Introducción a la Ciencia de Datos

Maestría en Ciencias de la Computación

Dr. Irvin Hussein López Nava

Lectura 3: Clasificación

Irvin Hussein Lopez Nava • 30 sept

100 puntos

Fecha de entrega: Hoy

Preparar un reporte de máximo dos cuartillas con un resumen del artículo adjunto (secciones 1-4). El reporte debe finalizar con una crítica al artículo, el cual será abordado en la sesión siguiente.

Gasparetto, A., Marcuzzo, M., Zangari, A., & Albarelli, A. (2022). A survey on text classification algorithms: From text to predictions. Information, 13(2), 83.

information-13-00083-v2.pdf

PDF

04

Clasificación


NO DISPAREN, SOY EXPERTO EN ML
PRUÉBALO: ¿QUÉ MODELOS APLICAS
PARA CADA PROBLEMA?
NAIVE BAYES

4.3 Bayesian learning


¿Qué es el Aprendizaje Bayesiano?

Definiciones

ChatGPT

Es un enfoque estadístico y de aprendizaje automático que se basa en el teorema de Bayes para actualizar y refinar las creencias o estimaciones a medida que se obtiene nueva evidencia. Se llama "bayesiano" en honor al estadístico británico Thomas Bayes.

https://chat.openai.com/

Wikipedia

Es un método de inferencia estadística en el que se utiliza el teorema de Bayes para actualizar la probabilidad de una hipótesis a medida que hay más evidencia o información disponible.

https://en.wikipedia.org/wiki/Bayesian_inference

Gemini

Es un enfoque de aprendizaje automático que utiliza la teoría de la probabilidad bayesiana para aprender de los datos. La teoría bayesiana ofrece un marco para calcular la probabilidad de una hipótesis dada la evidencia.

https://gemini.google.com/

Dos razones para usar aprendizaje Bayesiano

Proporciona algoritmos de aprendizaje prácticos:

- Aprendizaje Naive Bayes.
- Aprendizaje de Redes Bayesianas.
- Combina conocimientos previos (probabilidades previas) con datos observados.
- Requiere probabilidades previas.

Proporciona un marco conceptual útil:

- Proporciona un "*gold standard*" para evaluar otros algoritmos de aprendizaje.
- Información adicional sobre Occam’s razor.

Teorema de Bayes

$$P(h|D) = \frac{P(D|h)P(h)}{P(D)}$$

- P(h) = probabilidad previa de la hipótesis h.
- P(D) = probabilidad previa de los datos de entrenamiento D.
- P(h|D) = probabilidad de h dado D.
- P(D|h) = probabilidad de D dado h.

Eligiendo las hipótesis

- Generalmente se prefiere la hipótesis más probable dados los datos de entrenamiento, hipótesis **máxima a posteriori** ^{h}MAP:

$$
\begin{aligned}
h_{MAP} &= arg \max_{h \in H} P(h|D) \\
&= arg \max_{h \in H} \frac{P(D|h)P(h)}{P(D)} \\
&= arg \max_{h \in H} P(D|h)P(h)
\end{aligned}
$$

- Si se asume que $P(h_i) = P(h_j)$, entonces se puede simplificar aún más y elegir la hipótesis de **máxima verosimilitud** (^{h}ML):

$$
h_{ML} = arg \max_{h \in H} P(D|h_i)
$$

Ejemplo

- ¿El paciente tiene cáncer o no?
- Un paciente se hace una prueba de laboratorio y el resultado es positivo. La prueba arroja un resultado positivo correcto sólo en el 98% de los casos en los que la enfermedad está realmente presente, y un resultado negativo correcto en sólo el 97% de los casos en los que la enfermedad no está presente. Además, el 0.008 de toda la población padece este cáncer.

P(cáncer) =
P(+ | cáncer) =
P(+ | ¬cáncer) =

P(¬cáncer) =
P(- | cáncer) =
P(- | ¬cáncer) =

Ejemplo

- ¿El paciente tiene cáncer o no?
- Un paciente se hace una prueba de laboratorio y el resultado es positivo. La prueba arroja un resultado positivo correcto sólo en el 98% de los casos en los que la enfermedad está realmente presente, y un resultado negativo correcto en sólo el 97% de los casos en los que la enfermedad no está presente. Además, el 0.008 de toda la población padece este cáncer.

$P(\text{cáncer}) = 0.008$ $\quad$ $P(\neg\text{cáncer}) = 0.992$

$P(+ \mid \text{cáncer}) = 0.98$ $\quad$ $P(- \mid \text{cáncer}) = 0.02$

$P(+ \mid \neg\text{cáncer}) = 0.03$ $\quad$ $P(- \mid \neg\text{cáncer}) = 0.97$

Fórmulas básicas

- Regla del producto: probabilidad de una conjunción de dos eventos A y B:

$$P(A \wedge B) = P(A|B)\, P(B) = P(B|A)\, P(A)$$

- Regla de la suma: probabilidad de una disyunción de dos eventos A y B:

$$P(A \vee B) = P(A) + P(B) - P(A \wedge B)$$

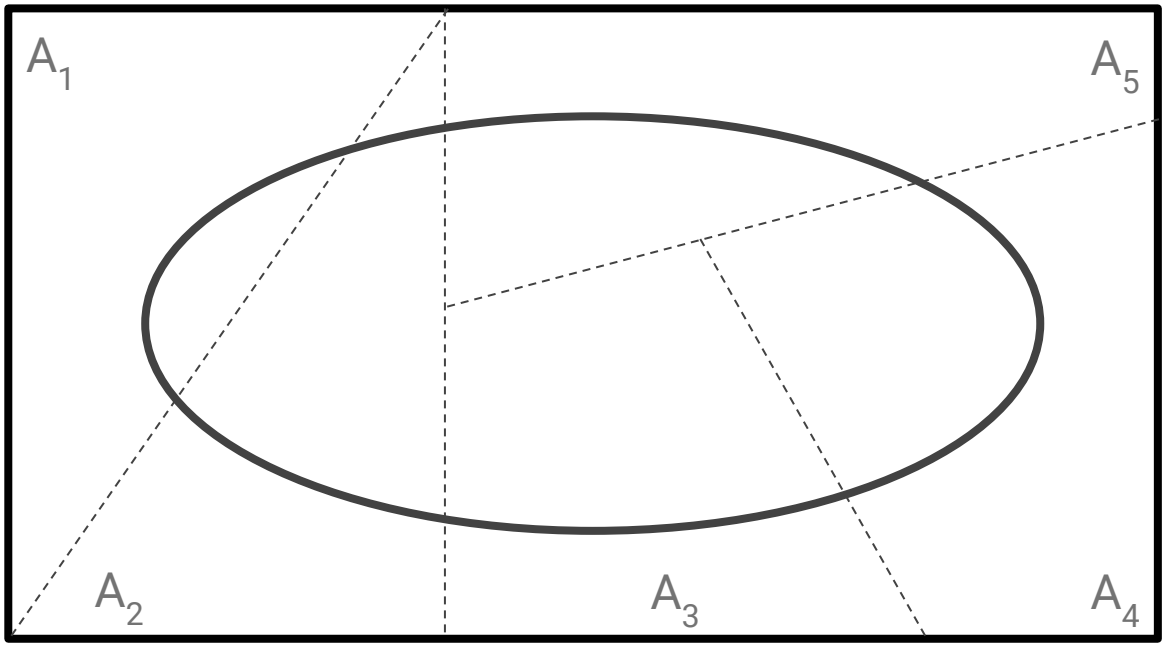
- Teorema de probabilidad total: si los eventos A_1, ..., A_n son mutuamente excluyentes con la suma $P(A_i) = 1$, entonces

$$P(B) = \sum_{i=1}^{n} P(B|A_i)\ P(A_i)$$

Teorema de la probabilidad total

A

B



Hipótesis MAP a fuerza bruta

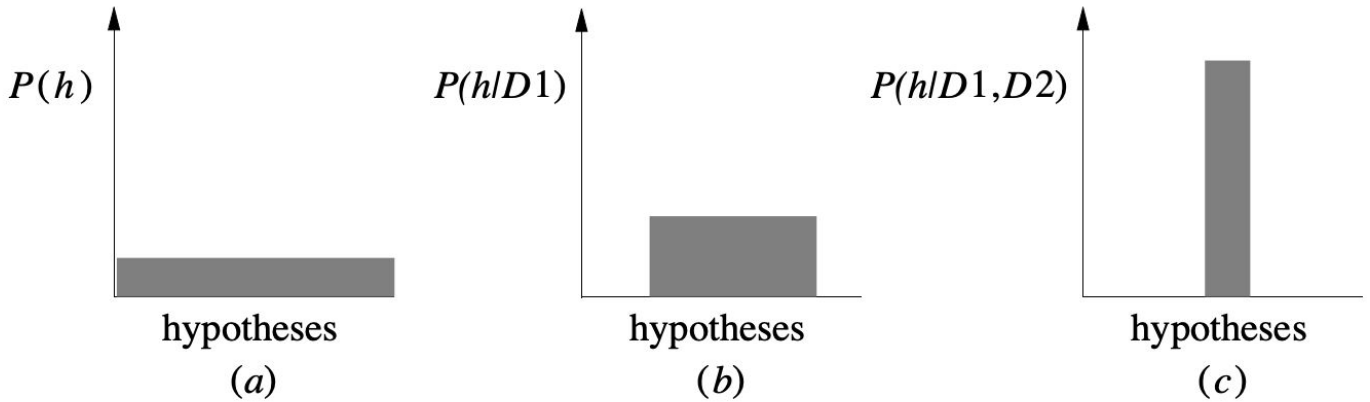
1. Para cada hipótesis h en H, calcular la **probabilidad posterior**.

$$P(h|D) = \frac{P(D|h)P(h)}{P(D)}$$

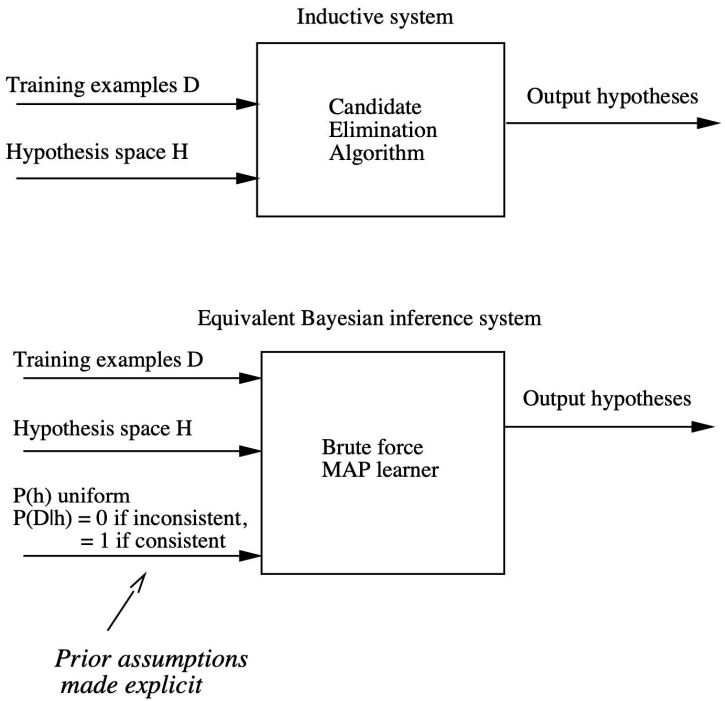
2. Generar la hipótesis h_{MAP} con la **probabilidad posterior** más alta.

$$h_{MAP} = arg \max_{h \in H} P(h|D)$$

Evolución de las probabilidades posteriores



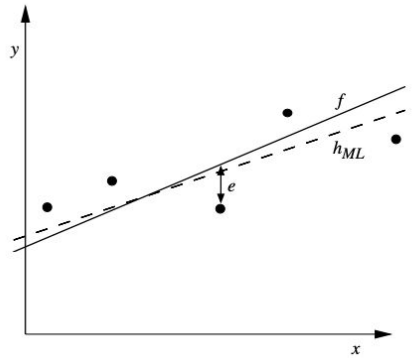
Algoritmos de aprendizaje vs MAP *learners*



Aprender una función con valor real

Considere cualquier función objetivo f de valor real. Ejemplos de entrenamiento $\{x_i, d_i\}$, donde d_i es un valor de entrenamiento ruidoso

- $d_i = f(x_i) + e_i$
- e_i es una variable aleatoria (ruido) extraída independientemente para cada x_i según alguna distribución gaussiana con media = 0.



Entonces la hipótesis de **máxima verosimilitud** ^hML es la que minimiza la suma de errores al cuadrado:

$$h_{ML} = arg \min_{h \in H} \sum_{i=1}^{m} (d_i - h(x_i))^2$$

Clasificación más probable de nuevas instancias

Hasta ahora se ha buscado la hipótesis más probable dados los datos D (^{h}MAP).

- Dada la nueva instancia **x**, ¿cuál es su clasificación más probable?
- ¡$^{h}MAP(x)$ no es la clasificación más probable!

Considerar:

- Tres posibles hipótesis: $P(h_1 \mid D) = 0.4$, $P(h_2 \mid D) = 0.3$, $P(h_3 \mid D) = 0.3$
- Dada la nueva instancia **x**, $h_1(x) = +$, $h_2(x) = -$, $h_3(x) = -$
- ¿Cuál es la clasificación más probable de **x**?

Clasificador óptimo de Bayes

- Ejemplo:

$$P(h_1|D) = 0.4,\ \ P(-|h_1) = 0,\ \ P(+|h_1) = 1$$
$$P(h_2|D) = 0.3,\ \ P(-|h_2) = 1,\ \ P(+|h_2) = 0$$
$$P(h_3|D) = 0.3,\ \ P(-|h_3) = 1,\ \ P(+|h_3) = 0$$

- por lo tanto

$$\sum_{h_i \in H} P(+|h_i)P(h_i|D) = 0.4$$
$$\sum_{h_i \in H} P(-|h_i)P(h_i|D) = 0.6$$

- y

$$arg\ \max_{v_j \in V} \sum_{h_i \in H} P(v_j|h_i)P(h_i|D) = -$$

Clasificador Naive Bayes

- Junto con los árboles de decisión, las redes neuronales y los k-vecinos más cercanos, es uno de los métodos de aprendizaje más prácticos.
- Cuándo usar:
  - Conjunto de entrenamiento moderado o grande.
  - Los atributos que describen las instancias son **condicionalmente independientes** dada una clasificación.
- Algunas aplicaciones exitosas:
  - Diagnóstico médico.
  - Clasificación de documentos de texto.

Formulación

- Suponga funciones objetivo $f : X \rightarrow V$, donde cada instancia x se describe mediante los atributos $\{a_1, a_2, ..., a_n\}$. El valor más probable de $f(x)$ es:

$$
\begin{aligned}
v_{MAP} &= arg\max_{v \in V} P(v_j | a_1, a_2, ..., a_n) \\
v_{MAP} &= arg\max_{v \in V} \frac{P(a_1, a_2, ..., a_n | v_j)\ P(v_j)}{P(a_1, a_2, ..., a_n)} \\
&= arg\max_{v \in V} P(a_1, a_2, ..., a_n | v_j)\ P(v_j)
\end{aligned}
$$

- Suposición ingenua de Bayes: $P(a_1, a_2, ..., a_n | v_j) = \prod_i P(a_i | v_j)$

- lo que da un clasificador Naive Bayes: $v_{NB} = arg\max_{v \in V} P(v_j) \prod_i P(a_i | v_j)$

Algoritmo Naive Bayes

Naive_Bayes_Learn(examples)

For each target value v_j

$$\hat{P}(v_j) \leftarrow estimate\ P(v_j)$$

For each attribute value a_i of each attribute a

$$\hat{P}(a_i|v_j) \leftarrow estimate\ P(a_i|v_j)$$

Classify_New_Instance(x)

$$v_{NB} = arg \max_{v_j \in V} \hat{P}(v_j) \prod_{a_i \in x} \hat{P}(a_i|v_j)$$

Nuestro ejemplo

day	outlook	temp	humidity	wind	play
D1	Sunny	Hot	High	Weak	No
D2	Sunny	Hot	High	Strong	No
D3	Overcast	Hot	High	Weak	Yes
D4	Rain	Mild	High	Weak	Yes
D5	Rain	Cool	Normal	Weak	Yes
D6	Rain	Cool	Normal	Strong	No
D7	Overcast	Cool	Normal	Strong	Yes
D8	Sunny	Mild	High	Weak	No
D9	Sunny	Cool	Normal	Weak	Yes
D10	Rain	Mild	Normal	Weak	Yes
D11	Sunny	Mild	Normal	Strong	Yes
D12	Overcast	Mild	High	Strong	Yes
D13	Overcast	Hot	Normal	Weak	Yes
D14	Rain	Mild	High	Strong	No

Nuestro ejemplo con Naive Bayes

- Considere una nueva instancia:
- {outlook = sunny, temperature = cool, humidity = high, wind = strong}

- Se quiere calcular: $$v_{NB} = arg \max_{v_j \in V} P(v_j) \prod_i P(a_i|v_j)$$

- *P(y) P(sol | y) P(frío | y) P(alto | y) P(fuerte | y) =*
- *P(n) P(sol | n) P(frío | n) P(alto | n) P(fuerte | n) =*
- *v_{NB} = ?*

Nuestro ejemplo con Naive Bayes

- Considere una nueva instancia:
- {outlook = sunny, temperature = cool, humidity = high, wind = strong}

- Se quiere calcular: $$v_{NB} = arg \max_{v_j \in V} P(v_j) \prod_i P(a_i|v_j)$$

- *P(y) P(sol | y) P(frío | y) P(alto | y) P(fuerte | y) = 0,005*
- *P(n) P(sol | n) P(frío | n) P(alto | n) P(fuerte | n) = 0,021*
- *v_{NB} = n*

Sutilezas de Naive Bayes

1. A menudo se viola el supuesto de independencia condicional

$$P(a_1, a_2, ..., a_n|v_j) = \prod_i P(a_i|v_j)$$

... pero de todos modos funciona sorprendentemente bien. Tener en cuenta que no es necesario que los posteriores estimados sean correctos; solo se necesita

$$arg\max_{v_j \in V} \hat{P}(v_j) \prod_i \hat{P}(a_i|v_j) = arg\max_{v_j \in V} P(v_j)\ \ P(a_1, ..., a_n|v_j)$$

Los posteriores Naive Bayes a menudo se acercan de manera poco realista a 1 o 0.

Sutilezas de Naive Bayes

2. ¿Qué pasa si ninguna de las instancias de entrenamiento con el valor objetivo v_j tiene el valor del atributo a_i? Entonces

$$\hat{P}(a_i|v_j) = 0, and... \quad \hat{P}(v_j)\prod_i \hat{P}(a_i|v_j) = 0$$

La solución típica es la estimación bayesiana con

$$\hat{P}(a_i|v_j) \leftarrow \frac{n_c + mp}{n + m}$$

dónde
n es el número de ejemplos de entrenamiento para los cuales v = v_j,
nc número de ejemplos para los cuales v = v_j y a = a_i,
p es la estimación previa de ^P(a_i | v_j), y
m es el peso dado al anterior (i.e., el número de ejemplos "virtuales").

Algoritmo Naive Bayes

Algorithm : Naive Bayes algorithm

```
Given a data set S(x,c);
while N training examples are available do
    for i = 0 .. L do
        estimate P(C=ci) with example in S ;
        for j = 0 .. n do
            for k = 0 .. N do
                estimate P(Xj = xjk | C=ci) ;
            end
        end
    end
end
while testing examples is available do
    get the greatest conditional probabilities calculated for each class
end
```

Redes Bayesianas

Interesantes porque:

- La suposición de Naive Bayes sobre la independencia condicional es demasiado restrictiva.
- Pero es intratable sin algunas de esas suposiciones...
- Las redes de creencia Bayesianas (también conocidas como redes Bayesianas) describen la **independencia condicional** entre subconjuntos de variables.
- Permite combinar conocimientos previos sobre (in)dependencias entre variables con datos de entrenamiento observados.

Independencia condicional

- **Definición**: X es condicionalmente independiente de Y dado Z si la distribución de probabilidad que rige a X es independiente del valor de Y dado el valor de Z; i.e., si

$$(\forall x_i, y_j, z_k) P(X = x_i | Y = y_j, Z = z_k) = P(X = x_i | Z = z_k)$$

- De manera más compacta

$$P(X|Y, Z) = P(X|Z)$$

Independencia condicional

Ejemplo:

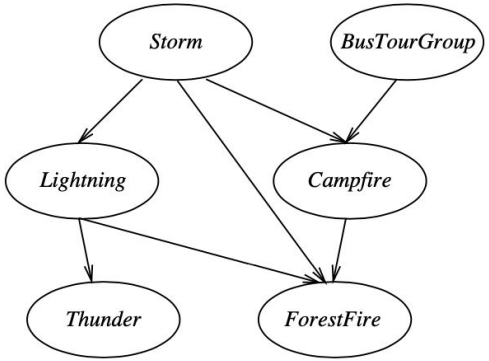
- *Los truenos son* condicionalmente independiente de *la lluvia*, dados los relámpagos

P(Thunder | Rain, Lightning) = P(Thunder | Lightning)

Naive Bayes utiliza independencia condicional para justificar

$$P(X|Y,Z) = P(X|Y,Z)\ P(Y|Z)$$
$$= P(X|Z)\ P(Y|Z)$$

Red de creencia Bayesiana



	S,B	*S,¬B*	*¬S,B*	*¬S,¬B*
C	0.4	0.1	0.8	0.2
¬C	0.6	0.9	0.2	0.8

Campfire

La red representa un conjunto de afirmaciones de **independencia condicional**:

- Se afirma que cada nodo es condicionalmente independiente de sus no descendientes, dados sus predecesores inmediatos.
- Gráfico Acíclico Dirigido (DAG).

Red de creencia Bayesiana

Representa la distribución de probabilidad conjunta sobre todas las variables:

- e.g., *P(Tormenta, BusTourGroup,..., ForestFire)*,
- en general,

$$P(y_1, ..., y_n) = \prod_{i=1}^{n} P(y_i | Parents(Y_i))$$

- donde Parents(Y_i) denota los predecesores inmediatos de Y_i en el grafo,
- entonces, la distribución conjunta está completamente definida por el grafo, más *P (y_i | Parents (Y_i))*.

Inferencia en las BNs

¿Cómo se pueden **inferir** las (probabilidades de) valores de una o más variables de la red, dados los valores observados de otras?

- Bayes Net contiene toda la información necesaria para esta inferencia.
- Si solo hay una variable con valor desconocido, es fácil inferirla.
- En el caso general, el problema es NP-hard.

En la práctica, puede tener éxito en muchos casos:

- Los métodos de inferencia exacta funcionan bien para algunas estructuras de red.
- Los métodos de Monte Carlo "simulan" la red aleatoriamente para calcular soluciones aproximadas.

Aprendizaje de las BNs

Variantes para la tarea de aprendizaje:

- La estructura de la red puede ser conocida o desconocida.
- Los ejemplos de entrenamiento pueden proporcionar valores de todas las variables de la red, o solo de algunas.

Si se conoce la estructura y se observan todas las variables:

- Entonces es tan fácil como entrenar un clasificador Naive Bayes.

Aprendizaje de las BNs

Suponga que se conoce la estructura y que las variables son parcialmente observables.

p.ej. *observar ForestFire, Storm, BusTourGroup, Thunder, pero no Lightning, Campfire,...*

- Similar al entrenamiento de redes neuronales con unidades ocultas.
- De hecho, se pueden aprender tablas de probabilidad condicional de la red mediante el ascenso de gradiente.
- Converjer a la red ***h*** que maximiza (localmente) $P(D \mid h)$.

Ascenso de gradiente para BNs

Sea w_{ijk} una entrada en la tabla de probabilidad condicional para la variable Y_i en la red:

$w_{ijk} = P(Y_i = y_{ij} \mid Parents(Y_i))$ = la lista u_{ik} de valores

e.g., si $Y_i = Campfire$, entonces u_{ik} podría ser $\{Storm = T, BusTourGroup = F\}$

Realizar un ascenso de gradiente repetidamente

1. Actualizar todo w usando datos de entrenamiento D

$$w_{i,j,k} \leftarrow w_{i,j,k} + \eta \sum_{d \in D} \frac{P_h(y_{ij}, u_{i,k} \mid d)}{w_{i,j,k}}$$

2. luego, renormalizar el w_{ijk} para asegurar $\sum_j w_{i,j,k} = 1$ y $0 \leq w_{i,j,k} \leq 1$

Más sobre cómo aprender BNs

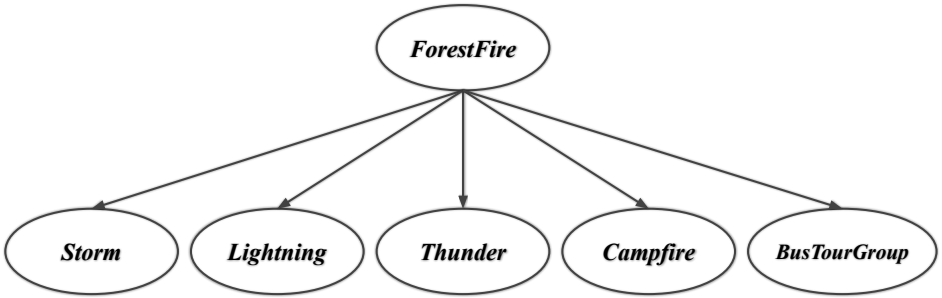
También se puede utilizar el algoritmo EM. Repetidamente:

- Calcular las probabilidades de las variables no observadas, asumiendo *h*,
- Calcular el nuevo w_{ijk} para maximizar $\mathbb{E}\,[ln\ P(D \mid H)]$, donde D ahora incluye variables observadas y (probabilidades calculadas de) no observadas.

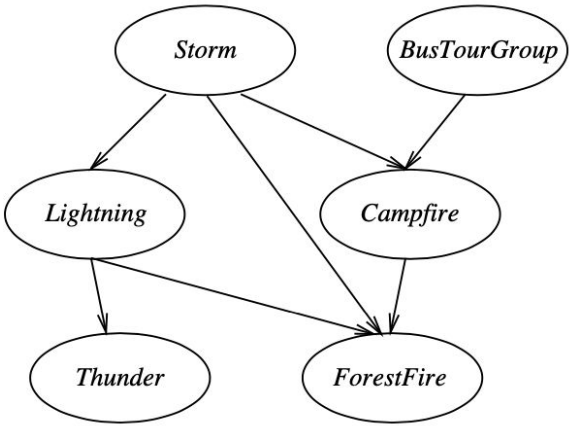
Cuando se desconoce la estructura...

- Los algoritmos utilizan búsqueda *greedy* para sumar/restar aristas y nodos.
- Es un tema de investigación activo.

Resumen: *Naive Bayes vs. Bayes Net*

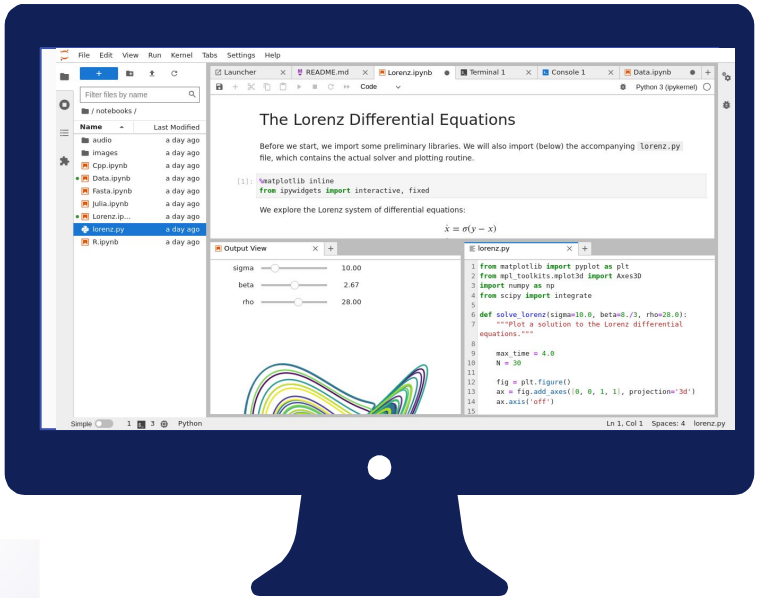


Naive Bayes

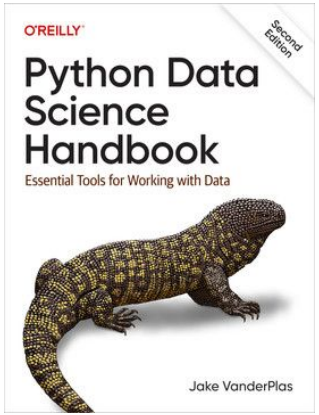


Bayes Net

(Go to live notebook)



Extra Libro



GitHub

05.05-Naive-Bayes.ipynb

Gracias!

¿Alguna pregunta?

hussein@cicese.mx

https://sites.google.com/view/husseinlopeznava

SCAN ME

CREDITS: This presentation was based on a template by **Slidesgo**, and includes icons by **Flaticon**.