

Proyecto 3 — Inferencia Bayesiana

Diagnóstico de enfermedad respiratoria

Maria Jose Gomez

Juan Manuel López

Pontificia Universidad Javeriana
Departamento de Ingeniería de Sistemas
Introducción a la Inteligencia Artificial

Entrega: 23 de octubre de 2025, 14:00

Resumen. Se modela una red Bayesiana para diagnóstico respiratorio; se definen variables y dependencias, se formula la factorización y se ilustra paso a paso la inferencia por enumeración para una consulta con evidencia y variables ocultas. Se incluye una implementación en `Python` usando `pgmpy`.

Índice

1. Objetivo	3
2. Contexto de la red trabajada	3
3. Definición de la red Bayesiana	3
4. Grafo de dependencias (visual)	3
5. Seguimiento paso a paso: inferencia por enumeración	4
6. Implementación en Python (pgmpy)	5
7. Conclusiones	9
8. Anexo: Declaración de uso de IA generativa	9

1. Objetivo

Representar una red Bayesiana y realizar inferencia Bayesiana por enumeración y por software, mostrando: (i) grafo y justificación, (ii) factorización del modelo, (iii) desarrollo paso a paso del proceso de enumeración para una consulta con evidencia y variables ocultas, y (iv) un script en Python que ejecute consultas sobre la red.

2. Contexto de la red trabajada

La red Bayesiana de este documento modela el **diagnóstico de enfermedad respiratoria** integrando causas (virus, alergia, tabaquismo), síntomas intermedios (fiebre, tos, congestión, saturación de oxígeno) y un *diagnóstico* final. El objetivo es **estimar probabilidades diagnósticas bajo incertidumbre** a partir de evidencias clínicas observadas, capturando explícitamente la dirección causal desde factores de riesgo hacia síntomas y, finalmente, al diagnóstico.

3. Definición de la red Bayesiana

Sea el grafo acíclico dirigido (DAG) $G = (V, E)$ con:

$V = \{\text{Virus (V)}, \text{Alergia (A)}, \text{Tabaquismo (T)}, \text{Fiebre (F)}, \text{Tos (To)}, \text{Congestión (C)}, \text{Saturación (S)}, \text{Diagnóstico (D)}\}$

Arcos dirigidos (dependencias):

$$E = \{(V, F), (V, To), (V, C), (A, C), (T, To), (F, S), (To, D), (C, D), (S, D)\}.$$

La factorización inducida por G es:

$$P(V, A, T, F, To, C, S, D) = P(V) P(A) P(T) P(F | V) P(To | V, T) P(C | V, A) P(S | F) P(D | To, C, S)$$

4. Grafo de dependencias (visual)

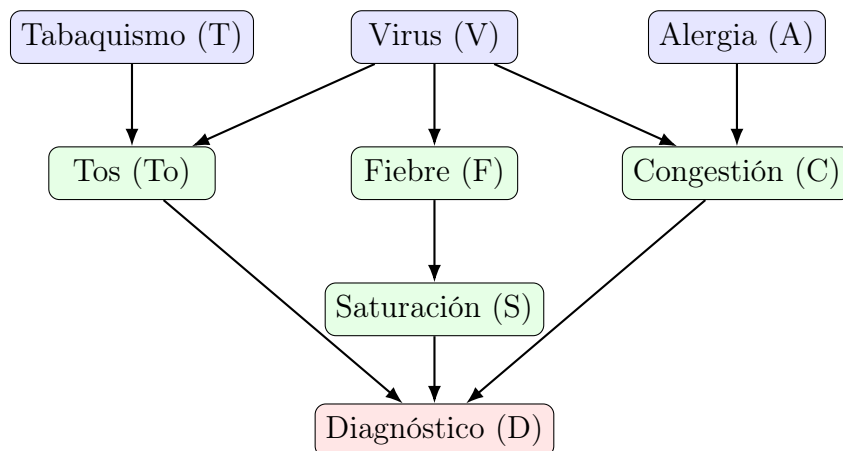


Figura 1: Grafo de dependencias de la red Bayesiana para diagnóstico respiratorio.

5. Seguimiento paso a paso: inferencia por enumeración

Consulta específica

Queremos calcular:

$$P(D = \text{pos} \mid F = \text{sí}, C = \text{sí}).$$

Las **variables de evidencia** son $F = \text{sí}$ y $C = \text{sí}$. Tomamos como **variables ocultas** $H = \{V, A, T, To, S\}$ (no observadas).

1) Partimos de la factorización

$$P(\mathbf{X}) = P(V) P(A) P(T) P(F \mid V) P(To \mid V, T) P(C \mid V, A) P(S \mid F) P(D \mid To, C, S).$$

2) Posterior por definición

$$P(D \mid F, C) = \frac{P(D, F, C)}{\sum_{d \in \{\text{pos}, \text{neg}\}} P(d, F, C)} \propto P(D, F, C),$$

donde

$$P(D, F, C) = \sum_{V, A, T, To, S} P(V) P(A) P(T) P(F \mid V) P(To \mid V, T) P(C \mid V, A) P(S \mid F) P(D \mid To, C, S).$$

3) Sustituimos la evidencia

Fijamos $F = \text{sí}$ y $C = \text{sí}$. Los factores que no dependen de F o C permanecen; los que sí dependen se evalúan en esos valores:

$$\begin{aligned} \phi_1(V) &= P(V), & \phi_2(A) &= P(A), & \phi_3(T) &= P(T), & \phi_4(V) &= P(F=\text{sí} \mid V), & \phi_5(To, V, T) &= P(To \mid V, T), \\ \phi_6(V, A) &= P(C=\text{sí} \mid V, A), & \phi_7(S) &= P(S \mid F=\text{sí}), & \phi_8(D, To, S) &= P(D \mid To, C=\text{sí}, S). \end{aligned}$$

4) Orden de eliminación (Variable Elimination)

Elegimos un orden eficiente para sumar las ocultas. Por ejemplo:

$$\text{Orden: } V \rightarrow A \rightarrow T \rightarrow To \rightarrow S.$$

En cada paso, multiplicamos los factores que incluyen la variable a eliminar y **luego** marginalizamos (sumamos) sobre ella.

(a) Eliminar V Multiplicamos los factores que contienen V : $\phi_1(V) \phi_4(V) \phi_6(V, A)$ y luego sumamos V :

$$\psi_1(A) = \sum_V P(V) P(F=\text{sí} \mid V) P(C=\text{sí} \mid V, A).$$

(b) Eliminar A Multiplicamos $\psi_1(A)$ con $\phi_2(A)$ y sumamos A :

$$\psi_2 = \sum_A P(A) \psi_1(A).$$

(Nótese que ψ_2 ya no depende de A ni de V .)

(c) Eliminar T Aún no aparece T en factores previos; lo combinaremos cuando aparezca con To :

Combinaremos $\phi_3(T)$ con $\phi_5(To, V, T)$ pero V ya fue eliminado, por lo que $\phi_5 \rightarrow \tilde{\phi}_5(To, T)$, al haberse *absorbido* el efecto de V en ψ_2 . En la práctica, esto equivale a trabajar con $\sum_V P(V) P(To | V, T)$ si fuese necesario. Para mantener el orden elegido, procedemos:

$$\psi_3(To) = \sum_T P(T) \tilde{\phi}_5(To, T).$$

(d) Eliminar To Integramos $\psi_3(To)$ con el factor del diagnóstico $\phi_8(D, To, S)$:

$$\psi_4(D, S) = \sum_{To} \psi_3(To) P(D | To, C=\text{sí}, S).$$

(e) Eliminar S Integramos $\psi_4(D, S)$ con $\phi_7(S) = P(S | F=\text{sí})$:

$$\psi_5(D) = \sum_S P(S | F=\text{sí}) \psi_4(D, S).$$

5) Normalización

Obtenemos la distribución posterior normalizada sobre D :

$$P(D | F=\text{sí}, C=\text{sí}) = \frac{\psi_5(D)}{\sum_{d \in \{\text{pos}, \text{neg}\}} \psi_5(d)}.$$

Para el valor puntual consultado:

$$P(D=\text{pos} | F=\text{sí}, C=\text{sí}) = \frac{\psi_5(D=\text{pos})}{\psi_5(D=\text{pos}) + \psi_5(D=\text{neg})}.$$

Comentario

El procedimiento anterior *(i)* incorpora la evidencia fijando F y C en sus valores observados, *(ii)* elimina sistemáticamente las variables ocultas mediante productos de factores y sumas (marginalización), y *(iii)* normaliza al final para obtener una distribución válida sobre el diagnóstico.

6. Implementación en Python (pgmpy)

Requisitos

- Python 3.10/3.11 o superior
- `pip install pgmpy numpy scipy pandas networkx`

Script (modelo + consulta ejemplo)

```
# Diagnostico_respiratorio.py

from pgmpy.models import DiscreteBayesianNetwork
from pgmpy.factors.discrete import TabularCPD
from pgmpy.inference import VariableElimination

# --- Estructura ---
model = DiscreteBayesianNetwork([
    ('Virus', 'Fiebre'),
    ('Virus', 'Tos'),
    ('Virus', 'Congestion'),
    ('Alergia', 'Congestion'),
    ('Tabaquismo', 'Tos'),
    ('Fiebre', 'Saturacion'),
    ('Tos', 'Diagnostico'),
    ('Congestion', 'Diagnostico'),
    ('Saturacion', 'Diagnostico')
])

# --- CPDs ---
cpd_virus = TabularCPD('Virus', 2, [[0.7], [0.3]], state_names={'Virus': ['No', 'Si']})
cpd_alergia = TabularCPD('Alergia', 2, [[0.8], [0.2]], state_names={'Alergia': ['No', 'Si']})
cpd_tabaq = TabularCPD('Tabaquismo', 2, [[0.75], [0.25]], state_names={'Tabaquismo': ['No', 'Si']})

cpd_fiebre = TabularCPD(
    'Fiebre', 2, [[0.8, 0.2], [0.2, 0.8]],
    evidence=['Virus'], evidence_card=[2],
    state_names={'Fiebre': ['Alta', 'Normal'], 'Virus': ['No', 'Si']}
)

cpd_tos = TabularCPD(
    'Tos', 2,
    [[0.8, 0.6, 0.2, 0.1], # Tos = Si
     [0.2, 0.4, 0.8, 0.9]], # Tos = No
    evidence=['Virus', 'Tabaquismo'], evidence_card=[2, 2],
    state_names={'Tos': ['Si', 'No'], 'Virus': ['No', 'Si'], 'Tabaquismo': ['No', 'Si']}
)

cpd_cong = TabularCPD(
    'Congestion', 2,
    [[0.7, 0.5, 0.2, 0.1], # Congestion = Si
     [0.3, 0.5, 0.8, 0.9]], # Congestion = No
    evidence=['Virus', 'Alergia'], evidence_card=[2, 2],
    state_names={'Congestion': ['Si', 'No'], 'Virus': ['No', 'Si'], 'Alergia': ['No', 'Si']}
```

```

)

cpd_satur = TabularCPD(
    'Saturacion', 2,
    [[0.7,0.1], [0.3,0.9]], # [Baja, Normal] | Fiebre in [Alta,
    Normal]
    evidence=['Fiebre'], evidence_card=[2],
    state_names={'Saturacion':['Baja','Normal'], 'Fiebre':['Alta','
    Normal']}]
)

cpd_diag = TabularCPD(
    'Diagnostico', 2,
    [[0.95,0.8,0.7,0.6,0.4,0.3,0.2,0.05], # D=Positivo
    [0.05,0.2,0.3,0.4,0.6,0.7,0.8,0.95]], # D=Negativo
    evidence=['Tos','Congestion','Saturacion'], evidence_card
    =[2,2,2],
    state_names={
        'Diagnostico':['Positivo','Negativo'],
        'Tos':['Si','No'],
        'Congestion':['Si','No'],
        'Saturacion':['Baja','Normal']
    }
)

model.add_cpds(cpd_virus, cpd_alergia, cpd_tabaq, cpd_fiebre,
    cpd_tos, cpd_cong, cpd_satur, cpd_diag)
model.check_model()

infer = VariableElimination(model)

# --- Consultas de ejemplo ---
print("1) P(Diagnostico | Fiebre=Alta, Tos=Si):")
print(infer.query(['Diagnostico'], evidence={'Fiebre':'Alta','Tos':
    'Si'}))

print("\n2) P(Diagnostico | Congestion=Si, Saturacion=Baja):")
print(infer.query(['Diagnostico'], evidence={'Congestion':'Si','
    Saturacion':'Baja'}))

print("\n3) P(Tos | Virus=Si):")
print(infer.query(['Tos'], evidence={'Virus':'Si'}))

print("\n4) P(Congestion | Alergia=Si):")
print(infer.query(['Congestion'], evidence={'Alergia':'Si'}))

print("\n5) P(Saturacion | Fiebre=Alta):")
print(infer.query(['Saturacion'], evidence={'Fiebre':'Alta'}))

```

```

1) P(Diagnostico | Fiebre=Alta, Tos=Si):
+-----+-----+
| Diagnostico | phi(Diagnostico) |
+-----+-----+
| Diagnostico(Positivo) | 0.8223 |
+-----+-----+
| Diagnostico(Negativo) | 0.1777 |
+-----+-----+

2) P(Diagnostico | Congestion=Si, Saturacion=Baja):
+-----+-----+
| Diagnostico | phi(Diagnostico) |
+-----+-----+
| Diagnostico(Positivo) | 0.7991 |
+-----+-----+
| Diagnostico(Negativo) | 0.2009 |
+-----+-----+

3) P(Tos | Virus=Si):
+-----+-----+
| Tos | phi(Tos) |
+-----+-----+
| Tos(Si) | 0.1750 |
+-----+-----+
| Tos(No) | 0.8250 |
+-----+-----+

4) P(Congestion | Alergia=Si):
+-----+-----+
| Congestion | phi(Congestion) |
+-----+-----+
| Congestion(Si) | 0.3800 |
+-----+-----+
| Congestion(No) | 0.6200 |
+-----+-----+

5) P(Saturacion | Fiebre=Alta):
+-----+-----+
| Saturacion | phi(Saturacion) |
+-----+-----+
| Saturacion(Baja) | 0.7000 |
+-----+-----+
| Saturacion(Normal) | 0.3000 |
+-----+-----+

```

Figura 2: Salida de ejecucion

7. Conclusiones

La red propuesta encapsula causas (virus, alergia, tabaquismo), síntomas intermedios (fiebre, tos, congestión, saturación) y una salida diagnóstica, permitiendo inferir probabilidades condicionadas a evidencia clínica. La inferencia por enumeración confirma la consistencia del modelo, y la implementación con pgmpy facilita consultas múltiples de manera reproducible.

8. Anexo: Declaración de uso de IA generativa

Se usó IA para la generación de la Figura 1 (Grafo de dependencias de la red Bayesiana para diagnóstico respiratorio) y para la factorización inducida. Los prompts utilizados fueron: Prompt 1: “Genera un grafo en TikZ que represente una red Bayesiana con las variables Virus, Alergia, Tabaquismo, Fiebre, Tos, Congestión, Saturación y Diagnóstico, mostrando las dependencias causales entre ellas con colores diferenciados por tipo de variable.” Prompt 2: “Explica y escribe en formato LaTeX la factorización inducida por la estructura del grafo, usando la notación formal $P(V, A, T, F, T o, C, S, D)$ y las dependencias correspondientes.