

# APPLYING THE AFFECTIVE AWARE PSEUDO ASSOCIATION METHOD TO ENHANCE THE TOP-N RECOMMENDATIONS DISTRIBUTION TO USERS IN GROUP EMOTION RECOMMENDER SYSTEMS

John Kalung Leung<sup>1</sup>, Igor Griva<sup>2</sup> and William G. Kennedy<sup>3</sup>

<sup>1</sup>Computational and Data Sciences Department, Computational Sciences and Informatics, College of Science, George Mason University, 4400 University Drive, Fairfax, Virginia 22030, USA

<sup>2</sup>Department of Mathematical Sciences, MS3F2, Exploratory Hall 4114, George Mason University, 4400 University Drive, Fairfax, Virginia 22030, USA

<sup>3</sup>Center for Social Complexity, Computational and Data Sciences Department, College of Science, George Mason University, 4400 University Drive, Fairfax, Virginia 22030, USA

## ABSTRACT

*Recommender Systems are a subclass of information retrieval systems, or more succinctly, a class of information filtering systems that seeks to predict how close is the match of the user's preference to a recommended item. A common approach for making recommendations for a user group is to extend Personalized Recommender Systems' capability. This approach gives the impression that group recommendations are retrofits of the Personalized Recommender Systems. Moreover, such an approach not taken the dynamics of group emotion and individual emotion into the consideration in making top-N recommendations. Recommending items to a group of two or more users has certainly raised unique challenges in group behaviors that influence group decision-making that researchers only partially understand. This study applies the Affective Aware Pseudo Association Method in studying group formation and dynamics in group decision making. The method shows its adaptability to group's moods change when making recommendations.*

## KEYWORDS

*User Behavioral Analysis, Text-based Group Emotion-aware Recommender System, Emotion Prediction, Personality, Cross Information Domain Pseudo Users Association*

## 1. INTRODUCTION

There are two keys design considerations in Group Recommender Systems (GRS). All commercial Recommender Systems in the market are Group Recommender Systems . [50]. Group Recommender Systems would focus its design on making and distributing personalized recommendations to users in the group, and regardless the grouping formation is initiated by users or by the system. Some Group Recommender Systems provide users with group formation functions to self-create, manage, maintain, and disband a group. In most if not all Recommender Systems, for performance and throughput reasons, the system groups individual users or groups of users with similar preferences and tastes into a group without their awareness. When a Group Recommender System makes a top-N recommendation to an active user in a group, the same top-N list will serve all group members. Such a system users' grouping strategy is known as the system simulcast group (SSG) and is the most common type of system users' grouping strategy.

Another system users' grouping strategy is a system broadcast group (SBG), which refers to distributing the same information to all SSG. SSG and SBG message distribution is that SSG is intra-group oriented, whereas SBG is inter-group oriented.

The second Group Recommender Systems design point is taking from a user perspective about user grouping types. A user can be a single member of a uni-group (UG) or a multigroup (MG) member. Thus, the top-N list distribution could involve a network of SSG groups where the user is a member. Making top-N recommendations for an active user is relatively straight forward for a Recommender System. All it requires is to access the user's preference profile to guide the top-N recommendations-making process in evaluating and selecting recommendation candidates. However, making recommendations for a group of users is more challenging than making personalized recommendations. The Recommender System must adopt a group decision-making strategy to guide the top-N recommendations making for all group members. In other words, the Recommender System must consider the group preferences profile and the individual active user's preference profile when making the top-N recommendations for a group. Hence, regardless of whether the preference profile is group-oriented or personal oriented, the profile values are ever-changing as the group member consumed an item.

People love socializing. From a user's perspective, a Recommender system should support group formation management functions for group creation, group deletion, adding members to a group, removing members from a group, listing members in a group, setting group member privileges, notifying members in a group, and member vote accounting. Users should not be burden by the system in preference profile maintenance nor respond to query from the system about the user's preferences. The ideal system should know all about its users' likes and dislikes. It can derive the needed users' preference of information by mining the system's implicit metadata, such as system transactions and log files.

From a system perspective, it needs to find a viable solution to support all the above functions and more. For practical purposes, the minimum number of users in a group is two (2). One must find a means to compare the differences of profiles between a pair of users. This study draws the lesson learned from prior works in [34] and [37] that apply the Affective Aware Pseudo Association Method (AAPAM) to facilitate the affective computing of pairwise object's preference profile for the system and users group related operations. This study intends to address what feasible a Recommender System can do to provide the necessary support for the system's and users' group related operations while addressing the ever-changing users' profile issue.

## **2. RELATED WORK IN GROUP RECOMMENDER SYSTEMS**

Two or more persons can form a group—for example, friends who meet every weekend for dinner. A group can temporarily include random people to honor a March of Dimes walk for some worthy causes. Regardless of the formation of a group is either a recurring basis or an ephemeral basis, making a recommendation to a group faces two significant challenges: the semantics of group recommendation and the efficient way to compute for group recommendation [3].

To solve problems such as cold start, data sparsity, and scalability, many commercial websites that deployed Personalized Recommender Systems have incorporated methods by grouping users with similar preferences to share a list of recommendations for each group and [41]. Alternatively, in making a recommendation for groups, conventional approaches in information retrieval and filtering extend Personalized Recommender Systems [53]. However, this approach gives the impression that group recommendations are retrofits of their personalized counterparts.

Although some researchers argue that recommendations for groups using the extension approach perform well, such Group Recommender Systems are not designed with the proper algorithms to solve problems specific to a group setting [4].

## 2.1. State of Group Recommender Systems

Group Recommender Systems (GRS), despite the lack of notoriety in the field of Recommender Systems (RS), have been around as long as their glamorous sibling, the Personalized Recommender Systems (PRS), which were introduced in the mid-1990s to expedite personalized information retrieval on the Web which based on users' preferences [1]. Later, when smartphones become prevalent, research on Mobile Recommender Systems become a significant thrust [2]. Recommender Systems use different information sources to provide users with predictions and recommendations of items while balancing factors like accuracy, novelty, dispersity, and stability in the process. Collaborative Filtering is the favorite choice of method in recommendation processing, which often couples with other information filtering techniques to form a Hybrid Recommender System for achieving higher performance and better user experience [1], and [10]. For example, other popular types of information filtering methods include Content-Based, Knowledge-Based, Demographic-Based, and Social Networking Based Kompan et al. [30], Adomavicius et al. [1], Kywe et al. [32], De et al. [15], Bobadilla et al. [8], and Burke et al. [10]. Nonetheless, many of the personalized recommended items are often (or mostly) used by groups rather than by individuals, for example, FlyTrap for music [13], Pocket Restaurant Finder for restaurants [42], e-Tourism for tourism [17], PolyLens for movies [45], LET'S BROWSE for group web surfing [38], and YuTV for TV shows [61] are all Group Recommenders.

In recent years, various Group Recommender Systems have emerged. Besides building from scratch, these Group Recommenders most augment a Personalized Recommender to become a Group Recommender System through one of the two recommendation strategies. The first group recommendation strategy is "aggregating recommendations" of personalized recommendations into recommendations for the whole group [12]. It is similar to the concept, "aggregated predictions," as denoted by Senot et al. [56], are the results of aggregating predictions from an individual user into a group prediction. The second group recommendation strategy is "aggregating preferences" of the users' preference model into a group's preference model [5]. Similarly, "aggregated models," as described by Cantador et al. [11], refer to aggregate individual user data into group data.

From a user perspective, when a Group Recommender System allows users to create and manage groups, the grouping behavior is explicit [25]. In contrast, a Group Recommender System derives groups through aggregating recommendations or aggregating preferences or aggregated predictions or aggregated models; such grouping behaviors denote as the implicit grouping [51]. One of the benefits of grouping users into groups is eradicating the cold start problem that all Recommender Systems face [48].

## 2.2. Grouping Types

Since different groups exist, group recommender systems aim to manage the heterogeneity of groups. Boratto et al. [9] speculated that the formation of a group would affect its model and thus the predictive capability of a Recommender System. Building on the work of Jameson et al. ([25], which described the four tasks of a recommender system in detail, Boratto et al. [9] further extended these four tasks to four different variants of a group:

- Established group: people who share common interests and explicitly choose to join a group;
- Occasional group: people who occasionally meet over some activities;
- Random group: people share a resource on occasion without their explicit consent;
- Auto group: People with shared preferences and automatically identified and grouped to share some shared resources.

### 2.2.1. Established Group

O'Connor et al. [45] described the established groups which they observed in PolyLens, a movie Recommender for group use, have a persistent property that users who joined a group not only shared common interests such as movie watching but also actively participated in group activities such as rating watched movies. Once a user joined an established group, the user tends to stay with the group for a long time. Many Group Recommender Systems, such as PolyLens, apply the Collaborative Filtering method to construct user profiles for an individual member in the group and then build the group profile by merging individual profiles from the group members [28]. To produce recommendations for each group member, PolyLens uses a Collaborative Filtering algorithm to compute each movie's rating score that meets the user's preferences. Movies with the highest recommended rates become the Top-N candidates for recommendation. PolyLens uses a "least misery" selection strategy for making the group's recommendation [45], i.e., the recommended rating for a group is the lowest predicted rating for a movie, to ensure every member is satisfied.

However, some Group Recommenders such as Jukola[46] and PartyVote[58] are two music recommenders able to make music recommendations to an established social group of people attending a social event. These two music recommenders work without requiring any user profiles. Instead, these recommenders allow any event attendees to express their preferences by selecting a song, album, artist, or genre from a digital music collection. The rest of the group votes to play songs from the selected list. The Recommender computes the probability of the voted songs and plays the song which has the highest probability.

### 2.2.2. Auto Group

Let  $v_i$  be the vector of the ratings of user  $i$  for the items and  $v_j$  be the vector of the ratings of user  $j$  for the items. Cosine similarity measures the similarity of  $s_{ij}$  between users  $i$  and  $j$  as expressed in Boratto et al. [9] proposed group recommendation algorithm that works in four steps:

- 1) Step 1. Cosine similarity between users' ratings matrixes measures users' similarity. The evaluation procedure [19] illustrates as follows.

$$\psi(u, v) = \cos(\vec{u} \odot \vec{v}) = \frac{\vec{u} \odot \vec{v}}{\|\vec{u}\|_2 \times \|\vec{v}\|_2} = \frac{\sum_{i \in I_{u,v}} r_{u,i} r_{v,i}}{\sqrt{\sum_{i \in I_{u,v}} r_{u,i}^2} \sqrt{\sum_{i \in I_{u,v}} r_{v,i}^2}}$$

Where " $\odot$ " is denoted as the dot product operator and " $\times$ " denoted as the multiplication operator.

- 2) Step 2. Communities detection algorithm proposed by [7] can apply to the user's similarity network and generate partitions of different granularities.
- 3) Step 3. Rating prediction for items rated by enough users of a group by aggregating the arithmetic mean of users involved for the group. So, for each item  $i$ , its rating  $r_i$  is expressed as:

$$\psi(u, v) = \cos(\vec{u} \odot \vec{v}) = \frac{\vec{u} \odot \vec{v}}{\|\vec{u}\|_2 \times \|\vec{v}\|_2} = \frac{\sum_{i \in I_{u,v}} r_{u,i} r_{v,i}}{\sqrt{\sum_{i \in I_{u,v}} r_{u,i}^2} \sqrt{\sum_{i \in I_{u,v}} r_{v,i}^2}}$$

- 4) Step 4. Ratings prediction for the remaining unrated items by considering both the rating and the similarity ( $t_{ij}$ ) of its top similar items:

$$\psi(u, v) = \cos(\vec{u} \odot \vec{v}) = \frac{\vec{u} \odot \vec{v}}{\|\vec{u}\|_2 \times \|\vec{v}\|_2} = \frac{\sum_{i \in I_{u,v}} r_{u,i} r_{v,i}}{\sqrt{\sum_{i \in I_{u,v}} r_{u,i}^2} \sqrt{\sum_{i \in I_{u,v}} r_{v,i}^2}}$$

Where, “ $\odot$ ” is denoted as the dot product operator.

### 2.3. User Perspective in Grouping

From a user's perspective, groups' persistence is related to both usage patterns and privacy issues. If users want to repeatedly receive recommendations for the same group of people, making the group persistent saves time and effort. Moreover, if groups form and dissociate after a single use, the ephemeral approach is better to meet the need. Whether groups are private, known only to group members, or public and accessible to all. Users will behave differently in addressing their privacy, security, net appearance, and perception needs.

#### 2.3.1. Pseudo User

Pseudo user is not a new concept in the field of recommender systems. However, the pseudo-user takes on different semantics and roles under different context as described by O'Connor et al. [45], Melville et al. [43], Resnick et al. [52], Martinez et al. [40], Vozalis et al. [60], and Lee et al. [33].

#### 2.3.2. Pseudo Grouping

In this research study context, pseudo grouping and virtual grouping are synonyms. Research studies use pseudo users in recommender systems, as in Martinez et al. [40]. However, no study has investigated leverages pseudo groups as a grouping strategy and plan to apply pseudo grouping to solve the cold start problem.

#### 2.3.3. Combine Similar Groups to form Super Group

Several studies of recommender systems have applied Supergroup mostly in the study of the semantic context under content filtering, such as Hu et al. [22]. The supergroup concept is yet widely applied in the grouping strategy of recommender systems research.

### 2.4. System Perspective in Grouping

There are times when a Recommender System is the initiator of the grouping strategy. For example, to improve the performance of making recommendations to users, Recommender Systems may select to partition users with similar taste into a group and make the same set of recommendations to all the users in that group to minimize search through the entire rating database or comparing the similarity of pair-users' profiles. In doing so, the Recommender

System sacrifices accuracy for performance gain by making similar recommendations to a group of individual users.

## **2.5. Ephemeral and Persistence**

The notion of ephemeral as a nature of group had been examined exhaustively by O'Connor et al. [45], Jameson [24], Schafer et al. [55], and Good et al. [20]. On the other hand, Gartrell et al. [18], O'Connor et al. [45], Erickson [16], Smeaton et al. [57] had researched the persistence and related issues of groups.

## **2.6. Merging Strategies**

O'Connor et al. [45] developed PolyLens, a collaborative filtering recommender system designed to recommend movies for a group of viewers rather than individuals. PolyLens merges the individual user's profile among similar taste users to form the group profile and applies the similarity function against the rating matrix for suitable items for making a recommendation to the group. This dissertation research will evaluate other merging strategies besides the simple aggregation approach.

### **2.6.1. Weight Settings for Influential Members**

Kim et al. [29] articulated that social influence impacts e-Commerce decision-making. Few studies have considered a social influence in an e-Commerce decision support system because until recently, data about social interaction does not adequately capture in e-Commerce. It becomes apparent that the customer decision process is influenced by information from trusted people, not from product manufacturers or recommendation systems. The social influence from high-quality reviews written by previous consumers can have a direct, positive effect on potential consumers' decision making, and this effect can propagate through a social network Huang et al. [23]. One of the aims of this dissertation research is to examine a balanced method in extracting the degree of influence by weighing that members exert among each other through decision-making.

## **2.7. Group Recommendation Making Strategies**

There are several ways of extending a Personalized Recommender System to a Group Recommender System as described by De et al. [14]. Merging strategy and Virtual User strategy are two conventional approaches for grouping. According to Kagita et al. [26], there are three ways to implement the Merging strategy: merged profiles, merging recommendation, and merging score. Incidentally, Kagita et al. [27] also illuminated the Virtual User strategy. The aspect of grouping strategies for a system is different from that of a user. Next, to examine are the differences between them. Specifically, an examination on the nature of ephemeral group versus persistence group from a user perspective and the following grouping strategies from a system perspective:

- 1) Making group recommendations for a clustered of users,
- 2) Making group recommendations by considering transitive precedence relation through a virtual user model,
- 3) Adapting personalized recommendation for a group through aggregation strategies,
- 4) Automatic identification groups of users with similar interests, and
- 5) Making group recommendations through group behavior modeling.

### 2.7.1. Making Group Recommendations for a Clustered of Users

To avoid an exhaustive search through the entire user preference database to match a particular user's preferences, Ntoutsis et al. [44] advocated a group recommendation system that enhances recommendations partitioning users into clusters based on similar preferences. Aggregated preferences of clustered members drive the decision making of recommendations for users. Popescu et al. also studied aggregated user preference [49] and Masthoff et al. [41]. The algorithms to estimate the relevance of an item for a user deployed by Ntoutsis et al. [44] for the group recommendations framework come in two flavors:

- 1) In studies of Amer et al. [3] and Konstan et al. [31]. both have illustrated personalized recommendations produced from evaluating relevance scores for unrated items of an unusually active user based on collaborative filtering techniques. However, users typically rate only a few items against the vast amount of the available items. Thus, the notion of support measures the percentage of the active user's friends who have expressed preferences for the item to minimize the rating matrix's skewness. The relevance and support scores are then combined to estimate a recommendation worthiness score of an item for a user.
- 2) In addition to personalized recommendations, group recommendations deploy a model built on context information of users in the group as illustrated in Amer et al. [3] by combining all individual users' preferences. Moreover, Ntoutsis et al. [44] further refined these group preferences by three aggregation design methods that carry different semantics. Firstly, the Least Misery design will capture cases where strong user preferences act as a veto. Next, the Fair design will capture more common cases where most of the group members are satisfied. Lastly, the Most Optimistic design will capture cases where the most satisfied member of the group acts as the most influential member. After applying these three design methods appropriately, a Top-K list then uses in making recommendations to users in the group.

### 2.7.2. Making Group Recommendations through Aggregation Functions

Two dominant strategies for making a Personalized Recommender to become a Group Recommender [5]. The first grouping strategy is to aggregate individual preferences into a recommendation list. In effect, this approach creates a pseudo-user for a group based on its group members and then makes recommendations based on the pseudo user's preferences. The second grouping strategy is to aggregate the individual member's recommendation list to form the group recommendation list. In other words, every single user in the group will receive an individual recommendation list. All individual recommendation lists combine to form a recommendation list for the group. The aim of making a group recommendation is to compute a recommendation score for each candidate item that reflects the interests and preferences of all group members. An acceptable approach to obtain a consensus of group ranking recommendation score for a candidate item is through some aggregation functions. Popular aggregation functions are as follows.

Least-misery:

$$\psi(u, v) = \cos(\vec{u} \odot \vec{v}) = \frac{\vec{u} \odot \vec{v}}{\|\vec{u}\|_2 \times \|\vec{v}\|_2} = \frac{\sum_{i \in I_{u,v}} r_{u,i} r_{v,i}}{\sqrt{\sum_{i \in I_{u,v}} r_{u,i}^2} \sqrt{\sum_{i \in I_{u,v}} r_{v,i}^2}}$$

Average:

$$\psi(u, v) = \cos(\vec{u} \odot \vec{v}) = \frac{\vec{u} \odot \vec{v}}{\|\vec{u}\|_2 \times \|\vec{v}\|_2} = \frac{\sum_{i \in I_{u,v}} r_{u,i} r_{v,i}}{\sqrt{\sum_{i \in I_{u,v}} r_{u,i}^2} \sqrt{\sum_{i \in I_{u,v}} r_{v,i}^2}}$$

Average without Misery:

$$\psi(u, v) = \cos(\vec{u} \odot \vec{v}) = \frac{\vec{u} \odot \vec{v}}{\|\vec{u}\|_2 \times \|\vec{v}\|_2} = \frac{\sum_{i \in I_{u,v}} r_{u,i} r_{v,i}}{\sqrt{\sum_{i \in I_{u,v}} r_{u,i}^2} \sqrt{\sum_{i \in I_{u,v}} r_{v,i}^2}}$$

where

$$\psi(u, v) = \cos(\vec{u} \odot \vec{v}) = \frac{\vec{u} \odot \vec{v}}{\|\vec{u}\|_2 \times \|\vec{v}\|_2} = \frac{\sum_{i \in I_{u,v}} r_{u,i} r_{v,i}}{\sqrt{\sum_{i \in I_{u,v}} r_{u,i}^2} \sqrt{\sum_{i \in I_{u,v}} r_{v,i}^2}}$$

### 2.7.3. Adapting Personalized Recommendation for Group through Aggregation Strategies

The main problem of group recommendation stems from adopting personalized recommendation based on individual user preference. Recommender Systems have to learn users' preferences from user rating records. After learning individual user preferences, recommender systems then aggregate user rating information to provide group recommendations. Aggregation strategies research is ongoing. Researchers such as Mastho et al. [41] developed various aggregation strategies that later research built-on. Highlight below is a few of their aggregation strategies. Merging individual user profiles as the group's preferences is the superior strategy.

The utilitarian strategy considers utility values for group recommendation. The utility values are of two types, i.e., additive or multiplicative. An example of a movie recommender system first adds/multiply all the movie ratings separately for a user group. Then display the movies list in the highest aggregate utility value as recommended items.

Most pleasure strategy generates a group rating list based on the maximum of individual ratings. For example, movies with the highest universal rating values will add to the recommended list. The least misery strategy generates a group rating list based on the minimum of individual ratings. Then the item with a high minimum individual rating will be recommended. The idea behind this strategy is that a group is as happy as its least happy member.

Padmanabhan et al. [47] implemented a rule-based aggregation strategy called RTL strategy to group recommender systems by combining all of the above three strategies. This algorithm performs better regarding accuracy in group recommendation as compared to the preceding three grouping strategies. However, RTL has no provision for creating a better group (regrouping). Pujahari et al. [50] solved the regrouping deficiency by applying Predictive Rule Mining algorithms [39], which featured learning rules to learn individual users' preferences applying a new algorithm for making group recommendation.

## 3. DATASETS

Developing an Emotion Aware Recommender System requires an emotion labeled dataset to work with various machine learning algorithms. However, there is no readily available emotion labeled dataset in any publicly accessible repository. One can overcome the deficiency by mining emotion features from existence metadata in the dataset. One can demonstrate making top-N movie recommendations to users in a group through an Emotion Aware Recommender System.



Thus, it is necessary to mine the needed emotion label from the subjective text in movie branch mark datasets from the MovieLens repository and The Movie Database (TMDb) website, where the movie overview contains the subjective text metadata. One can download a variety of MovieLens movie datasets stored in the GroupLens repository. However, one needs to scrape the TMDb website for needed metadata. Before performing any emotion modeling on a dataset, one needs to develop a text-based emotion classifier to classify the movie overview's emotion features.

### 3.1. MovieLens Datasets

In the article Leung et al. [35], it worked with [54] advocated the four MovieLens movie benchmark datasets, namely, ml-latest-small (a.k.a. mlsm), ml-20m (a.k.a. ml20m), ml-25m (a.k.a. ml25m), and ml-latest (a.k.a. ml27m hereafter) datasets. [35] also illustrated the need to scrape the TMDb for movie overview metadata to use as input to the Tweets Affective Classifier (TAC) to classify the scraped TMDb dataset's emotion labels before merging with MovieLens datasets. Table 1 illustrates the statistics of each mentioned MovieLens dataset after merging with the emotion labeled TMDb dataset. The name of a MovieLens dataset reflects the number of ratings it contains. Please note that the Number of Users column does not match up the Number of Overviews column. The reason is that the collection of movies in MovieLens and TMDb is not identical.

Table 1. Statistics of MovieLens datasets.

Attribute Dataset	No. Users	No. Movies	No. Ratings	No. Overviews
mlsm	610	9742	100836	9625
ml20m	138493	27278	20000263	26603
ml25m	162541	62423	25000095	60494
ml27m	283228	58098	27753444	56314

In each of the four MovieLens datasets, there are two data files named ratings, and tags contain user ids as a unique identifier. MovieLens stated that user ids found in ratings and tags data files are consistent within the same dataset but are not uniform across different datasets [21]. For example, in [35], user id 400 is only compatible within the same MovieLens MLSM dataset and is not across ml20m, ml25m, ml27m datasets. In other words, user id 400 in other MovieLens datasets are not the same user id 400 as in the MLSM dataset. However, [35] has demonstrated by using the Affective Aware Pseudo Association Method (AAPAM), the disjoint user id 400 in MLSM can correctly connect to the proper user id in MovieLens datasets, as depicted in Table 2. For example, user id 400 of MLSM can Pseudo Associate Connect (PAC) to user id 66274 of MLSM or PAC to user id 95450 of ml25m or PAC to user id 89195 of ml27m. All mentioned user id are disjoint users of each other. In the context of this study, disjoint users refer to two or more individual mutually exclusive users whose user id are different and reside in different datasets within the same or different information domains whose emotion profiles, UVEC, are highly similar or even identical. One can consider the disjoint users as identical pseudo users.

### 3.2. Affective Aware Pseudo Association Method

The Affective Aware Pseudo Association Method (AAPAM) computes the Affective Index Indicator (AII) using the Cosine Similarity algorithm [6], as depicted in Equation 2, Cosine Similarity expresses the closeness of the emotion profiles between two users or items. When using AAPAM to compare pairwise between User id 400 of MLSM against users in other MovieLens datasets, AII reveals, as depicted in Table 2, the closest other users' emotion profiles that match the candidate user. User id 400 in MLSM can make a pseudo associate connection

(PAC) to user id 66274 with AII 0.999916 in ml20m or to user id 95449 with AII 0.999999 ml25m, or user id 89195 with AII 0.999999 in ml27m, respectively.

$$Inner(x, y) = \sum_i x_i y_i = \langle x, y \rangle \quad (1)$$

$$CosSim(x, y) = \frac{\sum_i x_i y_i}{\sqrt{\sum_i x_i^2} \sqrt{\sum_i y_i^2}} = \frac{\langle x, y \rangle}{||x|| ||y||} \quad (2)$$

AAPAM also worked with The Movie Database (TMDb) [59]. By scraping from TMDb, it yields the movie metadata for movie overviews, poster images, and other metadata. AAPAM applied the Tweets Affective Classifier (TAC), a method developed in [36], to classify a movie emotion profile. A movie emotion profile is also known as a movie vector, MVEC, which represents a multi-dimensional embedding of a probability distribution of seven primary human emotions: neutral, happiness, sadness, hate, anger, disgust, and surprise. Each user in [36] also has a user emotion profile, UVEC, where it contains the average value of all movies MVECs the user has watched.

Table 2. Pseudo Association Connection of ml-latest-small user id 400 to other users in different datasets through affective index indicator.

Dataset	MLsm	ml20m	ml25m	ml27m
User1 ID PAC	400	66274	95459	89195
User1 Movie Count	43	22	43	43
User1 Watched List movieID	6 47 50 260 ... 122886 134130 164179 168252	47 260 300 307 ... 2628 2797 3418 3481	6 47 50 260 ... 122886 134130 164179 168252	6 47 50 260 ... 122886 134130 164179 168252
User1 UVEC				
Neutral	0.16353	0.16250	0.16353	0.16353
Happiness	0.08874	0.08609	0.08874	0.08874
Sadness	0.12709	0.12654	0.12709	0.12709
Hate	0.20332	0.20701	0.20332	0.20332
Anger	0.11934	0.11776	0.11934	0.11934
Disgust	0.15881	0.16005	0.15881	0.15881
Surprise	0.13918	0.14005	0.13918	0.13918
User1 Affective Index Indicator	1.0	0.99992	0.99999	0.99999

As illustrated in Table 2, user id 400 in the rating data file of the MLSM dataset has watched 43 movies; taking the average of all the 43 movies' MVECs yields the UVEC for user id 400. As

mentioned in [36], a movie's MVEC is static and stays unchanged throughout the film's life; whereas, a user's UVEC changes its value each time the user watches a movie. The user's UVEC reflects the up-to-date movie taste and preference of the user. A movie MVEC is unique, while a UVEC may not be unique when two users watched the same movie set.

Besides, the AAPAM method can PAC connect disjoint users from different datasets within the same domain; this study believes the same technique can PAC connect disjoint users and items among different datasets across different domains. Unlike MovieLens datasets, some other movie datasets such as TMDb highlight the average voting score, the sentiment rating value on a scale of 1 (lowest) to 10 (highest), of a movie by a group of users who have watched and rated the movie through the voting count attribute in the data file instead of individual user's sentiment. TMDb does not maintain user information, and neither contains a user-id field in the dataset. When applying the AAPAM to connect MovieLens and TMDb domains, the PAC connection applies to movie items between MovieLens and TMDb. Here, the PAC connection between movie A in MovieLens to movie B in TMDb indicates how similar the two movies' emotion profiles, MVECs, form a one-to-one relationship. However, when applying the PAC to connect a user in MovieLens and a movie item in TMDb, the movie item MVEC in TMDb must first be normalized with the respective voting count. The normalized MVEC represents the average UVEC of the group of users who have rated it. Thus, the PAC connection between user A in MovieLens to the normalized MVEC of movie B in TMDb indicates how similar user A to a group of users B is in the form of a one-to-many relationship.

#### 4. METHODOLOGY

This study utilized affective features in two data sources that it uses. The affective features applied to users' emotion profiles, UVECs, and items' emotion profiles, IVECs, or MVECs equivalent. No disjoint users and items can interconnect without adding affective features across data sources of MovieLens, and TMDb. Affective aware features are added to the data sources through Tweets Affective Classifier (TAC) as developed in [36]. Table 3 depicts samples after added affective features UVEC and MVEC in this study's data sources.

Table 3. TMDb movie emotion profile example.

tmbId	2	525662
movieId	4470	189111
Mood	Disgust	Hate
Neutral	0.15705037	0.11876434
Happiness	0.08608995	0.05086204
Sadness	0.15583897	0.12669845
Hate	0.07506061	0.3391073
Anger	0.08469571	0.13069303
Disgust	0.26612538	0.13746719
Surprise	0.17513901	0.096407644

##### 4.1. Tweets Affective Classifier

Leung et al. in [36] illustrated the Tweets Affective Classifier (TAC) development, a text-based tweets emotion classifier capable of detecting and recognizing six basic human emotions advocated by Paul Ekman. The six emotions are happiness, sadness, fear, anger, disgust, and surprise. For affective computing convenience, one designs TAC capable of detecting and recognizing natural emotion. One then feeds TMDb's movie overview to TAC to classify the overview's moods or the movie's emotion profile, MVEC depicted in Table 4. After joining the

emotion-labeled TMDb data file with the rating data file in MovieLens, it yields an emotion-labeled MovieLens data file depicted in Table 5. With the joined emotion-labeled data file, one computes the users' emotion profiles, UVEC, by taking the MVEC from all the movies that the user has watched.

Table 4. First few rows of the cleansed emotion labeled MovieLens data file.

Index	tid	mid	iid	mood	neutral	happy	sad	hate	anger	disgust	surprise
1	2	4470	94675	disgust	0.157	0.086	0.156	0.075	0.085	0.266	0.175
2	5	18	113101	disgust	0.121	0.060	0.098	0.128	0.133	0.244	0.216
3	6	479	107286	hate	0.075	0.114	0.054	0.433	0.095	0.128	0.100
4	11	260	76759	neutral	0.299	0.262	0.079	0.030	0.017	0.083	0.230
5	12	6377	266543	surprise	0.150	0.080	0.055	0.083	0.103	0.153	0.376

Table 5. Sample of users' emotion profiles derived from taking the average of the user's moods values from all movies the user has watched

mlsm	neutral	Joy	sadness	hate	anger	disgust	surprise
400	0.1635299	0.0887352	0.12708998	0.2033184	0.1193381	0.1588128	0.13917538
414	0.1663518	0.0973058	0.1180924	0.1641951	0.1151779	0.1725031	0.16637367
474	0.1688583	0.0997465	0.1187206	0.1608771	0.1126127	0.1719196	0.16726495
448	0.1728330	0.0968581	0.11604573	0.1612073	0.1122760	0.1709857	0.16979389

## 4.2. System Grouping

This paper advocates a method for the system to group users in a system simulcast group (SSG) according to the similarity in users' emotion profiles, UVEC, so that all SSG users share the same top-N recommendations list. However, to simulate the personalized recommendation functionality before pushing the SSG top-N list to an active user in the SSG, the system reranked the top-N list's MVEC against the active user's UVEC, thereby ensuring all users in the SSG receives personalized individual top-N recommendations list. The benefit of grouping users in SSG is improving the system's throughput while performing fewer top-N recommendations made by the Recommender System. For example, a system hosts  $n$  million users. The system must run a top-N recommendation process per-user or  $n$  million top-N recommendations operations to make  $O(n)$ . personalized recommendations. By grouping users into  $m$  users per SSG, the system only needs to make  $\frac{n}{m}$  top-N or  $O\left(\frac{n}{m}\right)$ . It is an  $m$  fold decrease in top-N recommendations making. Thus, by grouping users in SSG, the system can expand the handling of  $\approx m$  more users without upsizing the system.

This paper has devised the following scheme to demonstrate an SSG's working in an Emotion Aware Recommender System. MovieLens ml-latest-small (MLSM) ratings data file contains 100,836 ratings of 9743 movies by 610 users. Some users rated 20 movies, the minimum number of movies a user must have rated in MovieLens sampled datasets, while some users have rated thousands of movies. By sorting the MLSM rating data file in descending order with users rated most movies at the top, draw out the top ten users to act as the dominant user in one of the ten SSG groups. Table 7 depicts ten dominant users to act as anchor user for the ten SSG group. Table 7 also shows the sample list of SSG group members. With each SSG's dominant user's UVEC, it computed against all undraw users through pairwise Cosine Similarity of UVEC. It ranked the result in descending order. In effect, the result shows the Affective Index Indicator (AII) of each member's UVEC relative to the UVEC of the dominant user. Draw out the top 60 users who have the most similar emotion profile of the dominant user to join the SSG. Continue

the same step to draw out members for the other nine SSG. The grouping scheme forming ten SSG each has a dominant user who rated most movies and 60 members with closely matched emotion profiles against the dominant user’s UVEC.

IMDb periodically publishes a list containing the “IMDb 100 Greatest Movies of All Time.” Using the list as top-N recommendations, one can feed to all ten SSG. Doing so, one simulates the system broadcast group (SBG) broadcast message function. Before sending the top-N recommendations to an SSG member, the system reranked the user’s UVEC against all movie emotion profiles, MVEC, on the list through the pairwise Cosine Similarity. This paper picks three users from one of the SSG to demonstrate the reranking process. Hence the SSG organized by ranking all members’ UVEC by computing their Affective Index Indicator (AII). The dominant member occupies the top of the AII list. The least-misery member occupies the bottom of the AII list, and the 30th member in the AII list is the average member of the SSG.

Table 6 showing the reranked top 10 of the “IMDb 100 Greatest Movies of All Time” for MovieLens ml-latest-small user ID 414 using UVEC of 20% and 100% movie watching history to compute the pairwise AII of Cosine Similarity. The two reranked lists of user ID 414 show a different rank order from the movie ranking of IMDb and the more movies the user has watched. Users with different UVEC affect the reranked result.

Table 6. User ID 414 Reranked IMDb 100 Greatest Movies of All Time using 20% and 100% Movie Watching UVEC

Ranking 100 All Time Greatest Movies	Moive ID	Movie Title	UserId414 Reranked (UVEC 20% watched)	UserId414 Reranked Movie Title (UVEC 20% Watched)	UserId414 Reranked (UVEC 100% watched)	UserId414 Reranked (AII - UVEC 100% Watched)
1	858	The Godfather (1972)	1252	Chinatown (1974)	1252	Chinatown (1974)
2	1221	The Godfather: Part II (1974)	1213	Goodfellas (1990)	1213	Goodfellas (1990)
3	2019	Seven Samurai (1954)	899	Singin' in the Rain (1952)	318	The Shawshank Redemption (1994)
4	296	Pulp Fiction (1994)	318	The Shawshank Redemption (1994)	899	Singin' in the Rain (1952)
5	1203	12 Angry Men (1957)	912	Casablanca (1942)	8125	Sunrise (1927)
6	5618	Spirited Away (2001)	8125	Sunrise (1927)	26150	Andrei Rublev (1966)
7	527	Schindler's List (1993)	26150	Andrei Rublev (1966)	912	Casablanca (1942)
8	912	Casablanca (1942)	89759	A Separation (2011)	89759	A Separation (2011)
9	1219	Psycho (1960)	1251	8½ (1963)	1237	The Seventh Seal (1957)
10	1213	Goodfellas (1990)	1237	The Seventh Seal (1957)	1251	8½ (1963)

### 4.3. User Grouping

One significant difference between a Group Recommender System and Personalized Recommender System is that Group Recommender System supports users with functions to initiate group formation, and Personalized Recommender System cannot. A Group Recommender System allows the system and users to create a multi-group (MG) and explicitly join the group. The system provides functions for the user to self-manage group administration and group maintenance. An MG's joining can be by invitation only known as private multi-group (PMG) or open to public multi-group (OMG). When the system makes a top-N recommendation to an MG, all MG members share the top-N list, and no need to re-rank the top-N is necessary for individual users in the MG. However, the Group Recommender System may apply a decision-making support strategy in top-N recommendations to assist the group decision making in selecting a recommendation in the top-N list. This study focuses on applying affective awareness recommendations to support group decision-making by the most dominant member strategy, average members' mood strategy, and the Least-misery member strategy. The most dominant member in an MG has the largest number of movies a member has watched among the group. The least-misery member is the member whose emotion profile is most dissimilar to the MG's most dominant member. The least-misery member of the group determines by the pairwise Affective Index Indicator (AII) of all members' emotion profiles in the MG against the most dominant member emotion profile, then rank the AII list in the descending order, the least-misery member occupies the lowest rank. Depicted in Table 8 is a user MG of size 5. User ID 195 is the dominant member of the group, and user ID 463 is the least-misery member.

## 5. RESULTS

### 5.1. Group Formation by System

One needs to determine the number of simulcast group (SSG),  $g$ , that wants to form. The system can make a rank list on user interaction and pick a reasonable number for  $g$ . In the MovieLens dataset, the system can compute the number of movies watching history as the system interaction criteria. Each SSG will anchor with a member pick from the top of the interaction rank list. The system computes the pairwise Cosine Similarity between the anchor member of the SSG against other ungrouped users. After ranking the pairwise list, the system will move  $m$  number of users from the top of the ranked list to the SSG. Depicted in Table 7 is 10 SSG on MovieLens ml-latest-small dataset. Each SSG anchors with a dominant user with most movie watching history. Each SSG contains 61 members, including the anchor member. The system picks members by the Affective Index Indicator (AII) values highly similar to the anchor member.

Table 7. 10 System Simulcast Groups Formed by Using MovieLens ml-latest-small

Rank	Dominant User ID	Movie Count	System Simulcast Group Member User ID	Member Count
1	414	2698	(1) 414, (2) 232, ..., (31) 212, ..., (61) 167	61
2	599	2478	(1) 599, (2) 477, ..., (31) 198, ..., (61) 474	61
3	474	2108	(1) 474, (2) 560, ..., (31) 260, ..., (61) 350	61
4	448	1864	(1) 448, (2) 226, ..., (31) 453, ..., (61) 389	61
5	274	1346	(1) 274, (2) 330, ..., (31) 424, ..., (61) 477	61
6	610	1302	(1) 610, (2) 160, ..., (31) 405, ..., (61) 218	61
7	68	1259	(1) 68, (2) 414, ..., (31) 212, ..., (61) 593	61
8	380	1218	(1) 380, (2) 160, ..., (31) 218, ..., (61) 318	61
9	606	1115	(1) 606, (2) 177, ..., (31) 6, ..., (61) 66	61
10	288	1055	(1) 288, (2) 483, ..., (31) 555, ..., (61) 167	61

## 5.2. Grouping by Users

Once a Group Recommender System provides users with group formation functionality, users can use the group formation to create and maintain a multi-users group (MG). In the following illustration, all members subsampled randomly from the derived emotion labeled movie dataset MLSM. Depicted below is a five-member multi-user group. The group average UVEC derived from the mean of five members' UVEC. User ID 195 watched most movies, thus the dominant member, whereas user ID 463 is the least-misery member.

Table 8. A Five Member Multi-User Group

Rank	UserId	Watched	UVEC
1	195	187	0.1639455, 0.0902557, 0.1176815, 0.1726736, 0.1185870, 0.1777129, 0.1591437
2	602	135	0.1639545, 0.0869860, 0.1168919, 0.16947266, 0.1156349, 0.1817310, 0.1653290
3	190	66	0.1603803, 0.0849701, 0.1254172, 0.17182250, 0.1135154, 0.1797844, 0.1641099
4	521	40	0.1574143, 0.0944750, 0.1240710, 0.14589457, 0.1083259, 0.1795868, 0.1902323
5	463	33	0.1558253, 0.0968441, 0.1140474, 0.19975860, 0.1226243, 0.1571110, 0.1537890
Average	Group	UVEC	0.1603040, 0.0907061, 0.1196220, 0.17192440, 0.1157376, 0.1751852, 0.1665208

## 5.3 Group Decision Making Strategies

A reasonable way to handle a group decision making using the dominant member strategy is to generate the top-N recommendations based on the dominant user's preference. Similarly, to support the group decision making using the least-misery member strategy, the top-N recommendations are generated based on the least-misery member's preference. Instead of using a movie database to generate a disparate top-N list based on different decision-making strategies, this paper limits the overall movie selection range to generate top-N listed from the "IMDb 100 Greatest Movies of All Time". Each of the movies in the IMDb list has computed an emotion profile MVEC. By computing the Affective Index Indicator (AII) pairwise between the UVEC of the preferred user group decision-making strategy and the IMDb MVEC and reranked the result in descending order will yield the desire top-N list. Depicted in Table 9 listed the top-10 of dominant user ID 195, least-misery user ID 463, and using the five-member MG average UVEC. Table 9 shows the top-10 recommendations list generated to support MG users' decision-making strategies of the dominant member, least-misery member, and average group user profile.

Table 9. Top-10 Generated by Dominant Member, Least-misery Member and Average Group User Profile Decision-making Strategies

Top-10 Rank	UserId195 Dominant User Strategy Top-10	UserId195 Dominant user Top-10 Movie Title	UserId463 Least-misery User Strategy Top-10	UserId463 Least-misery User Top-10 Movie Title	Average Group User Strategy Top-10	Average Group User Top-N Movie Title
1	858	The Godfather (1972)	1252	Chinatown (1974)	1252	Chinatown (1974)
2	1221	The Godfather: Part II (1974)	1213	Goodfellas (1990)	1213	Goodfellas (1990)

3	2019	Seven Samurai (1954)	899	Singin' in the Rain (1952)	318	The Shawshank Redemption (1994)
4	296	Pulp Fiction (1994)	318	The Shawshank Redemption (1994)	899	Singin' in the Rain (1952)
5	1203	12 Angry Men (1957)	912	Casablanca (1942)	8125	Sunrise (1927)
6	5618	Spirited Away (2001)	8125	Sunrise (1927)	26150	Andrei Rublev (1966)
7	527	Schindler's List (1993)	26150	Andrei Rublev (1966)	912	Casablanca (1942)
8	912	Casablanca (1942)	89759	A Separation (2011)	89759	A Separation (2011)
9	1219	Psycho (1960)	1251	8½ (1963)	1237	The Seventh Seal (1957)
10	1213	Goodfellas (1990)	1237	The Seventh Seal (1957)	1251	8½ (1963)

## 6. FUTURE WORK

The author et al. Plans to widen the study of applying the Affective Aware Pseudo Association Method in other aspects of Group Recommender Systems besides group formation and group decision-making support. The current study relied on pre-processed users' and items' emotion profiles. Future studies will investigate the real-time processing of objects' emotion profile to accommodate the new arrival of objects to the Recommender Systems.

## 7. CONCLUSION

Human preference in decision-making is highly influenced by their moods. By capturing and modeling emotion features in users and items can better reflect human preferences. This paper advocated applying the Affective Index Indication (AII) developed by the Affective Aware Pseudo Association Method (AAPAM) to obtain pairwise similarity metrics for comparing the closeness between two objects. In the study, the AII was applied to compare two users' emotion profiles UVEC to determine the closeness between users. The same technique was also applied to find the closeness between a user's emotion profile UVEC and a list of movies' emotion profiles MVEC. Provided that movies and users have precomputed with MVEC and UVEC, to make movie top-N recommendations by AII technique is straight forward. The AII generated top-N list also capable of adapting to the dynamic of the user's mood change. AII technique can support various decision-making strategies from group users and support system and user-oriented group formation.

## REFERENCES

- [1] Adomavicius, Gediminas, & Tuzhilin, Alexander, (2005) "Toward the next generation of recommender systems: A survey of the state-of-the-art and possible extensions", IEEE Transactions on Knowledge and Data Engineering, Vol. 17, No. 6, pp734–749.
- [2] Adomavicius, Gediminas, & Tuzhilin, Alexander, (2011) "Context-aware recommender systems", In Recommender systems handbook, Springer, pp217–253.
- [3] Amer-Yahia, Sihem, Roy, SenjutiBasu, Chawlat, Ashish, Das, Gautam, & Yu, Cong, (2009) "Group recommendation: Semantics and efficiency", Proceedings of the VLDB Endowment, Vol. 2, No. 1, pp754–765.



- [4] Baltrunas, Linas, Makcinskas, Tadas, & Ricci, Francesco, (2010) "Group recommendations with rank aggregation and collaborative filtering", Proceedings of the Fourth ACM Conference on Recommender Systems, pp119–126.
- [5] Berkovsky, Shlomo, & Freyne, Jill, (2010) "Group-based recipe recommendations: analysis of data aggregation strategies", Proceedings of the Fourth ACM Conference on Recommender Systems, ACM, pp111–118.
- [6] Bigdeli, Elnaz, & Bahmani, Zeinab, (2008) "Comparing accuracy of cosine-based similarity and correlation-based similarity algorithms in tourism recommender systems", Management of Innovation and Technology, 2008. ICMIT 2008. 4th IEEE International Conference On, IEEE, pp469–474.
- [7] Blondel, Vincent D, Guillaume, Jean-Loup, Lambiotte, Renaud, & Lefebvre, Etienne, (2008) "Fast unfolding of communities in large networks", Journal of Statistical Mechanics: Theory and Experiment, 2008(10), P10008.
- [8] Bobadilla, Jesús, Ortega, Fernando, Hernando, Antonio, & Gutiérrez, Abraham, (2013) "Recommender systems survey", Knowledge-Based Systems, 46, pp109–132.
- [9] Boratto, Ludovico, & Carta, Salvatore, (2010) "State-of-the-art in group recommendation and new approaches for automatic identification of groups", In Information retrieval and mining in distributed environments, Springer, pp. 1–20.
- [10] Burke, Robin, (2002) "Hybrid recommender systems: Survey and experiments", User Modeling and User-Adapted Interaction, Vol. 12, No. 4, pp331–370.
- [11] Cantador, Iván, & Castells, Pablo, (2011) "Extracting multilayered Communities of Interest from semantic user profiles: Application to group modeling and hybrid recommendations", Computers in Human Behavior, Vol. 27, No. 4, pp1321–1336.
- [12] Cantador, Iván, & Castells, Pablo, (2012) "Group recommender systems: new perspectives in the social web", In Recommender systems for the social web, Springer, pp139–157.
- [13] Crossen, Andrew, Budzik, Jay, & Hammond, Kristian J, (2002) "Flytrap: intelligent group music recommendation", Proceedings of the 7th International Conference on Intelligent User Interfaces, ACM, pp184–185.
- [14] De Pessemier, Toon, Doods, Simon, & Martens, Luc, (2012) "Design and evaluation of a group recommender system", Proceedings of the Sixth ACM Conference on Recommender Systems, ACM, pp225–228.
- [15] De Pessemier, Toon, Doods, Simon, & Martens, Luc, (2014) "Comparison of group recommendation algorithms", Multimedia Tools and Applications, Vol. 72, No. 3, pp2497–2541.
- [16] Erickson, Thomas, (2003) "Designing visualizations of social activity: six claims", CHI'03 Extended Abstracts on Human Factors in Computing Systems, ACM, pp846–847.
- [17] Garcia, Inma, Sebastia, Laura, & Onaindia, Eva, (2011) "On the design of individual and group recommender systems for tourism", Expert Systems with Applications, Vol. 38, No. 6, pp7683–7692.
- [18] Gartrell, Mike, Xing, Xinyu, Lv, Qin, Beach, Aaron, Han, Richard, Mishra, Shivakant, & Seada, Karim, (2010) "Enhancing group recommendation by incorporating social relationship interactions", Proceedings of the 16th ACM International Conference on Supporting Group Work, ACM, pp97–106.
- [19] Gfeller, David, Chappelier, Jean-Cédric, & De Los Rios, Paolo, (2005) "Finding instabilities in the community structure of complex networks", Physical Review E, Vol. 72, No. 5, 056135.
- [20] Good, Nathaniel, Schafer, J Ben, Konstan, Joseph A, Borchers, Al, Sarwar, Badrul, Herlocker, Jon, ... others, (1999) "Combining collaborative filtering with personal agents for better recommendations", AAAI/IAAI, pp439–446.
- [21] grouplens.org, (2019) "Movielens ml-20m dataset readme", Retrieved October 8, 2019, from <http://files.grouplens.org/datasets/movielens/ml-20m-README.html>
- [22] Hu, Changwei, Rai12, Piyush, & Carin, Lawrence, (2016) "Non-negative Matrix Factorization for Discrete Data with Hierarchical Side-Information", Proceedings of the 19th International Conference on Artificial Intelligence and Statistics, pp1124–1132.
- [23] Huang, Zhao, & Benyoucef, Morad, (2013) "From e-commerce to social commerce: A close look at design features", Electronic Commerce Research and Applications, Vol. 12, No. 4, pp246–259.
- [24] Jameson, Anthony, (2004) "More than the sum of its members: challenges for group recommender systems", Proceedings of the Working Conference on Advanced Visual Interfaces, ACM, pp48–54.
- [25] Jameson, Anthony, & Smyth, Barry, (2007) "Recommendation to groups", In The adaptive web, Springer, pp596–627.

- [26] Kagita, Venkateswara Rao, Padmanabhan, Vineet, & Pujari, Arun K, (2013) “Precedence Mining in Group Recommender Systems”, In *Pattern Recognition and Machine Intelligence*, Springer, pp701–707.
- [27] Kagita, Venkateswara Rao, Pujari, Arun K, & Padmanabhan, Vineet, (2013) “Group Recommender Systems: A Virtual User Approach Based on Precedence Mining”, In *AI 2013: Advances in Artificial Intelligence*, Springer, pp434–440.
- [28] Kim, Jae Kyeong, Kim, HyeaKyeong, Oh, Hee Young, & Ryu, Young U, (2010) “A group recommendation system for online communities”, *International Journal of Information Management*, Vol. 30, No. 3, pp212–219.
- [29] Kim, Young, & Srivastava, Jaideep, (2007) “Impact of social influence in e-commerce decision making”, *Proceedings of the Ninth International Conference on Electronic Commerce*, ACM, pp293–302.
- [30] Kompan, Michal, (2012) “Group Inspired Single-User Collaborative Filtering”, *Information Sciences and Technologies Bulletin of the ACM Slovakia*, Vol. 4, No. 2, pp35–37.
- [31] Konstan, Joseph A, Miller, Bradley N, Maltz, David, Herlocker, Jonathan L, Gordon, Lee R, & Riedl, John, (1997) “GroupLens: applying collaborative filtering to Usenet news”, *Communications of the ACM*, Vol. 40, No. 3, pp77–87.
- [32] Kywe, Su Mon, Lim, Ee-Peng, & Zhu, Feida, (2012) “A survey of recommender systems in twitter”, In *Social Informatics*, Springer, pp420–433.
- [33] Lee, Tong Queue, Park, Young, & Park, Yong-Tae, (2008) “A time-based approach to effective recommender systems using implicit feedback”, *Expert Systems with Applications*, Vol. 34, No. 4, pp3055–3062.
- [34] Leung, John Kalung, Griva, Igor, & Kennedy, William G., (2020a) “An Affective Aware Pseudo Association Method to Connect Disjoint Users Across Multiple Datasets – an enhanced validation method for Text-based Emotion Aware Recommender”, *International Journal on Natural Language Computing (IJNLC)*, Vol. 9, pp11–31, <https://doi.org/10.5121/ijnlc.2020.9402>
- [35] Leung, John Kalung, Griva, Igor, & Kennedy, William G., (2020b) “An Affective Aware Pseudo Association Method to Connect Disjoint Users Across Multiple Datasets – an enhanced validation method for Text-based Emotion Aware Recommender”, *International Journal on Natural Language Computing (IJNLC)*, Vol. 9, No. 4, <https://doi.org/10.5121/ijnlc.2020.9402>
- [36] Leung, John Kalung, Griva, Igor, & Kennedy, William G., (2020c) “Using Affective Features from Media Content Metadata for Better Movie Recommendation Making”, *Proceedings of the 12th International Joint Conference on Knowledge Discovery, Knowledge Engineering and Knowledge Management*, Vol. 1, pp161–168, <https://doi.org/10.5220/0010056201610168>
- [37] Leung, John Kalung, Griva, Igor, & Kennedy, William G., (2021) “Making Cross-Domain Recommendations by Associating Disjoint Users and Items Through the Affective Aware Pseudo Association Method”, *Proceedings of 8th International Conference on Artificial Intelligence and Applications (AIAP 2021)*, Vol. 11, pp113–129, <https://doi.org/10.5121/csit.2021.110108>
- [38] Lieberman, Henry, Van Dyke, Neil W, & Vivacqua, Adrian S, (1998) “Let’s browse: a collaborative Web browsing agent”, *Proceedings of the 4th International Conference on Intelligent User Interfaces*, ACM, pp65–68.
- [39] Machado, FP, (2003) “CPAR: Classification based on predictive association rules”, *IEICE Transactions on Information and Systems*.
- [40] Martínez, Ana Belén Barragáns, Arias, José J Pazos, Vilas, Ana Fernández, Duque, Jorge García, & Nores, Martin Lopez, (2009) “What’s on TV tonight? An efficient and effective personalized recommender system of TV programs”, *IEEE Transactions on Consumer Electronics*, Vol. 55, No. 1, pp286–294.
- [41] Masthoff, Judith, (2015) “Group Recommender Systems: Aggregation, Satisfaction and Group Attributes”, In *Recommender Systems Handbook*, Springer, pp743–776.
- [42] McCarthy, Joseph F, (2002) “Pocket restaurantfinder: A situated recommender system for groups”, *Workshop on Mobile Ad-Hoc Communication at the 2002 ACM Conference on Human Factors in Computer Systems*.
- [43] Melville, Prem, Mooney, Raymond J, & Nagarajan, Ramadass, (2002) “Content-boosted collaborative filtering for improved recommendations”, *Aaai/Iaai*, pp187–192.
- [44] Ntoutsis, Irene, Stefanidis, Kostas, Norvag, Kjetil, & Kriegel, Hans-Peter, (2012) “grecs: A group recommendation system based on user clustering”, *International Conference on Database Systems for Advanced Applications*, Springer, pp299–303.

- [45] O’connor, Mark, Cosley, Dan, Konstan, Joseph A, &Riedl, John, (2001) “PolyLens: a recommender system for groups of users”, ECSCW 2001, Springer, pp199–218.
- [46] O’Hara, Kenton, Lipson, Matthew, Jansen, Marcel, Unger, Axel, Jeffries, Huw, & Macer, Peter (2004). “Jukola: democratic music choice in a public space”, Proceedings of the 5th Conference on Designing Interactive Systems: Processes, Practices, Methods, and Techniques, 145–154. ACM.
- [47] Padmanabhan, Vineet, Seemala, Siva Krishna, &Bhukya, Wilson Naik, (2011) “A rule based approach to group recommender systems”, In Multi-disciplinary Trends in Artificial Intelligence, Springer, pp26–37.
- [48] Park, Seung-Taek, & Chu, Wei, (2009) “Pairwise preference regression for cold-start recommendation”, Proceedings of the Third ACM Conference on Recommender Systems, ACM, pp21–28.
- [49] Popescu, Gheorghe, & Exam, EDIC Candidacy, (2011) “Preference Aggregation in Group and Social Recommender Systems”, Signature.
- [50] Pujahari, Abinash, & Padmanabhan, Vineet, (2015) “A New Grouping Method Based on Social Choice Strategies for Group Recommender System”, In Computational Intelligence in Data Mining, Vol. 1, pp325–332.
- [51] Rashid, Al Mamunur, Karypis, George, &Riedl, John, (2008) “Learning preferences of new users in recommender systems: an information theoretic approach”, AcmSigkdd Explorations Newsletter, Vol. 10, No. 2, pp90–100.
- [52] Resnick, Paul, & Varian, Hal R, (1997) “Recommender systems”, Communications of the ACM, Vol. 40, No. 3, pp56–58.
- [53] Ricci, Francesco, Rokach, Lior, & Shapira, Bracha, (2011) “Introduction to recommender systems handbook”, Springer.
- [54] Riedl, J, &Konstan, J, (2015) “MovieLens Latest Datasets (2015) ”, Retrieved April 19, 2018, from <https://grouplens.org/datasets/movielens/latest/>
- [55] Schafer, J Ben, Konstan, Joseph, &Riedl, John, (1999) “Recommender systems in e-commerce”, Proceedings of the 1st ACM Conference on Electronic Commerce, ACM, pp158–166.
- [56] Senot, Christophe, Kostadinov, Dimitre, Bouzid, Makram, Picault, Jérôme, Aghasaryan, Armen, & Bernier, Cédric, (2010) “Analysis of strategies for building group profiles”, International Conference on User Modeling, Adaptation, and Personalization, Springer, pp40–51.
- [57] Smeaton, Alan F, & Callan, Jamie, (2005) “Personalisation and recommender systems in digital libraries”, International Journal on Digital Libraries, Vol. 5, No. 4, pp299–308.
- [58] Sprague, David, Wu, Fuqu, & Tory, Melanie, (2008) “Music selection using the PartyVote democratic jukebox. Proceedings of the Working Conference on Advanced Visual Interfaces, ACM, pp433–436.
- [59] TMDb, (2018) “TMDb About”, Retrieved May 11, 2018, from <https://www.themoviedb.org/about?language=en>
- [60] Vozalis, Emmanouil, &Margaritis, Konstantinos G., (2003) “Analysis of recommender systems algorithms”, The 6th Hellenic European Conference on Computer Mathematics & Its Applications, pp732–745.
- [61] Yu, Zhiwen, Zhou, Xingshe, Hao, Yanbin, & Gu, Jianhua, (2006) “TV program recommendation for multiple viewers based on user profile merging”, User Modeling and User-Adapted Interaction, Vol. 16, No. 1, pp63–82.

## AUTHORS

**John K. Leung** is a Ph.D. candidate in Computational and Data Sciences Department, Computational Sciences and Informatics at George Mason University in Fairfax, Virginia. He has over twenty years of working experience in information technology research and development capacity. Formerly, he worked in the T. J. Watson Research Center at IBM Corp. in Hawthorne, New York. John has spent more than a decade working in Greater China, leading technology incubation, transfer, and new business development.



**Igor Griva** is an Associate Professor in the Department of Mathematical Sciences at George Mason University. His research focuses on the theory and methods of nonlinear optimization and their application to problems in science and engineering.



**William G. Kennedy**, PhD, Captain, USN (Ret.) is an Associate Professor in the Department of Computational and Data Sciences and is a Co-Director of the Center for Social Complexity at George Mason University in Fairfax, Virginia. He has over 10-years' experience in leading research projects in computational social science with characterizing the reaction of the population of a mega-city to a nuclear WMD event being his most recent project. His teaching, research, and publication activities are in modeling cognition and behavior from individuals to societies.

