

# Sistemas Recomendadores

## IIC-3633

Sistemas Recomendadores secuenciales y grupales

# Esta clase

1. Sistemas recomendadores secuenciales
2. Sistemas recomendadores grupales

# Tips para presentación de papers

# Estructura (1/2)

**Motivación:** ¿Para qué se hizo el paper? ¿Por qué es importante? ¿Qué se obtuvo que lo hace distinto?

**Trabajo relacionado:** ¿Que se ha hecho antes? ¿Por qué el paper se diferencia de lo que ya se ha hecho antes?

**Datos:** ¿Qué datos se utilizan? En qué dominio? Por que el método propuesto se puede evaluar mejor con estos datos?

## Estructura (2/2)

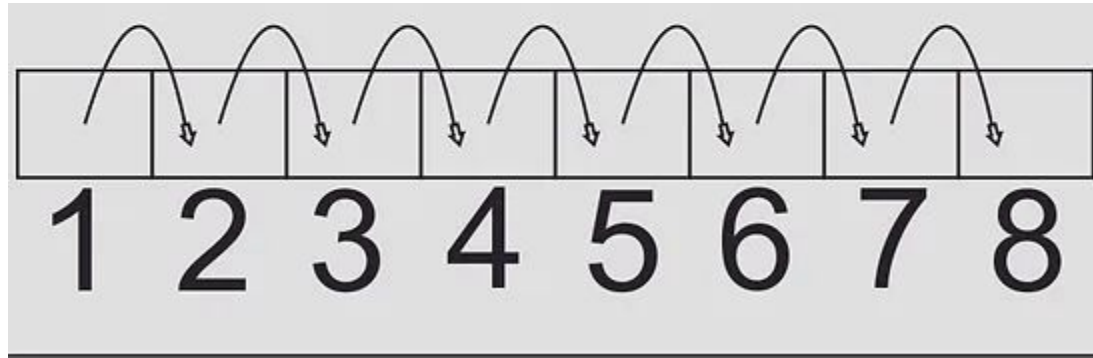
**Métodos:** ¿Qué método proponen (describir cual es la intuición, diagramas , arquitecturas, etc..) ? ¿Con qué métodos se comparan?

**Resultados:** Principales resultados obtenidos (metricas), comparación de métodos, análisis de sensibilidad, cómo se comparan en los distintos conjuntos de datos.

**Conclusiones y discusión:** Que obtuvieron, que se puede sacar en limpio del paper, que falta por mejorar, que cosas curiosas se encontraron?

# Sistemas recomendadores secuenciales

¿Por qué es importante la secuencialidad?



# Importancia de secuencialidad en recomendación

Preferencias y comportamientos de los usuarios no son estáticos; evolucionan con el tiempo.

Secuencialidad permite

- Capturar cambios dinámicos en los intereses del usuario.
- Detectar patrones temporales, como el hecho de que ciertos productos o contenidos sean más populares en determinados momentos.
- Mejorar la relevancia de las recomendaciones, especialmente en escenarios donde el orden de las interacciones importa (por ejemplo, en una lista de reproducción de música).



# Ejemplo recomendación con sesiones de navegación.

¿ Qué es una sesión?

- Usuario Anónimo
- Id de sesión
- Secuencia de clicks. Visitas orgánicas a distintos productos

# Ejemplo

Feedback implícito: si el **usuario / sesión** visitó el ítem y cuando

Session	Item	Timestamp
1	24	5
1	34	10
1	17	15
2	98	18
...	...	...

# Task

## Next-item prediction

Dada una secuencia de ítems previamente visitados, predecir el/los siguiente/s ítem/s que el usuario verá.

En la práctica, se recomienda la lista top-N predicha:

**(24,34) → ?**

**[5,17,68,99,7]**

# Trabajo previo

## Item k-Nearest Neighbors

- Matriz de similaridad: co-ocurrencia de items en base a clicks. Usa solamente el último click como “prior”

## Markov Decision Process:

- < estados , acciones (recomendaciones) , recompensas >
- No escala si consideramos todas las secuencias de acciones (ítems) posibles

## General Factorization Framework:

- Usa información pasada, pero no considera el orden de la secuencia.

# Uno de los primeros papers de datos secuenciales para recomendación

## SESSION-BASED RECOMMENDATIONS WITH RECURRENT NEURAL NETWORKS

**Balázs Hidasi \***

Gravity R&D Inc.  
Budapest, Hungary  
balazs.hidasi@gravityrd.com

**Alexandros Karatzoglou**

Telefonica Research  
Barcelona, Spain  
alexk@tid.es

**Linas Baltrunas †**

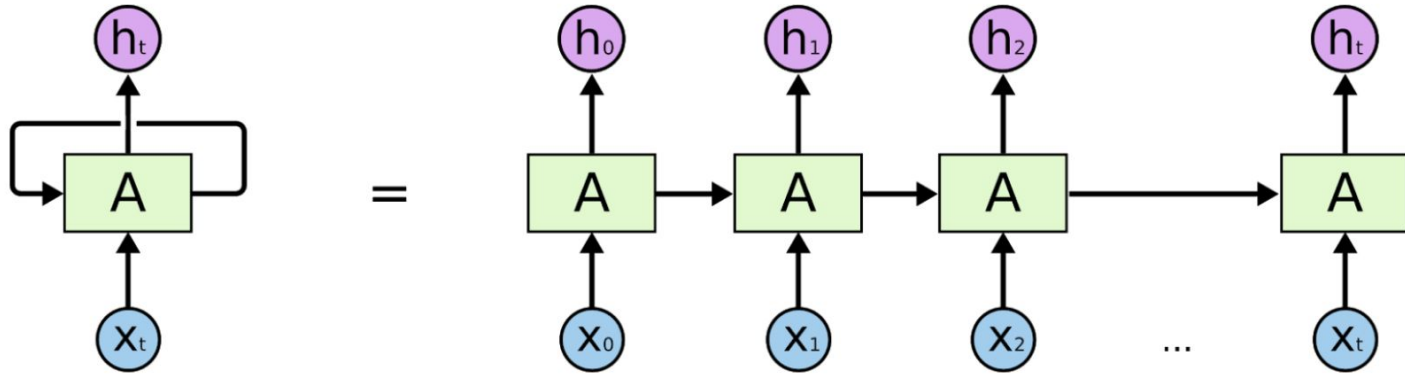
Netflix  
Los Gatos, CA, USA  
lbaltrunas@netflix.com

**Domonkos Tikk**

Gravity R&D Inc.  
Budapest, Hungary  
domonkos.tikk@gravityrd.com

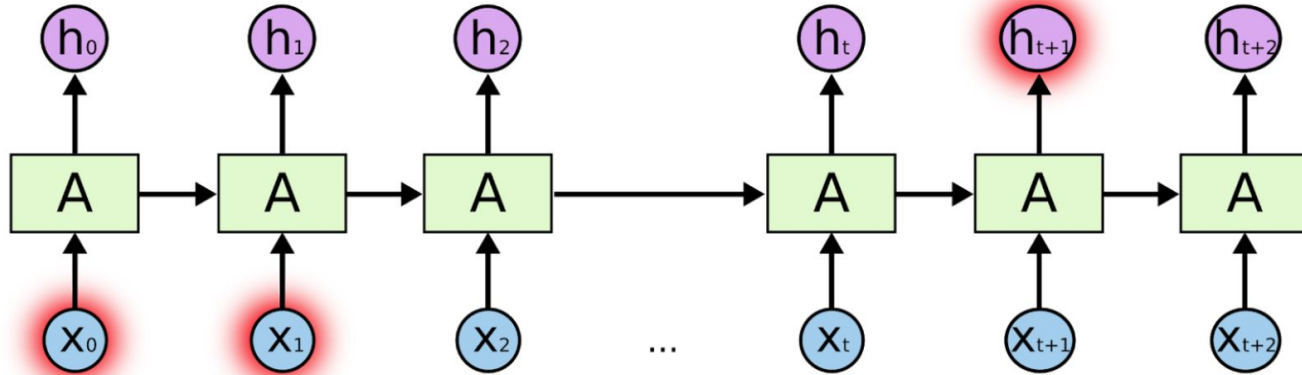
# Recurrent Neural Networks

Tratan datos secuenciales



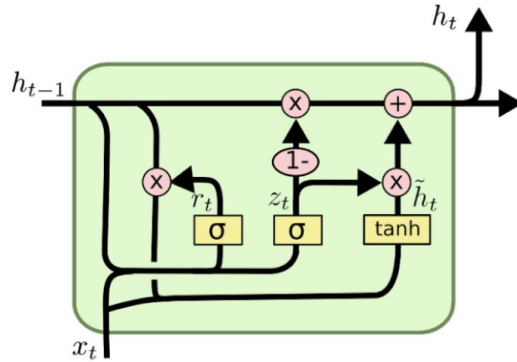
# RNN: Vanishing Gradient

Las RNN no es capaz de capturar dependencias de largo plazo



# GRU: Gated Recurrent Units

Resuelven el problema del gradiente. Propuestas por Bengio et al. en 2014.



$$z_t = \sigma(W_z \cdot [h_{t-1}, x_t])$$

$$r_t = \sigma(W_r \cdot [h_{t-1}, x_t])$$

$$\tilde{h}_t = \tanh(W \cdot [r_t * h_{t-1}, x_t])$$

$$h_t = (1 - z_t) * h_{t-1} + z_t * \tilde{h}_t$$

Permite recordar y olvidar, selectivamente, por intervalos indeterminados de tiempo. Gradiente no desaparece ni explota.



# GRU4REC

Idea:

- Aprovechar dependencias de largo plazo en la sesión para mejorar las recomendaciones.
- Arquitectura GRU

# Resultados reportados en el paper

Method	Recall@20	MRR@20
<b>POP</b>	0.0050	0.0012
<b>S-POP</b>	0.2672	0.1775
<b>BPR-MF</b>	0.2574	0.0618
<b>GRU4Rec</b>	<b>0.5781</b>	<b>0.2375</b>

**POP** : Most Popular

**S-POP** : Session Most Popular

**BPR - MF** : Factorización  
Matricial usando BPR



# Transformer4Rec

## **Transformers4Rec: Bridging the Gap between NLP and Sequential / Session-Based Recommendation**

Gabriel de Souza Pereira  
Moreira  
gmoreira@nvidia.com  
NVIDIA  
São Paulo, Brazil

Sara Rabhi  
srabhi@nvidia.com  
NVIDIA  
Ontario, Canada

Jeong Min Lee\*  
jeongmin@fb.com  
Facebook AI  
California, United States

Ronay Ak  
ronaya@nvidia.com  
NVIDIA  
Florida, United States

Even Oldridge  
eoldridge@nvidia.com  
NVIDIA  
British Columbia, Canada

# Transformer4Rec

- Basado en la arquitectura Transformer.
- Usa mecanismos de autoatención para modelar contexto.

Objetivo: Predecir un momento específico, no sólo el futuro.

- En lugar de palabras el input son ítems en una sesión.
- Incorpora otros features de los ítems (ej. categoría)

# Arquitectura

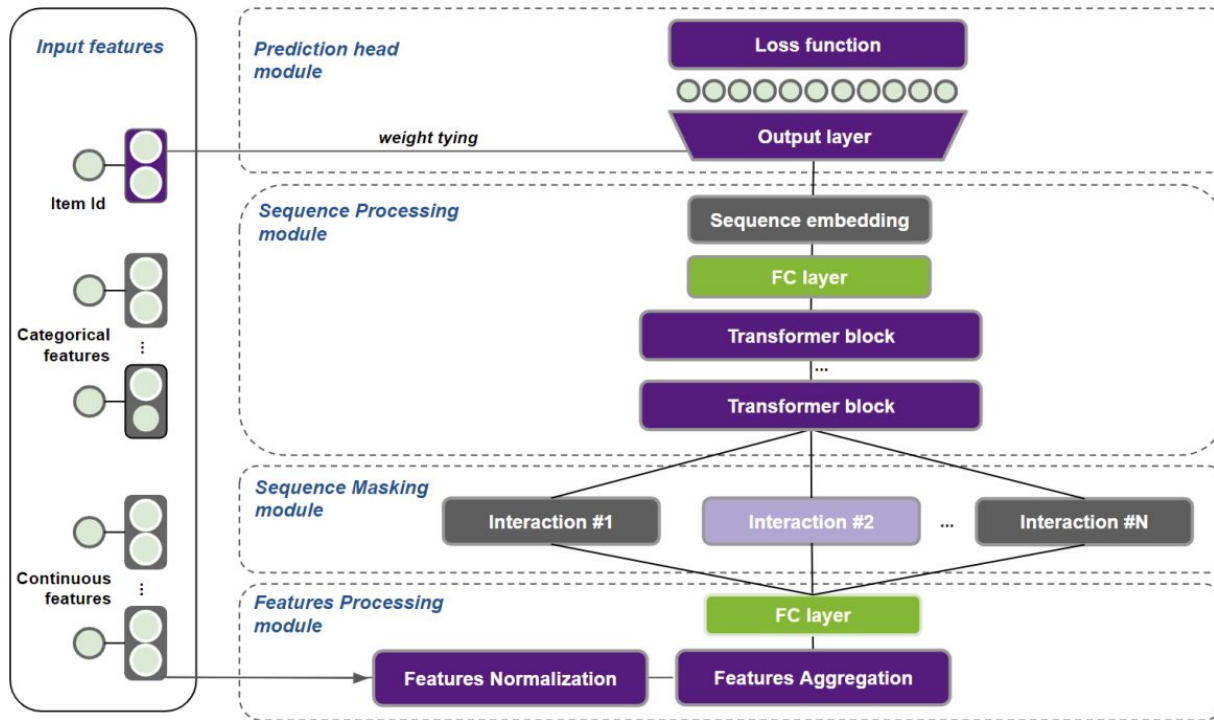


Figure 3: Transformers4Rec neural meta-architecture

# Preguntas de investigación

**RQ1: ¿Pueden las arquitecturas basadas en transformers proporcionar predicciones precisas del siguiente clic para las secuencias de usuario más cortas que se encuentran en la tarea de recomendación basada en sesiones?**

**RQ2: ¿Cómo se desempeñan comparativamente las técnicas de entrenamiento de Causal LM, Masked LM, Permutation LM y Replacement Token Detection para la tarea de recomendación basada en sesiones?**

**RQ3: ¿Cuáles son los enfoques efectivos para integrar características adicionales, comúnmente referidas como información adicional o side information, en arquitecturas transformer con el fin de mejorar la precisión de las recomendaciones?**

# Resultados

**Table 2: Experimental Results: RQ1 / RQ2 / RQ3**

		REES46 eCommerce		YOOCHOOSE eCommerce		G1 news		ADRESSA news	
Algorithm		NDCG@20	HR@20	NDCG@20	HR@20	NDCG@20	HR@20	NDCG@20	HR@20
<b>RQ1</b>	V-SkNN	0.2187	0.4662	0.2975	0.5110	0.3511	0.6601	0.3590	0.7210
	STAN	0.2194	0.4797	0.3082	0.5196	0.3570	<b>0.6681</b>	0.3635	0.7246
	VSTAN	0.2200	<b>0.4857*</b>	0.3097	0.5206	<u>0.3586</u>	<u>0.6668</u>	0.3617	0.7241
	GRU4Rec (FT)	<u>0.2231</u>	0.4414	<u>0.3442</u>	0.5891	0.2596	0.5029	0.3007	0.6052
	GRU4Rec (SWT)	0.2204	0.4359	0.3431	0.5885	0.2666	0.5183	0.2967	0.5948
	GRU (CLM)	0.2139	0.4315	0.2975	<u>0.6129</u>	0.3549	0.6632	<u>0.3799</u>	<b>0.7413</b>
	GPT-2 (CLM)	0.2165	0.4338	0.2975	0.6065	0.3560	0.6620	0.3790	0.7398
	Transformer-XL (CLM)	0.2197	0.4404	0.3585	0.6133	0.3294	0.6192	<b>0.3811*</b>	0.7382
	BERT (MLM)	0.2218	0.4672	<b>0.3750*</b>	<b>0.6349*</b>	0.3549	0.6549	0.3725	0.7221
	ELECTRA (RTD)	<b>0.2430</b>	0.4768	0.3722	0.6294	<b>0.3588</b>	0.6600	0.3729	0.7226
	XLNet (PLM)	0.2422	0.4760	0.3681	0.6282	0.3551	0.6634	0.3673	0.7212
<b>RQ2</b>	XLNet (PLM) - original	0.2422	0.4760	0.3681	0.6282	0.3551	<b>0.6634*</b>	0.3673	0.7212
	XLNet (CLM)	0.2108	0.4219	0.3557	0.6079	0.3551	0.6508	0.3770	<b>0.7378*</b>
	XLNet (RTD)	<b>0.2546*</b>	<b>0.4886*</b>	<b>0.3776</b>	0.6373	<b>0.3609</b>	0.6611	0.3816	0.7329
	XLNet (MLM)	0.2428	0.4763	<b>0.3776</b>	<b>0.6384*</b>	0.3607	0.6605	<b>0.3822</b>	0.7349
<b>RQ3</b>	XLNet (MLM) - item id	0.2428	0.4763	0.3776	0.6384	0.3607	0.6605	0.3822	0.7349
	Concat. merge	0.2522	0.4782	-	-	0.3652	0.6714	<b>0.3912*</b>	<b>0.7488*</b>
	Concat. merge + SOHE	<b>0.2542*</b>	<b>0.4858</b>	-	-	<b>0.3675*</b>	<b>0.6721*</b>	0.3886	0.7463
	Element-wise merge	0.2529	0.4854	-	-	0.3614	0.6678	0.3892	0.7433



# Sistemas recomendadores grupales

# ¿Por qué recomendación para grupos?

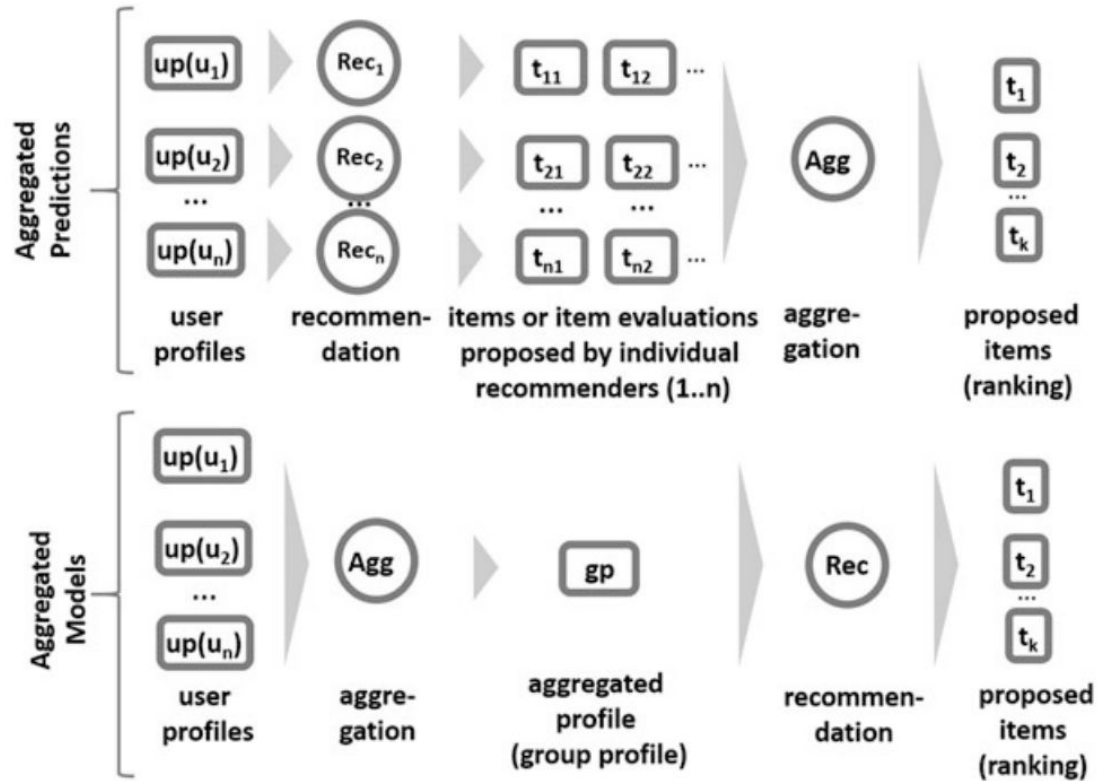
Las decisiones a menudo se toman en grupo: ver películas, elegir restaurantes, planificar viajes.

La recomendación individual puede no satisfacer a todos los miembros del grupo.



# Estrategias de agregación

1. **A nivel de recomendaciones:** Generar primero para individuos, luego agregar.
2. **A nivel de perfiles:** Combinar perfiles y luego recomendar.



**Fig. 2.1** Two basic aggregation strategies in group recommendation: (1) recommendation based on *single user profiles* with a downstream aggregation of items (or evaluations/ratings) recommended to group members/users (*aggregated predictions*) and (2) recommendation based on *aggregated models* (*group profiles*)

# Métodos de agregación (clásicos)

**Promedio:** Tomar el promedio de las preferencias.

**Votación:** Opción más popular entre miembros.

**Mínimo Descontento:** Menor insatisfacción general.

**Optimización:** Modelos matemáticos para maximizar satisfacción general.

# Más métodos de agregación

Aggregation strategy	Description	Recommendation
Additive Utilitarian (ADD) [C]	Sum of item-specific evaluations	$\underset{(t \in I)}{\operatorname{argmax}}(\sum_{u \in G} \operatorname{eval}(u, t))$
Approval Voting (APP) [M]	Number of item-specific evaluations above an approval threshold	$\underset{(t \in I)}{\operatorname{argmax}}( \{u \in G : \operatorname{eval}(u, t) \geq \operatorname{threshold}\} )$
Average (AVG) [C]	Average of item-specific evaluations	$\underset{(t \in I)}{\operatorname{argmax}}(\frac{\sum_{u \in G} \operatorname{eval}(u, t)}{ G })$
Average without Misery (AVM) [C]	Average of item-specific evaluations (if all evaluations are above a defined threshold)	$\underset{(t \in I : \nexists u \in G   \operatorname{eval}(u, t) \leq \operatorname{threshold})}{\operatorname{argmax}}(\frac{\sum_{u \in G} \operatorname{rating}(u, t)}{ G })$
Borda Count (BRC) [M]	Sum of item-specific scores derived from item ranking	$\underset{(t \in I)}{\operatorname{argmax}}(\sum_{u \in G} \operatorname{score}(u, t))$
Copeland Rule (COP) [M]	Number wins ( $w$ )—number losses ( $l$ ) in pair-wise evaluation comparison	$\underset{(t \in I)}{\operatorname{argmax}}( w(t, I - \{t\})  -  l(t, I - \{t\}) )$
Fairness (FAI) [C]	Item ranking as if individuals ( $u \in G$ ) choose them one after the other	$\underset{(t \in I)}{\operatorname{argmax}}(\operatorname{eval}(u, t))$ [in each iteration]
Least Misery (LMS) [B]	Minimum item-specific evaluation	$\underset{(t \in I)}{\operatorname{argmax}}(\operatorname{mineval}(t))$
Majority Voting (MAJ) [B]	Majority of evaluation values per item	$\underset{(t \in I)}{\operatorname{argmax}}(\operatorname{majorityeval}(t))$
Most Pleasure (MPL) [B]	Maximum item-specific evaluation	$\underset{(t \in I)}{\operatorname{argmax}}(\operatorname{maxeval}(t))$
Most Respected Person (MRP) [B]	Item-evaluations of most respected user	$\underset{(t \in I)}{\operatorname{argmax}}(\operatorname{eval}(u_{\operatorname{mrp}}, t))$
Multiplicative (MUL) [C]	Multiplication of item-specific evaluations	$\underset{(t \in I)}{\operatorname{argmax}}(\prod_{u \in G} \operatorname{eval}(u, t))$
Plurality Voting (PLU) [M]	Item with the highest #votes from $u \in G$	$\underset{(t \in I)}{\operatorname{argmax}}(\operatorname{votings}(t))$ [in each iteration]

# Ejemplo

**Table 2.6** Rating predictions and corresponding *scores* (scores are used by *BRC*)

Name Item		Rating predictions $\hat{r}_{ij}$ (scores)					Aggregation		
		$u_1$	$u_2$	$u_3$	$u_4$	$u_5$	AVG	BRC	LMS
$t_1$	Vienna	5.0(9)	3.5(2)	1.0(0)	4.5(7)	5.0(9)	3.8	27	1.0
$t_2$	Yellowstone	2.5(0)	4.0(4)	3.0(3)	2.0(0)	1.1(0)	2.5	7	1.1
$t_3$	New York	4.9(8)	3.8(3)	4.0(7)	3.3(4)	4.0(5)	4.0	27	3.3✓
$t_4$	Blue Mountains	3.1(2)	5.0(9)	4.2(8)	2.4(1)	4.4(8)	3.8	28	2.4
$t_5$	London	4.0(4)	4.3(7)	3.3(5)	4.1(6)	2.9(3)	3.7	25	2.9
$t_6$	Beijing	4.5(6)	4.1(5)	5.0(9)	3.2(3)	4.2(6)	4.2✓	29✓	3.2
$t_7$	Cape Town	4.2(5)	4.2(6)	3.4(6)	3.1(2)	3.8(4)	3.7	23	3.1
$t_8$	Yosemite	3.4(3)	2.6(0)	1.6(1)	5.0(9)	2.4(2)	3.0	15	1.6
$t_9$	Paris	4.7(7)	3.1(1)	2.7(2)	3.6(5)	2.2(1)	3.3	16	2.2
$t_{10}$	Pittsburgh	2.6(1)	4.5(8)	3.1(4)	4.6(8)	4.3(7)	3.8	28	2.6

Recommendations are derived on the basis of aggregation functions (*AVG*, *BRC*, *LMS*). The ✓ symbol indicates the item with the best evaluation

**AVG:** Promedio  
Rating

**BRC:** Suma scores  
borda-count entre  
paréntesis.

**LMS:** Least Misery.  
Rating menor más  
grande.

# Desafíos en la Recomendación Grupal

Conflictos de preferencia entre miembros.

Peso de las opiniones: ¿todos iguales?

Cambios en la composición del grupo  
(ej género, edad, raza).





# Evaluación de Recomendadores Grupales

Métricas para satisfacción grupal.

Desafíos: No todos los miembros pueden estar igual de satisfechos.

# Evaluación de Recomendadores Grupales

## 1. Satisfacción del Grupo:

- **Agregación de Satisfacción Individual:** Mide la satisfacción de cada miembro y la agrupa, a través de promedios, mínimos, o mediana de las satisfacciones individuales.

## 2. Descontento del Grupo:

- **Descontento Mínimo:** Busca minimizar el descontento del miembro menos satisfecho.
- **Descontento Máximo:** Trata de minimizar la máxima insatisfacción dentro del grupo.
- **Descontento Promedio:** Intenta minimizar el promedio de insatisfacción entre los miembros.

## 3. Fairness (Justicia):

- **Fairness Individual:** Garantiza que cada miembro del grupo reciba al menos una recomendación que le agrade.
- **Paridad de Fairness:** Busca asegurar que todas las preferencias de los miembros del grupo se consideren por igual.