Improving Recommendation Diversity with Architectural Modifications to DGRec

João Pedro Oliveira Batisteli

January, 2025

Introduction

- We live in an era of information overflow, with data created every moment too large to digest in time.

[1] L. Yang *et al.*, **"Dgrec: Graph neural network for recommendation with diversified embedding generation,"** in *Proceedings of the sixteenth ACM international conference on web search and data mining*, 2023, pp. 661–669.
[2] Y. Zheng *et al.*, **"Dgcn: Diversified recommendation with graph convolutional networks,"** in *Proceedings of the Web Conference 2021*, 2021, pp. 401–412.

Introduction

- We live in an era of information overflow, with data created every moment too large to digest in time.
- Recommender systems are essential components of online platforms, navigating users through vast amounts of data to discover relevant items.

[1] L. Yang *et al.*, **"Dgrec: Graph neural network for recommendation with diversified embedding generation,"** in *Proceedings of the sixteenth ACM international conference on web search and data mining*, 2023, pp. 661–669.
[2] Y. Zheng *et al.*, **"Dgcn: Diversified recommendation with graph convolutional networks,"** in *Proceedings of the Web Conference 2021*, 2021, pp. 401–412.

Introduction

- We live in an era of information overflow, with data created every moment too large to digest in time.
- Recommender systems are essential components of online platforms, navigating users through vast amounts of data to discover relevant items.
- While maximizing accuracy has been a central concern in traditional recommender system research, studies have increasingly emphasized the importance of recommendation diversity[1][2]

[1] L. Yang *et al.*, **"Dgrec: Graph neural network for recommendation with diversified embedding generation,"** in *Proceedings of the sixteenth ACM international conference on web search and data mining*, 2023, pp. 661–669.

[2] Y. Zheng *et al.*, **"Dgcn: Diversified recommendation with graph convolutional networks,"** in *Proceedings of the Web Conference 2021*, 2021, pp. 401–412.

Introduction

- Graph Neural Networks are widely used across various fields due to their ability to process complex problems represented as graphs effectively.

Introduction

- Graph Neural Networks are widely used across various fields due to their ability to process complex problems represented as graphs effectively.
- Graph-based methods offer significant advantages, including the ability to represent users' historical interactions and diverse relationship types, leveraging the information encoded in their edges.

Diversified GNN-based Recommender System

To tackle the issue of limited recommendation diversity in current methods, the Diversified GNN-based Recommender System, Yang *et al.* [1] propose the **D**iversified **G**NN-based **Rec**ommender System (DGRec), which incorporates three key modules:

Diversified GNN-based Recommender System

To tackle the issue of limited recommendation diversity in current methods, the Diversified GNN-based Recommender System, Yang *et al.* [1] propose the **D**iversified **G**NN-based **Rec**ommender System (DGRec), which incorporates three key modules:

- **Submodular neighbor selection**: Identifies a diverse subset of neighbors for aggregation, enhancing representation diversity.

Diversified GNN-based Recommender System

To tackle the issue of limited recommendation diversity in current methods, the Diversified GNN-based Recommender System, Yang *et al.* [1] propose the **D**iversified **G**NN-based **Rec**ommender System (DGRec), which incorporates three key modules:

- **Submodular neighbor selection**: Identifies a diverse subset of neighbors for aggregation, enhancing representation diversity.
- **Layer attention**: Applies attention weights to the outputs of each GNN layer, mitigating over-smoothing and leveraging high-order connections.

Diversified GNN-based Recommender System

To tackle the issue of limited recommendation diversity in current methods, the Diversified GNN-based Recommender System, Yang *et al.* [1] propose the **D**iversified **G**NN-based **Rec**ommender System (DGRec), which incorporates three key modules:

- **Submodular neighbor selection**: Identifies a diverse subset of neighbors for aggregation, enhancing representation diversity.
- **Layer attention**: Applies attention weights to the outputs of each GNN layer, mitigating over-smoothing and leveraging high-order connections.
- **Loss reweighting**: Adjusts the learning process to emphasize items from long-tail categories, balancing accuracy and diversity.

Proposal

- In this work, we propose novel modifications to the DGRec architecture to further promote recommendation diversity.

Proposal

- In this work, we propose novel modifications to the DGRec architecture to further promote recommendation diversity.
- Specifically introducing a gated attention mechanism and additional transformations within the graph processing modules.

Methodology

Problem Statement

For diversified recommendation task, we have a set of users $U = u_1, u_2, ..., u_{|U|}$, a set of items $I = i_1, i_2, ..., i_{|I|}$, and a mapping function $C(\cdot)$ that maps each item to its category.

Problem Statement

For diversified recommendation task, we have a set of users $U = u_1, u_2, ..., u_{|U|}$, a set of items $I = i_1, i_2, ..., i_{|I|}$, and a mapping function $C(\cdot)$ that maps each item to its category.

The observed user-item interactions can be represented as an interaction matrix $\mathbf{R} \in \mathbb{R}^{|U| \times |I|}$, where $R_{ui} = 1$ if user u has interacted with item i, or $R_{ui} = 0$ otherwise.

Problem Statement

For diversified recommendation task, we have a set of users $U = u_1, u_2, ..., u_{|U|}$, a set of items $I = i_1, i_2, ..., i_{|I|}$, and a mapping function $C(\cdot)$ that maps each item to its category.

The observed user-item interactions can be represented as an interaction matrix $\mathbf{R} \in \mathbb{R}^{|U|\times|I|}$, where $R_{ui} = 1$ if user u has interacted with item i, or $R_{ui} = 0$ otherwise.

For a graph-based recommender model, the historical interactions are represented by a user-item bipartite graph $G = (U, I, E)$, where U and I are the disjoint sets of user and item nodes, and there is an edge $e_{ui} \in \mathcal{E}$ between u and i if $R_{ui} = 1$.

Training Framework

Embedding Layer is a look-up table that maps the user/item ID to a dense vector:

$$\mathbf{E}^{(0)} = \left(\mathbf{e}_1^0, \mathbf{e}_2^0, ..., \mathbf{e}_{|U|+|I|}^0\right)$$

where $e^{(0)} \in \mathbb{R}^d$ is the d-dimensional dense vector for user/item.

Training Framework

The convolution operation used in this work is Light Graph Convolution (LGC)[3].

[3] X. He *et al.*, **"Lightgcn: Simplifying and powering graph convolution network for recommendation,"** in *Proceedings of the 43rd International ACM SIGIR conference on research and development in Information Retrieval*, 2020, pp. 639–648.

Training Framework

The convolution operation used in this work is Light Graph Convolution (LGC)[3].

Each LGC layer generates an embedding vector for every user or item node.

[3] X. He *et al.*, **"Lightgcn: Simplifying and powering graph convolution network for recommendation,"** in *Proceedings of the 43rd International ACM SIGIR conference on research and development in Information Retrieval*, 2020, pp. 639–648.

Training Framework

The convolution operation used in this work is Light Graph Convolution (LGC)[3].

Each LGC layer generates an embedding vector for every user or item node.

$$\mathbf{e}_{\mathcal{T}}^{\ell+1} = \sum_{i \in \mathcal{N}_{\mathcal{T}}} \frac{1}{\sqrt{|\mathcal{N}_I|}\sqrt{|\mathcal{N}_U|}} \mathbf{e}_{\mathcal{T}}^{\ell} \tag{1}$$

where $\mathbf{e}_{\mathcal{T}}^{\ell}$ are the embedding at layer ℓ for nodes of type $\mathcal{T}$. $\mathcal{N}_I$ and $\mathcal{N}_U$ represent the neighborhoods of item and user nodes.

[3] X. He *et al.*, **"Lightgcn: Simplifying and powering graph convolution network for recommendation,"** in *Proceedings of the 43rd International ACM SIGIR conference on research and development in Information Retrieval*, 2020, pp. 639–648.

Training Framework

Each LGC layer produces an embedding vector for each user $(e_u^0, e_u^1, ..., e_u^\ell)$ and item node $(e_i^0, e_i^1, ..., e_i^\ell)$.

Training Framework

Each LGC layer produces an embedding vector for each user $(e_u^0, e_u^1, ..., e_u^\ell)$ and item node $(e_i^0, e_i^1, ..., e_i^\ell)$.

An attention mechanism is then applied to these embeddings to obtain the final user and item representations.

$$e_u = \text{Attention}\left(e_u^0, e_u^1, ..., e_u^L\right)$$

$$e_i = \text{Attention}\left(e_i^0, e_i^1, ..., e_i^L\right)$$

DGRec - Submodular Neighbor Selection

In GNN-based recommender systems, user/item embedding is obtained by aggregating information from all neighbors.

DGRec - Submodular Neighbor Selection

In GNN-based recommender systems, user/item embedding is obtained by aggregating information from all neighbors.

The submodular neighbor selection module aims to select a subset of neighbors that maximizes diversity and information content for aggregation.

DGRec - Submodular Neighbor Selection

In GNN-based recommender systems, user/item embedding is obtained by aggregating information from all neighbors.

The submodular neighbor selection module aims to select a subset of neighbors that maximizes diversity and information content for aggregation.

The objective is to find a subset $\mathcal{S}_u \subseteq \mathcal{N}_u$ that maximizes the sum of the maximum similarities between each item $i \in \mathcal{N}_u \setminus \mathcal{S}_u$ and the items in $\mathcal{S}_u$

DGRec - Loss Reweighting

Borrowing the idea of class-balanced loss, the DGRec reweighted the sample loss during training based on its category.

DGRec - Loss Reweighting

Borrowing the idea of class-balanced loss, the DGRec reweighted the sample loss during training based on its category.

Each sample (u, i) is reweighted based on the category number of items. This is achieved by incorporating a weight $w_{C(i)}$ into the loss function.

DGRec - Proposed Changes

1. Additional Transformations
2. Gated Attention

Additional Transformations

- Building upon the insights from[4] and[5], we employ a Graph Processing Module (GP Module) designed to enhance the diversity of node representations.

[4] K. Han *et al.*, **"Vision gnn: An image is worth graph of nodes,"** in *NeurIPS*, 2022.
[5] J. P. O. Batisteli *et al.*, **"Multi-scale image graph representation: A novel gnn approach for image classification through scale importance estimation,"** in *2023 IEEE International Symposium on Multimedia (ISM)*, 2023, pp. 62–68.

Additional Transformations

- Building upon the insights from[4] and[5], we employ a Graph Processing Module (GP Module) designed to enhance the diversity of node representations.
- This module incorporates linear and normalization layers before and after each graph convolution.

[4] K. Han *et al.*, **"Vision gnn: An image is worth graph of nodes,"** in *NeurIPS*, 2022.

[5] J. P. O. Batisteli *et al.*, **"Multi-scale image graph representation: A novel gnn approach for image classification through scale importance estimation,"** in *2023 IEEE International Symposium on Multimedia (ISM)*, 2023, pp. 62–68.

Additional Transformations

The original convolution Equation is modified to:

Additional Transformations

The original convolution Equation is modified to:

$$\mathbf{e}_{\mathcal{T}}^{\ell+1} = BN(W_2(BN(\sum_{i \in \mathcal{N}_{\mathcal{T}}} \frac{1}{\sqrt{|\mathcal{N}_I|}\sqrt{|\mathcal{N}_U|}} BN(W_1 \mathbf{e}_{\mathcal{T}}^{\ell})))) \tag{2}$$

where W_1 and W_2 are linear transformations, and BN stands for Batch Normalization.

Gated Attention

Aiming to aggregate information while maintaining feature diversity and ensuring robust representations for less frequent items, we adopted gated attention mechanics instead of the original Layer Attention.

Gated Attention

Aiming to aggregate information while maintaining feature diversity and ensuring robust representations for less frequent items, we adopted gated attention mechanics instead of the original Layer Attention.

This mechanism introduces learnable gates that modulate the flow of information, allowing the model to focus on relevant features and enhance the representation.

Experiments

Dataset

TaoBao[6]: contains user behavior on the TaoBao platform, encompassing various types of interactions such as clicking, purchasing, adding items to carts, and favoriting items.

[6] Y. Zheng *et al.*, **"Dgcn: Diversified recommendation with graph convolutional networks,"** in *Proceedings of the Web Conference 2021*, 2021, pp. 401–412.

Graph Construction

- The dataset is represented as a bipartite graph with user and item nodes.

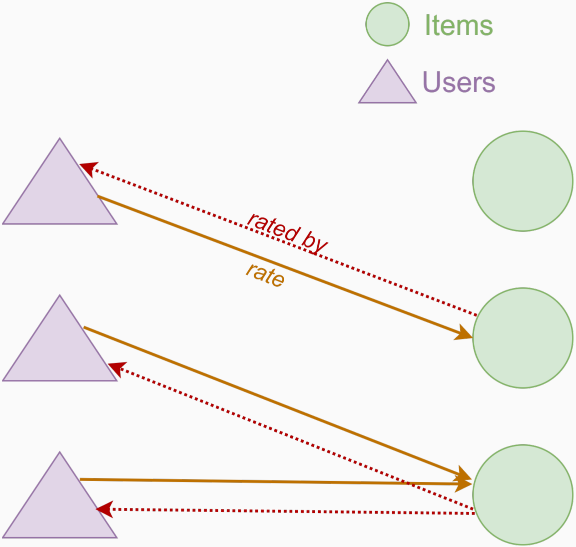
Graph Construction

- The dataset is represented as a bipartite graph with user and item nodes.
- All observed interactions are treated as positive samples, reflecting the direct connection between users and the items.

Graph Construction

- The dataset is represented as a bipartite graph with user and item nodes.
- All observed interactions are treated as positive samples, reflecting the direct connection between users and the items.
- The connections are represented by edges of type "*rated*" (user to item) and "*rated by*" (item to user).

Graph Construction



Metrics

Two metrics are used to evaluate accuracy, while a separate metric is employed to assess diversity.

Metrics

Two metrics are used to evaluate accuracy, while a separate metric is employed to assess diversity.

- Accuracy: Recall and Hit Ratio (HR).

Metrics

Two metrics are used to evaluate accuracy, while a separate metric is employed to assess diversity.

- Accuracy: Recall and Hit Ratio (HR).
- Diversity: Coverage.

Quantitative analysis

- Table 1 presents a comparative analysis of the original DGRec model [1] and our two proposed variants, DGRec_T and DGRec_A.

[7] L. Yang *et al.*, **"Dgrec: Graph neural network for recommendation with diversified embedding generation,"** in *Proceedings of the sixteenth ACM international conference on web search and data mining*, 2023, pp. 661–669.

Quantitative analysis

- Table 1 presents a comparative analysis of the original DGRec model [1] and our two proposed variants, DGRec_T and DGRec_A.

Method	Recall@100	Recall@300	HR@100	HR@300	Coverage@100	Coverage@300
DGRec[7]	**0.0455**	**0.0916**	**0.2946**	**0.4714**	39.5123	90.2758
DGRec_T	0.0204	0.0401	0.1569	0.2727	52.3410	**126.1431**
DGRec_A	0.0037	0.0116	0,0398	0.1075	**54.3406**	114.1998

[7] L. Yang *et al.*, **"Dgrec: Graph neural network for recommendation with diversified embedding generation,"** in *Proceedings of the sixteenth ACM international conference on web search and data mining*, 2023, pp. 661–669.

Conclusion

- This work introduces architectural modifications to the DGRec model [1] with the goal of enhancing the diversity of recommended items.

Conclusion

- This work introduces architectural modifications to the DGRec model [1] with the goal of enhancing the diversity of recommended items.
- Evaluation on the TaoBao dataset demonstrated that the proposed architectural changes effectively enhanced recommendation diversity.

Conclusion

- This work introduces architectural modifications to the DGRec model [1] with the goal of enhancing the diversity of recommended items.
- Evaluation on the TaoBao dataset demonstrated that the proposed architectural changes effectively enhanced recommendation diversity.
- This increase in diversity, however, was accompanied by a decrease in accuracy metrics.

Conclusion

- This work introduces architectural modifications to the DGRec model [1] with the goal of enhancing the diversity of recommended items.
- Evaluation on the TaoBao dataset demonstrated that the proposed architectural changes effectively enhanced recommendation diversity.
- This increase in diversity, however, was accompanied by a decrease in accuracy metrics.
- Future work will focus on exploring architectural modifications to achieve a better balance between the diversity and accuracy of recommendations, as well as utilizing multigraph representations to explicitly model various types of user-item relationships.

Thank you!