Aproximación probabilística al RP

Índice

Referencia principal

- Cap.5 de "Python Data Science Handbook", sección "In Depth: Naive Bayes Classification"
- Capítulo 3 de *"Machine Learning. A probabilistic perspective"*
- Documentación de Scikit-Learn sobre Naive Bayes

Al terminar este tema tendremos las herramientas necesarias para entender nuevos métodos de ML, desde una aproximación probabilística y generativa.

1. Introducción

Hasta ahora hemos visto diferentes métodos para ajustar los parámetros de un modelo matemático con un conjunto de ejemplos etiquetado. Una vez aprendido, implementado y puesto en marcha, éste recibirá nuevos ejemplos que deberá etiquetar. Es el caso, por ej., de los reconocedores automáticos de matrícula, o del reconocedor de huella dactilar de algunos móviles.

El concepto de probabilidad sólo ha aparecido cuando se explicó la regresión logística y la función *Softmax*. La utilidad de estas funciones era modular las respuestas del clasificador de modo que su suma fuera 1, lo cual nos permitía asociar dicha respuesta con una medida de probabilidad, pero que igualmente podría ser con una medida de confianza.

Una interpretación probabilística del RP, y del ML en general, exige más profundidad pero en cambio ofrece nuevos métodos.

Proceso de aprendizaje. Es muy similar al modo que hemos aprendido clasificadores en las semanas anteriores. Consiste en un problema de optimización que devolverá los mejores parámetros del modelo que hayamos utilizado. Este aprendizaje se realiza con el conjunto de ejemplos etiquetados. La diferencia con los métodos ya vistos es que ahora el 'modelo' consiste en la densidad de probabilidad conjunta de los datos y las etiquetas $p(\mathbf{X}, \mathbf{t})$, donde $\mathbf{X}$ es el conjunto de datos y $\mathbf{t}$ el conjunto de etiquetas (es decir, se mantiene la notación que hemos llevado siempre). Para construir esa densidad conjunta utilizaremos una familia parametrizada (Bernoulli, Multinomial, Normal, Weibull, etc), o una mezcla de ellas.

Proceso de inferencia. Desde este nuevo punto de vista la etiqueta estimada (i.e. predicha o inferida) para un nuevo ejemplo $\mathbf{z}$ será aquella que maximiza dicha densidad conjunta que hemos construido, condicionada por dicho ejemplo, es decir $\hat{t} = \arg\max_t \left(p(t|\mathbf{z})\right)$.

1.1. Distribuciones de probabilidad

Una variable continua o discreta puede tomar varios valores. Cuando dicha variable es una incógnita no sabemos su valor, pero una vez determinado permanecerá invariable a no ser que cambien las condiciones que nos llevaron a él. De una **variable aleatoria** nunca podemos determinar su valor. Lo mejor que podemos saber es como se distribuyen los valores que puede tomar y que probabilidad tiene cada uno de estos. Por ejemplo lanzar una moneda al aire es una variable aleatoria que puede tomar dos valores {cara, cruz}, cada uno de ellos equiprobable e igual a 1/2. Lanzar un dado es una variable aleatoria que puede tomar seis valores {1,2,3,4,5,6}, cada uno de ellos también equiprobable e igual a 1/6. La probabilidad de que una bolita caiga en uno de los diez cubos de un tablero de Galton (ver Figura 1) ya no es uniforme. El cubo central es mucho más probable que los demás.

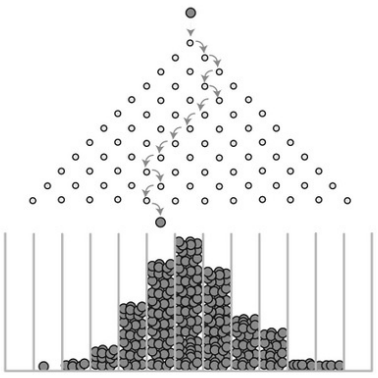

Figura 1: Tablero de Galton para mostrar la distribución binomial. [Fuente: Anónimo @ Internet]

En todos estos casos, el valor de la probabilidad de un suceso concreto es mayor que cero, y a su distribución se le denomina **función de distribución de masa de probabilidad** o, por sus siglas en inglés, **pmf**. En otras palabras, cuando la variable aleatoria x toma valores discretos, su distribución viene dada por la pmf $p(x)$, y la probabilidad de que ocurra un suceso concreto $x = x^*$ es exactamente $p(x^*)$.

¿Y si la variable aleatoria fuera continua? Si compramos un billete de lotería de navidad, la probabilidad de que ganar el Gordo es muy pequeña, 1/99999 porque hay 5 cifras en cada billete. Por tanto, si hubiera infinitas cifras, la probabilidad de ganar sería cero. Supongamos también un tablero de Galton que en vez de 10 cubos tuviera infinitos. La probabilidad de que una bolita cayese en uno también sería cero.

El concepto de probabilidad en variables continuas parece contradecir a la intuición: "si yo lanzo una bolita, en un tablero de Galton tiene que acabar cayendo en un cubo, aunque haya infinitos". Podemos intentar justificarlo diciendo que si hubiera infinitos cubos, entonces habría infinitos niveles de obstáculos sobre ellos y la bolita nunca acabaría de caer. Pero la razón real es la aparición del infinito. Además, si las variables son continuas ese infinito es *denso*, es decir que entre dos valores cualesquiera de dicha variable hay infinitos valores más. Se habla entonces de la distribución de la **densidad de probabilidad** de dicha variable aleatoria (**pdf**). La probabilidad se calcula para un intervalo de valores de la variable, y no para un valor concreto, pues sería cero.

En la Figura 2 se muestran tres pmf y tres pdf típicas. Podemos observar que la continuidad es una propiedad de las pdfs puesto que sus variables aleatorias (eje horizontal) son continuas.

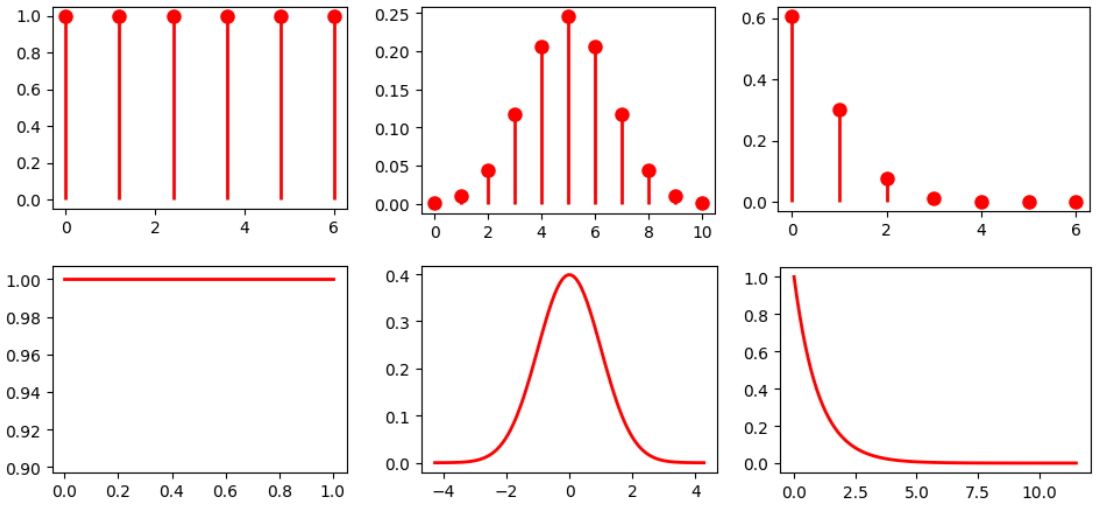

Figura 2: Arriba: PMF Uniforme, Binomial y Poisson. Abajo: PDF Uniforme, Normal o Gaussiana, Exponencial. [Fuente: Original de A. Cuesta]

La probabilidad de que puedan ocurrir varios sucesos es la suma de cada uno de ellos en el caso discreto, y la integral (es decir, el **área**) del intervalo en el caso continuo. La distribución acumulada (**cdf**) es lo que se denomina **función de distribución** de una variable aleatoria, tanto si es continua como discreta, y se suele representar con la mayúscula de la letra empleada para su masa (si era discreta) o densidad (si era continua). La siguiente tabla resume estos conceptos.

Discreta	**Continua**
pmf	**pdf**
$p(x = x^*) \geq 0$	$p(x = x^*) = 0$
cdf	
$P(a \leq x \leq b) = \sum_{x=a}^{x=b} p(x)$	$P(a \leq x \leq b) = \int_{x=a}^{x=b} p(x)$
$P(-\infty < x < +\infty) = 1$	
Ejemplos	
Uniforme	Uniforme
Binomial	Normal o Gaussiana
Poisson	Exponencial
Dirichlet	Weibull
⋮	⋮

Si consideramos la probabilidad de n variables aleatorias que suceden al mismo tiempo (conjuntas), entonces el dominio de las funciones pdf y cdf pasan a ser $n-$dimensionales, y se denominan **pdf conjunta** y **distribución conjunta** respectivamente. Por ejemplo, para $n = 2$ variables continuas distribuidas normal y conjuntamente, su densidad sería una campana de Gauss similar a la Figura 3. En este caso es el **volumen** bajo la densidad (y no el área) el que debe sumar 1. Y si hubiera más dimensiones, entonces hablaríamos del hipervolumen.

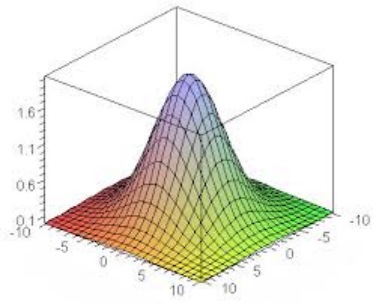

Figura 3: Densidad Normal conjunta de 2 variables. [Fuente: Anónimo @ Internet]

Se denomina **Distribución marginal** de una variable al resultado de acumular la distribución conjunta en todas las dimensiones excepto en la de dicha variable.

Por ejemplo, para 2 variables (x, y) la marginal de x,

Por tanto, al marginalizar una distribución conjunta en una de sus variables aleatorias obtenemos la distribución individual de dicha variable. El camino contrario, en general, no es posible.

$$\text{Distribución conjunta de } (x,y) = P(x,y) \begin{cases} \nearrow \text{ Distribución marginal de } x = P(x) \\ \searrow \text{ Distribución marginal de } y = P(y) \end{cases}$$

El concepto de marginal se puede generalizar a más dimensiones. Por ejemplo, si $n = 4$ variables aleatorias discretas, la marginal de (x, y) sería

$$P(x,y) = \sum_{z=-\infty}^{z=+\infty} \sum_{w=-\infty}^{w=+\infty} P(x,y,z,w).$$

En algunas técnicas de ML es habitual **marginalizar** una distribución, que consiste en añadir variables aleatorias a la distribución mediante esta técnica. Así, por ejemplo, en la expresión

anterior, hemos pasado de 2 variables a 4. Por ejemplo, es un paso necesario para la obtención de expresiones de filtrado y suavizado en Modelos Ocultos de Markov (HMM).

Construir funciones de distribución conjunta no es sencillo. Normalmente con datos multidimensionales se intenta estimar la mejor normal multivariada, que usualmente se denomina **MVN** por sus siglas en inglés (Multi-Variate Normal), es decir una "campana de Gauss" en n dimensiones; o bien a una mezcla de MVNs, como estudiaremos más adelante.

Una técnica muy importante es **condicionar** la distribución de una variable al conocimiento de otra, que se representa $p(x|y)$ y se lee "la probabilidad de x dado y". La pdf conjunta de 2 variables aleatorias x e y se puede escribir como la pdf de una de ellas multiplicada por la pdf condicionada de la otra, es decir: $p(x, y) = p(x)p(y|x) = p(y)p(x|y)$. Con más variables debemos *encadenar* las distribuciones condicionadas, por ejemplo una distribución conjunta de 4 variables podría ser: $p(x, y, z, w) = p(w)p(z|w)p(y|z, w)p(x|y, z, w)$. Más aún, esta **regla de la cadena** permite agrupar las variables de muchos otros modos, siempre que se respete el encadenamiento de las distribuciones. Por ejemplo:

$$p(x, y, z, w) = p(x, y)p(z, w|x, y) = p(x, w)p(y, z|x, w) = p(y)p(x, z, w|y) = \cdots$$

Esta técnica es muy costosa si las variables son discretas y no escala si son continuas, pero se simplifica si se hacen algunas suposiciones.

1. Si dos variables son **independientes** entonces el conocimiento de una no debería influir en la densidad de la otra, por tanto su densidad conjunta es simplemente la multiplicación de las marginales,
$$p(x_1, x_2, \ldots, x_n) = \prod_{i=1}^{n} p(x_i)$$
2. Cuando el conocimiento de una variable x convierte a otras dos v y w en independientes, se dice que éstas son **condicionalmente independientes**. Formalmente se escribe
$$p(v, w|x) = p(v|x)p(w|x)$$

1.2. Teorema de Bayes

El teorema surge directamente de aplicar la regla de la cadena a la densidad conjunta.

$$\begin{cases} p(A, B) = p(B)p(A|B) & \text{despejando se obtiene } p(A|B) = p(A, B)/p(B) \\ p(A, B) = p(A)p(B|A) & \text{que introducimos arriba, a la derecha de la igualdad} \end{cases}$$

En definitiva,

$$p(A|B) = p(B|A)p(A)/p(B) \tag{1}$$

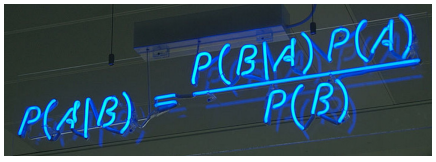

Figura 4: ¡Es tan importante que alguno se ha hecho un neón con él! [Fuente: Anónimo @ Internet]

En ML el teorema nos permite incorporar nuestra conocimiento previo o nuestra *creencia* sobre una hipótesis (etiqueta estimada), y modificarla si las evidencias (el vector de características de cada ejemplo) nos llevan a ello.

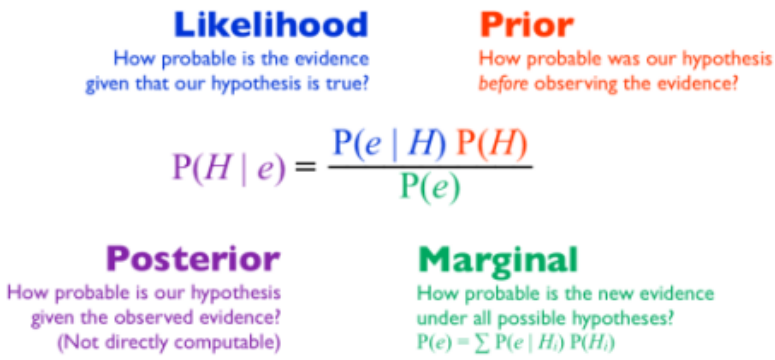

Figura 5: Relaciones en el Teorema de Bayes [Fuente: Anónimo @ Internet]

2. Clasificador Bayesiano Ingenuo

Más conocido por su nombre en inglés, *Naive Bayes*, o por las siglas NBC. Se trata de un clasificador que construye la distribución conjunta del vector de características a partir de la hipótesis de que estas son independientes entre sí para cada ejemplo etiquetado.

Método de aprendizaje. Hasta ahora utilizábamos un modelo lineal que tenía un vector de parámetros, los pesos $\mathbf{w} = (w_1, \ldots, w_n)$, y una función de coste $J(\mathbf{w}; \mathbf{X}, \mathbf{t})$. Al resolver el problema de optimización obteníamos el vector de parámetros óptimo $\mathbf{w}^*$ para dicho modelo.

De manera similar, ahora nuestro modelo será una pdf conjunta $p(\mathbf{X}, \mathbf{t}; \mathbf{w})$, que también dependerá de un vector de parámetros. Los parámetros serán óptimos cuando la probabilidad conjunta de los datos y las etiquetas sea máxima. En definitiva:

$$\mathbf{w}^* = \arg\max_{\mathbf{w}} p(\mathbf{X}, \mathbf{t}; \mathbf{w})$$

En el NBC, la pdf conjunta $p(\mathbf{X}, \mathbf{t}; \mathbf{w})$ se construye admitiendo las siguientes suposiciones:

1. **El conjunto de datos es iid** (independiente y está idénticamente distribuido).

   Es decir, se asume que cada par ejemplo-etiqueta se podría haber obtenido como resultado de muestrear una vez $p(\mathbf{X}, \mathbf{t}; \mathbf{w})$. Por tanto la probabilidad conjunta de todos los ejemplos, ya que hemos asumido que estos son independientes, es el producto de la probabilidad de cada uno de ellos,

   $$p(\mathbf{X}, \mathbf{t}) = \prod_{i=1}^{m} p(\mathbf{x}^{(i)}, t_i).$$

2. **Las características son condicionalmente independientes** dada la etiqueta.

   Es decir, si cada ejemplo es un vector de características n-dimensional, $\mathbf{x} = (x_1, \ldots, x_n)^T$, entonces

   $$p(\mathbf{x}|t) = \prod_{j=1}^{n} p(x_j|t).$$

Aplicado la regla de la cadena a la primera suposición obtenemos

$$p(\mathbf{X}, \mathbf{t}) = \prod_{i=1}^{m} p(t_i) p(\mathbf{x}^{(i)}|t_i) = \prod_{i=1}^{m} p(t_i) \prod_{i=1}^{m} p(\mathbf{x}^{(i)}|t_i).$$

Aplicando ahora la segunda suposición a cada ejemplo i en la expresión de arriba obtenemos

$$p(\mathbf{X}, \mathbf{t}) = \prod_{i=1}^{m} p(t_i) \prod_{i=1}^{m} \prod_{j=1}^{n} p(x_j^{(i)}|t_i).$$

El truco 'log-sum-exp'. En la expresión anterior, aparece un producto de términos que casi siempre son menores que 1. Por tanto la operación puede llegar a provocar errores debidos a como se codifican los números en el ordenador. Para evitarlo se puede recurrir al truco 'log-sum-exp' que transforma el producto de varios términos en la suma del logaritmo de dichos términos. Por ejemplo, si queremos calcular $v = p_1 p_2 p_3 p_4$ el truco 'log-sum-exp' funciona del siguiente modo: primero se añade un logaritmo a ambos términos de la igualdad, $\log v = w = \log(p_1 p_2 p_3 p_4)$, después se aplica la regla de logaritmos para un producto, $w = \log p_1 + \log p_2 + \log p_3 + \log p_4$. Finalmente se recupera v con la exponencial de w, es decir $v = e^w$.

Para aplicar el truco 'log-sum-exp' a la última expresión se añade el logaritmo a ambos lados, con lo que se llega a la función de coste genérica para NBC:

$$\ell(\mathbf{w}; \mathbf{X}, \mathbf{t}) = \log p(\mathbf{X}, \mathbf{t}; \mathbf{w}) = \sum_{i=1}^{m} \log p(t_i) + \sum_{i=1}^{m} \sum_{j=1}^{n} \log p(x_j^{(i)} | t_i). \tag{2}$$

El óptimo de la función (2) en el vector de parámetros $\mathbf{w}^*$. Para encontrarlo debemos derivar respecto de cada una de las dimensiones de $\mathbf{w} = (w_1, w_2 \ldots)$ e igualar a cero; es decir para cada peso w_k debemos encontrar

$$w_k^* = \arg\underset{w_k}{\text{cero}} \frac{\partial}{\partial w_k} \ell(\mathbf{w}; \mathbf{X}, \mathbf{t})$$

Pero por la construcción que hemos hecho de $\log p(\mathbf{X}, \mathbf{t}; \mathbf{w})$ podemos dividir el vector $\mathbf{w}$ en varios *subvectores* como se muestra a continuación:

$$\mathbf{w} = (\underbrace{\eta_1, \eta_2, \ldots}_{\text{parm. de } p(t)} , \overbrace{\xi_1, \xi_2, \ldots}^{\text{parm. de } p(x_1|t)} , \overbrace{\zeta_1, \zeta_2 \ldots}^{\text{parm. de } p(x_2|t)} , \ \ldots \ )$$

Es decir, algunos parámetros ($\eta_1, \eta_2, \ldots$, los que sean necesarios) modelan la distribución de las etiquetas. Otros parámetros ($\xi_1, \xi_2, \ldots$, los que sean necesarios) modelan la distribución de los valores de la primera característica, condicionada a cada uno de los valores de la etiqueta. Igualmente, otros parámetros ($\zeta_1, \zeta_2, \ldots$, los que sean necesarios) modelan la distribución de los valores de la segunda característica, condicionada a cada uno de los valores de la etiqueta. Y así sucesivamente con cada característica.

Por tanto la búsqueda de los parámetros se divide en:

$$\eta_k^* = \arg\underset{\eta_k}{\text{cero}} \frac{\partial}{\partial \eta_k} \sum_{i=1}^{m} \log p(t_i;\ \eta_1, \eta_2, \ldots) \tag{3}$$

$$\xi_k^* = \arg\underset{\xi_k}{\text{cero}} \frac{\partial}{\partial \xi_k} \sum_{i=1}^{m} \log p(x_1^{(i)} | t_i;\ \xi_1, \xi_2, \ldots) \tag{4}$$

$$\zeta_k^* = \arg\underset{\zeta_k}{\text{cero}} \frac{\partial}{\partial \zeta_k} \sum_{i=1}^{m} \log p(x_2^{(i)} | t_i;\ \zeta_1, \zeta_2, \ldots) \tag{5}$$

$$\vdots$$

En breve pasaremos a crear diferentes NBC y particularizaremos estas expresiones que ahora mismo son totalmente genéricas, y por tanto bastante abstractas y difíciles de comprender.

Método de inferencia. Tal y como hemos explicado en la introducción, en este tema la inferencia de la etiqueta para un nuevo ejemplo $\mathbf{z}$ es:

$$\hat{t} = \arg\underset{t}{\text{máx}}\, p(t|\mathbf{z}).$$

Aplicando el teorema de Bayes podemos reescribir dicho problema como

$$\hat{t} = \arg\max_t \left(\frac{p(t)p(\mathbf{z}|t)}{p(\mathbf{z})} \right) = \arg\max_t \left(p(t)p(\mathbf{z}|t) \right).$$

¿Por qué ha desaparecido el denominador en el último paso? Buscamos la etiqueta t que maximiza la fracción dentro del paréntesis, pero el denominador no depende de t y además es un valor positivo (es una probabilidad) por lo que el máximo se alcanzará en el mismo t si multiplicamos la fracción por $p(\mathbf{z})$. En otras palabras, el valor del máximo sí que cambia cuando quitamos el denominador, pero su posición, o sea t (que es lo que buscamos) NO.

Aplicando la suposición 2 (independencia condicional de las características) obtenemos:

$$\hat{t} = \arg\max_t \left(p(t) \prod_{i=1}^{n} p(z_i|t) \right)$$

Para aplicar el truco 'log-sum-exp' a la expresión anterior se añade el logaritmo a la función que queremos optimizar, es decir a la derecha de la igualdad, con lo que se llega a

$$\hat{t} = \arg\max_t \left(\log p(t|\mathbf{z}) \right) = \arg\max_t \left(\log p(t) + \sum_{i=1}^{n} \log p(z_i|t) \right) \tag{6}$$

Como se trata de un problema de optimización, y el logaritmo es una función monótona creciente, el argumento que maximiza (6) es el mismo que si no hubieramos aplicado el truco, aunque el valor del máximo que se obtenga con cada uno de ellos será diferente.

2.1. Distribuciones del clasificador

2.1.1. Conocimiento o creencia previa sobre la etiqueta

La distribución $p(t_i)$ se denomina conocimiento a priori, en inglés *prior*. Modelando esta distribución imponemos el conocimiento previo sobre cómo pensamos que están distribuidas las etiquetas. Podemos no imponer nada, haciendo que todas las etiquetas sean equiprobables. O podemos dar más probabilidad a una frente a las demás, por ejemplo contando el número de ejemplos de cada clase y dividiendo entre el total de ejemplos en el conjunto de entrenamiento.

Problema de clasificación binaria Comenzamos por el caso más sencillo de clasificación, en el que cada ejemplo i-esimo sólo puede tener una posible etiqueta $t_i = \{0, 1\}$. Para dicho ejemplo el modelo de distribución más sencillo es aquel que otorga una cierta masa de probabilidad q a la etiqueta 1 y el resto, $1 - q$ a la etiqueta 0; es decir se trata de una distribución **Bernoulli**

$$p(t_i) = \text{Ber}(t_i; q) = q^{t_i} (1-q)^{(1-t_i)} = \begin{cases} q & \text{cuando } t_i = 1 \\ 1-q & \text{cuando } t_i = 0 \end{cases}$$

Como los ejemplos son iid, el logaritmo de la distribución a priori del vector de etiquetas $\mathbf{t}$ será

$$\log p(\mathbf{t}; q) = \sum_{i=1}^{m} \log \left(q^{t_i} (1-q)^{(1-t_i)} \right) = \sum_{i=1}^{m} (t_i) \log (q) + (1-t_i) \log (1-q)$$

Esta distribución sólo tiene el parámetro q, por tanto según la expresión (3) para encontrar q^* derivamos la expresión anterior respecto de q e igualamos a cero:

$$\frac{\partial}{\partial q} \log p(\mathbf{t}; q) = \sum_{1=1}^{m} \frac{t_i}{q} + \frac{1-t_i}{1-q}(-1) = \frac{1}{q} \sum_{1=1}^{m} t_i + \frac{1}{1-q} \sum_{1=1}^{m} (t_i - 1) = 0$$

Manipulamos para que ambos términos tengan como denominador común $q(1-q)$ de modo que

$$\frac{\partial}{\partial q}\log p(\mathbf{t};q) = \frac{(1-q)\sum t_i}{q(1-q)} + \frac{(q)\sum(t_1-1)}{q(1-q)} = \frac{\sum(t_i) - (q)\sum(t_i) + (q)\sum(t_i) - (q)\sum 1}{q(1-q)} = 0$$

En esta expresión el símbolo $\sum \cdot$ representa $\sum_{i=1}^{m} \cdot$, y además:

- Como $q > 0$ (de otro modo nunca habría etiquetas "1") el denominador se puede ir al otro lado de la igualdad y, al multiplicarse por 0, desaparece.
- Los términos $-(q)\sum(t_i)$ y $+(q)\sum(t_i)$ se cancelan.
- El sumatorio $\sum 1 = \sum_{i=1}^{m} = 1 + 1 + 1 + \ldots (m \text{ veces}) \ldots + 1 = m$

Por lo que tenemos:

$$\frac{\partial}{\partial q}\log p(\mathbf{t};q) = \sum_{i=1}^{m}(t_i) - (q)m = 0;$$

y despejando llegamos a

$$q^* = \frac{\sum_{i=1}^{m}(t_i)}{m}$$

Finalmente, puesto que t_i es la etiqueta del ejemplo i-esimo, y ésta es 0 ó 1, entonces la suma de todas las etiquetas es igual al número de ejemplos con etiqueta "1", que podemos representar, por ejemplo como $|t_1|$, de modo que:

$$q^* = \frac{|t_1|}{m} \tag{7}$$

En definitiva, hemos estimado el valor del parámetro que maximiza la probabilidad a priori de la etiqueta para el problema binario o biclase. Por este motivo, al valor q^* se llama **estimador máximo-verosimil** o MLE (*maximum-likelihood estimator*) de la distribución de Bernoulli; y coincide con la idea *intuitiva* de que la probabilidad es el ratio de casos favorables entre el total de casos, que habíamos introducido al comienzo de esta subsección.

Problema Multiclase Si, en vez de dos clases, tenemos K, podemos utilizar la representación *One-hot* para codificar la etiqueta. Así el problema se modela con una distribución Bernoulli por cada una de las K etiquetas distintas que haya. La distribución a priori sobre la etiqueta del ejemplo i-esimo será entonces la distribución **Categórica** con K parámetros, $p(t_i) = \text{Cat}(t_i; q_1, \ldots, q_K)$

$$\begin{array}{cccc} t_1 & t_2 & \cdots & t_K \\ \downarrow & \downarrow & \cdots & \downarrow \\ q_1^* = \frac{|t_1|}{m} & q_2^* = \frac{|t_2|}{m} & \cdots & q_K^* = \frac{|t_K|}{m} \end{array}$$

Es importante notar que la distribución satisface que $\sum_{k=1}^{K} q_k^* = 1$, es decir que la probabilidad acumulada es 1. Y, de nuevo, este estimador máximo-verosímil es el que nos dicta la intuición.

2.1.2. Verosimilitud de los datos suponiendo una etiqueta

La distribución $p(x^{(i)}|t_i)$ se denomina verosimilitud del ejemplo i-esimo, en inglés *likelihood*. Es lógico que la probabilidad de un ejemplo, condicionada a la etiqueta, sea mayor cuando la etiqueta es la correcta; es decir será más verosimil pensar que pertenece a esa etiqueta que a ninguna otra.

Generalmente los NBC tienen todos la misma distribución a priori que ya hemos visto según el problema sea biclase o multiclase; y se diferencian en la distribución que modela la verosimilitud. En la siguiente sección construiremos varios NBC con este método.

2.1.3. Conocimiento o creencia posterior sobre la etiqueta

La distribución $p(t|z)$ en la expresión (6) se denomina distribución a posteriori de la etiqueta dado el ejemplo z, en inglés *posterior*. Esta distribución se utiliza para hacer inferencia, es decir para predecir la etiqueta de nuevos ejemplos. Puesto que buscamos la etiqueta que maximiza esta distribución, la expresión (6) se conoce "Máximo a Posteriori" o MAP.

2.2. Aprendizaje de modelos de verosimilitud

Vamos a estimar los parámetros de la distribución de verosimilitud de para datos continuos y discretos

2.2.1. NBC para características continuas

Se construye asumiendo que cada una de las características de los ejemplos de una misma clase tiene una densidad de probabilidad que se puede modelar mediante una familia paramétrica; por ej.: Gaussiana (media y desviación), Weibull (escala y forma), t-Student (grado de libertad), etc.

La suposición más habitual, y la que está implementada en Scikit-Learn, es asumir que están distribuidas normalmente, es decir según la pdf normal o Gaussiana.

Formalmente, por cada característica j y para cada ejemplo i cuya etiqueta sea t_i:

$$p(x_j^{(i)}|t_i) \sim \mathcal{N}(\mu_j, \sigma_j)$$

Como hay dos parámetros por cada característica x_j, la media μ_j y la desviación σ_j, el proceso para obtener su estimador máximo-verosímil según las expresiones (4-5) es:

$$\mu_j^* = \arg\underset{\mu_j}{\text{cero}} \frac{\partial}{\partial \mu_j} \sum_{i=1}^{m} \log \mathcal{N}(x_j^{(i)}|t_i; \mu_j, \sigma_j) \quad , \quad \text{para cada } j = 1, \ldots, n$$

$$\sigma_j^* = \arg\underset{\sigma_j}{\text{cero}} \frac{\partial}{\partial \sigma_j} \sum_{i=1}^{m} \log \mathcal{N}(x_j^{(i)}|t_i; \mu_j, \sigma_j) \quad , \quad \text{para cada } j = 1, \ldots, n$$

Vamos a obtener el estimador máximo-verosímil de la media de la característica j-esima para un problema biclase, es decir $t_i = \{0, 1\}$.

Para ello, recordamos que la densidad normal de una variable aleatoria continua x es:

$$\mathcal{N}(x; \mu, \sigma) = \frac{1}{\sqrt{2\pi}\sigma} \exp\left(-\frac{(x-\mu)^2}{2\sigma^2}\right)$$

Y su logaritmo

$$\log \mathcal{N}(x; \mu, \sigma) = \log\left(\frac{1}{\sqrt{2\pi}\sigma}\right) - \left(\frac{(x-\mu)^2}{2\sigma^2}\right)$$

Como vamos a derivar su logaritmo respecto de μ consideramos que σ es constante, de modo que podemos reescribir el logaritmo la densidad como

$$\log \mathcal{N}(x; \mu) \propto \left(-\frac{(x-\mu)^2}{2}\right)$$

Y su derivada respecto de μ es

$$\frac{\partial}{\partial \mu} \mathcal{N}(x; \mu) \propto (x - \mu)$$

Introduciendo este resultado en la expresión de μ_j^* tenemos

$$\mu_j^* = \arg\underset{\mu_j}{\text{cero}} \sum_{i=1}^{m} \left(x_j^{(i)} - \mu_j \right) = \sum_{i=1}^{m} \left(x_j^{(i)} \right) \mathbf{1}_{[t_i=1]} - \mu_j \sum_{i=1}^{m} \mathbf{1}_{[t_i=1]}$$

donde hemos introducido la función $\mathbf{1}_{[t_i=1]}$ que devuelve 1 sólo cuando se cumple la igualdad de su argumento, es decir cuando el ejemplo i-esimo tiene la etiqueta "1" y "0" en caso contrario.

En esta última expresión el término $\sum_{i=1}^{m} \mathbf{1}_{[t_i=1]}$ es un sumatorio de tantos unos como ejemplos haya con etiqueta $t = 1$, que con la notación que se introdujo anteriormente se representaría $|t|$.

En definitiva, para encontrar μ_j^* igualamos a cero la expresión obtenida y despejamos μ_j:

$$\begin{aligned}
\sum_{i=1}^{m} \left(x_j^{(i)} \right) \mathbf{1}_{[t_i=1]} - \mu_j \sum_{i=1}^{m} \mathbf{1}_{[t_i=1]} = 0 \quad , \\
\sum_{i=1}^{m} \left(x_j^{(i)} \right) \mathbf{1}_{[t_i=1]} = \mu_j \sum_{i=1}^{m} \mathbf{1}_{[t_i=1]} \quad , \\
\sum_{i=1}^{m} \left(x_j^{(i)} \right) \mathbf{1}_{[t_i=1]} = \mu_j |t| \quad , \\
\mu_j^* = \frac{\sum_{i=1}^{m} \left(x_j^{(i)} \right) \mathbf{1}_{[t_i=1]}}{|t|}
\end{aligned}$$

Es decir que el estimador máximo-verosimil del primero de los dos parámetros de la distribución normal, de los valores de la característica j-esima, de los ejemplos con etiqueta $t = 1$, es la suma dichos valores dividido por el número de ejemplos con dicha etiqueta.

Esto concuerda con nuestra idea intuitiva de cómo se calcula la media de una muestra o conjunto de datos: se suman los valores del conjunto y se divide por el número de elementos en el conjunto.

Podríamos proceder de la misma manera con el segundo parámetro y comprobaríamos que se llega al estimador máximo-verosímil de la desviación.

Si fuera un problema multiclase con K clases, al igual que con el aprendizaje del *prior*, podemos utilizar una codificación *one-hot* de manera que al final el denominador sería $|t_1|$ para la clase 1, $|t_2|$ para la clase 2 y así sucesivamente.

En resumen el proceso de aprendizaje de un NBC Gaussiano para un problema multiclase es el siguiente algoritmo:

- Para cada etiqueta t_j y para cada característica x_i:
  1. Copiar en un vector auxiliar los valores correspondientes a la etiqueta t_j y la característica x_i
  2. Calcular la media (μ_j^*) y la desviación (σ_j^*) del vector auxiliar

Si en vez de Gaussianas fueran otras pdf habría que utilizar los estimadores máximo-verosímiles de sus parámetros. Esto es un problema ya resuelto para multitud de familias de probabilidad continuas, por lo que no entraremos en más desarrollos de este tipo.

2.2.2. NBC para variables discretas

De manera similar al anterior, éste se construye eligiendo una familia paramétrica de distribución de masa de probabilidad. Vamos a comenzar suponiendo que las características son binarias y después generalizamos, como ya se hizo en la obtención del *prior*.

Comenzamos recordando que una variable aleatoria discreta y binaria es aquella que sólo puede tomar dos valores posibles $\{0, 1\}$. Por tanto su pmf será $p(x = 1)$ y $p(x = 0) = 1 - p(x = 1)$. Es decir, basta con asignar una masa de probabilidad a uno de los dos sucesos y el otro se calcula automáticamente. Por tanto dicho valor se convierte en el parámetro de la distribución, que denominaremos q. En definitiva, la distribución de Bernoulli, como ya vimos antes, es

$$\text{Ber}(x) = q^x \, (1-q)^{1-x} = \begin{cases} q & \text{si } x = 1 \\ 1-q & \text{si } x = 0 \end{cases}$$

Para utilizar esta distribución en un NBC es necesario que las características sean binarias, es decir que el conjunto de datos sea una matriz de ceros y unos. Las características continuas se pueden binarizar en el preprocesado de datos mediante una condición booleana del tipo "Si es menor de tal valor se asigna 0, y si no uno". Por ejemplo, en la Figura 6 las características x_1 y x_2 se han binarizado asignando 0 para valores inferiores a 0.5 y 1 para los demás.

x_1	x_2		x_1	x_2
0.5	0.7	$\rightarrow$	1	1
0.1	0.9	$\rightarrow$	0	1
0.2	0.3	$\rightarrow$	0	0

Figura 6: Transformación de variables continuas a binarias.[Fuente: Original de A. Cuesta]

Como ya hemos visto, el estimador máximo-verosimil q^* de una distribución Bernoulli es la proporción de ejemplos con valor 1 entre el total de ejemplos. Por tanto, para la j-esima característica del conjunto de datos, tendremos:

$$q_j^* = \frac{\sum_{i=1}^{m} \mathbf{1}_{\left[x_j^{(i)}=1\right]}}{m},$$

es decir el número de veces que la etiqueta j-esima es "uno" entre el número total de ejemplos.

Las características discretas también se pueden binarizar del mismo modo. Pero además, si el rango de valores no es muy grande, se puede crear una representación *one-hot* para cada característica. La Figura 7 muestra un ejemplo.

x_1	x_2		x_{1a}	x_{1b}	x_{1c}	x_{2a}	x_{2b}	x_{2c}
a	b	$\rightarrow$	1	0	0	0	1	0
b	a	$\rightarrow$	0	1	0	1	0	0
a	c	$\rightarrow$	1	0	0	0	0	1
c	c	$\rightarrow$	0	0	1	0	0	1

Figura 7: Transformación de variables discretas a binarias.[Fuente: Original de A. Cuesta]

Por tanto podríamos utilizar la distribución Bernoulli para cada una de ellas.

Alternativamente se puede emplear la distribución categórica. En general, si los valores de la característica j de aquellos ejemplos de la etiqueta t_i pueden tomar D valores distintos $\{1, 2, \ldots, D\}$, la distribución Categórica modela su pmf con D parámetros

$$\text{Cat}(x_j^{(i)}; q_{j1}, \ldots, q_{jD}) = \begin{cases} q_{j1} & \text{si } x_j^{(i)} = 1, \\ q_{j2} & \text{si } x_j^{(i)} = 2, \\ & \vdots \\ q_{jD} & \text{si } x_j^{(i)} = D, \end{cases}$$

El estimador máximo-verosímil de cada peso es

$$q^*_{jk} = \frac{\sum_{i=1}^{m} \mathbf{1}_{\left[x_j^{(i)}=k\right]}}{\sum_{i=1}^{m} \mathbf{1}_{\left[x_j^{(i)}=1\right]}},$$

En resumen el proceso de aprendizaje de un NBC Discreto para un problema multiclase es el siguiente algoritmo:

- Para cada etiqueta t_j y para cada característica x_i:
  1. Copiar en un vector auxiliar los valores correspondientes a la etiqueta t_j y la característica x_i
  2. $C_{ijk} \leftarrow$ número de apariciones del valor k en el vector auxiliar
  3. $N_{ij} \leftarrow$ tamaño del vector auxiliar
  4. $q^*_{jk} = C_{ijk}/N_{ij}$

2.3. Inferencia

El proceso de inferencia consiste en evaluar la distribución a posteriori de un nuevo ejemplo $\mathbf{z}$ para cada etiqueta posible y después elegir aquella más probable. Es decir:

1. Para cada etiqueta t_j calculamos $p(t_j|\mathbf{z}) = p(t_j)p(\mathbf{z}|t_j)$ y lo guardamos en un array.
2. La etiqueta asignada es $\hat{t} = \text{máx}\{t_j\}$.

2.4. Comentarios

Aproximación objetiva vs. subjetiva Los estimadores máximo-verosímiles que hemos obtenido son frecuentistas porque dependen sólo de los valores del conjunto de datos, es decir es una estimación "objetiva". Por ej. la media es la suma de los valores dividida por el número de ejemplos, o el peso de un variable discreta es el número de veces que aparece (frecuencia) dividido por el número de ejemplos.

En una aproximación "subjetiva" se puede imponer un valor inicial (a priori) sobre dichos parámetros y después corregirlo con los datos. Si ese valor inicial es muy pequeño los datos lo corregirán rápidamente, y si es muy grande tardarán más. A este conocimiento a priori se le suele denominar *belief* o creencia inicial.

En otras palabras, se trata de hacer una estimación Bayesiana de los parámetros. Esto se puede hacer dentro de un NBC pero también fuera, son dos cosas completamente independientes.

La estimación bayesiana de los parámetros de una distribución dado un conjunto de valores también se conoce como MAP porque su objetivo es encontrar el valor de dichos parámetros que maximiza su distribución a posteriori. Es decir, si $p(w)$ es la distribución a priori que asumimos para el parámetro w , y $p(\mathbf{x}|w)$ es la verosimilitud de un vector de datos $\mathbf{x}$ dado dicho parámetro, entonces $w^* = \arg\max_w p(w)(\mathbf{x}|w)$.

Categórica vs. Multinomial Frecuentemente la distribución Categórica y la Multinominal se confunden. La distribución Multinomial modela la distribución de masa de probabilidad de un conjunto discreto de casos, donde cada uno de ellos ocurre una cierto número de veces en un total de "intentos". La distribución Categórica es un caso particular de la distribución multinomial, en la que sólo hay un intento.

La distribución Multinomial se emplea habitualmente en clasificación de textos con bastantes buenos resultados y por ese motivo se ve a menudo en la literatura, a veces incluso usándolo para referirse a la distribución Categórica.

En el caso que hemos descrito para datos discretos se ha utilizado la distribución Categórica porque se ajusta más al planteamiento del problema; sólo hay 1 "intento", que consiste en el vector formado por los valores $x_j^{(i)}$.

En cualquier caso, los estimadores máximo-verosímiles de ambas se calculan exactamente igual.

Implementación en Scikit-Learn La implementación en Python incluye:

- el NBC Gaussiano para variables continuas,
- el NBC Bernoulli para variables binarias,
- el NBC Multinomial para variables discretas; y que:
  - sirve igualmente como NBC Categórico
  - tiene una estimación de parámetros que puede ser
    frecuentista ($\alpha = 0$)
    bayesiana ($\alpha \neq 0$)

3. Introducción a las Redes Bayesianas

Los modelos gráficos probabilísticos (PGM por su nombre en inglés) son un modo de codificar en un grafo la construcción de una distribución de probabilidad conjunta. Un caso particular de PGM son las redes bayesianas (BN en inglés), en el que el grafo es dirigido y no produce ciclos. Estas dos condiciones se resumen en las siglas DAG (*directed acyclic graph*)

En cada nodo o vértice del grafo de una BN se escribe una de las variables aleatorias de la distribución conjunta. Cada arista o arco tiene un nodo de salida, que se denomina "nodo padre" y uno de llegada que se denomina "nodo hijo"; de tal manera que se dan los siguientes casos:

- Una nodo A que no tiene padres codifica la distribución $p(A)$
- Un nodo B que es hijo de A codifica la distribución condicionada $p(B|A)$
- Dos o más nodos $B, C, \ldots$ hijos del mismo padre (es decir "hermanos") son condicionalmente independientes, dado el padre A, por tanto codifican la distribución conjunta $p(B|A)p(C|A)\cdots$
- Un nodo A con varios padres $B, C, \cdots$ codifica la distribución condicionada $p(A|B, C, \ldots)$

La distribución conjunta es la multiplicación de cada nodo. En la Figura 8 se muestran los casos vistos arriba y un ejemplo de BN compleja (de 5 variables).

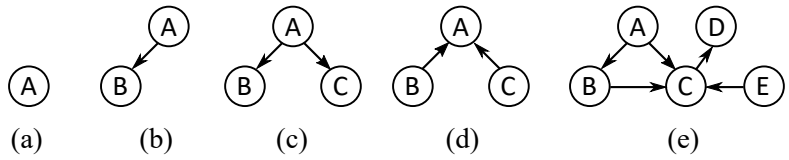

Figura 8: Representación en forma de PGM de las siguientes distribuciones de probabilidad conjunta: (a) $p(A)$, (b) $p(A)p(B|A)$, (c) $p(A)p(B|A)p(C|A)$, (d) $p(B)p(C)p(A|B,C)$, (e) $p(A)P(E)p(B|A)p(C|A,B,E)p(D|C)$ [Fuente: Original de A. Cuesta]

Las BN se suelen utilizar con datos discretos o discretizados si son continuos porque su tratamiento continuo está limitado a distribuciones normales.

El aprendizaje de las BN es complejo y computacionalmente costoso cuando el número de variables es grande debido al número de tablas que se deben construir. En la Figura 9 podemos ver un ejemplo de una BN con tres variables, de las cuales sólo una de ellas es "padre", otra es hija de esta y la última es hija de las otras dos. Se puede ver como el proceso de construcción de las tablas de cada una es más complejo cuantos más antecesores tiene el nodo.

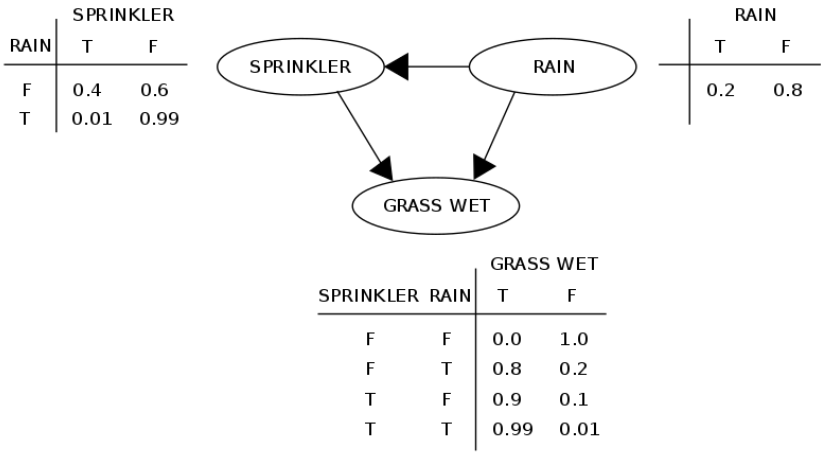

Figura 9: BN que relaciona la probabilidad de lluvia, la probabilidad de que haya funcionado el aspersor y la probabilidad de que la hierba esté húmeda. Las tablas que se aprenden son más grandes según se avanza en las dependencias entre variables. [Fuente: Original de Wikipedia]

El NBC se puede codificar en el PGM de la Figura 10

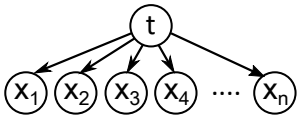

Figura 10: PGM de un NBC [Fuente: Original de A. Cuesta]

Finalmente, Scikit-Learn no implementa métodos para aprender redes bayesianas, salvo por los NBC; aunque se pueden encontrar implementaciones para Python en internet.

Índice alfabético