
Clase 221 — Cold-start problem

Parte: 6 — Sistemas de Recomendación · Fuente: Schein, Popescul, Ungar, Pennock, Methods and Metrics for Cold-Start Recommendations (SIGIR 2002) + Aggarwal cap. 13.

Duración estimada: 70 min.

Clase 221 — Cold-start problem

Parte: 6 — Sistemas de Recomendación · Fuente: Schein, Popescul, Ungar, Pennock, *Methods and Metrics for Cold-Start Recommendations (SIGIR 2002)* + Aggarwal cap. 13. Duración estimada: 70 min.

Objetivo

Cuándo un user o item es "nuevo" (0 interacciones), CF (Clase 216-217) no funciona. Estrategias concretas para los 3 tipos de cold-start: user cold-start (onboarding), item cold-start (catalog launch), system cold-start (lanzamiento del producto). Las estrategias correctas son la diferencia entre un recomendador útil desde el día 1 vs uno inútil hasta el mes 6.

Resultados de aprendizaje

Al finalizar, el estudiante podrá:

- Diferenciar user cold-start (usuario nuevo), item cold-start (item nuevo), system cold-start (todo nuevo).
- Aplicar popularity fallback con shrinkage Bayesiano $((c + m \times C) / (n + m))$ para users sin historia.
- Diseñar onboarding explícito (pedir N preferencias antes de personalizar).
- Usar content features (Clase 218) para items nuevos.
- Aplicar exploration-exploitation con bandits (epsilon-greedy, Thompson sampling) para users mid-cold-start.

Temas

#	Tema	Por qué importa
1	3 tipos de cold-start	Cada uno necesita estrategia distinta.
2	Popularity fallback	Default sano cuando no sabés nada.
3	Bayesian shrinkage para popularity	Evita que items con 2 ratings 5/5 dominen.
4	Onboarding: preguntar al user	"Elegí 3 géneros" cambia el juego.
5	Content-based para item cold-start	Embeddings de texto/imágenes.
6	Bandits para explorar	Cuando offline metrics no alcanzan.

Definiciones y características

- User cold-start: usuario sin historial (recién registrado). El modelo no tiene p_u . Estrategias: popularity, onboarding, demographics.
- Item cold-start: item nuevo sin ratings. El modelo no tiene q_i . Estrategias: content features (descripción, categoría), boost temporal "nuevos".
- System cold-start: lanzamiento del producto, ni users ni items con historia. Estrategias: editorial picks, externos (importar gustos desde Facebook/Google), popularity por geo/demographic.
- Popularity shrinkage Bayesiano: $score = (\sum ratings + m \times C) / (n + m)$ donde C es la media global, m es el "weight" del prior. Items con $n=2$ ratings 5/5 no superan items con $n=1000$ ratings 4.2/5.
- Onboarding explícito: pedir al user N preferencias (géneros, intereses) antes de personalizar. Estilo Spotify/Netflix.

- Bandits: framework de RL para exploration vs exploitation. Epsilon-greedy: con prob ϵ explora random, con prob $1-\epsilon$ recomienda el "best". Thompson sampling: muestra de la posterior de cada arm.
- Contextual bandits: el contexto (user features) influye en la decisión. Más sofisticado que vanilla bandits.

Dataset / recursos

- Dataset: MovieLens 100K + new users / new items simulados.
- Librerías: numpy, opcional vowpalwabbit (bandits), scikit-learn.

Ejercicios

1. Popularity baseline: rankear items por $n_ratings$. Top-10 son siempre los mismos. Evaluar $recall@10$ para users cold-start (con 0 interactions).
2. Bayesian shrinkage: implementar $(sum + m \times C) / (n + m)$ con $m=10$, $C=mean_rating$. Comparar top-10 con popularity vanilla. Items con pocos ratings caen al promedio.
3. Onboarding 3 géneros: simular que user nuevo elige ["Action", "Sci-Fi", "Comedy"]. Recomendar top-10 movies de esos géneros (content-based + popularity tiebreaker).
4. Item cold-start: agregar 10 movies nuevas con descripción pero 0 ratings. Content-based (Clase 218) las puede rankear; CF no. Demostrar.
5. Epsilon-greedy: en cada slot del top-10, con prob $\epsilon=0.1$ recomendar item random (explore), con prob 0.9 recomendar el "best" del modelo (exploit). Medir coverage en N usuarios.

Homework verificable

Repo con:

1. Recomendador full con manejo explícito de cold-start:
 - User nuevo: onboarding (elegir géneros) → content-based.
 - User con ≤ 5 interactions: weighted hybrid con α bajo (más content).
 - User maduro: CF puro.
1. Item nuevo: content-based hasta acumular 10 ratings, después ALS.
2. Bayesian shrinkage en todos los rankings "popularity-based".
3. Bandit epsilon-greedy en producción (simulado): tasa $\epsilon=0.05$, decay a 0.01 después de 30 días.
4. Evaluación por segmento (cold-start, warm) con $NDCG@10$. Mostrar que estrategia adaptativa supera a CF puro para cold.

Criterio de aceptación: cold-start users tienen $NDCG@10 > 0.05$ (vs ~ 0 con CF puro); items nuevos aparecen en recomendaciones dentro de las primeras 48h.

Errores comunes

Síntoma / mensaje	Causa y cómo arreglar
Popularity baseline domina todo	Tu CF no está aprovechando — probable bug
Items nuevos nunca son recomendados ("rece	Pure content sin boost por novelty. Fix: a
Onboarding pide 20 géneros — los users se	UX trade-off. Fix: pedir 3-5 max; mejor 1
ϵ -greedy estanca: nunca aprende	ϵ muy bajo desde día 1. Fix: arrancar con
Bayesian shrinkage me da scores casi igual	m muy alto. Fix: tunear m por validación —

Geographic popularity sesga a country mayo

Sin segmentación. Fix: popularity por (cou

Preguntas frecuentes

¿Onboarding pregunta directa o implícita?

- Directa (elegir 3 géneros): rápida, controlable, pero rompe el flujo. Spotify la usa al sign-up.
- Implícita (observar primeras N interacciones, no recomendar hasta tener señal): seamless, pero los primeros 5 clicks son perdidos.

Híbrido: pregunta directa básica (1-2 items) + observar primeras N.

¿Mostrar items "Nuevo" boost por cuánto tiempo?

Depende del catálogo. News: minutos-horas. Movies: días-semanas. Productos e-commerce: días. Decay logarítmico: $\text{boost} = 1 / (1 + \text{days})$.

¿Bandits valen la pena vs A/B test?

A/B test mide impacto de una variante vs otra. Bandits explotan automáticamente la mejor opción mientras siguen explorando. Para múltiples variantes (10+ tipos de recomendación) o cuando el costo de no explorar es alto: bandits. Para 2-3 variantes y experimentación controlada: A/B (Clase 204).

¿Cold-start de sistema cómo arranco?

(1) Importar gustos desde otra plataforma (Facebook Connect, Google sign-in). (2) Editorial picks curados. (3) Promociones para incentivar primer rating. (4) Partner con plataforma que ya tiene data.

¿Cuándo termina el cold-start?

Heurística: cuando $\text{NDCG}@10$ personal supera consistente a $\text{NDCG}@10$ de popularity baseline. Típicamente: 5-20 interactions. Depende del catálogo.

¿Hay solo CF + content para resolverlo?

No. Cross-domain transfer (gustos en Spotify → recomendaciones en Audible), demographic CF (gente como vos suele consumir X), conversational ("¿qué te interesa hoy?") son alternativas avanzadas.

Referencias

- Schein, A. et al. Methods and Metrics for Cold-Start Recommendations (SIGIR 2002).
- Aggarwal Recommender Systems: The Textbook (Springer, 2016), cap. 13 — Advanced Topics (incluye cold-start).
- Sutton & Barto, Reinforcement Learning: An Introduction (2nd ed., 2018), cap. 2 — Multi-armed bandits.
- Vowpal Wabbit contextual bandits.
- Two Decades of Recommender Systems at Amazon (Smith & Linden, IEEE, 2017) — perspectiva histórica.

Siguiente clase

Clase 222 — Librerías: LightFM, Implicit, Surprise

Apéndice: notebook (primer bloque)

Estrategias para users/items sin historia. Demos sintéticos.

```
import numpy as np, pandas as pd

# Items con n_ratings y mean_rating distintos
items = pd.DataFrame({
    'item': range(1, 11),
    'n': [2, 5, 10, 50, 100, 500, 1000, 1500, 3, 1],
    'mean': [5.0, 4.8, 4.5, 4.4, 4.3, 4.2, 4.1, 4.0, 5.0, 5.0],
})
items['naive_score'] = items['mean']

# Popularity vanilla: ordenar por mean → ganan items con 1-2 ratings 5/5 (overconfident)
print('--- Popularity vanilla (por mean) ---')
print(items.sort_values('naive_score', ascending=False).head(5).to_string(index=False))
```

Archivos complementarios

- notebook.ipynb