Monte Carlo Simulation: Methods and Applications

Presentado por: Laura Díaz Sánchez

Supervisado por Dr. Emilio Carrizosa Priego



FACULTAD DE MATEMÁTICAS Departamento de Estadística e Investigación Operativa Sevilla, Junio 2021

Agradecimientos

En primer lugar, me gustaría dar las gracias a mis padres por el apoyo incondicional que me han dado siempre, por no soltarme nunca de la mano y por confiar en mí más que yo misma. Soy quién soy gracias a vosotros. No olvidéis nunca que sois mi motor.

En segundo lugar, me gustaría agradecer a mi pareja todo lo que ha hecho por mí, por levantarme cada vez que me he caído y por darme siempre fuerzas para seguir.

Al profesorado que me ha acompañado durante este camino, que no sólo siente pasión por esta ciencia, sino también por transmitirla.

A mis compañeros y amigos con los que he compartido buenos y malos momentos, ha sido un verdadero placer compartir esta etapa con vosotros.

Por último, pero no menos importante, a mi tutor, Emilio, por estar ahí siempre que lo he necesitado, por implicarte tanto, por enseñarme una infinidad de cosas durante este trayecto y por hacer que aumente aún más mi pasión por las matemáticas.

Índice general

Re	esum	en	7
Al	ostra	$\operatorname{\mathbf{ct}}$	9
1.	Intr	oducción	11
2.	Gen	eración de variables aleatorias	12
	2.1.	Simulación uniforme	12
	2.2.	La transformación inversa	15
	2.3.	Métodos de transformación generales	16
		2.3.1. Distribuciones discretas	17
		2.3.2. Representaciones de mixturas	18
	2.4.	Método Acepto-Rechazo	19
3.	Inte	egración Monte Carlo	23
	3.1.	Introducción	23
	3.2.	Muestreo por importancia	25
		3.2.1. Remuestreo por importancia	28
		3.2.2. Selección de la función de importancia	29
4.	Esta	adística Bayesiana	33
	4.1.	Muestreo de Gibbs	34
5.	Apl	icación de los métodos Monte Carlo a la optimización	

	estocástica	37
	5.1. El método SAA	40
6.	Estimación del valor de Shapley	44
	6.1. Conceptos generales	45
Bi	ibliografía	61

Resumen

En el ámbito de las matemáticas y la ingeniería aparecen frecuentemente integrales multidimensionales que son difíciles de resolver. Se puede solventar este problema proporcionando estimadores puntuales e intervalos de confianza para el valor determinado de la integral usando técnicas de muestreo.

En este trabajo, estudiaremos el *método Monte Carlo* básico y algunas variantes, como el *muestreo por importancia*, que lo hacen más eficiente.

Además, ilustramos la técnica con diversas aplicaciones en estadística bayesiana, optimización estocástica y teoría de juegos.

Abstract

Complex multidimensional integral calculus can be found often in the field of mathematics and engineering. This issue can be solved providing point estimates and confidence intervals to the particular value of the integral by using sampling techniques.

In the presented academic work, the basic *Monte Carlo method*, as well as some of its variants that make it more efficient, such as the *Importance Sampling*, has been investigated.

Moreover, the technique has been illustrated with diverse applications to Bayesian statistics, stochastic optimization and game theory.

Capítulo 1

Introducción

En matemáticas e ingeniería aparecen integrales multidimensionales y sumas que son difíciles de resolver. Se puede solventar este problema proporcionando estimadores puntuales e intervalos de confianza para el valor determinado de la integral o suma usando técnicas de muestreo.

En este trabajo, se estudian el método Monte Carlo básico y algunas de sus variantes que lo hacen más eficiente, además de algunas de sus diversas aplicaciones.

Los métodos Monte Carlo principalmente dependen de la posibilidad de generar variables aleatorias de distribuciones nuevas o conocidas, por lo que en primer lugar se desarrollarán técnicas que generan variables aleatorias en el Capítulo 2.

En el Capítulo 3 se introducirá el problema que se quiere resolver junto con el uso de los métodos Monte Carlo para solventar este problema y algunas de sus variantes, como el *muestreo por importancia*.

En el Capítulo 4 se mostrará la aplicación del método Monte Carlo en la estadística bayesiana. Además, se desarrollará una técnica muy conocida en este ámbito, el *muestreo de Gibbs*.

En el Capítulo 5 se verá la aplicación de los métodos Monte Carlo en la optimización estocástica junto con un método muy conocido para abordar este tipo de problemas, el método SAA.

Por último, en el Capítulo 6, se desarrollará la aplicación de los métodos Monte Carlo en teoría de juegos, en particular, en la estimación del valor de Shapley.

Capítulo 2

Generación de variables aleatorias

En este capítulo, basado en el artículo [1], se van a desarrollar técnicas que generan variables aleatorias. En todos los casos, los métodos dependen de la disponibilidad de generaciones de muestras independientes idénticamente distribuidas a una variable uniforme, proporcionadas por el comando de R, runif.

Los *métodos Monte Carlo* principalmente dependen de la posibilidad de producir variables aleatorias de distribuciones nuevas o conocidas.

2.1. Simulación uniforme

El generador de uniformes en R es la función runif, cuya entrada es el número de valores que se quiere generar.

Todos los métodos que se van a ver, incluido runif, producen números pseudo-aleatorios, basados en un valor inicial u_0 de una muestra de una uniforme $\mathcal{U}(0,1)$ y una transformación D. El generador de uniformes produce una sucesión $(u_i) = (D^i(u_0))$ de valores en (0,1), pero el resultado tiene las mismas propiedades estadísticas que una muestra aleatoria iid.

Runif tiene implementado un algoritmo, llamado $algoritmo\ MT$, que se describe a continuación.

Algoritmo MT

Se denotan por \mathbf{x}_i vectores w-dimensionales sobre el espacio $\mathbb{F}_2 = \{0, 1\}$.

El algoritmo MT genera una sucesión de estos vectores, considerados números pseudoaleatorios entre 0 y $2^w - 1$. Dividiendo por este factor, se consideran los vectores números en el intervalo [0, 1].

El algoritmo está basado en la siguiente recurrencia lineal

$$\mathbf{x}_{k+n} := \mathbf{x}_{k+m} \oplus (\mathbf{x}_k^u | \mathbf{x}_{k+1}^l) A, \quad k = 0, 1, \dots$$

donde

- n: grado de la recurrencia
- $0 \le r \le w 1$
- $1 \le m \le n$
- A: matriz $w \times w$ con entradas en \mathbb{F}_2
- \mathbf{x}_k^u : vector formado por los primeros w-r bits de \mathbf{x}_k
- \mathbf{x}_{k+1}^l : vector formado por los últimos r bits de \mathbf{x}_{k+1}
- $(\mathbf{x}_k^u | \mathbf{x}_{k+1}^l)$: concatenación de los dos vectores anteriores en ese orden
- ⊕: suma binaria

Pseudocódigo algoritmo MT

Paso 1 Se toman $\mathbf{x}_0, \mathbf{x}_1, \ldots, \mathbf{x}_{n-1}$ iniciales.

Paso 2 Tomando $k=0,1,2,\ldots$, usando la recurrencia, se generan $\mathbf{x}_n,\,\mathbf{x}_{n+1},\,\mathbf{x}_{n+2},\,\ldots$

Paso 3 Se obtienen \mathbf{x}_k^u , \mathbf{x}_{k+1}^l .

Paso 4 Se calcula $(\mathbf{x}_k^u \mid \mathbf{x}_{k+1}^l)$.

Paso 5 Se realiza la siguiente multiplicación $(\mathbf{x}_k^u \mid \mathbf{x}_{k+1}^l) A$.

Paso 6 Se realiza la suma binaria de \mathbf{x}_{k+m} con el vector anterior y se genera el vector \mathbf{x}_{k+n} .

Se elige la matriz A de forma que la multiplicación $(\mathbf{x}_k^u \mid \mathbf{x}_{k+1}^l) A$ sea rápida de realizar. En el artículo [8] se propone como matriz A la siguiente

$$A = \begin{pmatrix} 1 & & & & \\ & & 1 & & \\ & & & \ddots & \\ & & & & 1 \\ a_{w-1} & a_{w-2} & \cdots & \cdots & a_0 \end{pmatrix},$$

luego el cálculo de $\mathbf{x}A$ se puede realizar usando sólo operaciones binarias:

$$\mathbf{x}A = \begin{cases} \text{Mover hacia la derecha los bits de } \mathbf{x} & \text{si } x_0 = 0 \\ \text{Mover hacia la derecha los bits de } \mathbf{x} \oplus \mathbf{a} & \text{si } x_0 = 1 \end{cases}$$

donde
$$\mathbf{a} = (a_{w-1}, a_{w-2}, \dots, a_0), \mathbf{x} = (x_{w-1}, x_{w-2}, \dots, x_0).$$

Ejemplo 2.1.1

Fijemos n=2 y w=2. Por tanto, $0 \le r \le 1$ y $1 \le m \le 2$. Tomamos m=1, r=1.

Elegimos como matriz A la siguiente

$$A = \left(\begin{array}{cc} 0 & 1\\ 1 & 0 \end{array}\right).$$

 \rightarrow Calculemos \mathbf{x}_2 .

Tomamos $\mathbf{x}_0 = (1 \, 1), \, \mathbf{x}_1 = (0 \, 1).$

Usando la recurrencia para k = 0 se tiene que

$$\mathbf{x}_2 := \mathbf{x}_1 \oplus (\mathbf{x}_0^u \,|\, \mathbf{x}_1^l) \, A$$

Cálculos

$$\mathbf{x}_0^u = (1) \\
\mathbf{x}_1^l = (1)$$

$$\Rightarrow (\mathbf{x}_0^u \mid \mathbf{x}_1^l) = (11)$$

$$(11) \cdot A = (11)$$

$$\mathbf{x}_1 \oplus (11) = (01) \oplus (11) = (10)$$

$$\Longrightarrow \mathbf{x}_2 = (10)$$

\rightarrow Calculemos \mathbf{x}_3 .

Usando la recurrencia para k = 1 se tiene que

$$\mathbf{x}_3 := \mathbf{x}_2 \oplus (\mathbf{x}_1^u \,|\, \mathbf{x}_2^l) \, A$$

Cálculos

$$\mathbf{x}_{1}^{u} = (0) \\ \mathbf{x}_{2}^{l} = (0) \end{cases} \Longrightarrow (\mathbf{x}_{1}^{u} | \mathbf{x}_{2}^{l}) = (0 0)$$

$$(0 0) \cdot A = (0 0)$$

$$\mathbf{x}_{2} \oplus (0 0) = (1 0) \oplus (0 0) = (1 0)$$

$$\Longrightarrow \mathbf{x}_{3} = (1 0)$$

Para más información sobre el algoritmo MT se puede consultar el artículo [8].

La generación de variables aleatorias uniformes es fundamental en el comportamiento de los métodos de simulación para otras distribuciones de probabilidad, ya que sus distribuciones pueden ser representadas como una transformación de variables aleatorias uniformes.

2.2. La transformación inversa

Existe una transformación muy común conocida como transformación integral de la probabilidad, que permite transformar cualquier variable aleatoria en una variable aleatoria uniforme y viceversa.

Proposición 2.2.1 Sea X una variable aleatoria absolutamente continua con fdd f y FdD F invertible, y sea $U \sim \mathcal{U}(0,1)$, entonces $F^{-1}(U) \sim X$.

Demostración

$$P[F^{-1}(U) \le x] = P[U \le F(x)] \stackrel{U \sim \mathcal{U}(0,1)}{=} F(x) = P[X \le x]$$

Ejemplo 2.2.2

En este ejemplo se va a simular la distribución exponencial usando la transformación inversa, y se va a comprobar la eficacia de este método dibujando el histograma resultante.

Sea $X \sim \mathcal{E}xp(1)$, entonces $F(x) = 1 - e^{-x}$. Se toma $u = 1 - e^{-x}$, entonces $x = -\log(1 - u) = F^{-1}(u)$. Por tanto, si $U \sim \mathcal{U}(0,1)$, se tiene que

$$X = -\log U \sim \mathcal{E}xp(1)$$

(tanto U como 1-U son uniformes).

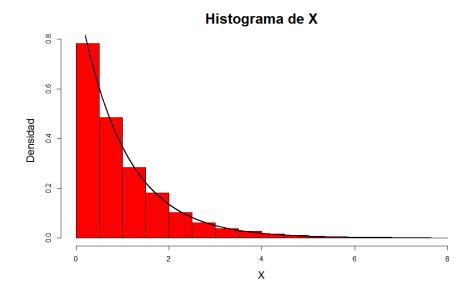


Figura 2.1: Histograma de la exponencial usando la transformación inversa. Además, se ha incluído en negro la fdd de la exponencial.

2.3. Métodos de transformación generales

Cuando una distribución con fdd f está relacionada con otra distribución que es fácil de simular, se puede utilizar esta relación para construir un algoritmo que simule variables de f.

Ejemplo 2.3.1

Veamos algunas de las variables aleatorias que pueden ser generadas a partir de una distribución exponencial. Se denotan por X_i 's variables aleatorias iid según una $\mathcal{E}xp(1)$, se pueden obtener tres distribuciones conocidas mediante

$$Y = 2 \sum_{j=1}^{\nu} X_j \sim \chi_{2\nu}^2, \quad \nu \in \mathbb{N},$$

$$Y = \beta \sum_{j=1}^{a} X_j \sim \mathcal{G}(a, \beta), \quad a \in \mathbb{N},$$

$$Y = \frac{\sum_{j=1}^{a} X_j}{\sum_{j=1}^{a+b} X_j} \sim \mathcal{B}e(a, b), \quad a, b \in \mathbb{N}.$$

16

Ventaja: estas transformaciones son bastantes fáciles de usar.

Desventaja: tiene limitaciones para su utilidad, como el rango de variables que pueden ser generadas y la eficiencia.

2.3.1. Distribuciones discretas

Usando la transformación inversa de la sección 2.2, se puede construir un algoritmo que trabajará formalmente con cualquier distribución discreta.

Generación de $X \sim P_{\theta}$, donde P_{θ} tiene soporte en los enteros no negativos

Paso 1 Definimos $p_{-1} = P_{\theta}[X \le -1] = 0$.

Paso 2 Se calculan las siguientes probabilidades y se almacenan,

$$p_0 = P_{\theta}[X \le 0], \quad p_1 = P_{\theta}[X \le 1], \quad p_2 = P_{\theta}[X \le 2], \dots$$

Paso 3 Se genera $U \sim \mathcal{U}(0,1)$.

Paso 4 Se toma X = k si $p_{k-1} < U \le p_k$.

Los algoritmos específicos suelen ser más eficientes principalmente porque tienen que almacenarse todas las probabilidades. Se puede mejorar el algoritmo eligiendo qué probabilidades calcular, tomando aquellas tales que $P[p_{k-1} < U \le p_k]$ sea suficientemente alta.

Ejemplo 2.3.2

Se va a simular una Poisson para grandes valores de λ mediante el método anterior y se compara el resultado obtenido con la función de probabilidad de la Poisson.

Se van a calcular las probabilidades p_k con $k \in (\max\{0, \lambda - 3\sqrt{\lambda}\}, \lambda + 3\sqrt{\lambda})$, pues fuera de este intervalo las probabilidades son casi cero y prácticamente 1, por lo que si no se calculan el algoritmo es más eficiente.

En este ejemplo se toma $\lambda = 100$.

→ En la primera iteración se obtiene

$$\begin{array}{l} U = 0.5580847 \\ p_{100} = 0.526562199 < U \leq 0.566028532 = p_{101} \end{array} \right\} \Longrightarrow X = 101$$

→ En la **segunda iteración** se obtiene

$$U = 0.7160623$$

 $p_{105} = 0.712807882 < U \le 0.745261333 = p_{106}$ $\Longrightarrow X = 106$

Repitiendo este proceso 10^4 veces se obtiene una muestra cuyo histograma representamos a continuación.

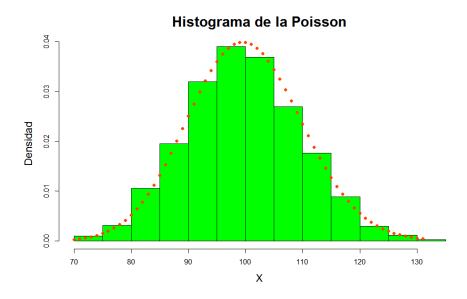


Figura 2.2: Histograma de la Poisson usando el método anterior. Además, se ha añadido en naranja la función de probabilidad de la Poisson.

2.3.2. Representaciones de mixturas

Una distribución de probabilidad puede ser representada como una *mixtura de distribuciones*, es decir, se puede escribir de la forma

$$f(x) = \int_{\mathcal{Y}} g(x|y) p(y) dy$$
 ó $f(x) = \sum_{i \in \mathcal{V}} p_i f_i(x),$

dependiendo de si el espacio \mathcal{Y} es continuo o discreto, donde g y p son distribuciones conocidas que pueden ser simuladas fácilmente.

Para generar una variable aleatoria X, podemos generar primero una variable Y de la distribución mixta y luego generar X de la distribución condicionada seleccionada. Es decir,

si
$$y \sim p(y)$$
 y $X \sim f(x|y)$, entonces $X \sim f(x)$ Caso continuo
si $\gamma \sim P[\gamma = i] = p_i$ y $X \sim f_{\gamma}(x)$, entonces $X \sim f(x)$ Caso discreto

2.4. Método Acepto-Rechazo

Hay muchas distribuciones para las cuales el método de la transformación inversa y las transformaciones generales no son suficientes para generar variables aleatorias. En estos casos, se recurren a métodos *indirectos*, métodos en los que se genera un candidato a variable aleatoria y se acepta si y sólo si verifica una cierta condición. Este método es muy potente y nos permitirá simular prácticamente cualquier distribución multivariante.

Elementos:

- **Densidad objetivo**: función de densidad f de interés.
- Densidad candidata: función de densidad q fácil de simular.

Condiciones que tiene que verificar la función de densidad g:

- 1. f y g tienen soportes compatibles (g(x) > 0 cuando f(x) > 0)
- 2. Existe una constante M tal que $\frac{f(x)}{g(x)} \leq M \quad \forall x$

En este caso, se puede simular X de la siguiente manera. En primer lugar, se genera $Y \sim g$ y, por otro lado, se genera $U \sim \mathcal{U}(0,1)$. Si

$$U \le \frac{1}{M} \frac{f(Y)}{g(Y)},$$

se toma X = Y. En otro caso, se descarta Y y U, y se vuelve a empezar.

Algoritmo del método Acepto-Rechazo

- 1. Generar $Y \sim g$, $U \sim \mathcal{U}(0,1)$.
- 2. Aceptar X = Y si $U \leq f(Y)/Mg(Y)$.
- 3. Volver a 1 en otro caso.

Proposición 2.4.1 La función de densidad de la variable aleatoria aceptada, $P[Y \le x | U \le f(Y)/Mg(Y)]$, es la función de densidad de X.

Demostración

$$P[Y \le x \,|\, U \le f(Y)/Mg(Y)] = \frac{P[Y \le x, \, U \le f(Y)/Mg(Y)]}{P[U \le f(Y)/Mg(Y)]}$$

$$= \frac{\int_{-\infty}^{x} \int_{0}^{f(y)/Mg(y)} du \, g(y) \, dy}{\int_{-\infty}^{\infty} \int_{0}^{f(y)/Mg(y)} du \, g(y) \, dy}$$

$$= \frac{\int_{-\infty}^{x} [f(y)/Mg(y)] \, g(y) \, dy}{\int_{-\infty}^{\infty} [f(y)/Mg(y)] \, g(y) \, dy}$$

$$= \frac{\int_{-\infty}^{x} f(y) \, dy}{\int_{-\infty}^{\infty} f(y) \, dy} = P[X \le x]$$

Ejemplo 2.4.2

Utilizando el método Acepto-Rechazo se generan variables aleatorias de una $\mathcal{B}e(a,b)$ con fdd

$$B(x, a, b) = \frac{x^{a-1} (1-x)^{b-1}}{\Gamma(a)\Gamma(b) / \Gamma(a+b)} \quad 0 < x < 1, \tag{2.1}$$

donde a > 0, b > 0 son parámetros y $\Gamma(z)$ es la función gamma.

Se tiene que elegir M de forma que satisfaga la condición $M \geq f/g$. En particular, se puede tomar

$$\begin{array}{rcl} M^* = & \max & f(x)/g(x) \\ & s.a: & x \in (0,1) \end{array},$$

para calcularla se utilizan las funciones optimize y objective de R.

Notar que la probabilidad de aceptación dada una realización es el área bajo la fdd f dividido por el área bajo la función M^*g , es decir, $1/M^*$ (f y g son fdd).

Se toman a = 2.7, b = 6.3.

Se va a aplicar el método para dos funciones candidatas distintas:

Caso 1: $g = \mathbf{I}_{(0,1)}$ (función de densidad de una $\mathcal{U}(0,1)$). En este caso, $M^* \approx 2.67$, por lo que se aceptan aproximadamente $1/2.67 \approx 37\%$ de los valores.

Caso 2: g = B(x, 2, 6). En este caso, $M^* \approx 1.67$, por lo que se aceptan aproximadamente $1/1.67 \approx 60 \%$ de los valores.

Notar que generar $U \sim \mathcal{U}(0,1)$ y multiplicar por M es equivalente a generar $U \sim \mathcal{U}(0,M)$.

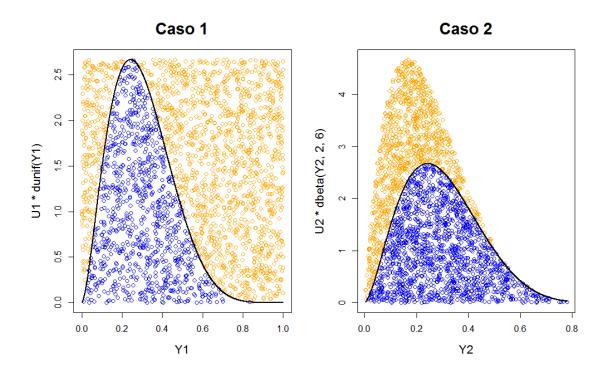


Figura 2.3: Generación de variables aleatorias de $\mathcal{B}e(2.7,6.3)$ usando el método Acepto-Rechazo. Se generan 2500 pares (Y,U) de g y $\mathcal{U}(0,M^*)$ respectivamente. Los puntos naranjas son los rechazados por el método y los azules los aceptados. Además, se ha incluido en negro la fdd f.

Propiedades del algoritmo Acepto-Rechazo

- 1. El intervalo $f \leq Mg$ no tiene que ser pequeño necesariamente, el algoritmo funciona cuando M es grande.
- 2. La probabilidad de aceptación es 1/M, por lo que M debería de ser lo más pequeña posible para mejorar la eficiencia del algoritmo.

Se puede mejorar el algoritmo de Acepto-Rechazo eligiendo la función de densidad candidata g en una familia paramétrica determinando el valor del parámetro que minimice la constante M.

21

Desventaja: genera valores "inútiles" de g cuando rechaza, incluso aquellos necesarios para validar el resultado siendo generados por f.

El método de *muestreo por importancia* puede solventar este problema, descrito en el capítulo 3.

Capítulo 3

Integración Monte Carlo

La posibilidad de producir infinitas variables aleatorias distribuidas de acuerdo a una distribución dada permite usar resultados frecuentistas y asintóticos mucho más fácilmente que en el ámbito lineal de la inferencia estadística, donde el tamaño de la muestra suele estar fijado. La ley fuerte de los grandes números y el teorema central del límite permiten estudiar la convergencia de los métodos de simulación.

Los métodos numéricos como la regla de Simpon o del trapecio no sirven para aproximar integrales multidimensionales, que son muy frecuentes en los problemas estadísticos.

El contenido de este capítulo se ha extraído del artículo [1].

3.1. Introducción

Problema: evaluar la siguiente integral, que se supone finita:

$$I = \mathbb{E}_f[h(X)] = \int_{\mathcal{X}} h(x) f(x) dx, \tag{3.1}$$

donde \mathcal{X} denota el soporte de X, que suele coincidir con el soporte de la función de densidad multivariante f.

Aplicaciones:

1. Divergencia KL

Es una manera de medir la diferencia entre dos distribuciones de probabilidad. La divergencia KL para dos distribuciones p(y) y q(y) se define

como

$$D(q(\cdot)||p(\cdot)) := \mathbb{E}_{y \sim q(y)} \left[\log \frac{q(y)}{p(y)} \right] = \int q(y) \log \frac{q(y)}{p(y)} dy,$$

definida sólo si q(y) > 0 para cualquier y tal que p(y) > 0.

2. Cálculo de la probabilidad de un suceso asociado a una densidad multivariante f.

Tomando $h = \mathcal{X}_A$ se tiene que

$$\int_{\mathcal{X}} \mathcal{X}_A f(x) \, dx = P[X \in A].$$

La estrategia de los métodos Monte Carlo para aproximar (3.1) es generar una muestra $\{X_1, \ldots, X_n\}$ de la función de densidad f y toma como aproximación la media muestral

$$\overline{h_n} = \frac{1}{n} \sum_{j=1}^n h(X_j).$$

Propiedades

- 1. $\overline{h_n}$ converge casi seguro a $\mathbb{E}_f[h(X)]$ por la ley fuerte de los grandes números.
- 2. Cuando $\mathbb{E}_{\underline{f}}[h^2(X)]$ es finita, se puede estimar la velocidad de convergencia de $\overline{h_n}$, del orden de $O(\sqrt{n})$, y calcular su varianza:

$$\operatorname{var}(\overline{h_n}) = \frac{1}{n} \int_{\mathcal{X}} (h(x) - \mathbb{E}_f[h(X)])^2 f(x) \, dx.$$

3. Se puede estimar $var(\overline{h_n})$ tomando la muestra $\{X_1, \ldots, X_n\}$ mediante

$$v_n = \frac{1}{n^2} \sum_{j=1}^{n} (h(x_j) - \overline{h_n})^2.$$

4. Debido al teorema central del límite, para n suficientemente grande se tiene que

$$\frac{\overline{h_n} - \mathbb{E}_f[h(X)]}{\sqrt{v_n}} \sim \mathcal{N}(0, 1),$$

lo que permite la construcción de un test de convergencia e intervalos de confianza para la aproximación de $\mathbb{E}_f[h(X)]$.

Aunque se tiene la garantía de que el método Monte Carlo proporciona un buen estimador, se debe tener en cuenta que es fiable si y sólo si v_n es un buen estimador de la varianza de $\overline{h_n}$. En los casos en los que v_n no converge lo suficientemente rápido como para poder aplicar el teorema central del límite, el estimador y las regiones de confianza asociadas no son fiables.

Los métodos Monte Carlo proporcionan buenas aproximaciones de (3.1) en la mayoría de los casos, pero existen alternativas más eficientes que no solo evitan la simulación de f, sino que también pueden ser usadas repetidamente para integrales de la forma (3.1), para una familia de funciones h o de densidad f, como los que se estudian en la sección 3.2. Para conseguir un estimador con una cierta precisión se tienen que realizar un número de simulaciones de f lo suficientemente grande.

3.2. Muestreo por importancia

Problema: se tiene interés en calcular la probabilidad de un suceso muy raro, por tanto, si se usan simulaciones de f, esto requeriría una gran cantidad de simulaciones para conseguir una aproximación estable. Sin embargo, gracias al *muestreo por importancia* se puede mejorar dicha aproximación disminuyendo el número de simulaciones.

El muestreo por importancia depende de las llamadas funciones de importancia, que son las distribuciones que vamos a usar en lugar de las distribuciones originales. De hecho, una aproximación de (3.1) basada en simulaciones de f no suele ser óptima en el sentido de que usando distribuciones alternativas podemos mejorar la varianza del estimador de (3.1).

Método de muestreo por importancia

Paso 1 Dada una función de densidad arbitraria g, estrictamente positiva cuando $h \cdot f$ es distinto de cero, se reescribe (3.1) de la forma:

$$\mathbb{E}_f[h(X)] = \int_{\mathcal{X}} h(x) \frac{f(x)}{g(x)} g(x) dx = \mathbb{E}_g \left[\frac{h(X) f(X)}{g(X)} \right], \quad (3.2)$$

donde \mathcal{X} es el soporte de X, que puede ser más pequeño que el soporte de la función de densidad g.

Paso 2 Se toma como estimador

$$\hat{I}_g = \frac{1}{n} \sum_{j=1}^n \frac{f(X_j)}{g(X_j)} h(X_j), \tag{3.3}$$

donde $\{X_1, \ldots, X_n\}$ es una muestra aleatoria iid de g.

Propiedad

Haciendo algunas manipulaciones se llega a que $\operatorname{var}(\hat{I}_g) = \sigma_g^2/n$, donde

$$\sigma_g^2 = \int_{\mathcal{X}} \left(\frac{f(x) h(x)}{g(x)} - I \right)^2 g(x) dx = \int_{\mathcal{X}} \frac{f^2(x) h^2(x)}{g(x)} dx - I^2.$$

Teorema 3.2.1 (Identidad fundamental del muestreo por importancia) Por la ley fuerte de los grandes números, se tiene que

$$\frac{1}{n} \sum_{j=1}^{n} \frac{f(X_j)}{g(X_j)} h(X_j) \longrightarrow \mathbb{E}_f[h(X)]. \tag{3.4}$$

La identidad (3.4) se tiene para cualquier elección de g, siempre que $sop(g) \supset sop(h \cdot f)$.

El muestreo por importancia es muy interesante pues impone muy pocas restricciones en la elección de la función de densidad g, que puede ser elegida de distribuciones fáciles de simular o eficientes en la aproximación de la integral. Además, la muestra generada por g puede ser usada repetidamente no sólo para diferentes funciones h, sino también para diferentes funciones de densidad f.

Ejemplo 3.2.2

Sea $Z \sim \mathcal{N}(0,1)$, se pretende calcular $P[Z > 4.5] = 3.398 \cdot 10^{-6}$, un valor muy pequeño. Para estimar esta probabilidad, se tendrían que realizar muchas iteraciones para obtener un estimador fiable. La solución a este problema es el muestreo por importancia.

Se pretende calcular la siguiente integral

$$I = \int_{-\infty}^{+\infty} f(z) h(z) dz = \int_{-\infty}^{+\infty} f(z) \mathbf{I}_{z>4.5} dz = P[Z > 4.5],$$

donde f es la fdd de una $\mathcal{N}(0,1)$.

→ Como se ha visto anteriormente, los **métodos Monte Carlo** toman como estimador la media muestral

$$\overline{h_n} = \frac{1}{n} \sum_{j=1}^n h(Z_j) = \frac{1}{n} \sum_{j=1}^n \mathbf{I}_{Z_j > 4.5},$$

donde $\{Z_1, \ldots, Z_n\}$ es una muestra de la función de densidad f.

El resultado obtenido mediante los métodos Monte Carlo para $n = 10^3$ es 0.

→ A continuación, se aplica el muestreo por importancia.

Si se considera una distribución con soporte restringido a $(4.5, \infty)$, la variación del estimador Monte Carlo debido a la simulación de ceros (cuando z < 4.5) desaparece.

Se toma g la fdd de $\mathcal{E}xp(1)$ truncada en 4.5,

$$g(z) = e^{-z} / \int_{4.5}^{\infty} e^{-z} dz = e^{-(z-4.5)} \mathbf{I}_{z>4.5}$$
.

El estimador de muestreo por importancia de la probabilidad P[Z > 4.5] es

$$\frac{1}{n}\sum_{j=1}^{n}\frac{f(Z_{j})}{g(Z_{j})}h(Z_{j}) = \frac{1}{n}\sum_{j=1}^{n}\frac{e^{-Z_{j}^{2}/2 + Z_{j} - 4.5}}{\sqrt{2\pi}},$$

donde $\{Z_1, \ldots, Z_n\}$ es una muestra iid de g.

Para generar variables aleatorias de $e^{-(z-4.5)} \mathbf{I}_{z>4.5}$ se generan variables aleatorias de $\mathcal{E}xp(1)$ y se le suman 4.5.

Se obtiene como resultado el valor $3.293 \cdot 10^{-6}$ para una muestra de tamaño $n=10^3$.

Existen otras variantes del muestreo por importancia, como el recocido o secuencial, que se pueden ver en el artículo [2].

Muestreo Importancia Método Monte Carlo valor exacto

Convergencia del muestreo por importancia

Figura 3.1: Convergencia del estimador de muestreo por importancia de la probabilidad P[Z>4.5] basada en la simulación de una distribución exponencial trasladada.

3.2.1. Remuestreo por importancia

El método de muestreo por importancia va más alla de aproximar integrales, proporciona una manera alternativa de simular distribuciones complejas. Recordemos que el método proporciona una muestra aleatoria iid $\{X_1, \ldots, X_n\}$ de g junto con sus "pesos de importancia" $f(X_i)/g(X_i)$. Esta muestra puede ser transformada mediante un remuestreo multinomial en una muestra (casi) de f.

Proposición 3.2.3 Para cualquier conjunto de Borel A, si se realiza un muestreo con remplazamiento de $\{X_1, \ldots, X_n\}$ tomando X_i con probabilidad $f(X_i)/ng(X_i)$, se obtendría una variable aleatoria X^* con función de densidad f.

Demostración

$$P[X^* \in A] = \sum_{i=1}^n P[X^* \in A, X^* = X_i] = \int_A \frac{f(x)}{g(x)} g(x) dx = \int_A f(x) dx$$

Este método produciría una simulación exacta de f. Sin embargo, la suma de los valores $f(X_i)/ng(X_i)$ no es 1, incluso algunas son mayores que 1, no representan una probabilidad, se tienen que normalizar.

Método de remuestreo por importancia

Paso 1 Se normalizan los pesos de importancia:

$$w_i = \frac{1}{n} \frac{f(X_i)}{g(X_i)} / \frac{1}{n} \sum_{j=1}^n \frac{f(X_j)}{g(X_j)} \quad \forall i = 1, \dots, n$$

Paso 2 Se usa el remuestreo multinomial para aproximar muestras de f.

Se puede observar que en el **Paso 1**, mientras que el denominador está convergiendo casi seguro a 1, la normalización produce un cambio en la distribución de los valores remuestreados. No obstante, para tamaños de muestras suficientemente grandes, este cambio es insignificante.

El uso de pesos normalizados en el estimador de muestreo por importancia produce el estimador de muestreo por importancia normalizado

$$\sum_{i=1}^{n} h(X_i) \frac{f(X_i)}{g(X_i)} / \sum_{i=1}^{n} \frac{f(X_j)}{g(X_j)}, \tag{3.5}$$

que puede ser usado en situaciones donde conozcamos f ó g salvo constante multiplicativa.

Observación 3.2.4

Los pesos de importancia solo proporcionan una "evaluación" de como de adecuada es la muestra simulada para la función de densidad objetivo, en el cuál indican la probabilidad de que X_i sea simulado por f comparado con X_j . Por ejemplo, si X_i tiene un peso normalizado próximo a 1, eso no significa que la probabilidad de que X_i sea generado por f sea muy alta, sino que tiene mucha más probabilidad de ser generado por f que el resto de valores simulados.

3.2.2. Selección de la función de importancia

Uno de los factores de los que depende la elección de la función de importancia q es la varianza del estimador resultante (3.3).

Mientras que se tiene (3.4), supuesto que la esperanza (3.1) existe, la varianza de este estimador es finita si y sólo si la siguiente esperanza es finita

$$\mathbb{E}_g\left[h^2(X)\frac{f^2(X)}{g^2(X)}\right] = \mathbb{E}_f\left[h^2(X)\frac{f(X)}{g(X)}\right] = \int_{\mathcal{X}} h^2(x)\frac{f^2(x)}{g(x)}\,dx < \infty.$$

Aunque no se prohíbe el uso de funciones de importancia con colas más ligeras que las de f, esto conduce a proporciones no acotadas f/g. Esta condición indica que esas funciones suelen conducir a estimadores de la varianza infinitos.

Cuando la proporción f/g no está acotada, los pesos de importancia $f(x_j)/g(x_j)$ tienen un rango de valores muy amplio, dando demasiada importancia a algunos valores x_j , por lo que disminuye la eficiencia del estimador (3.3). Por otro lado, distribuciones de importancia g con colas más pesadas que las de f aseguran que el comportamiento de la proporción f/g no es la causa de la divergencia de $\mathbb{E}_f[h^2(X)f(X)/g(X)]$.

La condición para que la varianza del estimador normalizado (3.5) sea finita es la misma que para el caso no normalizado.

Recomendación: usar distribuciones g tales que |h|f/g sea casi constante o al menos tenga un comportamiento de las colas controlado.

Casos particulares

- Cuando consideramos la aproximación de funciones h tal que $\mathbb{E}_f[h(X)]$ existe pero $\mathbb{E}_f[h^2(X)]$ no existe, se deben usar funciones de importancia con colas más pesadas. En estos casos, no se pueden usar los métodos Monte Carlo, ya que no existe la varianza de nuestro estimador.
- Cuando se tienen funciones h con soportes restringidos, se deben escoger funciones g con el mismo soporte que h, a no ser que no sea posible por la complejidad de h. Esto requiere tomar nuevas funciones de importancia por cada integrando h.

Estas restricciones en las colas de f no suelen ser fáciles de implementar, especialmente en altas dimensiones. Sin embargo, existe una alternativa basada en la incorporación de colas pesadas en la función de importancia g. Esta alternativa se conoce como $muestreo\ defensivo$.

Muestreo defensivo

Se basa en sustituir la función de densidad q por una densidad mixta

$$\rho q(x) + (1 - \rho)l(x), \quad 0 < \rho < 1,$$
 (3.6)

donde ρ está próximo a 1 y l es una función de densidad con colas pesadas.

Si se genera de (3.6), las observaciones son generadas con probabilidad ρ de g y con probabilidad $1-\rho$ de l, lo cual no afecta a los pesos de importancia. De hecho, estos pesos son $f(x)/(\rho g(x)+(1-\rho)l(x))$ para todos los valores generados.

Proposición 3.2.5 Si se generan ρn x_i 's de g y $(1-\rho)n$ y_i 's de l, se obtiene un estimador insesgado suponiendo que ρn es entero, es decir, el estimador de muestreo por importancia

$$\frac{1}{n} \sum_{i=1}^{\rho n} h(x_i) \frac{f(x_i)}{\rho g(x_i) + (1-\rho) l(x_i)} + \frac{1}{n} \sum_{i=1}^{(1-\rho)n} h(y_i) \frac{f(y_i)}{\rho g(y_i) + (1-\rho) l(y_i)}$$

tiene una esperanza igual a $\mathbb{E}_f[h(X)]$.

Propiedad

Si se toma $g_{\rho}(x) \equiv \rho f(x) + (1 - \rho)g(x)$ se tiene que

$$\sigma_{g_{\rho}}^{2} = \int_{\mathcal{X}} \frac{h^{2}(x) f^{2}(x)}{g_{\rho}(x)} dx - I^{2} \le \frac{1}{\rho} \left(\int_{\mathcal{X}} h^{2}(x) f(x) dx \right) - I^{2} = \frac{1}{\rho} \left(\sigma_{f}^{2} + I^{2} (1 - \rho) \right).$$

Por tanto, si la función de densidad f tiene varianza finita, el muestreo defensivo también.

Veamos un ejemplo donde, utilizando el muestreo por importancia, la varianza es infinita y el muestreo defensivo nos permite solventar este problema. Se va a utilizar la función de densidad de una beta definida en (2.1).

Ejemplo 3.2.6

Se toman

- $\mathcal{X} = (0,1)^5$.
- Función de densidad: $\underline{X} \sim \mathcal{U}(0,1)^5 \Longrightarrow f(\underline{x}) = 1$.
- Función

$$h(\underline{x}) = 0.9 \prod_{j=1}^{5} B(x^{j}, 20, 20) + 0.1 \prod_{j=1}^{5} B(x^{j}, 2, 2).$$

• Función de importancia

$$g(\underline{x}) = \prod_{j=1}^{5} B(x^{j}, 20, 20).$$

Por tanto,

$$I = \int_{\mathcal{X}} h(\underline{x}) f(\underline{x}) d\underline{x} = \int_{\mathcal{X}} \left(0.9 \prod_{j=1}^{5} B(x^{j}, 20, 20) + 0.1 \prod_{j=1}^{5} B(x^{j}, 2, 2) \right) d\underline{x}.$$

Se observa que I=1 pues el integrando es una fdd.

Por otro lado, se tiene que $\sigma_g^2 = \infty$, y por tanto $\text{var}(\hat{I}_g) = \infty$, pues desarrollando $\int_{\mathcal{X}} f^2(\underline{x}) \, h^2(\underline{x}) \, /g(\underline{x}) \, d\underline{x}$ se tiene

$$\int_{\mathcal{X}} \frac{1}{\prod_{j=1}^{5} B(x^{j}, 20, 20)} \left(0.81 \prod_{j=1}^{5} B(x^{j}, 20, 20)^{2} + 0.01 \prod_{j=1}^{5} B(x^{j}, 2, 2)^{2} + 0.18 \prod_{j=1}^{5} B(x^{j}, 20, 20) B(x^{j}, 2, 2) \right) d\underline{x} =$$

$$= \int_{\mathcal{X}} \left(0.81 \prod_{j=1}^{5} B(x^{j}, 20, 20) + 0.01 \prod_{j=1}^{5} \frac{B(x^{j}, 2, 2)^{2}}{B(x^{j}, 20, 20)} + 0.18 \prod_{j=1}^{5} B(x^{j}, 2, 2) \right) d\underline{x},$$

donde la integral del segundo sumando es divergente en \mathcal{X} pues

$$\frac{B(x^{j},2,2)^{2}}{B(x^{j},20,20)} = k \frac{x^{j^{2}} (1-x^{j})^{2}}{x^{j^{19}} (1-x^{j})^{19}} = k \frac{1}{x^{j^{17}} (1-x^{j})^{17}},$$

donde k es una constante.

En cambio, si se aplica el muestreo defensivo se obtendrá una varianza del estimador finita.

Nota 3.2.7

El muestreo defensivo es una buena alternativa en los casos donde $var(\hat{I}_g) = \infty$. Sin embargo, cuando la varianza es finita, el muestreo defensivo aumenta la varianza y, por tanto, proporciona una peor estimación de la integral.

Capítulo 4

Estadística Bayesiana

Sea y_1, \ldots, y_m una m.a.s generada con la función de densidad $p(y|\theta)$, donde θ es un parámetro que se considera variable aleatoria. Se denota $y = \{y_1, \ldots, y_m\}$ una variable aleatoria con función de probabilidad p(y).

Elementos

- Funciones de probabilidad: familia de funciones de densidad condicionada de y dado θ , denotadas $p(y|\theta)$.
- Distribución a priori: representa la información que tenemos sobre cómo son generados los datos antes de observarlos, denotada $p(\theta)$.
- Distribución a posteriori: función de densidad condicionada de θ dadas las observaciones, denotada $p(\theta|y)$.

Objetivo: calcular la distribución a posteriori.

Proposición 4.0.1 Usando las funciones de probabilidad y la distribución a priori, se puede obtener la distribución a posteriori usando la regla de Bayes,

$$p(\theta|y) = \frac{p(y|\theta) p(\theta)}{p(y)} \stackrel{probabilidad \ total}{=} \frac{p(y|\theta) p(\theta)}{\int p(y|\theta') p(\theta') d\theta'}.$$

La integral del denominador, llamada función de partición, suele ser intratable, sobre todo cuando el parámetro θ toma valores en un espacio de altas dimensiones.

Se pueden usar los métodos Monte Carlo para evitar el cálculo de esta integral.

Aplicaciones de los métodos Monte Carlo

1. Distribución posterior de predicción

$$p(y_{m+1}|y) = \int p(y_{m+1}|y,\theta) \, p(\theta|y) \, d\theta = \int p(y_{m+1}|\theta) \, p(\theta|y) \, d\theta.$$

2. Distribución de probabilidad marginal

$$p(x) = \int p(x|\theta) p(\theta) d\theta.$$

Para más información sobre la aplicación de los métodos Monte Carlo en la estadística bayesiana se pueden consultar los artículos [4] y [5].

4.1. Muestreo de Gibbs

El muestreo de Gibbs es una técnica de muestreo Markov Chain Monte Carlo (MCMC) muy potente para los casos en los que se tienen acceso a las distribuciones condicionadas. Es un caso particular del algoritmo Metropolis-Hastings mucho más rápido, pero sólo puede utilizarse en estas situaciones.

Se expresa $\theta \in \mathbb{R}^d$ como $\theta = (\theta_1, \dots, \theta_d)$. Se supone que, aunque no se puedan extraer muestras de $p(\theta|y)$ debido a la función de partición, se puedan extraer muestras de la distribución condicionada

$$p(\theta_j \mid \theta_1, \dots, \theta_{j-1}, \theta_{j+1}, \dots, \theta_d, y).$$

Descripción: El muestreo de Gibbs actualiza las variables $\theta_1, \ldots, \theta_d$ una por una. En cada paso, todas ellas permanecen constantes en su estado actual excepto θ_j , que es actualizada muestreando la distribución condicionada $p(\theta_j | \theta_1, \ldots, \theta_{j-1}, \theta_{j+1}, \ldots, \theta_d, y)$. A continuación, se mantiene el valor de θ_j en su estado y se procede con la actualización de la siguiente variable de manera similar.

Algoritmo del muestreo de Gibbs

Paso 1 Se toma $\theta^0 = (\theta^0_1, \dots, \theta^0_d)$. Se establece t = 1.

Paso 2 Para $j \in \{1, ..., d\}$, se obtiene θ_i^t de muestrear

$$p(\theta_j \mid \theta_1^t, \dots, \theta_{i-1}^t, \theta_{i+1}^{t-1}, \dots, \theta_d^{t-1}, y).$$

Paso 3 Mientras que la distribución estacionaria, distribución a la que converge el algoritmo, o el número de muestreos no se hayan alcanzado, se toma $t \leftarrow t + 1$ y se vuelve al **Paso 2**.

En cada iteración del muestreo de Gibbs, se actualizan secuencialmente cada componente de θ^t .

Ejemplo 4.1.1

Se consideran las distribuciones

$$X \mid \theta \sim \mathcal{B}in(n,\theta), \quad \theta \sim \mathcal{B}e(a,b),$$

junto con la distribución conjunta

$$f(x,\theta) = \binom{n}{x} \frac{\Gamma(a+b)}{\Gamma(a)\Gamma(b)} \theta^{x+a-1} (1-\theta)^{n-x+b-1}.$$

En este ejemplo, se quieren generar variables de X y θ aplicando el algoritmo anterior.

Se puede ver en el artículo [4] que $\theta \mid x \sim \mathcal{B}e(x+a, n-x+b)$.

Se toman n = 15, a = 3 y b = 7.

Realicemos dos iteraciones del algoritmo.

En primer lugar, se toma $\theta_0 = 0.5141716$.

→ En la primera iteración se obtiene

$$X_1 \sim f_{X|\theta}(\cdot | \theta_0) \implies X_1 = 5$$

 $\theta_1 \sim f_{\theta|X}(\cdot | X_1) \implies \theta_1 = 0.1918344$

→ En la **segunda iteración** se obtiene

$$X_2 \sim f_{X|\theta}(\cdot | \theta_1) \implies X_2 = 1$$

 $\theta_2 \sim f_{\theta|X}(\cdot | X_2) \implies \theta_2 = 0.1476743$

En la Figura 4.1 se tiene el histograma de X y θ para una muestra de tamaño 5000.

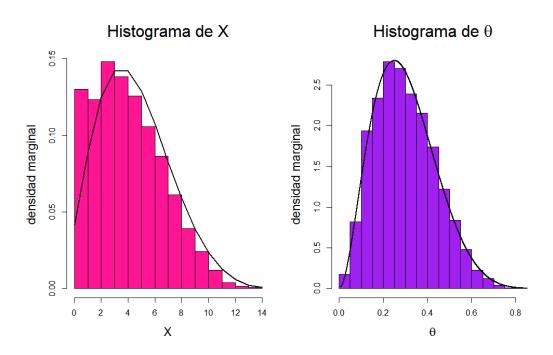


Figura 4.1: Histograma de las distribuciones marginales de X y θ usando el muestreo de Gibbs basado en 5000 iteraciones para n=15, a=3, b=7. La distribución marginal de θ es $\mathcal{B}e(a,b)$ y la distribución marginal de X es una beta-binomial.

La demostración de la convergencia del muestreo de Gibbs se encuentra en el artículo [7]. Para más información sobre el muestreo de Gibbs se puede recurrir al artículo [6].

Capítulo 5

Aplicación de los métodos Monte Carlo a la optimización estocástica

Muchos problemas se resuelven en el ámbito de la *optimización estocástica*, un área que consta de modelización y optimización de problemas que dependen de variables aleatorias.

El contenido de este capítulo se ha extraído del artículo [9].

Formulación de un problema de optimización estocástica irrestringido

$$\min_{s.a:} g_0(x) := \mathbb{E}[G_0(x,\zeta)]$$

$$s.a: x \in X$$
(5.1)

donde:

- x: vector de decisión.
- ζ : vector aleatorio.
- $X \subset \mathbb{R}^{d_x}$, con d_x la dimensión del vector x, las restricciones deterministas
- $\Xi \subset \mathbb{R}^{d_{\zeta}}$ el soporte de ζ , con d_{ζ} la dimensión del vector ζ .

Se asume que ζ sigue una distribución conocida que se denota P, independiente de x, y que las esperanzas respecto de ζ en (5.1) están bien definidas y son finitas para todo $x \in X$.

Existen muchas situaciones en las que es prácticamente imposible obtener todas las soluciones del problema. En estos casos, una buena alternativa sería usar *técnicas de muestreo*.

Existen diversas maneras de usar métodos de muestreo en el problema (5.1). Una manera genérica de describirlas es aproximando el problema como sigue.

Aproximación del problema de optimización estocástica irrestringido

Paso 1 Se considera una familia $\{g_n(\cdot)\}$ de aproximaciones aleatorias de la función $g(\cdot)$,

$$g_n(x) := \frac{1}{n} \sum_{j=1}^{N} G(x, \zeta^j),$$

donde $\{\zeta^1, \ldots, \zeta^n\}$ es una muestra de ζ .

Paso 2 Se construye el problema aproximado

$$\begin{array}{ll}
\min & g_n(x) \\
s.a: & x \in X
\end{array} \tag{5.2}$$

Cuando ζ^1, \ldots, ζ^n son independientes, $g_n(x)$ es llamado estimador Monte Carlo de g(x).

Notar que para cada $x \in X$, $g_n(x)$ es una variable aleatoria, depende de la muestra $\{\zeta^1, \ldots, \zeta^n\}$. Por tanto, las soluciones óptimas y el valor óptimo de (5.2) son variables aleatorias.

Construcción del problema determinista

Paso 1 Se toma una realización de la muestra, $\{\hat{\zeta}^1, \dots, \hat{\zeta}^n\}$, y se define

$$\hat{g}_n(x,\hat{\zeta}^1,\ldots,\hat{\zeta}^n) := \frac{1}{n} \sum_{j=1}^n G(x,\hat{\zeta}^j).$$

Paso 2 Se construye el problema determinista

$$\min_{s,a:} \hat{g}_n(x,\hat{\zeta}^1,\dots,\hat{\zeta}^n)
s.a: x \in X$$
(5.3)

Dado x, $\hat{g}_n(x, \hat{\zeta}^1, \dots, \hat{\zeta}^n)$ depende de la realización de la muestra. Se asume que las realizaciones son generadas usando los métodos Monte Carlo.

Una estrategia para resolver el problema (5.3) es la siguiente.

Algoritmo

- 1. Se elige una solución inicial x^0 , se toma k := 1.
- 2. Se obtiene una realización $\{\hat{\zeta}^{k,1},\ldots,\hat{\zeta}^{k,n_k}\}$ de $\{\zeta^1,\ldots,\zeta^{n_k}\}$.
- 3. Se realizan algunos pasos de optimización en la función $\hat{g}_{n_k}(\cdot,\hat{\zeta}^{k,1},\ldots,\hat{\zeta}^{k,n_k})$ (se puede utilizar información de iteraciones anteriores) para obtener x^k .
- 4. Se verifica un cierto criterio de optimalidad, si no se satisface, se toma k := k + 1 y se vuelve al Paso 2.

Variantes

- 1. Sample Average Approximation (SAA) Es un caso particular del Algoritmo donde en el Paso 3 se minimiza la función $\hat{g}_{n,1}(\cdot,\hat{\zeta}^{1,1},\ldots,\hat{\zeta}^{1,n_1})$, por lo que el Algoritmo para en una iteración.
- 2. Aproximación Estocástica (SA) Este método está definido por la recursión

$$x^{k+1} := x^k - \alpha_k \eta^k \quad k > 0,$$

donde

- η^k es una dirección aleatoria, usualmente un estimador de $\nabla g(x^k)$. Por ejemplo, $\nabla G(x^k, \zeta)$.
- α_k es la longitud de paso.

El método SA es un caso particular del Algoritmo donde $n_k = 1$ para todo k y el Paso 3 consiste en una iteración de la recurrencia $x^{k+1} := x^k - \alpha_k \eta^k$.

Se tratará con más detalle el método SAA en la sección 5.1.

En el ámbito de la optimización, donde se pretende resolver "problemas estocásticos de programación matemática", la función $g(\cdot)$ tiene ciertas propiedades como convexidad, continuidad, diferenciabilidad, etc.

Por otro lado, en el ámbito de la simulación, basado en métodos que no plantean hipótesis sobre la estructura de g, el objetivo es elegir x de manera que proporcione una solución que tenga determinadas propiedades, como optimalidad asintótica, o garantías de ser una buena solución.

5.1. El método SAA

El método SAA se utiliza para resolver problemas de la forma (5.3). Se toma una realización de $\{\zeta^1, \ldots, \zeta^n\}$ y se resuelve el problema de minimización.

En este método se asume que el problema aproximado puede ser resuelto de manera exacta.

Ejemplo 5.1.1 (Problema del vendedor de periódicos) Datos del problema

- Decisión: x, cantidad de inventario al principio de una temporada de ventas.
- Demanda: ζ , variable aleatoria no negativa con función de distribución F.
- Coste de inventario por unidad: c.
- Precio del producto por unidad: r.
- Precio de productos no vendidos en el inventario por unidad: v.
- Función objetivo: g(x), coste total si parto de x unidades en el inventario al principio de la temporada.

Se pretende resolver el problema

$$\min_{x} \left\{ g(x) = \mathbb{E}[cx - r \min\{x, \zeta\} - v \max\{x - \zeta, 0\}] \right\}. \tag{5.4}$$

Si v < c < r, cualquier x^* que satisfaga

$$F(x) \le \frac{r-c}{r-v} \quad \forall x < x^* \quad y \quad F(x) \ge \frac{r-c}{r-v} \quad \forall x > x^*$$
 (5.5)

es solución óptima. El conjunto de soluciones óptimas viene dado por el conjunto de γ -cuantiles de la distribución de ζ ,

$$S := \{ z \in \mathbb{R} \mid P[\zeta \ge z] \ge 1 - \gamma, \, P[\zeta \le z] \ge \gamma \},$$

donde $\gamma = (r-c)/(r-v)$. Notar que S es un intervalo cerrado no vacío para todo $\gamma \in (0,1)$.

Aplicación del método SAA

En primer lugar, se aproxima el problema (5.4)

$$\min_{x} \left\{ \hat{g}_{n}(x, \hat{\zeta}^{1}, \dots, \hat{\zeta}^{n}) := \frac{1}{n} \sum_{i=1}^{n} \left[cx - r \min\{x, \hat{\zeta}^{i}\} - v \max\{x - \hat{\zeta}^{i}, 0\} \right] \right\},$$

donde $\{\hat{\zeta}^1, \dots, \hat{\zeta}^n\}$ es una realización de ζ .

Cualquier muestra γ -cuantil es solución óptima del problema. Por ejemplo, se puede tomar $\hat{x}_n = \hat{\zeta}^{(\lceil \gamma n \rceil)}$ como solución óptima, donde $\hat{\zeta}^{(1)}, \dots, \hat{\zeta}^{(n)}$ representa un orden de las realizaciones.

Numéricamente, se toman c=10, r=30, v=5 y $F(x)=1-e^{-x}\mathbf{I}_{x\geq 0}$ $(\zeta \sim \mathcal{E}xp(1))$.

Sustituyendo en (5.5), se busca x^* tal que

$$F(x) \le \frac{4}{5} \quad \forall x < x^* \quad \text{y} \quad F(x) \ge \frac{4}{5} \quad \forall x > x^*.$$

- → Solución exacta: $x^* = F^{-1}\left(\frac{4}{5}\right) = 1.609438.$
- \rightarrow Solución aproximada: se genera una muestra $\{\zeta^1, \dots, \zeta^n\}$ de ζ y se construye la función de distribución empírica

$$F_n(\zeta) = \frac{1}{n} \sum_{i=1}^n \mathbf{I}_{\zeta \le \zeta^i}.$$

De este modo, $x^* = F_n^{-1}\left(\frac{4}{5}\right) = 1.615616$ para una muestra de tamaño 1000.

41

Enución objetivo diagram a g(x) diagram a g(x) valor óptimo

Problema del vendedor de periódicos

Figura 5.1: Representación de las funciones g(x) y $\hat{g}_n(x)$ con sus soluciones óptimas correspondientes.

3

Χ

4

5

Consistencia de SAA

Notación

- x_n^* : solución óptima del problema (5.2).
- x^* : solución óptima del problema (5.1).
- \bullet ν_n^* : valores objetivos óptimos del problema (5.2).

2

- ν^* : valores objetivos óptimos del problema (5.1).
- Π_n^* : conjunto de soluciones óptimas del problema (5.2).
- π^* : conjunto de soluciones óptimas del problema (5.1).
- Distancia euclídea de un punto x a un conjunto B: $d(x, B) = \inf_{y \in B} ||x y||$.
- Distancia entre dos conjuntos $A, B \subset \mathbb{R}^q$: $D(A, B) = \sup\{d(x, B) \mid x \in A\}$.

Condición suficiente para la convergencia de x_n^* a x^*

$$\sup_{x \in X} |g_n(x) - g(x)| = \sup_{x \in X} \left| \frac{1}{n} \sum_{i=1}^n G(x, \zeta^i) - \mathbb{E}[G(x, \zeta)] \right| \stackrel{n \to \infty}{\longrightarrow} 0$$

Es una modificación de la ley fuerte de los grandes números.

Definición 5.1.2 (Consistencia) El método SAA es consistente si $\nu_n^* \to \nu^*$ cuando $n \to \infty$.

Teorema 5.1.3 Si existe un subconjunto compacto $C \subset \mathbb{R}^d$ tal que

- (i) π^* es no vacío y está contenido en C,
- (ii) $\{g_n(x)\}\$ converge a g(x) uniformemente en C cuando $n \to \infty$,
- (iii) para n suficientemente grande, Π_n^* es no vacío y está contenido en C, entonces $\nu_n^* \to \nu^*$. Además, si la función $g(\cdot)$ es continua en C, se tiene que $D(\Pi_n^*, \pi^*) \to 0$ cuando $n \to \infty$.

La demostración de este teorema se encuentra en el artículo [10].

Observación 5.1.4

Si x_n^* es una solución exacta de (5.2), se tiene que $d(x_n^*, \pi^*) \to 0$ cuando $n \to \infty$. Además, si el problema (5.1) tiene una única solución óptima x^* , entonces $x_n^* \to x^*$.

Capítulo 6

Estimación del valor de Shapley

El valor de Shapley es el valor de un jugador en relación a su cooperación con el resto de jugadores.

La computación del valor de Shapley es un **problema NP completo**, puede ser resuelto en tiempo polinómico. Considerando la aplicación de la teoría de juegos a los problemas del mundo real, donde no suelen existir soluciones exactas, es necesario utilizar algoritmos que faciliten esta aproximación.

Notación

- N: conjunto finito de jugadores.
- S: coalición, $S \subseteq N$.
- Función recompensa:

$$\begin{array}{cccc} v: & 2^N = \{S \,|\, S \subseteq N\} & \longrightarrow & \mathbb{R} \\ & S \subseteq N & \longmapsto & v(S) \in \mathbb{R}, \end{array}$$

donde v(S) es el valor de la coalición S, es decir, la recompensa que la coalición S puede obtener por si misma si todos sus miembros trabajan conjuntamente. Se toma $v(\emptyset) = 0$.

- (N, v): juego cooperativo en forma de coalición donde N y v son los descritos anteriormente.
- n: cardinal del conjunto N.
- s: cardinal del conjunto S.

Definición 6.1.1 (Valor de Shapley 1) Se define el valor de Shapley como sique

$$Sh_i(v) = \sum_{\substack{S \subseteq N \\ i \notin S}} \frac{(n-s-1)! \, s!}{n!} \, (v(S \cup \{i\}) - v(S)), \quad i \in N,$$

donde $Sh_i(v)$ para cada jugador es una media ponderada de sus contribuciones marginales.

Existe una caracterización alternativa del valor de Shapley en función de todas las ordenaciones posibles de los jugadores.

Notación

- $O: N \to N$, permutación (ordenación) de los jugadores.
- $\pi(N)$: conjunto de todas las permutaciones posibles del conjunto de jugadores N.
- $Pre^{i}(O)$: conjunto de jugadores previos al jugador i en el orden de O, con $O \in \pi(N)$.
- $x(O) = (x(O)_i)_{i \in N}$: vector de contribuciones marginales dada una permutación $O \in \pi(N)$, definido como

$$x(O)_i = v(Pre^i(O) \cup \{i\}) - v(Pre^i(O)), \quad i \in N.$$

Definición 6.1.2 (Valor de Shapley 2) El valor de Shapley se puede expresar como la siguiente esperanza, donde se asume que todas las permutaciones tienen la misma probabilidad:

$$Sh_i(v) = \sum_{O \in \pi(N)} \frac{1}{n!} x(O)_i, \quad i \in N.$$

Problema: se tienen n! sumandos, número de ordenaciones posibles de los jugadores, y por la Fórmula de Stirling se tiene que

$$n! \approx \sqrt{2\pi n} \frac{n^n}{e} \approx n^n.$$

El valor de Shapley es una esperanza, por lo que Castro [13] propuso una estimación basada en un proceso de muestreo.

Algoritmo

- 1. La población del proceso de muestreo, P, es el conjunto de todas las ordenaciones posibles de los jugadores, es decir, $P = \pi(N)$. Cada muestra representa una permutación $O \in \pi(N)$.
- 2. Se toma como parámetro al vector $Sh = (Sh_i)_{i \in N}$.
- 3. Las características observadas en cada muestra, $O \in \pi(N)$, es el vector de las contribuciones marginales para cada permutación O, $x(O) = (x(O)_i)_{i \in N}$.
- 4. Se considera una muestra M subconjunto de $P \times P \times \cdots \times P$, es decir, una muestra con reemplazamiento.
- 5. La estimación del parámetro Sh es la media de las contribuciones marginales en la muestra M, es decir, $\hat{Sh} = (\hat{Sh}_i)_{i \in N}$ donde

$$\hat{Sh}_i = \frac{1}{m} \sum_{O \in M} x(O)_i, \quad i \in N.$$

Se observa que la muestra M tendrá el orden de alguna permutación $O \in \pi(N)$ con probabilidad $\frac{1}{n!}$.

Observación 6.1.3

Para asegurar que \hat{Sh} puede calcularse en tiempo polinómico, se asume que el valor de cualquier coalición S, v(S), puede calcularse en tiempo polinómico. Es suficiente asumir que el vector de las contribuciones marginales de todas las ordenaciones puede obtenerse en tiempo polinómico.

Desventaja: el valor de cualquier coalición debe poder ser computada polinomialmente.

Podemos considerar otros métodos de muestreo alternativos para reducir la varianza del estimador del método Monte Carlo. Por ejemplo, Maleki [14] propuso el muestreo aleatorio estratificado para reducir el error de estimación en el método descrito por Castro [13].

Para saber más sobre este método se puede consultar el artículo [12].

Ejemplo 6.1.4

En este ejemplo, vamos a obtener el valor de Shapley para cada una de las provincias andaluzas con respecto a una votación.

Elementos

- Jugadores: provincias andaluzas (8 jugadores).
- Coalición: S, existen 2⁸ coaliciones.
- Peso del jugador: población de la provincia.
- Función recompensa:

$$v(S) = \left\{ \begin{array}{ll} 1 & \text{si se consigue agrupar al} & 75\,\% & \text{de la población} \\ 0 & \text{en caso contrario} \end{array} \right.$$

Consideramos el siguiente conjunto de datos con la población de cada una de las provincias de Andalucía en el año 2021. Los datos se han extraído de https://www.citypopulation.de/es/spain/andalucia/.

Provincia	Población
Almería	727.945
Cádiz	1.244.049
Córdoba	781.451
Granada	919.168
Huelva	524.278
Jaén	631.381
Málaga	1.685.920
Sevilla	1.950.219
Total	8.464.411

Cuadro 6.1: Población de las provincias de Andalucía

En el cálculo del valor de Shapley exacto el número de sumandos es 40.320, para estimarlo tomamos m = 1.000 y m = 10.000 sumandos.

Para diferentes tamaños de muestra se obtienen los siguientes resultados:

Provincia	Valor de Shapley exacto	Valor de Shapley estimado (m=1.000)	Valor de Shapley estimado (m=10.000)
Almería	0.085	0.083	0.0844
Cádiz	0.127	0.132	0.1206
Córdoba	0.085	0.09	0.0889
Granada	0.118	0.114	0.113
Huelva	0.065	0.063	0.0669
Jaén	0.075	0.087	0.0791
Málaga	0.223	0.223	0.2225
Sevilla	0.223	0.208	0.2246

Cuadro 6.2: Valor de Shapley exacto y estimado para las provincias de Andalucía

Tenemos una buena estimación, pero hay casos como Málaga-Sevilla y Almería-Córdoba que tienen el mismo valor de Shapley y no tienen el mismo valor de Shapley estimado.

Ejemplo 6.1.5

Nos encontramos en la misma situación que en el ejemplo anterior, solo que ahora vamos a estimar el valor de Shapley para las 8 capitales de provincia de Andalucía.

Consideramos el siguiente conjunto de datos con la población de cada uno de los municipios de Andalucía en el año 2020.

Fuente: Instituto de Estadística y Cartografía de Andalucía. Explotación del Padrón municipal de habitantes del INE.

Los resultados que se muestran a continuación se han obtenido para m=500 sumandos.

Municipio	Valor de Shapley estimado
Almería (capital)	0.038
Cádiz (capital)	0.014
Córdoba (capital)	0.04
Granada (capital)	0.026
Huelva (capital)	0.014
Jaén (capital)	0.028
Málaga (capital)	0.06
Sevilla (capital)	0.07

Cuadro 6.3: Valor de Shapley estimado para las capitales de provincia de Andalucía $\,$

_

Cuadro 6.4: Población de los municipios de Andalucía

Municipio	Población	Municipio	Población	Municipio	Población	Municipio	Población
Abla	1248	Obejo	2008	Villanueva de las Torres	562	Alozaina	2061
Abrucena	1183	Palenciana	1465	Villanueva Mesía	2021	Alpandeire	263
Adra	25412	Palma del Río	20928	Víznar	993	Antequera	41318
Albanchez	687	Pedro Abad	2815	Zafarraya	2120	Árchez	385
Alboloduy	609	Pedroche	1482	Zagra	885	Archidona	8168
Albox	12083	Peñarroya- Pueblonuevo	10561	Zubia (La)	19330	Ardales	2506
Alcolea	844	Posadas	7318	Zújar	2544	Arenas	1177
Alcóntar	514	Pozoblanco	17204	Alájar	759	Arriate	4082
Alcudia de Monteagud	138	Priego de Córdoba	22367	Aljaraque	21474	Atajate	169
Alhabia	677	Puente Genil	29943	Almendro (El)	840	Benadalid	233
Alhama de Almería	3691	Rambla (La)	7529	Almonaster la Real	1788	Benahavís	7748
Alicún	201	Rute	9898	Almonte	24507	Benalauría	438
Almería (capital)	201322	San Sebastián de los Ballesteros	804	Alosno	3933	Benalmáde- na	69144
Almócita	176	Santaella	4614	Aracena	8255	Benamargosa	1535
Alsodux	125	Santa Eufemia	743	Aroche	3054	Benamocarra	3046
Antas	3315	Torrecampo	1040	Arroyomolino de León	s 947	Benaoján	1453
Arboleas	4624	Valenzuela	1118	Ayamonte	21104	Benarrabá	446
Armuña de Almanzora	300	Valsequillo	348	Beas	4341	Borge (El)	922
Bacares	250	Victoria (La)	2289	Berrocal	303	Burgo (El)	1792

Capítulo 6. Estimación del valor de Shapley

Balanegra	3105	Villa del Río	7067	Bollullos Par del Condado	14387	Campillos	8387
Bayárcal	312	Villafranca de Córdoba	4860	Bonares	6060	Canillas de Aceituno	1680
Bayarque	216	Villaharta	621	Cabezas Rubias	706	Canillas de Albaida	759
Bédar	984	Villanueva de Córdoba	8671	Cala	1159	Cañete la Real	1633
Beires	120	Villanueva del Duque	1448	Calañas	2768	Carratraca	740
Benahadux	4481	Villanueva del Rey	1015	Campillo (El)	2024	Cartajima	242
Benitagla	58	Villaralto	1118	Campofrío	713	Cártama	26738
Benizalón	262	Villaviciosa de Córdoba	3200	Cañaveral de León	396	Casabermeja	3629
Bentarique	240	Viso (El)	2517	Cartaya	20083	Casarabonela	2514
Berja	12563	Zuheros	635	Castaño del Robledo	228	Casares	6883
Canjáyar	1189	Agrón	260	Cerro de Andévalo (El)	2327	Coín	22673
Cantoria	3365	Alamedilla	569	Chucena	2223	Colmenar	3409
Carboneras	8105	Albolote	18962	Corteconcepc	ión 536	Comares	1330
Castro de Filabres	120	Albondón	712	Cortegana	4602	Cómpeta	3967
Chercos	298	Albuñán	409	Cortelazor	299	Cortes de la Frontera	3065
Chirivel	1489	Albuñol	7353	Cumbres de Enmedio	51	Cuevas Bajas	1370
Cóbdar	178	Albuñuelas	802	Cumbres de San Bartolomé	376	Cuevas del Becerro	1597
Cuevas del Almanzora	14455	Aldeire	630	Cumbres Mayores	1749	Cuevas de San Marcos	3601
Dalías	4066	Alfacar	5488	Encinasola	1310	Cútar	616
Ejido (El)	83758	Algarinejo	2467	Escacena del Campo	2287	Estepona	70228

Enix	514	Alhama de Granada	5867	Fuenteheridos	709	Faraján	261
Felix	636	Alhendín	9509	Galaroza	1382	Frigiliana	3062
Fines	2104	Alicún de Ortega	471	Gibraleón	12737	Fuengirola	82837
Fiñana	1984	Almegíjar	337	Granada de Río-Tinto (La)	243	Fuente de Piedra	2662
Fondón	994	Almuñécar	25927	Granado (El)	532	Gaucín	1589
Gádor	3034	Alpujarra de la Sierra	969	Higuera de la Sierra	1291	Genalguacil	393
Gallardos (Los)	2902	Alquife	580	Hinojales	330	Guaro	2280
Garrucha	9520	Arenas del Rey	653	Hinojos	3961	Humilladero	3297
Gérgal	1080	Armilla	24340	Huelva (capital)	143837	Igualeja	752
Huécija	486	Atarfe	18960	Isla Cristina	21393	Istán	1494
Huércal de Almería	17917	Baza	20430	Jabugo	2260	Iznate	907
Huércal- Overa	19432	Beas de Granada	980	Lepe	27880	Jimera de Líbar	379
Íllar	405	Beas de Guadix	329	Linares de la Sierra	262	Jubrique	523
Instinción	448	Benalúa	3310	Lucena del Puerto	3261	Júzcar	214
Laroya	198	Benalúa de las Villas	1054	Manzanilla	2118	Macharaviaya	513
Láujar de Andarax	1527	Benamaurel	2260	Marines (Los)	399	Málaga (capital)	578460
Líjar	394	Bérchules	712	Minas de Riotinto	3812	Manilva	16439
Lubrín	1449	Bubión	298	Moguer	21867	Marbella	147633
Lucainena de las Torres	564	Busquístar	279	Nava (La)	258	Mijas	85397

Capítulo 6. Estimación del valor de Shapley

Lúcar	760	Cacín	568	Nerva	5169	Moclinejo	1251
Macael	5480	Cádiar	1465	Niebla	4116	Mollina	5276
María	1230	Cájar	5205	Palma del Condado (La)	10801	Monda	2697
Mojácar	6778	Calahorra (La)	668	Palos de la Frontera	11742	Montecorto	586
Mojonera (La)	9191	Calicasas	638	Paterna del Campo	3457	Montejaque	949
Nacimiento	496	Campotéjar	1229	Paymogo	1162	Nerja	21144
Níjar	31666	Caniles	4021	Puebla de Guzmán	3092	Ojén	3816
Ohanes	558	Cáñar	377	Puerto Moral	278	Parauta	239
Olula de Castro	191	Capileira	563	Punta Umbría	15355	Periana	3089
Olula del Río	6256	Carataunas	201	Rociana del Condado	7939	Pizarra	9444
Oria	2239	Cástaras	235	Rosal de la Frontera	1697	Pujerra	308
Padules	414	Castilléjar	1321	San Bartolomé de la Torre	3761	Rincón de la Victoria	48768
Partaloa	1035	Castril	2020	San Juan del Puerto	9411	Riogordo	2723
Paterna del Río	365	Cenes de la Vega	8153	Sanlúcar de Guadiana	403	Ronda	33730
Pechina	4103	Chauchina	5552	San Silvestre de Guzmán	614	Salares	173
Pulpí	10358	Chimeneas	1251	Santa Ana la Real	475	Sayalonga	1568
Purchena	1664	Churriana de la Vega	15689	Santa Bárbara de Casa	1043	Sedella	576
Rágol	303	Cijuela	3415	Santa Olalla del Cala	2022	Serrato	482

Rioja	1453	Cogollos de Guadix	642	Trigueros	7862	Sierra de Yeguas	3318
Roquetas de Mar	98433	Cogollos de la Vega	2007	Valdelarco	231	Teba	3744
Santa Cruz de Marchena	199	Colomera	1302	Valverde del Camino	12750	Tolox	2136
Santa Fe de Mondújar	461	Cortes de Baza	1844	Villablanca	2885	Torremolinos	69166
Senés	289	Cortes y Graena	982	Villalba del Alcor	3366	Torrox	17943
Serón	2033	Cuevas del Campo	1744	Villanueva de las Cruces	387	Totalán	737
Sierro	379	Cúllar	4091	Villanueva de los Castillejos	2825	Valle de Abdalajís	2502
Somontín	443	Cúllar Vega	7547	Villarrasa	2211	Vélez- Málaga	82365
Sorbas	2436	Darro	1634	Zalamea la Real	3054	Villanueva de Algaidas	4148
Suflí	196	Dehesas de Guadix	422	Zarza- Perrunal (La)	1253	Villanueva de la Concepción	3288
Tabernas	3717	Dehesas Viejas	688	Zufre	796	Villanueva del Rosario	3362
Taberno	974	Deifontes	2600	Albanchez de Mágina	1011	Villanueva del Trabuco	5308
Tahal	336	Diezma	764	Alcalá la Real	21709	Villanueva de Tapia	1460
Terque	365	Dílar	2092	Alcaudete	10483	Viñuela	2055
Tíjola	3548	Dólar	628	Aldeaquemada	a 476	Yunquera	2869
Tres Villas (Las)	565	Domingo Pérez de Granada	878	Andújar	36615	Aguadulce	2035
Turre	3781	Dúdar	327	Arjona	5515	Alanís	1706
Turrillas	248	Dúrcal	7111	Arjonilla	3571	Albaida del Aljarafe	3201

Capítulo 6. Estimación del valor de Shapley

Uleila del	822	Escúzar	791	Arquillos	1722	Alcalá de	75533
Campo						Guadaíra	
Urrácal	350	Ferreira	301	Arroyo del Ojanco	2287	Alcalá del Río	12186
Velefique	230	Fonelas	978	Baeza	15791	Alcolea del Río	3347
Vélez- Blanco	1874	Fornes	556	Bailén	17548	Algaba (La)	16503
Vélez-Rubio	6546	Freila	939	Baños de la Encina	2576	Algámitas	1253
Vera	16996	Fuente Vaqueros	4384	Beas de Segura	5100	Almadén de la Plata	1307
Viator	5978	Gabias (Las)	21543	Bedmar y Garcíez	2633	Almensilla	6175
Vícar	26899	Galera	1071	Begíjar	3009	Arahal	19533
Zurgena	2956	Gobernador	229	Bélmez de la Moraleda	1569	Aznalcázar	4588
Alcalá de los Gazules	5226	Gójar	5909	Benatae	445	Aznalcóllar	6083
Alcalá del Valle	5043	Gor	744	Cabra del Santo Cristo	1782	Badolatosa	3079
Algar	1428	Gorafe	383	Cambil	2683	Benacazón	7222
Algeciras	123078	Granada (capital)	233648	Campillo de Arenas	1751	Bollullos de la Mitación	10931
Algodonales	5506	Guadahortuna	1872	Canena	1814	Bormujos	22180
Arcos de la Frontera	30818	Guadix	18436	Carboneros	591	Brenes	12501
Barbate	22556	Guájares (Los)	1053	Cárcheles	1332	Burguillos	6766
Barrios (Los)	23777	Gualchos	5190	Carolina (La)	15160	Cabezas de San Juan (Las)	16358
Benalup- Casas Viejas	6986	Güéjar Sierra	2857	Castellar	3292	Camas	27560
Benaocaz	673	Güevéjar	2640	Castillo de Locubín	4049	Campana (La)	5238

Bornos	7654	Huélago	401	Cazalilla	803	Cantillana	10736
Bosque (El)	2137	Huéneja	1170	Cazorla	7352	Cañada Rosal	3325
Cádiz (capital)	115439	Huéscar	7195	Chiclana de Segura	955	Carmona	28834
Castellar de la Frontera	3057	Huétor de Santillán	1882	Chilluévar	1410	Carrión de los Céspedes	2547
Chiclana de la Frontera	85150	Huétor Tájar	10399	Escañuela	944	Casariche	5429
Chipiona	19246	Huétor Vega	12035	Espeluy	620	Castilblanco de los Arroyos	4899
Conil de la Frontera	22775	Íllora	10065	Frailes	1585	Castilleja de Guzmán	2833
Espera	3869	Ítrabo	991	Fuensanta de Martos	3038	Castilleja de la Cuesta	17516
Gastor (El)	1729	Iznalloz	5120	Fuerte del Rey	1355	Castilleja del Campo	640
Grazalema	2013	Játar	623	Génave	590	Castillo de las Guardas (El)	1468
Jerez de la Frontera	213105	Jayena	1052	Guardia de Jaén (La)	4983	Cazalla de la Sierra	4684
Jimena de la Frontera	6707	Jérez del Marquesado	955	Guarromán	2739	Constantina	5864
Línea de la Concepción (La)	63630	Jete	927	Higuera de Calatrava	614	Coria del Río	30908
Medina Sidonia	11773	Jun	3853	Hinojares	373	Coripe	1238
Olvera	8062	Juviles	135	Hornos	594	Coronil (El)	4717
Paterna de Rivera	5453	Láchar	3513	Huelma	5744	Corrales (Los)	3938
Prado del Rey	5590	Lanjarón	3530	Huesa	2490	Cuervo de Sevilla (El)	8656

Capítulo 6. Estimación del valor de Shapley

Puerto de							
Santa María (El)	88703	Lanteira	564	Ibros	2827	Dos Hermanas	135050
Puerto Real	41700	Lecrín	2156	Iruela (La)	1879	Écija	39893
Puerto Serrano	6942	Lentegí	333	Iznatoraf	942	Espartinas	15824
Rota	29282	Lobras	133	Jabalquinto	2017	Estepa	12497
San Fernando	95001	Loja	20419	Jaén (capital)	112757	Fuentes de Andalucía	7152
San José del Valle	4433	Lugros	310	Jamilena	3271	Garrobo (El)	782
Sanlúcar de Barrameda	69205	Lújar	483	Jimena	1269	Gelves	10193
San Martín del Tesorillo	2801	Malahá (La)	1784	Jódar	11634	Gerena	7674
San Roque	31571	Maracena	22306	Lahiguera	1686	Gilena	3727
Setenil de las Bodegas	2741	Marchal	414	Larva	474	Gines	13428
Tarifa	18183	Moclín	3622	Linares	57353	Guadalcanal	2608
Torre Alháquime	796	Molvízar	2772	Lopera	3652	Guillena	12903
Trebujena	7016	Monachil	8007	Lupión	820	Herrera	6464
Ubrique	16605	Montefrío	5313	Mancha Real	11328	Huévar del Aljarafe	3060
Vejer de la Frontera	12622	Montejícar	2110	Marmolejo	6763	Isla Mayor	5825
Villaluenga del Rosario	455	Montillana	1129	Martos	24343	Lantejuela	3846
Villamartín	12134	Moraleda de Zafayona	3117	Mengíbar	9997	Lebrija	27578
Zahara	1379	Morelábor	572	Montizón	1689	Lora de Estepa	873
Adamuz	4137	Motril	58460	Navas de San Juan	4528	Lora del Río	18630
Aguilar de la Frontera	13382	Murtas	455	Noalejo	1932	Luisiana (La)	4578
Alcaracejos	1470	Nevada	1072	Orcera	1791	Madroño (El)	287

Almedinilla	2370	Nigüelas	1185	Peal de Becerro	5240	Mairena del Alcor	23698
Almodóvar del Río	7932	Nívar	1036	Pegalajar	2875	Mairena del Aljarafe	46555
Añora	1530	Ogíjares	14348	Porcuna	6156	Marchena	19329
Baena	19045	Orce	1185	Pozo Alcón	4731	Marinaleda	2592
Belalcázar	3235	Órgiva	5784	Puente de Génave	2159	Martín de la Jara	2665
Belmez	2921	Otívar	1011	Puerta de Segura (La)	2234	Molares (Los)	3518
Benamejí	4963	Villa de Otura	6982	Quesada	5196	Montellano	7053
Blázquez (Los)	664	Padul	8694	Rus	3531	Morón de la Frontera	27633
Bujalance	7320	Pampaneira	317	Sabiote	3886	Navas de la Concepción (Las)	1542
Cabra	20347	Pedro Martínez	1138	Santa Elena	890	Olivares	9466
Cañete de las Torres	2923	Peligros	11492	Santiago de Calatrava	675	Osuna	17621
Carcabuey	2371	Peza (La)	1167	Santiago- Pontones	2911	Palacios y Villafranca (Los)	38548
Cardeña	1471	Pinar (El)	865	Santisteban del Puerto	4463	Palmar de Troya (El)	2343
Carlota (La)	14079	Pinos Genil	1490	Santo Tomé	2129	Palomares del Río	8843
Carpio (El)	4363	Pinos Puente	9894	Segura de la Sierra	1803	Paradas	6857
Castro del Río	7767	Píñar	1103	Siles	2220	Pedrera	5158
Conquista	366	Polícar	263	Sorihuela del Guadalimar	1098	Pedroso (El)	2009
Córdoba (capital)	326039	Polopos	1722	Torreblascope	edro 2508	Peñaflor	3639

Capítulo 6. Estimación del valor de Shapley

Doña	4615	Pórtugos	368	Torredelcampo	14142	Pilas	13952
Mencía	4010		300	Torredeleampo	14142	1 1185	10302
Dos Torres	2394	Puebla de Don Fadrique	2238	Torredonjimeno	13632	Pruna	2605
Encinas Reales	2254	Pulianas	5466	Torreperogil	7222	Puebla de Cazalla (La)	10932
Espejo	3289	Purullena	2306	Torres	1372	Puebla de los Infantes (La)	2980
Espiel	2405	Quéntar	947	Torres de Albánchez	770	Puebla del Río (La)	11920
Fernán- Núñez	9651	Rubite	382	Úbeda	34329	Real de la Jara (El)	1494
Fuente Carreteros	1148	Salar	2621	Valdepeñas de Jaén	3679	Rinconada (La)	39062
Fuente la Lancha	342	Salobreña	12513	Vilches	4409	Roda de Andalucía (La)	4175
Fuente Obejuna	4501	Santa Cruz del Comercio	531	Villacarrillo	10673	Ronquillo (El)	1377
Fuente Palmera	9766	Santa Fe	15222	Villanueva de la Reina	3038	Rubio (El)	3369
Fuente- Tójar	666	Soportújar	264	Villanueva del Arzobispo	8078	Salteras	5564
Granjuela (La)	439	Sorvilán	527	Villardompardo	968	San Juan de Aznalfa- rache	21774
Guadalcázar	1559	Taha (La)	670	Villares (Los)	6020	Sanlúcar la Mayor	13943
Guijarrosa (La)	1379	Torre- Cardela	716	Villarrodrigo	391	San Nicolás del Puerto	596
Guijo (El)	352	Torrenueva Costa	2796	Villatorres	4298	Santiponce	8538
Hinojosa del Duque	6707	Torvizcón	625	Alameda	5387	Saucejo (El)	4235

Hornachuelos	4480	Trevélez	736	Alcaucín	2279	Sevilla (capital)	691395
Iznájar	4134	Turón	232	Alfarnate	1065	Tocina	9486
Lucena	42733	Ugíjar	2517	Alfarnatejo	364	Tomares	25455
Luque	2976	Valderrubio	2066	Algarrobo	6556	Umbrete	9000
Montalbán de Córdoba	4467	Valle (El)	907	Algatocín	814	Utrera	50962
Montemayor	3856	Valle del Zalabí	2105	Alhaurín de la Torre	41170	Valencina de la Concepción	7776
Montilla	22739	Válor	677	Alhaurín el Grande	25358	Villamanrique de la Condesa	4477
Montoro	9293	Vegas del Genil	11432	Almáchar	1817	Villanueva del Ariscal	6591
Monturque	1943	Vélez de Benaudalla	2890	Almargen	1947	Villanueva del Río y Minas	4847
Moriles	3701	Ventas de Huelma	657	Almogía	3792	Villanueva de San Juan	1090
Nueva Carteya	5341	Villamena	934	Álora	13112	Villaverde del Río	7842
						Viso del Alcor (El)	19324

Bibliografía

- [1] Christian P. Robert, George Casella, *Introducing Monte Carlo Methods with R*, Springer Science+Business Media, LLC 2010.
- [2] Surya T. Tokdar and Robert E. Kass, *Importance Sampling:* a review, 2009 John Wiley & Sons, Inc. Volume 2, January/February 2010.
- [3] Rebecca C.Steorts, Module 1: Introduction to Bayesian Statistics, Part I.
- [4] Ben Letham and Cynthia Rudin, 15097: Probabilistic Modeling and Bayesian Analysis.
- [5] ROBERT L. WOLPERT, Monte Carlo Integration in Bayesian Statistical Analysis, Institute of Statistics and Decision Sciences, DP# 90-A-06.
- [6] Alan E. Gelfand, Gibbs Sampling, Journal of the American Statistical Association, Vol. 95, No. 452. (Dec., 2000), pp. 1300-1304.
- [7] A. F. M. SMITH AND G. O. ROBERTS, Bayesian Computation Via the Gibbs Sampler and Related Markov Chain Monte Carlo Methods, Journal of the Royal Statistical Society. Series B (Methodological), 1993, Vol. 55, No. 1 (1993), pp. 3-23.
- [8] MAKOTO MATSUMOTO AND TAKUJI NISHIMURA, Mersenne Twister: A 623-Dimensionally Equidistributed Uniform Pseudo-Random Number Generator, ACM Transactions on Modeling and Computer Simulation, Vol. 8, No.1, January, 1998.
- [9] TITO HOMEM-DE-MELLO AND GÜZIN BAYRAKSAN, Monte Carlo Sampling-Based Methods for Stochastic Optimization, January 22, 2014.

- [10] MICHAEL C. Fu, *Handbook of Simulation Optimization*, International Series in Operations Research & Management Science.
- [11] ART OWEN AND YI ZHOU ASSOCIATE, Safe and Effective Importance Sampling, Journal of the American Statistical Association March 2000, Vol. 95, No. 449, Theory and Methods.
- [12] JAVIER CASTRO, DANIEL GÓMEZ, ELISENDA MOLINA, JUAN TEJA-DA, Improving polynomial estimation of the Shapley value by stratified random sampling with optimum allocation, Computers and Operations Research 82 (2017) 180-188.
- [13] Castro, J, Gómez, D, Tejada, J, 2009. Polynomial calculation of the Shapley value based on sampling. Comput. Oper. Res. 36 (5), 1726-1730.
- [14] Maleki, S., Tran-Thanh, L., Hines, G., Rahwan, T., Rogers, A., 2014. Bounding the estimation error of sampling-based Shapley value approximation. ArXiv:1306.4265.