». Ninguna de estas condiciones puede deducirse, por lo que no puede inferirse el éxito del plan. Este es un ejemplo del **problema de la cualificación.**

Si un agente lógico no puede concluir que cualquier secuencia concreta de acciones alcance su objetivo, entonces será incapaz de actuar. La planificación condicional puede derrotar a la incertidumbre hasta cierto punto, sólo si las acciones del agente pueden obtener la información que se requiere y sólo si no hay demasiados imprevistos diferentes. Otra posible solución sería dotar al agente con una simple pero errónea teoría del mundo que le *permita* deducir un plan; presumiblemente, esos planes funcionarán *la mayoría* de las veces, pero los problemas se presentan cuando los acontecimientos contradicen la teoría del agente. Más aún, el tratamiento del equilibrio entre la precisión y la utilidad de la teoría del agente parece en sí mismo que requiere razonamiento sobre incertidumbre. En resumen, los agentes que no sean puramente lógicos serán capaces de concluir que el plan *A*90 es el correcto a desempeñar.

Aún con todo, supongamos que *A*90 es el apropiado a realizar. ¿Qué queremos decir al afirmar esto? Se espera que el *A*90 maximice la medida de desarrollo del agente, dada la información que tiene sobre el entorno. La medida de desarrollo incluye el llegar a tiempo al aeropuerto para el vuelo, evitar una espera larga e improductiva en el aeropuerto, y evitar prisas en poner etiquetas cerca del pasaje. La información que tiene el agente no puede garantizar ninguno de estos resultados finales para *A*90, pero puede proporcionar algún grado de creencia de que pueda lograrse. Otros planes, tal como el *A*120, podrían incrementar la creencia del agente de que llegará al

aeropuerto a tiempo, pero también pueden incrementar la posibilidad de una mayor espera. *Lo correcto a realizar (la **decisión racional**) depende tanto de la importancia relativa de los distintos objetivos como de la verosimilitud y el grado con el cual se conseguirán*.

Manipulación del Conocimiento Incierto

El diagnóstico (ya sea para medicina, reparación de automóviles, o lo que quiera) es una tarea que casi siempre involucra incertidumbre. Intentemos escribir las reglas para diagnóstico dental usando la lógica de predicados de primer orden, para que así podamos ver cómo fracasa la aproximación lógica. Considere la siguiente regla:

$$\forall p \; Síntoma(p, Dolor - de - muelas) \;\Rightarrow\; Enfermedad(p, Caries)$$

El problema es que esta regla es errónea. No todos los pacientes con dolor de muelas tienen caries; algunos tienen dolencia de encías, una acumulación de pus (absceso), u otro problema distinto. Desafortunadamente, para hacer la regla cierta, tenemos que añadir una lista casi ilimitada de causas posibles. El único modo para arreglar la regla es hacerla lógicamente exhaustiva: incrementar el lado izquierdo de la regla con todos los requisitos necesarios para que una caries cause un dolor de muelas. Incluso entonces, para los propósitos del diagnóstico, uno debe también tener en cuenta la posibilidad de que el paciente podría tener un dolor de muelas y una caries que no estén relacionados.

Nuestra principal herramienta para tratar con grados de creencia será la **teoría de la probabilidad**, que asigna a cada oración un grado numérico de creencia entre 0 y 1. La probabilidad proporciona una manera de **resumir** la incertidumbre. No podríamos saber con seguridad lo que aqueja a un paciente particular, pero podríamos creer que hay, un 80 por ciento de posibilidad (esto es, una probabilidad de 0,8) de que el paciente tiene una caries si se tiene un dolor de muelas. Esto es, hasta donde llega el conocimiento del agente esperamos que de todas las situaciones que son indistinguibles de la situación actual, el paciente tendrá una caries en el 80 por ciento de ellas. Esta creencia podría provenir de datos estadísticos (el 80 por ciento de los pacientes observados hasta este momento con dolor de muelas han tenido caries) o de algunas reglas generales, o de una combinación de fuentes de indicios. El 20 por ciento restante resume todas las otras causas posibles de dolor de muelas para los que somos demasiado perezosos o ignorantes en confirmar o desmentir.

Asignar probabilidad 0 a una oración determinada corresponde a una creencia inequívoca de que la oración es falsa, mientras que asignar una probabilidad de 1 corres ponde a una creencia rotunda de que la oración es cierta. Las probabilidades entre 0 y 1 corresponden a grados intermedios de creencia en la veracidad de la oración. Una probabilidad de 0,8 no significa «80 por ciento verdadero» sino un 80 por ciento de grado de creencia, eso es, una expectativa realmente fuerte.

Todas las oraciones de probabilidad deben así indicar la evidencia con respecto a la cual se está calculando la probabilidad. Cuando un agente recibe nuevas percepciones, sus valoraciones de probabilidad se actualizan para reflejar la nueva evidencia. Antes de que la evidencia se obtenga, hablaremos de probabilidad *a priori* o **incondicional**; después de obtener la evidencia, hablaremos de probabilidad *a posteriori* o **condicional**. En la mayoría de los casos, un agente tendrá alguna evidencia de sus percepciones y será interesante calcular las probabilidades posteriores de las consecuencias que le preocupan.

Probabilidad a priori o incondicional

La probabilidad a priori o incondicional asociada a una proposición A es el grado de creencia que se le otorga en ausencia de cualquier otra información y se escribe como $P(A)$. Por ejemplo, si la probabilidad priori de que tenga una caries es 0,1, entonces deberíamos escribir $P(Caries = cierto) = 0{,}1$ *o* $P(caries) = 0{,}1$.

Es importante recordar que *P*(*a*) puede usarse sólo cuando no hay otra información. Tan pronto como se conozca nueva información, debemos razonar con la probabilidad *condicional* de *a* dada esa nueva información. Las probabilidades condicionales se tratarán en la sección siguiente.

También usaremos expresiones como $\boldsymbol{P}(Tiempo, Caries)$ para denotar las probabilidades de todas las combinaciones de los valores de un conjunto de variables aleatorias. En ese caso, $\boldsymbol{P}(Tiempo, Caries)$ puede

representarse por una tabla de probabilidades de dimensión 4×2. Ésta se llama la **distribución de probabilidad conjunta** de *Tiempo* y *Caries*.

A veces será útil pensar sobre el conjunto completo de variables aleatorias que se utilicen para describir el mundo. Una distribución de probabilidad conjunta que considere este conjunto completo se llama la **distribución de probabilidad conjunta completa**. Por ejemplo, si el mundo consta exactamente de las variables *Caries*, *Dolor-de-muelas* y *Tiempo*, entonces la distribución conjunta completa viene dada por $\boldsymbol{P}(Caries, Dolor_{de_{muelas}}, Tiempo)$. Esta distribución conjunta puede representarse como una tabla de dimensión $2 \times 2 \times 4$ con 16 entradas. Una distribución conjunta completa especifica la probabilidad de cada suceso atómico y es así una especificación completa de la incertidumbre que tiene uno sobre el mundo en cuestión.

Probabilidad condicional o a posteriori

Una vez que el agente obtiene alguna evidencia que afecta a la variable X, las probabilidades a priori ya no son aplicables a X. En su lugar usamos probabilidades a *posteriori o condicionales*. La probabilidad condicional del evento A dada la ocurrencia del evento B se escribe como $P(a|b)$. $P(caries|dolor\_de\_muelas) = 0{,}8$ indica que si se observa a un paciente que tiene un dolor de muelas y todavía no se tiene otra información disponible, entonces la probabilidad de que el paciente tenga una caries será 0,8.

Es la probabilidad de que ocurra un evento ***A*** dado que ocurre otro evento ***B*** (evidencia). No implica causa-efecto. Ejemplo: $P(Test = Positivo|Enfermedad = Presente) = 0{,}7$

Las probabilidades condicionales pueden definirse en términos de probabilidades no condicionales. La ecuación que la define es:

$$P(a|b) = \frac{P(a \wedge b)}{P(b)}$$

Donde $P(a \wedge b)$ es la probabilidad conjunta de a y b, es decir, la probabilidad de que a y b ocurran al mismo tiempo. De la ecuación anterior se obtiene:

$$P(a \wedge b) = P(a|b)P(b)$$

Conocida como la *regla del producto*. Esta última ecuación es más intuitiva, ya que expresa que la probabilidad de que ocurran a y b es igual a la probabilidad de que ocurra a dado b, siempre y cuando ocurra b, o sea, por la probabilidad de b. La probabilidad condicional es una herramienta muy útil para representar información causal de la forma $P(efecto|causa)$.

Inferencia utilizando las distribuciones conjuntas totales

Describiremos un método simple para **inferencia probabilista** (es decir, el cálculo de probabilidades posteriores para proposiciones-pregunta a partir de la evidencia observada). Usaremos la distribución conjunta completa como la «base de conocimiento» desde la que se deducirán las respuestas de todas las preguntas.

El dominio consta exactamente de tres variables booleanas *Dolor-de-muelas*, *Caries* e *Infectase.* La distribución conjunta completa es la siguiente tabla:

	dolor-de-muelas		*¬dolor-de-muelas*	
	infectarse	*¬infectarse*	*infectarse*	*¬infectarse*
caries	0,108	0,012	0,072	0,008
¬caries	0,016	0,064	0,144	0,576

Figura 13.3 Una distribución conjunta completa para el mundo *Dolor-de-muelas*, *Caries* e *Infectarse*

Las probabilidades de la distribución conjunta suman 1, como se requiere por los axiomas de la probabilidad.

Para calcular la probabilidad de cualquier proposición, simple o compleja simplemente identificamos aquellos sucesos atómicos en los que la proposición es cierta y sumamos sus probabilidades. Por ejemplo, hay seis sucesos atómicos en los que se verifica $caries \vee dolor\_de\_muelas$:

$$P(caries \vee dolor_{de_{muelas}}) = 0{,}108 + 0{,}012 + 0{,}072 + 0{,}008 + 0{,}016 + 0{,}064 = 0{,}28$$

Una operación particular muy común es obtener la distribución definida sobre algún subconjunto de variables o una sola variable. Por ejemplo, sumando las entradas de la primera fila se obtiene la **probabilidad marginal** o incondicional de *caries*:

$$P(caries) = 0{,}108 + 0{,}012 + 0{,}072 + 0{,}008 = 0{,}2$$

Estaremos interesados en el cálculo de probabilidades *condicionales* de algunas variables, dada la evidencia sobre otras. Podemos calcular la probabilidad de una caries, dada la evidencia de dolor de muelas, como sigue:

$$P(caries|dolor\_de\_muelas) = \frac{P(caries \wedge dolor\_de\_muelas)}{P(ddolor\_de\_muelas)}$$

$$P(caries|dolor\_de\_muelas) = \frac{0{,}108 + 0{,}012}{0{,}108 + 0{,}012 + 0.016 + 0.064} = 0{,}6$$

Para verificarlo, calculamos también la probabilidad de que no hay caries, dado un dolor de muelas:

$$P(\neg caries|dolor\_de\_muelas) = \frac{P(\neg caries \wedge dolor\_de\_muelas)}{P(dolor\_de\_muelas)}$$

$$P(\neg caries|dolor\_de\_muelas) = \frac{0{,}016 + 0{,}064}{0{,}108 + 0{,}012 + 0.016 + 0.064} = 0{,}4$$

Regla o teorema de Bayes

La regla del producto puede escribirse de dos formas por la conmutatividad de la conjunción:

$$P(a \wedge b) = P(a|b)P(b)$$

$$P(a \wedge b) = P(b|a)P(a)$$

Si igualamos los miembros derechos y dividiendo por P(a), obtenemos:

$$P(b|a) = \frac{P(a|b)\,P(b)}{P(a)}$$

Esta ecuación se conoce como **regla de Bayes** (también como ley de Bayes o teorema de Bayes). Me permite calcular una probabilidad condicional cuando tenemos la probabilidad condicional en sentido contrario.

Por ejemplo, si una alarma se dispara (por la razón que sea) con probabilidad $P(alarma)$, los robos ocurren con probabilidad $P(robo)$ y sabemos que la probabilidad de que nuestra alarma se dispare cuando hay un robo es $P(alarma|robo)$, con la regla de Bayes podemos calcular la probabilidad de que esté ocurriendo un robo cuando suena la alarma, $P(robo|alarma)$.

La regla de Bayes es útil en la práctica porque hay muchos casos donde disponemos de buenas estimaciones probabilistas para estos tres números y necesitamos calcular el cuarto. En una tarea como el diagnóstico médico, con frecuencia tenemos probabilidades condicionales de relaciones causales y necesitamos deducir una diagnóstico. Un médico sabe que la enfermedad de meningitis causa al paciente que el cuello se agarrote un 50 por ciento de las veces. El médico también conoce algunos hechos incondicionados como que la probabilidad *a priori* de que un paciente tenga meningitis es 1/50.000 y la probabilidad priori de que cualquier paciente tenga un cuello agarrotado

es 1/20. Si tomamos *s* como la proposición de que el paciente tiene un cuello agarrotado y *m* como la proposición de que el paciente tiene meningitis, tenemos

$$P(s|m) = 0.5$$
$$P(m) = 1/50000$$
$$P(s) = 1/20$$
$$P\left(\frac{m}{s}\right) = \frac{P(s|m)P(m)}{P(s)} = \frac{0{,}5 \times \frac{1}{50000}}{\frac{1}{20}} = 0{,}0002$$

Es decir, esperamos que sólo un paciente de entre 5.000 (1/P) con un cuello agarrotado tenga meningitis. Nótese que, aun pensando en que una meningitis apunta a un cuello agarrotado con bastante contundencia (con probabilidad 0,5), la probabilidad de meningitis en el paciente se queda pequeña. Esto es porque la probabilidad a priori de un cuello agarrotado es mucho más alta que la de meningitis.

Una cuestión obvia sobre la regla de Bayes es por qué uno debe disponer de la probabilidad condicional en un sentido, pero no en el otro. En el dominio de la meningitis, quizás el médico sabe que un cuello agarrotado implica meningitis en uno de cada 5.000 casos; es decir, el médico tiene información cuantitativa en el sentido de los síntomas a las causas, el sentido **diagnóstico**. Tal médico no tiene necesidad de usar la regla de Bayes. Desafortunadamente, *el conocimiento diagnóstico es con frecuencia más frágil que el conocimiento causal*. Si hay una epidemia repentina de meningitis, la probabilidad incondicional de meningitis, $P(m)$, aumentará. El médico que obtuvo la probabilidad diagnóstica $P(m|s)$ directamente de la observación estadística de pacientes antes de la epidemia no tendrá ni idea de cómo actualizar el valor, pero el médico que calcula $P(m|s)$ a partir de los otros tres valores verá que $P(m|s)$ aumentará proporcionalmente a $P(m)$. Lo que es más importante, la información causal $P(s|m)$ no se ve *afectada* por la epidemia, ya que sencillamente refleja el modo en que actúa la meningitis. El uso de este tipo de conocimiento causal directo o basado en modelos proporciona la robustez decisiva necesaria que hace viables a los sistemas probabilistas en el mundo real.

Independencia

Dos eventos a y b son independientes si se cumple alguna de las siguientes condiciones:

1. $P(a|b) = P(a)\ y\ P(a) \neq 0, P(b) \neq 0.$
2. $P(a) = 0\ o\ P(b) = 0.$

Es decir, si $P(a)$ y $P(b)$ no son nulas, a y b son independientes cuando $P(a|b) = P(a)$. En ese caso, claramente la probabilidad de ocurrencia del evento a no cambia si ocurre o no ocurre b.

Solo si a y b son independientes, partiendo de $P(a|b) = P(a)$ y usando la regla del producto se puede ver que $P(a \wedge b) = P(a)P(b)$. Es importante tener en mente que la dependencia entre dos variables o eventos no implica que uno sea la causa del otro.

Independencia Condicional

Dos eventos a y b son condicionalmente independientes dado c, si $P(c) \neq 0$ y se cumple alguna de las siguientes afirmaciones:

1. $P(a|b \wedge c) = P(a|c)$ y $P(a|c) \neq 0, P(b|c) \neq 0.$

2. $P(a|c) = 0$ o $P(b|c) = 0.$

Un ejemplo para comprender esta propiedad. Supongamos que en un pueblo existen dos vecinos que no tienen ningún tipo de interacción entre ellos. Cada vecino tiene cierta probabilidad de salir de su casa con paraguas cuando hay pronóstico de lluvias. Llamemos p_1 y p_2 a los eventos "salir con paraguas" para el vecino 1 y para el vecino 2 respectivamente. Naturalmente, $P(p_1)$ y $P(p_2)$ son altas en caso de pronóstico positivo. Cuando uno de los vecinos sale con paraguas es más probable que el otro lo haga también, es decir, p_1 y p_2 no son independientes. Esto no

significa que se influyan mutuamente en el mundo real. El comportamiento se debe a que ambos eventos tienen la misma causa. Entonces, para un observador que no conoce el pronóstico, los eventos son dependientes. Si el observador conoce el pronóstico, los eventos se vuelven independientes dado el pronóstico. Es decir, si sabemos que va a llover, la probabilidad de ocurrencia del evento p_1 es condicionalemente independiente de la ocurrencia del evento p_2 y se escribe asi: $P(p_1|lluvia \wedge p_2) = P(p_1|lluvia)$, donde lluvia es el evento que representa al pronóstico positivo de lluvia.

La independencia condicional tiene particular importancia en la utilización de las redes bayesianas porque, como veremos más adelante, permiten representar toda la información necesaria mediante un conjunto reducido de probabilidades condicionales.

Redes Bayesianas

Vimos que la distribución de probabilidad conjunta completa puede responder a cualquier pregunta sobre el dominio, pero:

- Puede llegar a ser enormemente intratable cuando el número de variables crece.
- La especificación de probabilidades para sucesos atómicos es bastante antinatural y puede ser muy difícil.
- Las relaciones de independencia y de independencia condicional entre variables pueden reducir enormemente el número de probabilidades que se necesitan establecer para definir la distribución de probabilidad conjunta.

La **red bayesiana** es una estructura de datos que sirve para representar las dependencias entre las variables y para mostrar una descripción escueta de *cualquier* distribución de probabilidad conjunta completa. Es un grafo dirigido en el que cada *nodo* está comentado con información probabilista cuantitativa. La especificación completa es:

1. Un conjunto de variables aleatorias forman los nodos de la red. Las variables pueden ser discretas o continuas.

2. Un conjunto de enlaces dirigidos o flechas conectan pares de nodos. Si hay una flecha de un nodo X a un nodo Y, se dice que X es un *padre* de Y.

3. Cada nodo X_i tiene una distribución de probabilidad condicionada $P(X_i|Padres(X_i))$ que cuantifica el efecto de los padres del nodo.

4. El grafo no tiene ciclos dirigidos (y así es un grafo acíclico dirigido, o GAD).

La topología de la red especifica las relaciones de independencia condicional que existen en el dominio. El significado intuitivo de un arco que sale de X y apunta a Y es, habitualmente, que X tiene una influencia directa sobre Y. Es generalmente sencillo para un experto del dominio decidir qué influencias directas existen en el área. Una vez que la topología de la red bayesiana está diseñada, necesitamos sólo especificar una distribución de probabilidad condicional para cada variable dados sus padres. La combinación de la topología y las distribuciones condicionales son suficientes para definir la distribución conjunta completa para todas las variables.

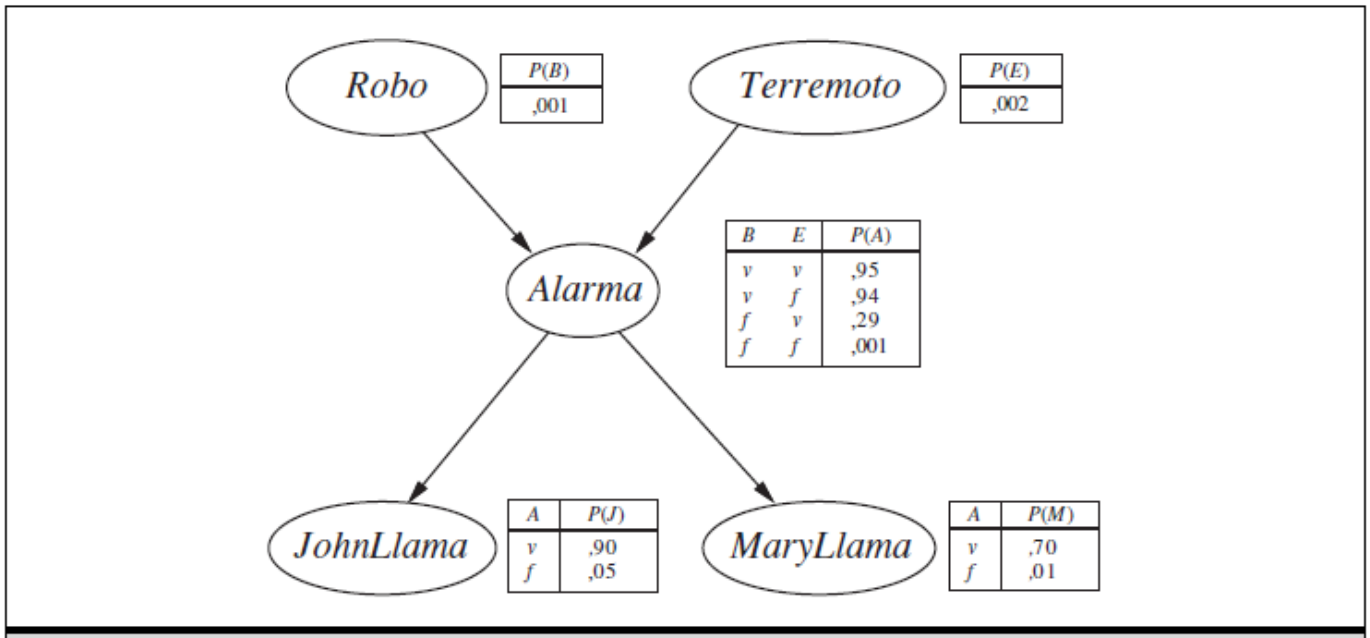


Figura 14.2 Una red bayesiana típica, mostrando tanto la topología de la red como las tablas de probabilidad condicional (TPCs). En las TPCs, las letras B, E, A, J y M significan *Robo*, *Terremoto*, *Alarma*, *JohnLlama* y *MaryLlama*, respectivamente.

En la figura, cada distribución se muestra como una **tabla de probabilidad condicional**, o TPC. Cada fila de una TPC contiene la probabilidad condicional de cada valor del nodo para un **caso de condicionamiento**. Un caso de condicionamiento es una combinación posible de valores de los nodos padres (un suceso atómico en miniatura, si prefiere). Cada fila debe sumar 1, ya que las entradas representan un conjunto exhaustivo de casos para la variable. Para variables booleanas, una vez que conoce que la probabilidad de un valor verdad es *p*, la probabilidad de falso debe ser $1 - p$.

Una red bayesiana proporciona una descripción completa del dominio. Cada entrada de la distribución de probabilidad conjunta puede calcularse a partir de la información de la red. Una entrada genérica en la distribución conjunta es la probabilidad de una conjunción de asignaciones concretas a cada variable, tal como $P(X_1 = x_1 \wedge \ldots \wedge X_n = x_n)$. Para ésta usaremos la notación abreviada $P(x_1 \ldots, x_n)$. El valor de esta entrada está dado por la fórmula:

$$P(x_1 \ldots x_n) = \prod_{i=1}^{n} P(x_i | padres(X_i))$$

donde $padres(X_i)$ denota los valores específicos de las variables de $Padres(X_i)$. Así, cada entrada de la distribución conjunta está representada por el producto de los elementos apropiados de las tablas de las probabilidades condicionales (TPCs) de la red bayesiana. Las TPCs proporcionan una representación descompuesta de la distribución conjunta. Podemos calcular, como ejemplo, la probabilidad de que la alarma ha sonado, pero no ha ocurrido ni un robo ni un terremoto, y tanto John como Mary llaman.

$$P(j \wedge m \wedge a \wedge \neg b \wedge \neg e) = P(j|a) \cdot P(m|a) \cdot P(a|\neg b \wedge \neg e) \cdot P(\neg b) \cdot P(\neg e)$$
$$= 0{,}90 \cdot 0{,}70 \cdot 0{,}001 \cdot 0{,}999 \cdot 0{,}998 = \mathbf{0{,}00062}$$

Inferencia exacta en redes bayesianas

La tarea básica de cualquier sistema de inferencia probabilista es computar la distribución de probabilidad *a posteriori* para un conjunto de **variables pregunta**, dado algún **evento** observado (esto es, alguna asignación de valores para un conjunto de **variables evidencia**). Usaremos la notación:

- X denota a la variable pregunta
- $\boldsymbol{E}$ denota al conjunto de variables evidencias $E_1 \ldots, E_m$ y e es un evento observado particular

- $\boldsymbol{Y}$ denotará las variables no evidencia $Y_1 \ldots, Y_l$ (a veces llamadas **variables ocultas**).

Así, el conjunto completo de variables es $\boldsymbol{X} = \{X\} \cup \boldsymbol{E} \cup \boldsymbol{Y}$.

Una cuestión típica pide la distribución de probabilidad *a posteriori* $\boldsymbol{P}(X|e)$.

En la red del robo, podemos observar el evento en el que $JohnLlama = cierto$ y $MaryLlama = cierto$. Podríamos entonces preguntarnos por la probabilidad de que haya ocurrido un robo:

$$\boldsymbol{P}(Robo|JohnLlama = cierto, MaryLlama = cierto) = \langle 0{,}284, 0{,}716 \rangle$$

Las variables ocultas para esta pregunta son *Terremoto* y *Alarma*.