

Meta Graph Learning para recomendaciones con cola larga



Meta Graph Learning for Long-tail Recommendation

Chunyu Wei*
Tsinghua University
Beijing, China
weicy15@icloud.com

Jian Liang*
Independent Researcher
Beijing, China
liangjianzb12@gmail.com

Di Liu
Alibaba Group
Beijing, China
wendu.ld@alibaba-inc.com

Zehui Dai
Alibaba Group
Beijing, China
zehui.dzh@alibaba-inc.com

Mang Li
Alibaba Group
Hangzhou, China
mang.li@alibaba-inc.com

Fei Wang
Cornell University
New York, United States
few2001@med.cornell.edu

ABSTRACT

Highly skewed long-tail item distribution commonly hurts model performance on tail items in recommendation systems, especially for graph-based recommendation models. We propose a novel idea to learn relations among items as an auxiliary graph to enhance the graph-based representation learning and make recommendations collectively in a coupled framework. This raises two challenges.



Grupo: 10

Luis Arias
Benjamín Lillo
Darwin Sanhueza

Tabla de contenidos

01

Introducción al
problema y
motivación

02

Trabajo
relacionado y
propuesta

03

Datos

04

Metodología

05

Experimentos y
resultados

06

Conclusiones



01

Introducción al problema

GCNs: Graph Convolutional Networks

- Forma eficiente de integrar vecinos multi-hop
- Buen performance

GCNs: Graph Convolutional Networks + distribuciones de cola larga

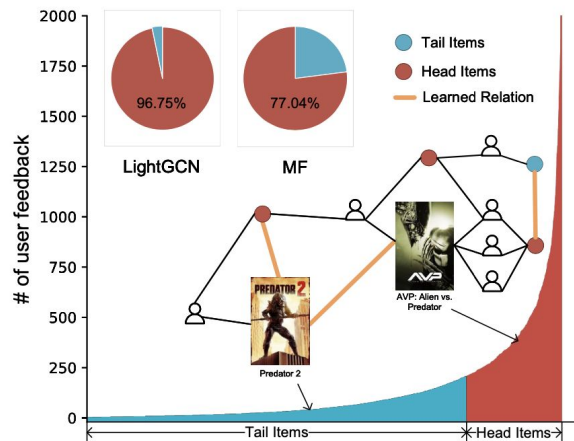


Figure 1: The long-tail distribution for items in Movielens-1M. Pie charts show recommending results trained on Movielens-1M. Orange lines is the learned item relations.

Obtenido del artículo Meta Graph Learning for Long-tail Recommendation

- Se produce overfitting para los ítems con más interacciones (sesgo de popularidad)
- Matthew effect: “the rich get richer”

Pocas interacciones != malo

Objetivo propuesto:

Un framework llamado Meta Graph Learning que aprende relaciones entre los ítems en un grafo auxiliar, que actualiza el grafo original para prevenir el traspaso del sesgo de popularidad desde la distribución hacia la recomendación.

Desafíos a afrontar

Skewed Downstream Information

Se debe encontrar una forma de mitigar el impacto negativo de la distribución desbalanceada de interacciones en el grafo aprendido para aumentar el desempeño.

Negative transfer

El grafo auxiliar puede introducir ruido en la recomendación de los ítems con más interacciones, por lo que se debe evitar en estos para enfocarse en mejorar a los ítems de la cola y no empeorar los de la cabeza.



02

Trabajo relacionado

En cuanto a distribuciones de cola larga

Estrategias para rebalancear un dataset

- Resampling
- Refinar la función de pérdida cambiando parámetros
- Añadir pesos a ciertas interacciones usuario-ítem

Aplicación de soluciones para problema de cold-start, como Información externa para aportar info sobre ítems en la cola.



En cuanto a modelos de aprendizaje de grafo

El grafo de input en GCN's, al ser construido o procesado por reglas definidas, puede perder aristas con info relevante o tener info redundante.

Se ha estudiado formas de añadir pesos a aristas significativas, y de reconstruir grafos de input a partir de distribuciones de probabilidad

Sin embargo, estos estudios no consideran distribuciones de cola larga, y se cree que pueden incrementar el sesgo de popularidad.



En cuanto a Meta-learning

Meta-learning intenta diseñar modelos que puedan aprender nuevas habilidades y adaptarse a nuevos entornos con muestras de entrenamiento.

Se ha intentado para entrenar redes neuronales con grafo.

En la propuesta, se usa meta-learning para mejorar el grafo auxiliar y solucionar problema de negative transfer.



03

Datos

Datasets para los experimentos

- Ambos con distribución de cola larga
- No se utilizan features de usuarios
- Para cada usuario, se parte el historial de interacciones en 8:1:1 para training, validación y testing.

Table 1: Descriptive statistics of the datasets.

	Movielens-1M	Bookcrossing
# Users	6,040	50,454
# Items	3,706	222,154
# Feedback	1,000,209	1,031,175
Density	0.04468	0.00009
Item Features	Title, Genres, Year	Title, Author, Year, Publisher
Feedback Portion of Top 20% Items	65.25%	50.84%

Obtenido del artículo Meta Graph Learning for Long-tail Recommendation

04

Metodología

Arquitectura Meta Graph Learning

- La idea principal de MGL es generar relaciones entre items adquiridas por un meta edge generator
- Introducción de una estrategia de aprendizaje contrastivo
- Análisis teóricos sobre MGL

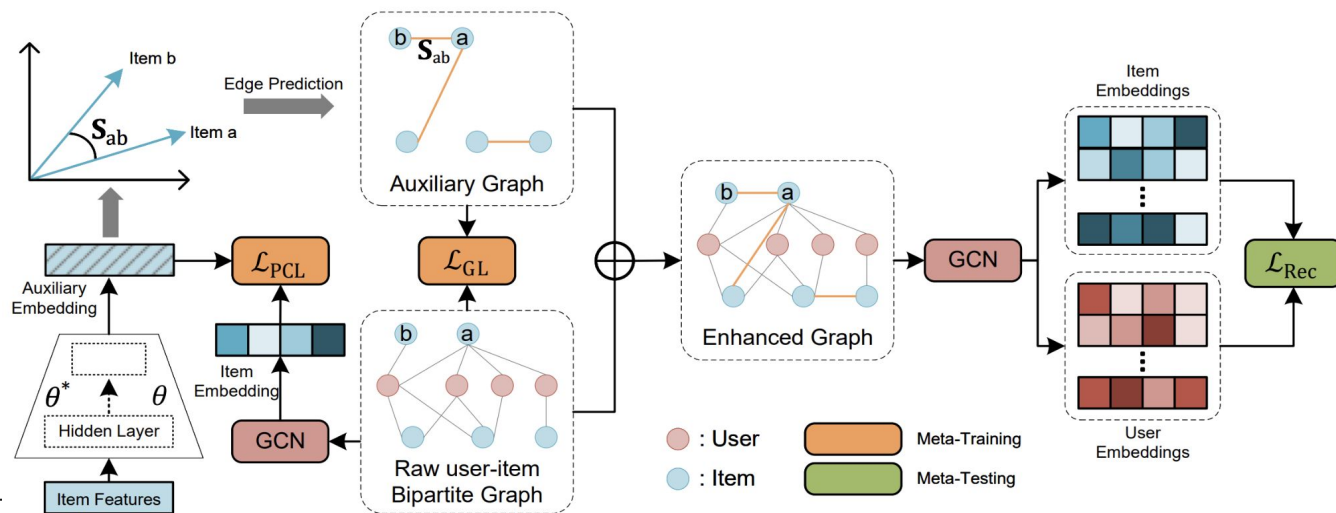


Figure 2: Framework of MGL.

- **Meta Edge Generator**
- Popularity-aware
- Contrastive Learning
- Neighbor Learning for Recommendation
- Optimización
- Análisis de MGL

- Se utiliza un item embedding encoder para aprender embeddings auxiliares de items a partir de features
- Se utiliza una función de similitud coseno para predecir si existe una relación entre dos items
- Para entrenar el meta edge generator de manera imparcial se construye una matriz de co-ocurrencia de items (S')
- Se limita el número de vecinos a k en la matriz de co-ocurrencia
- La función de pérdida para entrenar el meta edge generator se basa en las conexiones predichas

- Meta Edge Generator
- **Popularity-aware
Contrastive Learning**
- Neighbor Learning for
Recommendation
- Optimización
- Análisis de MGL

- Se busca mantener la consistencia entre dos grafos que representan items
- Los grafos contienen embeddings de items que codifican información sobre features y consumos de items
- Se utiliza un modelo recomendador GCN en el grafo original para obtener los embeddings originales de los elementos
- Se aplica una función lineal para alinear los embeddings auxiliares con los originales
- Se utiliza una pérdida de contraste que considera la popularidad de los items para mejorar la consistencia entre los grafos
- La pérdida de contraste se aplica selectivamente a items en función de su popularidad



- Meta Edge Generator
- Popularity-aware
- Contrastive Learning
- **Neighbor Learning for Recommendation**
- Optimización
- Análisis de MGL

- Después de entrenar un meta edge generator para reconstruir información colaborativa imparcial de elementos
- Se crea un grafo auxiliar, seleccionando principalmente elementos principales como candidato para reconstruir conexiones hacia todos los items
- Se calcula una matriz de similitud item-item
- Se mantienen las conexiones con las similitudes altas
- Se actualiza la estructura final del grafo y se utiliza GCN para obtener embeddings finales
- Se utiliza L_{rec} como función de pérdida para recomendaciones

- Meta Edge Generator
- Popularity-aware Contrastive Learning
- Neighbor Learning for Recommendation
- **Optimización**
- Análisis de MGL

- MGL emplea esquema meta-learning para entrenar un meta edge generator
- Se actualiza el generador en el entrenamiento y testeo
- El entrenamiento combina la pérdida convencional (L_{GL}) y una pérdida de contraste sensible a la popularidad (L_{PCL}) con un hiperparámetro λ que controla la ponderación de la pérdida de contraste
- La optimización se realiza simultáneamente en el meta-train y meta-test

- Meta Edge Generator
- Popularity-aware Contrastive Learning
- Neighbor Learning for Recommendation
- Optimización
- **Análisis de MGL**

- La optimización conjunta busca minimizar las pérdidas tanto en meta-train como meta-test
- Se maximiza la similitud entre las direcciones de mejora en ambas etapas para garantizar una optimización coordinada
- La regularización evita que el meta edge generator se enfoque en la tarea más fácil y garantiza una optimización equilibrada
- Previene el sobreajuste a datos sesgados en la etapa de recomendación



05

Experimentos

5.1 Descripción del Dataset

Table 1: Descriptive statistics of the datasets.

	Movielens-1M	Bookcrossing
# Users	6,040	50,454
# Items	3,706	222,154
# Feedback	1,000,209	1,031,175
Density	0.04468	0.00009
Item Features	Title, Genres, Year	Title, Author, Year, Publisher
Feedback Portion of Top 20% Items	65.25%	50.84%

5.2 Configuración Experimental

- Se utilizan dos métricas de evaluación ampliamente conocidas: Normalized Discounted Cumulative Gain@K (NDCG@K) y Hit Ratio@K (HR@K).
- Se comparan nueve métodos diferentes con MGL para demostrar su efectividad en la recomendación de larga cola basada en grafos.

5.3 Comparación de Rendimiento

Table 2: The recommendation performance of MGL versus baselines on *MovieLens-1M*.

Model	Overall				Head				Tail			
	NDCG@10	HR@10	NDCG@20	HR@20	NDCG@10	HR@10	NDCG@20	HR@20	NDCG@10	HR@10	NDCG@20	HR@20
BPR-MF [37]	0.04094	0.05960	0.05693	0.10419	0.05323	0.08358	0.07340	0.14458	0.01868	0.03387	0.02538	0.05788
NGCF [48]	0.04218	0.06306	0.05933	0.11081	0.05612	0.09072	0.07687	0.15245	0.01947	0.03540	0.02669	0.06095
LightGCN [18]	0.04807	0.06801	0.06508	0.11525	0.05556	0.08619	0.07486	0.14406	0.02299	0.04175	0.03125	0.07104
Over sampling [5]	0.04728	0.06655	0.06426	0.11345	0.05462	0.08317	0.07427	0.14194	<u>0.02331</u>	<u>0.04263</u>	<u>0.03199</u>	0.07330
Down sampling [16]	0.03449	0.04797	0.04639	0.08097	0.04200	0.06316	0.05785	0.11108	0.01968	0.03594	0.02649	0.05991
GAT [46]	0.04644	0.06770	0.06324	0.11409	<u>0.06116</u>	<u>0.09529</u>	<u>0.08265</u>	<u>0.16068</u>	0.02100	0.03720	0.02812	0.06220
EGLN [51]	<u>0.05010</u>	<u>0.06949</u>	<u>0.06778</u>	<u>0.11839</u>	0.06013	0.09149	0.07996	0.15089	0.02229	0.04084	0.03178	<u>0.07461</u>
MeLU [26]	0.04228	0.06217	0.05899	0.10864	0.05353	0.08430	0.07263	0.14135	0.02075	0.03727	0.02771	0.06155
MIRec [56]	0.04302	0.06364	0.05995	0.11065	0.05470	0.08760	0.07369	0.14409	0.02132	0.03574	0.02779	0.05844
MGL	0.05347	0.07551	0.07295	0.12898	0.06302	0.09610	0.08518	0.16213	0.02666	0.05082	0.03695	0.08761
Improv.	+6.73%*	+8.66%*	+7.64%*	+8.95%*	+3.04%*	+0.85%*	+3.07%*	+0.90%*	+14.38%*	+19.22%*	+15.51%*	+17.42%*

Table 3: The recommendation performance of MGL versus baselines on *Bookcrossing*.

Model	Overall				Head				Tail			
	NDCG@10	HR@10	NDCG@20	HR@20	NDCG@10	HR@10	NDCG@20	HR@20	NDCG@10	HR@10	NDCG@20	HR@20
BPR-MF [37]	0.00764	0.01273	0.00995	0.02034	0.01018	0.01974	0.01314	0.03066	0.00444	0.00812	0.00625	0.01433
NGCF [48]	0.01396	0.02265	0.01813	0.03648	0.01357	0.02576	0.01818	0.04289	0.01074	0.01723	0.01460	<u>0.03076</u>
LightGCN [18]	0.01652	0.02445	0.02062	0.03788	0.01738	0.02887	0.02109	0.04279	0.01113	0.01804	0.01430	0.02917
Over sampling [5]	0.01660	0.02356	0.02053	0.03778	0.01703	0.02894	0.02096	0.04173	0.01121	0.01882	0.01461	0.02913
Down sampling [16]	0.01216	0.01789	0.01470	0.02705	0.01328	0.02116	0.01533	0.03271	0.00940	0.01510	0.01249	0.02595
GAT [46]	0.01540	0.02305	0.01951	0.03648	0.01560	0.02676	0.01976	0.04209	0.01106	0.01794	0.01476	0.03066
EGLN [51]	<u>0.01816</u>	<u>0.02717</u>	<u>0.02302</u>	<u>0.04300</u>	<u>0.01819</u>	<u>0.03237</u>	<u>0.02361</u>	<u>0.04761</u>	<u>0.01161</u>	0.01844	<u>0.01495</u>	0.03016
MeLU [26]	0.01371	0.02237	0.01775	0.03572	0.01332	0.02426	0.01820	0.04285	0.01127	0.01885	0.01342	0.02642
MIRec [56]	0.01460	0.02134	0.02006	0.03908	0.01620	0.02906	0.02077	0.04619	0.01116	<u>0.01943</u>	0.01353	0.02765
MGL	0.01966	0.02916	0.02402	0.04649	0.01869	0.03251	0.02371	0.04829	0.01488	0.02345	0.01823	0.03545
Improv.	+8.23%*	+7.37%*	+4.35%*	+8.12%*	+2.72%*	+0.42%*	+0.44%*	+1.44%	+28.13%	+20.66%	+21.97%	+15.25%*

[†] Boldface denotes the highest score, and underline indicates the best result of the baselines. * denotes that the improvement is significant at $p < 0.05$ level with a two-sample t-test.

5.4 Ablation Studies

Se realizaron estudios de abstracción para investigar el efecto del enfoque de meta-aprendizaje. MGL se compara con dos variantes: **GL** y **RGL**.

- **GL** solo supervisa el generador de bordes mediante la pérdida de recomendación final.
- **RGL** supervisa el generador de bordes directamente con todas las pérdidas, incluida la pérdida de contraste.

5.4 Ablation Studies

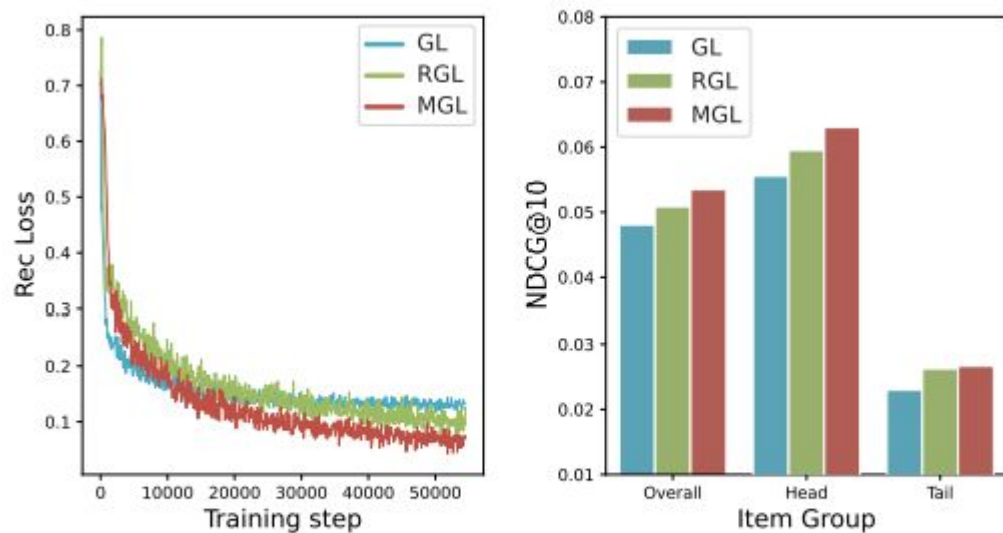


Figure 4: Effect of Meta-Learning on *Movielens-1M*

5.4.1 Estrategia de Aprendizaje por Contraste Diferente

Table 4: Comparison among models on *Movielens-1M*

Model	OVerall		Head		Tail	
	NDCG@10	HR@10	NDCG@10	HR@10	NDCG@10	HR@10
MGL _B	0.04297	0.06113	0.05206	0.07986	0.01123	0.02282
MGL _F	0.04842	0.06935	0.05590	0.08748	0.02452	0.04528
MGL	0.05347	0.07551	0.06302	0.09610	0.02666	0.05082

5.5 Visualización de Embeddings y Estudio de Caso

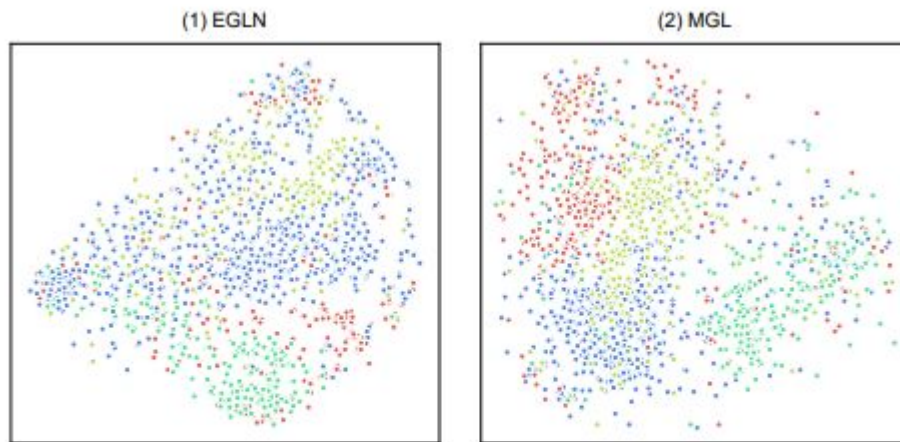


Figure 5: The 2-D visualization of tail items on *Movielens-1M*. The color represents the movie genres.

06

Conclusiones

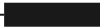
Conclusiones

- El modelo MGL se propone como un enfoque innovador para mejorar el rendimiento de los modelos de recomendación basados en grafos.
- El enfoque se basa en aprender relaciones entre elementos para construir un grafo auxiliar que se utiliza para actualizar la estructura del grafo principal de la recomendación.
- Se aplica un enfoque de meta-aprendizaje para aprender el grafo auxiliar.
- Se utiliza una estrategia de aprendizaje por contraste sensible a la popularidad para evitar la transferencia negativa en elementos principales.



Gracias!

Preguntas?



Anexos



- Meta Edge Generator
- Popularity-aware Contrastive Learning
- Neighbor Learning for Recommendation
- Optimización
- Análisis de MGL

$$S_{i,j} = \sigma \left(\frac{\langle \mathbf{p}_i, \mathbf{p}_j \rangle}{\|\mathbf{p}_i\| \|\mathbf{p}_j\|} \right) \quad (5)$$

$$\mathbf{S}' = \min (1, \mathbf{R}^T \mathbf{R}), \quad (6)$$

$$\mathbf{S}'_{mask} = \mathbf{M} \odot \mathbf{S}' \odot \mathbf{M}^T, \text{ where } \sum \mathbf{M}_{i.} = k, \quad (7)$$

$$\mathcal{L}_{GL} = \|\mathbf{S} - \mathbf{S}'_{mask}\|_F^2, \quad (8)$$

- Meta Edge Generator
- Popularity-aware Contrastive Learning
- Neighbor Learning for Recommendation
- Optimización
- Análisis de MGL

$$\mathbf{E}^{(l)} = [\mathbf{E}_{user}^{(l)}, \mathbf{E}_{item}^{(l)}] = \text{GCN}(\mathbf{E}^{(l-1)}, \mathcal{G}), \quad (9)$$

$$\mathcal{L}_{PCL} = - \sum_{v_i \in \mathcal{I}} \text{Pop}(v_i) \log \frac{\exp(s(h(\mathbf{p}_i), \mathbf{e}_i)/\tau')}{\sum_{v_j \in \mathcal{I}} \exp(s(h(\mathbf{p}_i), \mathbf{e}_j)/\tau')}, \quad (10)$$

$$\text{Pop}(v_i) = 1 - \frac{r}{r + \exp\left(\frac{\sum_{u \in \mathcal{U}} r_{ui}}{r}\right)}, \quad v_i \in \mathcal{I}, \quad (11)$$

- Meta Edge Generator
- Popularity-aware Contrastive Learning
- Neighbor Learning for Recommendation
- Optimización
- Análisis de MGL

$$\hat{\mathbf{p}}_i = g(\mathbf{y}_i; \theta')$$

$$\hat{\mathbf{S}}_{i,j} = \begin{cases} \sigma\left(\frac{\langle \hat{\mathbf{p}}_i, \hat{\mathbf{p}}_j \rangle}{\|\hat{\mathbf{p}}_i\| \|\hat{\mathbf{p}}_j\|}\right), & (\sum_{u \in \mathcal{U}} r_{uj}) > k_h, \\ 0, & \text{Otherwise} \end{cases}, \quad (12)$$

$$\hat{\mathbf{S}}_{i,j} = \begin{cases} \hat{\mathbf{S}}_{i,j}, & \hat{\mathbf{S}}_{i,j} \in \text{top}K(\hat{\mathbf{S}}_i) \\ 0, & \hat{\mathbf{S}}_{i,j} \notin \text{top}K(\hat{\mathbf{S}}_i) \end{cases}, \quad (13)$$

$$\hat{\mathbf{E}}^{(l)} = [\hat{\mathbf{E}}_{user}^{(l)}, \hat{\mathbf{E}}_{item}^{(l)}] = \text{GCN}(\hat{\mathbf{E}}^{(l-1)}, \mathcal{G}_F), \quad (14)$$

- Meta Edge Generator
- Popularity-aware Contrastive Learning
- Neighbor Learning for Recommendation
- Optimización
- Análisis de MGL

$$\mathcal{F}(\theta) = \mathcal{L}_{GL} + \lambda \mathcal{L}_{PCL}, \quad (15)$$

$$\mathcal{J}(\theta', \mathbf{w}, \mathbf{E}^{(0)}) = \mathcal{L}_{rec}, \quad (16)$$

$$\min_{\theta, \mathbf{w}, \mathbf{E}^{(0)}} \mathcal{J}(\theta - \alpha \mathcal{F}'(\theta), \mathbf{w}, \mathbf{E}^{(0)}) + \beta \mathcal{F}(\theta), \quad (17)$$

Algorithm 1 Meta Graph Learning

Input: User-item interaction matrix \mathbf{R} ; Item Attributes \mathbf{Y}

Init: Model Parameters $\theta, \mathbf{w}, \mathbf{E}^{(0)}$; Hyperparameters α, β .

- 1: Calculate \mathbf{S}'_{mask} with \mathbf{R} ;
 - 2: **for** i **in** iterations **do**
 - 3: **Meta-train:** Compute \mathcal{L}_{GL} , \mathcal{L}_{PCL} , and $\mathcal{F}(\theta)$; (Equ. 15)
 - 4: Update parameter $\theta' = \theta - \alpha \nabla_{\theta}$
 - 5: **Meta-test:** Compute $\mathcal{J}(\theta', \mathbf{w}, \mathbf{E}^{(0)})$; (Equ. 16).
 - 6: **Meta-optimization:** Update θ, \mathbf{w} , and $\mathbf{E}^{(0)}$; (Equ. 17)
 - 7: **end for**
-

- Meta Edge Generator
- Popularity-aware Contrastive Learning
- Neighbor Learning for Recommendation
- Optimización
- Análisis de MGL

$$\mathcal{J}(x) = \mathcal{J}(\dot{x}) + \mathcal{J}'(\dot{x}) \times (x - \dot{x}), \quad (18)$$

where \dot{x} is an arbitrary point close to x . We let $x = \theta - \alpha \mathcal{F}'(\theta)$, and we have $\dot{x} = \theta$. Then, we have:

$$\mathcal{J}(\theta - \alpha \mathcal{F}'(\theta)) = \mathcal{J}(\theta) + \mathcal{J}'(\theta) \cdot (-\alpha \mathcal{F}'(\theta)), \quad (19)$$

and the objective Equation 17 becomes:

$$\min_{\theta} \mathcal{J}(\theta) + \beta \mathcal{F}(\theta) - \alpha (\mathcal{J}'(\theta) \cdot \mathcal{F}'(\theta)), \quad (20)$$