



BOURBAKI

COLEGIO DE MATEMÁTICAS

Índice

01. Introducción _____ pág. 02

02. Lectura de referencia: la programación
probabilista _____ pág. 04

03. Simulación _____ pág. 05

 01. Método de monte carlo re-visitado –

 pág. 06

 02. Metropolis-Hastling _____ pág. 07

04. Modelos jerárquicos _____ pág. 09

 01. Independencia condicional _____ pág. 09

 02. Redes Bayesianas _____ pág. 10

 03. Algunas configuraciones _____ pág. 13

01 Introducción

Bienvenidos a nuestro curso de Matemáticas Avanzadas para la Ciencia de Datos, nuestro curso tiene tres módulos dedicados a estudiar las ideas matemáticas más útiles para comprender los algoritmos y modelos matemáticos más comunes en Ciencia de Datos. Los tres módulos son los siguientes

- Probabilidad y Estadística
- Álgebra Lineal
- Optimización y cálculo diferencial

Todos los módulos tienen una duración de 8 semanas. El curso está acompañado de ejercicios y tareas en Python para practicar y reforzar los conocimientos aprendidos así como las implementaciones en bases de datos de los algoritmos estudiados. Pueden consultar el repositorio de esta semana en [este link](#).

La estructura de cada una de las semanas es la siguiente:

1. Treinta minutos dedicados a estudiar un artículo de referencia que motivará los conceptos matemáticos de esta semana.
2. Dos horas dedicadas a estudiar el tema de la semana y algunos ejercicios.

3. Una hora y media dedicada a practicar lo aprendido utilizando Python.

El primer módulo de probabilidad consta de los siguientes temas:

1. Kolmogorov, independencia y condicionamiento
2. Variables aleatorias discretas y sus momentos
3. Ley de los grandes números y máxima verosimilitud
4. El teorema límite central y los intervalos de confianza
5. Tests estadísticos
6. Método de monte carlo para cadenas de markov
7. Inferencia bayesiana
8. Simulación y redes bayesianas

02 Lectura de referencia: la programación probabilista

La inferencia bayesiana ha motivado algunos cambios profundos incluso dentro de las bases de la computación. Recordando a Turing cuya teoría inspiró la creación de las computadoras tal y como las conocemos actualmente, una característica fundamental de las máquinas de Turing es que ellas son deterministas. Esto algunas veces podría ir en contra del razonamiento inteligente que hacemos los seres humanos.

Gracias a esta observación ha nacido el concepto de programación probabilista la cual combina ideas como la recursión o los condicionales provenientes de la teoría clásica de la computación con la simulación probabilista. Recientemente esta técnica ha sido implementada en distintos lenguajes de programación incluyendo Python.

La lectura sugerida de esta semana es ([1, 2]) el cual habla sobre la extensión de la teoría de la computación de turing cuando incluimos simulaciones estocásticas.

03 Simulación

En la semana anterior introducimos los conceptos básicos de la inferencia bayesiana, en el caso de las regresiones lineales bayesianas la ecuación se veía de la siguiente manera:

$$\mathbb{P}(\beta|X, Y) = \frac{\mathbb{P}(Y|\beta, X) \cdot \mathbb{P}(\beta|X)}{\mathbb{P}(Y|X)}$$

Por el momento nos concentraremos en el significado de $\mathbb{P}(Y|X)$ al cuál llamaremos nuestra evidencia del modelo. Vale la pena notar que esta cantidad no depende en lo absoluto de los modelos que haya elegido β . De hecho esta cantidad no había sido necesario considerarla pues la semana pasada resolvimos el problema de la inferencia bayesiana utilizando maximum a posteriori o conjugate priors los cuales evitan el cálculo de la evidencia.

Example 00.1. *En el ejemplo en el que deseamos modelar el precio Y de una familia de casas utilizando algunas de sus características X , esta cantidad el tipo de casas que estamos considerando, por ejemplo si son casas muy costosas o no. Es una probabilidad sobre nuestras bases de datos.*

A menos que tengamos acceso a todos valores $X = x, Y = y$ (es decir distintas bases de datos), esta cantidad es imposible de calcular. Debemos de pensar en X como distintas hipótesis, no sobre el modelo sino sobre

nuestras observaciones. Para hacer este cálculo un poco más sencillo lo que normalmente se hace es integrar sobre la familia de parámetros β :

$$\mathbb{P}(Y|X) = \int_u \mathbb{P}(Y|\beta=u, X) \mathbb{P}(\beta=u|X) du$$

Para el caso particular cuando β sea una distribución discreta obtenemos:

$$\mathbb{P}(Y|X) = \sum_u \mathbb{P}(Y|\beta=u, X) \mathbb{P}(\beta=u, X)$$

Exercise 00.2. Escribir la ecuación anterior cuando β solo tenga dos valores posibles: β y β^c .

Nuevamente el problema es que la integral $\int_u \mathbb{P}(Y|\beta=u, X) \mathbb{P}(\beta=u|X) du$ depende de nuestros parámetros del modelo β el cual podría ser muy grande, para poder resolver este tipo de problemas es necesario utilizar el método de monte carlo.

Método de monte carlo re–visitado

Recordemos que deseamos aproximar la distribución $\mathbb{P}(\beta|Y)$.

Sea $X_1, X_2, \dots, X_t, \dots$ una cadena de markov con valores en el espacio de parámetros de los modelos posibles β . Sea $M(\beta_i, \beta_j)$ su matriz de transición.

Supongamos además que nuestra cadena de markov satisface las hipótesis del teorema de la semana 6, es decir que converge a una distribución estacionaria π . El objetivo principal del método de monte carlo para la inferencia bayesiana es lograr que $\pi = \mathbb{P}(\beta|Y)$.

Metropolis-Hastling

A continuación proponemos un algoritmo para construir una cadena de markov tal que $\pi = \mathbb{P}(\beta|Y)$.

Para cada β_i en el espacio de parámetros de β , definimos una variable aleatoria Z_{t,β_i} gaussiana $\mathbf{N}(\beta_i, \sigma^2)$ y denotamos por $\alpha_t(\beta_i, \beta_j) = \mathbb{P}(Z = \beta_j | Z = \beta_i)$.

A esta distribución la llamaremos nuestra distribución propuesta para Metropolis-Hastling y debemos considerarla como un hiper-parámetro.

En el examen final vamos a utilizar una distribución distinta.

Definition 02.1. Utilizando la notación anterior, el algoritmo de Metropolis-Hastling para inferencia bayesiana consiste en lo siguiente:

1. Elegir uniformemente al azar un β_0 en el espacio de mis parámetros.
2. En el tiempo $t + 1$, si supongo que tengo acceso a un elemento β_t , voy a muestrear un $\hat{\beta}$ utilizando $\alpha_{t+1}(\beta_t, \hat{\beta})$.
3. Calcularé $\hat{\alpha} = \min\{1, \frac{\mathbb{P}(\hat{\beta}|Y)}{\mathbb{P}(\beta_t|Y)}\}$
4. Con probabilidad $\hat{\alpha}$ haremos $\beta_{t+1} = \hat{\beta}$, sino $\beta_{t+1} = \beta_t$

5. Continuar para t suficientemente grande, esto induce una distribución sobre el espacio de parámetros de β .

Theorem 02.1. *Utilizando la notación anterior, β_t es una cadena de markov y si tiene una distribución estacionaria entonces ella coincide con $\mathbb{P}(\beta|Y)$.*

04 Modelos jerárquicos

Para terminar con el contenido del curso hablaremos sobre otras generalizaciones además de las cadenas de markov para los procesos independientes e idénticamente distribuidos.

Independencia condicional

Antes de introducir en toda su generalidad a las redes bayesianas será necesario introducir un concepto técnico que nos ayudará a hablar con más fluidez de estos objetos.

Sean X, Y dos variables aleatorias cuya probabilidad conjunta nos interesa. Si ellas fueran absolutamente independientes entonces sería fácil representar su probabilidad conjunta de la siguiente forma:

$$\mathbb{P}(X, Y) = \mathbb{P}(X) \cdot \mathbb{P}(Y)$$

Desafortunadamente tal y como ya lo hemos estudiado, no es común pensar que ellas son independientes. Podrían sin embargo ser independientes condicionalmente.

Definition 01.1. Sean X, Y, Z tres variables aleatorias, diremos que X, Y son

independientes condicionadas a Z cuando

$$\mathbb{P}(X, Y|Z) = \mathbb{P}(X|Z) \cdot \mathbb{P}(Y|Z)$$

Example 01.1. Si X_1, \dots, X_t, \dots es una cadena de markov, entonces X_{t+1} y X_{t-1}, \dots, X_1 son independientes condicionadas a X_t .

Redes Bayesianas

Las aplicaciones de la estadística en campos como la bioinformática, o el procesamiento del lenguaje y de imágenes a menudo involucran modelos a gran escala en los que millones de variables interactúan de formas complejas. Los modelos jerárquicos proporcionan una metodología general para abordar estos problemas.

Un caso particular son las redes bayesianas los cuales a su vez generalizan a los modelos Naïve Bayes y Markov de los que hablamos anteriormente y son uno de los modelos de Machine Learning que arrojan resultados más transparentes.

02.1 Teoría de Grafos

En esta sección introduciremos los objetos que utilizaremos para describir los modelos de las redes bayesianas.

Definition 02.1. Sea $V = \{v_1, v_2, \dots, v_d\}$ un conjunto al que llamaremos los vértices de nuestro grafo G . Las aristas del grafo es un conjunto de parejas ordenadas entre los vértices. En este curso nos interesamos únicamente por los grafos dirigidos, es decir tal que las aristas (v_i, v_j) son distintas entre sí y además sin cilos, es decir que no existe una familia de aristas

$$(v_{i_1}, v_{i_2}), (v_{i_2}, v_{i_3}), (v_{i_3}, v_{i_4}), \dots, (v_{i_n}, v_{i_1})$$

02.2 Redes bayesianas

Los nodos en una red bayesiana representan variables explicativas y las aristas representan dependencias entre las variables. Las dependencias se calcularán utilizando probabilidades condicionales para cada vértice y los vértices que llegan a él.

Definition 02.2. Sea S una base de datos y G un grafo como en la sección anterior, una red bayesiana es un modelo estadístico asociado a la ecuación que vamos a definir a continuación.

Para cada vértice v_i , le llamaremos los antecedentes y denotaremos por I_i al conjunto de vértices tales que existe una arista entre ellos y v_i .

$$\mathbb{P}(v_1, \dots, v_d) = \mathbb{P}_S(v_1|I_1) \cdot \mathbb{P}_S(v_2|I_2) \cdot \dots \cdot \mathbb{P}_S(v_d|I_d)$$

Example 02.1. Sea G un grafo con 5 vértices como en el dibujo anterior, la

ecuación de la red bayesiana corresponde a

$$\mathbb{P}(v_1, v_2, v_3, v_4, v_5) = \mathbb{P}_S(v_1) \cdot \mathbb{P}_S(v_2|v_1) \cdot \mathbb{P}_S(v_3|v_1) \cdot \mathbb{P}_S(v_4|v_2, v_3) \cdot \mathbb{P}_S(v_5|v_4)$$

Remark 02.2. *Los modelos de clasificación de una red bayesiana corresponden al caso cuando las distribuciones son sobre el espacio de variables explicativas X junto a la variable dependiente Y .*

Independencia condicional

Ahora vamos a definir una noción fundamental que definirá determina las aristas de nuestros grafos en una red bayesiana.

Definition 02.3. Sea S una base de datos y $v_{i_1}, v_{i_2}, v_{i_3}$ tres variables. Diremos que condicionado a v_{i_3} las variables v_{i_1} son independientes cuando se satisfaga la siguiente ecuación:

$$\mathbb{P}(v_{i_1}, v_{i_2}|v_{i_3}) = \mathbb{P}(v_{i_1}|v_{i_3}) \cdot \mathbb{P}(v_{i_2}|v_{i_3})$$

Example 02.3. *Utilizando la imagen anterior, supongamos que nuestras tres variables son "estación de año", "mojado" y "lluvia", entonces:*

- *Condicionadas a estar mojado, las otras dos variables no son independientes.*

- *Condicionadas a la lluvia, las otras dos variables son independientes.*
- *Condicionadas a la estación del año, las otras dos variables no son independientes.*

Exercise 02.4. Utilizando las variables "estación de año", "aspersor", "lluvia", "mojado" y "resbaloso" argumentar cuáles parejas de variables son independientes condicionadas a alguna anterior.

Algunas configuraciones

Definition 03.1. Sean X, Y, Z tres variables aleatorias, diremos que ellas son una cadena cuando

$$\mathbb{P}(X, Y, Z) = \mathbb{P}(X)\mathbb{P}(Y|X)\mathbb{P}(Z|Y)$$

. En este caso el grafo acíclico correspondiente es $(v_X, v_Z), (v_Z, v_X)$.

Exercise 03.1. En este caso X, Y son independientes condicionadas a Z

Definition 03.2. Sean X, Y, Z tres variables aleatorias, diremos que ellas son un fork cuando

$$\mathbb{P}(X, Y, Z) = \mathbb{P}(X)\mathbb{P}(Y|X)\mathbb{P}(Z|X)$$

En este caso el grafo acíclico correspondiente es $(v_X, v_Z), (v_X, v_Y)$.

Exercise 03.2. *En este caso Y, Z son independientes condicionadas a X .*

Definition 03.3. Sean X, Y, Z tres variables aleatorias, diremos que ellas son una V-estructura cuando

$$\mathbb{P}(X, Y, Z) = \mathbb{P}(X)\mathbb{P}(Y)\mathbb{P}(Z|X, Y)$$

En este caso el grafo acíclico correspondiente es $(v_X, v_Z), (v_Y, v_Z)$.

Exercise 03.3. *En este caso X, Y son independientes y de hecho no lo son condicionadas a Z .*

Bibliografía

- [1] Turing's Legacy: Developments from Turing's Ideas in Logic. Lecture Notes in Logic. Cambridge University Press, 2014.
- [2] Cameron E. Freer, Daniel M. Roy, and Joshua B. Tenenbaum. Towards common-sense reasoning via conditional simulation: legacies of Turing in Artificial Intelligence. In Turing's Legacy, pages 195–252. Cambridge University Press, dec 2014.



escuela-bourbaki.com