Evaluación Reproducible de Sistemas de Recomendación Secuenciales

Matías Ossul Corbalán, Sebastián Terrazas Caviglia

Departamento de Ciencia de la Computación, Pontificia Universidad Católica de Chile

Objetivos

- Investigar el impacto de la reproducibilidad en modelos secuenciales.
- Analizar sensibilidad de GRU4Rec ante correcciones e hiperparámetros.
- Comparar GRU4Rec con baselines clásicos: Random, Popularity, Item-KNN.
- Evaluar métricas de ranking, diversidad y novedad.
- Mostrar ejemplos cualitativos de éxito y fallo.

Introducción

Los sistemas secuenciales predicen la siguiente interacción de usuario. Modelos de RNN (GRU4Rec) han demostrado gran promesa, pero su reproducibilidad y comparación con baselines simples no siempre está clara. Migramos a Yoochoose, aplicamos fixes y reportamos métricas de ranking, diversidad y sensibilidad.

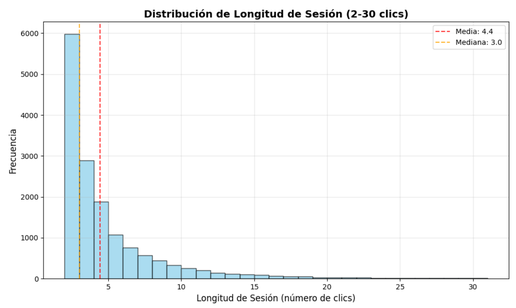


Figure 1:Distribución de longitud de sesión (2–30 clicks).

Definición de Métricas

Recall@K: $Recall@K(u) = \frac{|\text{relevantes} \cap \text{topK}|}{|\text{relevantes}|}$

MRR@K: $MRR@K(u) = \frac{1}{rank_u}$

ILD: $ILD(L) = \frac{2}{|L|(|L|-1)} \sum_{i<j} (1 - sim(i,j))$

Popular Bias: proporción de ítems top-populares en topK.

Datos y Recursos

- Yoochoose (subset 100 000 interacciones)
- `gru4rec_third_party_comparison`
- PyTorch, scikit-learn
- Preprocesamiento: `create_mini_dataset.py`

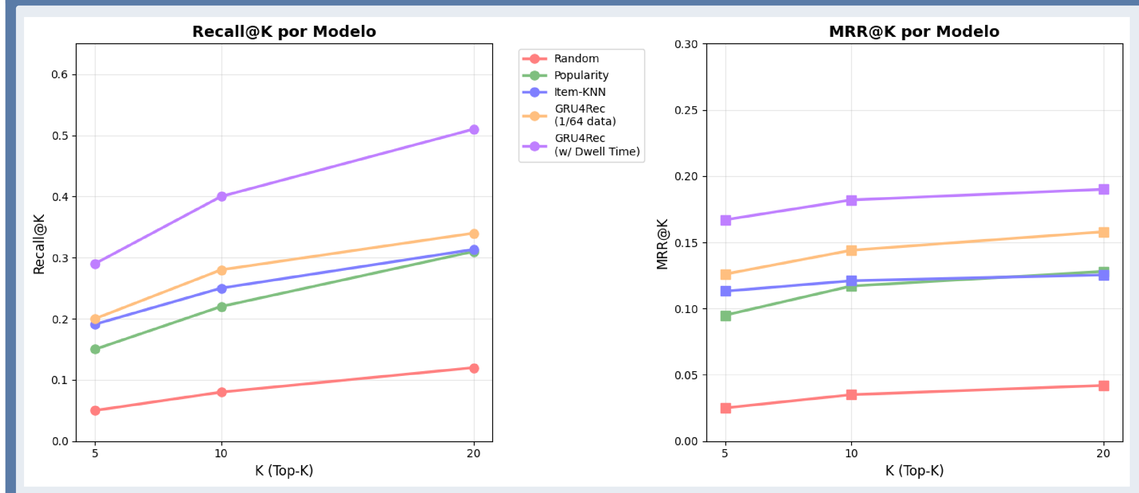
Desafíos:

- Compatibilidad Linux/Windows
- Limitación de memoria
- Tiempos de entrenamiento

Metodología

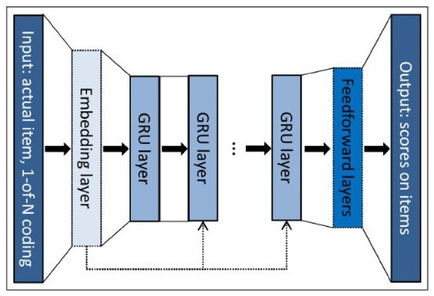
- Baselines: Random, Popularity, Item-KNN.
- GRU4Rec en 2 variantes: base, Major-Fix (w/ Dwell Time).
- Split temporal (80/10/10) y seed=42.
- Evaluación: Recall@K, MRR@K, ILD, Popularity Bias.

Resultado Clave



Recall@K y MRR@K para K=5,10,20. GRU4Rec Major-Fix mejora sobre base y Random, compite con Item-KNN para K bajos.

¿Cómo funciona GRU4Rec?



Comparación con el paper original

Hidasi et al. (2015) reportan en Yoochoose:

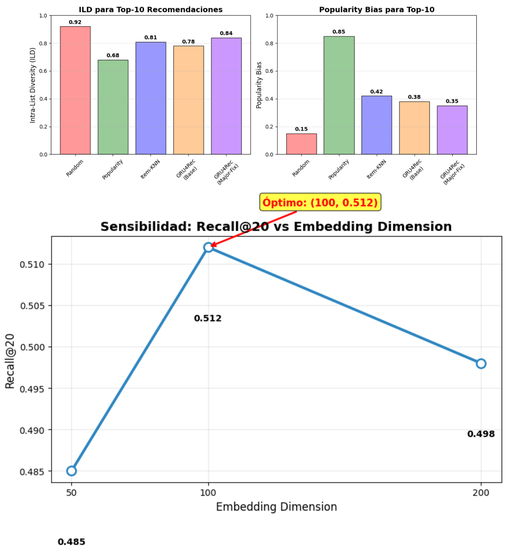
- GRU4Rec original: Recall@20 0.506
- Item-KNN: Recall@20 0.268

En nuestro experimento reproducible:

- GRU4Rec (w/ Dwell Time): **0.512** en Recall@20.
- Item-KNN logra **0.312** en Recall@20.

Confirmamos que GRU4Rec gana pero la brecha es menor, subrayando la necesidad de reproducibilidad.

Diversidad & Sensibilidad



(arriba) ILD y Popularity Bias para top-10. (abajo) Recall@20 vs. embedding_dim. para GRU4Rec (w/ Dwell Time)

Conclusiones

- GRU4Rec Major-Fix mejora Recall@20 (12→51%) vs base.
- Item-KNN alcanza 31.3% en R@20, mostrando robustez clásica.
- GRU4Rec ofrece mayor diversidad (ILD) y menor sesgo de popularidad.
- Sensibilidad óptima para embedding_dim 100.
- Casos cualitativos validan fortalezas y limitaciones según longitud de sesión.

Referencias

- Antonio Purificato et al. (2024). *A reproducible analysis of sequential recommender systems.* arXiv:2408.03964.
- Steffen Rendle, Christoph Freudenthaler, Zeno Gantner, Lars Schmidt-Thieme (2010). *Factorizing personalized Markov chains for next-basket recommendation.* WWW.
- Balázs Hidasi et al. (2016). *Session-based recommendations with recurrent neural networks.* ICLR.
- Wang-Cheng Kang, Julian McAuley (2018). *Self-attentive sequential recommendation.* ICDM.
- M. Ferrari Dacrema, P. Cremonesi, D. Jannach (2019). *Are we really making much progress?* RecSys.
- Fei Yuan et al. (2019). *A simple convolutional generative network for next item*