

Sobre la Utilidad de VGAE en LightGCL

Matías Francia

Diego Quezada

Contexto

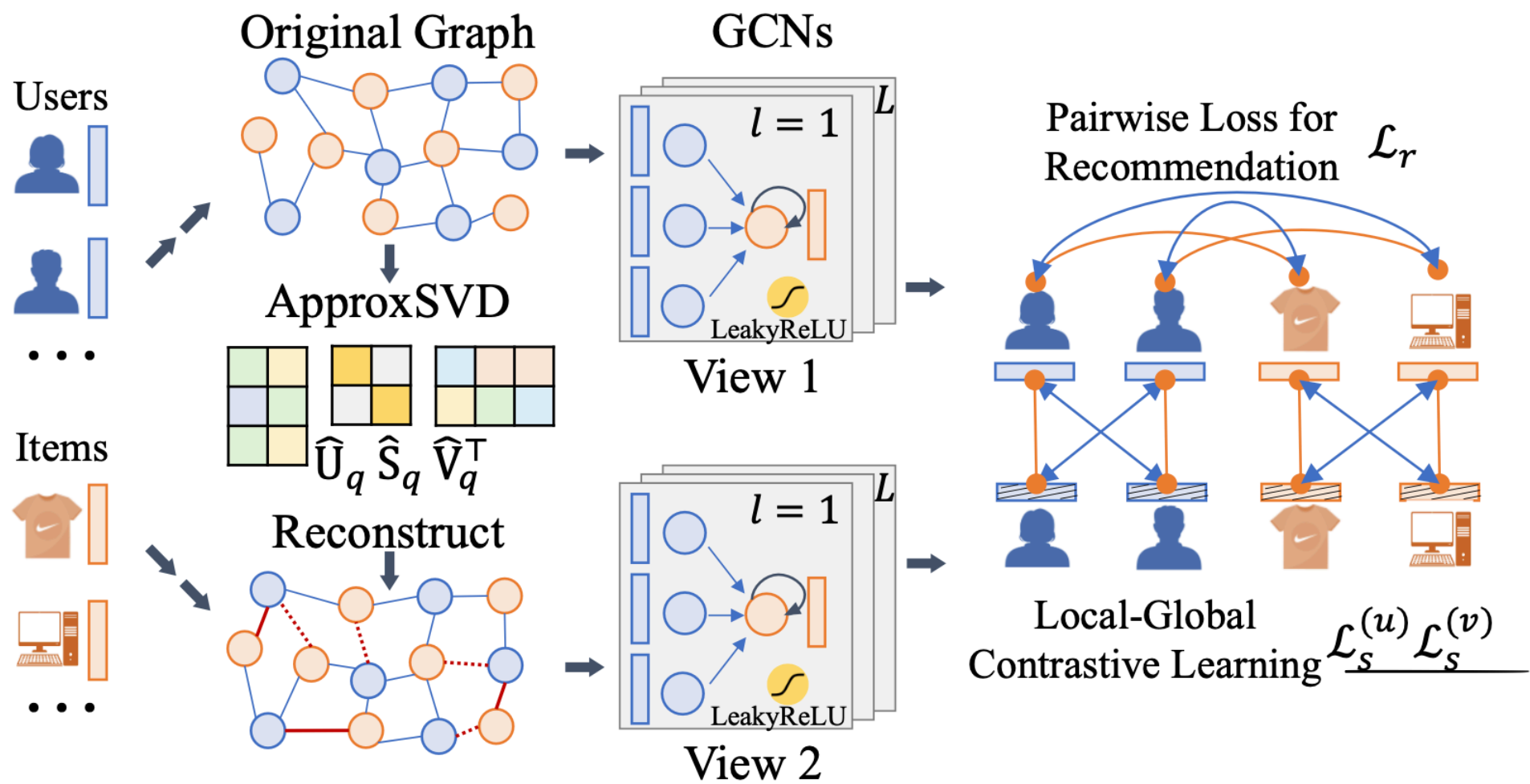
- GNNs han demostrado su efectividad en sistemas recomendadores basados en grafos
- La mayoría de los modelos de filtrado colaborativo basados en GNNs aprenden de manera supervisada, necesitando datos etiquetados
- En la práctica aprender representaciones de usuarios e ítems es un gran desafío debido a que la matriz de interacción es sparse
- GNNs integradas con contrastive learning han demostrado un desempeño superior en la tarea de recomendación junto a su esquema de data augmentation

Problema

Aprender representaciones de usuarios e ítems a partir de matrices de interacción sparse, capturando tanto información local como global del grafo

LightGCL

- Es un framework de aprendizaje contrastivo en grafos, basado en GCN y entrenado para tareas de recomendación
- Se basa en la generación de otra vista del grafo para el entrenamiento contrastivo, mediante el uso de una SVD aproximada
- La SVD aproximada busca generar una reconstrucción del grafo que contenga mayor información colaborativa global
- Muestra una mejora en la eficiencia de entrenamiento respecto a otros enfoques basados en GCN



Configuración

- Tarea: Link prediction
- Datos: Matriz de interacción
- Modelo: GCN
- Función de pérdida: BPR y CL

Profundizando

Datos de entrada

- Matriz de interacción original normalizada \tilde{A}
- Matriz de interacción reconstruida (SVD) \hat{A}

Configuración GCN:

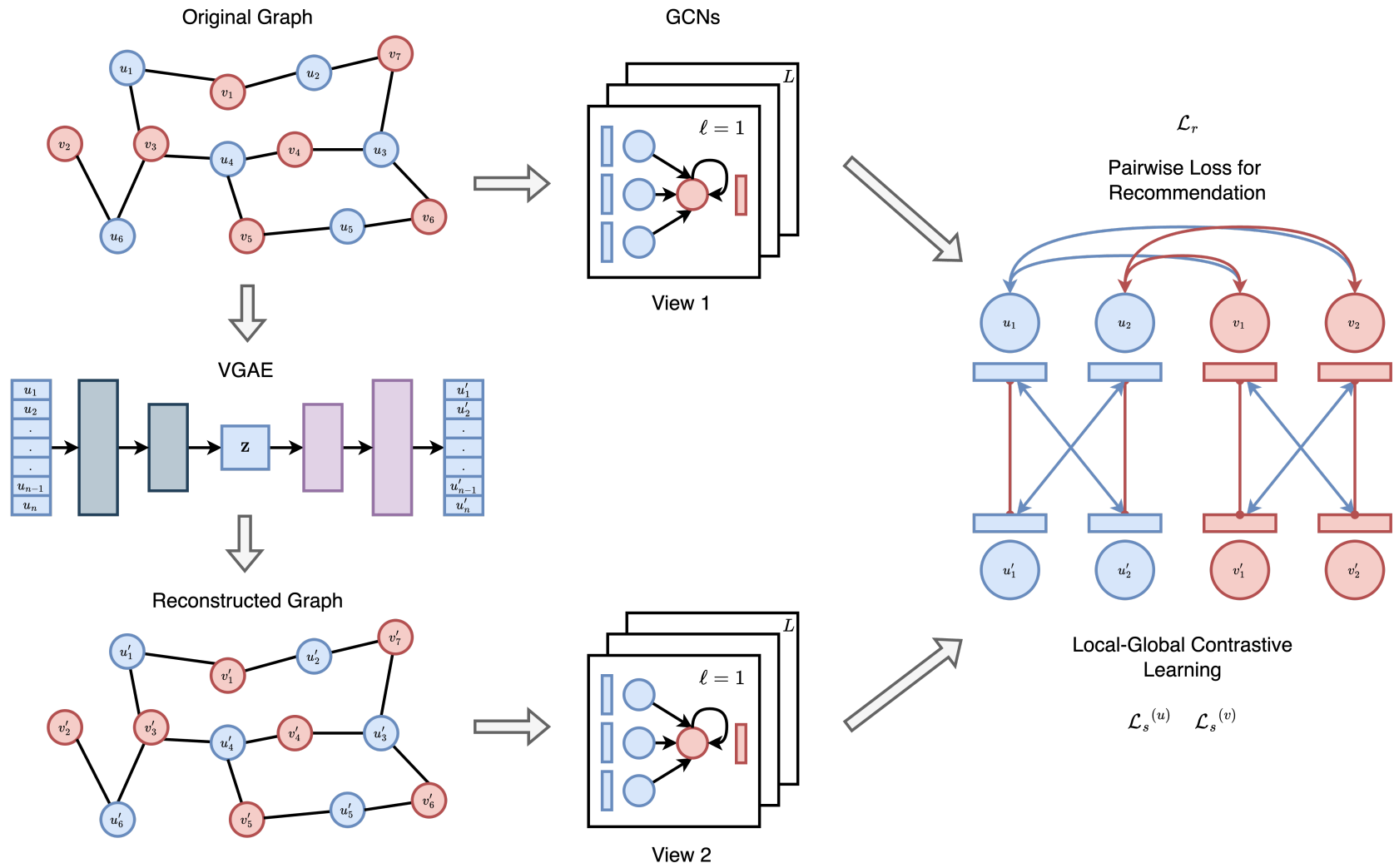
- l capas de paso de mensajes
- Listas de matrices de embeddings, todos con $l + 1$ dimensiones:
 - $\mathbf{E}_u, \mathbf{E}_v$: matrices usadas para paso de mensajes, primer elemento de cada una son los parámetros entrenables del modelo
 - $\mathbf{Z}_u, \mathbf{Z}_v / \mathbf{G}_u, \mathbf{G}_v$: embeddings de vista 1 y vista 2, respectivamente
- Activaciones: Sigmoides (BPR) y Softmax con temperatura τ (CL)

- Luego de las declaraciones se hace el forward pass y se actualizan los parámetros basándose en la unión de 2 funciones de pérdida: contrastiva (\mathcal{L}_s , InfoNCE) y de recomendación (\mathcal{L}_r , BPR).
- Función de pérdida:

$$\mathcal{L} = \mathcal{L}_r + \lambda_1 (\mathcal{L}_s^{(u)} + \mathcal{L}_s^{(v)}) + \lambda_2 \|\Theta\|_2^2$$

Contribución

- **Limitaciones LightGCL:** SVD asume que las relaciones entre las características son lineales, lo que puede limitar su capacidad para capturar relaciones no lineales
- **Inspiración VGAE** codifica la información colaborativa en el espacio latente, pues éste debe cumplir con la hipótesis de clustering. Esto debiese generar una mejor reconstrucción para la tarea de recomendación, en comparación con la SVD
- **Contribución:** Evaluar la utilidad de VGAE para reconstruir la matriz de interacción en el framework LightGCL



Avances

1. Análisis de datos
2. Revisión de código fuente y entrenamiento LightGCL

1. Análisis de datos

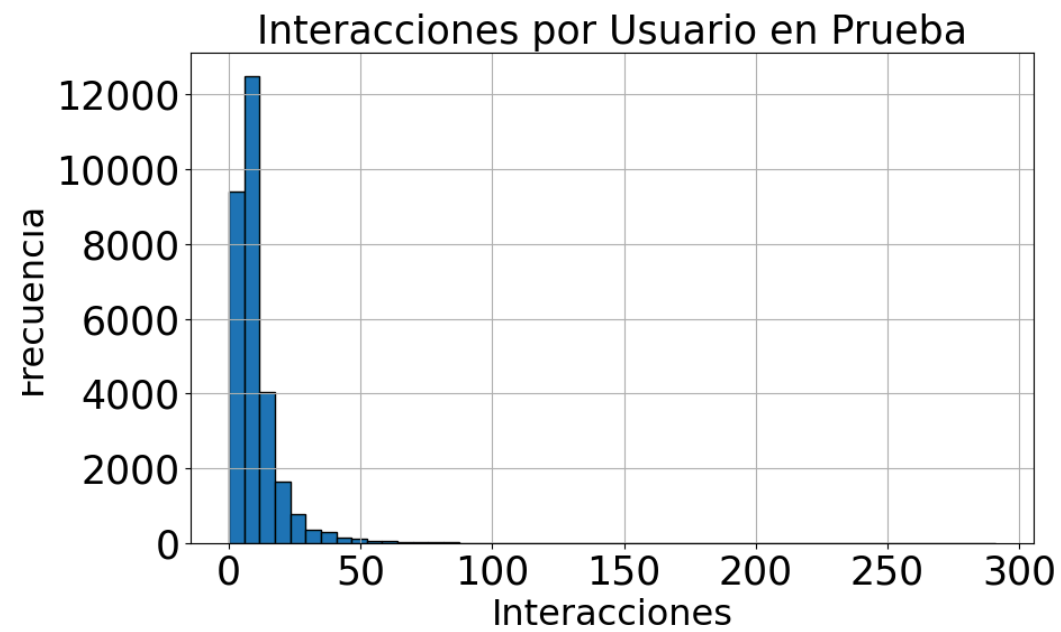
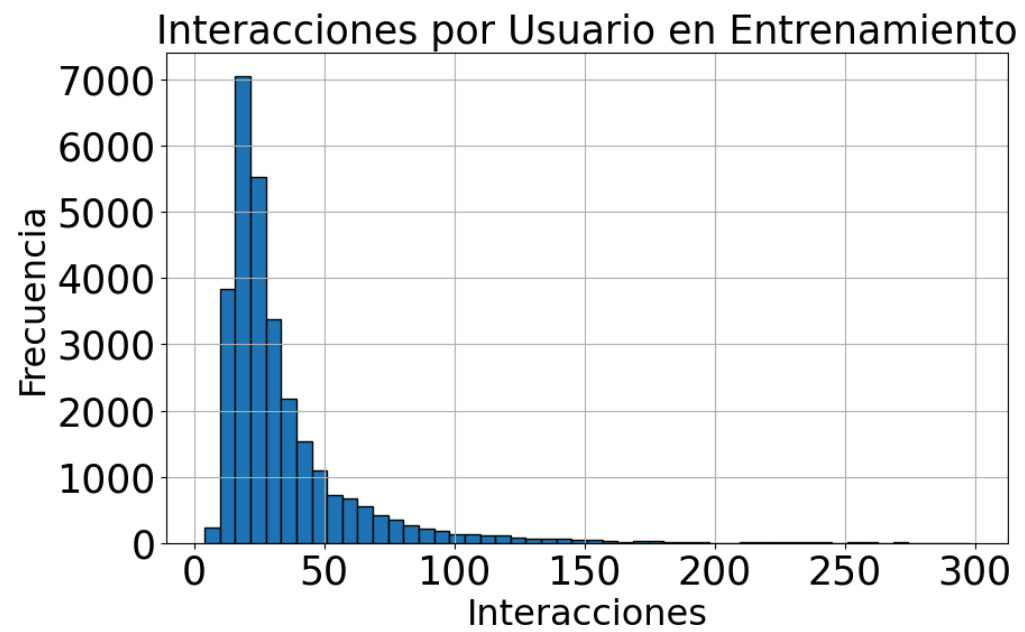
- LightGCL utiliza los siguientes *datasets*: **Yelp**, Gowalla, ML-10M, Amazon y Tmall.
- En este proyecto utilizaremos el conjunto de datos **Yelp**.
- Yelp es una plataforma donde usuarios califican y reseñan negocios locales.

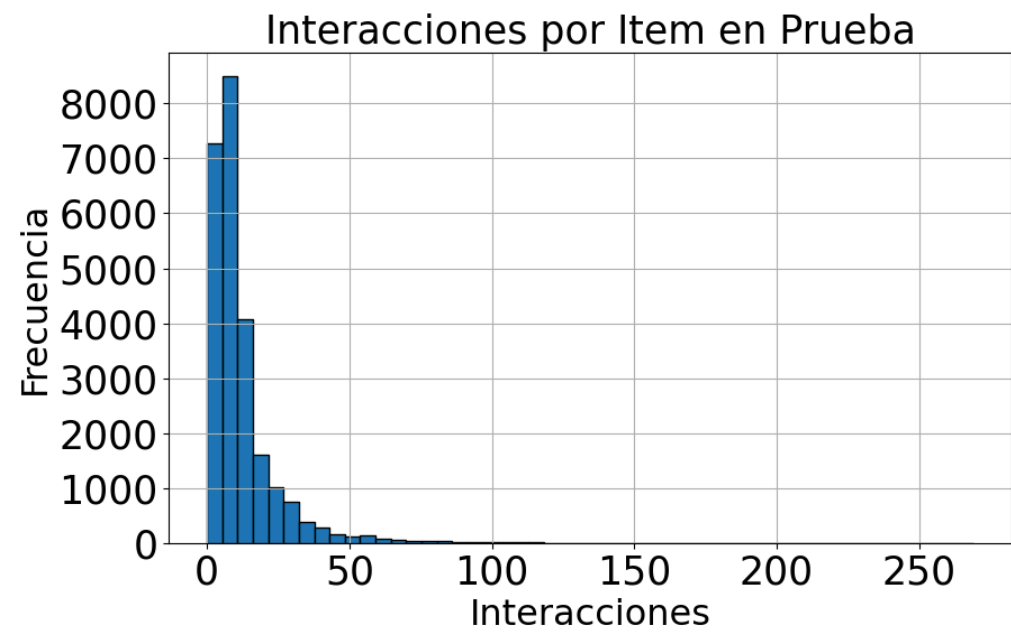
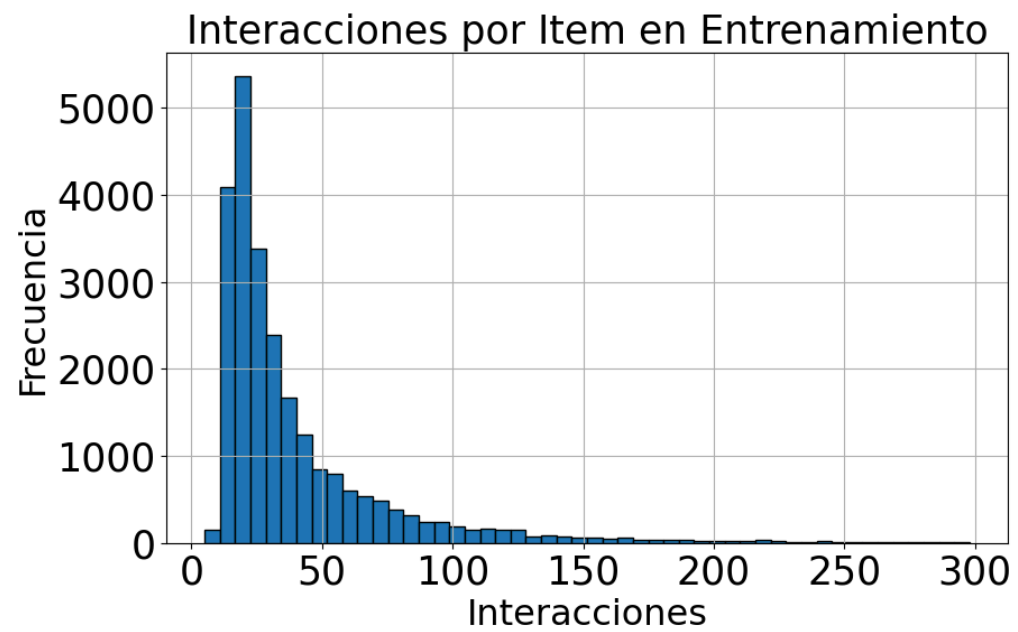
Métrica	Valor
Número de usuarios	29,601
Número de ítems	24,734
Número de interacciones	1,069,128
Promedio de interacciones por usuario	36.12
Promedio de interacciones por ítem	43.23
Máximo de interacciones por usuario	1,282
Máximo de interacciones por ítem	926
Densidad de la matriz	0.15%

Entrenamiento

Métrica	Valor
Número de usuarios	29,530
Número de ítems	24,679
Número de interacciones	305,466
Promedio de interacciones por usuario	10.32
Promedio de interacciones por ítem	12.35
Máximo de interacciones por usuario	370
Máximo de interacciones por ítem	269
Densidad de la matriz	0.04%

Prueba





2. Entrenamiento de LightGCL

```
# Install Python 3.9
sudo apt-get install python3.9
sudo update-alternatives --install /usr/bin/python3 python3 /usr/bin/python3.9 1

# Install pip
sudo apt-get install python3.9-distutils
wget https://bootstrap.pypa.io/get-pip.py
python3.9 get-pip.py

# Clone LightGCL
git clone https://github.com/HKUDS/LightGCL.git

# Install torch, numpy, tqdm
python -m pip install -r requirements.txt

# Start training
python main.py --data yelp
```

Epoch	Recall@20	Ndcg@20	Recall@40	Ndcg@40
1	0.058	0.049	0.096	0.063
25	0.096	0.082	0.15	0.10
50	0.099	0.085	0.16	0.11
100	0.10	0.085	0.16	0.11

Evaluación en conjunto de Prueba

Referencias

[1]: Cai, X., Huang, C., Xia, L., & Ren, X. (2023). LightGCL: Simple yet effective graph contrastive learning for recommendation. arXiv preprint arXiv:2302.08191.

[2]: Kipf, T. N., & Welling, M. (2016). Variational graph auto-encoders. arXiv preprint arXiv:1611.07308.