UNIVERSIDAD COMPLUTENSE DE MADRID

FACULTAD DE CIENCIAS MATEMÁTICAS

DEPARTAMENTO DE SISTEMAS INFORMÁTICOS Y COMPUTACIÓN

TRABAJO DE FIN DE GRADO

Métodos de Simulación de Monte-Carlo para la Optimización Heurística

Presentado por: Sergio González Montero

Dirigido por: Rafael del Vado Vírseda

Grado en Matemáticas

Curso académico 2024-25

Agradecimientos

Agradecimientos

Resumen

Resumen

Palabras clave:

Abstract

Resumen en inglés

Keywords*:*

Índice general

Capítulo 1

Introducción

Este primer capítulo se centra en presentar el marco contextual en el que se desarrolla este trabajo. Por un lado,

1.1. Motivación y objetivos

1.2. Contexto y antecedentes del trabajo

1.2.1. Problemas de optimización

1.2.2. Algoritmos heurísticos

1.2.3. Métodos de optimización de Monte-Carlo

1.3. Estructura de la memoria

La presente memoria se estructura del modo siguiente:

- En el presente **Capítulo 1** hemos presentado el tema de este trabajo, a través de
- El **Capítulo 2** se centra en
- En el **Capítulo 3** hablamos sobre
- El **Capítulo 4** se centra en
- Por último, en el **Capítulo 5** se presentan las conclusiones finales, el trabajo futuro y algunos trabajos publicados relacionados con la investigación realizada.

A lo largo de esta memoria se presentan diferentes cuadros de código que sirven para apoyar las experimentaciones realizadas y las explicaciones textuales. Sin embargo, la totalidad del código implementado puede encontrarse en el siguiente repositorio de Github `https://github.com/mavice07/TFG-Mates.git`.

Ejemplo de cita para lla bibliografía [1]

Capítulo 2

Optimización heurística

En este capítulo nos centraremos en

2.1. Conceptos generales

La optimización matemática es una técnica de investigación operativa utilizada en la toma de decisiones. El campo de la investigación operativa se sirve además de otras herramientas como la estadística, la teoría de la probabilidad, la simulación y el análisis de decisiones.

En particular, la optimización matemática, en su búsqueda de la mejor solución posible a un problema, requiere de tres elementos:

- Modelo de optimización, constituido de variables de decisión, restricciones y funciones objetivo.
- Datos, para las demandas del modelo.
- Algoritmo, para resolver la instancia del modelo.

Con esto, se buscan los valores de las variables de decisión que satisfagan todas las restricciones y optimicen las funciones objetivo.

2.2. Problemas de optimización

Para plantear un problema de optimización hay que definir los elementos anteriormente citados.

Las variables de decisión en un modelo de optimización representan los parámetros variables del problema. Sus valores varían según marque el algoritmo con el fin de obtener aquellos que mejor se ajusten a las funciones objetivo. Los aspectos más relevantes de estas variables son el dominio al cual pertenecen, los límites entre los que se disponen y que son delimitados por la frontera del dominio y, finalmente, el tipo de variable, ya que puede ser real, entero, booleano, etc. Es importante hacer una elección coherente de

estas características, pues ejerce un serio impacto en la redacción de las restricciones, la elección del algoritmo y, por tanto, en los resultados obtenidos.

Las restricciones representan los límites del conjunto de soluciones que se está dispuesto a admitir. Se elaboran mediante relaciones entre las variables de decisión, así como entre sus valores.

Las funciones objetivo en un modelo de optimización son representaciones matemáticas del ámbito que se desea optimizar. Estas funciones pueden ser maximizadas o minimizadas, y su tratamiento, ya sea individual o combinado en problemas con múltiples objetivos, determina el tipo de algoritmo de optimización.

Una vez sentados estos elementos y finalizado el algoritmo, se llega a la solución del problema, la cual puede ser una combinación de factible (satisface las restricciones), óptima (factible que alcanza el mejor valor posible), no factible (fuera de la región factible, viola alguna restricción) y no acotada (tiende a más o a menos infinito en maximización y en minimización, respectivamente).

Más formalmente, $\mathbf{x}$ es el vector de N variables de decisión del sistema, Ω representa el espacio sobre el cual están definidas las soluciones factibles, $f_i(\mathbf{x})$ cada una de las O funciones objetivo del problema y optG$(\cdot)$ el método para conjuntar y optimizar los objetivos de manera sincrónica. A su vez, el espacio Ω queda definido por el tipo $\boldsymbol{T}$ de cada variable, y las relaciones $g_j(\mathbf{x})$, $h_k(\mathbf{x})$ que definen las restricciones del problema.

$$\begin{aligned} &\text{optG}\left(f_1(\mathbf{x}), f_2(\mathbf{x}), \ldots, f_O(\mathbf{x})\right) \\ &\quad \mathbf{x} = [x_1, x_2, \ldots, x_N] \\ &\quad \mathbf{x} \in \Omega \end{aligned} \qquad \mathbf{x} \in \Omega = \begin{cases} x_i \in T_i \\ g_j(\mathbf{x}) \leqslant 0,\ j = 1 : R \\ h_k(\mathbf{x}) = 0,\ k = 1 : P \end{cases}$$

Existen, entre los métodos de combinación y optimización simultánea, dos grandes familias a destacar:

- Los métodos que convierten un problema con $O > 1$ objetivos en sólo un problema mediante una combinación (a menudo, lineal) de todos los objetivos y que optimizan el valor combinado, como por ejemplo $\min\limits_{\mathbf{x}} \sum\limits_{i=1}^{O} w_i f_i(\mathbf{x})$.
- Los métodos que optimizan de forma simultánea todos los objetivos del problema utilizando una definición en la que se establece si una solución es mejor que otra comparando cada uno de los objetivos por separado. Por ejemplo, con la definición de Pareto optimalidad, una solución $\mathbf{A}$ es mejor que una solución $\mathbf{B}$ si para todo i, $f_i(\mathbf{A})$ no es peor que $f_i(\mathbf{B})$ y existe un j tal que $f_j(\mathbf{A})$ es mejor que $f_j(\mathbf{B})$. Este procedimiento se conoce como frentes Pareto de soluciones igualmente óptimas.

Hay que remarcar que los problemas con un solo objetivo son un caso particular de los problemas con objetivos múltiples, y no es estrictamente necesario el uso de una función G$(\cdot)$.

Más definiciones formales de los problemas de optimización son:

- Vector de decisión factible: vector $\mathbf{x}$ que cumple las restricciones, $\mathbf{x} \in \Omega$.

- Óptimo local: un vector de decisión factible $\mathbf{x}^*$ representa un óptimo local de la función $\mathrm{G}(f_1(\mathbf{x}), f_2(\mathbf{x}), \ldots, f_O(\mathbf{x}))$ si existe $\varepsilon > 0$ tal que $\mathrm{G}(f_1(\mathbf{x}^*), f_2(\mathbf{x}^*), \ldots, f_O(\mathbf{x}^*))$ no es peor que $\mathrm{G}(f_1(\mathbf{x}), f_2(\mathbf{x}), \ldots, f_O(\mathbf{x}))$, $\forall \mathbf{x}$, en el entorno $\|\mathbf{x} - \mathbf{x}^*\| < \varepsilon$.
- Óptimo global: un vector de decisión factible $\mathbf{x}^*$ representa un óptimo global de la función $\mathrm{G}(f_1(\mathbf{x}), f_2(\mathbf{x}), \ldots, f_O(\mathbf{x}))$ si $\mathrm{G}(f_1(\mathbf{x}^*), f_2(\mathbf{x}^*), \ldots, f_O(\mathbf{x}^*))$ no es peor que $\mathrm{G}(f_1(\mathbf{x}), f_2(\mathbf{x}), \ldots, f_O(\mathbf{x}))$, $\forall \mathbf{x} \in \Omega$.
- Punto de silla: un vector de decisión factible $\mathbf{x}^* = [\mathbf{x}_A^*, \mathbf{x}_B^*]$ representa un punto de silla de la función $\mathrm{G}(f_1(\mathbf{x}), f_2(\mathbf{x}), \ldots, f_O(\mathbf{x}))$ si existe $\varepsilon > 0$ tal que $\mathrm{G}(f_1([\mathbf{x}_A, \mathbf{x}_B^*]), f_2([\mathbf{x}_A, \mathbf{x}_B^*]), \ldots,$ no es peor que $\mathrm{G}(f_1([\mathbf{x}_A^*, \mathbf{x}_B^*]), f_2([\mathbf{x}_A^*, \mathbf{x}_B^*]), \ldots, f_O([\mathbf{x}_A^*, \mathbf{x}_B^*]))$ y éste a su vez no es peor que $\mathrm{G}(f_1([\mathbf{x}_A^*, \mathbf{x}_B]), f_2([\mathbf{x}_A^*, \mathbf{x}_B]), \ldots, f_O([\mathbf{x}_A^*, \mathbf{x}_B]))$, $\forall \mathbf{x}$ en $\|[\mathbf{x}_A, \mathbf{x}_B] - [\mathbf{x}_A^*, \mathbf{x}_B^*]\| < \varepsilon$.

En cuanto a los problemas, atendiendo a distintos criterios, se pueden clasificar según:

1. Existencia de restricciones
   a) Sin restricciones: en Ω sólo se define el tipo de variable.
   b) Con reestricciones: en Ω se definen restricciones $g_j(\mathbf{x}) \leqslant 0$, o $h_k(\mathbf{x}) = 0$.
2. Naturaleza de las variables de decisión
   a) Estáticos: la variable tiempo no es relevante.
   b) Dinámicos: el tiempo es la variable de la que dependen el resto de variables.
3. Número de objetivos
   a) Mono-objetivo: se optimiza un sólo objetivo.
   b) Multi-objetivo: se optimizan varios objetivos.
4. Naturaleza de las funciones involucradas
   a) Problemas lineales: todas las funciones objetivo de las restricciones son lineales.
   b) Problemas no lineales: alguna de las funciones objetivo o restricciones no es lineal.
   c) Problemas de programación geométrica: las funciones objetivo y las restricciones son expresables en forma de polinomios de las variables de decisión.
   d) Problemas cuadráticos: caso especial de problema no lineal en el que las funciones objetivo son cuadráticas y las restricciones son lineales.
5. Valores permisibles de las variables de decisión
   a) Problemas de programación reales: las variables de decisión toman cualquier valor real.
   b) Problemas de programación entera: las variables de decisión toman sólo valores enteros (o discretos). Como caso particular, se tienen las de variable booleana (0 ó 1).
   c) Problemas de programación mixta: una mixtura de los anteriores.
6. Naturaleza determinista de las variables de decisión
   a) Problemas deterministas: todos los parámetros del problema se conocen con certeza y, por tanto, se pueden evaluar las funciones.

   b) Problemas estocásticos: existe incertidumbre sobre alguno o todos los parámetros del problema, y se tienen que dar definiciones probabilísticas para algunas funciones.

7. Separabilidad de las funciones:

   a) Problemas separable: tanto las funciones objetivo como las restricciones pueden separarse en suma de funciones de diferentes variable de decisión. Si la suma se realiza de manera independiente sobre cada una de las variables de decisión, se dice que los problemas son totalmente separables.

   b) Problemas no separables: caso contrario al anterior.

8. Complejidad computacional

   a) Problemas de clase P: se conocen algoritmos capaces de obtener la solución óptima del problema en un tiempo polinomial, según el tamaño del problema. Se suele expresar O(p(n)), con n el tamaño, p(·) polinomio, O(·) orden de cómputo.

   b) Problemas de clase NP: un algoritmo no puede resolverlo en tiempo polinomial, o bien no se conoce. A su vez, pueden ser NP-hard o NP-completos. Existen dos vías para enfrentar estos problemas: o bien utilizar un algoritmo que asegure la optimalidad pero el tiempo de cómputo sea excesivo, o bien encontrar una solución no óptima pero aceptable, en un tiempo razonable.

En este último criterio, los algoritmos del primer grupo se conocen como exactos y los del segundo grupo, aproximados. Algunos de estos algoritmos aproximados se los denomina heurísticos.

2.3. Algoritmos Heurísticos

En inteligencia artificial, el término heurístico hace referencia, en especial, a un procedimiento que intenta aportar soluciones razonablemente buenas teniendo en cuenta la calidad de las soluciones y los recursos empleados. En investigación operativa es similar, pero no se asegura la optimalidad ni la factibilidad de las soluciones, tampoco lo cerca de ellas que se está. También se usa el término heurística cuando se realizan modificaciones en el procedimiento de la solución en pos de mejorar su rendimiento.

En el contexto de las metaheurísticas, este término se utiliza para referirse a algoritmos heurísticos que integran un conjunto de mecanismos de control diseñados para optimizar el funcionamiento de la heurística y mejorar su rendimiento. Otra acepción sería un algoritmo basado en procedimientos heurísticos "inteligentes".

Las propiedades que se desea que posean estos algoritmos son sencillez, precisión (sin ambigüedades), eficiente (da soluciones, son de alta calidad y se aprovechan los recursos), general (aplicable a muchos problemas), robustez (su comportamiento apenas varía con modificaciones o cambio de contexto), multiplicidad de soluciones (para elección del usuario), flexibilidad en los parámetros y restricciones y, finalmente, modelable (para particularidades y generalidades de cada problema).

Atendiendo al modo en el que buscan y construyen sus soluciones, las cuatro grandes familias son:

1. Metaheurísticas de relajación: se resuelve versiones menos restrictivas y, por tanto, más fáciles de resolver que el problema original.
2. Metaheurísticas constructivas: se obtiene la solución del problema a partir del análisis y selección gradual de las componentes que la forman.
3. Metaheurísticas de búsqueda: se recorre el espacio de soluciones mediante transformaciones y movimientos que aprovechan la estructura del problema.
4. Metaheurísticas evolutivas: los valores de un conjunto de soluciones evolucionan de forma paralela, acercándose según aumenta el número de iteraciones, a la solución óptima.

2.3.1. Metaheurísticas de búsqueda

Las metaheurísticas de búsqueda establecen estrategias para recorrer el espacio de soluciones del problema transformando de forma iterativa y a través de reglas astutas una solución inicial en otra.

A su vez, hay dos grandes familias: las técnicas de búsqueda local y las técnicas de búsqueda global.

Por un lado, las técnicas de búsqueda local se basan en alguna regla inteligente que modifica una solución del problema. Ésta se aplica de forma iterativa hasta que no se pueda mejorar la solución previa. Las modificaciones sólo permiten trasladar una solución actual a una de su vecindad. Se dejarán de realizar estas modificaciones si no se encuentra ninguna solución mejor que la de partida. Si la hubiera, se utilizará la mejor solución hallada para la solución de partida de la siguiente iteración. Este tipo de técnicas están relacionadas con algoritmos de búsqueda monótona o con técnicas de escalado (*hill-climbing*). Además, al elegir la mejor solución posible determinada por la regla elegida, convierte estas metaheurísticas en algoritmos voraces (*greedy*). La desventaja primordial de las búsquedas locales es que pueden quedar fácilmente atrapadas en un óptimo local. Esto es, que no puede mejorarse una solución mediante las transformaciones de la heurística.

Por otro lado, las metaheurísticas de búsqueda global pretenden eliminar la localidad extendiendo la búsqueda de los algoritmos a otras regiones del espacio. Esto se consigue mediante varias mecánicas, como son las búsquedas de arranque múltiple (*Multi Start Search*), en la que se reinicia la búsqueda a partir de otra solución inicial distinta; las búsquedas por entorno variable (*Variable Neighbourhood Search, VNS*), en las que se modifica el tipo de movimiento en el entorno definido por la regla para evitar quedar atrapada en una región rígida; búsquedas no monótonas, en las que se admite que el movimiento empeore durante la búsqueda ya sea bien utilizando memoria para que con la información adquirida no se caiga en una zona concreta del espacio, evitando momentáneamente soluciones muy parecidas entre sí, como en el la búsqueda tabú (*Taboo Search, TS*, o bien se establecen normas para controlar la probabilidad de aceptar una solución igual o peor, como en el enfriamiento simulado (*Simulated Annealing*), en el que la probabilidad de aceptación viene dada por una función exponencial del empeoramiento producido.

2.3.2. Metaheurísticas evolutivas

Las metaheurísticas evolutivas establecen estrategias para guiar la evolución de un conjunto de posibles soluciones, al que llamaremos población, hacia la solución óptima.

La exploración del espacio de soluciones se hace en cada iteración con una población, modificando cada individuo teniendo en cuenta los valores del resto y las funciones de ajuste asociadas. Esto último no es estrictamente necesario, pues habrá regiones no explorables con la información exclusiva de la población, por lo que se pueden añadir mecanismos adicionales. En cada iteración, se sustituye la población con nuevas soluciones y, en ocasiones, parte de las ya existentes. En algunos algoritmos se guardarían los mejores individuos para tenerlos en cuenta posteriormente. La distinción entre metaheurísticas evolutivas se hace respecto a cómo combinan la información obtenida de la población.

Entre otros, existen tres grandes grupos:

Algoritmos genéticos (*Genetic Algorithms, GA*), en los que en cada iteración (generación) mantiene una población de posibles soluciones que evolucionan hacia las soluciones óptimas del problema mediante una selección de individuos (selección natural) y operadores genéticos (cruce y mutación). Se inspira en la teoría de la evolución de Darwin.

Nubes de partículas (*Particle Swarm Optimization, PSO*), en las que se mantiene un conjunto de posibles soluciones (partículas) que son desplazadas por el espacio de búsqueda teniendo en cuenta el desplazamiento anterior, el mejor valor hallado por la partícula (líder u mejor local) y el mejor valor hallado por todas las partículas (líder o mejor global). Está inspirado en el comportamiento de bandadas de aves o bancos de peces.

Evolución diferencial (*Differential Evolution, DE*), en la que a lo largo de las iteraciones, el conjunto de posibles soluciones son desplazadas incrementalmente (diferencial) teniendo en cuenta las diferencias entre pares de soluciones del conjunto, adaptándose a la diversidad de soluciones existentes. Los métodos de escalada o gradiente son de este estilo.

Capítulo 3

Métodos de simulación de Monte-Carlo

Este capítulo se centra en la introducción de varios métodos de integración sobre una función de probabilidad combinando la integración de Monte Carlo con alguna técnica de simulación estocástica. Además, se presentarán otros dos grupos de métodos de este tipo para la generación de muestras que, junto con lo anterior, permiten calcular el valor de integrales sobre funciones de probabilidad.

Para estudiar este método en profundidad, seguiremos la siguiente notación: U representa una variable aleatoria y $\boldsymbol{U}$ un conjunto de ellas, y en minúscula, u, $\boldsymbol{u}$, para valores específicos, respectivamente. $\boldsymbol{V}$ será el conjunto de todas las variables del problema, siendo X, x, $\boldsymbol{X}$, $\boldsymbol{x}$, las variables ocultas (desconocidas) y Y, y, $\boldsymbol{Y}$, $\boldsymbol{y}$, las observadas (medidas). Para las variables de un conjunto, se usará un subíndice ($U_g \in \boldsymbol{U}$), y para un subconjunto de variables indexadas usaremos un subíndice *k:s*. Para el i-ésimo valor de una muestra se indicará con el superíndice (i) y para su peso, $w(\cdot)$.

En cuanto a funciones, la densidad de probabilidad original sobre la que se desea integrar será $p(\cdot)$, mientras que $f(\cdot)$ será una función genérica. Además, una densidad de probabilidad genérica será $Pr(\cdot)$, la densidad auxiliar para obtener las muestras se denotará $q(\cdot)$, la operación de muestreo respecto a una densidad $s(\boldsymbol{U})$ como $\boldsymbol{u}^{(i)} \sim s(\boldsymbol{U})$ y finalmente la función delta de Dirac, $\delta(\cdot)$.

3.1. Integración de Monte-Carlo

Esta técnica consigue aproximar el resultado de la integral 3.1, valor esperado de una serie de funciones, útil para la resolución de problemas de aprendizaje, cuyo cálculo exacto se reduce a un conjunto muy específico de funciones.

$$I = \int f(\boldsymbol{U})\, p(\boldsymbol{U})\, d\boldsymbol{U} \tag{3.1}$$

Para evadir el problema de otros métodos, el cual reside en el espaciado correcto de la malla temporal para evaluación de esos puntos, pues no todos los nodos podrían corresponder

con momentos en los que $p(\boldsymbol{U})$ tome valores significativos, la técnica de integración de Monte Carlo evalúa $f(\boldsymbol{U})$ sobre un conjunto $\left\{\boldsymbol{u}^{(1)}, \boldsymbol{u}^{(2)}, \ldots, \boldsymbol{u}^{(N)}\right\}$ distribuidas según $p(\boldsymbol{U})$.

El método de integración, pues, aproxima 3.1 por medio de la siguiente expresión:

$$I_N = \frac{1}{N}\sum_{i=1}^{N} f\left(\boldsymbol{u}^{(i)}\right) \quad con \quad \boldsymbol{u}^{(i)} \sim p(\boldsymbol{U}) \tag{3.2}$$

lo que permite evaluar $f(\boldsymbol{U})$ en puntos de $\boldsymbol{U}$donde la probabilidad no sea nula. Esto se consigue mediante la aproximación de la función original $p(\boldsymbol{U})$ como sigue:

$$p_N(\boldsymbol{U}) = \frac{1}{N}\sum_{i=1}^{N} \delta\left(\boldsymbol{U} - \boldsymbol{u}^{(i)}\right) \quad con \quad \boldsymbol{u}^{(i)} \sim p(\boldsymbol{U}) \tag{3.3}$$

Sustituyendo en 3.1 $p(\boldsymbol{U})$ por $p_N(\boldsymbol{U})$ para comprobar la veracidad de la aproximación:

$$I = \int f(\boldsymbol{U})\, p(\boldsymbol{U})\, d\boldsymbol{U} \simeq \int f(\boldsymbol{U})\, p_N(\boldsymbol{U})\, d\boldsymbol{U} =$$

$$= \frac{1}{N}\sum_{i=1}^{N} \int f(\boldsymbol{U})\, \delta\left(\boldsymbol{U} - \boldsymbol{u}^{(i)}\right) d\boldsymbol{U} = \frac{1}{N}\sum_{i=1}^{N} f\left(\boldsymbol{u}^{(i)}\right) = I_N$$

A través de la ley fuerte de los grandes números se puede comprobar que si $N \to \infty$, entonces, $I_N \xrightarrow{c.s.} I$. También que para ciertos pares de funciones $f(\boldsymbol{U})$, $p(\boldsymbol{U})$, la varianza de I_N decrece según aumenta N.

3.2. Simulación estocástica

La simulación estocástica es el mecanismo responsable de generar el conjunto de muestras y, según sea su naturaleza, distinguir unos métodos de simulación de Monte Carlo de otros.

3.3. Técnicas de generación de muestras

Las siguientes técnicas forman parte de un amplio conjunto de métodos de muestreo auxiliar. Será clave la noción de *soporte* de una función $p(\boldsymbol{U})$, conjunto de regiones donde sucede $p(\boldsymbol{U}) > 0$. Generalmente, es necesario poder evaluar dicha función o, al menos, un múltiplo de ella.

3.3.1. Muestreo por rechazo

El muestreo por rechazo (*Rejection Sampling, RS*) genera un conjunto de muestras distribuidas según la función de densidad de probabilidad original $p(\boldsymbol{U})$ a través de:

1. Función de densidad auxiliar $q(\boldsymbol{U})$ para generar los candidatos $\boldsymbol{u}^{(i)}$.

2. Función de probabilidad uniforme $\mathcal{U}_{(0,1)}(A)$ para decidir si los candidatos $\boldsymbol{u}^{(i)}$ son aceptados con una probabilidad proporcional a la diferencia de $p\left(\boldsymbol{u}^{(i)}\right)$ y $q\left(\boldsymbol{u}^{(i)}\right)$.

La función auxiliar debe satisfacer $Mq(\boldsymbol{U}) - p(\boldsymbol{u}) > 0$, con $0 < M < \infty$, lo que fuerza a que tenga el mismo soporte que la original. Así, $\boldsymbol{u}^{(i)} \sim q(\boldsymbol{U})$ es aceptado si $a^{(i)} Mq\left(\boldsymbol{u}^{(i)}\right) < p\left(\boldsymbol{u}^{(i)}\right)$. En otro caso se rechaza y los valores de $\boldsymbol{u}^{(i)}$ y de $a^{(i)}$ se reemplazan por otro par.

$i = 1$
while $(i \leqslant N)$
  $\boldsymbol{u}^{(i)} \sim q(\boldsymbol{U}), \quad a^{(i)} \sim \mathcal{U}_{(0,1)}(A)$
  if $a^{(i)} Mq(\boldsymbol{u}^{(i)}) < p(\boldsymbol{u}^{(i)})$ then $i = i+1$

Figura 3.1: Método de muestreo por rechazo

Esta aceptación y rechazo sucede con una probabilidad $Pr\left(A|\boldsymbol{u}^{(i)}\right) = p\left(\boldsymbol{u}^{(i)}\right)/Mq\left(\boldsymbol{u}^{(i)}\right)$ y de $1 - Pr\left(A|\boldsymbol{u}^{(i)}\right)$, respectivamente. Por tanto, cuanto más cercanos sean $p(\boldsymbol{U})$ y $Mq(\boldsymbol{U})$ en el soporte de $p(\boldsymbol{U})$ más eficiente será el método. Es conveniente elegir $q(\boldsymbol{U})$ de modo que permita fijar valores pequeños de M, pues $Pr(A) = \int Pr(A|\boldsymbol{U})\, q(\boldsymbol{U})\, d\boldsymbol{U} = \dfrac{1}{M}\displaystyle\int p(\boldsymbol{U})\, d(\boldsymbol{U}) = \frac{1}{M}$.

También se da que $Pr(\boldsymbol{U}|A) = \dfrac{Pr(A|\boldsymbol{U})\, q(\boldsymbol{U})}{Pr(A)} = \dfrac{p(\boldsymbol{U})\, q(\boldsymbol{U})\, M}{Mq(\boldsymbol{U})} = p(\boldsymbol{U})$, luego $\boldsymbol{u}^{(i)}$ se distribuyen según la función $p(\boldsymbol{U})$ original.

Cabe destacar que el método de muestreo por rechazo es igualmente válido si se conoce una expresión de la forma $Cte \cdot p(\boldsymbol{U})$. Sin embargo, no es sencillo de implementar en problemas multidimensionales complejos, pues es difícil encontrar $q(\boldsymbol{U})$ y M adecuados para generar pocos rechazos, problema solventable haciendo uso de la versión adaptativa de este mismo método.

3.3.2. Muestreo enfatizado

En el muestreo enfatizado (*Importance Sampling, IS*), las muestras $\boldsymbol{u}^{(i)}$ se toman de una función de densidad auxiliar $q(\boldsymbol{U})$ con soporte igual o mayor que el de $p(\boldsymbol{U})$. Los pesos $w\left(\boldsymbol{u}^{(i)}\right)$ representan la proporción de la muestra dentro de $p(\boldsymbol{U})$, miden la disonancia entre la densidad original y la auxiliar. Por tanto, al tener que $w\left(\boldsymbol{u}^{(i)}\right) \propto \dfrac{p\left(\boldsymbol{u}^{(i)}\right)}{q\left(\boldsymbol{u}^{(i)}\right)}$, es suficiente con conocer una expresión proporcional a la densidad original ($Cte \cdot p(\boldsymbol{U})$). En este caso, $p(\boldsymbol{U})$ e I de 3.1 se aproximan como:

$$
\begin{aligned}
& p_{IS} = \sum_{i=1}^{N} w\left(\boldsymbol{u}^{(i)}\right) \delta\left(\boldsymbol{U} - \boldsymbol{u}^{(i)}\right); \quad I_{IS} = \sum_{i=1}^{N} w\left(\boldsymbol{u}^{(i)}\right) f\left(\boldsymbol{u}^{(i)}\right) \\
& \text{con} \quad \boldsymbol{u}^{(i)} \sim q(\boldsymbol{U}), \quad w\left(\boldsymbol{u}^{(i)}\right) \propto \frac{p\left(\boldsymbol{u}^{(i)}\right)}{q\left(\boldsymbol{u}^{(i)}\right)}, \quad \sum_{i=1}^{N} w\left(\boldsymbol{u}^{(i)}\right) = 1
\end{aligned}
\tag{3.4}
$$

El pseudocódigo de la figura 3.2 recoge el algoritmo para la generación de N muestras $\boldsymbol{u}^{(i)}$ con sus correspondientes pesos $w\left(\boldsymbol{u}^{(i)}\right)$.

$T = 0$

for $(i = 1 : N)$

$$\boldsymbol{u}^{(i)} \sim q(\boldsymbol{U}), \quad w\left(\boldsymbol{u}^{(i)}\right) = \frac{p\left(\boldsymbol{u}^{(i)}\right)}{q\left(\boldsymbol{u}^{(i)}\right)}, \quad T = T + w\left(\boldsymbol{u}^{(i)}\right)$$

for $(i = 1 : N)$

$$w\left(\boldsymbol{u}^{(i)}\right) = \frac{w\left(\boldsymbol{u}^{(i)}\right)}{T}$$

Figura 3.2: Método genérico de muestreo enfatizado

Las ecuaciones de 3.4 se derivan de la expresión 3.1 introduciendo la función de muestreo auxiliar $q\,(\boldsymbol{U}($. Además, se considera que la función de muestreo $w\,(\boldsymbol{U})\,q\,(\boldsymbol{U})$ puede aproximarse mediante $p_{IS}\,(\boldsymbol{U})$ combinando las probabilidades de densidad puntual $\delta\left(\boldsymbol{U} - \boldsymbol{u}^{(i)}\right)$ centradas en las N muestras $\boldsymbol{u}^{(i)}$ obtenidas de manera independiente con la función auxiliar $q\,(\boldsymbol{U})$ y pesadas según $w\left(\boldsymbol{u}^{(i)}\right)$.

$$I = \int f\,(\boldsymbol{U})\,p\,(\boldsymbol{U})\,d\boldsymbol{U} = \int f\,(\boldsymbol{U})\,\frac{p\,(\boldsymbol{U})}{q\,(\boldsymbol{U})} q\,(\boldsymbol{U})\,d\boldsymbol{U} = \int f\,(\boldsymbol{U})\,w\,(\boldsymbol{U})\,q\,(\boldsymbol{U})\,d\boldsymbol{U} \simeq$$
$$\simeq \int f\,(\boldsymbol{U})\,w\,(\boldsymbol{U})\,\frac{1}{N}\sum_{i=1}^{N}\delta\left(\boldsymbol{U} - \boldsymbol{u}^{(i)}\right)d\boldsymbol{U} = \int f\,(\boldsymbol{U})\,\frac{1}{N}\sum_{i=1}^{N} w\left(\boldsymbol{u}^{(i)}\right)\delta\left(\boldsymbol{U} - \boldsymbol{u}^{(i)}\right)d\boldsymbol{U}$$

La elección de $q\,(\boldsymbol{U})$ hace controlar la región de donde se obtienen las muestras y así introducir cualquier tipo de conocimiento previo que se tenga de la original $p\,(\boldsymbol{U})$. Además, como se quiere aproximar el valor integral, es conveniente minimizar la varianza. Para ello, la función auxiliar óptima es $q^*\,(\boldsymbol{U}) = \dfrac{|f\,(\boldsymbol{U})|p\,(\boldsymbol{U})}{\int|f\,(\boldsymbol{U})|p\,(\boldsymbol{U})}$, cuyo valor minimiza la varianza $E_{q(\boldsymbol{U})}\left[w^2\,(\boldsymbol{U})\,f^2\,(\boldsymbol{U})\right]$. De ella no se puede muestrear directamente pero nos señala que se pierde eficiencia si el numerador se anula. Si se quiere obtener los valores integrales para varias funciones generales $f\,(\boldsymbol{U})$, se puden emplear varias o una sóla $q\,(\boldsymbol{U})$, que ignore las funciones generales e intente aproximarse a $p\,(\boldsymbol{U})$.

Sin embargo, la elección de una función auxiliar eficiente es más complicado en el caso multidimensional. Exsisten técnicas para sortear este inconveniente, como el muestreo por importancia adaptativo o el muestreo enfatizado secuencial (*Sequential Importance Sampling, SIS*), implementado en los filtros de partículas.

3.3.3. Remuestreo por pesos

En el método de remuestreo por pesos (*Weighted Resampling, WR*) se aproxima una función de densidad $p_A\,(\boldsymbol{U})$, que es una combinación de N_A funciones de densidad puntuales

$\delta\left(\boldsymbol{U}-\boldsymbol{u}_A^{(i)}\right)$, centradas en $\boldsymbol{u}_A^{(i)}$ y ponderadas según $w\left(\boldsymbol{u}_A^{(i)}\right)$, por otra función $p_B\left(\boldsymbol{U}\right)$ de la misma clase.

Los centros de $p_B\left(\boldsymbol{U}\right)$ son muestreados entre los valores de $\boldsymbol{u}_A^{(i)}$ mediante $r\left(\boldsymbol{U}\right)$, la cual indica en qué regiones del espacio generar una determinada cantidad de muestras, y la función de probabilidad $q\left(I|\boldsymbol{u}_A^{(i)}\right)$ que elige las muestras de forma proporcional a su correspondiente valor $r\left(\boldsymbol{u}_A^{(i)}\right)$. Los pesos $w\left(\boldsymbol{u}_B^{(j)}\right)$ son el cociente de los pesos en la densidad original, $w\left(\boldsymbol{u}_A^{(i)}\right)$, entre el valor $r\left(\boldsymbol{u}_A^{(i)}\right)$, el cual toman toman en la función de remuestreo $\left(\boldsymbol{u}_B^{(j)}=\boldsymbol{u}_A^{(i)}\right)$.

$$
\begin{aligned}
&p_A\left(\boldsymbol{U}\right)=\sum_{i=1}^{N_A} w\left(\boldsymbol{u}_A^{(i)}\right)\delta\left(\boldsymbol{U}-\boldsymbol{u}_A^{(i)}\right);\quad p_B\left(\boldsymbol{U}\right)=\sum_{i=1}^{N_B} w\left(\boldsymbol{u}_B^{(j)}\right)\delta\left(\boldsymbol{U}-\boldsymbol{u}_B^{(j)}\right)\\
&\text{con}\quad \boldsymbol{u}_A^{(i)}=\boldsymbol{u}_B^{(j)},\quad i\sim q\left(I|\boldsymbol{u}_A^{(i)}\right),\quad q\left(I|\boldsymbol{u}_A^{(i)}\right)\propto r\left(\boldsymbol{u}_A^{(i)}\right)\\
&w\left(\boldsymbol{u}_B^{(j)}\right)\propto\frac{w\left(\boldsymbol{u}_A^{(i)}\right)}{r\left(\boldsymbol{u}_A^{(i)}\right)},\quad \sum_{i=1}^{N_B} w\left(\boldsymbol{u}_B^{(j)}\right)=1
\end{aligned}
\tag{3.5}
$$

El pseudocódigo de la figura 3.3 ilustra la generación de N muestras $w\left(\boldsymbol{u}_B^{(j)}\right)$ con sus correspondientes pesos $w\left(\boldsymbol{u}_B^{(j)}\right)$.

$T=0$
for $(j=1:N)$
  $I\sim q(I|\boldsymbol{u}_A^{(i)})$,

  $\boldsymbol{u}_B^{(j)}=\boldsymbol{u}_A^{(i)},\quad w\left(\boldsymbol{u}_B^{(j)}\right)=\dfrac{w\left(\boldsymbol{u}_A^{(i)}\right)}{r\left(\boldsymbol{u}_A^{(i)}\right)},\quad T=T+w\left(\boldsymbol{u}_B^{(j)}\right)$

for $(j=1:N)\quad w\left(\boldsymbol{u}_B^{(j)}\right)=\dfrac{w\left(\boldsymbol{u}_B^{(j)}\right)}{T}$

Figura 3.3: Método genérico de remuestreo por pesos

Estas aproximaciones garantizan que, cuando $N\to\infty$, ambas distribuciones converjan a condición de que el soporte de $r\left(\boldsymbol{U}\right)$ contenga al de $p_A\left(\boldsymbol{U}\right)$.

Además de elegir la función $r\left(\boldsymbol{U}\right)$, cuando se utiliza un método de remuestreo por pesos, es necesario seleccionar el método de muestreo proporcional $q\left(I|\boldsymbol{u}_A^{(i)}\right)$, usualmente el muestreo multinomial, el sistemático y el residual, los cuales no trataremos en esta sección.

La base del método de remuestreo por pesos se asemeja a la empleada por el método de muestreo enfatizado, dado que al incorporar una nueva función de muestreo proporcional a $r\left(\boldsymbol{U}\right)$, esta se encuentra en el denominador de la función que determina los pesos

$w\left(\boldsymbol{u}_B^{(j)}\right)$. La libertad de elección de $r\left(\boldsymbol{U}\right)$ nos permite generar muestras en del soporte que merecen ser estudiadas en profundidad. Es habitual elegir el valor de $r\left(\boldsymbol{u}_A^{(i)}\right)$ igual al de los pesos $w\left(\boldsymbol{u}_A^{(i)}\right)$ con el fin de que las nuevas muestras $\boldsymbol{u}_B^{(j)}$ se concentren en las regiones más probables de la función de densidad original y que $w\left(\boldsymbol{u}_B^{(j)}\right) = \frac{1}{N}$, lo cual cobra sentido en la creación de algunos filtros de partículas, sobre los que hablaremos en la sección Filtros de Partículas.

3.4. Métodos de simulación por cadenas de Markov

El fundamento de método de muestreo mediante cadenas de Markov (*Markov Chains, MC*) reside en que la siguiente muestra, $\boldsymbol{u}^{(i)}$, es generada mediante un proceso estocástico definido por una función de probabilidad, *kernel de transición*, $K\left(\boldsymbol{U}|\boldsymbol{u}^{(i-1)}\right)$, es decir, se depende sólo de la muestra inmediatamente anterior (propiedad de Markov). Para la aproximación de la función de densidad original, se usarán las N últimas muestras como sigue:

$$p_{MC} = \frac{1}{N}\sum_{j=B}^{N+B}\delta\left(\boldsymbol{U}-\boldsymbol{u}^{(j)}\right) \quad con \quad \boldsymbol{u}^{(j)} \sim K\left(\boldsymbol{u}^{(j)}|\boldsymbol{u}^{(j-1)}\right) \tag{3.6}$$

El pseudocódigo de la figura 3.4 muestra cómo se generan las $B+N$ muestras, de las cuales se desechan las B primeras.

$i = 1, \quad j = 1, \quad$ set $\boldsymbol{u}^{(0)}$
while $(i \leqslant N)$
  $\boldsymbol{u}^{(i)} \sim K\left(\boldsymbol{U} \mid \boldsymbol{u}^{(i-1)}\right)$
  if $j \geqslant B$ then $i = i + 1$,
  else $\boldsymbol{u}^{(i-1)} = \boldsymbol{u}^{(i)}, \quad j = j + 1$

Figura 3.4: Método genérico de muestreo por cadena de Markov

Para que el proceso definido con el kernel de transición, conocido como cadena de Markov, genere las muestras acorde a la función de probabilidad original $p\left(\boldsymbol{U}\right)$ se requiere de:

1. Convergencia de la cadena de Markov tras la toma de unas muestras iniciales posteriormente deshechadas.
2. La función de probabilidad estacionaria a la que converja sea $p\left(\boldsymbol{U}\right)$.

Esto es, que $K\left(\boldsymbol{U}|\boldsymbol{u}^{(i-1)}\right)$ cumpla las condiciones de ser irreducible (visita todo el soporte en tiempo limitado), aperiódica (no fluctuación entre distintas regiones del soporte de forma periódica) y recurrente de forma positiva (para la existencia de función estacionaria).

Una condición suficiente para 2 es que la cadena sea reversible y, por tanto, en el momento estacionario la velocidad de la transición sea igual en ambos sentidos:

$$p\left(\boldsymbol{u}^{(i-1)}\right) K\left(\boldsymbol{u}^{(i)}|\boldsymbol{u}^{(i-1)}\right) = p\left(\boldsymbol{u}^{(i)}\right) K\left(\boldsymbol{u}^{(i-1)}|\boldsymbol{u}^{(i)}\right) \tag{3.7}$$

Una función de probabilidad $p\left(\boldsymbol{U}\right)$ puede dar lugar a distintas cadenas de Markov, cada una con una velocidad a priori distinta, lo que hace más beneficioso el uso de unas frente a otras. Al tomar las últimas N muestras, antes de la converjencia las B primeras se suelen deshechar (*burn-in*), luego la aproximación de $p\left(\boldsymbol{U}\right)$ se encuentra en el rango entre B y $B+N$. La elección del valor inicial de $\boldsymbol{u}^{(0)}$ influye en el valor del *burn-in* lo que a su vez repercute, además del número necesario de muestras N, en el error de la aproximación mediante p_{MC}. Por ello, se suelen emplear otras técnicas avanzadas para su obtención. También es posible generar los datos utilizando cadenas independientes y/o utilizar los subconjuntos de muestras obtenidas una vez dada la convergencia. Finalmente, pese a la dificultad de construcción de una cadena $K\left(\boldsymbol{U}|\boldsymbol{u}^{(i-1)}\right)$, existen algunas cadenas generales ya definidas dentro de algunos métodos con un uso amplio plausible.

3.4.1. Cadena general para los métodos de Metrópolis-Hastings y Gibbs

El tipo de cadena elegida para estos procedimientos permite atacar problemas en los que se conoce $Cte{\cdot}p\left(\boldsymbol{U}\right)$ en vez de $p\left(\boldsymbol{U}\right)$. La generalidad de $K\left(\boldsymbol{U}|\boldsymbol{u}^{(i-1)}\right)$ recae en la posibilidad de modelar su comportamiento mediante la densidad condicionada $q\left(\boldsymbol{U}|\boldsymbol{u}^{(i-1)}\right)$.

La cadena genera valores para $\boldsymbol{u}^{(i)}$ a partir de la densidad condicionada anterior, con probabilidad de aceptación $\Pr\left(A|\,\boldsymbol{U},\boldsymbol{u}^{(i-1)}\right)$, e iguales a $\boldsymbol{u}^{(i-1)}$ con probabilidad de rechazo de $\Pr\left(\neg A|\boldsymbol{u}^{(i-1)}\right)$. La probabilidad de aceptación de $\Pr\left(A|\,\boldsymbol{U},\boldsymbol{u}^{(i-1)}\right)$ está supeditada a a los valores de $p(\cdot)$, $q(\cdot)$, evaluadas de manera directa e inversa sobre los valores de $\boldsymbol{U}=\boldsymbol{u}^{(i)}$ y $\boldsymbol{u}^{(i-1)}$.

Así, por construcción, se asegura que la cadena $K\left(\boldsymbol{U}|\boldsymbol{u}^{(i-1)}\right)$ sea reversible (3.7) y por tanto que la función de probabilidad estacionaria sea $p\left(\boldsymbol{U}\right)$. Además, forzando que el soporte de $q\left(\boldsymbol{U}|\boldsymbol{u}^{(i-1)}\right)$ incluya el soporte de $p\left(\boldsymbol{U}\right)$ se logra la irreductibilidad y la aperiodicidad por la posibilidad de no aceptar la muestra candidata mediante $q\left(\boldsymbol{U}|\boldsymbol{u}^{(i-1)}\right)$. Por tanto, con estas elecciones la cadena $K\left(\boldsymbol{U}|\boldsymbol{u}^{(i-1)}\right)$ consigue generar muestras según $p\left(\boldsymbol{U}\right)$.

Dicha cadena se define de la siguiente manera:

$$\begin{aligned} K\left(\boldsymbol{U}|\boldsymbol{u}^{(i-1)}\right) &= q\left(\boldsymbol{U}|\boldsymbol{u}^{(i-1)}\right) Pr\left(A|\,\boldsymbol{U},\boldsymbol{u}^{(i-1)}\right) + \\ &\quad + \delta\left(\boldsymbol{U}-\boldsymbol{u}^{(i-1)}\right) Pr\left(\neg A|\boldsymbol{u}^{(i-1)}\right) \end{aligned}$$

$$\begin{aligned} \text{con}\quad \Pr\left(A|\,\boldsymbol{U},\boldsymbol{u}^{(i-1)}\right) &= \text{mín}\left\{1,\frac{p\left(\boldsymbol{U}\right)q\left(\boldsymbol{u}^{(i-1)}|\,\boldsymbol{U}\right)}{p\left(\boldsymbol{u}^{(i-1)}\right)q\left(\boldsymbol{U}|\boldsymbol{u}^{(i-1)}\right)}\right\} \\ \Pr\left(\neg A|\boldsymbol{u}^{(i-1)}\right) &= \int Pr\left(\neg A|\,\boldsymbol{U},\boldsymbol{u}^{(i-1)}\right) q\left(\boldsymbol{U}|\boldsymbol{u}^{(i-1)}\right) d\boldsymbol{U} \\ \Pr\left(\neg A|\,\boldsymbol{U},\boldsymbol{u}^{(i-1)}\right) &= 1-\Pr\left(A|\,\boldsymbol{U},\boldsymbol{u}^{(i-1)}\right) \end{aligned} \tag{3.8}$$

Al aparecer el cociente $p(\boldsymbol{U})/p\left(\boldsymbol{u}^{(i-1)}\right)$, no es necesario conocer la expresión de $p(\boldsymbol{U})$, basta con un múltiplo.

3.4.2. Algoritmos de Metropolis-Hastings

Los algoritmos de esta familia se desarrollan utilizando directamente la cadena de Markov descrita en la sección anterior, implementada en un programa que la emula. El pseudocódigo mostrado en la figura 3.5 da una posible implementación.

El soporte de la densidad $q\left(\boldsymbol{U}|\boldsymbol{u}^{(i-1)}\right)$ debe incluir al de la original $p(\boldsymbol{U})$ y la elección de ésta influirá a la velocidad de convergencia del método.

Por un lado, una variante de esta familia de algoritmos es la de Metropolis, en la que la densidad auxiliar se toma simétrica, $q\left(\boldsymbol{U}|\boldsymbol{u}^{(i-1)}\right) = q\left(\boldsymbol{u}^{(i-1)}|\boldsymbol{U}\right)$, quedando entonces $\Pr\left(A|\boldsymbol{U},\boldsymbol{u}^{(i-1)}\right) = \text{mín}\left\{1, \dfrac{p(\boldsymbol{U})}{p\left(\boldsymbol{u}^{(i-1)}\right)}\right\}$.

$i = 1, \quad j = 1, \quad$ set $\boldsymbol{u}^{(0)}$

while $(i \leqslant N)$

$\boldsymbol{u}^{(i)} \sim q\left(\boldsymbol{U} \mid \boldsymbol{u}^{(i-1)}\right), \quad a^{(i)} \sim \mathcal{U}_{(0,1)}(A)$

$$\Pr\left(A|\boldsymbol{u}^{(i)},\boldsymbol{u}^{(i-1)}\right) = \text{mín}\left\{1, \frac{p\left(\boldsymbol{u}^{(i)}\right) q\left(\boldsymbol{u}^{(i-1)}|\boldsymbol{u}^{(i)}\right)}{p\left(\boldsymbol{u}^{(i-1)}\right) q\left(\boldsymbol{u}^{(i)}|\boldsymbol{u}^{(i-1)}\right)}\right\}$$

if $a^{(i)} > \Pr\left(A|\boldsymbol{u}^{(i)},\boldsymbol{u}^{(i-1)}\right)$ then $\boldsymbol{u}^{(i)} = \boldsymbol{u}^{(i-1)}$

if $j \geqslant B$ then $i = i + 1$

else $\boldsymbol{u}^{(i-1)} = \boldsymbol{u}^{(i)}, \quad j = j + 1$

Figura 3.5: Algoritmo de Metropolis-Hastings

Por otro lado, la variante conocida como muestreador independiente asume que $q\left(\boldsymbol{U}|\boldsymbol{u}^{(i-1)}\right) = q(\boldsymbol{U})$, de ahí el nombre, lo que resulta en $\Pr\left(A|\boldsymbol{U},\boldsymbol{u}^{(i-1)}\right) = \min\left\{1, \dfrac{w(\boldsymbol{U})}{w\left(\boldsymbol{u}^{(i-1)}\right)}\right\}$, siendo $w(\boldsymbol{U}) = \dfrac{p(\boldsymbol{U})}{q(\boldsymbol{U})}$, función de peso ya presentada en el método IS (3.3.2). Este método crea una dependencia entre las muestras, de modo que si el peso de una muestra $\boldsymbol{u}^{(i)}$ es mayor que la anterior, se acepta y en caso contrario, se rechaza o se acepta según lo fuerte que sea dicha discrepancia.

3.4.3. Método de Gibbs

El método de simulación de Gibbs, frecuente en problemas multivariables, consiste en una implementación en serie de la cadena de Markov de 3.4.1 y unas funciones de muestreo auxiliar $q(\cdot)$ determinadas.

Durante el proceso se actualiza cada una de las U_g de $\boldsymbol{U} = \{U_1, U_2, \ldots, U_G\}$ mediante una cadena distinta $K\left(\boldsymbol{U}_g|\boldsymbol{u}_{1:g-1}^{(i)}, \boldsymbol{u}_{g+1:G}^{(i-1)}\right)$, con una probabilidad de aceptación que se rije por la función $p\left(\boldsymbol{U}_g|\boldsymbol{u}_{1:g-1}^{(i)}, \boldsymbol{u}_{g+1:G}^{(i-1)}\right)$ y funciones auxiliares $q\left(\boldsymbol{U}_g|\boldsymbol{u}_{1:g-1}^{(i)}, \boldsymbol{u}_{g+1:G}^{(i-1)}\right)$. Para cada subíndice g, $p(\cdot)$ y $q(\cdot)$ coinciden, luego $Pr\left(A|\,\boldsymbol{U}_k, \boldsymbol{u}_{1:g-1}^{(i)}, \boldsymbol{u}_{g+1:G}^{(i-1)}\right) = 1$ y por tanto el proceso de aceptación del método de Metropolis-Hastings no es necesario, pues se aceptan todas las muestras.

El pseudocódigo que aparece en la figura 3.6 ilustra las diferentes etapas del método de Gibbs. Cada elemento del bucle interno se relaciona con una de las cadenas muestreadas en serie. El muestreado de $u_k^{(i)}$ de la función $p\left(\boldsymbol{U}_g|\boldsymbol{u}_{1:g-1}^{(i)}, \boldsymbol{u}_{g+1:G}^{(i-1)}\right)$ hace que, como se ha comentado antes, desaparezca el proceso de aceptación típico de Metropolis-Hastings.

$i = 1, \quad j = 1, \quad$ set $\boldsymbol{u}^{(0)} = \left\{u_1^{(0)}, u_2^{(0)}, \ldots, u_G^{(0)}\right\}$

while $(i \leqslant N)$

  $g = 1$

  while $(g \leqslant G)$

    $\boldsymbol{u}_g^{(i)} \sim p\left(\boldsymbol{U}_g|\boldsymbol{u}_{1:g-1}^{(i)}, \boldsymbol{u}_{g+1:G}^{(i-1)}\right), \quad g = g + 1$

  if $j \geqslant B$ then $\quad i = i + 1$

  else $\boldsymbol{u}^{(i-1)} = \boldsymbol{u}^{(i)}, \quad j = j + 1$

Figura 3.6: Método de Gibbs

A la hora de aplicar el método de Gibbs, la dificultad radica principalmente en determinar las funciones de distribución condicional $p\left(\boldsymbol{U}_g|\boldsymbol{u}_{1:g-1}^{(i)}, \boldsymbol{u}_{g+1:G}^{(i-1)}\right)$ a partir de la función de probabilidad original $p(\boldsymbol{U})$. Sin embargo, esto se puede sortear parcialmente si se modela el problema con una red bayesiana. **(la explico, en anexo...?)**

3.5. Métodos secuenciales de Monte-Carlo

Aplicables a problemas multivariables, estos métodos hacen uso de una implementación secuencial del muestreo enfatizado combinado, a elección, con una implementación secuencial del remuestreo por pesos. Por medio de las independencias condicionales de las $U_g \in \boldsymbol{U}$, generan los N valores $\boldsymbol{u}_g^{(i)}$ de U_g teniendo en cuenta los valores de $\boldsymbol{u}_k^{(i)}$ de las variables $\{U_k|k \subset \{1:G\}\}$ ya muestreadas.

En el diseño del algoritmo interfieren las independencias condicionales propias del problema a resolver y, por tanto, su implementación se adapta a las características específicas de dicho problema. Dado que el método depende del problema, resulta útil escoger un tipo particular de problema para ejemplificar el funcionamiento de esta familia de técnicas. Por ejemplo, para problemas modelables mediante redes bayesianas dinámicas genéricas,

un conjunto de métodos secuenciales son los filtros de partículas (*Particle Filters, PF*).

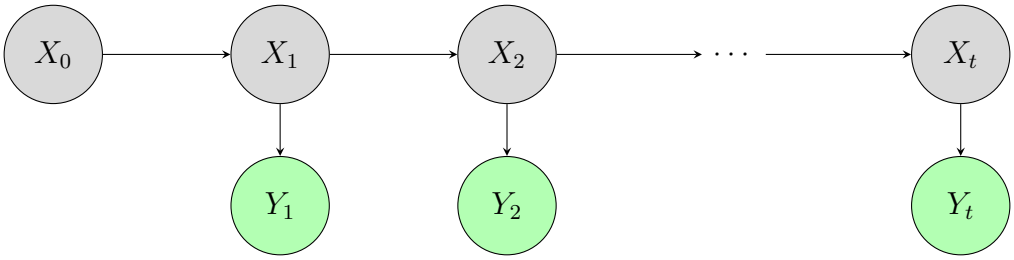

Figura 3.7: Red bayesiana para el desarrollo de los PF

En la red bayesiana 3.7 se declara que el valor de la variable oculta X_t depende únicamente del valor de X_{t-1}, y que el valor de la variable observada Y_t solamente depende de X_t. El comportamiento de cada problema se modela según la elección de las funciones $p\left(X_t|X_{t-1}\right)$, $p\left(Yt|Xt\right)$ para las dependencias anteriores. Sobre esta misma red se pueden definir varios problemas de estimación. En particular, para ejemplificar el funcionamiento se escogerá estimar $p\left(X_{0:t}|y_{1:t}\right)$ por medio de $p_N\left(X_{0:t}|y_{1:t}\right) = \sum_{i=1}^{N} w\left(x_{0:t}^{(i)}\right) \delta\left(X_{0:t} - x_{0:t}^{(i)}\right)$.

Este tipo de problemas se dan en diversas áreas del conocimiento: seguimiento de objetos, reconocimiento de habla, economía, etc. Un ejemplo del primer tipo es la determinación de la trayectoria $X_{0:t}$ seguida por una partícula móvil a partir de las medidas $y_{1:t}$ captadas por uno o varios sensores.

El tipo de problema y función de densidad a aproximar elegidos ayudan a ejemplificar los filtros de partículas frecuentes en la literatura.

3.5.1. Filtros de Partículas

El elemento principal de los PF es una implementación secuencial del método de muestreo enfatizado (*Importance Sampling, IS*, 3.3.2). Para aproximar $p\left(X_{0:t}|y_{1:t}\right)$ se usa una función de muestreo auxiliar $q\left(X_{0:t}|y_{1:t}\right)$ y se calcula el peso de $x_{0:t}^{(i)} \sim q\left(X_{0:t}|y_{1:t}\right)$ según la expresión $w\left(x_{0:t}^{(i)}\right) = \dfrac{p\left(x_{0:t}^{(i)}|y_{1:t}\right)}{q\left(x_{0:t}^{(i)}|y_{1:t}\right)}$. Utilizando el teorema de Bayes, la regla de la cadena y las relaciones de independencia condicional del problema, se manipulan las funciones de densidad para lograr la secuencialidad.

$$\begin{aligned} p\left(X_{0:t}|y_{1:t}\right) &= p\left(X_{0:t}, y_{1:t}\right) p\left(y_{1:t}\right) \quad \propto \\ &\propto \quad p\left(y_t|X_{0:t}, y_{1:t-1}\right) p\left(X_t|X_{0:t-1}, y_{1:t-1}\right) p\left(X_{0:t-1}, y_{1:t-1}\right) \\ &= p\left(y_t|X_t\right) p\left(X_t|X_{t-1}\right) p\left(X_{0:t-1}, y_{1:t-1}\right) p\left(y_{1:t-1}\right) \propto \\ &\propto \quad p\left(y_t|X_t\right) p\left(X_t|X_{t-1}\right) p\left(X_{0:t-1}, y_{1:t-1}\right) \end{aligned} \tag{3.9}$$

$$\begin{aligned} q\left(X_{0:t}|y_{1:t}\right) &= q\left(X_t|X_{0:t-1}, y_{1:t}\right) q\left(X_{0:t-1}|y_{1:t}\right) \simeq \\ &\simeq q\left(X_t|X_{0:t-1}, y_{1:t}\right) q\left(X_{0:t-1}|y_{1:t-1}\right) \end{aligned} \tag{3.10}$$

Sustituyendo en la expresión de los pesos, se ve que pesos en tiempos consecutivos están relacionados, se obtiene una función auxiliar $q(\cdot)$ para el muestreo de $x_t^{(i)}$ y las funciones de densidad condicionales $p(\cdot)$ que definen el comportamiento dinámico del problema en su totalidad.

$$\begin{aligned} w(x_{0:t}^{(i)}) &= \frac{p\left(x_{0:t}^{(i)}|y_{1:t}\right)}{q\left(x_{0:t}^{(i)}|y_{1:t}\right)} \simeq \frac{p\left(y_t|x_t^{(i)}\right) p\left(x_t^{(i)}|x_{t-1}^{(i)}\right) p\left(x_{0:t-1}^{(i)}|y_{1:t-1}\right)}{q\left(x_t^{(i)}|x_{0:t-1}^{(i)}, y_{1:t}\right) q\left(x_{0:t-1}^{(i)}, y_{1:t-1}\right)} = \\ &= \frac{p\left(y_t|x_t^{(i)}\right) p\left(x_t^{(i)}|x_{t-1}^{(i)}\right)}{q\left(x_t^{(i)}|x_{0:t-1}^{(i)}, y_{1:t}\right)} w\left(x_{0:t-1}^{(i)}\right) \end{aligned} \tag{3.11}$$

Lo anteriormente expuesto se recoge en el pseudocódigo mostrando el PF normalmente denominado *Sequential Importance Sampling, SIS*. La generación secuencial de las muestras, es decir, la obtención de $x_t^{(i)}$ correspondiente a la muestra $x_{0:t}^{(i)}$, es controlada por el bucle exterior, mientras que los valores asociados a las N muestras son controlados por el bucle interior. Las etapas opcionales muestran otras variantes, pero estos procesos no se realizan en el SIS.

Diferentes filtros de partículas tipo SIS surgen de la elección de distintas $q\left(X_t|X_{0:t-1}, y_{1:t}\right)$. La elección óptima, sin tener en cuenta la función general $f\left(X_{0:t}\right)$, es $q^*\left(X_t|X_{0:t-1}, y_{1:t}\right) = p\left(X_t|X_{t-1}, y_t\right)$. Por lo general, no es posible generar muestras a partir de la misma por lo que lo más sencillo es elegir $q\left(X_t|X_{0:t-1}, y_{1:t}\right) = p\left(X_t|x_{t-1}^{(i)}\right)$, que figura en las primeras versiones de los PF tipo SIS, útil porque facilita la generación de muestras y los cálculos de los pesos $w\left(x_{0:t}^{(i)}\right) = p\left(y_t|x_t^{(i)}\right) w\left(x_{0:t-1}^{(i)}\right)$.

Sin embargo, una elección distinta a la óptima hace que con el tiempo el número de muestras que toman un valor de peso nulo se incremente, provocando una degeneración de las muestras. Al no contribuir al cálculo integral, hace menos eficiente el algoritmo, por lo que se requiere de otras técnicas para compensarlo: se combina el SIS con etapas de remuestreo por pesos (*Weighted Resampling, WR*, 3.3.3).

Sobre el pseudocódigo de la figura 3.8, en la etapa asociada a la parte *optionally, WR for SIR*, utiliza $r\left(x_{0:t}^{(i)}\right) = w\left(x_{0:t}^{(i)}\right)$. En esta modalidad del filtro de partículas *Sampling Importance Resampling, SIR*, los pesos de las muestras tras el remuestreo son $w\left(x_{0:t}^{(i)}\right) = \frac{1}{N}$, y las muestras $x_{0:t}^{(i)}$ se acumulan en regiones donde previamente los pesos toman valores no nulos. El problema que surge de aplicar esto indiscriminadamente es el contrario al anterior, el empobrecimiento. Ahora, la variable X_k tiende a tomar el mismo valor en todas las muestras $x_{0:t}^{(i)}$, para algún $k < t$. Para evitarlo, puede aplicarse solamente a las iteraciones t en las que se detecte la degeneración mediante algún proceso auxiliar como el cálculo del número efectivo de partículas.

Además, en la parte del pseudocódigo *optionally, WR for Auxiliary PF*, se utiliza $r\left(x_{0:t-1}^{(i)}\right) = p\left(y_t|x_{0:t-1}^{(i)}\right)$, haciendo que el remuestreo considere las dependencias entre la nueva medida y los valores disponibles de la muestra. Calcular el valor de la función no es, en la mayoría

for $(i = 1 : N)$

set $x_0^{(i)}$, set $w\left(x_0^{(i)}\right)$

$t = 1$

while (true)

$F = 0$

optionally, WR for Auxiliary PF

for $(i = 1 : N)$

$x_t^{(i)} \sim q\left(X_t | X_{0:t-1}, y_{1:t}\right)$, $\dfrac{p\left(y_t|x_t^{(i)}\right) p\left(x_t^{(i)}|x_{t-1}^{(i)}\right)}{q\left(x_t^{(i)}|x_{0:t-1}^{(i)}, y_{1:t}\right)} w\left(x_{0:t-1}^{(i)}\right)$

$F = F + w\left(x_{0:t}^{(i)}\right)$

for $(i = 1 : N)$

$w\left(x_{0:t}^{(i)}\right) = \dfrac{w\left(x_{0:t}^{(i)}\right)}{F}$

optionally, WR for SIR

$t = t + 1$

Figura 3.8: Filtro de partículas Sequential Importance Sampling (SIS) con las etapas opcionales que lo convierten en Sequential Importance Resampling (SIR) y Auxiliary Particle Filter (APF).

de los casos, sencillo, de modo que se usa la aproximación 3.12. Esta etapa, WR, junto a la función de muestreo $q\left(X_t | X_{0:t-1}, y_{1:t}\right) = p\left(X_t | x_{t-1}^{(i)}\right)$, se utilizan de manera auxiliar en el conocido como *Auxiliary PF (APF)*, por lo que los pesos de las partículas al finalizar cada iteración son $w\left(x_{0:t}^{(i)}\right) \propto p\left(y_t|x_t^{(i)}\right) / p\left(y_t|m_t^{(i)}\right)$.

$$\begin{aligned} p\left(y_t|x_{0:t-1}^{(i)}\right) &= \int p\left(y_t|X_t\right) p\left(X_t|x_{t-1}^{(i)}\right) dX_t \simeq \\ &\simeq p\left(y_t|m_t^{(i)}\right) p\left(X_t|x_{t-1}^{(i)}\right) dX_t = p\left(y_t|m_t^{(i)}\right); \quad \text{con} \quad m_t^{(i)} = E_{p\left(X_t|x_{t-1}^{(i)}\right)}[X_t] \end{aligned} \tag{3.12}$$

Todas las combinaciones vistas de $q\left(X_t | X_{0:t-1}, y_{1:t}\right)$ y etapas de WR generan distintos algoritmos secuenciales de simulación con el objetivo final es generar las muestras en las zonas importantes del espacio, por lo que cualquier conocimiento previo que pueda facilitar la generación de las muestras en las regiones de interés pueden ser una guía para la elección de $q\left(X_t | X_{0:t-1}, y_{1:t}\right)$ y de las funciones $r\left(X_{0:t}\right)$ para las etapas de WR.

3.6. Métodos de simulación generales

Los métodos introducidos hasta ahora, (RS, IS, WR, MC), pueden ser caracterizados por aproximar una función de densidad original $p(\boldsymbol{U})$ mediante una auxiliar $p_N(\boldsymbol{U})$ comprendida por múltiples $p\left(\boldsymbol{U}-\boldsymbol{u}^{(i)}\right)$ obtenidas a partir de un muestreo realizado con las funciones de muestreo auxiliar $q(\cdot)$ y ponderadas por medio de $w\left(\boldsymbol{u}^{(i)}\right)$. Sin embargo, se pueden destacar importantes discrepancias entre ellos:

1. Tipo de $p(\boldsymbol{U})$: para WR está compuesta por varias funciones de densidad puntual, mientras que en el resto es una densidad genérica.

2. Función de muestreo auxiliar $q(\cdot)$:

   a) RS genera las muestras $\boldsymbol{u}^{(i)}$ según una función genérica $q(\boldsymbol{U})$ y decide su aceptación o rechazo en función de la discrepancia entre $p\left(\boldsymbol{u}^{(i)}\right)$ y $q\left(\boldsymbol{u}^{(i)}\right)$, por lo que su eficiencia la marca la cantidad de valores rechazados.

   b) IS muestrea las $\boldsymbol{u}^{(i)}$ por medio de $q(\boldsymbol{U})$, pero asigna un peso $w\left(\boldsymbol{u}^{(i)}\right)$ proporcional a la discrepancia anterior.

   c) WR se asemeja a IS, sólo que utiliza una función muestreo proporcional a una redistribución de las muestras según $r\left(\boldsymbol{u}^{(i)}\right)$.

   d) MC genera las muestras siguiendo la cadena $K\left(\boldsymbol{U}|\boldsymbol{u}^{(i-1)}\right)$, con ciertas propiedades, rechazando las muestras anteriores a la convergencia.

3. Los pesos $w\left(\boldsymbol{u}^{(i)}\right)$ asociados a las funciones de densidad puntual $p\left(\boldsymbol{U}-\boldsymbol{u}^{(i)}\right)$: en RS y MC todas las muestras tienen la misma importancia $\left(w\left(\boldsymbol{u}^{(i)}\right)=N^{-1}\right)$, mientras que en IS y WR, los pesos dependen de la función de la función de distribución elegida. Por eso en unos se ignora y en otros se trabaja con el par muestra-peso

4. Si el método funciona cuando desconocemos la expresión de $p(\boldsymbol{U})$ pero se dispone de $Cte \cdot p(\boldsymbol{U})$: aplica a RS, IS, WR, pero no a MC, sus cadenas no necesariamente tienen por qué.

Adicionalmente, también se debe tener en cuenta la paralización de cada uno de los cuatro métodos, pues no todos generan las muestras a la misma velocidad. En IS y WR, el valor de las muestras y de los pesos sin normalizar se realiza de forma independiente, de modo que puede hacerse de manera simultánea. RS funciona de manera similar, pues el valor de cada muestra y su aceptación o rechazo se realiza de forma independiente, salvo que discrepa en la cantidad de muestras generadas ya que depende de cuántas haya rechazado, hasta llegar a N. Finalmente, se recuerda que en MC las cadenas generan valores para la muestra de manera condicionada al anterior valor, $K\left(\boldsymbol{U}|\boldsymbol{u}^{(i-1)}\right)$, de modo que es necesario hacerlo de manera secuencial. Por otro lado, también es posible generar de forma simultánea varios conjuntos de muestras mediante varias cadenas de Markov ejecutadas en paralelo, de forma independiente.

Se recuerda también que en IS la función óptima de muestreo es proporcional a $|f(\boldsymbol{U})|$, por lo que es vano generar puntos en regiones donde dicha función sea nula. Esto se puede extender al resto de métodos, pues tales puntos no aportan valor a la aproximación integral.

Para finalizar, se contrastan los métodos RS frente a una combinación de una etapa de IS con una de WR, donde $r\left(\boldsymbol{u}^{(i)}\right)$ sean idénticas a los de la función original $\sum \alpha^{(i)}\delta\left(\boldsymbol{U}-\boldsymbol{u}^{(i)}\right)$. En RS, la generación de las posibles muestras $\boldsymbol{u}^{(i)}$ generadas con $q\left(\boldsymbol{U}\right)$ se da en la etapa de muestreo de IS con $q\left(\boldsymbol{U}\right)$. Los pesos $w\left(\boldsymbol{u}^{(i)}\right)=\dfrac{p\left(\boldsymbol{u}^{(i)}\right)}{q\left(\boldsymbol{u}^{(i)}\right)}$ en IS para las muestras son proporcionales a $\Pr\left(A|\boldsymbol{u}^{(i)}\right)=\dfrac{p\left(\boldsymbol{u}^{(i)}\right)}{Mq\left(\boldsymbol{u}^{(i)}\right)}$ en el proceso de aceptación/rechazo para cada una de las muestras de RS. En la última etapa, mientras que RS hace el proceso de aceptación/rechazo de la muestra de forma independiente a las demás, el uso de una etapa de WR tras una de IS con $r\left(\boldsymbol{u}^{(i)}\right)=\dfrac{p\left(\boldsymbol{u}^{(i)}\right)}{q\left(\boldsymbol{u}^{(i)}\right)}$ hace que las muestras admitidas (y posiblemente repetidas) sean aquellas con valores $\dfrac{r(\boldsymbol{u}^{(i)})}{\sum r(\boldsymbol{u}^{(i)})}$ más significativos. Tras la etapa de WR, los pesos de las N muestras son equiponderantes, y el mismo valor se les asigna tras la etapa de RS.

Ambos métodos se comportan de forma parecida para $N\to\infty$ debido a que, en RS, sólo se usa un valor M a lo largo de todo el soporte de $p(\boldsymbol{U})$, al igual que su análogo $\sum_{i=1}^{N} r(\boldsymbol{u}^{(i)})$ en WR.

Capítulo 4

Técnicas de simulación estocástica en algoritmos de optimización heurística

Este capítulo se enfoca en

El código completo está disponible en el repositorio de Github: `https://github.com/mavice07/TFG-Mates.git`.

4.1. Métodos de optimización de Monte-Carlo

4.2. Problema de aprendizaje de la distribución de probabilidades

4.3. Aplicaciones prácticas

Capítulo 5

Conclusiones y Trabajo Futuro

El presente capítulo pretende dar una visión global de las conclusiones a las que se ha llegado durante el desarrollo de esta memoria. Esto nos servirá para poder evaluar las contribuciones personales y el impacto de los resultados obtenidos, y, al mismo tiempo, descubrir las líneas de investigación y trabajo que quedan abiertas, y que podrían ser la base de nuevos proyectos en el futuro.

5.1. Contribuciones

5.2. Conclusiones

5.3. Trabajos relacionados

A continuación, presentaremos brevemente algunos trabajos relacionados con el nuestro. Estos trabajos han servido, desde el inicio, para obtener inspiración en el punto de partida, para usarlos como comparación, o incluso como propuestas de lecturas complementarias a lo largo de todo este trabajo.

- **Trabajo1**
- **Trabajo2**
- **Trabajo3**

5.4. Trabajo futuro

En cuanto a las líneas de investigación que quedan abiertas, y sobre las que se podrían desarrollar nuevos trabajos en el futuro, destacan las siguientes:

- **Trabajo1**:
- **Trabajo2**:

Bibliografía

[1] E. Sánchez-Jiménez, Y. Hernandez y J. Ortiz. "Técnicas de Optimización de Hiperparámetros en Modelos de Aprendizaje Automático para Predicción de Enfermedades Cardiovasculares". En: *Proceedings of the Conference*. Nov. de 2022.

Anexos

Info Anexo 1

Info anexo 2

Info anexo 3