PROGRAMACIÓN BAYESIANA DE AGENTES ARTIFICIALES

Metodología de programación F+D

Fundamentos epistemológicos

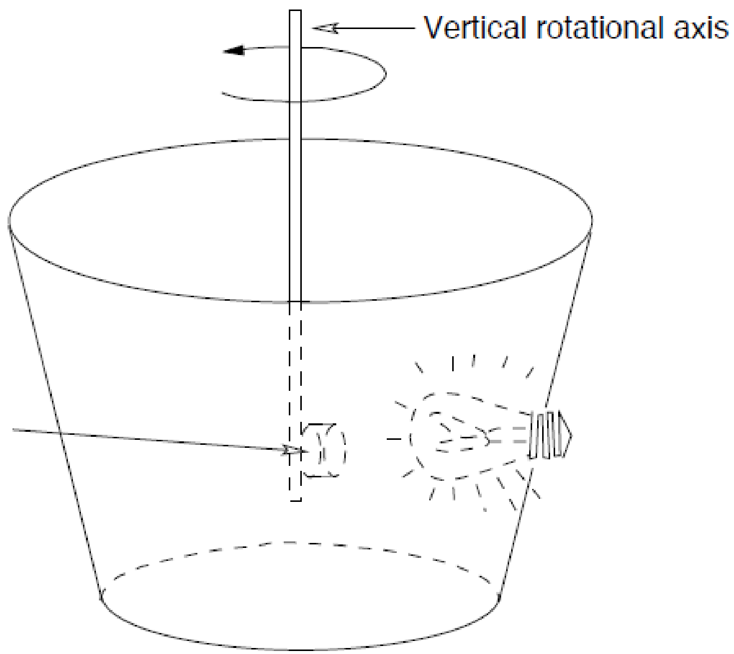
Donde se explica cómo la intrínseca incompletud de los modelos conduce a la incertidumbre y cómo la Programación Bayesiana permite enfrentar este problema en el contexto de la IA (fundamentos que dan sustento a la lógica bayesiana como método de construcción de agentes artificiales inteligentes).

Una causa muy pequeña que se nos escapa determina un efecto considerable que no podemos dejar de ver, y entonces decimos que el efecto se debe al azar.

H. Poincare

Haz en la Cubeta: noción de incompletud

Sistema sensorio-motriz

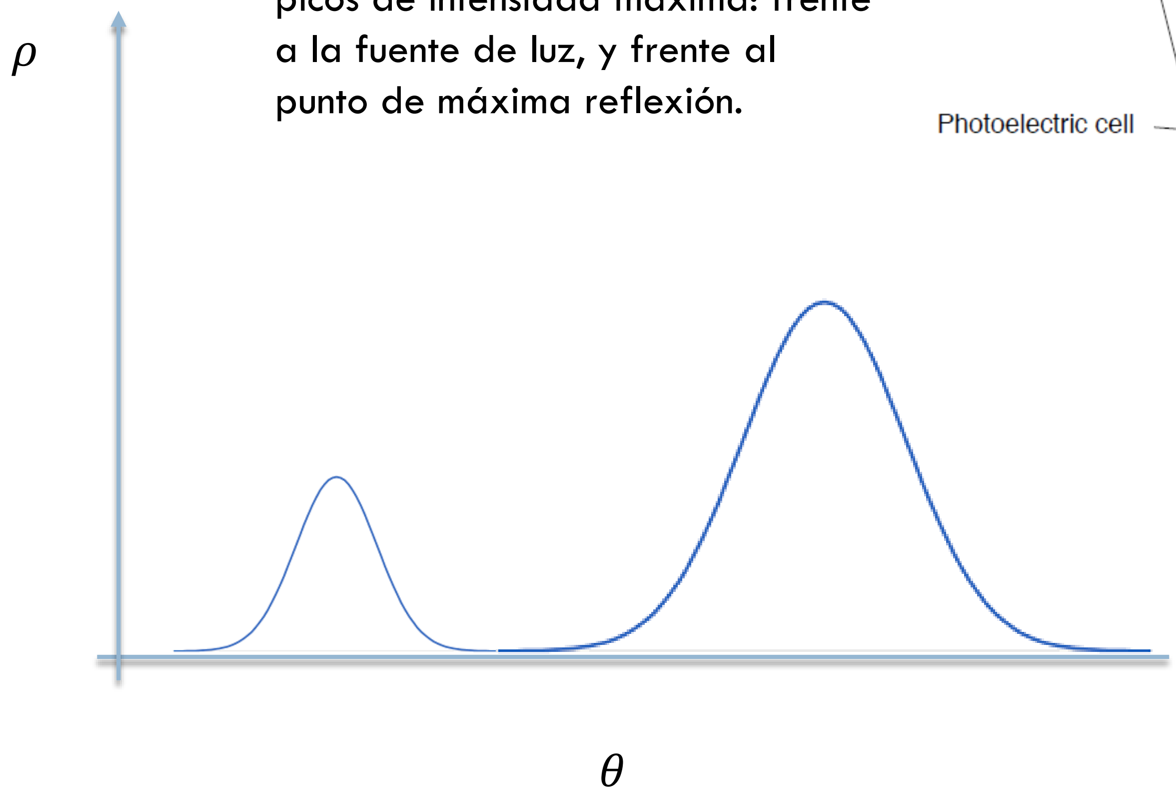


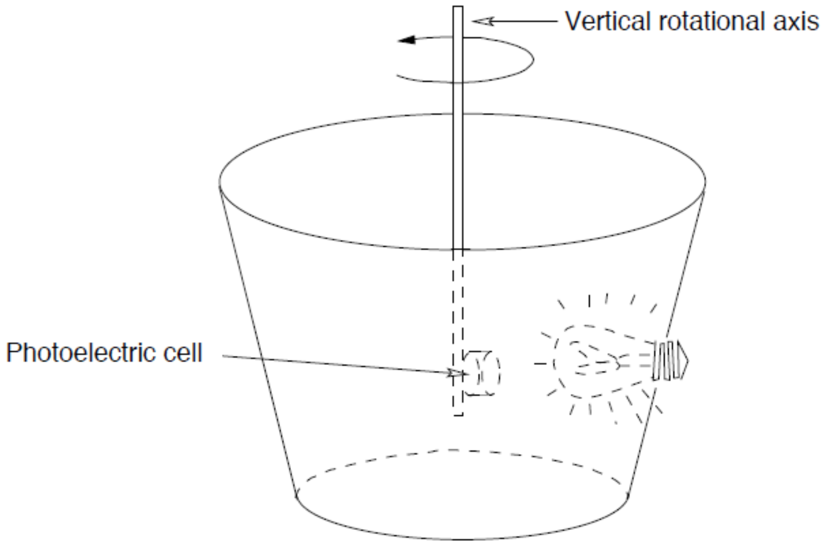
El montaje experimental "haz en la cubeta". Una celda fotoeléctrica explora 72 posiciones diferentes alrededor de un eje vertical midiendo el haz reflejado de una lámpara. Tanto la celda como la luz se colocan dentro de una cubeta.

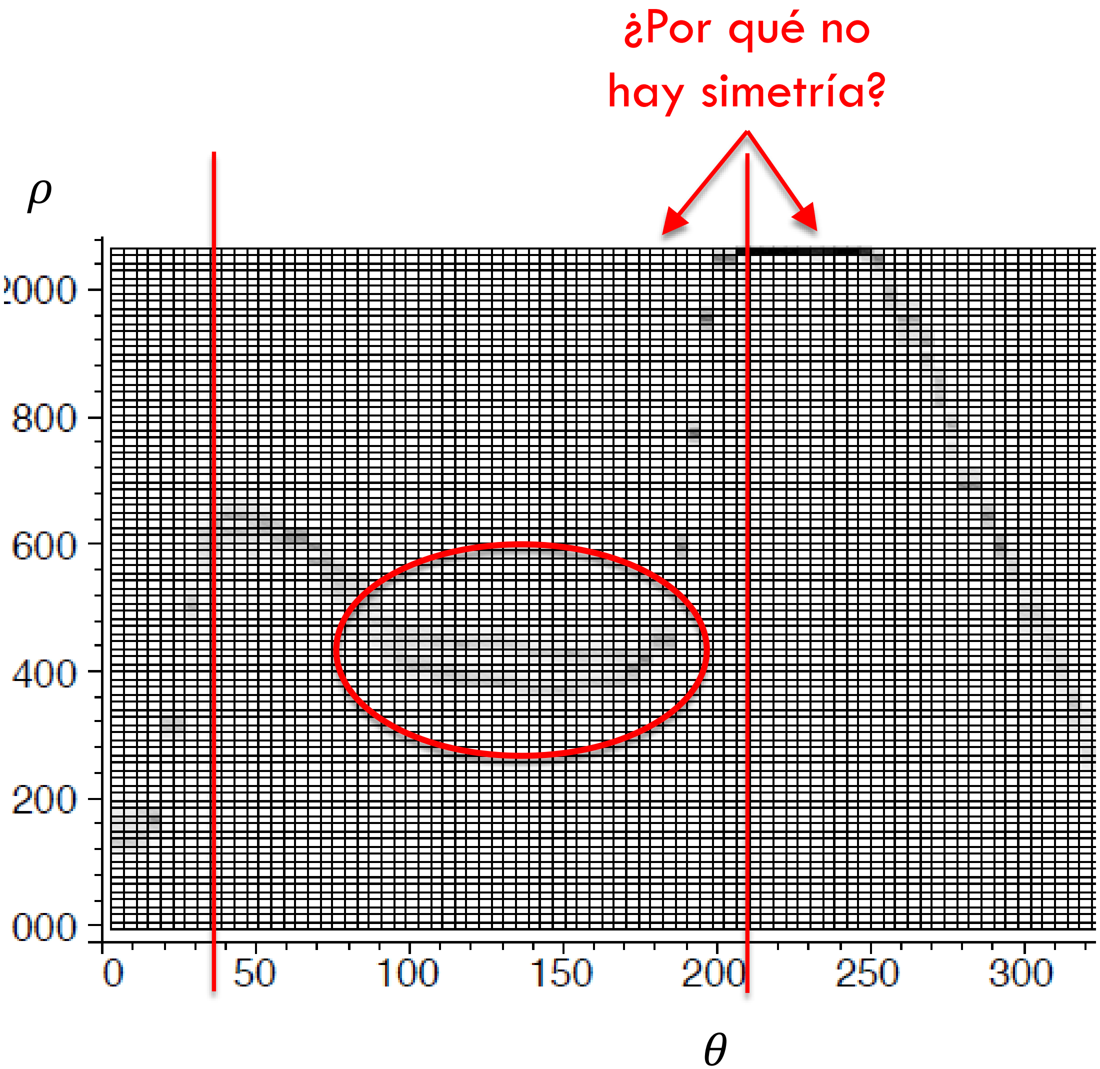
- Eje de rotación centrado y vertical.
- Se rota la celda, explorando 72 posiciones distintas θ (360° en incrementos de 5°)
- Registramos la salida de la celda ρ, [0 – 2047].
- Cada posición se mide 100 veces.
- Usando estas 7200 mediciones se construye un histograma, a partir del cuál se puede graficar un gráfico de densidad.

Un modelo para este experimento

Podríamos esperar que haya dos picos de intensidad máxima: frente a la fuente de luz, y frente al punto de máxima reflexión.

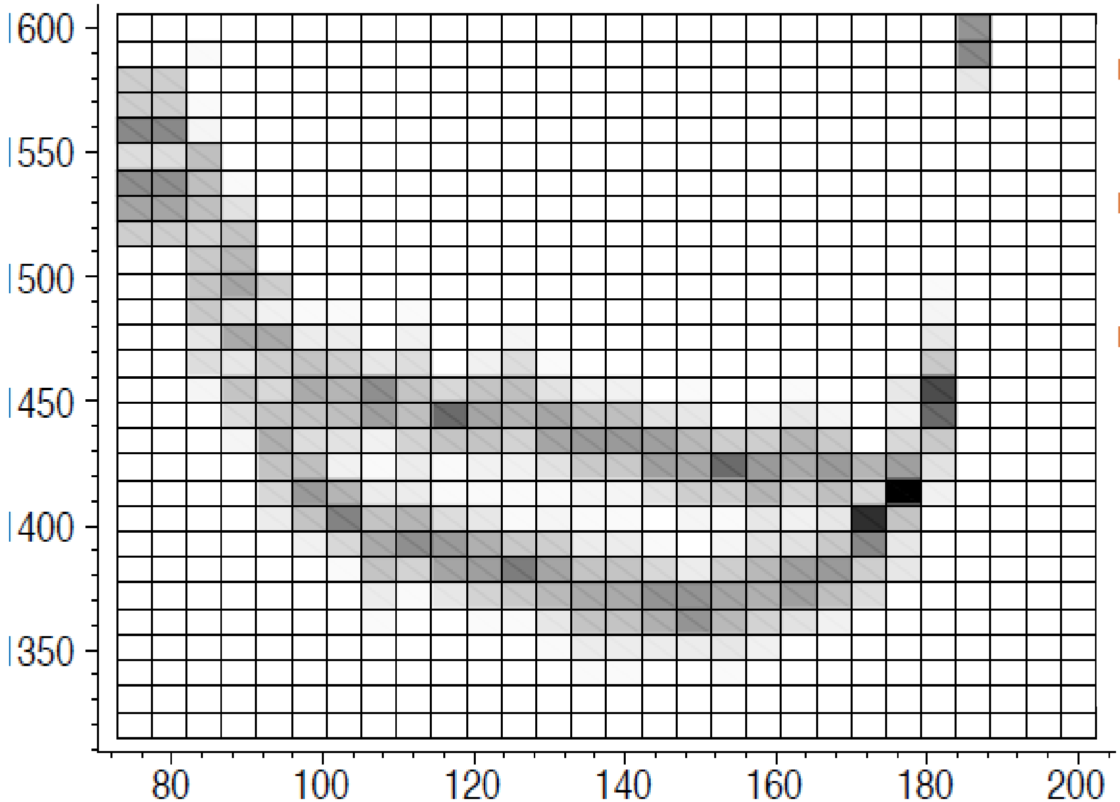






- Se verificó que la posición y la verticalidad del eje fueran las correctas.
- Se eliminaron otras fuentes de luz (neón, ventanas, etc.).
- Se verificó que la superficie interior de la cubeta estuviera limpia y brillante.
- Se verificó la calidad del ángulo de rotación controlado por un robot industrial.
- Se descubrió que la celda sufre de una histéresis: la lectura en el tiempo t no es independiente de las anteriores: zonas de mayor intensidad lumínica sesgan el sensor.

El pico más alto corresponde con la posición angular donde la celda está frente a la luz (210°). Otro pico se observa en la dirección opuesta (30°) donde la celda está frente al reflejo principal.



- No pudieron encontrar la razón exacta de este fenómeno.
- Las mediciones se tomaron en dos días distintos.
- Es posible que algunas condiciones de temperatura, humedad, y ruido (del sensor) hayan cambiado de un día al otro.

Este ejemplo nos habla de que un modelo siempre será incompleto, ya que éste nunca tendrá el mismo comportamiento que el fenómeno real debido al sinfín de variables que no es posible tomar en cuenta en el modelo.

Algunas lecciones

- Todo modelo de un fenómeno real (i.e. no formal) es *incompleto*.
  - ***Siempre hay variables ocultas, no consideradas en el modelo*** *que influencian el fenómeno: no es posible construir un modelo exacto sin variables ocultas.*
- El efecto de estas variables ocultas es que el modelo y el fenómeno nunca tendrán un comportamiento reproducible exactamente.
  - *La* ***incertidumbre aparece como consecuencia directa de la incompletud****. El modelo puede no tomar en cuenta completamente los datos y puede que no prediga exactamente el comportamiento del fenómeno.*

Preguntas centrales

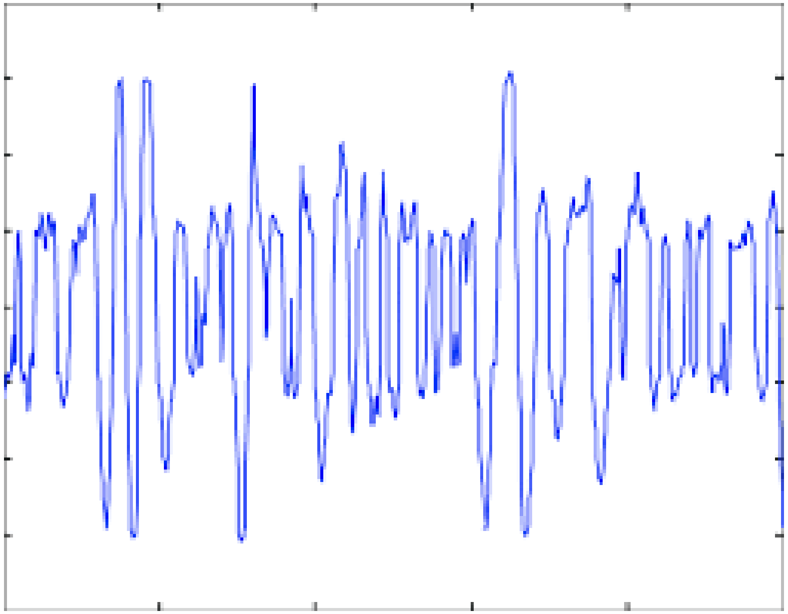
- Cualquier programa que modele un fenómeno real debe enfrentar una dificultad central:

¿Cómo podría este programa usar un modelo incompleto del fenómeno para razonar, decidir y actuar de manera efectiva?

Y entonces:

¿Cómo construir artefactos inteligentes con este programa?

¿Símbolos o datos?



Paradigma simbólico de la programación (enfoque **F+I**)

- El fenómeno se modela mediante un sistema **formal.**
- La realidad se **abstrae** mediante un **conjunto** de **símbolos** y sus relaciones especificadas por el programador.
- El modelo se construye **para predecir la realidad** mediante una **interpretación** de los símbolos.
- El problema del agente es **mapear sus lecturas sensoriales a los símbolos** para saber qué interpretación es la correcta y tomar decisiones.

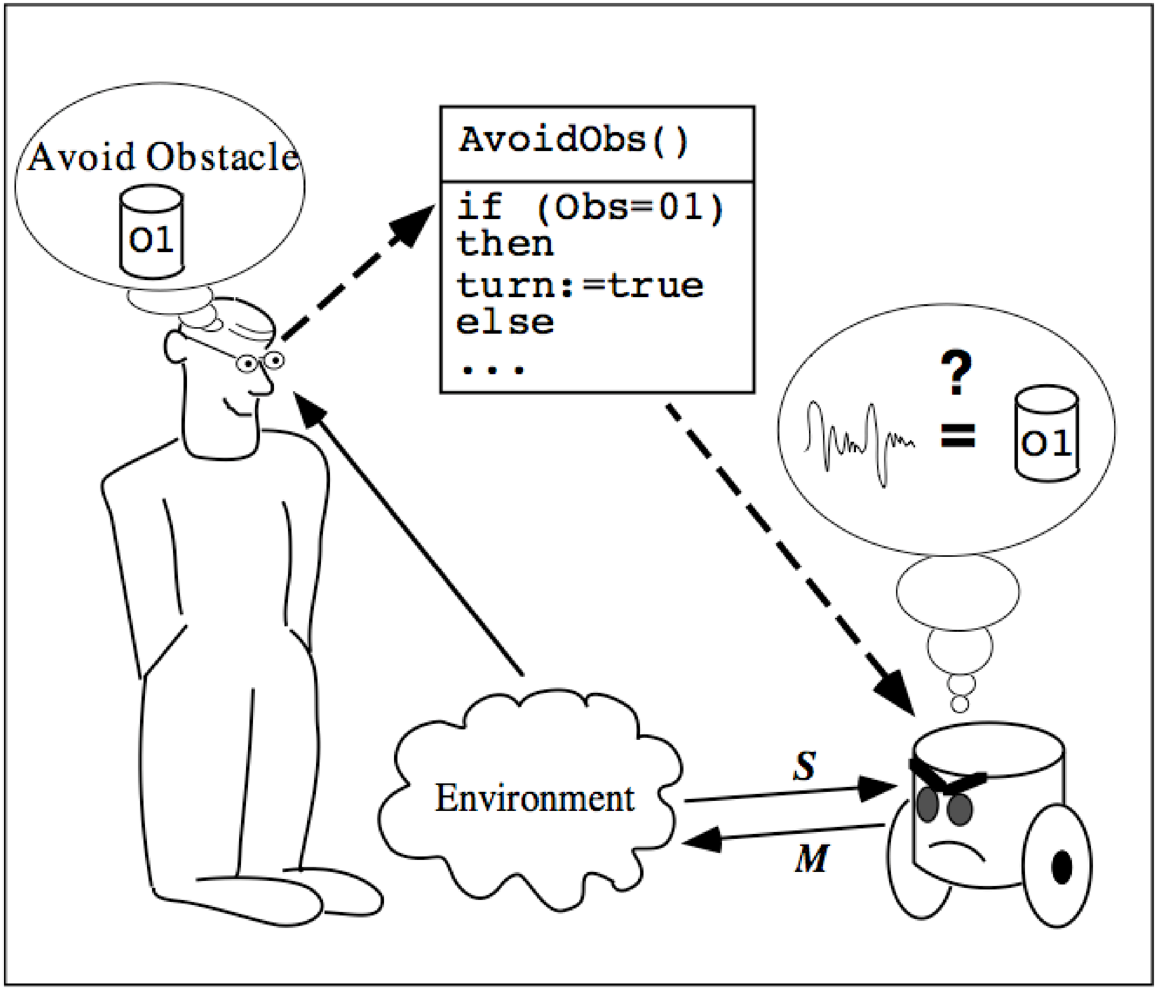


Figure 1: The symbolic approach in robotics.

Paradigma probabilista de la programación (enfoque **F+D**)

- El fenómeno se modela mediante un sistema **formal**.
- La realidad se **abstrae** mediante una **descripción** de una estructura probabilista que relaciona las variables del modelo.
- El modelo se construye, mediante la experiencia o el conocimiento a priori, para **dar forma a la relación** entre las variables.
- El problema del agente es **inferir (predecir) los valores de las variables** para tomar decisiones en función del flujo de datos.

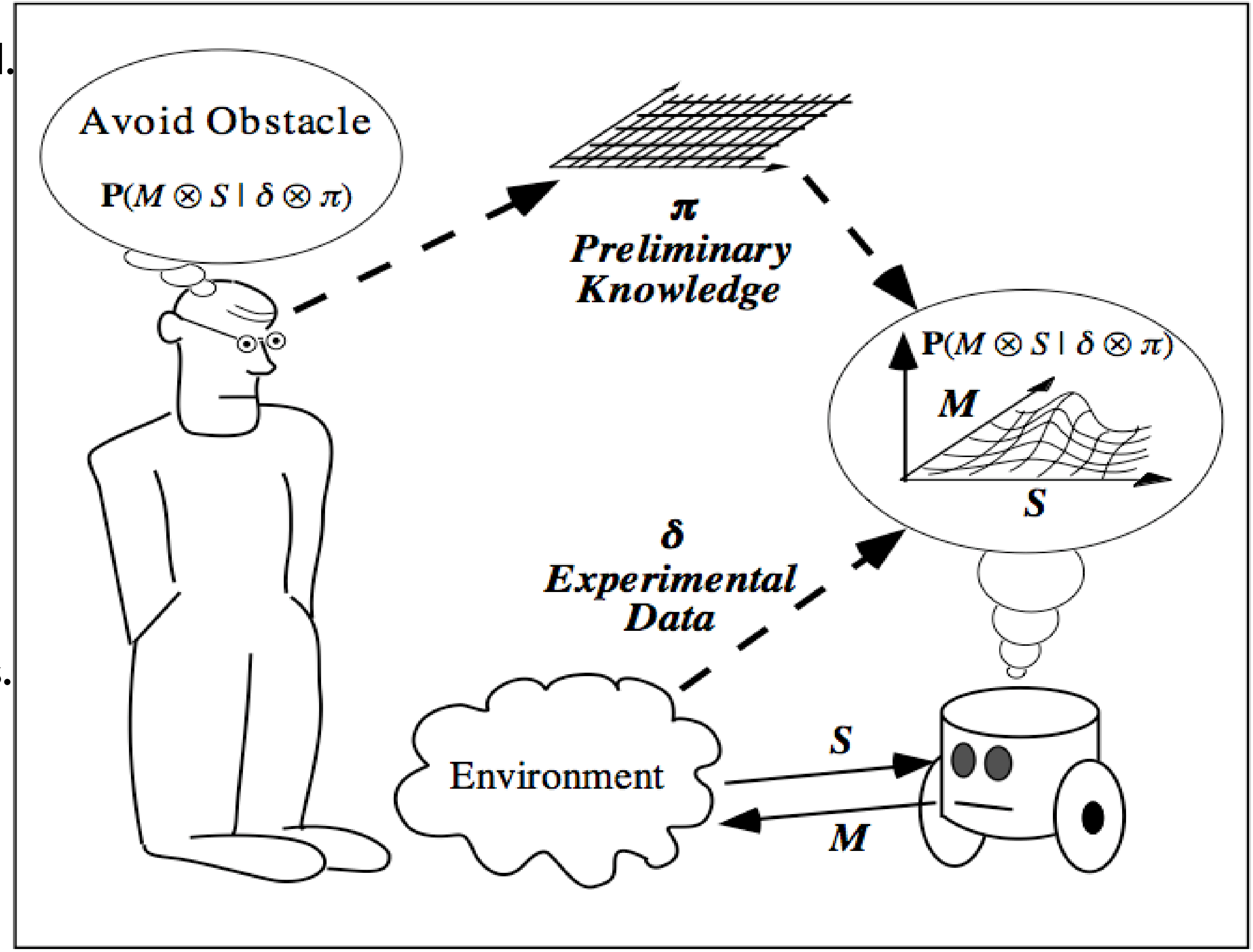


Figure 2: The probabilistic approaches in robotics.

Programación Bayesiana

- El propósito de la Programación Bayesiana es construir artefactos inteligentes con una teoría bien establecida: el cálculo probabilista.
- En este paradigma, el programador no propone un modelo exacto sino que expresa un "lienzo" probabilista, un marco con parámetros abiertos sobre el cual los datos experimentales (del fenómeno) darán forma y figura a la relación que guardan las variables del modelo.

Fundamento teórico

Incompleteness

Preliminary Knowledge
+
Experimental Data
=
Probability Distributions

Maximum Entropy Principle

$$-\sum p_i \ln(p_i)$$

Uncertainty

Bayesian Inference

$$P(a \quad b \mid \pi) = P(a \mid \pi) P(b \mid a \quad \pi) = P(b \mid \pi) P(a \mid b \quad \pi)$$

$$P(\neg a \mid \pi) + P(\neg a \mid \pi) = 1$$

Decision

¿Qué es la Programación Bayesiana?

Donde se introduce la metodología.

El azar es sólo la medida de nuestra ignorancia. Para emprender cualquier cálculo de probabilidades, e incluso para que este cálculo tenga un sentido, tenemos que admitir, como punto de partida, una hipótesis o una convención, que siempre comprende una cierta dosis de arbitrariedad. En la elección de esta convención, sólo podemos guiarnos por el principio de razón suficiente. Desde este punto de vista, toda ciencia no sería más que aplicaciones inconscientes del cálculo de probabilidades. Condenar este cálculo sería condenar toda la ciencia.

Henri Poincaré, La science et l'hypothèse (Poincaré, 1902)

Programa Bayesiano

Más vale una respuesta aproximada a la pregunta correcta, que a menudo es imprecisa, que una respuesta exacta a la pregunta incorrecta, que siempre se puede precisar.
John W. Tuckey

$$\text{Program}(\mathbf{Pr})\begin{cases}\text{Description}(\mathbf{Ds})\begin{cases}\text{Specification}(\mathbf{Sp})\begin{cases}\text{Variables}(\mathbf{Va})\\ \text{Decomposition}(\mathbf{Dc})\\ \text{Forms}(\mathbf{Fo})\end{cases}\\ \text{Identification}(\mathbf{Id})\end{cases}\\ \text{Question}(\mathbf{Qu})\end{cases}$$

Programa Bayesiano = Descripción + Pregunta

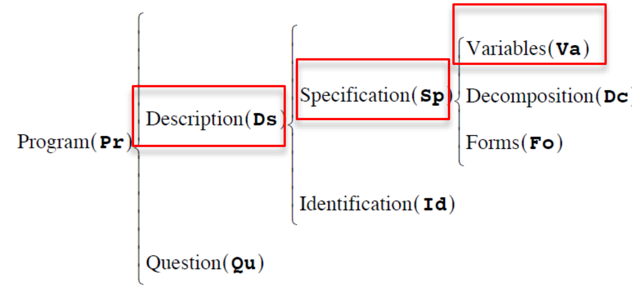
Un ejemplo: filtrado Spam

- Objetivo: Queremos clasificar textos (correos) en una de las dos categorías “nospam” (“ham”) o “spam”.
- La única información que podemos usar para clasificar los correos es su contenido: un conjunto de palabras.

Un ejemplo: filtrado Spam

- El clasificador debería además ser capaz de adaptarse a su usuario y aprender de la experiencia.
  - A partir de una configuración estándar, el clasificador debería modificar sus parámetros internos cuando el usuario esté en desacuerdo con su propia decisión.
  - Debe entonces adaptarse a los criterios del usuario para elegir entre “ham” y “spam”.
  - Deberá mejorar sus resultados conforme va encontrando cada vez más correos clasificados.
  - El clasificador usa un diccionario de N palabras. Cada correo será clasificado según la presencia o ausencia de cada una de las palabras del diccionario.

Variable



- Las variables necesarias para escribir este programa son las siguientes:
  1. $Spam$: una variable binaria, verdadera si el correo es spam, falsa de lo contrario.
  2. $W_0, W_1, \dots, W_{N-1}$: N variables binarias. W_n es verdadera si la palabra n del diccionario está presente en el texto.

Estas N+1 variables binarias son toda la información que tenemos sobre un correo.

Probabilidad

- Una variable puede tener un valor, y sólo uno, en un tiempo dado, por lo tanto el valor de $Spam$ es verdadero o falso, en tanto el correo puede ser spam o no.
- Sin embargo, este valor puede ser desconocido. Desconocido no significa que no disponemos de ninguna información relativa a $Spam$. Por ejemplo, podríamos saber que en promedio hay un 25% de correos ham. Esto se formaliza así:
  1. $P([Spam = \text{falso}]) = P(\overline{Spam}) = 0.25$
  2. $P([Spam = \text{verdadero}]) = P(Spam) = 0.75$

Regla de la normalización

- En vista de lo anterior, los *grados de credibilidad sobre los valores de la variable* $Spam$ son complementarios. Es decir:

$$P(\overline{Spam}) + P(Spam) = 1.0$$

- Como hemos dicho, esta propiedad es verdadera para cualquier variable X discreta, por lo que su distribución de probabilidad debe estar necesariamente *normalizada*:

$$\sum_{x \in X} P([X = x]) = \sum_{X} P(X) = 1.0$$

Probabilidad condicional

- Podríamos estar interesados en la probabilidad de una cierta variable sobre la base de alguna información. A esto se le llama *probabilidad condicional*.
- Por ejemplo, podríamos interesarnos en **la probabilidad de la aparición de una cierta palabra en un mensaje spam**: $P(W_j|Spam)$

$$P([W_j = \text{verdadero}]|[Spam = \text{verdadero}]).$$

El símbolo | separa las variables en dos conjuntos: a la derecha están las variables cuyos valores se conocen con certeza; a la izquierda las variables sondeadas (buscadas).

Probabilidad condicional

- $P(W_j|Spam)$, en palabras, *es la distribución probabilidad sobre la presencia de* W_j *sabiendo que el correo es spam*. Por ejemplo:
  1. $P(\overline{W_j}|Spam) = 0.9996$
  2. $P(W_j|Spam) = 0.0004$
- La regla de la normalización aplica de manera análoga:

$$\sum_{X} P(X \mid Y) = 1.0$$

- Por lo tanto:

$$\sum_{W_j} P(W_j \mid Spam) = 1.0$$

Conjunción de variables

- Podemos interesarnos también a la conjunción de dos variables: $P\left(Spam \wedge W_j\right)$
- La conjunción $Spam \wedge W_j$ es una nueva variable que puede tomar 4 valores distintos:

$$\{(\text{falso}, \text{falso}), (\text{falso}, \text{verdadero}), (\text{verdadero}, \text{falso}), (\text{verdadero}, \text{verdadero})\}$$

- Esto se generaliza a la conjunción de un número arbitrario de variables. Por ejemplo, nos vamos a interesar en la *distribución de probabilidad conjunta* de la conjunción de N+1 variables:

$$P\left(Spam \wedge W_0 \wedge \cdots \wedge W_j \wedge \cdots \wedge W_{N-1}\right)$$

Regla de la conjunción (Teorema de Bayes)

- La probabilidad conjunta de dos variables X y Y se puede calcular gracias a la *regla de la conjunción*:

$$P(X \wedge Y) = P(X) \times P(Y \mid X)$$
$$= P(Y) \times P(X \mid Y)$$

- Esta regla adquiere otra forma muy popular llamada *Teorema de Bayes*:

$$P(X \mid Y) = \frac{P(X) \times P(Y \mid X)}{P(Y)}$$

Regla de la conjunción (Teorema de Bayes)

- Por ejemplo

$$P\left(Spam \wedge W_j\right) = P(Spam) \times P\left(W_j \mid Spam\right)$$
$$= 0.75 \times 0.0004$$

- $$= 0.003$$
- $$= P\left(W_j\right) \times P(Spam \mid W_j)$$

Regla de la marginalización

- Esta regla dice que:

$$\sum_{X} P(X \wedge Y) = P(Y)$$

Distribución conjunta y preguntas

- La distribución conjunta nos da toda la información que necesitamos saber acerca de la conjunción de dos o más variables. En efecto, usando las reglas de la conjunción y la marginalización tenemos:

1. $P(Y) = \sum_X P(X \wedge Y)$
2. $P(X) = \sum_Y P(X \wedge Y)$
3. $P(Y|X) = \frac{P(X \wedge Y)}{\sum_Y P(X \wedge Y)}$
4. $P(X|Y) = \frac{P(X \wedge Y)}{\sum_X P(X \wedge Y)}$

Distribución conjunta y preguntas

- Para nuestro ejemplo de correos spam, necesitamos la distribución conjunta

$$P\left(Spam \wedge W_0 \wedge \cdots \wedge W_j \wedge \cdots \wedge W_{N-1}\right)$$

para calcular cualquiera de las posibles $3^{(N+1)} - 2^{(N+1)}$ preguntas que podemos imaginar con N+1 variables.

- Una *pregunta* se define mediante la partición del conjunto de variables en tres subconjuntos:
  - Las variables buscadas (a la izquierda de la barra).
  - Las variables conocidas (a la derecha de la barra).
  - Las variables libres.

Distribución conjunta y preguntas

- Ejemplos: 6 preguntas para nuestro ejemplo:

$$P\left(Spam \wedge W_0 \wedge \cdots \wedge W_j \wedge \cdots \wedge W_{N-1}\right)$$

$$P(Spam) = \sum_{W_0 \wedge \cdots \wedge W_{N-1}} P\left(Spam \wedge W_0 \wedge \cdots \wedge W_j \wedge \cdots \wedge W_{N-1}\right)$$

$$P\left(W_j\right) = \sum_{Spam \wedge W_0 \wedge \cdots \wedge W_{j-1} \wedge W_{j+1} \ldots \wedge W_{N-1}} P\left(Spam \wedge W_0 \wedge \cdots \wedge W_j \wedge \cdots \wedge W_{N-1}\right)$$

$$P(W_j|Spam) = \frac{P(Spam \wedge W_j)}{P(Spam)} =$$

$$\frac{\displaystyle\sum_{W_0 \wedge \cdots \wedge W_{j-1} \wedge W_{j+1} \ldots \wedge W_{N-1}} P(Spam \wedge W_0 \wedge \cdots \wedge W_j \wedge \cdots \wedge W_{N-1})}{\displaystyle\sum_{W_0 \wedge \cdots \wedge W_{N-1}} P(Spam \wedge W_0 \wedge \cdots \wedge W_j \wedge \cdots \wedge W_{N-1})}$$

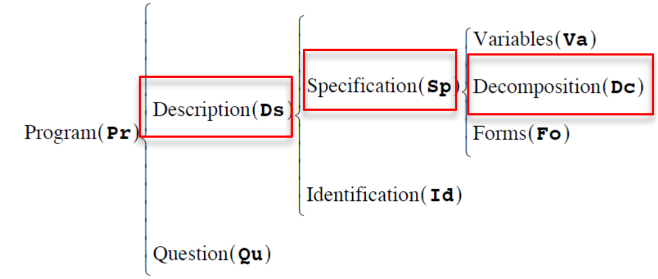
$$P\left(Spam|W_j\right) = \frac{P\left(Spam \wedge W_j\right)}{P\left(W_j\right)} =$$

$$\frac{\sum_{W_0 \wedge \cdots \wedge W_{j-1} \wedge W_{j+1} \ldots \wedge W_{N-1}} P\left(Spam \wedge W_0 \wedge \cdots \wedge W_j \wedge \cdots \wedge W_{N-1}\right)}{\sum_{Spam \wedge W_0 \wedge \cdots \wedge W_{j-1} \wedge W_{j+1} \ldots \wedge W_{N-1}} P\left(Spam \wedge W_0 \wedge \cdots \wedge W_j \wedge \cdots \wedge W_{N-1}\right)}$$

$$P\left(Spam|W_0 \wedge \cdots \wedge W_j \wedge \cdots \wedge W_{N-1}\right) =$$

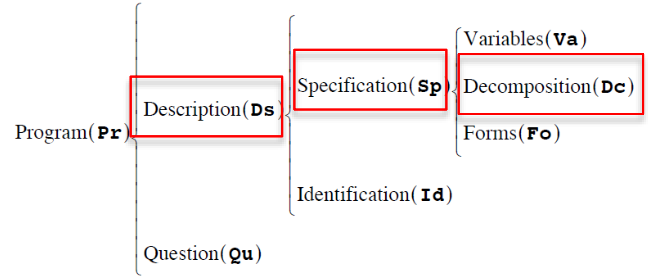
$$\frac{P\left(Spam \wedge W_0 \wedge \cdots \wedge W_j \wedge \cdots \wedge W_{N-1}\right)}{\sum_{Spam} P\left(Spam \wedge W_0 \wedge \cdots \wedge W_j \wedge \cdots \wedge W_{N-1}\right)}$$

Descomposición



- El reto clave de un programador Bayesiano es especificar una forma de calcular la distribución conjunta que tenga tres cualidades:
  - *Buen modelo*
  - *Fácil de calcular*
  - *Fácil de identificar*
- Esto se logra mediante una *descomposición* que reescriba la distribución conjunta como el producto de distribuciones más sencillas.

Descomposición



- Comenzando con la distribución conjunta y aplicando recursivamente la regla de la conjunción obtenemos:

$$P\left(Spam \wedge W_0 \wedge \cdots \wedge W_j \wedge \cdots \wedge W_{N-1}\right)$$
$$= P(Spam) \times P(W_0|Spam) \times P(W_1|Spam \wedge W_0)$$
$$\times \cdots \times P(W_{N-1}|Spam \wedge W_0 \wedge \cdots \wedge W_{N-2})$$

- Esta es una expresión matemática exacta.

Suposiciones de simplificación

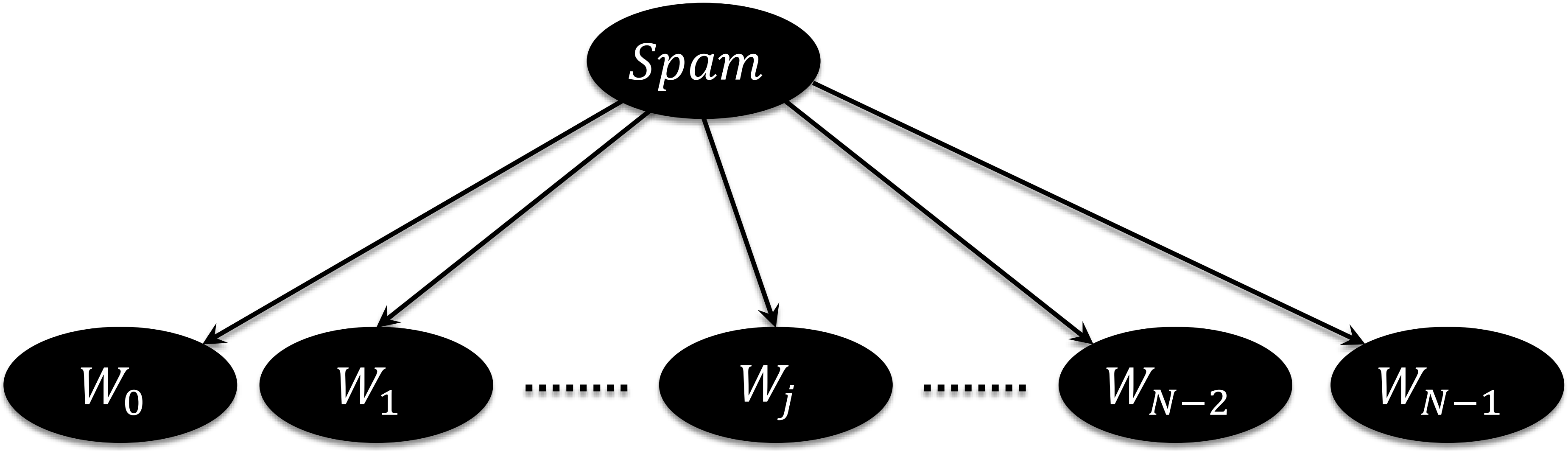
- Podemos simplificar esta descomposición de manera drástica *suponiendo que la probabilidad de que una palabra aparezca, conociendo la naturaleza del texto (spam o no), es independiente de la aparición de otras palabras*:

$$P(W_1|Spam \wedge W_0) = P(W_1|Spam)$$

- Finalmente obtenemos:

$$P\left(Spam \wedge W_0 \wedge \cdots \wedge W_j \wedge \cdots \wedge W_{N-1}\right)$$
$$= P(Spam) \times \prod_{i=0}^{N-1} P(W_i|Spam)$$

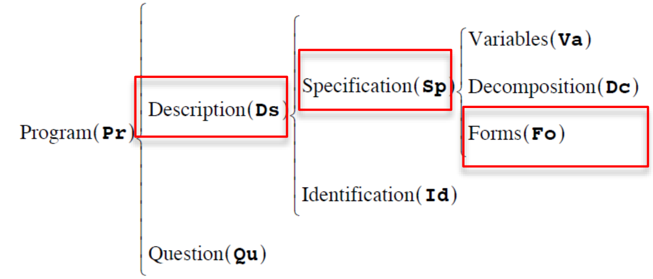
Modelo gráfico (red bayesiana)



Modelo gráfico del filtro bayesiano:

$$P(Spam) \times \prod_{i=0}^{N-1} P(W_i|Spam)$$

Formas paramétricas

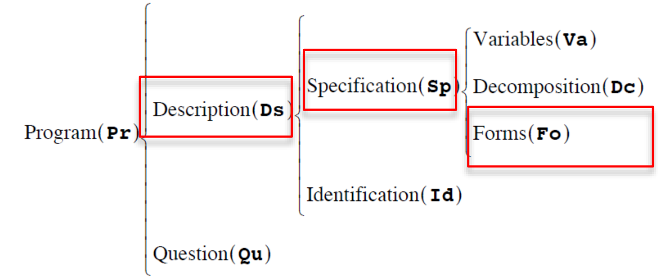


- Para calcular la distribución conjunta, debemos especificar ahora las N+1 distribuciones que aparecen en la descomposición.
- Ya especificamos $P(Spam)$

$P(Spam = \text{falso})$	$P(Spam = \text{verdadero})$
0.25	0.75

- Ahora debemos especificar cada una de las formas $P(W_i|Spam)$

Formas paramétricas



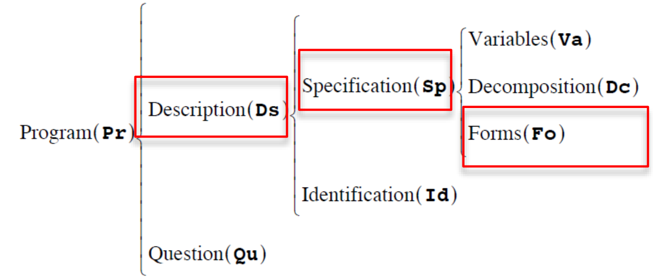
- La primera idea es contar el número de veces que una palabra i del diccionario aparece en ambos spam y ham. Esto nos llevaría ingenuamente a histogramas.

$$P(W_i|\overline{Spam}) = \frac{n_f^i}{n_f}, P(W_i|Spam) = \frac{n_v^i}{n_v}$$

  donde n_f^i (n_v^i) es el número de ocurrencias de W_i en correos ham (spam) y n_f (n_v) es el número total de correos ham (spam).

- El problema con esto es que cuando una observación no se hizo, la probabilidad es nula.
- Una suposición muy fuerte ya que estaría diciendo que lo que no se observó ¡es imposible!

Formas paramétricas



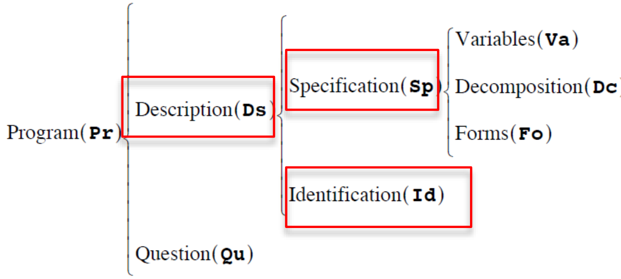
- Por lo tanto, vamos a suponer que las formas paramétricas de $P(W_i|Spam)$ siguen una sucesión de Laplace:

$$P(W_i|\overline{Spam}) = \frac{1+n_f^i}{\mathrm{Card}(W_i)+n_f}$$

$$P(W_i|Spam) = \frac{1+n_v^i}{\mathrm{Card}(W_i)+n_v}$$

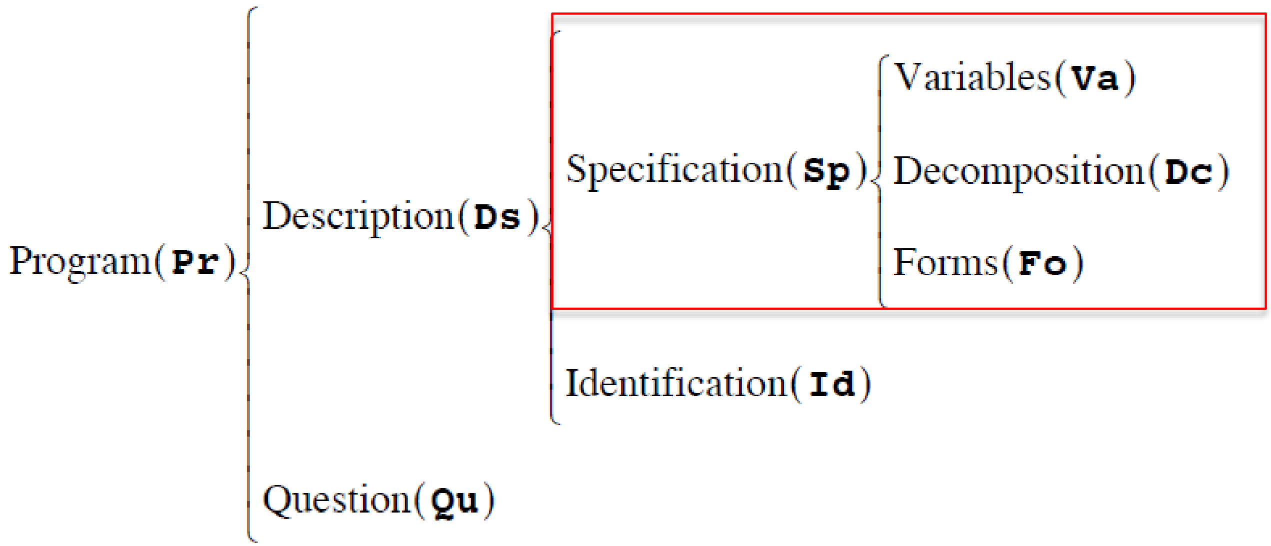
donde $\mathrm{Card}(W_i)$ es el número de valores posibles de W_i ; aquí $\mathrm{Card}(W_i) = 2$.

Identificación



- Las N formas $P(W_i|Spam)$ no se han especificado del todo todavía, porque los 2N+2 parámetros $n_f^{i=0,\ldots,N-1}, n_v^{i=0,\ldots,N-1}, n_f$ y n_v no tienen valores:
- Estos valores podrían definirse si se cuenta con un banco de datos (correos) etiquetados sobre los cuáles se pueden calcular, o bien
- Se pueden ir estimando conforme van llegando correos y se van etiquetando como spam o ham.
- Ambos métodos se pueden combinar.

Especificación=Variables+Descomposición+ Formas paramétricas



- *Variables*: la elección de las variables relevantes para el problema.
- *Descomposición*: la expresión de la distribución conjunta como producto de distribuciones más simples.
- *Formas paramétricas*: la elección matemática de las funciones de cada una de estas distribuciones.

Descripción=Especificación+Identificación

- Program(**Pr**)
  - Description(**Ds**)
    - Specification(**Sp**)
      - Variables(**Va**)
      - Decomposition(**Dc**)
      - Forms(**Fo**)
    - Identification(**Id**)
  - Question(**Qu**)

- *Identificación*: el cálculo de los parámetros de las formas de cada modelo en la descomposición.

Pregunta (interrogando al modelo)

- Program(**Pr**)
  - Description(**Ds**)
    - Specification(**Sp**)
      - Variables(**Va**)
      - Decomposition(**Dc**)
      - Forms(**Fo**)
    - Identification(**Id**)
  - Question(**Qu**)

- Una vez teniendo la descripción (una forma de calcular la distribución conjunta), es posible interrogar al modelo; es decir formular cualquier pregunta.

Pregunta (interrogando al modelo)

- Por ejemplo, luego de algunas simplificaciones, las respuestas a nuestras 6 preguntas son:

1. $P\left(Spam \wedge W_0 \wedge \cdots \wedge W_j \wedge \cdots \wedge W_{N-1}\right) =$

$$P(Spam) \times \prod_{i=0}^{N-1} P(W_i|Spam)$$

2. $P(Spam) = P(Spam)$

3. $P\left(W_j\right) = \sum_{Spam} P(Spam) \times P(W_j|Spam)$

$$P\left(W_j\right) = \left(0.25 \times \frac{1 + n_f^i}{2 + n_f}\right) + \left(0.75 \times \frac{1 + n_v^i}{2 + n_v}\right)$$

Pregunta (interrogando al modelo)

4. $P(W_j|Spam) = \frac{1+n_v^j}{2+n_v}$

Ya que la probabilidad de que aparezca la palabra j, sabiendo que el texto es spam fue especificada en la Descripción.

Pregunta (interrogando al modelo)

5. $P\left(Spam|W_j\right) = \frac{P(Spam) \times P(W_j|Spam)}{\sum_{Spam} P(spam) \times P(W_j|Spam)}$

La probabilidad de que el correo sea spam sabiendo que la palabra j aparece en el texto.

Pregunta (interrogando al modelo)

6. $P(Spam|W_0^{N-1}) = \dfrac{P(Spam) \times \prod_{i=0} P(W_i|Spam)}{\sum_{Spam} P(spam) \times \prod_{i=0} P(W_i|Spam)}$

- El denominador parece ser una constante de normalización.
- Para decidir si el correo es spam, un truco para evitar su cálculo es realizar el cociente:

$$\frac{P(Spam|W_0^{N-1})}{P(\overline{Spam}|W_0^{N-1})} = \frac{P(Spam) \times \prod_{i=0} P(W_i|Spam)}{P(\overline{Spam}) \times \prod_{i=0} P(W_i|\overline{Spam})}$$

Programa Bayesiano

$$\text{Program}(\mathtt{Pr})\begin{cases}\text{Description}(\mathtt{Ds})\begin{cases}\text{Specification}(\mathtt{Sp})\begin{cases}\text{Variables}(\mathtt{Va})\\ \text{Decomposition}(\mathtt{Dc})\\ \text{Forms}(\mathtt{Fo})\end{cases}\\ \text{Identification}(\mathtt{Id})\end{cases}\\ \text{Question}(\mathtt{Qu})\end{cases}$$

$$
\mathbf{Pr}\begin{cases}
\mathbf{Ds}\begin{cases}
\mathbf{Sp}\begin{cases}
\mathbf{Va} \rightarrow \mathit{Spam}, W_0, W_2, \ldots, W_{N-1} \\
\mathbf{Dc} \rightarrow \begin{cases}
\mathbf{P}(\mathit{Spam} \wedge W_0 \wedge \ldots \wedge W_j \wedge \ldots \wedge W_{N-1}) \\
= \mathbf{P}(\mathit{Spam}) \times \prod_{i=1}^{N} \mathbf{P}(W_i \mid \mathit{Spam})
\end{cases} \\
\mathbf{Fo} \rightarrow \begin{cases}
\mathbf{P}(\mathit{Spam}) \rightarrow \begin{cases}
\mathbf{P}([\mathit{Spam} = \text{false}]) = 0{,}25 \\
\mathbf{P}([\mathit{Spam} = \text{true}]) = 0{,}75
\end{cases} \\
\mathbf{P}(W_i \mid \mathit{Spam}) \rightarrow \begin{cases}
\mathbf{P}([W_i = \text{false}] \mid [\mathit{Spam} = \text{false}]) = 1 - \dfrac{1 + n_f^i}{2 + n_f} \\
\mathbf{P}([W_i = \text{true}] \mid [\mathit{Spam} = \text{false}]) = \dfrac{1 + n_f^i}{2 + n_f} \\
\mathbf{P}([W_i = \text{false}] \mid [\mathit{Spam} = \text{true}]) = 1 - \dfrac{1 + n_t^i}{2 + n_t} \\
\mathbf{P}([W_i = \text{true}] \mid [\mathit{Spam} = \text{true}]) = \dfrac{1 + n_t^i}{2 + n_t}
\end{cases}
\end{cases}
\end{cases} \\
\mathbf{Id} \rightarrow \text{Parameters learned from instances}
\end{cases} \\
\mathbf{Qu} \rightarrow \mathbf{P}(\mathit{Spam} \mid W_0 \wedge \ldots \wedge W_j \wedge \ldots \wedge W_{N-1})
\end{cases}
$$

Ejercicio

i	Word i	n_f^i	n_t^i
0	fortune	0	375
1	next	125	0
2	programming	250	0
3	money	0	750
4	you	125	375

Tabla 1

- Histograma derivado del análisis de 1000 correos.
- Restringimos nuestro filtro de spam a un diccionario de 5 palabras, por lo que podemos analizar $2^5 = 32$ subconjuntos.
- Suponga que en la fase de identificación se encontraron 250 correos spam y 750 ham.

TAREA 1

- Calcule la Tabla 2 usando la Tabla 1

i	$P(W_i \mid Spam = \text{falso})$		$P(W_i \mid Spam = \text{verdadero})$	
	$W_i = \text{falso}$	$W_i = \text{verdadero}$	$W_i = \text{falso}$	$W_i = \text{verdadero}$
0				
1				
2				
3				
4				

Tabla 2

TAREA 2

- Calcule $P(Spam|W_0 \wedge \cdots \wedge W_4)$
- ¡Esta Tabla tiene 32 entradas!

TAREA 3

- Analice la Tabla 2 que construyó. ¿Qué puede decir sobre la cuarta palabra?
- Analice los subconjuntos 3,11,12,15 y 27 y comente sus observaciones.