

TAREA 3 Parte 1

En el contexto bayesiano, si tenemos $\{x_1, x_2, \dots, x_n\} \sim p(x|\theta)$ y tenemos $p(\theta)$ la distribución inicial, entonces podemos estimar la distribución (final) de θ dada la información x_1, x_2, \dots, x_n de la siguiente manera,

$$p(\theta|x_1, x_2, \dots, x_n) \propto p(\theta)p(x_1, x_2, \dots, x_n|\theta), \quad (1)$$

donde la constante de normalización es $cte = \int p(\theta)p(x_1, x_2, \dots, x_n|\theta)d\theta$.

En el algoritmo de Naive Bayes supone que hay independencia condicional

$$p(x_1, x_2, \dots, x_n|\theta) = p(x_1|\theta)p(x_2|\theta) \cdots p(x_n|\theta),$$

y esto nos ayuda para que la distribución final de θ (en la ec(2)) se calcule a partir de

$$p(\theta|x_1, x_2, \dots, x_n) \propto \prod_{i=1}^n p(\theta)p(x_i|\theta). \quad (2)$$

Y todo muy bonito pero generar datos de una distribución en general no es un trabajo sencillo, así que se recurre a herramientas de simulación estocástica, estás a su vez están sustentadas por la teoría de procesos de Markov. La idea en general es construir un proceso de Markov que tenga como distribución estacionaria a $p(\theta|x_1, x_2, \dots, x_n)$. El algoritmo (en general) es el siguiente.

Algoritmo 1. (Metropolis-Hastings) Para generar una cadena de Markov con distribución estacionaria f . Inicializamos con x_0 , tal que $f(x_0) > 0$.

1. Dado x_t , generar $y_t \sim q(y|x_t)$
2. Generar $u \sim U(0, 1)$.
3. Hacer

$$x_{t+1} = \begin{cases} y & \text{si } u \leq \alpha(x_t, y_t) \\ x_t & \text{en otro caso.} \end{cases}$$

4. Hacer $t = t + 1$ y regresar a 1 hasta obtener convergencia.

Cómo escogemos la q ?

La propuesta original dada por Metropolis en 1953 es que q sea simétrica, es decir, que

$$q(y|x) = g(y-x),$$

donde g es una función simétrica, en este caso $\alpha(x, y) = \min\left\{1, \frac{f(y)}{f(x)}\right\}$. Es decir, la probabilidad de aceptación es independiente de q .

En la práctica se usan dos casos particulares.

1. Independencia. Cuando $q(y|x) = g(y)$.
2. Caminata Aleatoria. Cuando $q(y|x) = g(y-x)$.

Obviamente generar datos del proceso debe de ser fácil, pues de hecho en algunas ocasiones se necesitan construir mas de dos cadenas para lograr encontrar datos de una variable aleatoria. Un ejemplo se muestra a continuación.

Ejemplo 1.

Supongamos que tenemos una muestra $\{x_1, \dots, x_n\}$ tal que

$$x_1, \dots, x_m \sim P(X|\lambda) \text{ y} \\ x_{m+1}, \dots, x_n \sim P(X|\phi),$$

es decir, hay un punto de cambio en $m + 1$.

El objetivo del siguiente trabajo es estimar m , suponiendo que tanto λ como ϕ son desconocidos.

Si definimos

$$t_m = \sum_{j=1}^m x_j \text{ y } s_m = \sum_{j=m+1}^n x_j,$$

entonces la verosimilitud de (m, λ, ϕ) es

$$p(\tilde{y}|m, \lambda, \phi) \propto \lambda^{t_m} e^{-m\lambda} \phi^{s_m} e^{-(n-m)\phi}.$$

Hasta el momento sabemos que λ y ϕ son desconocidos. A partir de ahora, vamos a añadir la hipótesis de que tanto λ como ϕ y “ m ” tienen una distribución. Entonces, supongamos que

$$p(\lambda|\alpha, \beta) = Ga(\lambda|\alpha, \beta),$$

$$p(\phi|a, b) = Ga(\phi|a, b) \text{ y}$$

$$p(m) = \frac{1}{n-1} \text{ para toda } m \in \mathbb{N},$$

donde α, β, a y b son conocidos.

Se sigue entonces que

$$p(m, \lambda, \phi|\tilde{y}) \propto \lambda^{\alpha+t_m-1} e^{-\lambda(m+\beta)} \phi^{a+s_m-1} e^{-\phi(n-m+b)}.$$

Consecuentemente

$$p(\lambda|m, \tilde{y}) = Ga(\lambda|\alpha + t_m, \beta + m), \forall \lambda \in R^+,$$

$$p(\phi|m, \tilde{y}) = Ga(\phi|a + s_m, b + n - m), \forall \phi \in R^+ \text{ y}$$

$$p(m|\lambda, \phi, \tilde{y}) = \frac{\lambda^{t_m} e^{-\lambda m} \phi^{s_m} e^{-\phi(n-m)}}{\sum_{l=1}^{n-1} \lambda^{t_l} \phi^{s_l} e^{-\phi(n-l)} e^{-\lambda l}}, \forall m \in J_{n-1}.$$

De este modo, el algoritmo para estimar m queda

Algoritmo 2. Para estimar el punto de cambio m .

i) Dar $\{\alpha, \beta, a, b, n\}$ y $m_{(0)}$.

ii) Calcular $t_{m_{(i)}} = \sum_{j=1}^{m_{(i)}} x_j$ y $s_{m_{(i)}} = \sum_{j=m_{(i)}+1}^n x_j$ y

iii) Generar

$$\lambda_{(t+1)} \sim Ga(\lambda | \alpha + t_{m(t)}, \beta + m_{(t)}), \forall \lambda \in R^+,$$

$$\phi_{(t+1)} \sim Ga(\phi | a + s_{m(t)}, b + n - m_{(t)}), \forall \phi \in R^+ \text{ y}$$

$$m_{(t+1)} \sim \frac{\lambda_{t+1}^{t_{m(t)}} e^{-\lambda_{(t+1)} m_{(t)}} \phi_{(t+1)}^{s_{m(t)}} e^{-\phi_{(t+1)} (n - m_{(t)})}}{\sum_{l=1}^{n-1} \lambda_{t+1}^{t_l} \phi_{(t+1)}^{s_l} e^{-\phi_{(t+1)} (n-l)} e^{-\lambda_{(t+1)} l}}, \forall m \in \{1, 2, \dots, n-1\}.$$

iv) Repetir i) hasta alcanzar convergencia.

Y bueno, la tarea consiste en implementar el algoritmo (2).

Problema 1. *Resuelva lo siguiente.*

1. *Implementar el algoritmo 2.*
2. *Dibujar histogramas.*
3. *Dibujar trayectorias.*
4. *Dibujar como cambia los datos Poisson.*
5. *¿En donde se da el punto de cambio?*

Los datos x_i para esta tarea lo encontraran en la página del curso. Saludos, diviértanse.