PONTIFICIA UNIVERSIDAD CATÓLICA DE CHILE
ESCUELA DE INGENIERÍA
DEPARTAMENTO DE CIENCIA DE LA COMPUTACIÓN

IIC3633 – Sistemas Recomendadores

Entrega Avance

Matias Andrade – Yoav Navon

1. Análisis Exploratorio

Vamos a distinguir entre 3 datasets distintos. En primer lugar está el llamado Dataset Full, que consta con el 100 % de los datos. Después está el Mini Dataset, que es un subconjunto del full, y es entregado por Spotify en conjunto con el Dataset full para familiarizarse con los datos. Por último, está el que llamaremos Dataset03, que consiste en 1/330 del dataset full, o 0.3 % del original. Para el Dataset Mini utilizamos un split 80-20 para training-testing, en que ninguna sesión quedó separada, y se corroboró que los tracks de testing aparescan en training. Para Dataset03 se utilizó un split 50-50, nuevamente corroborando que los items aparescan anteriormente. Todos los resultados de aquí en adelante son en referencia al respectivo set de testing.

	Eventos	**Tracks**	**Sesiones**
Mini Dataset	167,880	50,704	10,000
Dataset Full	2,250,000,000	3,700,000	150,000,000
Dataset03	6,066,614	505,195	365,448

Tabla 1: Estádisticas del Dataset

Para todo los datasets, el mínimo número de eventos en una sesión es de 10, y el máximo es de 20. Se verificó que el dataset estuviera ordenado, de manera que todos los eventos de una misma sesión vinieran de manera contigua, y que estuvieran ordenados por su posición dentro de la sesión.

Column name	Column description	Example value
session id	unique session identifier	65_283174c5-551c-4c1b-954b-cb60ffcc2aec
session position	position of row within session	4
session length	number of rows in session	11
track id	unique track identifier	t_13d34e4b-dc9b-4535-963d-419afa8332ec
context id	unique context identifier	124321515
skip_1	boolean indicating if the track was only played very briefly	1
skip_2	boolean indicating if the track was only played briefly	1
skip_3	boolean indicating if most of the track was played	0
not_skipped	boolean indicating that the track was played in its entirety	0
context switch	boolean indicating if the user changed context between rows	0
no_pause_before_play	boolean indicating if there was no pause between current and previous playback	1
short_pause_before_play	boolean indicating if there was a short pause between current and previous playback	1
long_pause_before_play	boolean indicating if there was a long pause between current and previous playback	0
n_seekfwd	number of times the user did a seek forward within track	2
n_seekback	number of times the user did a seek back within track	0
shuffle	boolean indicating if the user encountered this track while shuffle mode was activated	0
hour_of_day	hour of day between 0 and 23	22
date	date in YYYY-MM-DD format	2018-09-18
premium	boolean indicating if the user was on premium or not	1
context_type	what type of context the playback occurred within	editorial playlist
reason_start	the user action which led to the current track being played	fwdbtn
reason_end	the user action which led to the current track playback ending	trackdone

Figura 1: Columnas de cada entrada del dataset

En la figura 2 se muestra la distribución de tracks por eventos en el Dataset03. Podemos notar que más de 400,000 canciones aparecen menos de 10 veces, por lo que trabajamos con datos altamente *sparse*.

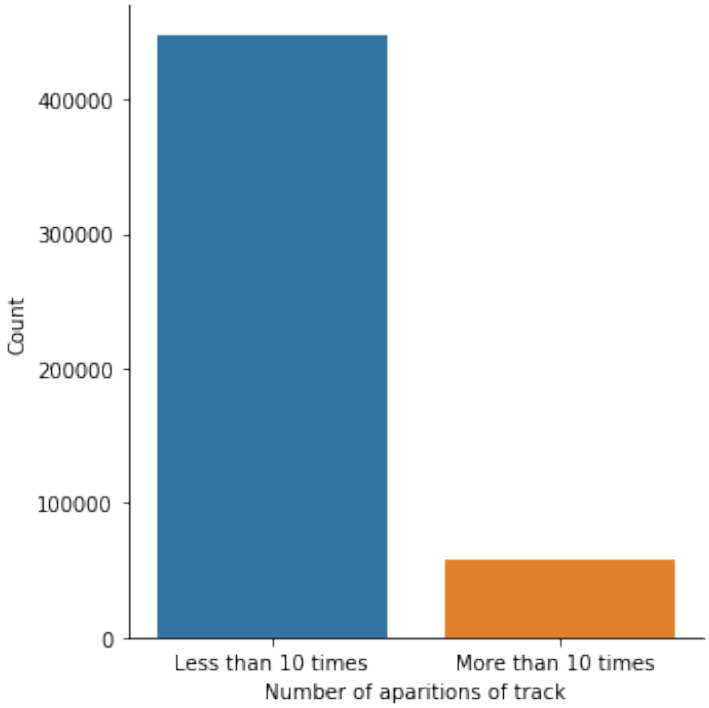

Figura 2: Distribución de tracks en eventos

2. Estado de Avance

Debido al gran tamaño del Dataset Full, solo se trabajó con el Dataset Mini y Dataset03. Cabe notar que el modelo utilizado demora 5 horas en 1 época en el Dataset03, por lo que con los recursos que se cuenta (Google Colab), utilizar el Dataset Full se vuelve intratable.

Se adaptó el modelo basado en RNN de pcerdam para que funcione para nuestro dataset de spotify. Además se le agregó al modelo la capacidad de utilizar como función de pérdida BPR. Por último se implementó el conocido baseline basado en Most Popular Items para recomendación no personalizada.

	GRU4REC (cross-entropy)		GRU4REC (bpr)		Most Popular	
	Recall@20	MRR@20	Recall@20	MRR@20	Recall@20	MRR@20
Mini Dataset	0.226	0.056	0.11	0.024	0.10	0.025
Dataset03	0.24	0.024	-	-	-	-

Tabla 2: Resultados del modelo según función de pérdida

Cabe mencionar que el modelo entrenado con BPR está obteniendo resultados equivalentes con el baseline de Most Popular, por lo que creemos que puede haber un error, ya que al ser BPR tan utilizado en la literatura esperamos que por lo menos sea capaz de superar el baseline.

2.1. Representaciones de Inputs

Para todos los modelos mencionados en la tabla 2, el input que se le entregó al modelo fueron vectores one-hot correspondientes al track. En esta sección exploramos como varían los resultados al utilizar otras representaciones.

La segunda representación utilizada (siendo la primera one-hot), es utilizar embeddings para cada track. Es decir, dado el id de la canción, el modelo hace lookup al embedding correspondiente y este pasa a ser el input del modelo. Para implementar esto solo fue necesario agregar la capa de Embedding luego del input, y esta es entrenada junto al modelo de manera end-to-end. Se probaron dos versiones, GRU4REC-emb50 y GRU4REC-emb100, que corresponden a la utilización de embeddings de tamaño 50 y 100 respectivamente.

La última representación que se utilizó, es la basada en contenido, ya que se utilizó la información que el dataset provee para cada track. Esta información corresponde a 29 features por canción, los que aparecen en la tabla 3. Estos fueron normalizados de manera que todos tengan promedio 0 y desviación 1. El modelo que utiliza esta representación corresponde a GRU4REC-content. Los resultados de todos los modelos se encuentran en la tabla 4.

duration	release year	popularity	acousticness	beat strength	bounciness
danceability	dyn range mean	energy	flatness	instrumentalness	key
liveness	loudness	mechanism	mode	organism	speechiness
tempo	time signature	valence	ac vector0	ac vector1	ac vector2
ac vector3	ac vector4	ac vector5	ac vector6	ac vector7	

Tabla 3: Features disponibles para cada track

	Mini Dataset		Dataset03	
	Recall@20	MRR@20	Recall@20	MRR@20
GRU4REC-onehot	0.226	0.056	0.24	0.024
GRU4REC-emb50	0.201	0.051	-	-
GRU4REC-emb100	0.209	0.054	-	-
GRU4REC-content	0.170	0.041	-	-

Tabla 4: Resultados de GRU4REC variando la representación de las tracks. Para todos los casos se utilizó la versión del modelo utilizando cross-entropy

3. Problemas

1. La dificultad más grande que nos hemos encontrado es la dimensión del dataset. Aunque estamos utilizando un subsample del original, aún son muchos datos, y los tiempos de entrenamiento no son menores.
2. BPR no ha funcionado como esperabamos, es posible que la implementación tenga algún error.
3. No tenemos información sobre la fecha de cada sesión de canciones, por lo que hacer un split training-test que tenga sentido es más difícil, ya que es posible que estemos testeando con sesiones con mucha distancia temporal, y sabemos que las tendencias en música son muy variables en el tiempo.

4. Plan de Avance

1. No hemos jugado demasiado con los parámetros del modelo (learning rate, optmizador, hidden units, batch size). Como el modelo original estaba optimizado para vectores one-hot, puede ser esta la razón que esta representación supere a las otras (embeddings, content-based). Proponemos realizar mayor fine-tuning de parámetros.
2. Para la versión content-based, estamos dando el mismo peso a todos los features. Proponemos pasar el vector de cada track por un MLP primero para que optimice la representación de cada track.
3. En el paper [1], utilizan una combinación de embeddings con vectores de features provistos por el dataset, por lo que proponemos experimentar esta opción.
4. Creemos que BPR no está funcionando como debe, por lo que habría que revisarlo. Además se propone implementar la función de perdida TOP1 mencionada en el paper original de GRU4REC [2].
5. Vamos a rellenar las celdas que quedaron vacías en las tablas de resultados.

Referencias

[1] Christian Hansen, Casper Hansen, Stephen Alstrup y col. "Modelling Sequential Music Track Skips using a Multi-RNN Approach". En: *arXiv preprint arXiv:1903.08408* (2019).

[2] Balázs Hidasi, Alexandros Karatzoglou, Linas Baltrunas y col. "Session-based recommendations with recurrent neural networks". En: *arXiv preprint arXiv:1511.06939* (2015).