Information Fusion 107 (2024) 102343

Contents lists available at ScienceDirect

Information Fusion

journal homepage: www.elsevier.com/locate/inffus

A graph neural approach for group recommendation system based on pairwise preferences

Roza Abolghasemi [a,*], Enrique Herrera Viedma [b], Paal Engelstad [a], Youcef Djenouri [c,d,e], Anis Yazidi [a]

[a] Department of Computer Science, Oslo Metropolitan University (OsloMet), Oslo, Norway
[b] Andalusian Research Institute in Data Science and Computational Intelligence, University of Granada, Granada, Spain
[c] University of South-Eastern Norway, Kongsberg, Norway
[d] Norwegian Research Center, Oslo, Norway
[e] IDEAS NCBR, Warsaw, Poland

ARTICLE INFO

Keywords:
Graph clustering
Pairwise preferences
Recommendation systems
Group decision making
Group recommendation systems

ABSTRACT

Pairwise preference information, which involves users expressing their preferences by comparing items, plays a crucial role in decision-making and has recently found application in recommendation systems. In this study, we introduce GcPp, a clustering algorithm that leverages pairwise preference data to generate recommendations for user groups. Initially, we construct individual graphs for each user based on their pairwise preferences and utilize a graph convolutional network to predict similarities between all pairs of graphs. These predicted similarity scores form the foundation of our research. We then construct a new graph where users are nodes and the edges are weighted according to the predicted similarities. Finally, we perform clustering on the graph's nodes (users). By evaluating various metrics, we found that employing a similarity metric based on a convolutional neural network (SimGNN) with our proposed ground truth called Top-K yielded the highest accuracy. The proposed approach is specifically designed for group recommendation systems and holds significant potential for group decision-making problems. Code is available at https://github.com/RozaAbolghasemi/Group_Recommendation_Syatem_GcPp_clustering.

1. Introduction

With the dramatic expansion of the web of things, recommendation systems are nowadays prevalent in our daily lives [1–3]. Recommendation systems explore the historical data represented by user information and preferences to decide which items might be recommended for new observations. Numerous recommendation algorithms have been suggested in the last decade, however, a few of them have been successfully utilized in real settings [4–6]. However, we are far from handling the pairwise preferences problem, which is considered a key ingredient in modern recommendation systems [7,8]. In order to efficiently explore pairwise preferences, more intelligent methods need to be incorporated into the recommendation process. Clustering has paramount importance in the exploration and analysis of data, and it has extensive applications in data mining, information retrieval, decision making, computer graphics, bioinformatics, and Very-large-scale integration (VLSI) design. For an overview of clustering techniques and applications, we refer the reader to [9]. Usually, clustering consists of discovering natural groups of similar elements in datasets. However, since the similarity can be represented using graphs, graph clustering has gotten very attention as an important variant of data clustering. Moreover, to have a more accurate clustering method, using an accurate similarity measure is very crucial. In the learning process, a good graph similarity function/model would be able to capture all the hidden information of the data and predict the relationships between the data points to be able to predict more accurate similarity scores. Additionally, in some areas like group decision making (GDM) [10,11] and more specifically group recommendation systems (GRS) [12], where the pairwise preferences of the users are available, a possible solution to find common decision/recommendation for the group, is to first find similar users. The intuition behind this is that similar users will likely have similar preferences, and their decisions are likely to be more similar compared to dissimilar users.

This paper addresses the problem of group recommendation based on graph clustering. In the proposed graph clustering, for each node, some information like its pairwise preferences is available. Among the conventional graph similarity methods, those based on neural

* Corresponding author.
E-mail address: rozaabol@oslomet.no (R. Abolghasemi).

https://doi.org/10.1016/j.inffus.2024.102343
Received 15 November 2023; Received in revised form 29 February 2024; Accepted 1 March 2024
Available online 2 March 2024

networks were shown to be more accurate on different types of graphs. Traditional similarity measures like Euclidean and cosine, are based on the input vectors, and graph similarity measures like graph edit distance (GED) [13] are based on the structure of the graph. Unlike these traditional methods, the neural networks learn the similarity not only based on the structure of the graph but also the attributes of the nodes. This is very crucial and shows the power of this type of similarity method on different types of data/graphs.

It is noteworthy that our work presents a novel approach to evaluating user similarity in recommender systems by shifting away from traditional feature vector reliance to item-based pairwise preferences, employing the GcPp algorithm and SimGNN model. This shift is motivated by the limitations of conventional similarity measures based on feature vectors, as highlighted in previous research [14–16]. Specifically, measures such as PCC, SPCC, CPCC, ACOS, MSD, Jaccard, PIP, and NHSM are found to exhibit drawbacks in terms of accuracy and complexity when calculating user similarity. Additionally, the advantages of employing pairwise preferences over pure rating values are discussed in [17,18], emphasizing the potential for more precise and accurate recommendations. In [12,19,20], the incorporation of pairwise preferences in group recommendation systems is examined, introducing Matrix factorization pair-score prediction (MFP) and comparing it with well-known approaches like BPR [21], and MPR [22]. The analysis underscores the superiority of utilizing pairwise comparisons in GRS, particularly when compared to single-rating data.

The main contributions of this work are listed in the following:

1. We transform users' pairwise preferences into graphs and devise a graph convolutional neural network (SimGNN) to assess the similarity between each pair of user graphs. To accomplish this, we introduce a novel metric named *Top-K* to associate users with the top-k most similar items.
2. We develop the GcPp technique, an abbreviation for "Graph Clustering based on Pairwise Preference data". This method leverages the dominant set clustering algorithm to consolidate users with the highest degrees of similarity into cohesive groups. Specifically, we enhance the dominant set clustering algorithm by incorporating SimGNN to gauge user similarity.
3. We evaluate the developed model using two pairwise preferences datasets, and compare SimGNN with well-known similarity metrics. The results reveal the superiority of SimGNN compared to the baseline metrics.

In our final step, we capitalized on the clusters of users, which represent groups of individuals with similar preferences. Leveraging these clusters enabled us to generate recommendations that are not only more accurate but also fairer. By considering the preferences and characteristics of users within each cluster, we could tailor recommendations more precisely to the specific needs and interests of the users within those groups. This approach enhances the overall recommendation system by ensuring that recommendations are relevant and equitable for all users.

2. Related work

This study is divided into two primary areas of focus: graph clustering and recommendation and group recommendation. In the upcoming sections, we will delve into the existing body of research pertaining to both of these subjects.

2.1. Graph clustering and recommendation

Sieranoja et al. [23] suggested two novel clustering networks and graphs techniques. The first one is a direct descendant of the k-means algorithm and is known as the K-algorithm. Similar iterative local optimization is used, but the means are not necessary. In terms of both good local optimization capabilities and a propensity to reach a local optimum, it shares features with k-means clustering. The second one called M-algorithm, iteratively enhances the K-output algorithm's to uncover new and superior local optima. It repeatedly separates and merges arbitrary clusters, then uses the K-algorithm to fine-tune the outcomes. Both algorithms can be applied to various cost functions, which makes them both generic. By lowering information correlation in two ways, Liu et al. [24] presented the Dual Correlation Reduction Network (DCRN), a revolutionary self-supervised deep graph clustering technique. They specifically created a siamense network first to encrypt samples. Then, they reduced the information correlation at the dual-level, enhancing the discriminative power of the resulting features by pushing the cross-view sample correlation matrix and cross-view feature correlation matrix to resemble two identity matrices, respectively. Additionally, they applied a propagation regularization term to the network's shallow network structure to help the network gain long-distance information while alleviating representation collapse brought on by over-smoothing in GCN. Liao et al. [25] put forth a new deep linear graph attention model for attributed graph clustering (DLGAMC), which is made up of a similarity-preserving module and an attention-based aggregation module. To design the attention for aggregation, which does not require learning additional attention parameters, the authors only took advantage of cosine similarity. They also suggested an adaptive technique to assess the smoothness of node representations, with intra-cluster distance and inter-cluster distance serving as the essential indicators in this process.

To address the challenging issue in a non-Euclidean space, Guo et al. [26] proposed an original end-to-end graph clustering architecture with a combined strategy. They provide a new variational graph auto-encoder algorithm based on the GCN for learning the graph embedding that takes into consideration the boosting effect of a joint generative model of the graph structure and node characteristics on the embedding output. They developed an auxiliary distribution based on the embedding representation to provide a self-training mechanism that improved the prediction of node categories and enabled the unsupervised clustering mode. In order to avoid huge clusters warping the embedding space, each cluster's loss contribution is also normalized. Liu et al. [27] introduced a multilayer graph contrastive clustering network, which is a general and efficient autoencoder framework for multilayer graph clustering (MGCCN). Three modules make up the MGCCN: (1) attentiveness for better node embeddings, a mechanism that is used to better capture the importance of nodes and their neighbors. (2) A contrastive fusion approach that effectively investigated the consistent data in various networks. (3) A self-supervised element that reinforces the node embedding and clustering iteratively. A novel unsupervised event-oriented graph clustering framework (EGC) was proposed [28], which does not require labeled data and can perform efficient clustering on huge datasets with little time overhead. To be more precise, EGC changes the textual data of social networks after first mining the potential relations included in social text data. By utilizing graph structure for the depiction of complicated relations, media can be converted into an event-oriented graph. Secondly, EGC reliably measures the weights of relations in event-oriented graphs using a keyword-based local importance method.

One of the applications of clustering is group decision making which can be done easier using a good clustering method. In [29,30] the idea of dealing with large-scale decision-making problems is discussed. In [31], large-group decision-making and conflict management are addressed. The authors propose a dynamic adaptive subgroup-to-subgroup conflict model that focuses on multicriteria decision-making. They introduce a compatibility index to quantify cognitive and interest conflicts among experts and utilize the fuzzy c-means clustering algorithm to classify experts into subgroups. The paper [32] investigates the achievement of cluster average consensus within a finite time frame in bidirectional networks. Through the design of distributed linear iterations using stochastic matrices that align with the network topology, the authors demonstrate the possibility of always attaining cluster average consensus in bidirectional networks that have cluster-spanning

trees. A clustering- and maximum consensus-based model with linguistic distribution is proposed for social network large-scale group decision making (SNLGDM) problems [33]. The model utilizes social network analysis (SNA) to determine the weights of decision groups and reduce the dimension of large-scale decision matrices. It involves the division of large-scale decision makers (DMs) into independent sub-groups based on trust relationships [34]. Linguistic distribution (LD) assessments are used to represent the preference relation of sub-groups [35]. A maximum consensus-based method is then employed to generate comprehensive weights for the sub-groups by maximizing the level of consensus between sub-groups and the collective matrix [36]. The final ranking of alternatives is obtained based on the collective preference relation [37]. The proposed model is verified through numerical examples, coefficient analysis, and comparative analysis. The paper in [38] proposes an expert clustering and information fusion method for large group decision-making (LGDM) problems with double information and heterogeneous experts. It introduces an optimization model to derive criteria weights of experts based on the minimization of deviation between double information. It presents a double clustering method that combines the similarity degrees of experts' fuzzy preference relations and their criteria weights to classify large-scale experts into clusters. A clustering validity index is introduced to objectively determine the clustering threshold, ensuring the rationality of the clustering results.

In addition to clustering, graphs have been utilized in various ways to generate effective recommendations. In the work by Chen et al. [39] a method named SR-HetGNN is introduced that is a novel session recommendation method. SR-HetGNN leverages a heterogeneous graph neural network (HetGNN) to enhance session-based recommendation systems. Also, [40] introduces Mandari, a novel approach that leverages a Multi-Modal Temporal Knowledge Graph for Next-POI recommendation. It addresses the challenges of modeling implicit associations in multi-modal data and capturing variations in user preferences over time intervals. A method named Satori [41] has been introduced that leverages a user interaction graph to capture relationships between users, items, and categories and utilizes a graph attention network to extract auxiliary features. Additionally, it adopts a self-attention mechanism to model user intention and preference, combining them to form a hybrid user representation. The framework in [42] employs multitask training to optimize the model with auxiliary tasks. It aims to enhance cross-domain recommendation performance through adaptive learning and graph-based techniques. To see how deep learning models offer effective recommendations we refer the reader to [43,44]. The survey extensively examines recent studies on serendipitous recommendations, particularly concentrating on deep learning recommendation models. It categorizes these models based on their integration of serendipity objectives at different stages of the recommendation process.

2.2. Group recommendation

Utilizing pairwise preference data within Group Recommender Systems (GRS) significantly enhances the precision of item recommendations, as it provides detailed insights into users' preferences (see [10,12]). A method known as MFP, introduced by [12], predicts personalized item scores based on such pairwise preference data. In developing the GRS, the study incorporates users' personality traits, specifically assertiveness, and cooperativeness, which closely resemble real-world decision-making scenarios. Furthermore, the application of an opinion dynamics model aids in achieving consensus within the system. In [10], a novel approach grounded in entropy is presented for the prediction of missing data within pairwise preference datasets. This method excels by capitalizing on user and item similarity for prediction, even when confronted with exceptionally sparse datasets, surpassing traditional methods that may yield no results in such scenarios. The principal domain of application for this innovative concept lies within Group Recommender Systems and Group DecisionMaking challenges.

Deep learning plays a pivotal role in both Recommendation systems and Group Recommender Systems (GRS) methods, as evident in recent works [45–47]. In the context of GRS, an innovative approach employing a Graph Attention Network (GAT) is presented in [48]. This method initially clusters users based on movie genre preferences and user similarities. Subsequently, it employs GAT to predict users' movie ratings by considering their preferences and group relationships. Additionally, Wu et al. [49] propose a GRS tailored for network document resource exploration, leveraging knowledge graphs and LSTM within edge computing. This approach effectively addresses issues related to information overload and resource tracking. Ait et al. [50] introduced a distributed group recommendation system built on Apache Spark. This system employs a novel recommendation method, dimension reduction techniques, and supervised and unsupervised learning to address the curse of the dimensionality problem, identify user groups, and enhance prediction quality. Ali-Yari et al. [51] dealt with the concern related to the uncertainty and ambiguity surrounding a tourist's group membership in group tourism. They developed a group recommendation system that incorporates uncertainty modeling using Bayesian networks, Pearson similarity factors, and SOM clustering. It models uncertainties and estimates tourism preferences for each group, reducing the impact of insufficient user information. The system also suggests tourist attractions and optimal routes via Google Maps for each user group, enhancing the recommendation experience. In order to reduce the clustering cost in group recommendation, Seo et al. [52] introduced a GRS based on the genre preferences of users. They established a genre preference vector and employed it for group clustering, resulting in more efficient time complexity due to the smaller number of genres compared to items. Furthermore, they introduced a novel item preference mechanism, incorporating genre weights to further refine user preferences.

2.3. Discussions

In this section, we provide a summary of the limitations and strengths of the related works, which can be found in the Tables 1 and 2.

3. Method design

3.1. Principle

Within this section, we elucidate the proposed approach, which encompasses two main steps including GcPp clustering and group recommendation subsequent to data preprocessing. Fig. 1 indicates the logical diagram of the proposed method. During the data preprocessing phase, the data is transformed into a standardized format as follows: if expert u expresses a preference for item i over item j, the corresponding pairwise preference $p_{ij}^{(u)}$ is assigned a value of 1. Conversely, if the two items are considered equally preferred, the preference value $p_{ij}^{(u)}$ is set to 0.5. In cases where expert u does not prefer item i over item j, the pairwise preference value $p_{ij}^{(u)}$ is 0. In the subsequent sections, we provide comprehensive explanations of both the GcPp clustering approach and the proposed group recommendation system, delving into their intricacies and details.

3.2. Clustering

In this section, we present a detailed description of the GcPp (graph clustering pairwise preference) method. An overview of the proposed approach is depicted in Fig. 2, and the pseudo-code implementation can be found in Section 3.2.5. The method encompasses four main phases as follows:

- The dataset, comprising pairwise preferences of users, is transformed into graphs, as described in Section 3.2.1 and illustrated in Fig. 2 (part a).

Table 1
Merits and limits of the related works in graph clustering.

Paper	Merit	Limit
Sami et al. [23]	Introducing two cost functions and two clustering methods that can be applied with various cost functions. Notably, the M-algorithm outperforms eight other state-of-the-art methods.	The M-algorithm incurs a high computational cost because it involves the repetitive merging and splitting of random clusters.
Liu et al. [24]	The suggested DCRN aims to diminish information correlation at sample and feature levels, a crucial aspect in preventing representation collapse, ultimately leading to improved clustering outcomes.	The method's computational complexity and scalability for larger graphs require further exploration for practical use cases.
Liao et al. [25]	Decreasing the attention parameters within GCN, which involves incorporating a nonlinear attention mechanism that includes both attribute information similarity and the local graph structure.	The node pair selection process for the introduced similarity-preserved module requires improvement.
Guo et al. [26]	The paper's innovative end-to-end graph clustering approach effectively integrates embedding and clustering, enhancing the accuracy of unsupervised clustering tasks.	The paper does not account for unknown prior knowledge of clusters, leaving potential room for improvement in handling real-world applications where such information may be valuable.
Liu et al. [27]	The paper introduces MGCCN, an innovative autoencoder framework for multilayer graph clustering, surpassing existing methods in handling complex network frameworks, often limited to multiview attributes or multiple networks.	The MGCCN framework, while effective, may face scalability challenges with extremely large or high-dimensional datasets, requiring careful consideration for practical use.
Hu et al. [28]	EGC effectively converts social text data into an event-oriented graph, delivering high-quality clustering performance, rapid query times, and eliminating the need for labeled data, facilitating timely public opinion analysis on social media.	The paper may face challenges in optimizing the event-oriented graph generation for a more lightweight representation of social text data and improving the execution time of the graph clustering algorithm.
Morente et al. [29]	The paper introduces an innovative GDM method that effectively reduces information overload by clustering a large set of alternatives into manageable groups for expert discussions and decision-making.	This method's effectiveness may depend on the quality of clustering, and it may not fully address the challenges of handling extremely complex decision environments with diverse alternatives.
Ding et al. [30]	The paper defines and categorizes Large Scale Decision Making (LSDM) frameworks, offering insights into managing complex decision processes involving diverse stakeholders.	It primarily focuses on theoretical models and future research directions, potentially requiring practical implementation and validation of LSDM solutions for real-world applicability.
Tang et al. [31]	The paper introduces a dynamic conflict resolution model for large-scale group decision-making, enhancing our understanding of conflict management in complex decision contexts.	Further research is needed to assess the model's applicability in diverse settings like social networks and its adaptability to alternative opinion representation methods.
Shang et al. [32]	The paper provides insights into achieving finite-time cluster average consensus on bidirectional networks with cluster-spanning trees, offering valuable guidance for noise-free scenarios.	The study does not address directed networks or communication noises, common in practical applications, and further research is needed to adapt the results to these scenarios and explore delay robustness.

Table 2
Merits and limits of the related works in group recommendation systems.

Paper	Merit	Limit
Roza et al. [12]	Attaining highly precise and equitable GRS by leveraging detailed information from pairwise preference data, incorporating user personalities, and employing opinion dynamics to reach consensus.	For users, the process of completing the TKI test to unveil their personality traits can be time-consuming, and turning to user their social media content may present a feasible alternative.
Roza et al. [10]	An entropy-based method excels at estimating missing values in pairwise preference data for GRS and GDM, using user and item similarity, making it stand out from conventional methods that struggle with sparse datasets.	Like other methods, the overall model fitting, when minimizing the cross-entropy loss function, tends to produce larger errors as the problem's dimension, specifically the number of alternatives, increases.
Liao et al. [48]	Utilizing GAT (Graph Attention Network), clustering based on movie genre preferences and user similarities leads to significantly improved recommendation accuracy in GRS.	High time complexity due to the nature of the deep learning techniques.
Wu et al. [49]	By employing a knowledge graph and LSTM within edge computing for GRS, it effectively addresses the challenges of information overload and resource tracking.	High time complexity due to the nature of the deep learning techniques.
Badr et al. [50]	A distributed GRS, built on Apache Spark, capable of managing large-scale data and addressing sparsity issues.	The absence of social relationship interactions to boost the performance of the architectures.
Ali et al. [51]	Modeling uncertainties reduces the impact of limited user information and enhances the recommendation experience by suggesting tourist attractions and optimal routes through Google Maps.	The proposed SOM-based method is not compared with other clustering methods.
Seo et al. [52]	Reduced clustering cost and enhanced time efficiency are achieved by clustering based on user genre preferences rather than item ratings.	It appears that all movies (items) sharing the same genre are assigned an equal weight (score), which could pose challenges when recommending a subset of items from a pool with the same genre.

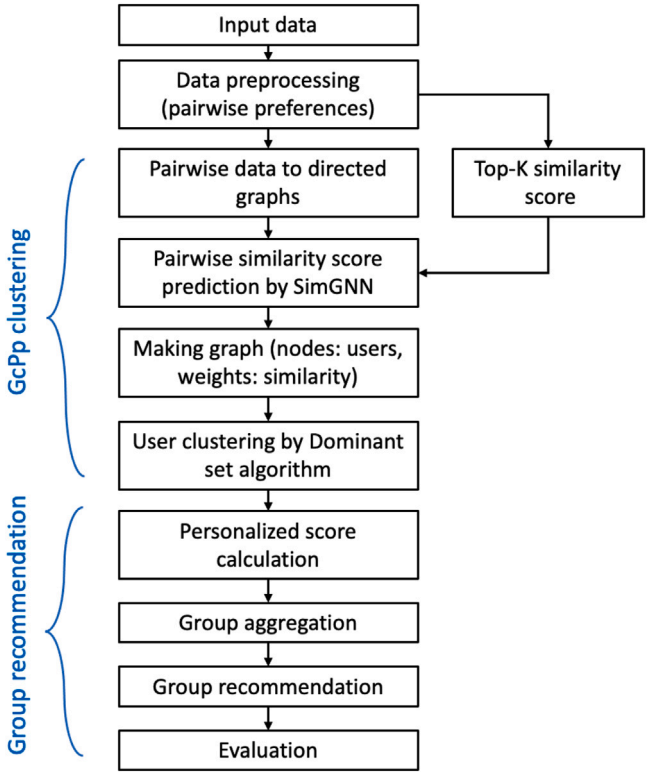

Fig. 1. Logical diagram of the process of the proposed method.

- The similarity between users (graphs of items) is predicted using the graph convolutional network called SimGNN [53], as depicted in Fig. 2 (part b) and described in Section 3.2.1.
- A graph is constructed with users represented as nodes, and the predicted similarity score between each pair of users is assigned as the weight of the corresponding edge, as shown in Fig. 2 (part c).
- The "Dominant-set" algorithm [54] is employed as a graph clustering method to group the users based on their relationships. This algorithm partitions the user graph into distinct clusters by identifying dominant sets of nodes. The process of clustering is depicted in Fig. 2 (part d), and you can find detailed explanations in Section 3.2.3.

In the final phase, we conducted an evaluation of our clustering approach using various metrics. In the subsequent sections, we will elaborate on the details of predicting similarity between pairs of graphs and the process of clustering a graph. This will provide a comprehensive understanding of how the similarity prediction and clustering steps were executed, allowing for a more in-depth analysis of our approach.

3.2.1. Similarity of graphs

In this section, we will outline our approach for predicting the similarity scores among users. Based on our datasets, which are described in Section 4.1, pairwise preferences on items are available for each user. This means that users have compared every pair of items and indicated their preferences. Specifically, we assign $p_{ij}^{(u)} = 1$ if user u prefers item i over item j, and $p_{ij}^{(u)} = 0$ otherwise. To capture these preferences, we convert the pairwise preferences into graphs for each user (refer to Fig. 2, part a). In these graphs, the items are represented as nodes, and a directed edge exists between two nodes if and only if $p_{ij}^{(u)} = 1$. Each graph represents the preferences of a specific user. We utilize these graphs to predict the similarity among users. Consequently, predicting the similarity between pairs of users is transformed into predicting the similarity scores between pairs of graphs. In this study, we adopt the concept of "a neural network approach to fast graph similarity computation (SimGNN)" introduced in [53]. An overview of this approach is depicted in Fig. 2 (part b). The SimGNN model employs a learnable embedding function based on graph convolutional networks (GCN) to generate an embedding vector for each graph, which serves as a condensed representation of the graph. Furthermore, the model incorporates a novel attention mechanism to highlight the significant nodes based on a specific similarity metric. This attention mechanism aims to emphasize the nodes that contribute most to the overall similarity computation. Additionally, a pairwise node comparison method is devised to enhance the graph-level embeddings by incorporating detailed node-level information. Finally, fully-connected neural networks are employed to predict the similarity scores. These networks leverage the graph-level embeddings and the fine-grained node-level information to estimate the similarity between pairs of graphs.

3.2.2. Top-K similarity

As SimGNN is a learning algorithm, it requires similarity scores for each pair of graphs as ground truth. To address this, we used the Top-k method which is introduced in this section. As previously mentioned, in our algorithm, each user corresponds to a graph. Thus, the similarity among users can be leveraged as the similarity between their corresponding graphs. To achieve this, we utilize the pairwise preferences of the users as follows: for each user, we arrange the items in descending order of preference based on personalized item scores calculated using the following equation:

$$v_i^{(u)} = \frac{\sum_{j \in I \setminus \{i\}} p_{ij}^{(u)}}{|I|}. \tag{1}$$

Here, $v_i^{(u)}$ represents the personalized item score of user u on item i. This score quantifies the degree of preference that user u has for item i relative to all other items in the set I (refer to [12] for more details). Using these personalized item scores, we sorted the items for each user and selected the top-k items with the highest scores as their most preferred items. Consequently, the Top-K similarity score between every pair of users is computed as follows:

$$S_{u_i,u_j}^{(Top-K)} = \frac{\left| u_i^{(Top-K)} \cap u_j^{(Top-K)} \right|}{K} \tag{2}$$

where u_i^{Top-K} represents the set of K-most preferred items for user u_i. The Top-K similarity scores, derived from the personalized item scores, serve as the ground truth for training the SimGNN model. By using these scores, the SimGNN model is incentivized to assign higher similarity scores to users whose top preferred items exhibit greater similarity. In other words, the training process encourages the SimGNN model to capture and reinforce the similarity patterns among users whose not only their pairwise preferences are similar, but also the top preferred items align closely with each other.

3.2.3. Clustering of graphs

Once we have predicted the similarity scores for pairs of users, our next step is to cluster the users based on these scores. The clustering method employed in this study is based on the concept of a "Dominant set". This approach, initially introduced by Pavan et al. [54], was later utilized by Hung et al. [55] for detecting F-formations in social environments, and by Yazidi et al. [56] for identifying unreliable sensors.

In the context of graph clustering, a dominant set refers to a maximal clique within an edge-weighted graph. By leveraging the dominant set algorithm, we aim to identify cohesive groups of users who exhibit similar preferences based on the obtained similarity scores. In the following, we will delve into the details of how the dominant set algorithm is applied for graph clustering.

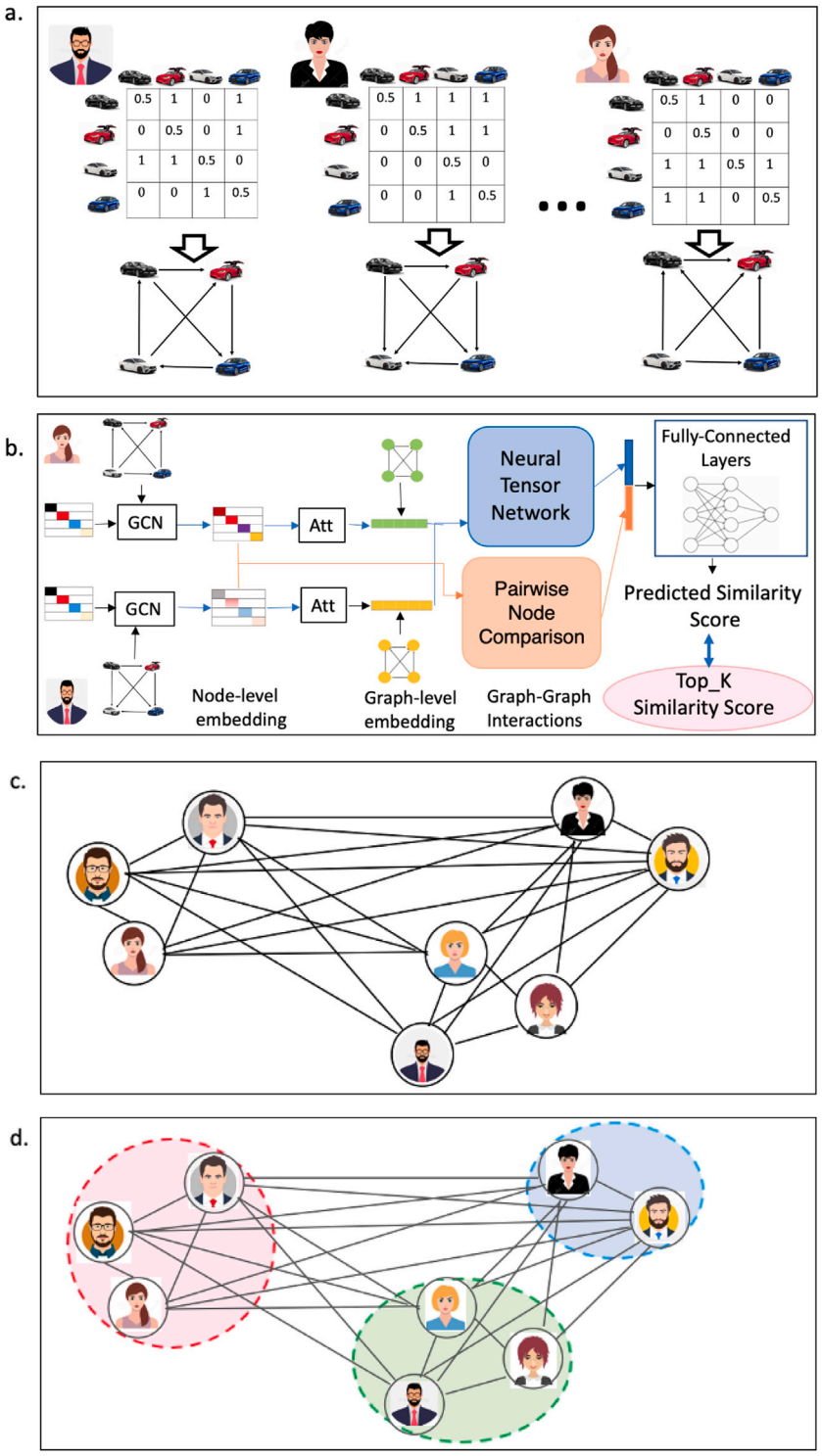

Fig. 2. General overview of the proposed GcPp method containing four main steps. a. converting the training data including pairwise preferences to graphs. b. predicting similarity between every pair of users (graphs) using SimGNN. c. graph of users, nodes are users, and edges are weighted by the similarity of pair of nodes. 4. clustering the users with Dominant set clustering.

We aim to cluster the data (users), using an undirected edge-weighted graph without self-loops. This graph, denoted as $G = (V, E, w)$, consists of a set of vertices V, a set of edges E, and a positive edge weight function w. In this context, the vertices correspond to the users, and the graph is assumed to be fully connected. However, the edges have varying weights that reflect the affinity between pairs of nodes.

To quantify the affinity, we utilize the predicted similarity scores between each pair of users (see Section 3.2.1) as the edge weights. This

relationship is captured by a weighted affinity matrix A, where each element a_{ij} represents the weight $w(i,j)$. When considering a subset S of vertices in graph G, we can assess the average weighted degree of a vertex $i \in S$ as follows:

$$k_S(i) = \frac{1}{|S|} \sum_{j \in S} a_{ij}. \tag{3}$$

The relative affinity between node $j \notin S$ and i can be expressed as follows:

$$\phi_S(i,j) = a_{ij} - k_S(i). \tag{4}$$

It should be noted that for all $i, j \in V$ if $i \neq j$, then $\phi_i(i,j) = a_{ij}$, and $\phi_S(i,j)$ can be either positive or negative. Ultimately, the weight of each node i with respect to a set S is defined recursively as follows:

$$w_S(i) = \begin{cases} 1, & if\ |S| = 1 \\ \sum_{j \in S \setminus \{i\}} \phi_{S\setminus\{i\}}(j,i) w_{S\setminus\{i\}}(j), & otherwise. \end{cases} \tag{5}$$

According to this definition, $w_S(i)$ represents the overall relative affinity (similarity) between node i and the other vertices in set S, weighted by the overall affinity of the vertices in S i. Thus, the following conditions establish the relationship between the internal nodes (in set S) and the external nodes (not in set S) of a dominant set S:

$$w_S(i) > 0, \forall i \in S \tag{6}$$

$$w_{S\cup\{i\}}(i) < 0, \forall i \notin S \tag{7}$$

3.2.4. Extracting dominating sets with replicator dynamics

The theory of replicator dynamics is attributed to the influential work of Taylor and Jonker [57] for modeling biological processes in population genetics. Replicator dynamics models a game of a population of competing agents where each agent possesses N strategies $a_1, a_2, \ldots, a_N$. The global state of the population is described by $x = (x_1(t), \ldots, x_N(t))$ with the constraint that $\sum_i x_i = 1$. For each $1 \leq i \leq N$, $x_i(t)$ denotes the proportion of the population using strategy a_i at time t. According to Taylor and Jonker [57], the agents encounter each other randomly. The random encounters yield a change in the fitness function according to a payoff matrix. The payoff matrix describes the utility A_{ij} of a player playing strategy i against a player playing strategy j. The replicator dynamics describes the evolution of the frequencies of strategies in a population. To each x_i we attach a payoff function f_i. f_i is given by $(Ax)_i$ which is the expected payoff of strategy a_i from a single interaction with a random individual in the large population. More precisely, the expected payoff f_i is written as $(Ax)_i = \sum_j a_{ij} x_j$, the average payoff of the population as a whole can be written as $x^T Ax = \sum_i \sum_j a_{ij} x_i x_j$. The frequency of each strategy is changed according to the difference between its expected payoff and the average payoff of the population. In formal terms, this is given by:

$$\frac{dx_i}{dt} = x_i(f_i - f) \tag{8}$$

The ratio of $\frac{dx_i}{dt}$ and x_i is called the relative growth success rate of the population. Based on the above expressions, we can present in more detail the replicator dynamics equation given by (Eq. (8)) as:

$$\frac{dx_i}{dt} = x_i(t)[(Ax(t))_i - x(t)^T Ax(t)], \quad i = 1, \ldots, N. \tag{9}$$

The theory of replicator dynamics has found a large set of applications in graph theory, including finding dominant sets [58,59], solving the maximum clique problem [60,61], reinforcement learning [62], resource allocation problems [63], to mention a few. We shall present a well-known theorem from the theory of replicator dynamics [57].

Theorem 1. *Let A be a nonnegative, real-valued symmetric $n \cdot n$ matrix. Then the function $x(t)^T Ax(t)$ increases with increasing t along any nonstationary trajectory $x(t)$ under continuous-time replicator dynamics. Furthermore, any such trajectory converges towards a stationary point x^*. Finally, a vector $x^* \in S_n$ is asymptotically stable if and only if x^* is a strict local maximizer of $x^T Ax$ in the simplex S_N given by $\sum_i^N x_i = 1$, $x_i \geq 0, i = 1, \ldots, N$.*

3.2.5. Pseudo code for GcPp method

Algorithm 1 GcPp

Input: $U = \{u_1, u_2, \ldots, u_n\}$: the set of n users where $u_u = \begin{bmatrix} p_{11}^{(u_n)} & \ldots & p_{1m}^{(u_n)} \\ \ldots & \ldots & \ldots \\ p_{m1}^{(u_n)} & \ldots & p_{mm}^{(u_n)} \end{bmatrix}$ is a matrix containing pairwise preference scores of user u on m items.
Output: $C = \{c_1, c_2, ..., c_T \mid c_t \subset U\}$: the set of all clusters of users.
for each u_i in U **do**
  $g_i \leftarrow graph(u_i)$;
end for
for every u_i and u_j in U **do**
  $S_{u_i,u_j}^{(Top-K)} \leftarrow$ Top-K(u_i, u_j);
  $w_{ij} \leftarrow SimGNN(g_i, g_j, S_{u_i,u_j}^{(Top-K)})$;
end for
{Graph G=(V, E, w), maximum number of clusters is T}
$V \leftarrow U$ {users as nodes of graph}
$C := \emptyset$
$t := 0$
while $V \neq \emptyset$ and $t < T$ **do**
  $c_t = Dominant\_set(G)$
  $C \leftarrow C \cup c_t$
  $V \leftarrow V \setminus c_t$
  $t \leftarrow t + 1$
end while
return C

A pseudocode for the proposed GcPp method is provided in Algorithm 1. This algorithm outlines the steps involved in clustering the users in U based on their pairwise preferences. In the algorithm, we start by converting the pairwise preference scores of each user u_i into a graph g_i (see Section 3.2.1). Then, we use the Top-K method to calculate similarity scores between pairs of users and incorporate them into SimGNN as inputs (see Sections 3.2.1 and 3.2.2). SimGNN predicts the similarity scores, which are then used as weights (w_{ij}) for the edges of a graph G, where the nodes V represent the users U. The algorithm proceeds with a while-loop, where in each iteration, the dominant set of nodes in G is determined and assigned to a new cluster. These nodes are then removed from the set of remaining nodes in G. The loop continues until the clustering algorithm cannot find any dominant sets among the remaining nodes. The pseudo-code provides a comprehensive representation of the GcPp method, showcasing the process of converting pairwise preferences to graphs, predicting similarity scores with SimGNN, constructing the weighted graph, and iteratively clustering the users based on dominant sets.

3.3. Group recommendation

In this section, we provide a detailed explanation of how we utilize the proposed GcPp clustering algorithm to offer effective recommendations for groups of users. The core concept of our proposed group recommendation system (GRS) is centered around the notion of maximizing fairness and accuracy when recommending items to a group of users. Unlike conventional GRS methods that often rely on random user selections within the groups, resulting in varying preferences and limited accuracy and fairness, we harness the power of our GcPp clustering algorithm to form groups of similar users. By leveraging the pairwise preferences and Top-K preferred items of users, we can establish a sense of similarity based on shared opinions regarding item scores. Consequently, the recommended items are expected to be appealing to the majority of group members, leading to higher levels of accuracy and fairness. Our proposed GRS comprises several key steps, namely: 1. personalized item score calculation, 2. group aggregation, and 3. group recommendation. Below, we provide a more comprehensive explanation of each step:

- In the initial stage, it is essential to determine the preferences of each user towards individual items. To achieve this, we calculate personalized item scores for every user using (1). However, if the dataset already includes user scores on items, this step can be skipped, and the existing scores can be directly utilized as personalized item scores.
- In this stage, we aim to determine the overall scores of items based on the preferences of users within each group. It is worth noting that our groups are created by clustering users using the GcPp method. Specifically, if $C = \{c_1, c_2, \ldots, c_T \mid c_t \subset U\}$ represents all the clusters obtained from the GcPp method, then $G = \{g_1, g_2, \ldots, g_T \mid g_t \subset U\}$ will represent all the corresponding groups where g_t is equal to c_t. Among various methods available for group aggregation, this paper focuses on two approaches: average and approval voting. The average group score for each item i in the group g_t denoted by $(v_i^{(g_t)})$ is calculated by taking the average of the personalized item scores (1) of all users u within that group. This provides a measure of the collective preference for each item within the group.

$$v_i^{(g_t)} = \frac{\sum_{u \in g_t} v_i^{(u)}}{|g_t|} \tag{10}$$

  In approval voting, which is a majority-based aggregation method [64], the group score for an item is determined by counting the number of users who approve of it. We use the following equation to calculate the group score for each item:

$$v_i^{(g_t)} = \frac{\sum_{u \in g_t} I_i^{(u)}}{|g_t|} \tag{11}$$

  Here, $I_i^{(u)} = 1$ if the individual user's score $v_i^{(u)}$ is above the threshold $(v_i^{(u)} > threshold)$, and $I_i^{(u)} = 0$ otherwise. In our experiments, we found that using a threshold of 0.4 produced better results.
  Based on the superior performance observed during the evaluation phase, we have chosen Average as the primary approach for aggregating group preferences in our system.
- During the group recommendation phase, we arrange the group scores $v_i^{(g_t)}$ for all items in descending order and provide recommendations to each group g_t based on the top-ranked items. To assess the performance of our group recommendation system, we evaluate it using various metrics including precision, recall, F1-score, and fairness.

4. Performance evaluation

4.1. Datasets

For this paper, we utilized two pairwise preference datasets to support our research and analysis. The details of these datasets are provided below:

- The food datasets used in this study were gathered through an online experiment called Consens@OsloMet, conducted at Oslo Metropolitan University in Norway. The primary objective of the experiment was to investigate how groups can achieve consensus or general agreement when presented with multiple food choices, including Chinese, French, Turkish, Italian, Japanese, and Mexican cuisines. The experiment was officially registered and approved by the Norwegian Centre for Research Data (NSD) with reference number 631862. The dataset consists of four trials, with five users providing their pairwise preferences for six different dishes in each trial. In this research, each user and experiment were treated as distinct individuals, resulting in a dataset that encompasses the pairwise preferences of 20 users across the six food items. For more details, please see section 3.2 in [10].

Table 3
Dataset characteristics.

Dataset	Users	Items	Comparisons	Sparsity
BookCrossing	271,379	278,858	5,893,374	99%
MovieLens 1M	6040	3952	104,931,478	92%
XING	784,687	1,029,480	180,601,043	99%
Movie online interface	46	100	2262	54%
Food	20	6	300	0%
Car	60	10	2700	0%

- The second dataset comprises car preferences[1] that were provided by Abbasnejad et al. [65] in 2013. The dataset was gathered from 60 users residing in the United States, who participated in the data collection process through Amazon's Mechanical Turk.[2] The dataset focuses on ten distinct cars, treated as individual items for comparison purposes. Each user in the dataset provided responses for all 45 possible pairs of items, resulting in a total of 90 observations for each expert. In addition to the pairwise preference scores, the dataset also includes two additional files containing users' attributes (education, age, gender, and region) and car attributes (body type, transmission, engine capacity, and fuel consumed). However, in our study, neither the users' attributes nor the cars' attributes were used during the training of the model.

Table 3 displays the characteristics of six real-world datasets utilized in recommendation systems. BookCrossing [66], MovieLens 1M,[3] and XING[4] encompass both implicit and explicit feedback from users regarding items, without incorporating pairwise preferences. Conversely, datasets such as Movie online interface [18], food and car datasets contain pairwise preferences expressed by users towards items. In their work [19] Kalloori et al. extracted both pairwise comparisons and ratings from the initial data of the first three datasets. For instance, the BookCrossing dataset comprises implicit (e.g., item clicks) and explicit user preferences on a 1–10 rating scale. Items with ratings exceeding 7 are deemed relevant, while the rest are considered irrelevant. Pairwise comparisons are derived by prioritizing items with implicit preferences over those deemed irrelevant. Similarly, in the XING dataset, interactions such as 'click', 'bookmark', 'apply', and 'delete' are recorded, with 'apply' interactions signifying relevance and 'delete' indicating irrelevance. Implicit preferences are inferred from 'click' and 'bookmark' actions, guiding the derivation of pairwise comparisons. For MovieLens 1M, items rated 4–5 stars are deemed relevant, while those rated 1–3 stars are considered irrelevant. All user ratings contribute to the derivation of pairwise comparisons, with differing ratings from the same user being subtracted.

Table 3 provides an overview of the datasets and their salient features. Notably, the sparsity metric is calculated as the ratio of missing elements to the total number of elements. Due to the substantial sparsity observed in the generated pairwise comparison datasets for all datasets except food and car, they are unsuitable for our method. Our method, which relies on SimGNN and GcPp, involves comparing user graphs, where nodes represent items and directed edges signify user preferences. The lack of pairwise comparisons makes it impossible to compare user graphs and compute user similarities. Therefore, the food and car datasets emerge as optimal choices for our proposed methodology.

[1] http://users.cecs.anu.edu.au/ u4940058/CarPreferences.html.
[2] http://mturk.com.
[3] https://grouplens.org/datasets/movielens/.
[4] 2https://github.com/recsyschallenge/2016.

4.2. Clustering performance

In this section, we provide a detailed explanation of the clustering evaluation methodology employed, the experiments conducted for the parameter optimization, and present the related results obtained for the GcPp clustering method and the SimGNN similarity score prediction. Please note that the evaluation and discussion related to the proposed Group Recommendation System (GRS) are presented in Section 4.3.

4.2.1. Metrics

According to an informal definition proposed by Jain et al. [67], "a cluster is a set of entities which are alike and entities from different clusters are not alike". Consequently, clustering approaches aim to achieve high internal homogeneity, which translates to high intra-cluster similarity (as shown in Eq. (12)), and at the same time, to maintain significant dissimilarity between entities within a cluster and those outside it, leading to low inter-cluster similarity (as expressed in Eq. (13)).

In line with this definition, we define the following equations. For object i within cluster c_I,

$$a(i) = \frac{1}{|c_I| - 1} \sum_{j \in c_I, i \neq j} d(i, j) \tag{12}$$

$a(i)$ represents the average distance between object i and the other objects within the same cluster (c_I), where $d(i, j)$ denotes the distance between objects i and j. A smaller value of $a(i)$ indicates a better assignment of the object to its respective cluster.

On the other hand, the dissimilarity of the object i to other clusters, such as c_J, is defined as the average distance between object i and all objects j within c_J (where $c_I \neq c_J$). For each object i, we calculate:

$$b(i) = \min_{J \neq I} \frac{1}{|c_J|} \sum_{j \in c_J} d(i, j) \tag{13}$$

defined as the minimum average distance of object i to all points in any other cluster to which i does not belong. By considering Eqs. (12) and (13), the *silhouette score* of an object i can be determined.

$$s(i) = \begin{cases} 1 - \frac{a(i)}{b(i)} & if\, a(i) < b(i) \\ 0 & if\, a(i) = b(i) \\ \frac{b(i)}{a(i)} - 1 & if\, a(i) > b(i) \end{cases} \tag{14}$$

From this equation, $-1 \leq s(i) \leq 1$, and a higher value of $s(i)$ indicates better clustering results.

4.2.2. Experiment and parameter setting (SimGNN)

In this section, we explain our conducted experiments to evaluate the GcPp clustering method.

Our first step was to determine the optimal parameters for the experiment. As mentioned earlier, In the process of clustering by GcPp, we utilized a similarity method called SimGNN, which is a graph convolutional neural network, to predict the similarity scores between pairs of graphs. However, since SimGNN is a learning-based method, we needed ground truth similarity scores for comparison. To facilitate this comparison, we obtained ground truth scores from two different similarity methods: graph edit distance (GED) [13] and the Top-K similarity that we introduced in this paper (see Section 3.2.2). In graph theory, GED calculates similarity based on the number of node and edge deletions and insertions required to transform one graph into another. This method emphasizes the shape of the graphs and its impact on their similarity. On the other hand, Top-K similarity compares the K-most preferred items of each pair of users, indicating that users are similar if their most preferred items are the same. We conducted the SimGNN experiment with 100 epochs using both Top-K and GED as ground truth, varying the training batch sizes (128, 256, and 512 for the car dataset, and 32, 64, and 128 for the food dataset). The corresponding results are shown in Fig. 3 for the car dataset and Figure Fig. 4 for the food dataset. Interestingly, the clustering based on the Top-K method achieved better test loss and exhibited lower validation and training loss compared to the clustering based on GED. This significant finding demonstrates that in calculating the users' similarities, incorporating valuable user information, such as their best items according to Top-K similarity, leads to improved similarity score predictions compared to GED, which solely relies on the graph structure. Regarding the batch sizes, we observed that a batch size of 128 and 256 performed better than 512 for the car dataset, while batch sizes of 32 and 64 were more effective for the food dataset. We reported the minimum validation and training loss as well as the test loss for all the experiments conducted on the car dataset in Table 4. The best results are highlighted in bold for easy reference.

4.2.3. Experiment and parameter setting (GcPp)

In the subsequent phase of the experiment, we employed the predicted similarity scores to cluster all of the users. Then, we evaluated our clustering with the silhouette score. As outlined in Section 4.2.1, the calculation of the silhouette score necessitates a distance metric. To this end, we utilized five distinct distance metrics, such as the Euclidean distance and cosine distance, which are computed based on the distance between the "feature vectors of the users". Notably, these feature vectors were not employed in the training of our proposed method. Additionally, we employed the "inverses of three similarity methods", namely SimGNN, GED, and Top-K, as three distance metrics. These three methods only used "similarity" between the "graphs of users" and not the "feature vectors of the users". The obtained silhouette scores based on these five distance metrics are presented in Table 5. Intriguingly, the inverse of SimGNN yielded the highest score in all experiments, irrespective of the batch sizes and ground truth methods. Even Top-K outperformed GED in terms of results. These remarkable findings underscore the effectiveness of our proposed method in clustering. There are two primary reasons for this: Firstly, we integrated the pairwise preferences of users, which provided valuable and detailed information about their preferences. This inclusion allowed us to capture the finer nuances of user preferences, leading to more accurate similarity predictions. Secondly, by using the Top-K similarity score as the ground truth, we incorporated the overall preferences of users in a broader sense. This approach allowed us to consider the general preferences and popular choices among users, resulting in a more comprehensive understanding of similarity. By combining these two approaches, we achieved improved performance in capturing and predicting similarity scores, resulting in better clustering. As anticipated, the Euclidean and cosine metrics did not perform as well as the other metrics. This can be attributed to the fact that we did not employ the feature vectors of users during the training process. Instead, training solely relied on the structure of graphs (users' pairwise preferences). In summary, our GcPp clustering method was implemented using the concept of "Dominant Set Clustering" and incorporated predicted similarity scores obtained from SimGNN, where the Top-K similarity score served as the ground truth. The construction of graphs was based on the pairwise preference data of the users. Our evaluation metrics demonstrated that the users were effectively clustered using this approach. Subsequently, we utilized these user clusters as groups in our Group Recommendation System (GRS) implementation. Through an ablation study, we optimized the system's parameters and evaluated its performance.

4.3. Group recommendation performance

In this section, we will elaborate on the evaluation methodology used for group recommendation. We will provide a comprehensive explanation of the experiments conducted to optimize the system's parameters and perform an ablation study. Furthermore, the results obtained from these experiments will be presented and discussed in detail.

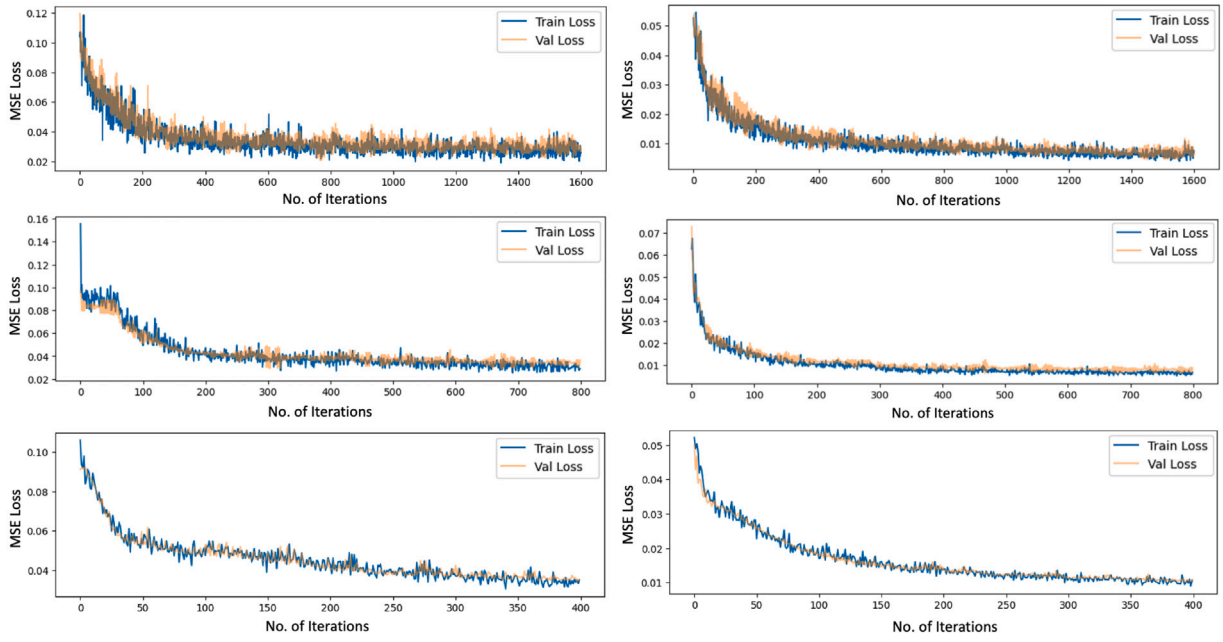

Fig. 3. Car dataset: Performance evaluation of SimGNN with 100 epochs and varying Training batch sizes (128, 256, 512). (Left) Ground truth: is Top-K, (Right) Ground truth is GED.

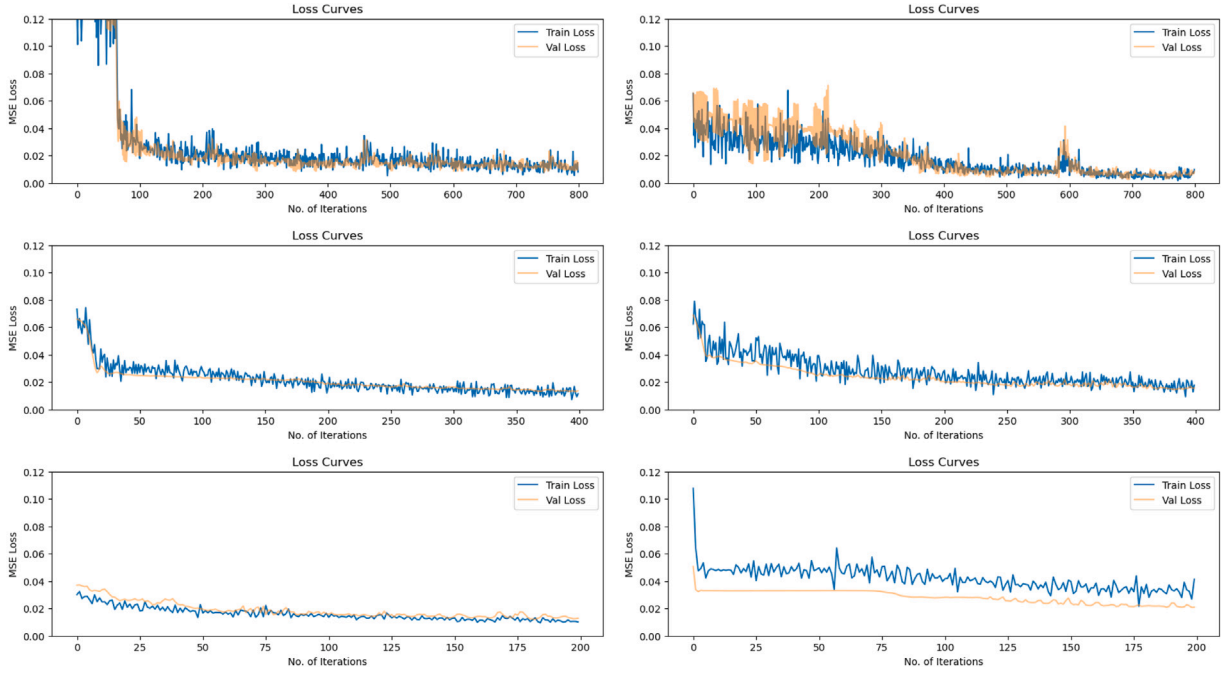

Fig. 4. Food dataset: Performance evaluation of SimGNN with 100 epochs and varying Training batch sizes (32, 64, 128). (Left) Ground truth: is Top-K, (Right) Ground truth is GED.

Table 4
SimGNN was implemented with various parameter configurations, including different ground truth methods (Top-K and GED) and batch sizes (128, 256, and 512). The minimum loss obtained for each setup is highlighted in bold.

	Ground truth: Top-K			Ground truth: GED		
	Train and validation batch			Train and validation batch		
	128	256	512	128	256	512
Min validation loss	**0.002**	**0.002**	0.003	0.004	0.006	0.010
Min train loss	**0.001**	0.002	0.003	0.004	0.005	0.009
Test loss	0.005	**0.004**	0.005	0.013	0.010	0.011

Table 5
Calculating Silhouette score for evaluation of our proposed GcPp clustering method by applying various parameter configurations, including different ground truth methods (Top-K and GED) and batch sizes (128, 256, and 512) and different distance metrics.

Distance metrics in Silhouette score		Ground truth: Top-K			Ground truth: GED		
		Train and validation batch			Train and validation batch		
		128	256	512	128	256	512
1	Euclidean distance	−0.332	−0.179	−0.240	−0.379	−0.411	−0.394
2	Cosine distance	−0.484	−0.504	−0.641	−0.616	−0.642	−0.553
3	Inverse of SimGNN	**−0.002**	**−0.003**	**−0.003**	**−0.019**	**−0.022**	**−0.021**
4	Inverse of GED	−0.026	−0.033	−0.032	−0.025	−0.023	−0.024
5	Inverse of Top-K	−0.006	−0.009	−0.008	−0.056	−0.048	−0.049

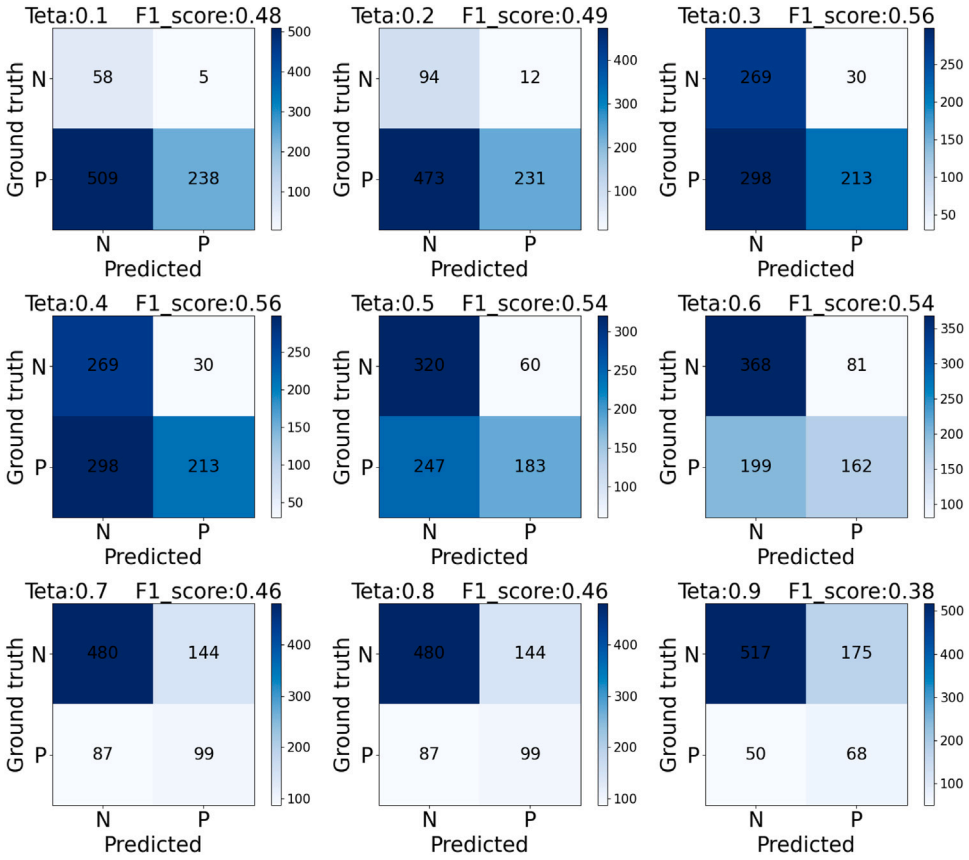

Fig. 5. Car dataset: Confusion matrices for different theta values, ranging from 0.1 to 0.9 (The letters N and P represent negative and positive, respectively).

4.3.1. Metrics

Our evaluation of the group recommendation model involves four key metrics: precision, recall, F1-score, and fairness. Below, I provide detailed explanations of how each of these metrics is calculated. We calculate TP (true positive), FP (false positive), and FN (false negative) based on the following equations (see section 5.2 in [12]):

$$TP_G = \{i \in R_G | \forall u \in G \quad \text{such that} \quad r_{u,i} \geq \theta\} \tag{15}$$

$$FP_G = \{i \in R_G | \exists u \in G \quad \text{such that} \quad r_{u,i} < \theta\} \tag{16}$$

$$FN_G = \{i \notin R_G | \forall u \in G \quad \text{such that} \quad r_{u,i} \geq \theta\} \tag{17}$$

Here, the set of items recommended to group G is denoted by R_G, while the rating of user u for item i is $r_{u,i}$. To measure whether a user likes or dislikes an item, we used a threshold θ. In the following, we describe the process of determining an appropriate value for θ.

4.3.2. Parameters setting

To determine the optimal threshold value for theta (θ), we conducted multiple iterations of the experiment using various theta values ranging from 0.1 to 0.9 for the car dataset and 0.0 to 0.8 for the food dataset.

As the heatmaps in Figs. 5 and 6 affirm, lower theta values are associated with higher values of TP and FN. Conversely, higher theta values result in higher TN and FP. Hence, a well-determined threshold is necessary to achieve high TP, TN, low FP, and FN, thereby ensuring high precision and recall. Based on our experiment, we selected a theta value of 0.4 for the car dataset and 0.1 for the food dataset. This implies that items with scores above this theta value are considered liked by the users, while those below, are deemed disliked.

4.3.3. Precision–recall curve

In order to demonstrate the effectiveness of utilizing GcPp clustering in the group recommendation system (GRS), we conducted two experiments: one with clustering and one without clustering. The precision–recall curve, depicted in Fig. 7, illustrates the evaluation outcomes at

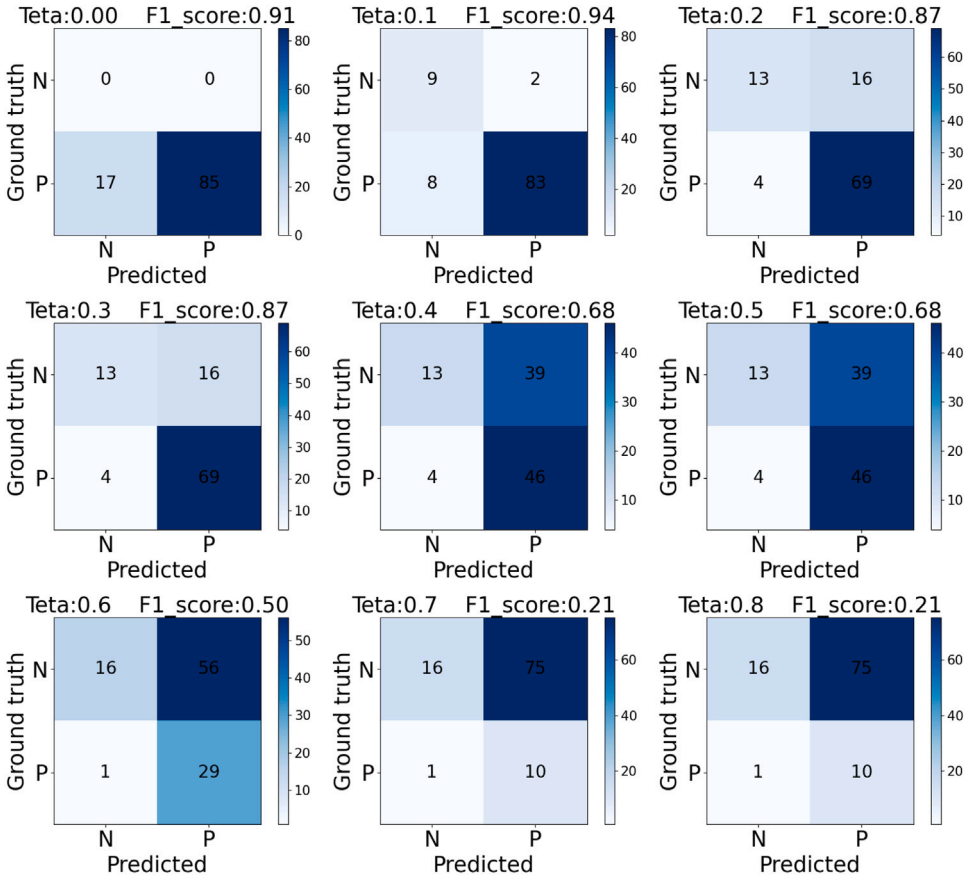

Fig. 6. Food dataset: Confusion matrices for different theta values, ranging from 0.1 to 0.9 (The letters N and P represent negative and positive, respectively).

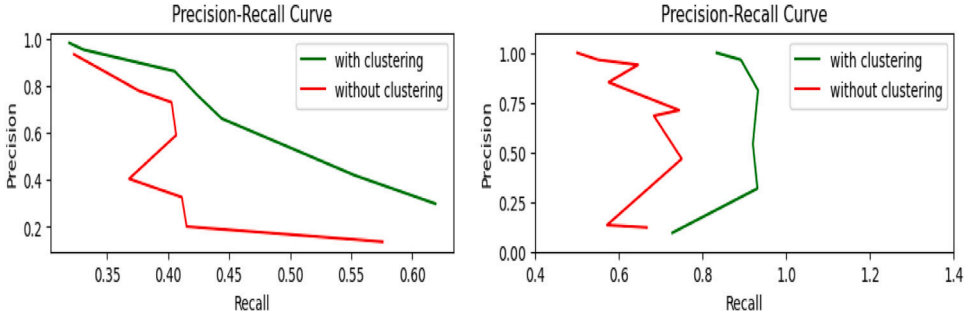

Fig. 7. Precision–recall curve for Group recommendation system with/without GcPp clustering on car dataset (left) and food dataset (right).

various thresholds (ranging from theta 0.1 to 0.9). This curve serves as evidence of the impact and performance of our group recommendation model with and without GcPp clustering. From these figures, it is observed that the curve for our model (with clustering) consistently lies above the curve for the model without clustering across all theta values. This indicates that our proposed model consistently achieves higher precision and recall compared to the other one. This means our model performs better in accurately identifying positive instances and minimizing false positives compared to the second model. This suggests that using GcPp clustering results more effectively in the task at hand, as it achieves a better balance between precision and recall. This comparison between the precision–recall curves of the two models highlights the superior performance of GRS with GCPp clustering.

4.3.4. Aggregation function setting

In the pursuit of identifying the most suitable aggregation function for our group recommendation System, we employed two aggregation methods: Approval Voting (AV) [68], which quantifies the number of ratings exceeding a specified threshold, and Average (Avg). These functions were applied to user groups derived from GcPp clustering and randomly assembled groups. The outcomes for the car dataset and food dataset on different thresholds (theta) are presented in Figs. 8 and 9, respectively.

Within the car dataset, Average exhibited a superior F1-score when contrasted with Approval Voting in both clustered and random user groups. Upon scrutinizing the left-hand figures (representing clustered groups) against the right-hand figures (depicting random groups), it becomes evident that GRS on groups clustered by GcPp demonstrates enhanced fairness and F1-score (a combination of precision and recall) compared to groups composed of random users. This observation underscores the advantageous impact of our proposed GcPp clustering approach within the context of GRS.

The summary of the parameter configuration of the proposed method based on the influence of various parameters on experimental

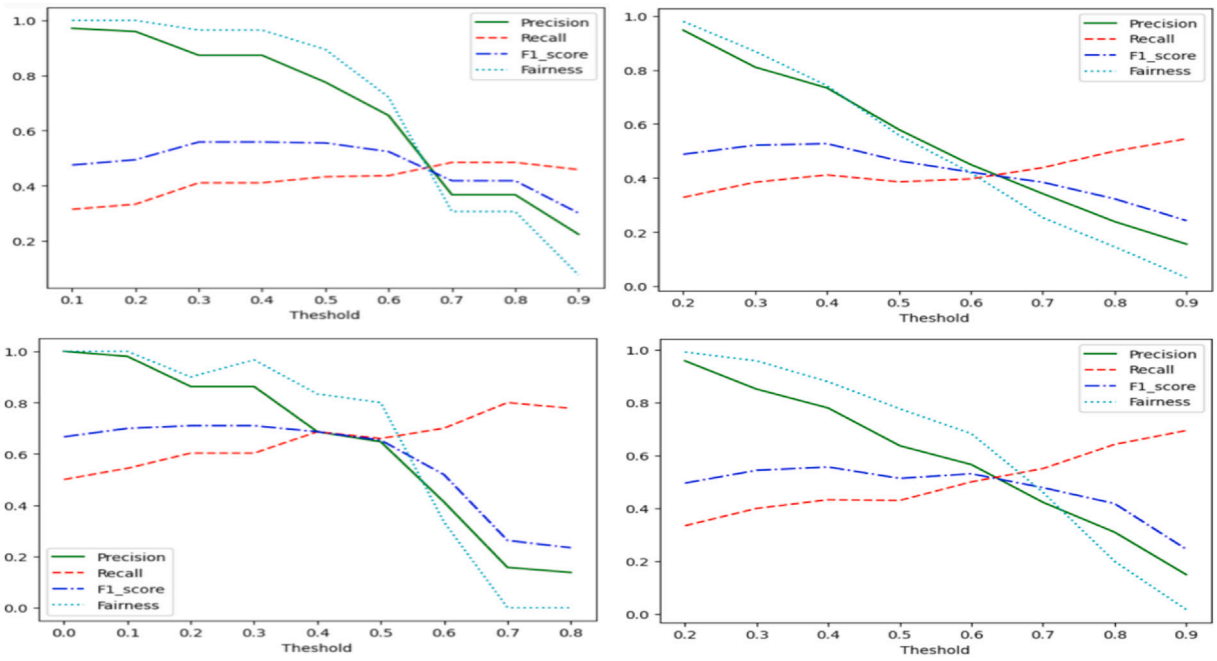

Fig. 8. Comparison of two aggregation functions: Approval Voting (upper panel) and Average (lower panel) applied to groups clustered by GcPp (left) and random groups (right), under varying theta values within the car dataset.

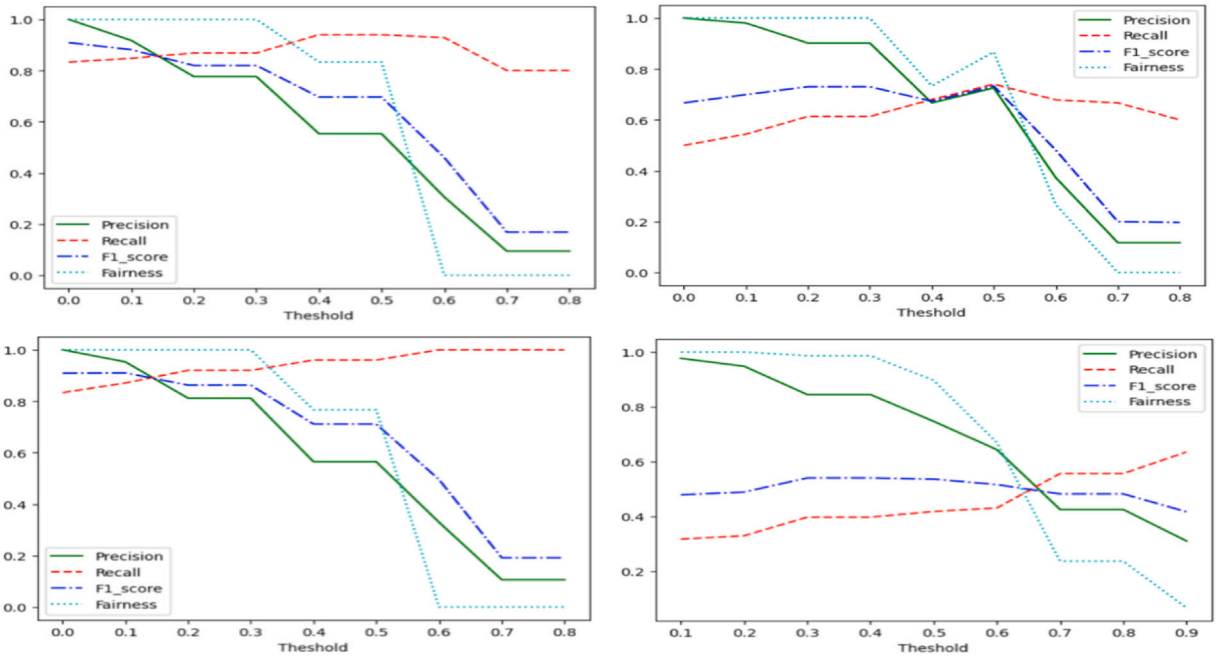

Fig. 9. Comparison of two aggregation functions: Approval Voting (upper panel) and Average (lower panel) applied to groups clustered by GcPp (left) and random groups (right), under varying theta values within the food dataset.

outcomes, comes in the following. According to Section 4.2.2, Table 4, and Figs. 3 and 4, the choice of "Top-K" as the ground truth for similarity in SimGNN is highlighted. Additionally, the train and validation batch sizes were set to 256 for the car dataset and 64 for the food dataset, driven by observations of lower test and validation losses. Furthermore, in accordance with Table 5 and the associated explanations in Section 4.2.3, "inverse of SimGNN" was selected as the distance metric for evaluating our clustering via the silhouette score. Having identified the optimal parameters for the clustering method, attention now shifts to leveraging these user clusters as groups for our group recommendation system. Towards this objective, as outlined in Section 4.3.4 and illustrated in Figs. 8 and 9, the aggregation function "Average" was chosen. Furthermore, guided by insights from Section 4.3.2 and depicted in Figs. 5 and 6, a theta value of 0.4 was adopted for the car dataset, while 0.1 was selected for the food dataset.

4.3.5. Comparison with state-of-the-art

In order to assess the effectiveness of our model, we conducted a comparative analysis with two widely recognized recommendation system methods, namely BPR [21] and MFP [12], which operate on pairwise-preference data. We evaluated various performance metrics, including precision, recall, F1-score, fairness, and execution time, to

Table 6
Comparison of evaluation results between our proposed GRS method with GcPp clustering and three alternative recommendation methods on the car dataset.

	Precision	Recall	F1-score	Fairness	Execution time
BPR	0.87	0.05	0.10	0.90	0.0061
MFP	0.78	0.30	0.43	0.82	0.0088
GRS without clustering	0.77	0.36	0.50	0.79	0.0088
GRS with GcPp clustering	**0.90**	**0.42**	**0.57**	**0.98**	**0.0077**

Table 7
Comparison of evaluation results between our proposed method with GcPp clustering and three alternative methods on the food dataset.

	Precision	Recall	F1-score	Fairness	Execution time
BPR	0.68	0.74	0.71	0.70	0.0087
MFP	0.71	0.73	0.72	0.78	0.0093
GRS without clustering	0.92	0.50	0.65	**1.00**	0.0016
GRS with GcPp clustering	**0.94**	**0.85**	**0.89**	**1.00**	**0.0014**

gauge the quality and efficiency of our model in relation to these established methods. Moreover, we compared our method with the proposed GRS without using GcPp clustering. The obtained results are displayed in Tables 6 and 7.

The results obtained from our evaluation demonstrate the superior performance of our proposed method over the other methods across all evaluation metrics. These findings highlight the significance of employing GcPp clustering in group recommendation systems. The high precision, recall, F1-score, and fairness achieved by our proposed method, which utilizes GcPp clustering, compared to BPR and MFP in the group recommendation system can be attributed to several key factors. Firstly, the incorporation of GcPp clustering in our proposed method allows for a more accurate and effective grouping of individuals with similar preferences. By clustering individuals based on their preferences, our method can capture the underlying patterns and similarities within the group more effectively. This clustering process enables us to create more homogeneous groups, where the members share similar tastes and preferences. As a result, our proposed method can generate recommendations that are more aligned with the individual preferences of the group members, leading to higher precision and recall compared to BPR and MFP. Furthermore, GcPp clustering facilitates targeted and precise recommendation aggregation within each cluster. By considering the preferences of the clustered group members, our method can identify items that are highly preferred by a significant portion of the group. This targeted aggregation enhances the relevance and accuracy of the recommendations, resulting in higher precision and recall scores. In terms of fairness, GcPp clustering helps to address the fairness concerns in group recommendations. By forming clusters based on similarity, our method ensures that individuals with similar preferences are grouped together. This helps in avoiding situations where certain individuals dominate the recommendation process, ensuring a fair distribution of recommendations among the group members. The recommendations provided by our proposed method take into account the preferences and interests of the entire group, promoting fairness in the recommendation outcomes. Moreover, GcPp clustering helps to mitigate the influence of personal biases or outliers within the group. By grouping individuals with similar preferences, our method can reduce the impact of extreme or outlier preferences, resulting in a more balanced and fair recommendation process. This contributes to higher fairness scores for our proposed method compared to BPR and MFP.

To assess the efficacy of the proposed GcPp clustering approach on Group Recommendation Systems (GRS), we employed Adversarial Preference Learning with Pairwise Comparisons (CRGAN) [69] as a recommendation technique suited for pairwise comparison data. For group recommendation, user groups were formed using two methods: firstly, by leveraging the GcPp clustering outcomes where each cluster represented a user group; secondly, by generating random groups of equivalent sizes to the clustered groups but composed of random users. To ensure impartiality, the process of generating random groups was repeated 20 times, and the final evaluation results represent the average performance across these iterations.

The evaluation encompassed 10 metrics, including HitRatio (HR), Normalized Discounted Cumulative Gain (NDCG), Area under the ROC Curve (AUC), Mean Average Precision (MAP), Mean Reciprocal Rank (MRR), Accuracy, Precision, Recall, F1-score, fairness, and also execution time. Evaluation results are presented in Table 8. Notably, CRGAN exhibited superior performance across all metrics when employing GcPp for group formation compared to random grouping, particularly excelling in precision, recall, F1-score, and fairness, pivotal evaluation criteria for group recommendation systems. This superiority stems from the clustering algorithm's ability to categorize users based on detailed (pairwise comparison) and top-K preferred items, resulting in similar preferences within clustered groups. Specifically, when group members share similar preferences, precision increases as a larger proportion of recommended items align closely with group preferences. Conversely, diversity within random groups poses challenges in matching items to varying preferences, resulting in lower precision. Additionally, in clustered groups, recommendations cater to common interests, enhancing recall by capturing relevant items more effectively. In contrast, random groups with diverse preferences struggle to identify universally relevant items, leading to lower recall. The F1-score reflects a balanced performance between recommendation accuracy and completeness. Fairness is also higher in clustered groups due to recommendations aligning with homogeneous group preferences, enhancing the satisfaction of each user and equitable representation compared to random groups with diverse preferences.

5. Conclusion

This paper presents a group recommendation approach using the introduced graph clustering method called GcPp, which relies on users' pairwise preferences. Initially, a neural network-based similarity score prediction process is employed, where the top-k preferred items act as the ground truth, determining the similarity scores between nodes (users) based on their favorite items and pairwise preference similarities. The integration of GcPp clustering enhances precision, recall, F1-score, and fairness in the group recommendation system. The clustering process enables accurate grouping and targeted recommendation aggregation, promoting fairness by considering the entire group's preferences and mitigating the impact of personal biases or outliers. Consequently, our proposed method outperforms BPR and MFP (widely used recommendation system methods based on pairwise preferences) in terms of precision, recall, F1-score, and fairness. In future work, we plan to explore GcPp in other applications such as intelligent transportation [70,71] or agent-based communication [72,73] to test the usability of the method in real settings.

Ethical and informed consent for data used

No ethical issue for data used.

CRediT authorship contribution statement

Roza Abolghasemi: Writing – review & editing, Writing – original draft, Visualization, Validation, Software, Conceptualization. **Enrique Herrera Viedma:** Writing – review & editing, Methodology, Conceptualization. **Paal Engelstad:** Writing – review & editing, Supervision, Conceptualization. **Youcef Djenouri:** Writing – review & editing, Supervision, Methodology, Formal analysis, Conceptualization. **Anis Yazidi:** Writing – review & editing, Supervision, Project administration, Methodology, Formal analysis, Conceptualization.

Table 8
Comparing assessment outcomes of the GRS through the utilization of CRGAN with GcPp clustering against random clusters across car and food datasets.

Dataset	Food Dataset		Car Dataset	
Method	CRGAN (random groups)	CRGAN (clustered groups)	CRGAN (random groups)	CRGAN (clustered groups)
HR@3	**1.00**	**1.00**	**1.00**	**1.00**
NDCG@3	0.77	**0.85**	0.89	**1.00**
AUC@3	0.70	**0.75**	0.87	**1.00**
MAP@3	0.85	**0.90**	0.91	**1.00**
MRR@3	0.47	**0.51**	0.42	**0.44**
Accuracy	0.71	**0.83**	0.85	**0.97**
Precision	0.78	**0.89**	0.88	**1.00**
Recall	0.77	**0.92**	0.84	**0.94**
F1-score	0.76	**0.89**	0.86	**0.97**
Fairness	0.63	**0.68**	0.68	**0.70**
Execution time	0.0114	0.0130	0.0116	0.0130

Declaration of competing interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Data availability

Data will be made available on request.

References

[1] J. Lin, M. He, W. Pan, Z. Ming, Collaborative filtering with sequential implicit feedback via learning users' preferences over item-sets, Inform. Sci. 621 (2023) 136–155.

[2] L. Zeng, J. Guan, B. Chen, MSBPR: A multi-pairwise preference and similarity based Bayesian personalized ranking method for recommendation, Knowl.-Based Syst. 260 (2023) 110165.

[3] Y. Xu, E. Wang, Y. Yang, Y. Chang, A unified collaborative representation learning for neural-network based recommender systems, IEEE Trans. Knowl. Data Eng. 34 (11) (2021) 5126–5139.

[4] E. Amigó, Y. Deldjoo, S. Mizzaro, A. Bellogín, A unifying and general account of fairness measurement in recommender systems, Inf. Process. Manage. 60 (1) (2023) 103115.

[5] B. Walek, P. Fajmon, A hybrid recommender system for an online store using a fuzzy expert system, Expert Syst. Appl. 212 (2023) 118565.

[6] X. Shen, H. Jiang, D. Liu, K. Yang, F. Deng, J.C. Lui, J. Liu, S. Dustdar, J. Luo, PupilRec: Leveraging pupil morphology for recommending on smartphones, IEEE Internet Things J. 9 (17) (2022) 15538–15553.

[7] X. Chen, Y. Zhang, I.W. Tsang, Y. Pan, J. Su, Toward equivalent transformation of user preferences in cross domain recommendation, ACM Trans. Inf. Syst. 41 (1) (2023) 1–31.

[8] W. Liu, X. Zheng, J. Su, L. Zheng, C. Chen, M. Hu, Contrastive proxy kernel stein path alignment for cross-domain cold-start recommendation, IEEE Trans. Knowl. Data Eng. (2023).

[9] A. Jain, M. Murty, P. Flynn, Data clustering: a review, ACM computing survey, Journal 31 (3) (1999).

[10] R. Abolghasemi, R. Khadka, P.G. Lind, P. Engelstad, E.H. Viedma, A. Yazidi, Predicting missing pairwise preferences from similarity features in group decision making, Knowl.-Based Syst. 256 (2022) 109860.

[11] A. Yazidi, M. Ivanovska, F.M. Zennaro, P.G. Lind, E.H. Viedma, A new decision making model based on rank centrality for GDM with fuzzy preference relations, European J. Oper. Res. 297 (3) (2022) 1030–1041.

[12] R. Abolghasemi, P. Engelstad, E. Herrera-Viedma, A. Yazidi, A personality-aware group recommendation system based on pairwise preferences, Inform. Sci. 595 (2022) 1–17.

[13] A. Sanfeliu, K.-S. Fu, A distance measure between attributed relational graphs for pattern recognition, IEEE Trans. Syst. Man Cybern. (1983) 353–362.

[14] A. Gazdar, L. Hidri, A new similarity measure for collaborative filtering based recommender systems, Knowl.-Based Syst. 188 (2020) 105058.

[15] B.K. Patra, R. Launonen, V. Ollikainen, S. Nandi, A new similarity measure using bhattacharyya coefficient for collaborative filtering in sparse data, Knowl.-Based Syst. 82 (2015) 163–177.

[16] T. Mahara, et al., A new similarity measure based on mean measure of divergence for collaborative filtering in sparse environment, Procedia Comput. Sci. 89 (2016) 450–456.

[17] N. Jones, A. Brun, A. Boyer, A. Hamad, An exploratory work in using comparisons instead of ratings, in: E-Commerce and Web Technologies - 12th International Conference, in: Lecture Notes in Business Information Processing, vol. 85, Springer, 2011, pp. 184–195, http://dx.doi.org/10.1007/978-3-642-23014-1_16.

[18] L. Blédaité, F. Ricci, Pairwise preferences elicitation and exploitation for conversational collaborative filtering, in: Proceedings of the 26th ACM Conference on Hypertext & Social Media, 2015, pp. 231–236.

[19] S. Kalloori, T. Li, F. Ricci, Item recommendation by combining relative and absolute feedback data, in: Proceedings of the 42nd International ACM SIGIR Conference on Research and Development in Information Retrieval, 2019, pp. 933–936.

[20] S. Kalloori, F. Ricci, M. Tkalcic, Pairwise preferences based matrix factorization and nearest neighbor recommendation techniques, in: Proceedings of the 10th ACM Conference on Recommender Systems, 2016, pp. 143–146.

[21] S. Rendle, C. Freudenthaler, Z. Gantner, L. Schmidt-Thieme, BPR: Bayesian personalized ranking from implicit feedback, 2012, arXiv preprint arXiv:1205.2618.

[22] R. Yu, Y. Zhang, Y. Ye, L. Wu, C. Wang, Q. Liu, E. Chen, Multiple pairwise ranking with implicit feedback, in: Proceedings of the 27th ACM International Conference on Information and Knowledge Management, 2018, pp. 1727–1730.

[23] S. Sieranoja, P. Fränti, Adapting k-means for graph clustering, Knowl. Inf. Syst. 64 (1) (2022) 115–142.

[24] Y. Liu, W. Tu, S. Zhou, X. Liu, L. Song, X. Yang, E. Zhu, Deep graph clustering via dual correlation reduction, in: Proceedings of the AAAI Conference on Artificial Intelligence, Vol. 36, 2022, pp. 7603–7611.

[25] H. Liao, J. Hu, T. Li, S. Du, B. Peng, Deep linear graph attention model for attributed graph clustering, Knowl.-Based Syst. 246 (2022) 108665.

[26] L. Guo, Q. Dai, Graph clustering via variational graph embedding, Pattern Recognit. 122 (2022) 108334.

[27] L. Liu, Z. Kang, J. Ruan, X. He, Multilayer graph contrastive clustering network, Inform. Sci. 613 (2022) 256–267.

[28] D. Hu, D. Feng, Y. Xie, EGC: A novel event-oriented graph clustering framework for social media text, Inf. Process. Manage. 59 (6) (2022) 103059.

[29] J.A. Morente-Molinera, S.R. Aguilar, R. González-Crespo, E. Herrera-Viedma, Using clustering methods to deal with high number of alternatives on group decision making, Procedia Comput. Sci. 162 (2019) 316–323.

[30] R.-X. Ding, I. Palomares, X. Wang, G.-R. Yang, B. Liu, Y. Dong, E. Herrera-Viedma, F. Herrera, Large-scale decision-making: Characterization, taxonomy, challenges and future directions from an artificial intelligence and applications perspective, Inf. Fusion 59 (2020) 84–102.

[31] M. Tang, H. Liao, E. Herrera-Viedma, C.P. Chen, W. Pedrycz, A dynamic adaptive subgroup-to-subgroup compatibility-based conflict detection and resolution model for multicriteria large-scale group decision making, IEEE Trans. Cybern. 51 (10) (2020) 4784–4795.

[32] Y. Shang, Finite-time cluster average consensus for networks via distributed iterations, Int. J. Control Autom. Syst. 15 (2017) 933–938.

[33] P. Liu, K. Zhang, P. Wang, F. Wang, A clustering- and maximum consensus-based model for social network large-scale group decision making with linguistic distribution, Inform. Sci. 602 (2022) 269–297.

[34] S. Yao, M. Gu, An influence network-based consensus model for large-scale group decision making with linguistic information, Int. J. Comput. Intell. Syst. 15 (1) (2022) http://dx.doi.org/10.1007/s44196-021-00058-1.

[35] F. Jin, J. Liu, L. Zhou, L. Martínez, Consensus-based linguistic distribution large-scale group decision making using statistical inference and regret theory, Group Decis. Negot. 30 (4) (2021) 1–33, http://dx.doi.org/10.1007/S10726-021-09736-Z.

[36] J. Saarinen, A large-scale group decision-making model with no consensus threshold based on social network analysis, Inform. Sci. 612 (2022) 361–383, http://dx.doi.org/10.1016/j.ins.2022.08.075.

[37] H. Liao, X. Li, M. Tang, How to process local and global consensus? A large-scale group decision making model based on social network analysis with probabilistic linguistic information, Inform. Sci. 579 (2021) 368–387, http://dx.doi.org/10.1016/J.INS.2021.08.014.
[38] X. Zhong, X. Xu, X. Chen, A clustering and fusion method for large group decision making with double information and heterogeneous experts, Soft Comput. (2022) 1–13.
[39] J. Chen, H. Li, X. Zhang, F. Zhang, S. Wang, K. Wei, J. Ji, SR-HetGNN: session-based recommendation with heterogeneous graph neural network, Knowl. Inf. Syst. (2023) 1–24.
[40] X. Liu, X. Li, Y. Cao, F. Zhang, X. Jin, J. Chen, Mandari: Multi-modal temporal knowledge graph-aware sub-graph embedding for next-POI recommendation, in: 2023 IEEE International Conference on Multimedia and Expo, ICME, IEEE, 2023, pp. 1529–1534.
[41] J. Chen, Y. Cao, F. Zhang, P. Sun, K. Wei, Sequential intention-aware recommender based on user interaction graph, in: Proceedings of the 2022 International Conference on Multimedia Retrieval, 2022, pp. 118–126.
[42] C.-W. Hsu, C.-T. Chen, S.-H. Huang, Adaptive adversarial contrastive learning for cross-domain recommendation, ACM Trans. Knowl. Discov. Data 18 (3) (2023) 1–34.
[43] Z. Fu, X. Niu, M.L. Maher, Deep learning models for serendipity recommendations: A survey and new perspectives, ACM Comput. Surv. (2023).
[44] D. Han, Y. Huang, J. Liu, K. Liao, K. Lin, LSAB: User behavioral pattern modeling in sequential recommendation by learning self-attention bias, ACM Trans. Knowl. Discov. Data 18 (3) (2024) 1–20.
[45] V.R. Yannam, J. Kumar, K.S. Babu, B. Sahoo, Improving group recommendation using deep collaborative filtering approach, Int. J. Inf. Technol. 15 (3) (2023) 1489–1497.
[46] T. Lalitha, P. Sreeja, Recommendation system based on machine learning and deep learning in varied perspectives: a systematic review, in: Information and Communication Technology for Competitive Strategies (ICTCS 2020) Intelligent Strategies for ICT, Springer, 2021, pp. 419–432.
[47] R. Shrivastava, D.S. Sisodia, N.K. Nagwani, Deep neural network-based multi-stakeholder recommendation system exploiting multi-criteria ratings for preference learning, Expert Syst. Appl. 213 (2023) 119071.
[48] W.-H. Liao, Y.-T. Lin, C.-Y. Lin, S.-C. Kuai, A group recommendation system for movies using deep learning, in: 2023 International Conference on Consumer Electronics-Taiwan, ICCE-Taiwan, IEEE, 2023, pp. 61–62.
[49] Y. Wu, Q. Liu, R. Chen, C. Li, Z. Peng, A group recommendation system of network document resource based on knowledge graph and LSTM in edge computing, Secur. Commun. Netw. 2020 (2020) 1–11.
[50] B. Ait Hammou, A. Ait Lahcen, S. Mouline, A distributed group recommendation system based on extreme gradient boosting and big data technologies, Appl. Intell. 49 (2019) 4128–4149.
[51] S. Ali-Yari, N. Neissani Samani, M.R. Jelokhani Nayarki, Uncertainty modeling of a group tourism recommendation system based on pearson similarity criteria, Bayesian network and self-organizing map clustering algorithm, Eng. J. Geosp. Inf. Technol. 8 (1) (2020) 39–61.
[52] Y.-D. Seo, Y.-G. Kim, E. Lee, H. Kim, Group recommender system based on genre preference focusing on reducing the clustering cost, Expert Syst. Appl. 183 (2021) 115396.
[53] Y. Bai, H. Ding, S. Bian, T. Chen, Y. Sun, W. Wang, Simgnn: A neural network approach to fast graph similarity computation, in: Proceedings of the Twelfth ACM International Conference on Web Search and Data Mining, 2019, pp. 384–392.
[54] M. Pavan, M. Pelillo, Dominant sets and pairwise clustering, IEEE Trans. Pattern Anal. Mach. Intell. 29 (1) (2006) 167–172.
[55] H. Hung, B. Kröse, Detecting f-formations as dominant sets, in: Proceedings of the 13th International Conference on Multimodal Interfaces, 2011, pp. 231–238.
[56] A. Yazidi, M.A. Pinto-Orellana, H. Hammer, P. Mirtaheri, E. Herrera-Viedma, Solving sensor identification problem without knowledge of the ground truth using replicator dynamics, IEEE Trans. Cybern. 52 (1) (2020) 16–24.
[57] P.D. Taylor, L.B. Jonker, Evolutionary stable strategies and game dynamics, Math. Biosci. 40 (1–2) (1978) 145–156.
[58] E.Z. Mequanint, L.T. Alemu, M. Pelillo, Dominant sets for constrained image segmentation, IEEE Trans. Pattern Anal. Mach. Intell. (2018).
[59] Y.T. Tesfaye, E. Zemene, A. Prati, M. Pelillo, M. Shah, Multi-target tracking in multiple non-overlapping cameras using fast-constrained dominant sets, Int. J. Comput. Vis. (2019) 1–18.
[60] M. Pelillo, A. Torsello, Payoff-monotonic game dynamics and the maximum clique problem, Neural Comput. 18 (5) (2006) 1215–1258.
[61] K. Avrachenkov, V.S. Borkar, Metastability in stochastic replicator dynamics, Dynam. Games Appl. (2018) 1–25.
[62] W. Barfuss, J.F. Donges, J. Kurths, Deterministic limit of temporal difference reinforcement learning for stochastic games, Phys. Rev. E 99 (4) (2019) 043305.
[63] Q. Wang, N. He, X. Chen, Replicator dynamics for public goods game with resource allocation in large populations, Appl. Math. Comput. 328 (2018) 162–170.
[64] A. Felfernig, L. Boratto, M. Stettinger, M. Tkalčič, et al., Group Recommender Systems: An Introduction, Springer, 2018.
[65] E. Abbasnejad, S. Sanner, E.V. Bonilla, P. Poupart, Learning community-based preferences via dirichlet process mixtures of gaussian processes, in: Twenty-Third International Joint Conference on Artificial Intelligence, 2013, pp. 1213–1219.
[66] C.-N. Ziegler, S.M. McNee, J.A. Konstan, G. Lausen, Improving recommendation lists through topic diversification, in: Proceedings of the 14th International Conference on World Wide Web, 2005, pp. 22–32.
[67] A.K. Jain, R.C. Dubes, Algorithms for Clustering Data, Prentice-Hall, Inc., 1988.
[68] L. Boratto, S. Carta, G. Fenu, Discovery and representation of the preferences of automatically detected groups: Exploiting the link between group modeling and clustering, Future Gener. Comput. Syst. 64 (2016) 165–174.
[69] Z. Wang, Q. Xu, K. Ma, Y. Jiang, X. Cao, Q. Huang, Adversarial preference learning with pairwise comparisons, in: Proceedings of the 27th ACM International Conference on Multimedia, 2019, pp. 656–664.
[70] B. Cao, J. Zhao, Z. Lv, P. Yang, Diversified personalized recommendation optimization based on mobile data, IEEE Trans. Intell. Transp. Syst. 22 (4) (2020) 2133–2139.
[71] J. Xu, K. Guo, X. Zhang, P.Z. Sun, Left gaze bias between LHT and RHT: a recommendation strategy to mitigate human errors in left-and right-hand driving, IEEE Trans. Intell. Veh. (2023).
[72] Y. Peng, Y. Zhao, J. Hu, On the role of community structure in evolution of opinion formation: A new bounded confidence opinion dynamics, Inform. Sci. 621 (2023) 672–690.
[73] J. Dong, J. Hu, Y. Zhao, Y. Peng, Opinion formation analysis for expressed and private opinions (EPOs) models: Reasoning private opinions from behaviors in group decision-making systems, Expert Syst. Appl. 236 (2024) 121292.