



UNIVERSIDAD AUTÓNOMA DEL ESTADO DE MORELOS
INSTITUTO DE INVESTIGACIÓN EN CIENCIAS BÁSICAS Y APLICADAS

Implementación de un motor de inferencia Bayesiano

T E S I S

QUE PARA OBTENER EL GRADO DE :
LICENCIADO EN CIENCIAS
ÁREA TERMINAL CIENCIAS COMPUTACIONALES Y
COMPUTACIÓN CIENTÍFICA

PRESENTA:
Roberto Reyes Brito

DIRECTOR DE TESIS:
Dr. Jorge Hermosillo Valadez

SINODALES:
Dr. Markus Franciscuz Müller Bender (Presidente)
Dr. Bruno Lara Guzmán (Secretario)
Dr. Jorge Hermosillo Valadez (Vocal)
Dr. José Daniel Arzate Mena (Suplente)
Dr. Raúl Salgado García (Suplente)

Cuernavaca, Morelos a 12 de agosto de 2024

Índice general

CAPÍTULO

1. Introducción	1
1.1. Antecedentes	2
1.1.1. La probabilidad como una extensión de la lógica	3
1.1.2. Probabilidad en variables	9
1.1.3. Programación Bayesiana	11
1.1.4. Redes de Bayes	14
1.1.5. ¿Qué es un Motor de Inferencia Bayesiano?	16
1.2. Planteamiento del Problema	19
1.3. Objetivo	20
1.4. Objetivos específicos	21
 2. Metodología (capítulo 3)	 23
2.1. Estructuras de datos	23
2.1.1. Clase para variables	24
2.1.2. Clase para distribuciones	24
2.1.3. Clases para el motor de inferencia	26
2.2. Diseño de algoritmos	27
2.3. Diseño experimental	29
2.3.1. Problema 1	29
2.3.2. Problema 2	29
2.3.3. Problema 3	30

Capítulo 1

Introducción

La modelación de fenómenos físicos no solo facilita la comprensión de su funcionamiento, sino que también permite predecir, medir, optimizar, simular, imitar y controlar dichos fenómenos. Con el avance tecnológico actual, es factible convertir modelos complejos en programas computacionales que realizan los cálculos necesarios para obtener respuestas pertinentes. No obstante, todo modelo y programa enfrenta el desafío de la incompletitud; es decir, ningún modelo de un fenómeno real captura todos sus aspectos. La incertidumbre es una consecuencia inevitable de esta incompletitud. Esto se debe a que, al desarrollar un modelo, existen variables ocultas en el entorno que no se consideran, ya sea por limitaciones tecnológicas o por falta de conocimiento. Por lo tanto, aunque los modelos pueden reproducir comportamientos similares a los fenómenos reales, nunca los replican exactamente. Bessiere y Mazer (2013)

En este estudio se busca mejorar la implementación de un motor de inferencia Bayesiano heredado, el cual utiliza la probabilidad como marco matemático para tratar con la incompletitud e la incertidumbre, este marco teórico maneja tanto la información certera como la incierta, y tiene la capacidad de ser un modelo de razonamiento. Este programa se basa en gran medida en la descripción de programas Bayesianos del libro *"Bayesian Programming"* (Bessiere y Mazer, 2013), el cual a su vez se fundamenta en la propuesta de la teoría de la probabilidad como un modelo de razonamiento de Edwar T. Jaynes en su libro *"Probability Theory: The Logic of Science"* (Jaynes, 2003).

El programa incluye una herramienta que facilita la creación de descripciones para modelos probabilísticos y reduce el espacio necesario para

almacenar las probabilidades. Sin embargo, esta herramienta por sí sola no es suficiente para abordar problemas del mundo real. La importancia de este motor de inferencia radica en su amplio alcance de uso. Puede emplearse tanto para describir modelos del entorno y hacer predicciones con un agente artificial, como para realizar inferencias sobre modelos probabilísticos de fenómenos en general. Por ello, es crucial desarrollar un motor de inferencia Bayesiano capaz de manejar problemas más complejos y así aplicar este método de inferencia a problemas del mundo real.

Este trabajo se enfoca en mejorar y extender las capacidades del motores de inferencia Bayesiana, superando las limitaciones actuales y permitiendo su aplicación efectiva en contextos prácticos y reales. Al hacerlo, se pretende avanzar en la precisión y utilidad de los modelos probabilísticos, abordando la incompletitud y la incertidumbre inherentes a la modelación de fenómenos físicos y otros sistemas complejos.

1.1. Antecedentes

Antes de plantear que es un motor de inferencia Bayesiano y cuales son sus problemas, es necesario establecer el por qué funciona la teoría de la probabilidad como modelo de razonamiento y porque es capaz de tratar con la incompletitud.

En el libro "Bayesian Programming" de Bessiere y Mazer (2013), se menciona que los modelos siempre presentan problemas de incompletitud e incertidumbre. En otras palabras, es imposible encontrar todas las causas que afectan a un problema o fenómeno, podríamos creer que hemos logrado encontrar todas las causas, pero no hay manera de probar afirmar que son todas las causas; y también es imposible el recolectar toda la información completa de un problema.

Por ejemplo, en 1781 Sir William Herschel describió el séptimo planeta del sistema solar, Urano. Mientras tanto, tanto Urbain Leverrier, astrónomo francés, como John Adan se interesaron por la trayectoria "incierta" de Urano. El planeta no estaba siguiendo exactamente la trayectoria que predecía la teoría de la gravedad de Newton. Ambos llegaron a la conclusión de que estas irregularidades deberían ser el resultado de una variable oculta no tomada en cuenta por el modelo: la existencia de un octavo planeta. Incluso fueron mucho más allá y encontraron la posición más probable de este octavo planeta. El observador de Berlín recibió la predicción de Leverrier el 23 de septiembre de 1846 y Johann Galle observó a Neptuno ese mismo día. Por lo tanto es necesario contar tanto con un marco matemático y compu-

tacional alternativo para poder abordar la incompletitud e incertidumbre que presentan todos los fenómenos.

La incompletitud es simplemente la brecha irreductible entre el conocimiento preliminar y el fenómeno, y la incertidumbre es una consecuencia directa y mensurable de esta imperfección. (Bessiere y Mazer, 2013)

La lógica es la base matemática del razonamiento y el principio fundamental de la computación, lo cual nos haría pensar que es la mejor herramienta para poder describir fenómenos para poder trabajarlos en una computadora y hacer cálculos de mayor escala. Sin embargo, la lógica no puede utilizarse para modelar fenómenos debido a su limitación a problemas que disponen de información certera y completa.

En el mismo libro, se propone a la teoría de la probabilidad como un marco matemático alternativo. La probabilidad es una exención de la lógica, tan matemáticamente sensata y simple como la lógica, pero con mayor poder expresivo.

Para entender como es que la teoría de la probabilidad proporciona un modelo de razonamiento racional en presencia de incompletitud e incertidumbre, es importante entender como la probabilidad es una exención de la lógica.

1.1.1. La probabilidad como una extensión de la lógica

Siguiendo la descripción discutida en Jaynes (2003), se considera la siguiente situación: un policía patrulla una calle aparentemente vacía durante una noche oscura. De repente, escucha una alarma antirrobo, voltea hacia el otro lado de la calle y observa una joyería con una ventana rota. En ese momento, ve a un hombre enmascarado saliendo por la ventana rota, llevando una bolsa llena de joyas. El policía, sin vacilar, concluye que este hombre es un ladrón. Pero, ¿cómo llega a esta conclusión? Primero, examinemos cuidadosamente la naturaleza de tales problemas.

La conclusión del policía no fue una deducción lógica basada en la evidencia; ya que podría haber otras explicaciones. Por ejemplo, el hombre podría ser el dueño de la joyería, quien, al volver de una fiesta de disfraces, olvidó sus llaves. Al pasar por su tienda, un camión podría haber lanzado una piedra que rompió la ventana, y él simplemente estaba protegiendo su propiedad.

Aunque el razonamiento del policía no fue una deducción lógica, podemos admitir que tenía un cierto grado de validez. Las pruebas no demost-

ban la deshonestidad del hombre con certeza, pero sí la hacían altamente probable. Este es un ejemplo de un tipo de razonamiento que todos hemos desarrollado de manera más o menos competente, necesariamente, mucho antes de estudiar teorías matemáticas. Es casi imposible pasar una hora de vigilia sin enfrentarnos a una situación (por ejemplo, ¿lloverá o no?) en la que no disponemos de suficiente información para hacer un razonamiento deductivo, pero aún así debemos decidir qué hacer de inmediato.

Todas las discusiones sobre estas cuestiones comienzan contrastando el razonamiento deductivo con el razonamiento plausible. El razonamiento deductivo puede analizarse mediante la aplicación repetida de dos silogismos fuertes:

$$\frac{\begin{array}{c} \text{Si } a \text{ es verdadero, entonces } b \text{ es verdadero} \\ a \text{ es verdadero} \end{array}}{\text{Por lo tanto, } b \text{ es verdadero}} \quad (1.1.1)$$

y su inverso:

$$\frac{\begin{array}{c} \text{Si } a \text{ es verdadero, entonces } b \text{ es verdadero} \\ b \text{ es falso} \end{array}}{\text{Por lo tanto, } a \text{ es falso}} \quad (1.1.2)$$

Este es el tipo de razonamiento que quisiéramos emplear siempre; sin embargo, como se ha mencionado, en casi todas las circunstancias que enfrentamos no contamos con la información necesaria para hacerlo. Por ello, recurrimos a silogismos más débiles:

$$\frac{\begin{array}{c} \text{Si } a \text{ es verdadero, entonces } b \text{ es verdadero} \\ b \text{ es verdadero} \end{array}}{\text{Por lo tanto, } a \text{ se vuelve más plausible}} \quad (1.1.3)$$

La evidencia no prueba que a sea verdadera, pero la verificación de una de sus consecuencias sí nos da más confianza en a . Por ejemplo, dado

- $a \equiv$ “Empezará a llover a más tardar a las 10 de la mañana.”
- $b \equiv$ “El cielo se nublará antes de las 10 de la mañana.”

Observar las nubes a las 9:45 a. m. no nos proporciona una certeza lógica de que continuará la lluvia; sin embargo, nuestro sentido común, guiado por un silogismo débil, puede llevarnos a modificar nuestros planes y actuar como si creyéramos que lloverá, especialmente si las nubes son lo suficientemente oscuras.

Otro silogismo débil:

$$\frac{\begin{array}{c} \text{Si } a \text{ es verdadero, entonces } b \text{ es verdadero} \\ a \text{ es falso} \end{array}}{\text{Por lo tanto, } b \text{ se vuelve menos plausible}} \quad (1.1.4)$$

En este caso, la evidencia no demuestra que b sea falsa; sin embargo, se ha eliminado una de las posibles razones que la sostendrían, lo que disminuye nuestra confianza en b . El razonamiento que utiliza un científico para aceptar o rechazar sus teorías se basa casi por completo en silogismos del segundo (1.1.3) y tercer tipo (1.1.4).

Ahora bien, el razonamiento de nuestro policía ni siquiera era del tipo mencionado anteriormente. Se describe mejor mediante un silogismo aún más débil:

$$\frac{\begin{array}{c} \text{Si } a \text{ es verdadero, entonces } b \text{ se vuelve más plausible} \\ b \text{ es verdadero} \end{array}}{\text{Por lo tanto, } a \text{ se vuelve más plausible}} \quad (1.1.5)$$

Aunque este argumento puede parecer débil cuando se presenta de manera abstracta en términos de a y b , reconocemos que la conclusión del policía tiene un gran poder persuasivo. Hay algo que nos lleva a crear que, en ese caso particular, su razonamiento es casi tan fuerte como una deducción lógica.

Estos ejemplos demuestran que el cerebro, al realizar un razonamiento plausible, no solo determina si algo es más o menos plausible, sino que también evalúa el grado de plausibilidad. La probabilidad de que llueva a las 10 a.m. depende en gran medida de lo oscuras que sean las nubes. Además, el cerebro utiliza tanto información previa como datos nuevos específicos del problema; al decidir que hacer, intentamos recordar nuestra experiencia pasada con nubes y lluvia, así como lo que predijo el meteorólogo la noche anterior.

Para demostrar que el policía también estaba haciendo uso de la experiencia pasada de los policías en general, supongamos que hechos como estos ocurrieran varias veces cada noche a todos los policías, y que en todos los casos el hombre resultara ser completamente inocente. Muy pronto, los policías aprenderían a ignorar estos incidentes triviales.

De esta manera, nuestro razonamiento depende en gran medida de la información previa que nos ayuda a evaluar el grado de verosimilitud de un nuevo problema. Este proceso de razonamiento ocurre de manera incons-

ciente, casi instantánea, y ocultamos su complejidad llamándolo sentido común.

Cada vez que podemos construir un modelo matemático que reproduzca una parte del sentido común, prescribiendo un conjunto definido de operaciones, esto nos muestra cómo construir una máquina” (es decir, escribir un programa de computadora) que opere con información incompleta y realice un razonamiento plausible en lugar de uno deductivo.

El primer concepto que usamos es la noción usual de una proposición lógica. Las proposiciones son denotadas por nombres en minúsculas. Se pueden componer proposiciones para obtener nuevas proposiciones utilizando operadores lógicos habituales: $a \wedge b$, denotando la conjunción de la proposición a y b , $a \vee b$ es la disyunción, y $\neg a$, la negación de la proposición a .

Para poder lidiar con la plausibilidad de las proposiciones (en otras palabras, lidiar con la incertidumbre), usamos la teoría de la probabilidad para asignar un valor numérico a la plausibilidad de las proposiciones. Consideremos que, para asignar una probabilidad a una proposición a , es necesario tener al menos algún conocimiento previo, resumido en una proposición π . En consecuencia, la probabilidad de una proposición a está siempre condicionada, al menos por π . Para cada diferente π , $P(\cdot|\pi)$ es una aplicación que asigna a cada proposición a un valor real único $P(a|\pi)$ en el intervalo $[0, 1]$.

Por supuesto, nos interesa razonar sobre las probabilidades de conjunciones, disyunciones y negaciones de proposiciones, denotadas respectivamente, por $P(a \wedge b|\pi)$, $P(a \vee b|\pi)$ y $P(\neg a|\pi)$.

También nos interesa la probabilidad de la proposición a condicionada tanto por el conocimiento preliminar π como por alguna otra proposición b . Esto denota $P(a|b \wedge \pi)$.

Razonamiento probabilístico

El razonamiento probabilístico requiere sólo dos reglas básicas:

1. La *regla de la conjunción*, o también llamada regla del producto, establece la probabilidad de una conjunción de proposiciones a y b de la siguiente manera:

$$\begin{aligned} P(a \wedge b|\pi) &= P(a|\pi) \times P(b|a \wedge \pi) \\ &= P(b|\pi) \times P(a|b \wedge \pi) \end{aligned} \tag{1.1.6}$$

2. La *regla de normalización*, que establece que la suma de las probabilidades de a y $a \neg$ es uno.

$$P(a|\pi) + P(a\neg|\pi) = 1 \quad (1.1.7)$$

en probabilidades discretas, estas dos reglas ((1.1.6) y (1.1.7)) son suficientes para cualquier cálculo. De hecho, podemos derivar todas las demás reglas de inferencia necesarias a partir de estas dos. Por ejemplo, la regla de la disyunción de proposiciones:

$$\begin{aligned} P(a \vee b|\pi) &= 1 - P(\neg a \wedge \neg b|\pi) \\ &= 1 - P(\neg a|\pi) \times P(\neg b|\neg a \wedge \pi) \\ &= 1 - P(\neg a|\pi) \times (1 - P(b|\neg a \wedge \pi)) \\ &= P(a|\pi) + P(\neg a \wedge b|\pi) \\ &= P(a|\pi) + P(b|\pi) \times P(\neg a|b \wedge \pi) \\ &= P(a|\pi) + P(b|\pi) \times (1 - P(a|b|\pi)) \\ &= P(a|\pi) + P(b|\pi) - P(a \wedge b|\pi) \end{aligned} \quad (1.1.8)$$

Es muy importante para adquirir una sensación intuitiva clara de lo que significa una probabilidad condicional y la regla de conjunción. Un primer paso hacia esta comprensión puede ser los silogismos lógicos clásicos en sus formas probabilísticas.

Toda esa lógica consiste en los dos silogismos fuertes (1.1.1), (1.1.2) y todo lo que se sigue de ellos; utilizando ahora el signo de implicación (\Rightarrow) para enunciar la premisa mayor.

$$\frac{a \Rightarrow b \quad a \text{ es verdadero}}{b \text{ es verdadero}} \quad (1.1.9)$$

$$\frac{a \Rightarrow b \quad b \text{ es falso}}{a \text{ es falso}} \quad (1.1.10)$$

y el flujo interminable de sus consecuencias. Si dejamos que π represente su premisa mayor:

$$\pi \equiv "a \Rightarrow b" \quad (1.1.11)$$

entonces estos silogismos corresponden a nuestra regla del producto (1.1.6) en las formas

$$P(b|a \wedge \pi) = \frac{P(a \wedge b|\pi)}{P(a|\pi)}, P(a|\neg b \wedge \pi) = \frac{P(a \wedge \neg b|\pi)}{P(\neg b|\pi)} \quad (1.1.12)$$

respectivamente. Pero de (1.1.9) y (1.1.10) tenemos $p(a \wedge b|\pi) = p(a|\pi)$ y $p(a \wedge \neg b|\pi) = 0$, y entonces (1.1.12) se reduce a

$$P(b|a \wedge \pi) = 1, P(a|\neg b \wedge \pi) = 0 \quad (1.1.13)$$

como se afirma en los silogismos (1.1.9) y (1.1.10).

Pero nuestras reglas tienen también lo que no está contenido en la lógica deductiva: una forma cuantitativa de los silogismos débiles (1.1.3) y (1.1.4). Para mostrar que esos enunciados cualitativos originales siempre se siguen de las reglas actuales, observemos que el primer silogismo débil

$$\begin{array}{c} a \Rightarrow b \\ b \text{ es verdadero} \\ \hline \text{Por lo tanto, } a \text{ se vuelve más plausible} \end{array} \quad (1.1.14)$$

corresponde a la regla del producto (1.1.6) en la forma

$$P(a|b \wedge \pi) = P(a|\pi) \frac{P(b|a \wedge \pi)}{P(b|\pi)}. \quad (1.1.15)$$

Pero de (1.1.9), $P(b|a \wedge \pi) = 1$, y dado que $P(a|\pi) \leq 1$, (1.1.15) da

$$P(a|b \wedge \pi) \geq P(a|\pi) \quad (1.1.16)$$

como se afirma en el silogismo. Asimismo, el silogismo (1.1.4)

$$\begin{array}{c} a \Rightarrow b \\ a \text{ es falso} \\ \hline \text{Por lo tanto, } b \text{ se vuelve menos plausible} \end{array} \quad (1.1.17)$$

corresponde a la regla del producto en la forma

$$P(b|\neg a \wedge \pi) = P(b|\pi) \frac{P(\neg a|b \wedge \pi)}{P(\neg a|\pi)} \quad (1.1.18)$$

Pero de (1.1.16) se sigue que $P(\neg a|b \wedge \pi) \leq P(\neg a|\pi)$; y por lo tanto (1.1.18) da

$$P(b|\neg a \wedge \pi) \leq P(b|\pi) \quad (1.1.19)$$

como se afirma en el silogismo.

Finalmente, el silogismo del policía (1.1.5), el cual también está contenido la regla del producto, enunciada en la forma (1.1.15). Si ahora π representa la premisa mayor, "Si a es verdadera, entonces b se vuelve más plausible", ahora toma la forma

$$P(b|a \wedge \pi) > P(b|\pi) \quad (1.1.20)$$

y (1.1.15) da inmediatamente

$$P(a|b \wedge \pi) > P(a|\pi) \quad (1.1.21)$$

1.1.2. Probabilidad en variables

Las reglas (1.1.6) y (1.1.7) se puede interpretar para distribuciones de probabilidad discretas. Pero para poder hacerlo es necesario establecer el concepto de la noción de *variable discreta*. Estas variables normalmente son denotadas por letras mayúsculas.

Las variables en teoría de la probabilidad son llamadas *variables aleatorias*, cada variable aleatoria es una función que asigna el dominio Ω a algún rango; el conjunto de valores posibles que puede tomar.

Por definición de Bessiere y Mazer (2013), una *variable discreta* X es un conjunto de proposiciones lógicas x_i , de manera que estas proposiciones son mutuamente excluyentes (para todo i, j con $i \neq j$, $x_i \wedge x_j$ es falso) y exhaustivas (al menos una de las proposiciones x_i es verdadero). x_i significa que la variable X toma su i -ésimo valor. $|X|$ denota la cardinalidad de el conjunto X (el número de proposiciones x_i).

La conjunción de dos variables X y Y , denotado por $X \wedge Y$, se define como el conjunto de proposiciones $x_i \wedge y_j$. $X \wedge Y$ es un conjunto de proposiciones mutuamente excluyente y exhaustivas. Como tal, es una nueva variable. Por supuesto, la conjunción de n variables también es una variable y, de esta forma, puede cambiar de nombre en cualquier momento y considerarse como una variable única en la secuela.

Por convención, cada vez que aparece una variable X en una fórmula probabilística $\Phi(X)$, debe entenderse como $\forall x_i \in X, \Phi(x_i)$.

Por ejemplo, dadas tres variables X, Y y Z :

$$P(X \wedge Y|Z) = P(X|Z \wedge \pi) \quad (1.1.22)$$

representa:

$$\begin{aligned} \forall x_i \in X, \forall y_j \in Y, \forall z_k \in Z \\ P(x_i \wedge y_i | z_k) = P(x_i | z_k \wedge \pi) \end{aligned} \quad (1.1.23)$$

Además de las distribuciones en una sola variable, necesitamos una notación para distribuciones en múltiples variables. Para esto, se utilizan comas. Por ejemplo, $P(X, Y)$ representa la probabilidad de todas las combinaciones posibles de los valores de X y Y (en otras palabras, $P(X \wedge Y)$). Esto forma una tabla de 4×2 probabilidades llamada *distribución de probabilidad conjunta* de X y Y . También podemos mezclar variables y valores específicos; $P(x_i, Y)$ sería un vector de dos elementos que muestra las probabilidades de todas las combinaciones de los valores de Y con x_i . Stuart J. Russell (2020)

Una vez definidas las variables y sus respectivas probabilidades, es posible establecer las reglas para el razonamiento probabilístico de variables de la siguiente manera:

1. La regla de la conjunción para variables:

$$\begin{aligned} P(X \wedge Y) &= P(X) \times P(Y|X) \\ &= P(Y) \times P(X|Y) \end{aligned} \quad (1.1.24)$$

De acuerdo con nuestra convención para fórmulas probabilísticas que incluyen variables, esto puede reformular de la siguiente manera:

$$\begin{aligned} \forall x_i \in X, \forall y_i \in Y \\ P(x_i \wedge y_i) &= P(x_i) \times P(y_i | x_i) \\ &= P(y_i) \times P(x_i | y_i) \end{aligned} \quad (1.1.25)$$

lo cual se deduce directamente de la regla de conjunción de para proposiciones (1.1.6).

2. La regla de la normalización para variables:

$$\sum_X P(X) = 1 \quad (1.1.26)$$

La regla de la normalización de deriva de la siguiente manera:

$$\begin{aligned}
1 &= P(x_1) + P(\neg x_i) \\
&= P(x_i) + P(x_2 \vee \cdots \vee x_{|X|}) \\
&= P(x_1) + P(x_2) + \cdots + P(x_{|X|}) \\
&= \sum_{x_i \in X} P(x_i)
\end{aligned} \tag{1.1.27}$$

donde la primera igualdad deriva de la regla de normalización para proposiciones (1.1.7), la segunda de la exhaustividad de las proposiciones x_i , y la tercera de aplicar la ecuación (1.1.8) y la exclusividad mutua de las proposiciones x_i .

3. Otra regla importante es la *regla de la marginalización*:

$$\sum_X P(X \wedge Y) = P(Y) \tag{1.1.28}$$

La regla de la marginalización deriva por la aplicación sucesiva de la regla de conjunción (1.1.24) y la regla de normalización (1.1.26):

$$\begin{aligned}
\sum_X P(X \wedge Y) &= \sum_X P(Y) \times P(X|Y) \\
&= P(Y) \times \sum_X P(X|Y) \\
&= P(Y) \times 1 \\
&= P(Y)
\end{aligned} \tag{1.1.29}$$

1.1.3. Programación Bayesiana

Sin embargo, la teoría de la probabilidad por sí sola no es suficiente para describir de manera intuitiva los fenómenos del mundo real como un modelo probabilístico, ya que no cuenta con una metodología para describir un fenómeno, ni para formular preguntas a dicho modelo con el fin de calcular respuestas. Para abordar esta limitación, se introduce el concepto de programación Bayesiana. Este enfoque no solo proporciona un marco conceptual, sino que también ofrece una metodología para generar descripciones probabilísticas tanto para los fenómenos como para las preguntas planteadas.

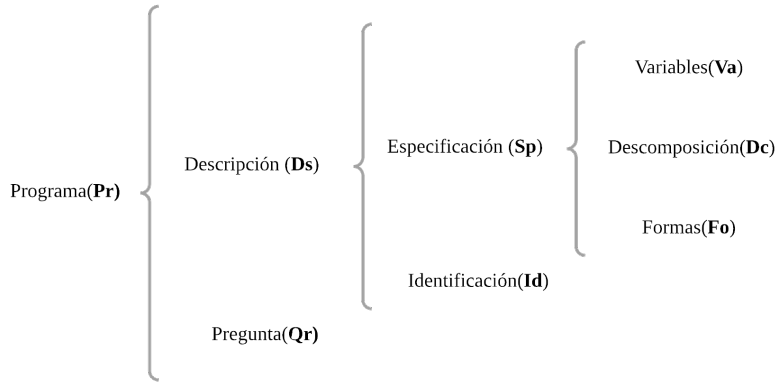


Figura 1.1.1: Estructura de un programa Bayesiano.

Según la definición de Bessiere y Mazer (2013), un programa Bayesiano es un medio para especificar una familia de distribuciones de probabilidad. Un programa Bayesiano constituye principalmente de una descripción y una pregunta:

- **Descripción:** Una descripción se construye utilizando alguna *especificación* dada por el programador y un proceso de *identificación* o aprendizaje para los parámetros no completamente especificados por la especificación, utilizando un conjunto de datos. El propósito de una descripción es especificar un método eficaz para calcular una distribución conjunta en un conjunto de variables $X = \{X_1, X_2, \dots, X_n\}$. Esta distribución conjunta es denotada como: $P(X_1, X_2, \dots, X_n)$.
- **Pregunta:** el propósito de una pregunta es especificar método para calcular probabilidades para proposiciones de consulta dada una descripción (es decir, dado $P(X_1, X_2, \dots, X_n)$), una pregunta se obtiene dividiendo $\{X_1, X_2, \dots, X_n\}$ en : las variables buscadas y las variables conocidas. Por ejemplo: $P(X_i|Known)$, donde $Known \subseteq X \setminus \{X_i\}$.

Especificación

En la descripción se establecen el modelo probabilístico, para dicho propósito se deberá realizar lo siguiente:

1. Definir el conjunto de **variables** relevantes $X = \{X_1, X_2, \dots, X_n\}$ en el que se define la distribución conjunta.
2. **Descomponer** la distribución conjunta: Dada una partición X en k subconjuntos, definimos k variables L_1, \dots, L_k , cada una correspondiente a uno de esos subconjuntos. Cada variable L_i se obtiene como la conjunción de las variables $\{X_{i_1}, X_{i_2}, \dots\}$ pertenecientes al subconjunto i . La regla de la conjunción (1.1.24), conduce a:

$$\begin{aligned}
 P(X_1, X_2, \dots, X_n) \\
 &= P(L_1)P(L_2|L_1) \dots P(L_k|L_{k-1}, \dots, L_2, L_1)
 \end{aligned}
 \tag{1.1.30}$$

L_i se define eligiendo algunas variables X_j entre las variables que aparecen en la conjunción $L_{i-1} \wedge \dots \wedge L_2 \wedge L_1$, llamando R_i a la conjunción de estas variables elegidas y estableciendo:

$$P(L_i|L_{i-1}, \dots, L_2, L_1) = P(L_i|R_i) \tag{1.1.31}$$

Obtenemos entonces:

$$\begin{aligned}
 P(X_1, X_2, \dots, X_n) \\
 &= P(L_1)P(L_2|R_2) \dots P(L_k|R_k)
 \end{aligned}
 \tag{1.1.32}$$

Esta simplificación de la distribución conjunta como producto de distribuciones más simples se denomina *descomposición*. Esto garantiza que cada variable aparezca como máximo una vez a la izquierda de una barra de condicionamiento, que es la condición necesaria y suficiente para escribir descomposiciones matemáticas válidas.

3. Definir las **formas** es la elección de las funciones matemáticas de cada una de estas distribuciones. Donde cada distribución $P(L_i|R_i)$ que aparece en el producto se asocia entonces como una forma paramétrica (es decir, una función $f_\mu((L_i))$ u otro programa Bayesiano. En general, μ es un vector de parámetros que depende de R_i . El aprendizaje tiene lugar cuando algunos de estos parámetros se calculan utilizando el conjunto de datos.

1.1.4. Redes de Bayes

Una vez establecidas las principales reglas para la inferencia en probabilidad, se puede usar la regla del producto, descrita por la ecuación (1.1.6), para descomponer una distribución conjunta, como se muestra en la ecuación (1.1.30). Sin embargo, esta forma de descomposición ya resulta complicada de tratar, y para un modelo probabilístico que tiene el fin de explicar un fenómeno es poco útil, ya que solo describe como descomponer la distribución conjunta.

Este tipo de descomposición requiere de especificar n distribuciones, y para poder aplicar las reglas de inferencia es necesario obtener la información de cada una de las distribuciones, siendo posible que algunas de las distribuciones descritas no tengan sentido alguno. Por esta razón surge la necesidad de encontrar alguna manera factible para poder desarrollar descomposiciones de conjuntos que no solo sean válidas matemáticamente sino también que tengan coherencia. Para poder generar modelos más sencillos se introduce el concepto de redes de Bayes, el cual también describe una metodología para desarrollar modelos probabilísticos coherentes.

Las redes Bayesianas, introducidas por primera vez por Judea Pearl en su libro *Probabilistic Reasoning in Intelligent Systems: Networks of Plausible Inference*, surgió de la unión entre la teoría de la probabilidad y teoría de grafos. Basado en la definición en Stuart J. Russell (2020), las redes de Bayes son un grafo dirigido en el cual cada nodo contiene la probabilidad cuantitativa. Estas redes constituyen una estructura de datos para representar las dependencias entre variables, las cuales pueden representar distribuciones de portabilidad conjuntas, que a su vez describen las relaciones entre las variables presentes en el dominio de un problema. El significado intuitivo que se le da a un arista que va del nodo X al nodo Y es que X tienen una *influencia* sobre Y ; en otras palabras, que el comportamiento de Y es afectado por X , o Y depende de X .

Una vez especificados la topología de la red de Bayes, se especifica la probabilidad condicional local dado por sus padres de cada variable. Dada una red ya construida, la probabilidad conjunta de todas las variables se define por la topología de la red así como su probabilidad se puede calcular a través de las probabilidades locales (probabilidad condicional local). La especificación completa es como sigue:

1. Cada nodo corresponde a una variable aleatoria, que puede ser discreta o continua.
2. Los enlaces dirigidos o flechas conectan pares de nodos. Si hay una flecha desde el nodo X al nodo Y , se dice que X es padre de Y .

El grafo no tiene ciclos dirigidos y, por lo tanto es un grafo acíclico dirigido.

3. Cada nodo X_i tiene información de probabilidad $\theta(X_i|Parents(X_i))$ que cuantifica el efecto de los padres en el nodo utilizando un número finito de parámetros.

Supongamos una red de Bayes que contenga n variables, X_1, \dots, X_n . La distribución conjunta $P(X_1 = x_1 \wedge \dots \wedge X_n = x_n)$, abreviada como $P(x_1, \dots, x_n)$, se describe por la red de Bayes de la siguiente manera:

$$P(x_1, \dots, x_n) = \prod_{i=1}^n \theta(x_i|parents(X_i)), \quad (1.1.33)$$

donde $parents(X_i)$ denota el valor de $Parents(X_i)$ que aparecen en x_1, \dots, x_n . De este modo, cada entra en la distribución condicional es representada por el producto de elementos apropiados de las distribuciones condicionales locales en la red de Bayes.

Resulta que los parámetros $\theta(x_i|parents(X_i))$ son exactamente las probabilidades condiciones $P(x_i|parents(X_i))$ implicadas en la distribución conjunta. Por lo tanto, podemos reescribir la ecuación (1.1.34) como

$$P(x_1, \dots, x_n) = \prod_{i=1}^n P(x_i|parents(X_i)) \quad (1.1.34)$$

Esto implica que, al estimar los valores para las distribuciones locales, deben reflejar las probabilidades condicionales reales de la variable, dadas sus variables parentales. El hecho de que cada parámetro de la red tenga un significado preciso en términos de un conjunto reducido de variables es de suma importancia para la robustez y la facilidad de especificación de los modelos. Basado en la definición, descrita anteriormente, una red de Bayes representa la distribución conjunta de un problema y esto sumamente importante ya que se puede utilizar para responder cualquier consulta, sumando todos los valores de probabilidad conjunta relevantes, cada uno se calcula multiplicando las probabilidades de las distribuciones condicionales locales.

Supongamos que tenemos n variables booleanas y se hace una red de Bayes, las variables pueden estar directamente influenciadas a lo máximo por otras k variables, para alguna $k \in \mathbb{Z}^+$. Entonces la cantidad de información necesaria para especificar cada tabla de probabilidad condicional será como máximo 2^k , y la red completa puede ser especificada por $2^k \times n$. Por otro

lado, si se especifica la distribución conjunta, sin una red de Bayes, entonces la cantidad de información para especificar la tabla sería de 2^n , incluso en problemas pequeños es notorio que la información necesaria para representar una distribución conjunta es demasiado costoso. Fijemos el valor de n a 30, por lo tanto tendríamos 30 variables booleanas, y el valor de k a 5, entonces la red de Bayes requiere de $2^5 \times 30 = 960$ registros y la distribución conjunta requiere 2^{30} y esto es más de un billón de registros.

La especificación de las tablas de probabilidad condicional para una red completa conectada, en el cual cada variable tiene todos los predecesores como padres, requiere la misma cantidad de información como es especificar la distribución conjunta en forma de tabla. Por esta razón, A menudo omitimos enlaces aunque exista una ligera dependencia, porque la ligera ganancia en precisión no compensa la complejidad adicional en la red.

Aunque las redes de Bayes son una herramienta muy útil para la descripción de descomposiciones y la reducción de la información necesaria para describirlas, presentan el sutil problema de que la interacción entre nodos puede volverse subjetiva. Sin embargo, a pesar de este inconveniente, la descomposición resultante adquiere coherencia, lo que permite que las tablas de probabilidad necesarias se conviertan en distribuciones que reflejan un comportamiento coherente de un fenómeno.

1.1.5. ¿Qué es un Motor de Inferencia Bayesiano?

Definamos un motor de inferencia Bayesiano como un sistema destinado a obtener respuestas a preguntas formuladas dentro de un modelo probabilístico, utilizando métodos de inferencia Bayesiana. En este contexto, se implementa el concepto de programación Bayesiana para la descripción de modelos y la generación de preguntas, y se utilizan las redes de Bayes para la descomposición de la distribución conjunta, es decir que la distribución conjunta se descompone de la siguiente manera.

Dada el conjunto de variables relevantes $X = \{X_1, X_2, \dots, X_n\}$ y una vez especificación de la topología de una red de Bayes, por la ecuación(1.1.34) se tiene la siguiente descomposición:

$$P(X_1, X_2, \dots, X_n) = \prod_{i=1}^n P(X_i | \text{parents}(X_i)) \quad (1.1.35)$$

A continuación, se describen los métodos empleados para realizar inferencia sobre distribuciones marginales y condicionales.

Inferencia Bayesiana exacta

Antes de presentar las reglas de inferencia probabilística es necesario establecer que es la inferencia. Según Jaynes (2003), la inferencia se refiere a dos tipos de razonamiento: deductivo, cuando siempre se dispone de suficiente información para permitirlo, e inductivo o probabilístico, cuando no se cuenta con toda la información necesaria, lo cual es común en la mayoría de los problemas reales. La inferencia Bayesiana se basa en el razonamiento probabilístico.

Las siguientes reglas detallan un método sencillo para la inferencia probabilística, el cual se basa en el cálculo de probabilidades para proposiciones de consulta. Para este propósito, se utiliza una distribución de probabilidad conjunta como la base de conocimientos, del cual se puede derivar respuestas a todas las proposiciones de consulta.

Dada la descomposición (1.1.35), se pueden obtener los siguientes ecuaciones para calcular dos distribuciones de con las cuales se pueden derivar todas las proposiciones de consulta de manera general.

1. **Marginal:** Por la ecuación (1.1.28) y (1.1.35) derivamos:

$$\begin{aligned} P(X_i) &= \sum_{X \setminus \{X_i\}} P(X_1, \dots, X_n) \\ &= \sum_{X \setminus \{X_i\}} \left(\prod_{j=1}^n P(x_j | \text{parents}(X_j)) \right) \end{aligned} \quad (1.1.36)$$

2. **Distribuciones condicionales:** Sea $Y, Z \subseteq X$ tal que $Y \cap Z = \emptyset$. Despejando la ecuación (1.1.24) se obtiene lo siguiente.

$$\begin{aligned} P(Y|Z) &= \frac{P(Y, Z)}{P(Z)} \\ &= \frac{\sum_{X \setminus Y \cup Z} P(X_1, \dots, X_n)}{\sum_{X \setminus Y} P(X_1, \dots, X_n)} \\ &= \frac{\sum_{X \setminus Y \cup Z} \prod_{i=1}^n P(X_i | \text{parents}(X_i))}{\sum_{X \setminus Y} \prod_{i=1}^n P(X_i | \text{parents}(X_i))} \end{aligned} \quad (1.1.37)$$

en la segunda igualdad se aplica la ecuación (1.1.28) y después se sustituyen las probabilidades conjuntas por (1.1.35).

Con las ecuaciones (1.1.36) y (1.1.37) se describen los dos siguientes algoritmos para realizar el calculo distribuciones de consulta.

Pero antes es necesario describir algunos datos auxiliares para facilitar la descripción de los algoritmos. Dado $X = \{X_1, X_2, \dots, X_n\}$ un conjunto de variables de un modelo probabilístico, sea $V = X$, k el subíndice de una variable del conjunto X y $e \in X_k$ el valor del evento a calcular, se describe el algoritmo (1), el cual realiza la inferencia sobre una distribución marginal.

Algoritmo 1 Algoritmo para el cálculo de marginales

```

procedure MARGINAL( $V = \{V_1, \dots, V_n\}, k, e$ ) ▷  $V = X$ 
   $V_k = \{e\}$ 
   $PV = V_1 \times V_2 \times \dots \times V_n$ 
   $p = 1$ 
  for  $i = 1, 2, \dots, |PV|$  do
     $p = P(X_1 = PV_{i_1}, \dots, X_n = PV_{i_n}) \cdot p$ 
  end for
  return  $p$ 
end procedure

```

Sea $Y, Z \subseteq X$ tal que $Y \neq Z$ y $V = X$. Definamos dos listas auxiliares H y O , donde $\forall H_i \in H, H_i \in Y_i$ y $\forall O_i \in O, O_i \in Z_i$, se describe el algoritmo (2), el cual realiza la inferencia sobre una distribución condicional.

Algoritmo 2 Algoritmo para el cálculo de condicionales

```

procedure CONDICIONAL( $Y, Z, X, H, O$ )
   $N = X \setminus \{Y \cup Z\}$ 
   $D = X \setminus Y$ 
   $NP = H \times O \times N_1 \times \dots \times N_k$ 
   $np = 1$ 
  for  $i = 1, 2, \dots, |NP|$  do
     $np = P(X_1 = NP_{i_1}, X_2 = NP_{i_2}, \dots, X_n = NP_{i_n}) \cdot p$ 
  end for
   $DP = H \times Z \times D_1 \times D_2 \times \dots \times D_p$ 
   $dp = 1$ 
  for  $i = 1, 2, \dots, |DP|$  do
     $np = P(X_1 = DP_{i_1}, X_2 = DP_{i_2}, \dots, X_n = DP_{i_n}) \cdot p$ 
  end for
  return  $\frac{np}{dp}$ 
end procedure

```

1.2. Planteamiento del Problema

Para realizar inferencia Bayesiana exacta, el motor de inferencia enfrenta un problema significativo relacionado con la memoria requerida. En la propuesta actual del motor de inferencia heredado (MIB1.0), el basada en los antecedentes descritos, para la inferencia exacta se genera una representación de todas las combinaciones posibles de eventos para las variables especificadas.

El espacio para almacenar el producto cartesiano de las variables para una marginal, del algoritmo (1) se puede representar con la siguiente ecuación:

$$E_m(V) = \prod_{V_i \in V} |V_i| \quad (1.2.1)$$

De la ecuación anterior podemos calcular la complejidad espacial y temporal de la siguiente manera.

Sea $c = \max\{|V_1|, |V_2|, \dots, |V_n|\}$, entonces

$$\prod_{V_i \in V} |V_i| \leq c^n \quad (1.2.2)$$

por lo tanto,

$$O(E_m(V)) = c^n \quad (1.2.3)$$

Como el algoritmo (1) necesita calcular la probabilidad conjunta sobre todas las combinaciones posibles de las variables, con el valor de una variable fijada, se puede concluir que la complejidad espacial es la misma que la complejidad temporal. Sea T_m la función que describe el tiempo del algoritmo (1) sobre la entrada V , detiene lo siguiente.

$$T_m(V) = E_m(V) \leq c^n \quad (1.2.4)$$

$$\therefore O(T_m(V)) = c^n \quad (1.2.5)$$

De manera similar, la complejidad espacial del algoritmo (2) se puede representar con la siguiente ecuación:

$$E_c(Y, Z, X) = \prod_{X_i \in X \setminus \{Y \cup Z\}} |X_i| + \prod_{X_i \in X \setminus Y} |X_i| \quad (1.2.6)$$

Sea $c = \max\{|X_1|, |X_2|, \dots, |X_n|\}$ y $k = \max\{n - |X \setminus Y|, n - |X \setminus \{Y \cup Z\}|\}$, entonces

$$\prod_{X_i \in X \setminus \{Y \cup Z\}} |X_i| + \prod_{X_i \in X \setminus Y} |X_i| \leq c^k + c^k \leq 2c^k \quad (1.2.7)$$

$$\therefore O(E_c(Y, Z, X)) = 2c^k \quad (1.2.8)$$

Como el algoritmo (2) necesita calcular la probabilidad conjunta sobre todas las combinaciones del conjunto NP y sobre todas las combinaciones del conjunto DP , esto quiere decir que la complejidad temporal se puede expresar con la siguiente ecuación:

$$T_c(Y, Z, X) = E_c(Y, Z, X) \leq 2c^k \quad (1.2.9)$$

$$\therefore O(T_c(Y, Z, X)) = 2c^k \quad (1.2.10)$$

Por lo tanto, el primer problema que surge es que la memoria requerida para la inferencia crece de manera exponencial. Incluso en problemas con variables exclusivamente booleanas, la memoria necesaria resulta demasiado costosa.

Otro de los principales problemas es que el motor de inferencia heredado requiere una forma para poder determinar qué valor de las variables le corresponde a las variables de la descomposición, ya que la forma de guardar las combinaciones es a través de una tupla con solo los valores, lo que hace que la implementación sea difícil de describir. Además, existe una falta de descripciones para variables aleatorias continuas, lo cual limita el uso del motor de inferencia para problemas reales.

1.3. Objetivo

El objetivo general de esta tesis es mejorar la eficiencia de una implementación del motor de inferencia Bayesiano (MIB2.0). Se pretende desarrollar una implementación con una estructura sólida y fácil de manejar.

Además, esta investigación busca proporcionar visibilidad sobre el uso de este motor de inferencia y presentar los principales desafíos asociados con la implementación de un motor de inferencia Bayesiano. De este modo, se espera que la comunidad científica pueda, en el futuro, llevar esta herramienta a un nivel superior y aplicarla a problemas de mayor dimensión.

1.4. Objetivos específicos

- Diseñar mejoras para la implementación de la estructura de datos con el fin de disminuir la cantidad de memoria necesaria para describir el modelo probabilístico de un fenómeno.
- Analizar la complejidad temporal y espacial de los algoritmos.
- Desarrollo e implementación de algoritmos para la inferencia.
- Validar la correctitud de la implementación del motor de inferencia Bayesiano.

Capítulo 2

Metodología (capítulo 3)

2.1. Estructuras de datos

Se propone utilizar el paradigma de programación orientada a objetos para la descripción de las estructuras de datos, aprovechando que este enfoque permite especificar atributos y métodos para una clase. Este paradigma ofrece una solución al problema de determinar qué valor de las variables corresponde a las variables de la descomposición. Especificando atributos para las variables que almacenen los valores posibles de cada variable y un método para establecer un evento y modificar los valores posibles para que solo considere este evento, sin perder la información de los demás eventos posibles.

Dado que en la propuesta las variables serán objetos, se propone la siguiente metodología para trabajar con el problema de determinar qué valor de las variables corresponde a las variables de la descomposición. La generación de combinaciones se realizará de manera que se cree una lista de valores, donde dada la tupla $(V_{i_1}, V_{i_2}, \dots, V_{i_n})$, cada valor V_{i_j} corresponderá a un evento a establecer para la variable X_j .

De forma que a las distribuciones se les para sus variables especificados como objetos, de forma que las distribuciones solo deberán consultar el atributo que contenga el valor del evento establecido. De este modo, a las distribuciones se les asignarán sus variables especificadas como objetos, de manera que las distribuciones solo necesitarán consultar el atributo que contenga el valor del evento establecido. Esta metodología simplifica el proceso de inferencia, facilitando el manejo y la manipulación de las variables y sus correspondientes valores dentro del motor de inferencia Bayesiano. Se

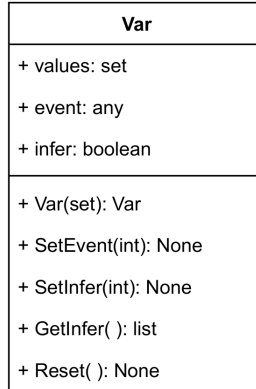


Figura 2.1.1: Diagrama de clase para las variables.

propone utilizar *pyhton* para la implementación de las clases, aprovechando la versatilidad que este lenguaje tiene en la especificación de tipos de datos en los parámetros. Lo cual nos permite establecer cualquier tipo de dato para la representación de valores para las variables. A continuación se especifica los diagramas de clases.

2.1.1. Clase para variables

Para la clase de las variables se propone el diagrama que se muestra en la imagen 2.1.2.

Se describen los atributos de la siguiente manera:

- *values*: Conjunto de valores posibles para las variables.
- *event*: Valor del evento, tiene que pertenecer a *values*.
- *infer*: Boleano para identificar que se realizara inferencia sobre esta variable.

2.1.2. Clase para distribuciones

Distribución marginal

Para la clase de las distribuciones marginales se propone el diagrama que se muestra en la imagen 2.1.2.

Se describen los atributos de la siguiente manera:

Distrib
+ var: Var
+ table: dict
+ Distrib(Var, dict): Distrib
- GetP(): float

Figura 2.1.2: Diagrama de clase para las distribuciones marginales.

CondDistrib
+ var: Var
+ indep: set
+ table: dict
+ CondDistrib(Var, dict): CondDistrib
- GetP(): float

Figura 2.1.3: Diagrama de clase para las distribuciones condicionales.

- *var*: Objeto de tipo Var, guarda la variable de la distribución.
- *table*: Diccionario con la de probabilidad da cada valor de la variable.

Distribución condicional

Para la clase de las distribuciones condicionales se propone el diagrama que se muestra en la imagen 2.1.3.

Se describen los atributos de la siguiente manera:

- *var*: Objeto de tipo Var, guarda la variable de la distribución.
- *indep*: Conjunto de objetos de de tipo Var, guarda las variables independientes en la distribución condicional.
- *table*: Diccionario con la de probabilidad da cada valor de la variables.

Distribución conjunta

Para la clase de las distribuciones condicionales se propone el diagrama que se muestra en la imagen 2.1.4.

Se describen los atributos de la siguiente manera:

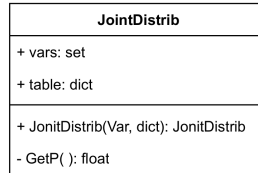


Figura 2.1.4: Diagrama de clase para las distribuciones conjuntas.

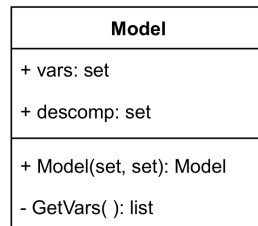


Figura 2.1.5: Diagrama de clase para los modelos probabilísticos.

- *vars*: Conjunto de objetos de de tipo Var, guarda las variables de la distribución.
- *table*: Diccionario con la de probabilidad da cada valor de la variables.

2.1.3. Clases para el motor de inferencia

Para la clase del motor de inferencia se proponen dos diagramas 2.1.5 y 2.1.6.

El diagrama de la imagen 2.1.5 representa la cales para especificar las variables y descomposición de un modelo probabilístico, además se propone un método para recuperar los valores posibles de las variables para una inferencia.

Se describen los atributos de la siguiente manera:

- *vars*: Conjunto de objetos de de tipo Var, guarda las variables de la distribución conjunta.
- *descomp*: Conjunto de objetos de tipo Distrib y CondDistrib para guardad la descomposición de la distribución conjunta.

El diagrama de la imagen 2.1.6 representa la cales para crear un motor de inferencia el cual tiene la tarea de realiza la inferencia.

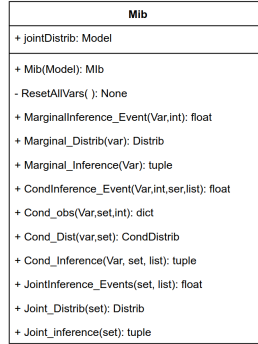


Figura 2.1.6: Diagrama de clase para el motor de inferencia.

Se describen los atributos de la siguiente manera:

- *jointDistrib*: Objeto de tipo Model.

2.2. Diseño de algoritmos

Para la inferencia exacta se proponen dos algoritmos: uno para el cálculo de las marginales y otro para las distribuciones condicionales.

Dada la complejidad temporal de la regla de marginales, descrita en la ecuación (1.2.3), se observa que la memoria requerida crece de manera exponencial. Para abordar este problema, se propone utilizar la función *'product'* de la librería *'itertools'*, que emplea el concepto de generadores en Python. Esto permite consultar uno a uno los elementos del producto cartesiano de los iterables pasados como parámetro, evitando así la necesidad de almacenar todas las combinaciones en memoria.

A continuación, se presentan los algoritmos para la inferencia exacta.

A partir de los algoritmos (3) y (4), se observa que, dado que ya no es necesario generar las combinaciones de las variables, la complejidad temporal se vuelve lineal. En el algoritmo (3), esta complejidad depende únicamente de los argumentos, mientras que en el algoritmo (4) depende tanto de los argumentos como de los subconjuntos de N y D . No obstante, la complejidad temporal sigue siendo la misma para ambos casos.

Algoritmo 3 Algoritmo para el cálculo de marginales con 'product'

```

procedure MARGINAL( $X, V = \{V_1, \dots, V_n\}, k, e$ )  $\triangleright V = X$ 
   $V_k = \{e\}$ 
   $p = 1$ 
  for  $PV$  in  $product([V_1, V_2, \dots, V_n])$  do
     $p = P(X_1 = PV_1, \dots, X_n = PV_n) \cdot p$ 
  end for
  return  $p$ 
end procedure

```

Algoritmo 4 Algoritmo para el cálculo de condicionales con 'product'

```

procedure CONDICIONAL( $Y, Z, X, H, O$ )
   $N = X \setminus \{Y \cup Z\}$ 
   $D = X \setminus Y$ 
   $np = 1$ 
  for  $NP$  in  $product([H, O, N_1, \dots, N_k])$  do
     $np = P(X_1 = NP_1, X_2 = NP_2, \dots, X_n = NP_n) \cdot p$ 
  end for
   $dp = 1$ 
  for  $DP$  in  $product([H, Z, D_1, \dots, D_n])$  do
     $np = P(X_1 = DP_1, X_2 = DP_2, \dots, X_n = DP_2) \cdot p$ 
  end for
  return  $\frac{np}{dp}$ 
end procedure

```

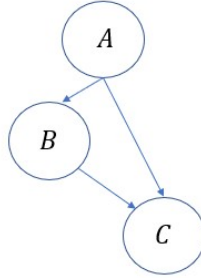


Figura 2.3.1: Red de Bayes del modelo probabilístico $P(ABC)$.

2.3. Diseño experimental

Para alcanzar el objetivo principal de esta tesis, se han diseñado tres problemas experimentales que permiten evaluar distintos aspectos del modelo probabilístico y su implementación. A continuación, se describen los tres problemas propuestos.

2.3.1. Problema 1

El primer problema se enfoca en analizar cómo varía el tiempo de ejecución a medida que aumenta la cardinalidad de las variables. Este análisis es esencial para comprender el comportamiento del modelo probabilístico en situaciones de escalabilidad.

Se emplearán tres variables: A , B y C , siguiendo el modelo probabilístico $P(ABC) = P(A)P(B|A)P(C|AB)$. Se realizarán 100 pruebas, donde la cardinalidad de las tres variables comenzará en 1 y en cada prueba posterior la cardinalidad aumentará en uno. Las distribuciones $P(A)$, $P(B|A)$ y $P(C|AB)$ serán distribuciones uniformes. Con esta configuración, se efectuará la inferencia sobre la distribución completa $P(B)$ y la inferencia del valor más alto de A en $P(A|B = 0, C = 0)$.

2.3.2. Problema 2

El segundo problema tiene como objetivo analizar el desempeño al realizar la inferencia sobre toda una distribución condicional.

Se tiene un conjunto de correos, en los cuales se sabe si son spam o no, y se quiere saber si dada las palabras presentes en el correo el correo pertenece

a spam.

Las variables están descritas de la siguiente manera:

- $Spam = \{0, 1\}$,
- $W_i = \{0, 1\}$

y la descomposición del modelo probabilístico:

$$P(Spam, W_0, W_1, \dots, W_{n-1}) = P(Spam) \prod_{i=0}^{n-1} P(W_i | Spam) \quad (2.3.1)$$

Cada una de las n formas de $P(W_i | Spam)$ debe ser especificado. Para el conteo de la i -ésima palabra que aparezca spam y no spam. $P(W_i | Spam)$:

$$P(W_i = 1 | Spam = 0) = \frac{1 + n_f^i}{|W_i| + n_f}, \quad (2.3.2)$$

$$P(W_i = 0 | Spam = 0) = 1 - P(W_i = 1 | Spam = 0), \quad (2.3.3)$$

$$P(W_i = 1 | Spam = 1) = \frac{1 + n_v^i}{|W_i| + n_v}, \quad (2.3.4)$$

$$P(W_i = 0 | Spam = 1) = 1 - P(W_i = 1 | Spam = 1) \quad (2.3.5)$$

donde, n_f^i es el número de apariciones de la i -ésima palabra en correos que no son spam y n_f es el número total de correos que no son spam, n_v^i es el número de apariciones de la i -ésima palabra en correos que son spam y n_v es el número total de correos que son spam. Una vez descrito el problema se plantea inferir sobre todos los valores de la distribución $P(W_0, W_1, \dots, W_{n-1} | Spam)$.

2.3.3. Problema 3

Para el tercer problema, se busca evaluar la exactitud y el desempeño del proceso de inferencia en diversas consultas utilizando un modelo probabilístico de contratación de una compañía hipotético. Se analizará tanto la precisión de los resultados obtenidos como el rendimiento del proceso de inferencia en diferentes escenarios planteados. La compañía ha recolectado datos históricos de los últimos 5 años sobre la evaluación de 500 candidatos y ha determinado varias proporciones y tasas relacionadas con las calificaciones, experiencia laboral y resultados de las entrevistas.

Datos Históricos

- Proporción de candidatos con calificaciones regulares: 30 %
- Proporción de candidatos con experiencia laboral: 60 %
- Probabilidad de obtener la más alta apreciación en la entrevista:
 - Calificaciones sobresalientes y experiencia laboral: 80 %
 - Calificaciones regulares y experiencia laboral: 30 %
 - Calificaciones sobresalientes y sin experiencia: 30 %
 - Calificaciones regulares y sin experiencia: 10 %
- Probabilidad de obtener la peor apreciación en la entrevista:
 - Calificaciones sobresalientes y experiencia laboral: 2 %
 - Calificaciones regulares y experiencia laboral: 10 %
 - Calificaciones sobresalientes y sin experiencia: 30 %
 - Calificaciones regulares y sin experiencia: 70 %
- Tasa de rechazo:
 - Apreciación favorable: 10 %
 - Apreciación regular: 60 %
 - Apreciación desfavorable: 99 %

Modelación

Variables:

- $C := \{0, 1\}$, donde es 0 si son calificaciones regulares, 1 otro caso.
- $E := \{0, 1\}$, donde es 0 si cuentan con experiencia y 1 si no cuentan con experiencia.
- $N := \{0, 1, 2\}$, donde es 0 si apreciación desfavorable, 1 si la apreciación regular y 2 si apreciación favorable.
- $O := \{0, 1\}$, donde es 0 si no es contratado y 1 si sí.

El modelo de contratación se representa mediante la siguiente distribución conjunta:

$$P(CENO) = P(C)P(E)P(N|CE)P(N|O) \quad (2.3.6)$$

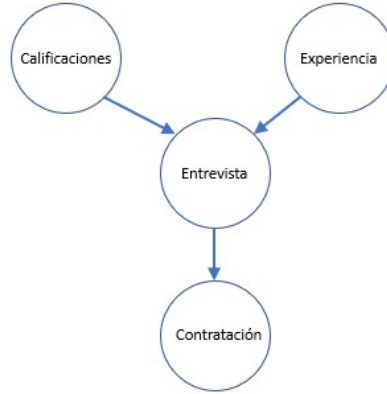


Figura 2.3.2: Red de Bayes del modelo probabilístico $P(CENO)$.

Preguntas

Para evaluar la precisión y el rendimiento del proceso de inferencia, se plantearán las siguientes preguntas:

1. ¿Cuál es su tasa de contratación? ($P(O)$)
2. ¿Cuántos candidatos ha contratado la compañía en los últimos 5 años? ($P(O = 0)$)
3. ¿Cuál es la tasa de apreciaciones favorables en la compañía? ($P(N = 0)$)
4. ¿Cuál es la tasa de apreciaciones regulares dado que se contrata a alguien? ($P(N = 1|O = 1)$)
5. ¿Cuál es la tasa de apreciaciones regulares dado que se contrata a alguien? ($P(N = 0|O = 0)$)
6. ¿Cuántos de los candidatos contratados obtuvieron una apreciación regular? ($P(N = 1|O = 0)$)
7. ¿Cómo se distribuye la contratación de candidatos en función de sus calificaciones? ($P(O|C)$)
8. ¿Cómo se distribuye la contratación de candidatos en función de su experiencia? ($P(O|E)$)

-
9. ¿Cuál es la probabilidad de que alguien con experiencia laboral tenga calificaciones sobresalientes? ($P(C = 0|E = 0)$)
 10. ¿Cuál es la distribución conjunta del personal contratado y la apreciación de la entrevista? ($P(ON)$)

Bibliografía

- P. Bessiere y E. Mazer. *Bayesian Programming*. Chapman and Hall/CRC, 2013.
- E. T. Jaynes. *Probability Theory: The Logic of Science*. Cambridge University Press, 2003.
- P. N. Stuart J. Russell. *Artificial Intelligence A Modern Approach*. Pearson, 2020.