

Universidad Nacional de Colombia
Facultad de Ciencias
Departamento de Matemáticas



Creación de canciones mediante composición algorítmica musical utilizando Cadenas de Markov

Proyecto final del curso Cadenas de Markov

Inka Michelle Hernández Vásquez
Daniel Santiago López Daza
Juliana Alejandra Nieto Cárdenas

Enlace al vídeo:

 Video Proyecto Cadenas de Markov

Profesor: Freddy Rolando Hernández Romero

9 de diciembre de 2025

ÍNDICE:

1. Resumen.
2. Introducción.
3. Marco teórico.
 - 3.1 Composición algorítmica computacional.
 - 3.2 Composición algorítmica utilizando cadenas de Markov.
 - 3.3 Analogique A.
4. Métodos actuales
 - 4.1 Método tradicional.
 - 4.2 Cadenas de Markov e interpolación de Lagrange.
5. Implementación del modelo y análisis de resultados.
 - 5.1 Desarrollo.
 - 5.2 Análisis de resultados.
 - 5.3 Trabajo futuro.
6. Conclusiones.
7. Bibliografía.

1. Abstract

En este documento se examina la capacidad de las cadenas de markov en la generación y estructuración algorítmica musical, retomando metodologías clásicas y sus diferentes intervenciones orientadas a capturar tendencias y emularlas sin comprometer la originalidad de los resultados. Se describen los fundamentos teóricos asociados a la computación algorítmica de melodías, el proceso de entrenamiento del modelo, trabajos relevantes, su alcance y principales contribuciones.

Finalmente, se presenta el diseño e implementación de un sistema de composición algorítmica basado en modelos estocásticos, centrado en la aplicación de cadenas de markov.

2. Introducción

El deseo de formalizar, sistematizar y estructurar algorítmicamente el entorno y los acontecimientos que en él se desarrollan resulta de algo inherente a la naturaleza humana. Desde los primeros registros del desarrollo de la capacidad de raciocinio de la especie humana, esta ha buscado la manera de **describir y comprender el mundo** como una secuencia finita de pasos o eventos, desde una simple receta de cocina hasta la formalización de estructuras sociales, físicas y matemáticas, dando lugar a los fundamentos que sostienen el funcionamiento de las sociedades modernas.

El arte y sus diferentes manifestaciones, en este caso, la música, a pesar de su carácter expresivo y su vínculo con la emoción humana no ha sido una excepción a esta tendencia. A lo largo de la historia se ha intentado describir la música mediante reglas siendo la composición algorítmica un tema de interés incluso antes de la llegada de la computación. A partir de ello surge la siguiente pregunta:

¿Cómo automatizar la creación musical sin comprometer la originalidad y creatividad características de la producción humana?

De este problema surge el atractivo de trabajar con procesos estocásticos, en particular, las cadenas de markov, dada su capacidad de capturar patrones existentes en obras humanas para generar nuevas secuencias de este material sin necesidad de comprometer la originalidad al reducirse a una simple repetición.

En el presente documento se explora esta idea desde un enfoque tanto teórico como práctico.

3. Marco teórico - Una mirada al pasado

"Algorithmic composition, the process of using some formal process to make music with minimal human intervention" (Alpern, 1995)

3.1. Composición algorítmica computacional

Los primeros documentos que trabajan con la composición algorítmica remontan al siglo XX. La primera composición computacionalmente generada de la cual se tiene registro es la **ILLIAC Suite**, atribuida a **Lejaren Hiller** y **Leonard Isaacson** en la Universidad de Illinois. Este proyecto consistió en una integración de experimentos relacionados con la computación asistida. Entre los métodos abordados en el resultado final se evidencia un fuerte enfoque probabilístico (este utilizaba reglas determinísticas para mantener estilo y coherencia), siendo las cadenas de markov con orden variable uno de los principales métodos utilizados para seleccionar notas, sucesiones de saltos, establecer el orden de las tonalidades, entre otros, permitiendo capturar tendencias locales e imitar estilos musicales al mismo tiempo que garantiza cierto nivel de variabilidad, no obstante, **estos primeros acercamientos aún necesitaban cierto grado de intervención humana** para traducir los resultados y seleccionar aquellos con mayor consistencia.

La composición algorítmica puede ser abordada desde tres enfoques: probabilístico, determinista y utilizando inteligencia artificial. En el presente documento se trabajará a partir de este primer enfoque centrándose específicamente en el uso de cadenas de markov para la generación musical.

3.2. Composición algorítmica utilizando cadenas de markov

Definición. (*Cadena de orden superior*) Una cadena de orden N es aquella en la que el resultado en una iteración X_n depende de los últimos N estados, es decir $X_{n-1}, X_{n-2} \dots X_{n-N}$.

Nota: a medida que aumenta el orden de una cadena de markov su matriz de transición pasa de ser una matriz a un tensor de N+1 dimensiones.

El uso de cadenas de Markov en la composición musical consiste en entrenar un modelo a partir del análisis de una base de datos musical. En dicho análisis se estudian las frecuencias, notas, patrones rítmicos y demás parámetros relevantes. Con base en estas relaciones se construye una matriz de transición que recoge la probabilidad de pasar de un evento musical a otro. Con esta matriz, la generación de nueva música se realiza de forma que cada evento depende de una secuencia de estados previos (Véase *Métodos Actuales*).

Una de las principales desventajas del análisis y automatización de composiciones utilizando cadenas de Markov consiste en que estos modelos solo pueden trabajar con dependencias a corto plazo, es decir, que solo consideran el contexto inmediato (un orden bajo) puesto que a medida que se toma en cuenta información distante el orden de la cadena aumenta y así mismo la dimensión de su tensor de transición correspondiente, necesitando una mayor cantidad de datos de entrenamiento a fin de evitar resultados incoherentes dada la escasez de datos resultando en proceso computacionalmente costoso, en el que de hecho, el modelo pierde la capacidad de “crear” música, puesto que este solo empieza a plagiar alguno de los contenidos suministrados, no obstante, esto no implica la necesidad de descartar los modelos de Markov para la creación de composiciones, puesto que dado condiciones, entrenamientos y prototipos bien establecidos se puede llegar a resultados interesantes.

3.3 Analogique A (*Xenaki*)

Iannis Xenaki es considerado uno de los principales pioneros en el uso de la cadenas de markov en composición musical.

Definición. (Pantalla) Una pantalla es una estructura matemática que limita los valores que pueden tomar un conjunto de variables estocásticas.

En la composición de Analogique A, Xenaki impone una estructura de Markov sobre los parámetros **F** (frecuencia), **G** (intensidad) y **D** (densidad, o cantidad de notas en un instante de tiempo) para que estos solo puedan tomar dos valores posibles. Así mismo, postula dos matrices de transición para determinar el comportamiento de estas variables. Por ejemplo:

$$\mathbf{P}_1 = \begin{bmatrix} 0.2 & 0.8 \\ 0.8 & 0.2 \end{bmatrix} \quad \mathbf{P}_2 = \begin{bmatrix} 0.85 & 0.15 \\ 0.4 & 0.6 \end{bmatrix}$$

Matrices de transición para las pantallas. Tomado de [2]

Así, el valor de la pantalla en un instante depende de la matriz seleccionada.

Para determinar qué matriz se utilizará para cada parámetro, se tienen en cuenta la información que dan los otros dos. Por ejemplo, para ver la matriz que modela el comportamiento de **G** en el siguiente paso es necesario verificar la información que brindan los valores de **F** y **D** actuales, para esto se utiliza un cuadro que contiene 3 filas.

| f_0 | f_1 | d_0 | d_1 | g_0 | g_1 | g_0 | g_1 | f_0 | f_1 | d_0 | d_1 |
|----------------|----------------|----------------|----------------|----------------|----------------|----------------|----------------|----------------|----------------|----------------|----------------|
| \mathbf{P}_1 | \mathbf{P}_2 | \mathbf{P}_1 | \mathbf{P}_2 | \mathbf{P}_1 | \mathbf{P}_2 | \mathbf{P}_2 | \mathbf{P}_1 | \mathbf{P}_1 | \mathbf{P}_2 | \mathbf{P}_1 | \mathbf{P}_2 |
| D | D | F | F | D | D | F | F | G | G | G | G |

Esquema de elección. Tomado de [2]

El significado de cada entrada de la tabla se describe a continuación:

Fila 1: Valor actual de la variable.

Fila 2: Matriz que se utilizará.

Fila 3: Parámetro para el cual se quiere definir la matriz.

¿Cómo se usa esta tabla? Si deseo conocer el valor del parámetro x entonces se ubica las columnas cuya tercera entrada sea el parámetro x y que el valor de la primera entrada coincida con el valor en la pantalla actual.

Por ejemplo, si se tiene la pantalla $(f_0 \ g_0 \ d_1)$ y se desea conocer la matriz con la que se decidirá el siguiente valor de f se ubican las columnas del esquema cuya 3ra entrada sea F y las primeras entradas sean g_0 y d_1 . La matriz en la segunda fila de dichas entradas serán las matrices buscadas. Si las matrices no coinciden, se elige una de forma aleatoria.

De modo que, cada forma de agrupar estos valores anteriores construye una pantalla con un valor establecido en la siguiente matriz de transición

| | A ($f_0 g_0 d_0$) | B ($f_0 g_0 d_1$) | C ($f_0 g_1 d_0$) | D ($f_0 g_1 d_1$) | E ($f_1 g_0 d_0$) | F ($f_1 g_0 d_1$) | G ($f_1 g_1 d_0$) | H ($f_1 g_1 d_1$) |
|---|------------------------|------------------------|------------------------|------------------------|------------------------|------------------------|------------------------|------------------------|
| A | 0.021 | 0.084 | 0.084 | 0.336 | 0.019 | 0.076 | 0.076 | 0.304 |
| B | 0.357 | 0.089 | 0.323 | 0.081 | 0.063 | 0.016 | 0.057 | 0.014 |
| C | 0.084 | 0.076 | 0.021 | 0.019 | 0.336 | 0.304 | 0.084 | 0.076 |
| D | 0.189 | 0.126 | 0.126 | 0.084 | 0.171 | 0.114 | 0.114 | 0.076 |
| E | 0.165 | 0.150 | 0.150 | 0.135 | 0.110 | 0.100 | 0.100 | 0.090 |
| F | 0.204 | 0.136 | 0.036 | 0.024 | 0.306 | 0.204 | 0.054 | 0.036 |
| G | 0.408 | 0.072 | 0.272 | 0.048 | 0.102 | 0.018 | 0.068 | 0.012 |
| H | 0.096 | 0.144 | 0.144 | 0.216 | 0.064 | 0.096 | 0.096 | 0.144 |

Matriz de transición de la cadena usando pantallas.. Tomado de [2]

Esta matriz al ser modelada por una cadena de markov converge relativamente rápido a su distribución estacionaria, un resultado que limitaría la variabilidad e innovación en las composiciones.

Para esto Xenaki propone perturbar este equilibrio de manera iterada, haciendo que cada vez que se llegue a este forzar una perturbación, enviando la cadena a una nueva pantalla, este proceso es descrito en el siguiente diagrama:

$$E \rightarrow P_A^0 \rightarrow P'_A \rightarrow E \rightarrow P'_C \rightarrow P_C^0 \rightarrow P_B^0 \rightarrow P'_B \rightarrow E$$

donde:

E : Distribución estacionaria

P_A^0 : El subíndice indica a qué distribución se asigna probabilidad 1.

$\overset{\circ}{P}_A$ Resultado de aplicar la matriz de transición a la perturbación.

¿Qué hace a este trabajo relevante?

A diferencia de **ILLIAC suite** la cual requería de reglas deterministas e intervención humana a la hora de rechazar resultados **Analogique A** fue la primera obra completamente automatizada utilizando únicamente procesos estocásticos.

4. Métodos actuales

4.1. Método tradicional.

Para generar música utilizando cadenas de markov es necesario que el conjunto de estados representen objetos sonoros, esto se puede lograr definiendo los estados como notas individuales, pares (nota, duración) o n-gramas que toman una secuencia de n-1 notas para definir el estado, no obstante, al trabajar con este último es necesario implementar una cadena de orden n-1.

A partir de un conjunto de entrenamiento se cuentan las apariciones de cada nota, así mismo la probabilidad de ir de una nota a partir de otra, diseñando una matriz de transición para cada parámetro sobre el cual que se quiere trabajar, esta matriz modela el comportamiento de la cadena, sin embargo, la música real no son solo notas elegidas de manera aleatoria, para esto es necesario controlar la estructura, esto se puede hacer por medio de reglas, temperaturas, perturbaciones como en Analogique A o penalizaciones, sin embargo, el desarrollo de esto depende completamente de lo que se quiera lograr y del estilo personal del responsable de la implementación.

4.2. Cadenas de Markov e interpolación de Lagrange.

Uno de los principales problemas al trabajar con cadenas de Markov consiste en la no consideración de la progresión de acordes y patrones rítmicos en la generación de notas, una posible solución a este problema consiste en utilizar las cadenas de Markov para generar progresiones de acordes (estructura) apoyándose en la interpolación de

Lagrange para generar melodías que siguen una curva suave orientada por estos acordes.

Pasos:

1. Se implementa la cadena de markov de manera tradicional tal que no genere melodías directamente sino una progresión armónica de acordes.
2. Para cada compás se toman puntos x, y de la forma (duración, acorde/nota) con los cuales se construye una curva suave utilizando interpolación de lagrange.
3. Se evalúa cada nota en la curva y se toma el valor más cercano a este punto, manteniendo una armonía global.

En la vida real existen notas que cruzan compases, estructuras rítmicas irregulares y variaciones métricas. Sin embargo, intentar incorporar estos elementos utilizando interpolación puede llegar a comprometer la uniformidad del sistema, perdiendo gran parte de la coherencia obtenida con su implementación.

5. Implementación del modelo y análisis de resultados

El objetivo de esta sección es contrastar los fundamentos teóricos con los resultados obtenidos al realizar una implementación práctica del método tradicional previamente descrito. Adicionalmente, se integran 1) Una caracterización de los estados de las cadenas de Markov inspirada en las pantallas de Xenaki y 2) La interpolación de Lagrange como mecanismo de regulación de las melodías generadas por la cadena.

Para esto, limitamos nuestro experimento a melodías usadas en consolas retro (8 bits). Dadas las limitaciones del [hardware](#) de la época, estas melodías cuentan con unas estructura mucho más limitada a las melodías de los videojuegos y canciones pop actuales. Lo cual es conveniente a la hora de estimar la distribución de probabilidad de las melodías.

Para nuestra implementación, consideramos los elementos esenciales para la construcción de una melodía:

- **Notas:** 12 notas por octava, 9 octavas disponibles en el estándar MIDI. En total, 108 notas distintas.
- **Duración:** sujeta al set de entrenamiento.
- **Acordes:** combinación de notas que se tocan en simultáneo. Sujeto al set de entrenamiento (la nota es un caso particular de acorde).

La implementación, analiza cada una de las notas de la secuencia MIDI para determinar las notas y acordes que están presentes en la canción con el fin de no construir estados *innecesarios* y ahorrar recursos. Así, cada estado de la cadena de

Markov es de la forma (*Acorde, Duración*). Para el análisis de las secuencias se tienen los siguientes parámetros:

- **Tolerancia de acorde (chord_threshold):** Este parámetro controla lo que se considera un acorde. Nace de la naturaleza del hardware de las consolas que no permiten directamente hacer acordes, por lo que se *emulan*. Así que al analizar cada secuencia, es necesario hacer el sampleo considerando estos problemas.
- **Orden:** Cuántas notas anteriores va a considerar para la estimación de la siguiente.

Para la generación de las secuencias se tienen el siguiente parámetro:

- **Temperatura:** Se implementa un parámetro de temperatura aplicado sobre la distribución de probabilidad, con el fin de modular el grado de exploración en la generación de notas y secuencias. Al igual que el recocido simulado, el parámetro de temperatura “destruye” la preferencia estimada por el procesamiento de los archivos y permite la exploración creativa de nuevas melodías.

5.1 Metodología

Este proyecto implementa dos métodos de generación melódica basados en cadenas de Markov de orden 2, aplicados a la generación de música retro de videojuegos 8-bit. Se comparan el método tradicional (Markov sobre notas) y el método de Xu et al. (2023) (Markov para acordes + interpolación de Lagrange para melodía).

Corpus Musical

Se seleccionaron tres canciones icónicas del género retro 8-bit, utilizando sus archivos MIDI:

`ZeldaFantasy_1_.mid`

`Bomberman2_-_Stage3.mid`

`3D_Worldrunner_Bonus.mid`

1. Extracción y procesamiento de datos MIDI

- Se extraen eventos musicales (notas, acordes, duraciones) de cada archivo MIDI
- Los eventos simultáneos se agrupan según un umbral temporal para identificar acordes

- Se organizan las secuencias por instrumento y offset temporal
- Detección automática de la tonalidad para mantener coherencia armónica

2. Construcción de Matriz de transición

- Método Tradicional: Análisis de frecuencias sobre secuencias de notas individuales con sus duraciones
- Método de Xu (Lagrange): Análisis separado de progresiones de acordes y construcción de melodías mediante interpolación de Lagrange estabilizada
- Construcción de matrices de transición de orden 2 (dependencia de dos estados previos)
- Almacenamiento de contextos iniciales para generación natural

3. Generar melodías

- Inicialización con secuencias reales del corpus original
- Recorrido probabilístico de la cadena de Markov
- Control de temperatura para ajustar aleatoriedad vs. determinismo
- Método Xu: Evita repeticiones agresivas y aplica restricciones de tonalidad

4. Resultados

Las melodías generadas se evalúan mediante múltiples métricas cuantitativas:

- Cross-Entropy: Mide la coherencia según el modelo probabilístico (objetivo: 3-8 bits)
- Perplexity: Cuantifica la diversidad efectiva de opciones por paso (objetivo: 5-20)
- JS Divergence: Evalúa la similitud distribucional con el corpus original (objetivo: < 0.15)
- Longest Copy: Detecta plagio o copia excesiva del material original (objetivo: < 15 notas)

Los archivos MIDI generados se exportan para análisis auditivo y comparación cualitativa.

5.2 Análisis de resultados

Los audios generados son

Para 3D_Worldrunner_Bonus:

- Generado con Método Tradicional:  WORLDRUNNER.wav
- Generado con Método Xu-Lagrange:  WORLDRUNNER LAGRANGE.wav

Para Bomberman2 - Stage3:

- Generado con Método Tradicional: BOMBERMAN.wav
- Generado con Método Xu-Lagrange: BOMBERMAN LAGRANGE.wav

Para sds:

- Generado con Método Tradicional: ZELDA.wav
- Generado con Método Xu-Lagrange: ZELDA LAGRANGE.wav

Los experimentos revelaron un trade-off fundamental entre coherencia y originalidad:

Método Tradicional (Markov sobre Notas)

- Cross-Entropy: 0.52 bits (valor extremadamente bajo)
- Perplexity: 1.43
- Longest Copy: 18 notas consecutivas
- Diagnóstico: Las melodías generadas presentan plagio significativo del material original. La cross-entropy muy baja indica que el modelo reproduce patrones del corpus casi textualmente, limitando la creatividad y originalidad de la composición.

Método de Xu (Lagrange)

- Cross-Entropy: 26.41 bits (valor excesivamente alto)
- Perplexity: 89,437,727
- JS Divergence: 0.2762 (por encima del umbral ideal de 0.15)
- Longest Copy: 5 notas
- Diagnóstico: La interpolación de Lagrange genera melodías altamente originales pero incoherentes. La perplejidad extremadamente alta revela una excesiva aleatoriedad en las transiciones, lo que resulta en secuencias poco naturales y musicalmente poco cohesivas. Si bien evita el plagio, sacrifica la coherencia estilística del género 8-bit.

Hallazgos Clave

- Detección de tonalidad: La implementación del método Xu incorpora detección automática de la tonalidad del MIDI original, lo cual es fundamental para mantener consistencia armónica en las melodías generadas.
- Rango ideal: Los valores óptimos se encuentran en un rango intermedio:
- Cross-Entropy: 3-8 bits (balance entre coherencia y variabilidad)
- Perplexity: 5-20 (diversidad controlada)
- Longest Copy: < 15 notas (originalidad sin perder estilo)

Limitaciones identificadas:

- El método tradicional tiende al sobreajuste (overfitting), memorizando el corpus
- El método Lagrange presenta subajuste (underfitting), generando transiciones demasiado impredecibles que rompen la estructura musical típica del género
- Implicaciones
- Se requiere un enfoque híbrido o ajuste de hiperparámetros (temperatura, orden de la cadena, estabilización de Lagrange) para alcanzar el balance ideal entre fidelidad estilística y creatividad melódica

6.3 Trabajo futuro

- Estudiar implementaciones de modelos híbridos que consideren diferentes géneros musicales y diversos instrumentos, con el fin de verificar la efectividad del modelo en cada uno de ellos.
- Implementar modelos híbridos que involucren la interacción humana en tiempo real (con el de continuación)
- Añadir más parámetros (volúmen, coherencia de patrones, etc)

6. Conclusiones

La investigación desarrollada confirma el valor de las cadenas de Markov como herramienta efectiva en la formalización algorítmica de la composición musical, demostrando capacidad robusta para modelar patrones locales, relaciones entre notas consecutivas y frecuencias rítmicas a partir de un corpus definido —en nuestro caso, archivos MIDI de música retro 8-bit—. La implementación exitosa de cadenas de Markov de orden 2, tanto en su formulación tradicional como en el método híbrido de Xu (2023), generó secuencias que capturan auténticamente las características estilísticas del género analizado. La evaluación cuantitativa reveló un espectro de posibilidades compositivas: el método tradicional logra alta fidelidad estilística (CE: 0.52 bits), replicando exitosamente la esencia del corpus; mientras que el método de Xu alcanza mayor originalidad creativa (longest copy: 5 notas), explorando espacios melódicos más diversos (CE: 26.41 bits, perplexity: 89M).

Estos hallazgos establecen las bases metodológicas para sistemas de generación musical más sofisticados, identificando claramente los parámetros que controlan el equilibrio entre fidelidad estilística y creatividad compositiva. Las métricas implementadas —cross-entropy, perplexity, JS divergence y longest copy— proveen un marco de evaluación objetivo que permite caracterizar y optimizar el comportamiento de futuros modelos generativos. La investigación demuestra que las

cadenas de Markov constituyen un fundamento teórico y práctico sólido para la generación algorítmica de música, cuyo potencial puede expandirse mediante su integración con restricciones armónicas, modelos jerárquicos o arquitecturas híbridas, abriendo caminos prometedores hacia sistemas de composición asistida algorítmicamente que combinen estructura probabilística con control creativo intencional.

7. BIBLIOGRAFÍA

1. Nierhaus, G. (s.f.). *Algorithmic composition: Paradigms of automated music generation*. Springer Wien New York.
2. Tremonte de Carvalho, H. (s.f.). *An introduction to Markov chains in music composition and analysis*. Federal University of Rio de Janeiro (UFRJ).
3. Shapiro, I., & Huber, M. (2021). *Markov chains for computer music generation*. *Journal of Humanistic Mathematics*, 11(2).
4. Xu, Y. (2023). *Music generator applying Markov chain and Lagrange interpolation*. In *Highlights in Science, Engineering and Technology: CMLAI 2023* (Vol. 39, p. 266).
5. Maurer, J. A. (1999). *A brief history of algorithmic composition*.
6. Pachet, F. (2010). *The Continuator: Musical interaction with style*. *Journal of New Music Research*, 32(3), 333–341.
7. Li, T., Choi, M., Fu, K., & Lin, L. (s.f.). *Music sequence prediction with mixture hidden Markov models*.
8. Hill, S. (s.f.). *Markov melody generator*