Sociedad de Ingeniería de Audio
Artículo de Congreso
Congreso Latinoamericano de la AES 2018
24 a 26 de Septiembre de 2018
Montevideo, Uruguay

Generación automática de melodías usando cadenas de Markov con restricciones

Verónica Rumbo,[1] Ernesto Mordecki[2] y Martín Rocamora[3]

[1] Facultad de Ingeniería Universidad de la República, Uruguay.

[2] Centro de Matemática, Facultad de Ciencias, Universidad de la República, Uruguay.

[3] Instituto de Ingeniería Eléctrica, Facultad de Ingeniería, Universidad de la República, Uruguay.

vrumbo@cmat.edu.uy, mordecki@cmat.edu.uy, rocamora@fing.edu.uy

RESUMEN

Proponemos una técnica para simular melodías aleatorias utilizando cadenas de Markov con restricciones para determinar las alturas de los sonidos cuyas duraciones han sido establecidas previamente. El empleo de restricciones favorece la creación de melodías a partir de otras preexistentes a modo de variaciones, preservando unas pocas notas de referencia y la tonalidad o la armonía subyacente.

0. CADENAS DE MARKOV CON RESTRICCIONES

Nuestro objetivo es generar aleatoriamente secuencias de símbolos, que luego asociaremos con (alturas de) notas musicales. Para ello una primera opción puede ser el uso de cadenas de Markov homogéneas, en las que la probabilidad de aparición de cada símbolo depende exclusivamente de cuál fue el simbolo anterior. Es decir, tenemos secuencias de variables aleatorias $X_1, X_2, \ldots$ que toman valores en el espacio de estados $S = \{0, 1, \ldots, K\}$ (asumimos que es finito) y que verifican

$$P(X_n = j | X_{n-1} = i, X_{n-2} = i_{n-2}, \ldots, X_0 = i_0) =$$
$$= P(X_n = j | X_{n-1} = i) \; \forall i, j \in S.$$

Así, una cadena de Markov (homogénea) queda definida por una *matriz de transición* $M \in \mathcal{M}_{K \times K}$ y el vector de probabilidades del primer símbolo $\mu \in \mathbb{R}^{\mathbb{K}}$, conocido como *distribución inicial*.

Si pensamos que los estados son notas, el uso de este tipo de cadenas de Markov parece muy poco apropiado en tanto no permite ningún tipo de control a futuro sobre el comportamiento de las secuencias. Para solucionarlo Pachet, Roy y Barbieri (2011) [1] proponen el uso de ciertas cadenas de Markov no homogéneas, construidas a partir de una cadena homogénea (i.e un par (M, μ) como antes) dada de modo que verifiquen ciertas *restricciones*.

Supongamos que queremos trayectorias finitas

$x_0, x_1, \ldots, x_N$ a partir de (M, μ), que verifiquen ciertas condiciones en los distintos instantes $0, 1, \ldots, N$. Dichas condiciones podemos clasificarlas como

- **Restricciones unitarias:** son condiciones que refieren a un único instante $k \in \{0, 1, \ldots N\}$ y consisten en indicar cuáles son los estados que puede tomar la variable X_k. Llamaremos $U_k \subset S$ al conjunto de los estados que pueden ser visitados en el instante k.
- **Restricciones binarias:** son condiciones que refieren a dos instantes consecutivos $k-1$ y k ($k \in \mathbb{N}$) e indican cuáles son las transiciones permitidas de X_{k-1} a X_k. Llamaremos $B_k \subset S \times S$ al conjunto de tales transiciones.

Además buscamos que $\tilde{P}$, probabilidad que el nuevo modelo asigna a cada trayectoria sea proporcional a P, probabilidad bajo la cadena homogénea, es decir

$$\tilde{P}(s) = \begin{cases} 0 & \text{si } s \notin T'_N, \\ P(s|s \in T'_N) & \text{si } s \in T'_N, \end{cases} \tag{1}$$

donde T'_N es el conjunto de trayectorias de largo N que satisfacen las restricciones.

Pachet et al., (2011) proponen un algoritmo que permite calcular las matrices de transición $\{\tilde{M}_1, \ldots, \tilde{M}_N\}$ y distribución inicial $\tilde{\mu_0}$ de la nueva cadena, y a partir del cual simularemos las notas de nuestras melodías.

1. EL ALGORITMO

Un aspecto no menor en la implementación de restricciones, es que deben definirse de modo tal que toda trayectoria que comience verficándolas debe poder finalizarse. Para ello, cada vez que impongamos una restricción debemos propagarla del siguiente modo

- **Fijación de estados:** si se quiere imponer una condición de la forma $U_k = \{a\}$, hay que eliminar de U_{k+1} todos los estados a los que no se puede acceder desde a. De modo similar, hay que quitar de U_{k-1} los estados que no pueden ir hacia a.
- **Remoción de estados:** Cada vez que se quite un estado a del conjunto U_k, habra que quitar de U_{k+1} todos los estados a los que sólo se puede acceder desde a. De U_{k-1} quitaremos los estados que sólo podían ir hacia a.

El proceso termina cuando en los pasos anteriores no hay más nada por hacer. Considerando las matrices de transición, el procedimiento consiste en construir una familia auxiliar de matrices $\{Z^{(n)}\}_{\{i \in 1, \ldots, N\}}$ que indica que estados y transiciones están permitidos (generalmente no son estocásticas). Luego se renormalizan las matrices obtenidas de modo que sean estocásticas[1] y generen trayectorias con las probabilidades deseadas.

[1] Abusando de la nomenclatura, admitiremos por estocásticas matrices que tengan filas de ceros.

Construimos la familia $\{Z^{(k)}\}_{n \in \{0, \ldots N\}}$ del siguiente modo:

- **Inicialización.** Definimos $Z^{(0)} = \mu$ y $Z^{(k)} = P \; \forall n \in \{1, \ldots, N\}$.
- **Remoción de estados.** Llamemos $z_{ij}^{(k)}$ a la entrada i, j de la matriz $Z^{(k)}$. Para cada $j \in E$ removido de U_n, se establece $z_{ij}^{(k)} = 0 \; \forall i \in E$ (es decir, se lleva la j-ésima columna de la matriz a 0).
- **Remoción de transiciones.** Las transiciones prohibidas imponen ceros en las matrices del siguiente modo: $\forall i, j \in E, \; k \in \{1, \ldots, N\}$ tales que $(i, j) \notin B_n$ se establece $z_{ij}^{(k)} = 0$.

A partir de las matrices $Z^{(k)}$ se construyen las matrices de transición $\tilde{P}_n$ y la distribución inicial $\tilde{\mu}$, cuyas entradas se definen recursivamente

$$\tilde{p}_{ij}^{(N)} = \frac{z_{ij}^{(N)}}{\alpha_i^{(N)}}, \qquad \alpha_i^{(N)} = \sum_{l \in S} z_{il}^{(N)},$$

$$\tilde{p}_{ij}^{(k)} = \frac{\alpha_j^{(k+1)} z_{ij}^{(k)}}{\alpha_i^{(k)}}, \quad \alpha_i^{(k)} = \sum_{l \in S} \alpha_l^{(k+1)} z_{il}^{(k)}, \; k < N$$

$$\tilde{\mu}_i = \frac{\alpha_i^{(1)} z_i^{(0)}}{\alpha^{(0)}}, \quad \text{con } \alpha^{(0)} = \sum_{l \in S} \alpha_l^{(1)} z_l^{(0)}.$$

En caso de que $\alpha_i^{(n)} = 0$ impondremos $p_{ij}^{(n)} = 0$.

2. EJEMPLO: VARIACIÓN DE UNA MELODÍA PREEXISTENTE

Aplicamos el modelo presentado en la sección anterior para generar variaciones de una melodía preexistente. Utilizamos la biblioteca de python *music21* [2].

En el siguiente ejemplo consideramos una melodía preexistente (“Arroz con leche”) y simulamos otras que puedan percibirse como variantes del tema original. Para ello preservamos notas que oficiarán como restricciones unitarias de un paseo al azar modificado para permitir transiciones de cada estado en sí mismo. Se conservan las notas que aparecen en cada cambio de acorde, y se considera como espacio de estados el conjunto de las notas del acorde subyacente. Además, generamos melodías en las que los patrones rítmicos entre restricciones también son aleatorios. Para ello utilizamos el *modelo jerárquico* propuesto por Temperley (2010) [3]. En la Figura 1 puede observarse la partitura de una de estas variaciones, con altura y ritmo aleatorios.

3. UNA CONSIDERACIÓN SOBRE LAS MELODÍAS GENERADAS

Las melodías simuladas con este procedimiento se perciben con una fuerte estructura métrica y tonal. En contrapartida, resultan poco interesantes en tanto permitimos únicamente el uso de notas del acorde subyacente. Una posible mejora consiste en ampliar el diccionario a notas dentro de una escala asociada al acorde.

Figura 1: Arriba, melodía de “Arroz con leche”, con los acordes correspondientes. Debajo, ejemplo de variación generada con el procedimiento descrito. Las notas señaladas son las que mantuvieron sus alturas de la melodía original.

REFERENCIAS

[1] François Pachet, Pierre Roy, and Gabriele Barbieri, “Finite-length markov processes with constraints,” *Proceedings of the Twenty-Second International Joint Conference on Artificial Itelligence*, pp. 638–642, 2011.

[2] “music21: A toolkit for computer-aided musicology,” http://web.mit.edu/music21/, Visitada el 24/1/2018.

[3] David Temperley, “Modeling common-practice rhythm,” *Music Perception*, vol. 27, no. 5, pp. 355–376, 2010, Available at https://openmusiclibrary.org/article/37125/.