Clasificación supervisada y no-supervisada

Escuela de Matemáticas Bourbaki

Febrero 2020, CDMX

1 Introducción

Estas notas son la bitácora de aprendizaje del Bootcamp Clasificación supervisada y no-supervisada que dimos en la Ciudad de México. El objetivo de este curso es estudiar las matemáticas y la implementación en 4 casos concretos de los siguientes algoritmos:

1. Perceptrón lineal
2. Árboles de decisión
3. K-nearest neighbours
4. 2-means

En la penúltima sección del curso se introduce el enfoque Bayesiano con el fin de invitar a los alumnos a estudiar aquellos métodos.

Al final de las notas se incluye un apéndice sobre probabilidad relevante para algunos pasos en el capítulo sobre el enfoque Bayesiano.

El curso tuvo una duración de 27 horas, 15 de ellas se dedicaron a estudiar el contenido de estas notas.

1.1 Hipótesis

A lo largo del curso supondremos que estamos en un problema clásico de aprendizaje supervisado (esta última hipótesis solo se olvidará en la sección de 2-means) tipo clasificación binaria en $\mathbb{R}^d$, donde buscamos entrenar a nuestro modelo con un conjunto de ejemplos

$$S = \{(x_1, y_1), \ldots, (x_N, y_N)\}$$

cuando $x_i \in \mathbb{R}^d, y_i \in \{-1, +1\}$.

2 Perceptrón

El perceptrón es una red neuronal de una sola capa y tantos nodos como cualidades tengan nuestros ejemplos.

Para aligerar la notación, en esta sección supondremos que la última coordanada de todos nuestras observaciones x_i es igual a uno i.e. $x_i = (x_{i,1}, x_{i,2}, \ldots, x_{i,d-1}, 1)$. Hacer esta suposición siempre es posible si en lugar de considerar nuestras observaciones en $\mathbb{R}^d$ lo hacemos en el hiperplano $X_{d+1} = 1$ en $\mathbb{R}^{d+1}$.

Haremos una suposición llamada de realizabilidad que puede parecer exagerada sin embargo será necesaria, en la sección **??** hablaremos más sobre por qué esta hipótesis no siempre es exagerada. Los

puntos de S son separables por una función lineal, eso quiere decir que existe un hiperplano $H \subseteq \mathbb{R}^d$ tal que de un lado de este hiperplano están todos los puntos clasificados con $+1$ y del otro lado estarán todos aquellos clasificados con -1. Formalmente la hipótesis de separabilidad significa lo siguiente:

Existe un vector $\beta^* \in \mathbb{R}^d$ de tal forma que siempre que $y_i = 1$ entonces $\langle x_i, \beta^* \rangle > 0$ y si $y_i = -1$ entonces $\langle x_i, \beta^* \rangle < 0$ (el caso en el que algún ejemplo x_i safisfaga la igualdad corresponde al hecho geométrico de pertenecer al plano definido por β^*). La justificación entre la idea intuitiva y la definición anterior es gracias al Teorema del valor Medio.

Suponiendo separabilidad, sea $\beta \in \mathbb{R}^d$ los parámetros del hiperplano que separa a nuestros ejemplos. Definamos la función $sign_{\beta^*} : \mathbb{R}^d \to \{-1, 1\}$ tal que $x \mapsto 1$ si $\langle x, \beta^* \rangle > 0$ y $x \mapsto -1$ si $\langle x, \beta^* \rangle < 0$. Utilizando esta función nuestra hipótesis de separabilidad se puede escribir de la siguiente manera $sign_{\beta^*}(x_i) = y_i$ para cualquier $i \leq N$. Eso implica que para cualquier $i \leq N$ se tiene $y_i \cdot sign_{\beta^*}(x_i) > 0$.

El algoritmo del perceptrón busca descubrir los valores de β^* únicamente conociendo S, se define inductivamente de la siguiente manera:

1. Comenzamos con $\beta_0 = (1, 1, \ldots, 1)$.
2. Si suponemos que β_i está definido, buscamos algún ejemplo $(x_j, y_j) \in S$ tal que lo siguiente NO sea cierto: $y_j \cdot sign_{\beta_i}(x_j) > 0$.
3. Si no encontramos algún índice j que cumpla los criterios de nuestra búsqueda entonces β_i es nuestra propuesta para clasificar puntos entrenada con S. Si por el contrario encontramos algún índice j que cumpla los criterios de nuestra búsqueda entonces actualizamos a $\beta_{i+1} := \beta_i + y_j x_j$.
4. Volvemos al paso dos.

Definamos $R = \max_{i \leq N} \langle x_i, x_i \rangle$ y $B = \min_{\beta \in \mathbb{R}^d} \{\langle \beta, \beta \rangle : y_1 \cdot \langle x_1, \beta \rangle > 0, \ldots, y_N \cdot \langle x_N, \beta \rangle > 0\}$.

El resultado positivo es el siguiente:

Theorem 2.1. *Bajo la hipótesis de separabilidad sobre S el algoritmo del perceptrón se detiene (es decir encuentra alguna β que clasifica correctamente a los ejemplos en S) en $(RB)^2$-pasos.*

Desafortunadamente la geometría del problema algunas veces implica que tiempo en el que el percetrón se detiene es demasiado grande:

Exercise 2.2. *Encontrar un ejemplo de puntos S que sean linealmente separables sin embargo $B \geq c^d$ para alguna constante c.*

3 Árboles de decisión

Algunas veces la información que buscamos clasificar no será separable linealmente, los árboles de decisión tienen la ventaja de poder clasificar ese tipo de información.

Definition 3.1. Sea $S = \{(x_i, y_i) : y_i \in \{\pm 1\}, x_i \in \{0, 1\}^d, i \leq N\}$, un árbol de decisión de tamaño l es una familia de subconjuntos $S_k^s \subseteq S$ donde $k \leq l, s \in \{0, 1\}$ y existe un $j \leq d$ tal que

1. $(x_i, y_i) \in S_k^0$ si y solo si $x_{i,j} = 0$
2. $(x_i, y_i) \in S_{k+1}^1$ si y solo si $x_{i,j} = 1$
3. Para cualquier $2 \leq k \leq l, s \in \{0, 1\}$, $S_k^0 \cup S_{k+1}^1 = S_{k-1}^s$.

El conjunto de los S_k^s se le llamará el conjunto de los nodos y al índice j se le llamará cualidad de división entre los nodos $k-1$ y $k, k+1$. La rama del árbol hasta el nodo S_k^s es el conjunto de todos los nodos anteriores a S_k^s.

Dado un árbol de decisión $S^0_{l-1}, S^1_l, S^1_{l-2}, \ldots, S^0_1, S^1_1 \subseteq S$ es posible definir un orden en el conjunto de índices $J = (j_1, j_2, \ldots, j_d)$ de la siguiente manera:

- j aparecerá antes que j' si y solamente si las longuitudes mínimas k, k' de las cadenas $S^s_k \subseteq S^{s'}_{k-1} \ldots \subseteq S$, $S^t_{k'} \subseteq S^{t'}_{k-1} \ldots \subseteq S$ sobre el conjunto de cadenas tal que j y j' son la cualidad de división entre los nodos $k, k-1$ y $k', k'-1$ respectivamente, satisfacen $k \leq k'$.

Al árbol de decisión junto a una elección del orden de sus índices lo llamaremos un árbol de decisión ordenado.

Exercise 3.1. *Si fijamos un orden $J = (j_1, \ldots, j_d)$ es posible construir un árbol de decisión de manera única cuyo orden tal que $J_S = J$.*

Una vez que tengamos un árbol de decisión ordenado es posible hacer una predicción mediante el siguiente algoritmo:
Sea $S^0_{l-1}, S^1_l, S^1_{l-2}, \ldots, S^0_1, S^1_1 \subseteq S$, $(k_1, \ldots, k_d)$ un árbol de decisión ordenado.

1. Dado un punto $x = (x_1, \ldots x_d) \in \{0,1\}^d$ buscamos predecir su clasificación $(x, \pm 1)$.
2. Caminaremos a lo largo del árbol de acuerdo al orden elegido de la siguiente manera:

   Si $x_{k_1} = 0$ visitamos el nodo S^0_1, de lo contrario visitamos el nodo S^1_2. Después si $x_{k_2} = 0$ visitamos el nodo S^0_3, de lo contrario visitamos el nodo S^1_4... Finalmente si $x_{k_d} = 0$ visitamos el nodo S^0_{l-1}, de lo contrario visitamos el nodo S^1_l. Le llamaremos hoja al último nodo visitado.
3. Si la mayoría de los puntos en la hoja están clasificados como 1 entonces predecimos $(x, 1)$, de lo contrario predecimos $(x, -1)$.

Ahora que sabemos qué es un árbol de decisión y cómo pueden ser utilizados para clasificar información, vamos a concentrarnos en idear algoritmos que nos permitan construir un árbol de decisión ordenado de manera eficaz. El primer intento es el llamado algoritmo de Hunt:

Definition 3.2. (Algoritmo de Hunt) Sea $S = \{(x_i, y_i) : y_i \in \{\pm 1\}, x_i \in \{0,1\}^d, i \leq N\}$. Construiremos un árbol de decisión ordenado inductivamente, en el paso k construimos a S^s_k de la siguiente forma:

- Si todos los puntos en S^s_k están clasificados de la misma forma entonces nos detenemos.
- Si existe algún punto en S^s_k mal clasificado entonces elegimos algún índice distinto a las cualidades de división anteriormente elegidas en esta misma rama del árbol (a este conjunto lo denotaremos como J_k), digamos j y separamos al conjunto de puntos en S^s_k de acuerdo a la coordenada j.

Exercise 3.2. *Escribir formalmente qué significa que exista un ejemplo mal clasificado en el nodo S^s_k*

Notemos que este algoritmo no incluye ninguna pista sobre cómo elegir al índice j. Si la información en S no es redundante entonces este algoritmo es teóricamente adecuado, sin embargo más adelante explicaremos sus desventajas.

Corollary 3.3. *Si existe una función f tal que $f(x_i) = y_i$ para todo $i \leq N$ entonces el algoritmo de Hunt se detiene eventualmente (quizás en 2^d-pasos).*

Ahora hablaremos del concepto de entropía el cual nos da una manera útil de elegir el índice j.

Definition 3.3. Sea $S^s_k = \{(x_i, y_i) : y_i \in \{\pm 1\}, x_i \in \{0,1\}^d, i \leq N_k\}$ un nodo en cierto árbol de decisión fijo, definimos las siguientes cantidades:

1. $p_{k,0} = \frac{N_{k,0}}{N_k}, p_{k,1} = \frac{N_{k,1}}{N_k}$ y $N_{k,0} = |S^0_{k+1}|, N_{k,1} = |S^1_{k+2}|$.

2. $Ent\left(S_k^s\right) = p_{k,0} log_2\left(p_{k,0}\right) + p_{k,1} log_2\left(p_{k,1}\right)$
3. $Gin\left(S_k^s\right) = 2p_{k,0} \cdot p_{k,1}$.
4. $Err\left(S_k^s\right) = \underset{i \leq 2}{min} p_{k,i}$.
5. Si $C\left(S_k^s\right)$ es alguna de las tres cantidades anteriores, definimos la ganancia entre los nodos S_k^s y S_{k+1}^0, S_{k+2}^1 de la siguiente manera: $Gain\left(S_k^s\right) = C\left(S_k^s\right) - p_{k,0} \cdot C\left(S_{k+1}^0\right) - p_{k,1} \cdot C\left(S_{k+2}^1\right)$. Es importante notar que todos estos cálculos dependen de árbol construido.

Exercise 3.4. *Converserce de que $p_{k,1} = 1 - p_{k,0}$.*

Definition 3.4. (Algoritmo ID3) Utilizando la notación anterior definimos la primera cláusula igual a la primera cláusula en el algoritmo de Hunt, la segunda es como siguie:

- Supongamos construido el árbol hasta el nodo S_k^s. Elegimos al índice j de la siguiente manera: calculamos $Gain\left(S_k\right)$ para los (a lo más d-posibles índices y elegimos aquel j que maximice esa cantidad.

4 K-nearest neighbours

El último algoritmo de clasificación supervisada que estudiaremos es K-nearest neighbours, es un algoritmo simple que aprovecha cuando la dimensión del problema es suficientemente baja.

Un corto recordatorio sobre la distancia euclidiana:

Definition 4.1. Sean $x = (x_1, \ldots, x_d), x' = (x'_1, \ldots, x'_d) \in \mathbb{R}^d$ dos puntos cualesquiera, definimos su distancia euclideana de la siguiente manera: $d\left(x, x'\right) = \sqrt{\left(x_1 - x'_1\right)^2 + \ldots + \left(x_d - x'_d\right)^2}$

Sea $S = \{(x_1, y_1), \ldots, (x_N, y_N)\}$ nuestra base de datos, recordemos que $x_i \in \mathbb{R}^d, y_i \in \{-1, +1\}$.

Definition 4.2. (1-NN) Sea $x \in \mathbb{R}^d$,

1. Calculemos las siguientes distancias: $d\left(x, x_1\right) =: d_1, \ldots d\left(x, x_N\right) =: d_N$
2. Calculamos y llamaremos d_x al menor de los números $d_1, \ldots d_N$, sea $d_x = d_i$.
3. Suponiendo que $(x_i, y_i) \in S$, predecimos (x, y).

Exercise 4.1. *Dibujar las fronteras de clasificación para $S = \{(1, 0, -1), (-1, 0, -1), (0, 1, 1), (0, -1, 1)\}$*

Exercise 4.2. *Si por un momento definimos $d\left(x, x'\right) = |x_1 - x'_1| + \ldots + |x_d - x'_d|$, dibujar las fronteras de clasificación de 1-NN.*

Definition 4.3. (K-NN) Sea $x \in \mathbb{R}^d$ y $K \leq N$

1. Calculemos las siguientes distancias: $d\left(x, x_1\right) =: d_1, \ldots d\left(x, x_N\right) =: d_N$
2. Ordenamos las distancias de menor a mayor: $d_{i_1} \leq d_{i_2} \leq \ldots \leq d_{i_K} \leq \ldots \leq d_{iN}$.
3. Definimos a y como aquella última coordenada que más se repita en los siguientes vectores en S: $(x_{i_1}, y_{i_1}), \ldots, (x_{i_K}, y_{i_K})$.
4. Suponiendo que $(x_i, y_i) \in S$, predecimos (x, y).

Exercise 4.3. *Si $K = N$ a qué corresponde nuestro algoritmo K-NN?*

Exercise 4.4. *Cuáles son las desventajas de un K cercano a 1? Cuáles son las desventajas de un K cercano a N?*

Exercise 4.5. *Qué significa información redundante para K-NN?*

Exercise 4.6. *Qué dificultad computacional podría encontrar K-NN?*

Exercise 4.7. *Qué dificultad en términos de generalización podría encontrar K-NN?*

5 2-means

El primer algoritmo no-supervisado que estudiaremos es el llamado K- means, este algoritmo lo utilizaremos también para resolver un problema de clasificación binaria.

Es fundamental notar que en los algoritmos anteriores se hacía intenso uso de la información que las coordenadas y_i de los puntos $(x_i, y_i) \in S$ nos daban. Esta vez debemos prescindir de esa información, es decir: sea $S = \{x_1, \ldots, x_N\}$ con $x_i \in \mathbb{R}^d$, buscamos partir a nuestra base de datos en dos subconjuntos disjuntos: $S_{+1}, S_{-1} \subseteq S$, $S_{+1} \cap S_{-1} = \emptyset$.

Definition 5.1. Sea $S = \{x_1, \ldots, x_N\}$ con $x_i \in \mathbb{R}^d$ y $S' \subseteq S$ un subconjunto arbitrario de S,

1. Definimos el centroide de S' de la siguiente forma: $\mu_{S'} := \underset{\mu \in \mathbb{R}^d, x_i \in S}{argmin} \, d(\mu_S, x_i)^2$

Definition 5.2. (2-Means) Sea $S = \{x_1, \ldots, x_N\}$ con $x_i \in \mathbb{R}^d$,

1. Comenzemos con cualquier par de centroides $\mu_1^0, \mu_2^0 \in \mathbb{R}^d$ (de algún par de subconjuntos en S), definimos
$$S_1^0 = \{x \in S : 1 = \underset{i \leq 2}{argmin} \, (d(\mu_i, x))\}$$
$$S_2^0 = \{x \in S : 2 = \underset{i \leq 2}{argmin} \, (d(\mu_i, x))\}$$
2. Una vez encontrados S_1^0 y S_2^0 debemos actualizar los centroides:
$$\mu_1^1 := \frac{1}{|S_1^0|} \sum_{x \in S_1^0} x$$
$$\mu_2^1 := \frac{1}{|S_2^0|} \sum_{x \in S_2^0} x$$
.
3. Repetir inductivamente para encontrar $S_1^i, S_2^i, \mu_1^i, \mu_2^i$.

Exercise 5.1. *Sea $S = \{1, 2, 3, 4\} \subseteq \mathbb{R}$, aplicar el algoritmo de 2-means.*

6 Examen

- Dado un conjunto S relacionado con un problema de aprendizaje supervisado, una función de pérdida relativa a un algoritmo es una función que mide el error que comete el algoritmo en el conjunto S. Para los algoritmos vistos en clase definir la función de pérdida adecuada.
- Describir el concepto de error para el algoritmo 2-means.

7 Acercamiento Bayesiano

Hasta este momento hemos considerado un acercamiento llamado "funcional" a los problemas de clasificación, sin embargo existe otro acercamiento llamado Bayesiano. En el acercamiento funcional damos prioridad a la búsqueda de los parámetros que definen la mejor función clasificadora. Por otro lado, es posible buscar esa función de manera indirecta mendiante la ley de distribución que genera nuestros ejemplos. Aquí debemos hacer una pausa fundamental para pensar qué siginifica la ley de distribución en términos de nuestro problema en Ciencia de Datos. Hasta ahora no hemos hecho ninguna suposición sobre la naturaleza de nuestro conjunto S, sin embargo para un estudio matemático formal esta hipótesis será fundamental.

7.1 1-Nearest Neighbour v.s. clasificador óptimo de Bayes

El acercamiento Bayesiano consiste en aproximar de la mejor manera la función f utilizando una base de datos S, la ventaja de conocer esa distribución es que podemos definir fácilmente una función clasificadora de la siguiente manera:

Definition 7.1. Sea D una ley de probabilidad sobre el conjunto $\mathbb{R}^d \times \{-1, +1\}$. Definimos al clasificador óptimo de Bayes como la función $h_D : \mathbb{R}^d \to \{-1, +1\}$ tal que:

1. $h_D(x) = 1$ si $\mathbb{P}_D(y = 1|x) \geq \frac{1}{2}$ y
2. $h_D(x) = -1$ en el caso contrario.

La demostración formal del siguiente enunciado está fuera de los objetivos del curso sin embargo es posible demostrar lo siguiente:

- Si uno conoce la ley de distribución D que sigue cierta base de datos S y suponemos que la distribución es arbitraria, entonces la función h_S anterior llamada el clasificador óptimo de Bayes es la mejor clasificación posible a un problema de predicción.

Antes de continuar vamos a introducir una clase de funciones súmamente importantes no solo en matemáticas sino también en nuestro contexto de Ciencia de Datos.

Definition 7.2. Sea $f : \mathbb{R}^n \to \mathbb{R}^m$ y $\nu > 0$, decimos que f es una función ν-Lipschitz cuando para $x, y \in \mathbb{R}^n$ se tiene:

$$||f(x) - f(y)||_2 \leq \nu ||x - y||_2$$

La intuición detrás de estas funciones para un Científico de Datos es la siguiente:

Sea D la distribución de la cuál el conjunto es un conjunto de ejemplos, eso significa que D es una ley de probabilidad sobre el conjunto $\mathbb{R}^d \times \{-1, +1\}$.

Sean X, Y variables aleatorias en $\mathbb{R}^d$ y $\{-1, +1\}$ cuyas distribuciones coinciden con las distribuciones marginales de D.

Definamos a la función f de la siguiente manera:

$$f(x) = \mathbb{P}_D(Y = 1|X)$$

Exercise 7.1. *¿Qué repercución tiene en Ciencia de Datos que la función f sea ν-Lipschitz?*

El siguiente resultado es un teorema matemático que permite garantizar la excelencia del algoritmo 1-nearest neighbour. Antes de eso es necesario definir la función de error de una función clasificadora:

Definition 7.3. Sea D una ley de probabilidad arbitraria sobre el conjunto $[0, 1]^d \times \{-1, +1\}$ y $h : [0, 1]^d \to \{-1, +1\}$ una función.

- Definimos el error de h relativo en un punto $(x, y) \in [0, 1]^d \times \{-1, +1\}$ como

  $$l(h, (x, y)) = 0$$

  si $h(x) = y$ y

  $$l(h, (x, y)) = 1$$

  si $h(x) \neq y$.

- Definimos el error de h relativo a la distribución D como

  $$err_D(h) = \mathbb{E}_{(x,y)\sim D}[l(h, (x, y))]$$

Theorem 7.2. *Sea D una ley de probabilidad arbitraria sobre el conjunto $[0,1]^d \times \{-1,+1\}$ y supongamos que f está definida como en el párrafo anterior y es ν-Lipschitz. Sea S un conjunto de N-experimentos con respecto a la distribución D y h_S la función clasificadora entrenada con el conjunto S, entonces:*

$$\mathbb{E}\left[err_D\left(h_S\right)\right] \leq err_D\left(h_D\right) + 4\nu\sqrt{d}N^{-\frac{1}{d+1}}$$

Este teorema concluye que el algoritmo K-NN en algunas circunstanias es tan bueno como el clasificador óptimo de Bayes, el cual a su vez ya mencionamos sin justificarlo es el mejor clasificador posible. Dos comentarios en este momento son pertinentes, es probable que las circunstancias de las que hablamos anteriormente no se cumplan y por tanto K-NN no es ideal. El segundo comentario tiene que ver con la imposibilidad virtual de tener a la mano la ley de distribución y por tanto el clasificador óptimo de bayes es en realidad una entelequia. En la siguiente sección vamos a hablar brevemente del llamado Muestro de Gibbs, el cuál es un método para poder aproximar aquellas distribuciones.

7.2 Muestro de Gibbs

Supongamos que tenemos un vector $x = (x_1, \ldots, x_d) \in \mathbb{R}^d$, como ya dijimos, el enfoque Bayesiano consiste en buscar la ley de distribución que rige a nuestros ejemplos x (quzás para después utilizar el clasificador óptimo de Bayes). Llamaremos a la distribución desconocida D sobre $\mathbb{R}^d$.

El muestreo de Gibbs consiste en aproximar la ley de distribución D utilizando las leyes de distribución condicionales $\mathbb{P}_D\left(x_i | x_1, \ldots, x_{i-1}, x_{i+1}, \ldots, x_d\right)$ que supondremos más simples de calcular.

Antes de adentrarnos en el algoritmo de Gibbs es necesario recordar la Ley de los Grandes números (descrita en el apéndice) que nos dice que para aproximar la esperanza de una variable aleatoria es suficiente con realizar experimentos (variables aleatorias) independientes respecto a la ley de distribución una cantidad suficientemente grande de veces, a este proceso le llamaremos muestreo. Si buscamos algo más concreto recomendamos buscar librerías en Python que son capaces de generar números pseudo-aleatorios con respecto a una distribución dada. Este tema requiere cuidado y esfuerzo.

Definition 7.4. Supongamos que tenemos un vector aleatorio $x = (x_1, x_2, x_3) \in \mathbb{R}^3$ con respecto a alguna distribución D. El algoritmo de Gibbs para D se define inductivamente de la siguiente manera:

- Supongamos que tenemos un muestreo fijo hasta el tiempo t (x_1^t, x_2^t, x_3^t).
- Actualizamos la primera cordenada mediante un muestreo respecto a la ley de probabilidad $\mathbb{P}_D\left(x_1 | x_2^t, x_3^t\right)$ y obtenemos algún x_1^{t+1}.
- Actualizamos la segunda cordenada mediante un muestreo respecto a la ley de probabilidad $\mathbb{P}_D\left(x_2 | x_1^{t+1}, x_3^t\right)$ y obtenemos algún x_2^{t+1}.
- Actualizamos la tercera cordenada mediante un muestreo respecto a la ley de probabilidad $\mathbb{P}_D\left(x_3 | x_1^{t+1}, x_2^{t+1}\right)$ y obtenemos algún x_1^{t+1}.
- Volvemos a comenzar, esta vez con el vector $\left(x_1^{t+1}, x_2^{t+1}, x_3^{t+1}\right)$.

Exercise 7.3. *Describir el algoritmo para dimensión 2 y 4 así como dar una interpretación geométrica en dimensión 2.*

Exercise 7.4. *Pensar en ejemplos en Ciencia de Datos cuando sea más fácil calcular las leyes de probabilidad condicionales.*

La esperanza es que el algoritmo de Gibbs eventualmente converga, existen razones matemáticas para pensar que esto podría suceder sin embargo aquella descripción está fuera del alcance de este curso pues es necesario el estudio de Cadenas de Markov entre otras cosas.

8 Apéndice: Elementos de Probabilidad

8.1 Espacios de Probabilidad

Definition 8.1. *Si Σ es un conjunto y $A, B \subseteq \Sigma$ son dos subconjuntos, denotaremos por:*

1. *$A \cup B$ a la unión entre A y B.*
2. *$A \cap B$ a la intersección entre A y B.*
3. *A^c al complemento de A.*

Definition 8.2. *Fijemos un conjunto finito Σ, una ley de probabilidad es una asignación numérica $\mathbb{P}$ a cada uno de los subconjuntos de Σ tal que si A, B son dos subconjuntos:*

1. $0 \leq \mathbb{P}(A) \leq 1 = \mathbb{P}(\Sigma)$
2. $\mathbb{P}(A^c) = 1 - \mathbb{P}(A)$
3. $\mathbb{P}(A \cup B) = \mathbb{P}(A) + \mathbb{P}(B)$, *si* $A \cap B = \emptyset$.

Example 8.3. *Ley uniforme: supongamos que Σ es un conjunto de tamano m. Definamos la siguiente ley de probabilidad $\mathbb{P}_{unif}(\sigma_1, \ldots, \sigma_i) = \frac{i}{m}$.*

Example 8.4. *Bernoulli: supongamos que Σ es un conjunto de tamano 2 i.e. $\Sigma = \{\sigma_1, \sigma_2\}$. Sea $0 \leq p \leq 1$ un número arbitrario. Definimos $\mathbb{P}_{Bernoulli}(\sigma_1) = 1 - p$ y $\mathbb{P}_{Bernoulli}(\sigma_2) = p$.*

Example 8.5. *Σ un conjunto de tamano n i.e. $\Sigma = \{\sigma_1, \sigma_2, \ldots, \sigma_n\}$, sea $0 \leq p \leq 1$ un número arbitrario. Definimos la ley de probabilidad siguiente¿ $\mathbb{P}_{Bin,p,b}(\sigma_i) = \left(\frac{n!}{(n-i)!i!}\right) p^i (1-p)^{n-i}$. Esta ley de probabilidad es llamada binomial con parámetros n, p. Esta ley generaliza a la ley de Bernouilli.*

8.2 Independencia

Definition 8.1. Sea $(\Sigma, \mathbb{P})$ una ley de probabilidad sobre un conjunto finito. Sean $A, B \subseteq \Sigma$ son dos eventos aleatorios, diremos que ellos son independientes si $\mathbb{P}(A)\mathbb{P}(B) = \mathbb{P}(A \cap B)$.

Exercise 8.6. *En la experiencia aleatoria de tirar dos dados, los eventos:*

- *Obtener un uno en el primer lanzamiento y un dos en el segundo son independientes.*
- *Obtener un número uno en el primer lanzamiento y que la suma de los dos dados sea mayor a ocho no son independientes.*

Exercise 8.7. *Si A, B son dos eventos independientes, cómo son los eventos A^c, B^c? Idependientes o dependientes?*

Definition 8.8. *Sea $(\Sigma, \mathbb{P})$ una ley de probabilidad (sobre un conjunto finito o infinito numerable). Sean $A, B \subseteq \Sigma$ son dos eventos aleatorios tales que $\mathbb{P}(B) > 0$. Definimos una nueva ley de probabilidad $\mathbb{P}_B(A) = \mathbb{P}\frac{(A \cap B)}{\mathbb{P}(B)}$ lo cual se lee como la ley de probabilidad condicionada al evento B.*

Exercise 8.9. *Si $\mathbb{P}$ es una ley de probabilidad y A, B son dos eventos aleatorios, qué número podría ser más pequeño? $\mathbb{P}_A(B)$ o $\mathbb{P}(B)$? Cuándo serán iguales?*

Exercise 8.10. *Supongamos que $(\Sigma, \mathbb{P})$ una ley de probabilidad sobre un conjunto finito y $B \subseteq \Sigma$ es un evento tal que $\mathbb{P}(B) > 0$. Demostrar que la asignación $\mathbb{P}_B(A)$ es una ley de probabilidad sobre el conjunto Σ.*

Exercise 8.11. *Si $\Sigma = \{1, 2, \ldots, 36\}$ y $\mathbb{P}_{Unif}$ es la ley uniforme sobre un conjunto de tamaño 36 (recuerden que esto corresponde a la experiencia aleatoria de tirar 2 dados justos), si $B = \{1, 2, 3, 4, 5, 6\}$, determinar los valores de $\mathbb{P}_B$.*

Theorem 8.12. *(Fórmula de Bayes). Supongamos que $(\Sigma, \mathbb{P})$ una ley de probabilidad sobre un conjunto finito y A, B dos eventos tales que $\mathbb{P}(B), \mathbb{P}(A) > 0$. Entonces $\mathbb{P}_A(B) = \frac{\mathbb{P}_B(A) \cdot \mathbb{P}(B)}{\mathbb{P}(A)}$.*

Exercise 8.13. *Deducir la fórmula de Bayes.*

Exercise 8.14. *Clasifiquemos los administradores de portafolios financieros en dos grupos, los bien informados y los mal informados. Supongamos que sabemos que con probabilidad igual a .8, si un administrador bien informado elige un activo financiero para su cliente, entonces el activo subirá de precio. Por el contrario la probabilidad de que eso ocurra si el administrador está mal informado es de .6. Supongamos que el 10 porciento de los administradores financieros están bien informados. La pregunta es la siguiente: un cliente elige al azar a un administrador y le pide que compre un activo financiero, suponiendo que el activo financiero subió, cuál es la probabilidad de que el administrador esté mal informado?*

Proof. Si M es el evento: el valor del activo sube e I es el evento: el administrador está bien informado. Gracias a la proposición **??**:

$$\mathbb{P}(M) = \mathbb{P}_I(M) \cdot \mathbb{P}(I) + \mathbb{P}_{I^c}(M) \cdot \mathbb{P}(I^c).$$

Utilizemos la fórmula de Bayes para concluir. □

8.3 Iteraciones de experiencias aleatorias

Supongamos que queremos repetir una misma experiencia aleatoria más de una vez, una primera idea para hacerlo es suponer que queremos que las experiencias sucesivas no dependan de las anteriores, para ello es necesario introducir el concepto de ley de probabilidad producto:

Definition 8.15. *Sean $(\Sigma, \mathbb{P}_\Sigma), (\Pi, \mathbb{P}_\Pi)$ dos leyes de probabilidad sobre Σ, Π dos conjuntos finitos. Definimos la ley de probabilidad producto de la siguiente manera: $(\Sigma \times \Pi, \mathbb{P}_{\Sigma\times\Pi})$ donde $\mathbb{P}_{\Sigma\times\Pi}(A \times B) = \mathbb{P}_\Sigma(A) \cdot \mathbb{P}_\Pi(B)$.*

Exercise 8.16.

1. *Calcular la ley de probabilidad producto de dos leyes de Bernouilli sobre el mismo conjunto (una con parámetro $\frac{1}{4}$ y la otra con un parámetro $\frac{1}{3}$ por ejemplo).*
2. *Calcular la ley de probabilidad de dos leyes uniformes sobre dos conjuntos finitos de tamanos distintos.*

Remark 8.17. *Si Σ, Π son dos conjuntos finitos entonces el producto de las leyes de probabilidad es única con respecto a la propiedad $\mathbb{P}_{\Sigma\times\Pi}(A \times B) = \mathbb{P}_\Sigma(A) \cdot \mathbb{P}_\Pi(B)$.*

Es posible construir otra ley de probabilidad sobre el producto de dos espacios que no ser'/a un producto de probabilidades.

Exercise 8.18. *Supongamos que $\Sigma = \{0, 1\}$, por tanto $\Sigma^2 = \{(0,0), (0,1), (1,0), (1,1)\}$. Definamos la ley de probabilidad $\mathbb{P}(0,1) = \frac{1}{2}, \mathbb{P}(1,0)$. Convencerse de que esta ley de probabilidad no es el producto de dos leyes de probabilidad.*

8.4 Variables aleatorias

La teoría moderna de probabilidades se basa en el manejo de las variables aleatorias trataremos de convencerlos para con numerosos ejemplos de sus ventajas prácticas.

Definition 8.2. Sea $(\Sigma, \mathbb{P})$ una ley de probabilidad. Sea $\mathbb{R}$ el conjunto de todos los números. Una variable aleatoria (real) es una función $X : \Sigma \to \mathbb{R}$.

Exercise 8.19. *Si $X : \Sigma \to \mathbb{R}$ es una variable aleatoria, demostrar que es posible definir una ley de probabilidad sobre $\mathbb{R}$ de la siguiente forma, para cada $A \subseteq \mathbb{R}$, $\mathbb{P}_X(A) = \mathbb{P}\left(X^{-1}(A)\right)$. Si $x \in \mathbb{R}$.*

Remark 8.20. *Si X es una variable aleatoria denotaremos por $Im(X)$ al conjunto de valores $x \in \mathbb{R}$ tales que existe algún $\sigma \in \Sigma$ con $X(\sigma) = x$ y por $p_x = \mathbb{P}_X(x)$.*

Exercise 8.21. *Si Σ es el espacio de la experiencia aleatoria de lanzar dos dados justos, definimos la siguiente variable aleatoria: $X(i,j) = i + j$. Demostrar que $\mathbb{P}_X(2) = \mathbb{P}((1,1))$, $\mathbb{P}_X(3) = \mathbb{P}(\{(1,2),(2,1)\})$ y $\mathbb{P}_X(4) = \mathbb{P}(\{(1,3),(3,1),(2,2)\})$.*

Remark 8.22. *El ejemplo anterior demuestra que aunque el tamano del conjunto imagen de una variable aleatoria no sea igual que el del conjunto Σ (porqué?), la ley de probabilidad sí se preserva. En particular la gráfica de una ley de probabilidad se ve muy distinta a una gráfica de la ley asociada a una variable aleatoria.*

Example 8.23. *Si E es la experiencia aleatoria de lanzar un dado justo hasta obtener el número seis, definimos la siguiente variable aleatoria: $X(\omega_1, \omega_2, \ldots) = \underset{j \geq 1}{inf}\{j : \omega_j = 6\}$*

Exercise 8.24. *En el ejemplo anterior calcular la probabilidad $\mathbb{P}_X(j)$. Hint: demostrar que $X^{-1}(j) = \{\omega \in \Omega : \omega_1 \neq 6, \omega_2 \neq 6, \ldots, \omega_{j-1} \neq 6, \omega_j = 6\}$.*

Exercise 8.25. *Definamos el conjunto $\Sigma = \{0,1\}^n$ y $\mathbb{P}_{Bernoulli,n}$ el producto de las leyes de probabilidad de Bernoulli. Para un $1 \leq k \leq$ y cada $\sigma = (\sigma_1, \ldots, \sigma_n) \in \Sigma$ definimos $X_k(\sigma) = \sigma_k$. Calcular $\mathbb{P}_{X_k}(1)$.*

Exercise 8.26. *Sean n y X_k como en el ejemplo anterior y $m \leq n$, definamos una variable aleatoria de la siguiente manera: $S_m = \sum_{k \leq m} X_k$. Notemos que esta nueva variable aleatoria toma valores en el conjunto $\{1, 2, \ldots, m\}$. Demostrar que para $1 \leq i \leq m$ $\mathbb{P}_{S_m}(i) = \mathbb{P}_{Bin,m,p}(i)$.*

Example 8.27. *Sea $X(i) = i$, si $(\Sigma, \mathbb{P}_{Poisson})$ es la ley de probabilidad de Poisson con parámetro lambda sobre $\Sigma = \{\sigma_1, \sigma_2, \ldots\}$, $\mathbb{P}_X(i) = e^{-\lambda}\frac{\lambda^i}{i!}$.*

8.5 Esperanza de las variables aleatorias

Definition 8.28. *Si $X : \Sigma \to \mathbb{R}$ es una variable aleatoria con Σ un conjunto finito o numerable y $\mathbb{P}$ una ley de probabilidad determinada por $p_i = \mathbb{P}(\sigma_i)$, definimos la esperanza de X de la siguiente forma:*

$$\mathbb{E}[X] = \sum_{\sigma_i \in \Sigma} p_i X(\sigma_i)$$

Exercise 8.29. *Definir una variable aleatoria que corresponda a las 10 calificaciones durante un semestre, todas ellas entre 5 y 10 de un estudiante. Calcular su esperanza.*

Proposition 8.30. *Si X es una variable aleatoria, entonces*

$$\mathbb{E}(X) = \sum_{x \in Im(X)} x \cdot p_x.$$

Proof. Por definición $\mathbb{E}(X) = \sum_{\sigma_i \in \Sigma} p_i X(\sigma_i) = \sum_{x \in Im(X)} \left(\sum_{\sigma_i : X(\sigma_i) = x} p_i \cdot x \right) = \sum_{x \in Im(X)} x \cdot p_x.$ □

Exercise 8.31. *Calcular la esperanza de las siguientes variables aleatorias:*

- $X_{i,j}$ *del ejercicio 8.21 utilizando la definición y después la proposición anterior,*
- X_k *del ejercicio 8.25*
- S_m *del ejercicio 8.26.*

Exercise 8.32. *Sunpongamos que un número entre el 1 y el 10 es eligido al azar. Debemos adivinarlo únicamente haciendo preguntas cuya respuesta sea sí o no. Vamos a considerar dos casos distintos:*

1. *Supongamos que las preguntas que hacemos son: 'el número es 1? el número es 2?, etc...' en ese orden. Sea* $\Sigma = \{1, \ldots, 10\}^2$ *con la siguiente ley de probabilidad* $p_{k,k} = \frac{1}{10}$ *y* $p_{i,j} = 0$ *siempre que* $i \neq j$*, para cualesquiera* $1 \leq i, j, k \leq 10$*. La ley de probabilidad* $p_{i,j}$ *corresponde a la siguiente experiencia aleatoria: elegimos un número al azar entre el 1 y el 10 y hacemos las preguntas descritas anteriormente, con qué probabilidad haremos* j *preguntas para adivinar un número aleatorio entre* $1 \leq j \leq 10$*. Definamos* $X(i,j) = j$ *entonces* $\mathbb{P}_X(k) = \frac{1}{10}$*. Calcular la esperanza de* X*.*

2. *Supongamos que ahora vamos a hacer las siguientes preguntas: el número es* ≤ 5*? Después preguntaremos: el número es* ≤ 2*? (si la respuesta fue sí en la pregunta anterior) o el número es* ≤ 7 *(si la respuesta fue no en la pregunta anterior). Después preguntaremos: el número es* ≤ 1*? o el número es* ≤ 3*? o el número es* ≤ 6*? o el número es* ≤ 8*? Dependiendo del resultado en la pregunta anterior. Después preguntamos: el número es* ≤ 4*? o el número es* ≤ 9*? Esto significa que necesitamos entre tres y cuantro preguntas para adivinar el número. Definir el espacio de probabilidad adecuado así como la variable aleatoria que represente el número de preguntas necesarias para adiviar un número. Por último calcular la esperanza.*

Si $X, Y : \Sigma \to \mathbb{R}$ dos variables aleatorias, definimos la variable aleatoria $Z = (X, Y)$ como la función $Z(x,y) = (X(x), Y(y))$.

Definition 8.33. *Sean* $X, Y : \Sigma \to \mathbb{R}$ *dos variables aleatorias, decimos que ellas son independientes si para todo* $A \subseteq Im(X), B \subseteq Im(Y)$*,* $\mathbb{P}_Z(A \times B) = \mathbb{P}_X(A) \cdot \mathbb{P}_Y(B)$*.*

Theorem 8.34. *(Ley de los grandes números) Sean* X_n *es una familia de variables aleatorias independientes e identicamente distribuidas entonces*

$$\lim_{n \to \infty} \frac{(X_1 + \ldots + X_n)}{n} = \mathbb{E}(X_1).$$