

Cadenas de Markov restringidas: una aplicación a la generación de melodías

SANTIAGO LOZANO SANDINO

Proyecto de grado presentado como parte de los requisitos para optar al grado de
Magíster en Estadística

Profesor supervisor:
MANUEL GALEA ROJAS

Santiago de Chile, noviembre de 2017

© MMXVII, Santiago Lozano Sandino

A mis viejitos amados.

RESUMEN

Este trabajo presenta una manera de generar secuencias melódicas basadas en cadenas de Markov restringidas. Con tal objetivo este documento entrega la teoría musical necesaria, así como la teoría básica de cadenas de Markov relevante para este trabajo. Además, se explica la propuesta de Pachet, Roy & Barberi (2011b) de cadenas de Markov restringidas, de la publicación *Finite-Length Markov Processes with Constraints*.

El producto final del trabajo es la programación del algoritmo que permite generar secuencias melódicas, contenida en el paquete `MarkovMusic`. Este documento incluye como ejemplo una de las melodías generadas con una cadena de Markov restringida de orden 2, y en el anexo digital se incluyen veintidós melodías más generadas con el algoritmo.

Contenidos

Introducción y motivación	i
1. Un poco de teoría musical	1
1.1. Tonalidad	1
1.2. Armonía y melodía	2
1.3. Blues de doce compases	3
2. Cadenas de Markov	5
2.1. Propiedades de las cadenas de Markov	7
2.2. Cadenas de Markov de orden d	12
2.3. Ejemplos de cadenas de Markov de orden 1, 2 y 3	14
2.4. Cadenas de Markov como modelos en grafo	17
3. Cadenas de Markov restringidas	19
3.1. Cadena de Markov restringida de orden 1	19
3.2. Cadena de Markov restringida de orden d	24
3.3. Arco-consistencia	27
3.4. Condiciones para la satisfacción de las restricciones	29
4. Implementación	31
4.1. Descripción del paquete <code>MarkovMusic</code>	31
4.2. La generación de un blues de doce compases	33
5. Conclusiones	38
Bibliografía	39

INTRODUCCIÓN Y MOTIVACIÓN

Existen distintas maneras de generar música mediante algoritmos, desde un enfoque tanto armónico (i.e., acordes) como melódico (secuencia de notas). Frank Brinkkemper (octubre 5, 2016), en un artículo de *The Asimov Institute*, enuncia seis herramientas distintas que existen en la actualidad para generar música, y que utilizan métodos como *long short-term memory*, redes neuronales y cadenas de Markov. Entre las herramientas mencionadas por el artículo de Brinkkemper está una desarrollada por los Laboratorios de Ciencia de la Computación de Sony (CSL Sony), con sede en París, que utiliza cadenas de Markov restringidas.

La página web de CSL Sony contiene un repositorio de las publicaciones producidas por el centro de investigación. En dichas publicaciones se explica la teoría e implementación de cadenas de Markov en la generación de melodías, progresiones armónicas, así como aplicaciones en otros campos. En particular, el documento *Finite-Length Markov Processes with Constraints* (Pachet *et al.*, 2011b), explica de manera sencilla el uso de cadenas de Markov restringidas para la generación de melodías, y es la base del presente trabajo.

Las cadenas de Markov restringidas se refieren a cadenas de Markov que cumplen cierta restricción en el espacio estado en un momento determinado. Es decir, si se impone una restricción en un momento L , en ese instante ocurrirá alguno de los estados pertenecientes al conjunto de estados impuestos arbitrariamente por el usuario, pero se permite que los estados anteriores o posteriores de la secuencia se generen aleatoriamente. Esta aproximación permite generar una gran variedad de secuencias, pues aparte de las restricciones que impone el usuario, los demás estados se producen aleatoriamente.

Un primer objetivo del presente documento es estudiar la propuesta de Pachet *et al.* (2011b), es decir, el uso de cadenas de Markov restringidas

aplicadas al caso de la generación de secuencias melódicas. Otro propósito de este trabajo es aplicar la metodología para generar secuencias de un largo finito, enunciando las condiciones para el cumplimiento de las restricciones impuestas. La metodología es implementada en la generación de una melodía que ajusta a la estructura armónica de un blues de doce compases, utilizando las reglas aprendidas por los datos de entrenamiento. Estas reglas consisten en melodías basadas en escalas de blues, y la imposición de las restricciones necesarias para ajustarse a la estructura armónica de un blues.

Este documento está dividido en cinco secciones. La primera sección hace una introducción sobre varios aspectos de teoría musical, necesarios para comprender la implementación de las cadenas de Markov para la generación de secuencias melódicas. La segunda sección hace un breve resumen de los aspectos básicos de las cadenas de Markov, enunciando las propiedades más relevantes para el desarrollo de este trabajo. La tercera sección aborda el tema de las cadenas de Markov restringidas, para cadenas de orden 1 y de orden d , explica el concepto de arco-consistencia, y enuncia las condiciones para la satisfacción de las restricciones. La cuarta sección presenta la descripción del paquete `MarkovMusic` programado en R especialmente la aplicación del algoritmo, y expone algunos los resultados de la generación de secuencias melódicas. Finalmente, la última sección está dedicada a las conclusiones, observaciones finales y discusión del trabajo.

1. UN POCO DE TEORÍA MUSICAL

En esta sección se presentan los conceptos básicos de teoría musical, necesarios para comprender el contexto de este trabajo. A pesar que se habla en términos absolutos durante gran parte de esta sección, es importante notar que en algunas ocasiones las reglas aquí descritas no aplican en el mundo real, y su interpretación es relativa. En general este trabajo utiliza la notación anglosajona para notas y acordes, aunque también se mencionan otros tipos de notaciones.

1.1. Tonalidad

Tradicionalmente la música occidental consta de doce notas o tonos (i.e., $C, C\#/D\flat, D, D\#/E\flat, \dots, A\#/B\flat, B$), que conforman la escala cromática, donde la distancia entre notas contiguas de dicha escala corresponde a un semitono. Estas doce notas se repiten cíclicamente en octavas, en un fenómeno conocido como circularidad tonal. Por ejemplo, A en la cuarta octava, o A_4 como también se suele notar, tiene una frecuencia de 440 Hz, y A_5 tiene una frecuencia de 880 Hz. Sin embargo, estas dos notas son equivalentes en un sentido musical, pues pueden la misma función, aunque la segunda nota sea más aguda que la primera. De todas maneras debe haber coherencia en la secuencia de notas, en el sentido que puede resultar extraño al oído pasar de A_3 a B_7 . Es decir, no es común que una melodía salte entre notas que pertenezcan a octavas distantes entre sí.

Hay varias maneras de representar las notas. Una de estas nomenclaturas se presentó anteriormente, que incluye el nombre de la nota y su respectiva octava (e.g., A_5). El número de la octava incrementa al pasar de B a C , es decir, C_5 le sigue a B_4 . Otro tipo de nomenclatura es la notación MIDI, que asigna un número entero a cada una de las notas, partiendo desde C en la octava -1 , que equivale a 0, y está contenido dentro del rango de frecuencias auditivo perceptible para el ser humano. Por ejemplo A_4 en notación MIDI equivale a 69, mientras que A_5 es 81, que está a doce semitonos de distancia.

Existen otras aproximaciones a la música que van más allá de lo tonal, como las variaciones microtonales, pero por simplicidad el trabajo actual se limita a las doce notas, ya que dicha aproximación permite usar espacio estado discreto. Además, este trabajo se enfoca únicamente en la información de la tonalidad. Es decir, las doce notas de la escala cromática (y el silencio) serán consideradas como la base para el espacio estado de las cadenas de Markov. En este sentido, las cadenas de Markov estudiados en este documento corresponden a procesos estocásticos de tiempo discreto y espacio estado discreto y finito.

1.2. Armonía y melodía

La armonía está compuesta por varias notas que suenan simultáneamente. Por ejemplo un acorde de mi menor o *Em* está compuesto de las notas *E*, *B* y *G* (aunque no necesariamente en ese orden, y cuyas octavas no suelen ser relevantes). Una secuencia de acordes se conoce como progresión o estructura armónica. Por otra parte, la melodía puede comprenderse como una secuencia de notas individuales. Las melodías están basadas en escalas o modos (e.g., escala cromática, diatónica mayor, modo dórico, etc.). Por ejemplo, la escala diatónica mayor es la más conocida de las escalas musicales, y consiste en la siguiente secuencia de notas: $C - D - E - F - G - A - B - C$. La última nota de la secuencia anterior *C*, es la misma *C* que la nota inicial, pero una octava más arriba.

A pesar que se pueden considerar como dos aspectos musicales distintos, tanto la melodía como la armonía sirven a un propósito común dentro de una pieza musical. La armonía funciona como la columna vertebral de la pieza. Por ejemplo, en la sección de improvisación de una canción de jazz, la progresión armónica brinda las bases al improvisador para construir la melodía: sobre un acorde de *Em* se puede improvisar o componer una melodía basada en alguna escala que contenga las notas *E*, *B* y *G*, e.g., la pentatónica mayor de *G* o la menor de *E*. De forma inversa, se puede construir una progresión armónica que

se ajuste a una melodía, y a este proceso se le conoce como armonización de la melodía.

La tonalidad de las melodías o armonías puede ser modificada, al trasponerlas un número determinado de semitonos. Por ejemplo, a partir del acorde de *Em* se puede encontrar el acorde de *Fm*, si se traspone cada una de las notas que lo componen un semitono hacia arriba. Es decir, *E* se reemplaza por *F*, *B* por *C*, y *G* por *Ab*.

Como se advirtió anteriormente, la explicación presentada en esta subsección es una simplificación, pues por ejemplo la armonía puede conformarse por varias melodías que suenan a la vez. Sin embargo, esta explicación permite comprender lo que se plantea en este trabajo: se utiliza la armonía para determinar el conjunto de notas o la escala que se puede utilizar para generar la melodía.

1.3. Blues de doce compases

Un compás es una medida de tiempo, que contiene un número específico de notas, dependiendo de su valor. El valor de una nota es su duración. Por ejemplo, la negra dura $1/4$, mientras que la corchea dura la mitad de una negra, y la blanca el doble de una negra. Si la métrica de una canción es $4/4$ quiere decir que en cada compás habrá el equivalente a la duración cuatro notas negras. Con el propósito de delimitar el trabajo, las secuencias generadas mediante las cadenas de Markov serán de notas de igual duración.

El blues de doce compases es un conjunto de progresiones armónicas utilizadas frecuentemente en la música popular, en géneros como blues, jazz, pop, rock, entre otros. Existen muchas variaciones del blues de doce compases, y una de ellas es la que aparece en la Figura 1.

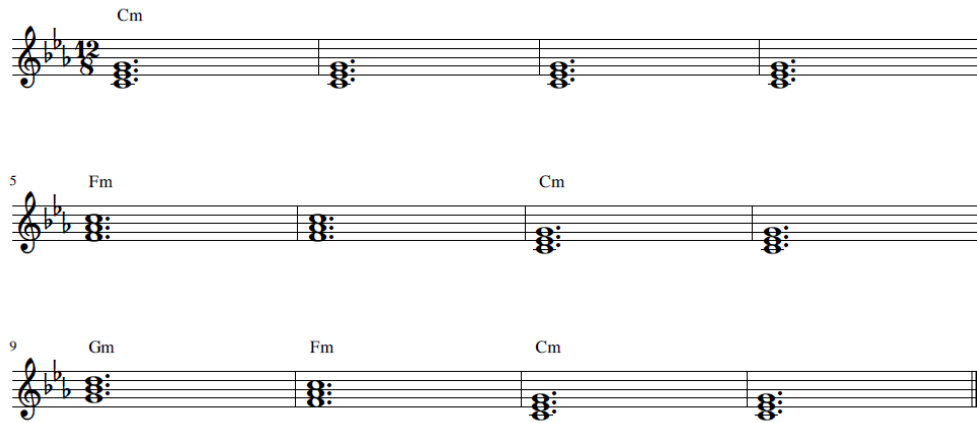


Figura 1. Blues de doce compases en *C* menor.

La Figura 1 describe un blues de doce compases en *Cm*, con una métrica de 12/8. Por ejemplo, sobre este tipo de estructura se puede improvisar usando una melodía basada en la escala menor de blues en *C* (i.e., melodías que incluyan *C*, *Eb*, *F*, *Gb*, *G*, *Bb*). Otra manera de abordar dicha progresión armónica es la siguiente: durante los primeros cuatro compases se puede improvisar una melodía basada en la escala menor de blues en *C*, en el quinto y sexto compás improvisar la melodía usando la escala menor de blues en *F* (i.e., la escala con notas *F*, *Ab*, *Bb*, *B*, *C*, *Eb*), y luego durante el resto de compases improvisar sobre la escala menor de blues en *C*. Sin embargo, las melodías deben tener coherencia entre sí, de tal manera que la transición entre una escala y la otra no sea repentina.

Uno de los propósitos de este trabajo es generar secuencias melódicas utilizando cadenas de Markov restringidas, de tal manera que se ajuste a la estructura de un blues de doce compases, como el descrito en la Figura 1.

2. CADENAS DE MARKOV

En Rincón (2012) se define un proceso estocástico como una colección de variables aleatorias $\{X_t: t \in T\}$, donde X_t es el estado en el tiempo t , y T es el conjunto de parámetros. Cuando $t \in T = \{0, 1, 2, \dots\}$, se trata de un proceso en tiempo discreto, mientras que si $t \in T = [0, \infty)$ es un proceso de tiempo continuo. Además $X_t \in A$, donde A se conoce como el espacio estado, y denota los posibles valores que puede tomar la variable aleatoria. Un proceso estocástico es un modelo que permite estudiar la manera en que un fenómeno se comporta a lo largo del tiempo.

El espacio estado A puede ser discreto o continuo. Un ejemplo de un espacio estado continuo puede ser una frecuencia auditiva medida en Hz, el cual está definido sobre el conjunto de números reales positivos. En cambio, un ejemplo de espacio estado discreto puede ser el tono de una melodía: los tonos o notas de una melodía están contenidas dentro del conjunto de posibles valores de la escala cromática.

De manera general, los procesos estocásticos pueden ser clasificados en cuatro tipos, dependiendo de las características del espacio estado y del conjunto de parámetros: (i) proceso estocástico de tiempo discreto y espacio estado discreto, (ii) proceso estocástico de tiempo continuo y espacio estado discreto, (iii) proceso estocástico de tiempo discreto y espacio estado continuo, y (iv) proceso estocástico de tiempo continuo y espacio estado continuo. El espacio estado discreto también puede ser finito o infinito.

Las cadenas de Markov consideran una relación particular entre las variables aleatorias, reflejada en el supuesto de Markov, que afirma que la probabilidad de transición al estado X_t depende únicamente del estado inmediatamente anterior X_{t-1} , también llamado prefijo. Es decir:

$$p(X_t = x_t | X_{t-1} = x_{t-1}, \dots, X_1 = x_1) = p(X_t = x_t | X_{t-1} = x_{t-1})$$

Donde t es el tiempo, que para este caso se considera discreto, y X_t es una variable aleatoria en el tiempo t , y $x_t \in A$ es una realización, elemento del espacio estado finito $A = \{a_1, \dots, a_k\}$. Este documento se enfoca en un tipo de proceso en particular: de tiempo discreto y espacio estado discreto y finito, que asume el supuesto de Markov como relación entre las notas.

En general se suele usar el término *cadena* o *proceso* de Markov de manera intercambiable. Sin embargo, con el fin de evitar confusiones, a lo largo de este documento se utiliza el término *cadena* para hacer referencia al modelo utilizado para la estimación de las probabilidades de transición. En cambio, el término *proceso* se utiliza para hacer referencia a la escala empleada en la generación de la secuencia. En este sentido, una cadena puede involucrar más de un proceso.

Las cadenas de Markov tradicionales consideran que las probabilidades de transición, $p(X_t = x_t | X_{t-1} = x_{t-1}) = p(x_t | x_{t-1})$, son invariantes en el tiempo. Es decir que si $p(x_i | x_{i-1}) = p(x_j | x_{j-1})$, para cualquier i y j , la cadena de Markov en cuestión es homogénea.

Si se considera una distribución de probabilidad inicial, $p(X_1 = x_1)$, la probabilidad de tener una secuencia en particular $\mathbf{s} = (x_1, \dots, x_L)$ con una cadena de Markov puede expresarse como:

$$p(\mathbf{s}) = p(x_1, \dots, x_L) = p(x_1) \prod_{t=2}^L p(x_t | x_{t-1})$$

Las probabilidades de transición de las cadenas de Markov de orden 1 pueden expresarse de forma matricial, de la siguiente manera:

$$M = \begin{pmatrix} p(X_t = a_1 | X_{t-1} = a_1) & p(X_t = a_2 | X_{t-1} = a_1) & \cdots & p(X_t = a_k | X_{t-1} = a_1) \\ p(X_t = a_1 | X_{t-1} = a_2) & p(X_t = a_2 | X_{t-1} = a_2) & \cdots & p(X_t = a_k | X_{t-1} = a_2) \\ \vdots & \vdots & \ddots & \vdots \\ p(X_t = a_1 | X_{t-1} = a_k) & p(X_t = a_2 | X_{t-1} = a_k) & \cdots & p(X_t = a_k | X_{t-1} = a_k) \end{pmatrix}$$

M es una matriz cuadrada, donde el número de filas y columnas coincide con la $\text{card}(A) = k$, i.e., el número de elementos del conjunto A . De manera

reducida, la matriz de transición puede representarse mediante su elemento típico:

$$M = \left(\left(p(X_t = a_i | X_{t-1} = a_j) \right) \right), \quad i, j = 1, \dots, k$$

En general M es una matriz estocástica, pues sus filas suman 1, salvo el caso que en el corpus (o secuencias usadas para la estimación) no contenga información sobre algún elemento cualquiera $a_i \in A$. Esto aplica si la matriz de transición se estima de manera frecuentista, a partir de la frecuencia relativa de transición entre eventos, provisto el corpus. En caso que suceda, la fila correspondiente a las probabilidades de transición desde el estado a_i , suma 0. Típicamente las filas que suman 0 se eliminan de la matriz de transición. Sin embargo por conveniencia este documento conserva las filas que suman 0, pues facilitan algunos cálculos, a pesar de no ser estocástica.

2.1. Propiedades de las cadenas de Markov

Existen varias propiedades de las cadenas de Markov que se deben considerar en el contexto de este documento. A continuación se explican de forma resumida las propiedades relevantes para el trabajo. Para ver en detalle estas propiedades se sugiere consultar Blanco, Arunachalan & Dharmaraja (2012), Hoel, Port & Stone (1972), Kulkarni (2010), y Rincón (2012).

Considere un conjunto finito de estados $A \in \{a_1, \dots, a_k\}$, con $a_i, a_j \in A$, dos estados cualquiera. Un estado a_i es accesible desde a_j , si la probabilidad la probabilidad de alcanzar el estado a_i desde el estado a_j en $n \geq 0$ pasos, es estrictamente positiva. Es decir, $p(X_n = a_i | X_0 = a_j) > 0$ (Blanco *et al.*, 2012, p. 353). La accesibilidad de a_i desde a_j puede expresarse como $a_j \rightarrow a_i$. Si además $a_j \rightarrow a_i$ y $a_i \rightarrow a_j$, entonces se puede decir que a_i y a_j comunican: $a_i \leftrightarrow a_j$. Note que según la definición, en número de pasos puede ser $n = 0$, en este caso se cumple que $p(X_0 = a_i | X_0 = a_i) = 1$, por lo que $a_i \rightarrow a_i$ (Rincón, p. 43). Un conjunto de estados que comunican entre sí se conoce como estados pertenecientes a una misma clase de comunicación (Rincón, p. 43). Una cadena

de Markov es irreducible si todos sus estados comunican entre sí (Blanco *et al.*, p. 354) o si todos pertenecen a una misma clase de comunicación (Rincón, p. 44 – 45).

Usando la definición empleada por Blanco *et al.* (2012, p. 357), sea $\varphi_{a_i a_j}^{(n)}$ la probabilidad que, partiendo de a_i , el proceso alcance el estado a_j por primera vez en el tiempo $n > 1$, es decir:

$$\varphi_{a_i a_j}^{(n)} = p(X_n = a_j, X_{n-1} \neq a_j, \dots, X_1 \neq a_j | X_0 = a_i)$$

Por otra parte se puede definir $\psi_{a_i a_j}^{(n)}$ como la probabilidad que el proceso alcance el estado a_j partiendo del estado a_i en el paso n :

$$\psi_{a_i a_j}^{(n)} = p(X_n = a_j | X_0 = a_i) = p(X_{t+n} = a_j | X_t = a_i)$$

La última parte de la igualdad se obtiene asumiendo que el proceso es homogéneo. Entonces $\varphi_{a_i a_j}^{(n)}$ se diferencia de $\psi_{a_i a_j}^{(n)}$ en el sentido que la primera expresión corresponde a la probabilidad de la primera visita al estado a_j partiendo de a_i luego de n pasos, mientras que el segundo es la visita al estado a_j partiendo de a_i luego de n pasos, sin importar si es o no la primera visita a ese estado. Sin embargo, existe la siguiente relación entre ambos:

$$\psi_{a_i a_j}^{(n)} = \sum_{r=0}^n \varphi_{a_i a_j}^{(r)} \psi_{a_j a_j}^{(n-r)}$$

Donde $\varphi_{a_i a_j}^{(0)} = 0$, $\psi_{a_j a_j}^{(0)} = 1$, y $\varphi_{a_i a_j}^{(1)} = \psi_{a_i a_j}$. Los estados de una cadena de Markov se pueden clasificar como recurrentes, transitorios y absorbentes. Partiendo de $\varphi_{a_j a_j}^{(n)}$ se puede definir la probabilidad que el proceso regrese al estado a_j en algún momento (habiendo partido de a_j) de la siguiente manera: $\varphi_{a_j a_j} = \sum_{n=1}^{\infty} \varphi_{a_j a_j}^{(n)}$. Si a_j es un estado recurrente con probabilidad 1 volverá a ocurrir en algún momento, i.e., $\varphi_{a_j a_j} = 1$. En cambio, si a_j es un estado

transitorio, entonces $\varphi_{a_j a_j} < 1$. Visto de otra manera, la probabilidad que el estado a_j no se vuelva a producir en la cadena es $1 - \varphi_{a_j a_j}$.

Si a_j es un estado absorbente, entonces se cumple que $\varphi_{a_j a_j} = 1$ y $\varphi_{a_j a_i} = 0, \forall a_j \neq a_i$. Es decir, una vez el proceso alcanza el estado a_j es incapaz de salir de él. Por definición entonces, si a_j es un estado absorbente entonces es recurrente, mientras que los demás estados $a_i \neq a_j$ son estados transitorios. Si una cadena de Markov no tiene estados absorbentes, necesariamente todos sus estados son recurrentes. Sin embargo, puede darse el caso que existan estados transitorios sin existir un estado absorbente.

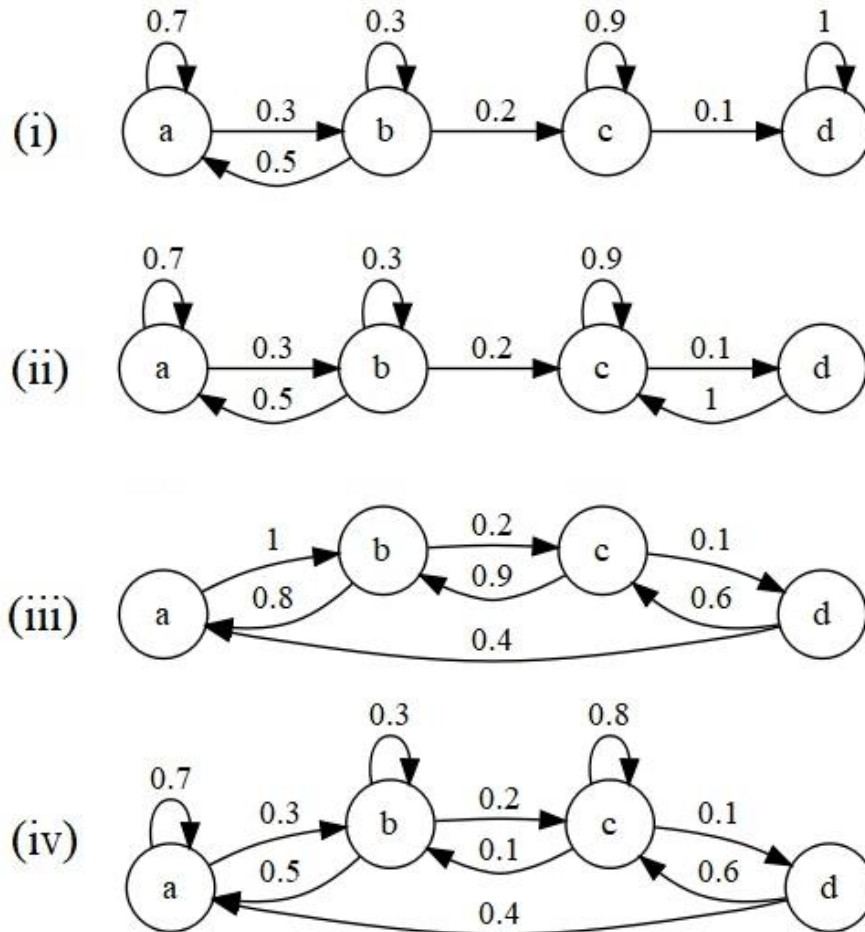


Figura 2. Ejemplos de cadenas de Markov representadas como diagramas de transición.

Para entender más fácilmente los tipos de estados se presenta la Figura 2, que ilustra diferentes cadenas de Markov, con un mismo espacio estado $A = \{a, b, c, d\}$. Más adelante se profundiza sobre este tipo de representaciones, pero de forma resumida cada círculo o nodo representa un estado, cada flecha o arco representa la relación de dependencia entre estados, y su etiqueta la probabilidad de transición entre ellos.

En el caso de (i), el estado d es absorbente, pues una vez la cadena alcanza ese estado seguirá estando en él; en cambio los estados a , b y c son transitorios. Por otra parte, en (ii) los estados a y b son transitorios, pues a pesar que la cadena no tiene estados absorbentes, una vez alcanza el estado c queda la cadena queda atrapada entre c y d , lo que implica que son un conjunto de estados absorbentes. Por último, en (iii) y (iv) todos los estados son recurrentes, pues es posible llegar a cualquiera de ellos desde algún otro. Las cadenas (i) y (ii) son absorbentes, porque tienen un estado o conjunto de estados que son absorbentes, mientras que las cadenas (iii) y (iv) son recurrentes.

El tiempo medio de recurrencia se puede entender de la siguiente manera: considere $n = \{1, 2, \dots\}$ como una variable aleatoria que representa el tiempo que demora el proceso en regresar al estado a_i y cuya distribución es $\varphi_{a_i a_i}^{(n)}$. Entonces el valor promedio de n se conoce como el tiempo de recurrencia, es decir, $\mu_{a_i} = \sum_{n=1}^{\infty} n \varphi_{a_i a_i}^{(n)}$. El tiempo medio de recurrencia se puede interpretar como el número promedio de pasos que demora la cadena en regresar al estado a_i . De forma general, se puede definir $\mu_{a_i a_j} = \sum_{n=1}^{\infty} n \varphi_{a_i a_j}^{(n)}$ como el tiempo de recurrencia del estado a_j , partiendo del estado inicial a_i , y se interpreta como el número promedio de pasos que toma a la cadena para llegar al estado a_j desde a_i .

Los estados recurrentes pueden ser clasificados en recurrentes positivos y recurrentes nulos, dependiendo del tiempo medio de recurrencia. Considere un

estado recurrente a_i ; si $\mu_{a_i} = \infty$, entonces a_i es recurrente nulo. Por otra parte, si $\mu_{a_i} < \infty$ entonces el estado a_i es recurrente positivo.

A partir del tiempo medio de recurrencia μ_{a_i} se puede definir la frecuencia relativa del número de visitas al estado a_j en el largo plazo como $\pi_{a_i} = 1/\mu_{a_i}$ (Rincón, p. 65), que es un resultado del teorema ergódico. En este sentido, si $\mu_{a_i} \rightarrow \infty$, entonces la frecuencia relativa del número de visitas al estado a_i es 0; a pesar que a_i es un estado recurrente la probabilidad que la cadena visite nuevamente ese estado es 0. Sin embargo, puede demostrarse que no existen estados recurrentes nulos en cadenas de Markov con espacio estado finito (Rincón, p. 65).

La ecuación de Chapman-Kolmogorov se define como:

$$\psi_{a_i a_j}^{(n)} = \sum_{a_k} \psi_{a_i a_k}^{(r)} \psi_{a_k a_j}^{(n-r)}$$

A partir de la ecuación de Chapman-Kolmogorov se puede demostrar que $\psi_{a_i a_j}^{(n)} = [p(X_t = a_j | X_{t-1} = a_i)]^n$, para cualquier $t \in T$. Este resultado implica que se puede estimar la probabilidad que la cadena alcance el estado a_j partiendo del estado a_i en el paso n , usando los elementos de la matriz de transición. Considere la matriz de transición M asociada a la cadena de Markov (iv) de la Figura 2:

$$M = \begin{pmatrix} 0.7 & 0.3 & 0 & 0 \\ 0.5 & 0.3 & 0.2 & 0 \\ 0 & 0.1 & 0.8 & 0.1 \\ 0.4 & 0 & 0.6 & 0 \end{pmatrix}$$

Note que la probabilidad de pasar del estado a al c en un paso es 0. Pero la probabilidad de pasar de a a c en dos pasos es 0.06. Esto puede obtenerse al multiplicar la matriz M por sí misma:

$$M^2 = \begin{pmatrix} 0.64 & 0.3 & 0.06 & 0 \\ 0.5 & 0.26 & 0.22 & 0.02 \\ 0.09 & 0.11 & 0.72 & 0.08 \\ 0.28 & 0.18 & 0.48 & 0.06 \end{pmatrix}$$

Entonces para estimar la probabilidad que la cadena alcance el estado a_j partiendo del estado a_i en el paso n , basta con estimar M^n y analizar el elemento de la matriz correspondiente a los estados deseados.

La periodicidad de un estado a_i se puede definir como:

$$\lambda(a_i) = \text{GCD}\{n \geq 1: p(X_n = a_i | X_0 = a_i) > 0\}$$

Donde *GCD* es el máximo común divisor (*greatest common divisor*) (Blanco *et al.*, p. 355). Si $\lambda(a_i) = 1$, entonces a_i es un estado aperiódico. Si además $\lambda(a_i) = 1, \forall a_i \in A$, entonces el proceso es aperiódico. En cambio, suponga que $\lambda(a_i) = k$; esto no implica que cada k pasos la cadena va a visitar el estado a_i , sino que el número de pasos entre cada visita a a_i es múltiplo de k .

Usando como referencia la Figura 2, el proceso descrito en (iii) no es aperiódico, pues todos sus estados tienen periodicidad 2: por ejemplo, si solo se puede regresar al estado d después de un número par de pasos. En cambio, el proceso (iv) es aperiódico: los estados a , b y c comunican consigo mismo, por lo que su periodicidad es 1. Por otra parte, al estado d del proceso (iv) se puede llegar en dos pasos mediante la trayectoria $d \rightarrow c \rightarrow d$, pero también mediante la trayectoria $d \rightarrow c \rightarrow c \rightarrow d$, es decir en tres pasos; su máximo común divisor es 1, lo que implica que es aperiódico.

2.2. Cadenas de Markov de orden d

El supuesto de Markov puede relajarse, de tal manera que la probabilidad del estado actual dependa de un número finito d de estados anteriores. Es decir:

$$p(x_t | x_{t-1}, \dots, x_1) = p(x_t | x_{t-1}, \dots, x_{t-d})$$

Bajo este supuesto se puede definir una cadena de Markov de orden d . Esto permite tener más memoria, en el sentido que considera estados anteriores.

Esto permite generar secuencias más parecidas a aquellas con las que se estimaron las probabilidades de transición. Sin embargo, existe una disyuntiva al utilizar más memoria en las cadenas de Markov. Si bien la generación de las secuencias pueden tener más sentido musical, la cadena de orden d va a tender a replicar el corpus durante secciones largas de secuencias, (Papadopoulos *et al.*, p. 2731, 2014), generando un sobreajuste.

Lo anterior resulta ser problemático si la longitud del corpus o secuencia de entrenamiento es limitada. De hecho, si no se tiene una secuencia de entrenamiento bastante larga puede que no exista información suficiente para estimar de manera adecuada las probabilidades de transición.

Las probabilidades de transición de las cadenas de Markov de orden d también se pueden representar de manera matricial, de la siguiente manera:

$$G = \left(\left(p(X_t = a_i | X_{t-1} = a_j, \dots, X_{t-d} = a_h) \right) \right), \quad i, j, \dots, h = 1, \dots, k$$

Lo anterior implica que la matriz de transición no es cuadrada. Si bien el número de columnas coincide con $\text{card}(A) = k$, el número de filas es $\text{card}(A)^d = k^d$, que es el número de d combinaciones posibles de elementos de A . Nuevamente es importante mencionar que si el corpus no incluye todas las combinaciones de secuencias de largo d , entonces las probabilidades de la fila correspondiente a esas secuencias ausentes en el corpus tendrán probabilidad 0 (o se pueden eliminar de la matriz, según convenga).

Considere la secuencia o vector de realizaciones $\mathbf{y}_t^d = (x_t, \dots, x_{t-d+1})$, conocido como un d -grama. Estas secuencias de largo d , pertenecen al conjunto que puede denotarse $A^{(d)}$, y corresponde a todas las posibles secuencias de largo d como combinaciones posibles de los elementos de A . En otras palabras, $\mathbf{y}_t^d \in A^{(d)}, \forall t$, donde $A^{(d)} = \{\mathbf{b}_1, \dots, \mathbf{b}_{k^d}\}$, y $\text{card}(A^{(d)}) = \text{card}(A)^d$. Note que si $d = 1$, entonces $\mathbf{y}_t^1 = (x_t)$, y $A^{(1)} = A$. Los d -gramas permiten denotar los elementos de la matriz de transición como:

$$G = \left(\left(p(X_t = a_i | Y_{t-1}^{(d)} = \mathbf{b}_j) \right) \right), \quad i = 1, \dots, k, \quad j = 1, \dots, k^d$$

Usando los d -gramas, también se puede construir una matriz cuadrada, tal que su elemento típico sea $G^* = \left(\left(p(Y_t^d = \mathbf{b}_i | Y_{t-1}^d = \mathbf{b}_j) \right) \right), i, j = 1, \dots, k^d$. A través de esta matriz cuadrada se pueden verificar varias propiedades de las matrices de transición, y en ocasiones la notación extensa correspondiente a la matriz de transición G^* puede resultar más conveniente que la notación abreviada G , a pesar de ser equivalentes.

Usando la notación extensa de la matriz de transición de las cadenas de orden d , este tipo de cadenas se asemejan a las de orden 1. En la subsección anterior se enunciaron algunas propiedades de las cadenas de Markov de orden 1, pero al considerar la notación extensa de las probabilidades de transición es evidente que estas propiedades también aplican para los procesos de orden d , pero en este caso deben estudiarse los d -gramas en lugar de los estados.

2.3. Ejemplos de cadenas de Markov de orden 1, 2 y 3

Considere el espacio estado $A = \{a, b\}$. Para iniciar una secuencia de una cadena de Markov se utiliza la distribución de probabilidad inicial:

$$\mathbf{v} = (p(a) \quad p(b))$$

Dados los estados en A , la matriz de transición de la cadena de orden 1 puede estimarse como frecuencia relativa condicionada al estado inmediatamente anterior:

$$M = \begin{pmatrix} p(X_t = a | X_{t-1} = a) & p(X_t = a | X_{t-1} = b) \\ p(X_t = b | X_{t-1} = a) & p(X_t = b | X_{t-1} = b) \end{pmatrix}$$

Ahora considere $A^{(2)} = \{(a, a), (b, a), (a, b), (b, b)\}$, el conjunto de todas las secuencias posibles de largo 2, dado el conjunto A . Para la matriz de transición de la cadena de orden 2 se tiene:

$$G = \begin{pmatrix} p(X_t = a|X_{t-1} = a, X_{t-2} = a) & p(X_t = b|X_{t-1} = a, X_{t-2} = a) \\ p(X_t = a|X_{t-1} = b, X_{t-2} = a) & p(X_t = b|X_{t-1} = b, X_{t-2} = a) \\ p(X_t = a|X_{t-1} = a, X_{t-2} = b) & p(X_t = b|X_{t-1} = a, X_{t-2} = b) \\ p(X_t = a|X_{t-1} = b, X_{t-2} = b) & p(X_t = b|X_{t-1} = b, X_{t-2} = b) \end{pmatrix}$$

Ahora sea el conjunto de todas las secuencias posibles de largo 3, dado el conjunto A :

$$A^{(3)} = \{(a, a, a), (b, a, a), (a, b, a), (b, b, a), (a, a, b), (b, a, b), (a, b, b), (b, b, b)\}$$

La matriz de transición de orden 3 asociada es:

$$H = \begin{pmatrix} p(a|a, a, a) & p(b|a, a, a) \\ p(a|b, a, a) & p(b|b, a, a) \\ p(a|a, b, a) & p(b|a, b, a) \\ p(a|b, b, a) & p(b|b, b, a) \\ p(a|a, a, b) & p(b|a, a, b) \\ p(a|b, a, b) & p(b|b, a, b) \\ p(a|a, b, b) & p(b|a, b, b) \\ p(a|b, b, b) & p(b|b, b, b) \end{pmatrix}$$

Para estimar las matrices de transición, considere el corpus correspondiente a la secuencia $\mathbf{z} = (a, a, b, b, a, b, a, b, b, a, a, b, a, a)$, ordenada de la más reciente a la más antigua, donde \mathbf{z} es la realización de un proceso Z , i.e., $\mathbf{z} \in Z$. Las probabilidades iniciales y condicionales se pueden estimar de forma frecuentista. Entonces la distribución inicial estimada es:

$$\hat{v}|\mathbf{z} = (8/15 \quad 7/15)$$

Para las matrices de transición de las cadenas de orden 1, 2 y 3, dada la secuencia de entrenamiento \mathbf{z} , se tiene:

$$\hat{M}|\mathbf{z} = \begin{pmatrix} 3/7 & 4/7 \\ 4/7 & 3/7 \end{pmatrix}, \quad \hat{G}|\mathbf{z} = \begin{pmatrix} 0 & 1 \\ 1/2 & 1/2 \\ 1/2 & 1/2 \\ 2/3 & 1/3 \end{pmatrix}$$

$$\hat{H}|\mathbf{z} = \begin{pmatrix} 0 & 0 \\ 1/2 & 1/2 \\ 1/2 & 1/2 \\ 1/2 & 1/2 \\ 0 & 1 \\ 1/2 & 1/2 \\ 1/2 & 1/2 \\ 1 & 0 \end{pmatrix}$$

Note que para estimar la frecuencia relativa de una matriz de transición de orden d , se descartan las primeras d observaciones, pues no se tiene información sobre qué generó el estado que inició la secuencia de entrenamiento. Por esta razón es imperativo que el orden de la cadena de Markov no supere la longitud del corpus.

Por otra parte, puede observarse que $\hat{H}|\mathbf{z}$ no es una matriz estocástica, pues su primera fila no suma 1. Esto sucede porque en la secuencia del corpus no existe ninguna secuencia (a, a, a) . Dicha fila podría eliminarse de la matriz, sin embargo su presencia no es problema al momento de generar una secuencia cualquiera, pues mediante la cadena nunca se va a llegar a (a, a, a) . Además, como se verá más adelante, en el contexto de las cadenas de Markov restringidas, conservar las filas que suman 0 es conveniente al momento de propagar las restricciones sobre las matrices, pues va a tener un número conocido de filas (i.e., k^d).

Se pueden utilizar cadenas de Markov de distinto orden al generar una secuencia. Por ejemplo, si no se tiene información previa, se podría iniciar la secuencia usando \mathbf{v} , e ir incrementando el orden de las matrices de transición secuencialmente.

Finalmente, como ejemplo de la notación extensa de las matrices de transición, considere el caso de orden 2:

$$G^* = \begin{pmatrix} p(a, a|a, a) & 0 & p(b, a|a, a) & 0 \\ 0 & p(a, b|b, a) & 0 & p(b, b|b, a) \\ p(a, a|a, b) & 0 & p(b, a|a, b) & 0 \\ 0 & p(a, b|b, b) & 0 & p(b, b|b, b) \end{pmatrix}$$

Cada uno de los elementos de la matriz de transición corresponde a la probabilidad de pasar de $X_{t-1} = a_j \cap X_{t-2} = a_h$ a $X_t = a_i \cap X_{t-1} = a_j$, para $a_i, a_j, a_h \in A$. Los 0 de la matriz de transición se explican porque X_{t-1} solo puede tomar un valor a la vez, por lo que $P(X_t = a, X_{t-1} = b | X_{t-1} = a, X_{t-2} = a) = 0$, por ejemplo.

2.4. Cadenas de Markov como modelos en grafo

Las cadenas de Markov de tiempo discreto y espacio estado discreto pueden ser representadas como un modelo en grafos. Los grafos son un conjunto de elementos, que en el contexto de las cadenas de Markov corresponden a los estados o prefijos. La relación que existe entre estados y prefijos se conoce como arcos o aristas, y la dirección expresa la dependencia condicional que existe entre ellos, mientras que valor del arco corresponde a la probabilidad condicional.

Anteriormente se presentó un tipo de representación de modelo en grafo; la Figura 2 presenta los diagramas de transición de algunas cadenas de Markov, considerando que son homogéneas. Pero las cadenas de Markov también se pueden representar indexando los estados o prefijos, lo cual es de gran utilidad si se tiene una cadena no homogénea o de orden mayor a 1. Por ejemplo, la Figura 3 presenta la representación gráfica de la cadena de Markov de orden 3 presentado como ejemplo en la subsección anterior.

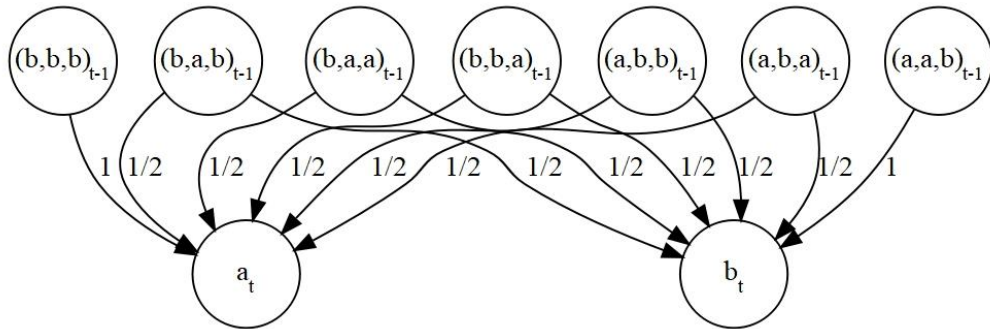


Figura 3. Representación del grafo asociado a la cadena de Markov $\hat{H}|z$.

En cambio, la Figura 4 presenta la representación del grafo asociado de la cadena de Markov de orden 2 de la subsección anterior, de manera extensa. Acá cada nodo representa un 2-grama.

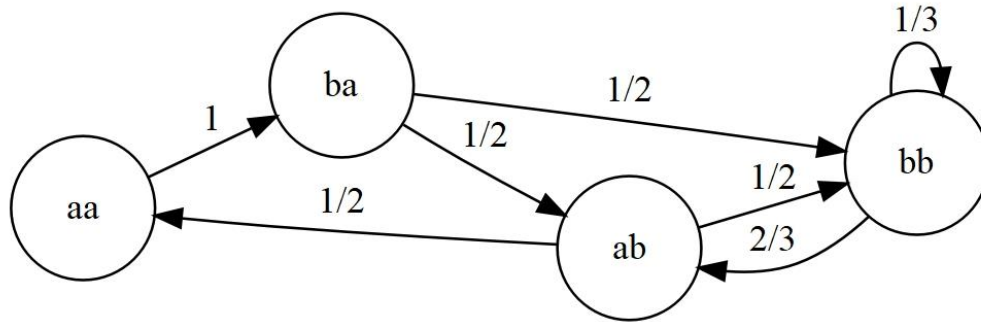


Figura 4. Representación del grafo asociado a la cadena de Markov $\hat{G}|z$, de forma extensa.

Entender las cadenas de Markov como modelos en grafo facilita el entendimiento sobre el concepto de arco-consistencia, que es esencial para la implementación de las cadenas de Markov restringidas propuestos por Pachet *et al.* (2011b).

3. CADENAS DE MARKOV RESTRINGIDAS

En esta sección se estudia la cadena de Markov restringida de orden 1, propuesta por Pachet *et al.* (2011b). Además se plantea la generalización para el caso de orden d . También se explica el concepto de arco-consistencia y las condiciones necesarias para el cumplimiento de las restricciones.

En resumen, las cadenas de Markov restringidas se construyen a partir de la modificación de las cadenas de Markov no restringidas. Además, varias características de las cadenas de Markov restringidas se pueden estudiar a partir de su contraparte no restringida, que permiten analizar si es posible satisfacer una restricción o no.

Una solución alternativa al problema de generación de secuencias que cumplan con ciertas condiciones consiste en generar muchas secuencias, y luego seleccionar aquellas que cumplan con los criterios deseados. Sin embargo, como argumentan Pachet *et al.* (2011a, p. 158), esto puede resultar costoso computacionalmente y existe la posibilidad que nunca se genere una secuencia que cumpla con los criterios deseados. Es por esta razón que los autores proponen usar cadenas de Markov restringidas.

3.1. Cadena de Markov restringida de orden 1

La propuesta de Pachet *et al.* (2011b) consiste en mantener el enfoque de caminata aleatoria que tienen las cadenas de Markov, pero restringiendo el espacio estado en un momento determinado de la secuencia, considerando únicamente un subconjunto del mismo. Es decir, $A_t \subset A$ para al menos un $1 < t < L$. Algunos espacios estado anteriores a L también deben modificarse para satisfacer la restricción. Al imponer restricciones sobre los elementos de la secuencia se está violando el supuesto de Markov (Pachet *et al.*, p. 152, 2011a), pues se está forzando la transición a un conjunto determinado de estados.

El resultado de imponer dichas restricciones sobre el espacio estado en un conjunto de matrices de transición convierte a la cadena en heterogénea. Esto se

debe a que existen varias matrices de transición diferentes, pues ahora las probabilidades de transición dependen del tiempo t .

Eliminar uno o más elementos del espacio estado en el instante L puede hacerse remplazando por 0 los elementos de la columna de la matriz de transición correspondiente a dichos estados, de tal manera que la probabilidad de producirlos en el contexto del proceso sea 0. Incluso algunos elementos de matrices de transición anteriores al instante L deben ser eliminados para satisfacer la restricción. Sin embargo, este procedimiento implica que las matrices dejan de ser estocásticas (pues los elementos de la fila no suman 1). Para solucionar esto se debe normalizar las matrices.

El procedimiento para normalizar las matrices de transición de la cadena de Markov de orden 1 puede entenderse como se plantea a continuación. Sea $A = \{a_1, \dots, a_k\}$ el espacio estado asociado a la cadena de Markov M , y x_t las realizaciones de X_t en el tiempo t . La realización x_t puede tomar cualquier valor $a_i \in A$. Además, sea $\mathbf{s} = (x_L, \dots, x_1)$ el vector ordenado, donde $\mathbf{s} \in S$, una secuencia en particular de largo L producida por la cadena M . Por producto cartesiano, $S = A^L$ es el conjunto de secuencias de largo L . Entonces la probabilidad de producir dicha secuencia, dada la cadena M , puede escribirse como:

$$p_M(\mathbf{s}) = p_M(x_1) \prod_{t=2}^L p_M(x_t | x_{t-1})$$

Por otra parte, por teorema de la probabilidad total, la probabilidad de generar una secuencia cualquiera \mathbf{s} con el proceso M puede descomponerse de la siguiente manera:

$$p_M(\mathbf{s}) = p_M(\mathbf{s} | \mathbf{s} \in S_c) p_M(\mathbf{s} \in S_c) + p_M(\mathbf{s} | \mathbf{s} \in S_c^\perp) p_M(\mathbf{s} \in S_c^\perp)$$

Donde $S_c \subset S$ el conjunto de secuencias de largo L que cumplen con la restricción, y S_c^\perp es su complemento, i.e., $S = S_c \cup S_c^\perp$. El problema exige que

se cumpla $\mathbf{s} \in S_c$, así que la probabilidad de producir tal secuencia con una cadena restringida debe ser proporcional a $p_M(\mathbf{s}|\mathbf{s} \in S_c)$.

La intención es construir una cadena de Markov restringida a partir de la cadena no restringida M . Suponga $A_t \subseteq A$ el espacio estado en el instante t , tal que $S_c = A_1 \times \dots \times A_L$. Más adelante se aclarará porqué imponer una restricción en L tiene consecuencias en los espacio estado anteriores. Es importante mencionar que es necesario que $A_t \neq \emptyset, \forall t$, para que se pueda generar la secuencia de largo L . Entonces la probabilidad de producir una secuencia, con la cadena M , que sea elemento de S_c , puede escribirse como:

$$p_M(\mathbf{s}|\mathbf{s} \in S_c) = p_M(x_1|x_1 \in A_1) \prod_{t=2}^L p_M(x_t|x_t \in A_t, x_{t-1})$$

Como $p_M(x_t|x_t \in A_t, x_{t-1}), \forall t$, son porbabilidades condicionales (al espacio estado A_t). Es necesario normalizar estas expresiones, de tal manera que sumadas sobre todo el espacio de las condicionales sea 1. De manera individual, estas probabilidades condicionales pueden ser normalizadas de la siguiente manera:

$$p_{\tilde{Z}}(X_1 = a_i) = \begin{cases} \frac{p_M(X_1 = a_i)}{\sum_{a_j \in A_1} p_M(X_1 = a_j)}, & \text{si } a_i \in A_1 \\ 0, & \text{si } a_i \notin A_1 \end{cases}$$

$$p_{\tilde{Z}}(X_t = a_i|X_{t-1} = a_h) = \begin{cases} \frac{p_M(a_i|a_h)}{\sum_{a_j \in A_t} p_M(a_j|a_h)}, & \text{si } a_i \in A_t, \\ 0, & \text{si } a_i \notin A_t \end{cases}, \quad \forall t = \{2, \dots, L\}$$

Lo anterior crea la cadena \tilde{Z} , como lo definen Pachet *et al.* (2011b, p. 638), el cual permite alcanzar la restricción mediante la normalización local de las probabilidades condicionales de transición. Entonces, la probabilidad de generar una secuencia $\mathbf{s} \in S_c$, con la cadena \tilde{Z} es:

$$p_{\tilde{Z}}(\mathbf{s}) = \frac{p_M(x_1|x_1 \in A_1)}{\sum_{x_1} p_M(x_1|x_1 \in A_1)} \times \prod_{t=2}^L \frac{p_M(x_t|x_t \in A_t, x_{t-1})}{\sum_{x_t} p_M(x_t|x_t \in A_t, x_{t-1})}$$

Sin embargo, el proceso \tilde{Z} no considera globalmente la probabilidad de generar una secuencia \mathbf{s} de largo L al momento de hacer la normalización, sino que lo hace localmente. En cambio, Pachet *et al.* (2011b, p. 638) definen la cadena \tilde{M} , que parte de normalizar la probabilidad de generar una secuencia $\mathbf{s} \in S_c$:

$$\begin{aligned} p_{\tilde{M}}(\mathbf{s}) &= \frac{p_M(\mathbf{s}|\mathbf{s} \in S_c)}{\sum_{\mathbf{s}} p_M(\mathbf{s}|\mathbf{s} \in S_c)} = \frac{1}{\alpha} p_M(\mathbf{s}|\mathbf{s} \in S_c) \\ &= \frac{p_M(x_1|x_1 \in A_1) \prod_{t=2}^L p_M(x_t|x_t \in A_t, x_{t-1})}{\sum_{x_1} \dots \sum_{x_L} [p_M(x_1|x_1 \in A_1) \prod_{t=2}^L p_M(x_t|x_t \in A_t, x_{t-1})]} \\ &= p_{\tilde{M}}(x_1) \prod_{t=2}^L p_{\tilde{M}}(x_t|x_{t-1}) \end{aligned}$$

Resulta evidente que $p_{\tilde{Z}}(\mathbf{s} \in S_c) \neq p_{\tilde{M}}(\mathbf{s} \in S_c)$. Esto implica que las secuencias generadas por la cadena \tilde{Z} son sesgadas, en términos globales de la secuencia de entrenamiento. Sin embargo, generar secuencias de \tilde{Z} es menos demandante computacionalmente hablando. Los elementos de las matrices de transición para la cadena \tilde{M} quedan definidos de la siguiente manera:

$$\begin{aligned} p_{\tilde{M}}(X_L = a_i | X_{L-1} = a_h) &= \begin{cases} \frac{p_M(a_i|a_h)}{\omega_{a_h}^{(L)}}, & \text{si } a_i \in A_L \\ 0, & \text{si } a_i \notin A_L \end{cases} \\ p_{\tilde{M}}(X_t = a_i | X_{t-1} = a_h) &= \begin{cases} \frac{\omega_{a_i}^{(t+1)} p_M(a_i|a_h)}{\omega_{a_h}^{(t)}}, & \text{si } a_i \in A_t, \forall t = \{2, \dots, L-1\} \\ 0, & \text{si } a_i \notin A_t \end{cases} \\ p_{\tilde{M}}(X_1 = a_i) &= \begin{cases} \frac{\omega_{a_i}^{(2)} p_M(a_i)}{\omega^{(1)}}, & \text{si } a_i \in A_1 \\ 0, & \text{si } a_i \notin A_1 \end{cases} \end{aligned}$$

Donde los factores y cocientes de estas expresiones están definidos como:

$$\omega_{a_j}^{(L)} = \sum_{a_i \in A_L} p_M(a_i|a_j)$$

$$\omega_{a_j}^{(t)} = \sum_{a_i \in A_t} \omega_{a_i}^{(t+1)} p_M(a_i | a_j), \forall t = \{2, \dots, L-1\}$$

$$\omega^{(1)} = \sum_{a_i \in A_1} \omega_{a_i}^{(2)} p_M(a_i)$$

Se puede notar que resolviendo recursivamente reemplazando en $\omega^{(1)}$ cada uno de los valores de $\omega^{(t)}$, para $t = 2, \dots, L$, se obtiene:

$$\omega^{(1)} = \sum_{x_1} \dots \sum_{x_L} \left[p_M(x_1 | x_1 \in A_1) \prod_{t=2}^L p_M(x_t | x_t \in A_t, x_{t-1}) \right]$$

Al estimar la probabilidad conjunta de generar una secuencia que satisfaga la restricción, cada uno de los cocientes $\omega_{a_j}^{(t)}, \forall t = 2, \dots, L$, y $\forall a_j \in A_t$, en t , se cancela con el factor $\omega_{a_j}^{(t)}$ en $t-1$. De esta manera, la probabilidad de generar una secuencia \mathbf{s} por el proceso restringido \tilde{M} , queda dada por:

$$\begin{aligned} p_{\tilde{M}}(\mathbf{s}) &= \frac{1}{\omega^{(1)}} p_M(x_1 | x_1 \in A_1) \prod_{t=2}^L p_M(x_t | x_t \in A_t, x_{t-1}) \\ &= p_{\tilde{M}}(x_1) \prod_{t=2}^L p_{\tilde{M}}(x_t | x_{t-1}) \end{aligned}$$

La expresión $1/\omega^{(1)}$ se puede interpretar como un factor de normalización de la probabilidad global de una secuencia. Este tipo de manipulaciones son frecuentes en probabilidad y estadística, cuando se quiere truncar una distribución. Por ejemplo, la distribución Poisson truncada en cero es un caso particular de la distribución Poisson tradicional. Si una variable X tiene una distribución Poisson truncada en cero, i.e., $X \sim ZTP(\lambda)$, entonces su soporte son los enteros positivos (en contraste con una variable con distribución Poisson tradicional, que tiene soporte sobre los enteros positivos incluyendo el 0). En consecuencia, la función de masa de probabilidad de la distribución Poisson truncada en cero difiere de la Poisson tradicional, en una constante que permite que la suma sobre todo su soporte sea 1.

A pesar de haber definido la normalización global de las cadenas de Markov, Pachet *et al.* (2011b, p. 639 – 640) discuten la necesidad de emplear dicha normalización, argumentando que la normalización global tiene un impacto pequeño sobre la generación de secuencias. Este argumento tiene sentido si se considera que al generar secuencias relativamente largas, la probabilidad de generar una secuencia en particular, de las muchas posibles, es muy pequeña. Por tal motivo, la programación del algoritmo solo considera la normalización local de las matrices de transición luego del procedimiento de arco-consistencia.

3.2. Cadena de Markov restringida de orden d

Defínase una cadena de Markov (no restringida) M con un espacio estado asociado $A = \{a_1, \dots, a_k\}$, $x_t \in A, \forall t = 1, \dots, L$, el estado en el tiempo t , $S = A^L$ el conjunto de secuencias de largo L , y $\mathbf{s} = (x_1, \dots, x_L) \in S$ una secuencia en particular de largo L . Sin embargo, en este caso la probabilidad de generar una secuencia \mathbf{s} mediante el proceso M está dada por:

$$p_M(\mathbf{s}) = p_M(x_1) \prod_{t=2}^L p_M(x_t | x_{t-1}, \dots, x_{t-\delta}), \quad \delta = \min\{t, d\}$$

Recuerde que los estados y prefijos pueden escribirse como un δ -grama: $\mathbf{y}_t^\delta = (x_t, \dots, x_{t-\delta+1}) \in A^{(\delta)}$, donde $A^{(\delta)}$ son todas las posibles secuencias de largo δ como combinaciones posibles de los elementos de A . En este caso, la secuencia \mathbf{s} puede representarse mediante estos δ -gramas como:

$$\mathbf{s} = \bigcap_{t=1}^L \mathbf{y}_t^\delta$$

Dado esto, la probabilidad de generar una secuencia de largo L usando la cadena no restringida M es equivalente a decir:

$$p_M(\mathbf{s}) = p_M(\mathbf{y}_1^\delta) \prod_{t=2}^L p_M(\mathbf{y}_t^\delta | \mathbf{y}_{t-1}^\delta)$$

En este punto resulta más conveniente usar la notación en términos de los δ -gramas. Con el fin de hacer cumplir la restricción se debe exigir arco-consistencia sobre cada uno de los conjuntos $A_t^{(\delta)}$, tal que $A_t^{(\delta)} \subseteq A^{(\delta)}$, para $t = 1, \dots, L$. Por ejemplo, si para satisfacer las restricciones es necesario restringir el conjunto $A^{(d)}$ de un prefijo de largo d en el momento t , entonces dicho subconjunto de $A^{(d)}$ será denominado $A_t^{(d)}$, el cual puede diferir del conjunto de otro prefijo de largo d en el momento r , el cual será $A_r^{(d)} \subseteq A^{(d)}$. Una explicación más detallada y un ejemplo para la comprensión de este concepto se dan en la próxima subsección, que trata sobre arco-consistencia.

La restricción se debe imponer sobre los espacios de los prefijos, sea por ejemplo $A_t^{(d)} \subseteq A^{(d)}$, para $t = 1, \dots, L - 1$, y no sobre los espacios estado anteriores. De lo contrario se está excluyendo la posibilidad de producir secuencias que efectivamente cumplan con la restricción. Por ejemplo, considere $\hat{H}|\mathbf{z}$; se quiere que $x_L = a$. Para esto es necesario evitar todos los estados anteriores que no lleven a ese resultado; es decir $\mathbf{y}_{L-1}^3 = (a, a, b)$ no debería generarse. Sin embargo, si se limita la ocurrencia de $x_{L-2} = a$, entonces otros prefijos que satisfacen la restricción, tales como (b, a, b) y (b, a, a) , no ocurrirían.

Definir la restricción como $x_L = a$, es equivalente a definir el espacio de los prefijos para L como $A_L^{(3)} = \{(a, b, a), (a, a, b), (a, b, b), (a, a, a)\}$. Note que en el caso del ejemplo no es posible generar la secuencia (a, a, a) bajo ninguna circunstancia, a pesar que ese estado hace parte del conjunto $A_L^{(3)}$.

El conjunto de secuencias de largo L que satisfacen la restricción es $S_c = A_1^{(1)} \times \dots \times A_L^{(\delta)}$. La cadena de Markov restringida \tilde{Z} se define de la misma manera que en orden 1, solo que ahora se consideran las secuencias anteriores de largo δ . Este proceso requiere una normalización de manera local de las probabilidades condicionales de transición. Entonces, la probabilidad de generar una secuencia $\mathbf{s} \in S_c$, con el proceso \tilde{Z} es:

$$p_Z(\mathbf{s}) = \frac{p_M(\mathbf{y}_1^1 | \mathbf{y}_1^1 \in A_1^{(1)})}{\sum_{\mathbf{y}_1^1} p_M(\mathbf{y}_1^1 | \mathbf{y}_1^1 \in A_1^{(1)})} \times \prod_{t=2}^L \frac{p_M(\mathbf{y}_t^\delta | \mathbf{y}_t^\delta \in A_t^{(\delta)}, \mathbf{y}_{t-1}^\delta)}{\sum_{\mathbf{y}_t^\delta} p_M(\mathbf{y}_t^\delta | \mathbf{y}_t^\delta \in A_t^{(\delta)}, \mathbf{y}_{t-1}^\delta)}$$

Asimismo, se puede definir la cadena de Markov restringida \tilde{M} , que normaliza las probabilidades de transición de manera global, tal que la probabilidad de generar una secuencia \mathbf{s} de largo L sea $p_{\tilde{M}}(\mathbf{s}) = \frac{1}{\alpha} p_M(\mathbf{s} | \mathbf{s} \in S_c)$. Este caso es equivalente al de la subsección anterior, con la diferencia que se utiliza los δ -gramas, y los ω quedan expresados en términos de los prefijos.

$$p_{\tilde{M}}(X_L = a_i | \mathbf{Y}_{L-1}^\delta = \mathbf{b}_h) = \begin{cases} \frac{p_M(a_i | \mathbf{b}_h)}{\omega_{\mathbf{b}_h}^{(L)}}, & \text{si } a_i \in A_L \\ 0, & \text{si } a_i \notin A_L \end{cases}$$

$$p_{\tilde{M}}(\mathbf{Y}_t^\delta = \mathbf{b}_i | \mathbf{Y}_{t-1}^\delta = \mathbf{b}_h) = \begin{cases} \frac{\omega_{\mathbf{b}_i}^{(t+1)} p_M(\mathbf{b}_i | \mathbf{b}_h)}{\omega_{\mathbf{b}_h}^{(t)}}, & \text{si } \mathbf{b}_i \in A_t^{(\delta)}, \forall t = \{2, \dots, L-1\} \\ 0, & \text{si } \mathbf{b}_i \notin A_t^{(\delta)} \end{cases}$$

$$p_{\tilde{M}}(\mathbf{Y}_1^1 = \mathbf{b}_i) = \begin{cases} \frac{\omega_{\mathbf{b}_i}^{(2)} p_M(\mathbf{b}_i)}{\omega^{(1)}}, & \text{si } \mathbf{b}_i \in A_1^{(1)} \\ 0, & \text{si } \mathbf{b}_i \notin A_1^{(1)} \end{cases}$$

De manera similar, los factores y cocientes de las expresiones corresponden a:

$$\omega_{\mathbf{b}_j}^{(L)} = \sum_{a_i \in A_L} p_M(a_i | \mathbf{b}_j)$$

$$\omega_{\mathbf{b}_j}^{(t)} = \sum_{\mathbf{b}_i \in A_t^{(\delta)}} \omega_{\mathbf{b}_i}^{(t+1)} p_M(\mathbf{b}_i | \mathbf{b}_j), \forall t = \{2, \dots, L-1\}$$

$$\omega^{(1)} = \sum_{a_i \in A_1} \omega_{\mathbf{b}_i}^{(2)} p_M(\mathbf{b}_i)$$

La interpretación de estos factores y cocientes es la misma a la que se presentó en la sección anterior. Sin embargo, note que en este caso los factores y cocientes están en función de los δ -gramas, y no de los estados.

3.3. Arco-consistencia

En el contexto de problemas de satisfacción de restricciones, la consistencia es una propiedad que permite alcanzar una solución al problema que satisfaga las restricciones (Pachet *et al.*, 2011b, p. 637). La arco-consistencia fue propuesta por Mackworth (1977, p. 104). Existen varios tipos de consistencia, entre ellas se encuentra la arco-consistencia. En el contexto de las cadenas de Markov restringidas, la arco-consistencia se basa en la eliminación los arcos que conducen a nodos que no satisfacen la restricción impuesta en el instante L , y de los nodos de los cuales proceden. Esto tiene implicancias sobre los nodos y arcos de los instantes anteriores al instante L .

Por ejemplo, si tenemos una cadena de Markov como el que se ilustra en la Figura 3, y se quisiera que $x_L = a$, entonces al ejercer arco-consistencia el modelo en grafo quedaría como en la Figura 5. Los arcos punteados representan aquellos que se deben eliminar para que se cumpla la restricción. Asimismo, el nodo punteado se deben eliminar para poder cumplir la restricción, es decir que se deben eliminar los arcos y nodos que llevan a $y_{L-1}^3 = (x_{L-1} = a, x_{L-2} = a, x_{L-3} = b)$.

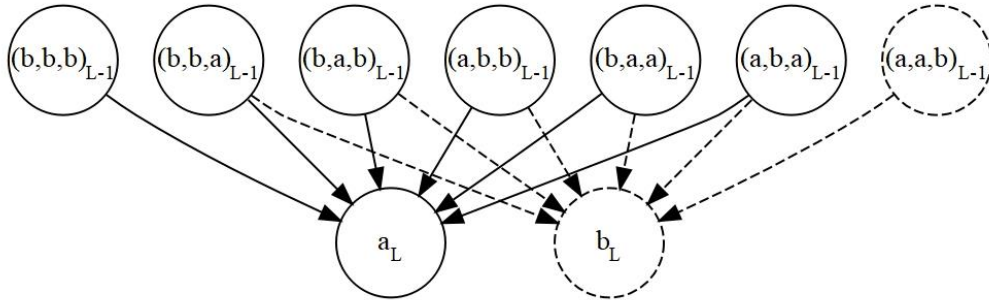


Figura 5. Representación del grafo asociado a la cadena de Markov $\hat{H}|\mathbf{z}$, en $t = L$, con la restricción $x_L = a$.

Al eliminar un nodo en los prefijos, es necesario modificar los arcos del modelo en grafo para $t = L - 1$, en un proceso llamado propagación hacia atrás. La Figura 6 presenta dicha modificación, que consiste en eliminar los arcos que provienen de $\mathbf{y}_{L-2}^3 = (a, b, a)$ y $\mathbf{y}_{L-2}^3 = (a, b, b)$ hacia $x_{L-1} = a$, evitando que se produzca la secuencia $\mathbf{y}_{L-1}^3 = (a, a, b)$.

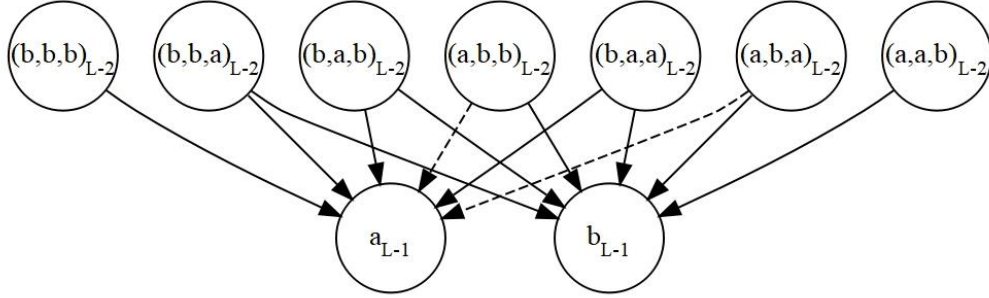


Figura 6. Representación del grafo asociado a la cadena de Markov $\hat{H}|\mathbf{z}$, en $t = L - 1$, con la restricción $x_L = a$.

Al eliminar algunos de los arcos, cambia la ponderación de los arcos que permanecen, dependiendo de si se emplea el proceso \tilde{Z} o el proceso \tilde{M} , descritos en la subsección anterior. Con el proceso \tilde{Z} solo se modifica la ponderación los modelos en grafo a los que se les elimina arcos, mientras que con \tilde{M} se afecta la ponderación de todos los modelos en grafo anteriores. En términos de matrices de transición, la eliminación de arcos implica que las columnas de la matriz de transición de los estados no permitidos debe ser 0, y la eliminación de algún nodo en el prefijo implica que el espacio estado de las matrices anteriores se debe modificar.

$$(\hat{H}|\mathbf{z}, t = L, x_L = a, \tilde{Z}) = \begin{pmatrix} 0 & 0 \\ 1 & 0 \\ 1 & 0 \\ 1 & 0 \\ 0 & 0 \\ 1 & 0 \\ 1 & 0 \\ 1 & 0 \end{pmatrix}, \quad (\hat{H}|\mathbf{z}, t = L - 1, x_L = a, \tilde{Z}) = \begin{pmatrix} 0 & 0 \\ 1/2 & 1/2 \\ 0 & 1 \\ 1/2 & 1/2 \\ 0 & 1 \\ 1/2 & 1/2 \\ 0 & 1 \\ 1 & 0 \end{pmatrix}$$

Acá se presenta las matrices de transición del ejemplo, para $t = L, L - 1$, luego de ejercer arco-consistencia y la normalización descrita por el proceso \tilde{Z} . También podría implementarse la normalización mediante el proceso \tilde{M} . Sin embargo, en el caso del ejemplo en cuestión las matrices de transición previas (que no requieren eliminación de arcos) son iguales a las no restringidas bajo el proceso \tilde{Z} , mientras que con la normalización del proceso \tilde{M} todas las matrices previas cambian.

3.4. Condiciones para la satisfacción de las restricciones

En la subsección anterior se menciona brevemente las condiciones para la satisfacción de las restricciones, desde la perspectiva de arco-consistencia y con enfoque a la representación de las cadenas de Markov como modelos en grafo. Esta subsección tiene la intención de profundizar más en el tema.

Las restricciones sobre una secuencia de largo L se pueden clasificar en varios tipos: (i) una sola restricción al inicio de la secuencia, (ii) una sola al final de la secuencia, (iii) dos restricciones, una al inicio o al final de la secuencia. Los casos en los que se tiene solo una restricción en medio de la secuencia, se puede entender como la unión de los casos (i) y (ii). En cambio, si se tienen dos restricciones en medio de la secuencia, se puede entender como una unión de tres casos.

Para el tipo (i), basta restringir el espacio estado, $A_0 \subset A = \{a_1, \dots, a_k\}$. De forma natural el proceso va a generar el resto de la secuencia de largo L . En este caso, si $X_1 = a_j \in A_1 \subset A$, donde a_j un estado absorbente, el resto de la secuencia será $(X_L = a_j, \dots, X_1 = a_j)$. Por esta razón no es necesario exigir que los estados del proceso sean todos recurrentes, pero resulta conveniente para tener variabilidad en la secuencia generada para este tipo de restricción.

Para el tipo (ii), es necesario hacer arco-consistencia, tal que no se generen secuencias que no cumplan con la restricción. De igual manera, no es necesario exigir la ausencia de estados absorbentes, pero sucede algo similar que en el tipo (i): si la restricción demanda que $X_L = a_j \in A_L \subset A$, donde a_j es

un estado absorbente, podría suceder que se llegue a ese estado antes. Por otra parte, si se desea continuar generando una secuencia luego de $t = L$, esa secuencia necesariamente será $X_r = a_j$, para $r > L$.

Por otra parte, cumplir las restricciones de tipo (iii) requiere varias condiciones adicionales. Acá se requiere hacer arco-consistencia, de modo que no se genere una secuencia que no permita satisfacer la restricción en $A_L \subset A$. Esto implica que $A_r \subseteq A$ para $1 \leq r < L$. Pero además se debe exigir que $A_1 \neq \emptyset$, para poder iniciar la secuencia. Otra manera de verlo es mediante $\psi_{a_i a_j}^{(L-1)}$, la probabilidad que el proceso alcance el estado a_j , partiendo del estado a_i , luego de $L - 1$ pasos, en el proceso no restringido: si $\psi_{a_i a_j}^{(L-1)} > 0$ para algún $a_i \in A_1$ y algún $a_j \in A_L$, es posible satisfacer la restricción. La probabilidad $\psi_{a_i a_j}^{(L-1)}$ podría estimarse considerando el resultado de la ecuación de Chapman-Kolmogorov, mediante la multiplicación de la matriz de transición $L - 1$ veces, pero basta con verificar que $A_r \neq \emptyset$, para $1 \leq r < L$, para garantizar que la restricción se puede alcanzar.

4. IMPLEMENTACIÓN

Para implementar el algoritmo descrito en este documento se programaron las funciones necesarias en R, que están contenidas en el paquete `MarkovMusic`, disponible en el anexo digital. Además de las funciones programadas, el paquete tiene ejemplos de su uso en la documentación. Pero en esta sección se presenta una descripción de los contenidos del paquete, así como la implementación del algoritmo, que corresponde a los resultados presentados en esta sección y demás elementos contenidos en el anexo digital.

Los datos de entrenamiento están en XML, que es un formato usual soportado por distintos software para componer piezas musicales. En el anexo digital está la melodía de entrenamiento completa, en formatos XML, PDF y MIDI¹. Adicionalmente, la melodía de entrenamiento también está incluida en el paquete `MarkovMusic`.

Existen muchos softwares para escribir y visualizar partituras. Durante el desarrollo de este trabajo se utilizó el software MuseScore 2² para componer la melodía de entrenamiento y visualizar las melodías producidas por el algoritmo. MuseScore 2 es un software libre, publicado bajo licencia GNU, que permite leer y escribir en formato XML, PDF y MIDI, entre otros.

A continuación se explica brevemente los contenidos del paquete, su funcionamiento y se presentan algunos resultados.

4.1. Descripción del paquete `MarkovMusic`

El paquete `MarkovMusic` contiene las funciones programadas para implementar el algoritmo. La documentación del paquete está disponible para la consulta del usuario, con ejemplos para el uso de cada una de las funciones. Sin embargo, en esta subsección se describen las funciones principales de este paquete y se presenta el código para la generación de las melodías de blues de

¹ No confundir con la notación musical MIDI.

² Disponible en la página web: <https://musescore.org/en/download>

doce compases. Antes de instalar el paquete, es necesario instalar el paquete XML disponible en el repositorio de CRAN³.

El paquete tiene la función `readMusicXML` que permite leer archivos en formato XML, y convertirlos en una lista de objetos que incluye una matriz con los tonos y la duración de las notas, además de otros atributos como la clave y la métrica del compás. Solo es posible leer melodías monofónicas, con una métrica invariante y que no presenten valores irregulares (como tresillos, etc.). La función `fromMidiToNote` permite convertir las melodías de notación MIDI a notación anglosajona.

Las funciones `priorVec` y `transMat` permiten estimar el vector de probabilidad inicial y las matrices de transición respectivamente. Si se desea utilizar más de un proceso, es decir generar una melodía que mezcle dos escalas distintas, es necesario especificar el estado espacio correspondiente a todos los tonos de ambas escalas en conjunto.

La función `arcConsistency` crea la arco-consistencia entre las matrices de transición, dada unas condiciones. Esta función se aplica sobre una lista de matrices y recibe como argumento una lista de restricciones a implementar. Como resultado entrega una lista de matrices que son arco-consistentes con las restricciones. Sin embargo, estas matrices no están normalizadas. Lo anterior no es problema pues la función `rmultinom` que se utiliza para generar la melodía normaliza el vector de probabilidades que recibe como argumento. Esto quiere decir que la función entrega como resultado un conjunto de matrices pertenecientes a una cadena \tilde{Z} , como la descrita en la sección anterior.

La función `genMelody` recibe como argumentos la cadena obtenido mediante la función `arcConsistency`, el espacio estado de los procesos

³ Para instalar el paquete XML puede escribir el comando `install.packages("XML")` en la consola de R. Para instalar el paquete `MarkovMusic` puede escribir el comando `utils::menuInstallLocal()` en la consola y seleccionar el archivo fuente "MarkovMusic_0.1.0.tar.gz".

utilizados, y la duración de todas las notas producidas. Y la función `writeMusicXML` permite exportar la melodía generada a formato XML.

El paquete también incluye los datos de entrenamiento, bajo el nombre "TrainingBlues.xml". En la próxima subsección se presenta la manera en que se emplean las funciones del paquete para producir un blues de doce compases.

4.2. La generación de un blues de doce compases

El propósito es generar una secuencia de notas que ajuste a la estructura armónica de un blues de doce compases: una secuencia de 144 notas corcheas. En el contexto de generación de secuencias usando cadenas de Markov esto significa que durante los primeros cuatro compases la generación de la secuencia se hace con una cadena de Markov basado en una melodía de la escala menor de blues en C , luego, durante el quinto y sexto compás se genera la secuencia a partir de un proceso basado en la escala menor de blues en F , etc.

Si bien los procesos asociados a las escalas son distintos, tienen elementos en común. Para el caso de las escalas menores de blues en C y F , ambas comparten las notas C , F y Bb . En este sentido, para crear coherencia entre la transición de un proceso a otro, se utiliza la cadena de Markov restringida, que hace que la generación de la secuencia de un proceso sea compatible con la del proceso siguiente.

Los datos de entrenamiento consisten en una melodía de 528 notas (corcheas) en la escala menor de blues en C . Esta melodía de entrenamiento se emplea para calcular las probabilidades de transición del proceso Cm . La Figura 7 muestra una fracción de la melodía de entrenamiento. Además se utiliza la melodía de entrenamiento para calcular las probabilidades de transición de los procesos Fm y Gm , trasponiéndola cinco y siete semitonos hacia arriba respectivamente.



Figura 7. Primeros nueve compases de la melodía de entrenamiento.

Con las melodías de entrenamiento se calculan las matrices de probabilidad de transición no restringidas, luego se aplica la arco-consistencia con el fin de impedir que se generen secuencias que no satisfagan restricciones impuestas. Con el proceso *Cm* se desea generar las primeras 48 notas, con *Fm* las siguientes 24, nuevamente con *Cm* otras 24, luego se generarán 12, 12 y 24, con *Gm*, *Fm* y *Cm* respectivamente, tal que se ajuste a la progresión armónica descrita en la Figura 1.

Las restricciones que se imponen son: (i) empezar la melodía en *C*, (ii) iniciar el quinto compás en *F*, (iii) inicia el séptimo compás en *C*, (iv) iniciar el noveno compás en *G*, (v) iniciar el décimo compás en *F*, (vi) iniciar el onceavo compás en *C*, y (vii) finalizar la melodía en *C*.

El código en R que aparece a continuación demuestra cómo utilizar las funciones del paquete para generar una secuencia de notas para el blues de doce compases, usando cadenas de Markov de orden 2.

```

library(MarkovMusic)

# Lectura de los datos de entrenamiento.
Cm <- readMusicXML(paste0(system.file(package = "MarkovMusic"),
                           "/TrainingBlues.xml"))
Cm <- Cm$song[, "note"] # Se preserva solo la información del tono.
Cm <- c(Cm, rev(Cm)) # Melodía en escala menor de blues en C.
Fm <- Cm + 5 # Melodía en escala menor de blues en F.
Gm <- Cm + 2

# Conversión de la melodía en notación MIDI a notación anglosajona.
Cm <- fromMidiToNote(Cm)
Fm <- fromMidiToNote(Fm)
Gm <- fromMidiToNote(Gm)

# Espacio estado común para las tres secuencias.
state.space <- unique(unlist(c(Cm, Fm, Gm)))

# Matrices de transición de los procesos.
M0Cm <- priorVec(Cm, state.space = state.space) # Prob. inicial.
M1Cm <- transMat(Cm, state.space = state.space) # Orden 1.
M2Cm <- transMat(Cm, order = 2, state.space = state.space) # Orden 2.
M2Fm <- transMat(Fm, order = 2, state.space = state.space)
M2Gm <- transMat(Gm, order = 2, state.space = state.space)

# Cadenas de Markov no restringidas.
M <- c(list(M0Cm), list(M1Cm), rep(list(M2Cm), 46), # compases 1-4.
      rep(list(M2Fm), 24), # 24 notas, compases 5 y 6.
      rep(list(M2Cm), 24), # 24 notas, compases 7 y 8.
      rep(list(M2Gm), 12), # 12 notas, compás 9.
      rep(list(M2Fm), 12),
      rep(list(M2Cm), 24))

# Definición de las restricciones.
constraints <- list(list(1, "C"), # La primera nota debe ser C.
                   list(49, "F"), # La nota 49 debe ser F.
                   list(73, "C"),
                   list(97, "G"),
                   list(109, "F"),
                   list(121, "C"),
                   list(144, "C"))

# Acro-consistencia.
Z <- arcConsistency(M, constraints)

# Generación de la melodía.
set.seed(69)
melody <- genMelody(Z, state.space)
writeMusicXML(melody, file = "MarkovMelody2_69.xml",
              beat = 12, beat.type = 8)

```

El resultado de ejecutar este código es la melodía que aparece en la Figura 8. Una vez se define la cadena de Markov restringido, es fácil generar otras melodías distintas. Por eso se generaron once melodías, con cadenas de orden 1 y 2. Las veintidós melodías generadas, incluyendo la presentada en la Figura 8, se incluyen en el anexo digital.

Al intentar generar melodías con cadenas de Markov de orden 3 el algoritmo presenta un error que corresponde a la incapacidad de ejercer arco-consistencia pues el espacio estado de algún instante es vacío. Una posibilidad de enfrentar este tipo de situaciones es relajar un poco las restricciones impuestas, buscando que se pueda producir la melodía. Por ejemplo, para relajar los supuestos se puede incluir otros estados en la restricción: que la primera nota del quinto compás pueda ser C , además de F o G , y lo mismo en el noveno compás.

En general es importante considerar las características de la melodía de entrenamiento, para poder decidir el orden que se desea utilizar. Se considera una buena práctica ser cuidadoso con que las notas iniciales y finales de la melodía de entrenamiento se aparezcan también en medio de la melodía, cuando se desea usar una cadena de orden 1. Esto implica que no hay estados absorbentes o transientes, lo cual es conveniente por las razones enunciadas en la sección anterior. Sin embargo, para usar procesos de orden d no solo se debe estudiar el comportamiento de la melodía de entrenamiento como secuencia de estados, sino el comportamiento de los d -gramas que la conforman.

Finalmente, al comparar las melodías generadas mediante las cadenas de orden 1 y 2, se puede notar que las primeras tienen poco sentido de frase, frente a las melodías producidas por las cadenas de orden 2. Si bien esto es una simple apreciación estética, tiene sentido pues las melodías producidas con las cadenas de orden 2 tienen mayor memoria.



Figura 8. Melodía generada por el algoritmo usando cadenas de Markov de orden 2 (semilla 69).

5. CONCLUSIONES

Las cadenas de Markov restringidas permiten generar secuencias melódicas que se ajusten a cierta estructura armónica. En este proyecto se aplica a la generación de una melodía para un blues de doce compases en C menor, con métrica 12/8.

Sin embargo, sería interesante aplicar este algoritmo a la generación de melodías que se ajusten a progresiones armónicas más complejas como las de jazz y funk, u otras variaciones de blues. Por ejemplo, la progresión armónica de la canción *Virtual Insanity* de Jamiroquai es de la forma $Ebm7 - Ab7 - Db9 - Ebm9 - Ab9 - Abm9 - Daug$. En este caso no basta con tener una única melodía de entrenamiento como se hace en este documento con la melodía basada en la pentatónica menor de blues en C , que se traspone y utiliza para definir las melodías que servirán de entrenamiento para los compases de Fm y Gm . En cambio se debe considerar melodías de entrenamiento distintas que compaginen con cada acorde.

La dimensión de las melodías de entrenamiento también es un aspecto importante de la implementación. En este proyecto se utilizó una melodía de entrenamiento de 528 notas, pero incluso al intentar generar una secuencia una cadena de orden 3 que fuera arco-consistente con las restricciones se obtuvieron espacios estados vacíos. Considerando esto, si se desea utilizar procesos de orden superior debe utilizarse melodías de entrenamiento más largas, que cuenten con más información y variabilidad de los d -gramas.

Una limitación de este algoritmo es que no considera una forma de modelar la duración de cada nota. Todas las notas producidas por el algoritmo tienen una duración que se determina por el usuario. Esto no es un obstáculo trivial, pues la escala de una melodía no sólo se identifica por las notas que utiliza, sino también por su predominancia.

BIBLIOGRAFÍA

- Blanco, L., Arunachalam, V., & Dharmaraja, S. (2012). *Introduction to Probability and Stochastic Processes with Applications*. E.E.U.U.: Wiley & Sons.
- Brinkkemper, F. (octubre 5, 2016). Analyzing Six Deep Learning Tools for Music Generation. *The Asimov Institute*. Recuperado el 27 julio de 2017, de: <http://www.asimovinstitute.org/analyzing-deep-learning-tools-music/>
- Hoel, P.G., Port, S.C., & Stone, C.J. (1972). *Introduction to Stochastic Processes*. E.E.U.U.: Waveland Press.
- Kulkarni, V.G. (2010). *Introduction to Modeling and Analysis of Stochastic Systems* (2da Edición). E.E.U.U.: Springer.
- Mackworth, A. K. (1977). Consistency in Networks of Relations. *Artificial Intelligence*, 8, pp. 99-118. North-Holland Publishing Company.
- Pachet, F., & Roy, P. (2011a). Markov Constraints: Steerable Generation of Markov Sequences. *Constraints*, 16, pp. 148 – 172. Springer.
Recuperado el 27 de abril de: 2017 de
<https://www.csl.sony.fr/downloads/papers/2011/pachet-09c.pdf>
- Pachet, F., Roy, P., & Barberi, G. (2011b). Finite-Length Markov Processes with Constraints. *Proceedings of the Twenty-Second International Joint Conference on Artificial Intelligence*, pp. 635 – 642. Recuperado el 27 de abril de 2017, de:
<https://www.csl.sony.fr/downloads/papers/2011/pachet-11b.pdf>

- Papadopoulos, A., Roy, P., & Pachet, F. (2014). Avoiding Plagiarism in Markov Sequence Generation. *Proceedings of the Twenty-Eight International Joint Conference on Artificial Intelligence*, pp. 2731 – 2737. Recuperado el 27 de abril de 2017, de:
<https://www.csl.sony.fr/downloads/papers/2014/papadopoulos-14a.pdf>
- Rincón, L. (2012). *Introducción a los procesos estocásticos*. Universidad Nacional Autónoma de México, Facultad de Ciencias – Departamento de Matemáticas. México. Recuperado el 27 de octubre de 2017, de:
<http://lya.fciencias.unam.mx/lars/libros/procesos2012.pdf>