



# Propuesta Inicial

Matias Andrade – Yoav Navon

---

## 1. Contexto del Problema

Métodos tradicionales para generar recomendación son el filtrado colaborativo (CF) y el filtrado basado en contenido (CB). Los métodos basados en alguno de estos dos paradigmas se aprovechan de lo mismo, en base a las interacciones pasadas del usuario generar recomendaciones, ya sea buscando ítems similares (CB), o utilizando la información de usuarios similares (CF). Sin embargo, dichos métodos ignoran los patrones transaccionales de corto plazo [1], considerando todo el historial del usuario como una unidad. De esta manera, el comportamiento de un usuario en un momento pasado puede influenciar negativamente la recomendación de un usuario que presenta patrones de comportamiento de corto plazo. Esta área de investigación se le conoce como Session Based Recommendation (SBRs), de manera que lo que se busca es dada una sesión de interacciones en un periodo corto de tiempo, predecir los ítems más valiosos para dicha sesión.

## 2. Problema y Justificación

Ya enmarcado en el contexto de SBRs, vamos a centrar nuestra investigación en la predicción top-N de canciones, dada una sesión de reproducción de múltiples canciones. Este problema es pertinente en la práctica, ya que dependiendo de distintos contextos (estudio, deporte, amigos, etc) vamos a querer distintas recomendaciones. Para nuestro problema, cada sesión es anónima, por lo que no sabemos que usuario efectuó dicha secuencia de tracks. Modelos como [2] y [3] no asumen independencia entre sesiones, por lo que rescatan información al corto y largo plazo, sin embargo para efecto de la investigación nos centraremos en el corto plazo. Esta es una manera de enfrentar el problema del cold-start, ya que con unas cuantas interacciones debemos ser capaces de recomendar.

## 3. Objetivos

- Evaluar el modelo en la tarea de rankear canciones para un dataset de grandes dimensiones como lo es el Music Streaming Sessions Dataset.
- Evaluar el impacto del uso de inputs content-based para el modelo, en contraste con ids como input.
- Contrastar el modelo basado en RNN con baselines más comunes como Item-KNN, Most Popular.

## 4. Solución Propuesta

### 4.1. Modelo General

El modelo propuesto está basado en [4], explicado en la sección 6. La propuesta de modelo consiste en una RNN basada en una celda GRU. A la red se le entregará un embedding de cada canción de forma

secuencial, de manera que primero se le entrega la primera canción de la sesión, y se busca que genere un ranking de posibles canciones, donde el primer elemento sería la predicción, y así sucesivamente. Finalmente, obtendremos una predicción de la canción que seguirá a la sesión. El diagrama se muestra en la figura 1, donde se le pasan las N canciones de cada sesión.

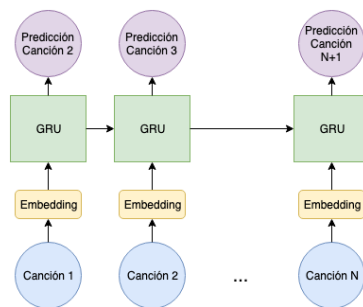


Figura 1: Diagrama de bloques RNN

## 4.2. Embeddings

Para el modelo propuesto es necesario computar embeddings para cada canción. El dataset a utilizar entrega 28 features para cada canción o track, por lo que la opción más simple de embedding a utilizar es entregarle como input a la RNN este vector de 28 features. Otra opción que se evaluará será el pasar los 28 features por una MLP antes de la RNN, de manera de generar un vector de más dimensiones. De esta manera, el modelo podrá determinar los features más relevantes para la tarea de recomendación, ya que la MLP será entrenado en conjunto con la RNN, manteniendo el carácter end-to-end.

## 4.3. Función de Pérdida

El método más simple para entrenar el modelo es utilizando cross-entropy, de manera de transformar la recomendación en un problema de clasificación entre las miles de tracks existentes. Otro método a evaluar será computar una pairwise loss, de manera de optimizar en la tarea de rankear, para realizar esto es posible utilizar BPR.

## 4.4. Entrenamiento

Para el entrenamiento del modelo, se va a dividir el dataset en training y testing. De esta manera, para cada sesión de training, se utilizarán todas las canciones de la sesión para optimizar el modelo. Luego de haber hecho esto, para cada sesión de testing le vamos a pasar las canciones de manera secuencial, sin pasarle las últimas 10 canciones (sin realizar back-propagation para testing), computando una métrica de evaluación.

# 5. Experimentos a Realizar

## 5.1. Dataset

Vamos a utilizar el Music Streaming Sessions Dataset [5], creado por Spotify. El dataset consiste en alrededor de 150 millones de sesiones con largo entre 10 y 20 tracks. Aparecen alrededor de 3.7 millones de tracks diferentes, por lo que el realizar recomendación es evidentemente un desafío. Además, para cada canción se cuenta con metadata particular de la canción, y metadata de la canción dentro de la sesión particular (e.g. si el usuario la terminó de escuchar, hizo skip, etc).

## 5.2. Métodos

- Se evaluará el uso de GRU y LSTM para el modelo.
- Se evaluará el impacto de estrategias de entrenamiento. Por ejemplo la performance vs. el número de canciones que ha visto la red de la sesión actual.
- Se probarán estrategias para generar los embeddings, ya sea utilizar los features del dataset directamente, o pasarlos primero por una MLP para generar un vector denso.
- Se evaluarán distintas formas de calcular la pérdida. Se experimentará con el método de cross-entropy, y métodos de pairwise loss como BPR y TOP1 (propuesto en [4]).

## 5.3. Evaluación

El modelo va a entregar scores para cada item, por lo que vamos generar un ranking de estos, esperando que el item que tenemos como ground-truth quede arriba en el ranking. Como el dataset cuenta con millones de items, es impráctico rankear todos los items, por lo que de la misma manera que en [4] se tomarán los 30.000 items más populares y el item deseado, de manera de rankear este subconjunto. Finalmente calcularemos recall@N, denotando la proporción de casos en que el item deseado quede en los primeros N items del ranking. También se evaluará con Mean Reciprocal Rank (MRR@N) para considerar el orden en que aparece el item deseado, donde si el item no aparece en las primeras N entradas del ranking se considera Reciprocal Rank igual a 0.

# 6. Bibliografía Relevante

## 6.1. Session-Based Recommendations With Recurrent Neural Networks [4]

Este es el principal modelo en que se basó la propuesta. A este modelo se le entregan como inputs codificaciones one-hot de los items a recomendar. Luego la arquitectura es una RNN basada en celdas GRU. El modelo es evaluado con datos de e-commerce correspondientes al RecSys Challenge 2015, y datos de recomendaciones de una plataforma de videos que los autores llaman VIDEO. La cantidad de items para cada dataset son del orden de 37.000 y 330.000 respectivamente, por lo que adaptar el modelo para el dataset de spotify con 3.7 millones es otro desafío de la propuesta. El modelo está disponible en [Github](#).

## 6.2. Session-based Sequential Skip Prediction via Recurrent Neural Networks [6]

Este modelo fue el ganador del *Spotify Sequential Skip Prediction Challenge*, utilizando el dataset de sesiones de Spotify que se propone utilizar. Además el modelo está basado en una RNN, por lo que demuestra que es posible utilizar redes recurrentes para recomendacion basada en sesión para este dataset. Una diferencia importante con nuestra propuesta es que dicho modelo es entrenado para *skip prediction*, es decir, si dentro de la sesion el usuario va a seleccionar *skip* o escuchará la canción.

## Referencias

- [1] Shoujin Wang, Longbing Cao y Yan Wang. “A Survey on Session-based Recommender Systems”. En: *CoRR* abs/1902.04864 (2019). arXiv: [1902.04864](https://arxiv.org/abs/1902.04864). URL: <http://arxiv.org/abs/1902.04864>.
- [2] Massimo Quadrana, Alexandros Karatzoglou, Balázs Hidasi y col. “Personalizing session-based recommendations with hierarchical recurrent neural networks”. En: *Proceedings of the Eleventh ACM Conference on Recommender Systems*. ACM. 2017, págs. 130-137.
- [3] Mei Wang, Weizhi Li y Yan Yan. *Time-weighted Attentional Session-Aware Recommender System*. 2019. arXiv: [1909.05414](https://arxiv.org/abs/1909.05414) [[cs.LG](#)].
- [4] Balázs Hidasi, Alexandros Karatzoglou, Linas Baltrunas y col. “Session-based recommendations with recurrent neural networks”. En: *arXiv preprint arXiv:1511.06939* (2015).
- [5] Brian Brost, Rishabh Mehrotra y Tristan Jehan. “The Music Streaming Sessions Dataset”. En: *The World Wide Web Conference*. ACM. 2019, págs. 2594-2600.
- [6] Lin Zhu y Yihong Chen. “Session-based Sequential Skip Prediction via Recurrent Neural Networks”. En: *arXiv preprint arXiv:1902.04743* (2019).