Computer Science and Information Systems 19(3):1133–1154 https://doi.org/10.2298/CSIS210925018S

Development of Recommendation Systems Using Game Theoretic Techniques

Evangelos Sofikitis[1] and Christos Makris[2]

[1] Department of Computer Engineering and Informatics, University of Patras, Rio 26500, Patras, Greece
sofikitis@ceid.upatras.gr
[2] Department of Computer Engineering and Informatics, University of Patras, Rio 26500, Patras, Greece
makri@ceid.upatras.gr

Abstract. In the present work, we inquire the use of game theoretic techniques for the development of recommender systems. Initially, the interaction of the two aspects of the systems, query reformulation and relevance estimation, is modelled as a cooperative game where the two players have a common utility, to supply optimal recommendations, which they try to maximize. Based on this modelling, three basic recommendation methods are developed, namely collaborative filtering, content based filtering and demographic filtering. The different methods are then combined to create hybrid systems. In the weighted combination, the use of game theoretic techniques is extended, as it is modelled as a *cooperative game*. Finally, the methods are combined with the use of a genetic algorithm where game theory is used for the parent selection process. Our work offers a baseline for the efficient combination of recommendation methods through game theory and in addition the novelty method, *Choice by Game*, for the parent selection process in genetic algorithms which offers consistent performance improvements.

Keywords: recommendation system, game theory, genetic algorithm.

1. Introduction

Information retrieval systems, consist of two aspects, namely the formulation of an optimal query to best represent the target user's need of information and the estimation of the documents' relevance to this query [29,16]. According to Rocchio's fundamental theory [26], the optimal query reformulation, in text retrieval, is achieved through relevance feedback [11]. In more detail, the query is reformulated through an iterative process, where the system returns results and based on the user's feedback on the results, expands the terms of the query. Depending on the individual situations, the feedback signal might be implicit, such as clicks or play-lists, or blinded, assuming the top-k returned results as relevant ones [31]. The same ground rules could be applied to recommendation systems after making some modifications. In recommender systems where there is no initial query provided, the user's need is represented by a profile based on its historical interactions s and may be inferred from other similar user profiles [30]. As far as the second aspect is concerned, the main goal is to assign relevance scores to the available documents given the query. The classic retrieval model [24] and its variations, BM25 [25], and language models [34], utilize term weighing to calculate the relevance scores.

In related work [36], based on game-theoretical analysis, a new equilibrium theory of information retrieval is proposed, according to which, the two basic aspects of Information Retrieval are correlated and participate in a cooperative game. More specifically, the query reformulation player would refine the query that is the best response to the results returned from the given retrieval model player, i.e., formulate an optimal query that would maximize its utility. Simultaneously, the retrieval model player would also need to produce the document relevant estimation that is the best response toward the formulated query. The game play provides the retrieval solutions that is in a Nash equilibrium or multiple Nash equilibria [19] when each becomes the best response to the other. In the recommendation systems setting, this novelty equilibrium theory is evaluated through a practical implementation in collaborative filtering tasks and the experiments show that, it outperforms the query reformulation and retrieval model reformulation, when applied separately.

In this paper, in order to confirm that, the modelling of the recommendation process as a cooperative game offers performance improvements, we apply this equilibrium theory [36] in more recommendation methods, namely collaborative filtering, demographic filtering, content based filtering and hybrid combinations of the three [7], while scaling the experiments in a larger dataset. The experiments show that the initial claim [36] is confirmed for the variations of the recommendation methods, however the same does not apply for the scaled experiments. Thereinafter, we extend the integration of game theory in recommendation systems by developing novelty algorithms. An algorithm for the weighted combination of two recommendation methods is developed, where the weights of every method derived from a cooperative game [6]. The methods are the players of the game and they are, initially, assigned with a weight. During the game, which is an iterative process, every player chooses a strategy, i.e. increase, or decrease its weight to maximize the common utility. An equilibrium is reached when the two players can not increase their utility by altering their weights. This algorithm is evaluated with experiments in two datasets where, although different combinations of methods are attempted, the weighed combination through game excels in performance compared to the individual methods combined. Finally, game theoretic techniques are also applied in genetic algorithms, where the parent selection process is modelled as a game. Two user-based recommendation methods are executed, and the resulting top-k users are combined to create an initial population for the genetic algorithm. In the method Choice by Game, the probability of two individuals being selected as parents, is calculated through a game and is proportional to the utility of the combination. The results of the experiments in two datasets show that, the Choice by Game method offers consistent performance improvements when compared to the existing Choice by Roulette method and to the individual methods from whom derived the initial population for the genetic algorithm.

Although a simple linear retrieval model is used for the basic recommendation methods, our work offers a baseline for the efficient development of hybrid recommendation systems through game theory, which could be applied to more refined systems. Utilizing our algorithms, two recommendation methods can be combined linearly or through a genetic algorithm resulting to a hybrid that provides more qualitative recommendations than the individual methods. Furthermore, the *Choice by Game* method can be used not only in recommendation systems, but in the various applications [27] of the genetic algorithms.
Briefly, the paper is organized as follows. In the following section (Section 2), we con-

duct a brief review of and how has game theory utilized to produce more efficient and improved recommender systems. Then, in Section 3, we present the baseline of previous research on the modelling of the recommendation process as a cooperative game. In Section 4, we elaborate on the basic recommendation methods that were developed using this modelling. The integration of game theory and recommendation systems is extended on Section 5, where we introduce two algorithms for the efficient combination of two recommendation methods, namely linearly and through a genetic algorithm. Both algorithms utilize game theoretic techniques. In Section 6 the practical implementation used for the experimental evaluation of all the above algorithms is explained, while in Section 7 the results of the experiments in two datasets are presented. In the epilogue of this paper (Section 8), the conclusions of our inquiry is discussed along with suggestions for future research.

2. Related Work

2.1. Recommender Systems

On the Internet, there is an overabundance of options, choices and products, rendering it impossible for users to browse and evaluate all of them, in order to choose the most desirable ones. In our daily lives we rely on the recommendations of others, such as friends or critics, for the selection of movies, restaurants, products, etc. On the Internet, this role is carried out by the recommendation systems [5]. These systems are a subset of information filtering systems and aim to present the most relevant products from a set, making predictions about users' preferences. In the initial stages of recommender systems, the collaborative filtering method was developed, where recommendations are based on the preferences of users who are relevant to the target user. Other methods such as demographic filtering and content based filtering were later developed [23]. The demographic filtering, like the collaborative filtering is based on users' similarity while content based filtering is based on item similarity, meaning that the systems suggest to the user items similar to those they have already evaluated and rated highly [17]. The ever-increasing need for efficient recommendations has led to more sophisticated systems that often utilize two or more methods, these are hybrid recommender systems [3], or draw on methods and techniques from other scientific fields [35]. The evolution of recommendation systems has led to extending their usage in other fields other than improving users' online experience, even in the education [20]. State of the art recommendation systems utilize neural networks and matrix factorization to develop better performing systems [22]. Hierarchical Recurrent Neural Networks are currently one of the most efficient methods to achieve felicitous recommendations [21]. In our research, however, we do not utilize such sophisticated methods but rather, try to find ways to efficiently combine recommendation methods and techniques.

2.2. Recommender Systems and Game Theory

Game Theory is the scientific field that studies games, which are interdependence situations of players [18]. Techniques from this field are widely applied in various scientific fields with machine learning being one of the latest. The combination of recommender systems and game theory is an innovative field of research with a relatively limited literature.

However, researchers have already utilized game theoretic techniques to various aspects of recommender systems. Such techniques have been used to balance accuracy and coverage of recommendations through rough sets [2], to counterpoise profit of strategic content providers to application usability [4] and to find an equilibrium between qualitative recommendations and data privacy [10]. Moreover, game theory has been used to locate trustworthy users in a set more efficiently and consequently provide more accurate recommendations [1] and even to provide a novel formulation of the recommendation process i.e., as a cooperative game between the user and the systems enhancing the process as a whole [33]. In our work, we follow a more straight-forward approach, as we attempt to provide more efficient and qualitative recommendations through game theoretic techniques.

3. Baseline of Previous Approach

As mentioned in the previous section, any retrieval process consists of two main aspects. The first aspect is to formulate the query to best represents the user's need for information. The second aspect is to calculate the relevance of the available documents or items to the query and select the most relevant ones. In recommendation systems instead of a query, a profile is used to represent the preferences of the target user. Since there is no initial query, the profile is based entirely on the user's historical interactions with the products, for example their ratings and comments. The goal of the retrieval models remains the same, that is, to estimate the relevance of available items to the profile and select the items that are more likely to satisfy the target user. When it comes to text retrieval, the classic models assign weights to query terms so that each word has a different weight when calculating relevance. Term weighting can also be adapted, in recommendation systems, utilizing past interactions as terms. Although recommendation systems are a subset of information filtering systems, they can be formulated using information retrieval techniques. In this case in particular, instead of filtering the non-relevant users, in every iteration of the algorithm the users are classified as relevant and non-relevant and both sets contribute to the query and retrieval parameters reformulation. Namely, both offer information to the system in order to conclude to the recommendations.

3.1. Modelling

According to related research [36], for the modelling of the recommendation process as a game, the profile of the target user q and a set of objects D are defined. Each object d_i of the set D can be a product, a service, or another user, since the recommendations for the target user can be derived from the relevant users. The profile and each object of the set are represented by a vector of attributes. The attributes vary depending on the recommendation method used. The recommendation process can be modelled as a cooperative game with three key elements: players, their strategies, and profit functions. The game is collaborative, as the common goal of the two players and the means of maximizing their utilities is to provide qualitative recommendations.

Definition 1. *(Game of Recommendation Process)*
The game is defined as follows:

- *Players: the query reformulation Q and the retrieval model reformulation M,*
- *Strategies: S_Q and S_M are two finite sets of strategies, which are available to players Q and M, respectively. A strategy $s_Q \in S_Q$ can for example be the weight gain of an item in the user profile which is more relevant than the rest of the items to better represent their user preferences,*
- *Utility Functions: u_Q and u_M define the utilities of players Q and M respectively and depend on the players' strategies. Equilibrium occurs when both players have no motivation to change strategy. That is, when a unilateral change of strategy will reduce the profit of the player who changed.*

The utility of the retrieval model M depends on the successful distinguishing the relevant objects from the non-relevant ones.

$$\begin{aligned} u_M(s_Q, s_M) =& \frac{1}{|D_r|} \sum_{d_i \in D_r} \log p(r = 1|d_i, q; w) - \\ & \frac{1}{|D_n|} \sum_{d_i \in D_n} \log p(r = 0|d_i, q; w) \end{aligned} \tag{1}$$

where:

- D_r, D_n are the sets of relevant and non-relevant items respectively,
- w is the weight vector of the retrieval model for each attribute,
- p is the probability that the object is relative $(r = 1)$ or non-relevant $(r = 0)$ given the profile q, the object d_i and the weights w

The probability is calculated using the sigmoid function

$$s(x) = \frac{1}{1 + e^{-x}} \tag{2}$$

where:

$$\sum_{j=1}^{N} w_j q_j d_{ij} \tag{3}$$

j being an element of the characteristic vectors of length N, x is the inner product of the vectors w, q, d_i. The gain of M increases if an object belongs to the relevant, i.e. $d_i \in D_r$ and the model has a high probability of distinguishing it as so and vice versa, decreases if the probability is low. The same applies to non-relevant items. Therefore, in the iterative process of reshaping the weights w, the strategy that appropriately shapes the probabilities and maximizes its profit is chosen. On the other hand, the utility of player Q is based on the feedback from the retrieval model

$$\begin{aligned} u_Q(s_Q, s_M) =& \frac{1}{|D_k|} \sum_{d_i \in D_k} \log p(r = 1|d_i, q; w) - \\ & \frac{1}{N - |D_k|} \sum_{d_i \notin D_k} \log p(r = 0|d_i, q; w) \end{aligned} \tag{4}$$

where:

- D_k are the top-k objects as classified, based on relativity, by the recovery model
- N is the set of all available objects

The method of calculating the probability remains the same. The player Q also tries to configure the probabilities appropriately in order to maximize its utility with the difference that its available strategies aim at reformulating the profile q instead of w. When there is no apriori knowledge of the relevant and non-relevant objects, we assume the top-k as relevant and the others as non-relevant. In that case, the players Q and M share a common utility function.

3.2. Cases: ConvQ - ConvM – Game

For the application of the above modelling of the recommendation process as a game, three cases are distinguished [36], the reformulation of the user's profile, the reformulation of the weights of the retrieval model and the simultaneous reformulation of both. Let q be the target user's profile, d_i an object of the set d, w the weights of the retrieval model and θ_i the relevance score of the object d_i to the profile q. In all three cases, players try to maximize their utility using gradient ascent. Gradient ascend is an iterative algorithm for finding the maximum of a differential function. In the case of *Retrieval Model Reformulation*, θ_i is calculated as follows:

$$\theta_i = sigmoid(q^\top w d_i) \tag{5}$$

while in the case of *Profile Reformulation*, the calculation method of θ_i is defined as:

$$\theta_i = sigmoid(q^\top d_i) \tag{6}$$

where $q^\top$ is the inverse vector of q, $q^\top w d_i$ is the inner product of q, w, d_i and respectively, $q^\top d_i$ is the inner product of q and d_i.

Case 1: Profile Reformulation In this case, the player Q tries to maximize its utility u_Q by reformulating the profile q while the weights w remain constant.

$$\begin{aligned} \frac{\partial u_Q(s_Q, s_M)}{\partial q} =& \frac{1}{|D_k|} \sum_{d_i \in D_k} (1-\theta_i) d_i - \\ & \frac{1}{N-|D_k|} \sum_{d_i \notin D_k} \theta_i d_i \end{aligned} \tag{7}$$

$$q \leftarrow q + \eta \frac{\partial u_Q(s_Q, s_M)}{\partial q} \tag{8}$$

q is updated with gradient ascent as shown in the above equations Eq. (8) and Eq. (10), where η is the learning rate. We call this case *ConvQ*.

Case 2: Retrieval Model reformulation Similarly, in this case player M tries to maximize its utility u_M by reformulating the weights w while the profile q remains constant. w is updated using the gradient ascent as shown in the following equations, where η is the learning rate. We call this case ConvM.

$$\begin{aligned}\frac{\partial u_M(s_Q, s_M)}{\partial w} = & \frac{1}{|D_r|} \sum_{d_i \in D_r} (1-]\theta_i) q d_i^\top - \\ & \frac{1}{|D_n|} \sum_{d_i \in D_n} \theta_i q d_i^\top\end{aligned} \tag{9}$$

$$w \leftarrow w + \eta \frac{\partial u_M(s_Q, s_M)}{\partial w} \tag{10}$$

Case 3: Game In the third case, which we call Game, both q and w change. Player Q changes their strategy and reshapes the profile q in response to the strategy chosen by M, who does the same by updating w based on the reformulation of q. Their interaction is a collaborative game.

4. Equilibrium of the Recommendation Process

The modelling of the recommendation process as a cooperative game is product of related work [36], in which, it was applied and evaluated on the collaborative filtering method. We expand the usage of this methodology to other two fundamental recommendation methods and four hybrid combinations of them.

4.1. Collaborative Filtering

In the Collaborative Filtering algorithm, the set of objects D consists of user profiles and each element of the vectors d_i and q represents a product from the set of available products to be recommended. The value of every elements is the rating of each user for the corresponding product. The recommendation system identifies the users $d_i \in d$ who are most relevant to user q, i.e. users who have similar ratings to the target user. The collaborative filtering method is based on the assumption that two users who have rated a set of products similarly, will have the same satisfaction or dissatisfaction from products that have not yet rated [8]. Therefore, the user for whom the recommendations are intended is likely to be satisfied with products that the relevant users have rated highly. The aim of the system is to predict the user's ratings for the products that he has not evaluated and to present to him the products for which he has provided a high rating. To do this, the system first identifies the k most relevant users. K is a number smaller than the total number of users and varies depending on the application. The predicted rating for each product is the average ratings of the top-k users for that product. The products with the highest predicted rating are then recommended to the target user.

4.2. Demographic Filtering

The demographic filtering method is based on the assumption that users with similar demographics will have the same preferences on products. In this algorithm, the profile q, and the objects d_i of set D are still user profiles, for the representation of which vectors are used. The elements of each vector represent a demographic characteristic, such as age, gender, occupation, place of residence and more. The same steps as described above for the collaborative filtering algorithm are followed, to generate recommendations.

4.3. Content Based Filtering

For the content based filtering algorithm the d_i objects are products and the elements of the vectors representing each $d_i \in D$ are characteristic of the products. The user q profile is also a feature vector where each attribute has a value according to the preferences of the user. The profile is created based on previous ratings of the user. More specifically, we consider a set of N products $P \subset D$, each product $p_i \in P$ for $i \in [1, N]$ has been rated by the user with r_i. Each of these products is represented by a vector of attributes c and $p_{i,j}$ is the value of the product for the attribute $c_j \in c$. The element q_j in the profile q represents the preferred value of the attribute c_j by the target user, and it is calculated by the formula:

$$q_j = \frac{\sum_{i=1}^{N} p_{i,j} * r_i}{\sum_{i=1}^{N} r_i} \tag{11}$$

So instead of the bipartite user to item graph an item to item graph is utilized to result in the recommendations for the target user.

4.4. Hybrid Methods

Hybrid1 and Hybrid2 algorithms are hybrid recommendation algorithms that use the above methods. They are serial filtering algorithms, in the sense that the algorithms are executed sequentially, and the results of the first are used to reinforce the recommendations of the second. The cases developed and presented below are combinations of collaborative filtering and demographic filtering or content based filtering.

Hybrid1.1: Demographic Filtering after Collaborative Filtering The Hybrid1.1 algorithm is the combination of collaborative filtering and demographic filtering. It is a simple algorithm that identifies users who have similar demographic characteristics to the target user, among users who have already been judged to be relevant based on their ratings. Running collaborative filtering results in a set of users who are relevant to the target user. This set is then used in demographic filtering to generate the k final users from whom the recommendations derived.

Hybrid1.2: Collaborative Filtering after Demographic Filtering Hybrid1.2 combines the two algorithms like Hybrid1.1 but in a different order. More specifically, running demographic filtering creates a set of relevant users from which, after collaborative filtering, the top-k most relevant are chosen. Essentially, the Hybrid1 algorithms apply double filtering on the set of users D. Once with demographic criteria and once with their product ratings.

Hybrid2.1: Content Based Filtering after Collaborative Filtering In the combination of content based filtering and collaborative filtering where collaborative filtering is performed first, the procedure is as follows, the system performs collaborative filtering from which the top-k users derived. Then, a set of products is created, containing the products that are highly rated by the top-k users. This set is used for the content based filtering algorithm. In other words, the system selects the products that the relevant users liked and filters them for the final recommendations.

Hybrid2.2: Collaborative Filtering after Content Based Filtering In the case of Hybrid2.2, first content based filtering is performed resulting in the top-k products, which are more relevant to the target user. These products are given more weight to the calculation of the most relevant users when performing collaborative filtering. More specifically, in q and d_i, which are vectors and each of their elements represents a product, a second elements is created for the representation of each of the top-k products. Thus, these products have a double weight and affect the relevance score of users that collaborative filtering will identify.

5. Game Theory in Hybrid Systems

In this section, we present two methods of integrating game theory to recommendation systems. Initially, the linear combination of two recommendation methods is modelled as a game. In the second hybrid system, two methods are combined with a genetic algorithm and the game takes place in the selection stage where the novelty method *Choice by Game* is used.

5.1. Equilibrium in Linear Combination

The Hybrid3 algorithm is a hybrid algorithm of linear combination of two recommendation methods. The algorithms of each combination run simultaneously and independently and each produces a score vector according to its predictions. Each element of the vectors corresponds to a product and the value of the element is the rating that the algorithm predicted for it. These vectors are combined linearly to obtain the final ratings predictions. If R_A, R_B are the predicted ratings and α, β are the weights of the algorithms A, B respectively, then the final prediction of the ratings of a product, R results from the combination as follows:

$$R = R_A * \alpha + R_B * \beta \tag{12}$$

Initially the weights have values $\alpha = 1$ and $\beta = 0$ which change while their sum remains constant $(\alpha + \beta = 1)$. The final weights occur after a cooperative game.

Definition 2. *(Game of Recommendation Methods Combination)*
The game is defined as follows:

- *Players: A, B are the algorithms that are combined to produce recommendations and have initial weights α, β respectively*
- *Strategies: s_A and s_B for players A, B respectively are to reduce or increase their weights*

– *Utility function: u is common for both players and they try to maximize it. A state of equilibrium occurs when neither of the players benefits from changing their strategies or when one of the two weights reaches a certain upper limit.*

More specifically, let r_A, r_B be the vectors of estimated scores of the algorithms A, B respectively and q the target user's profile. The scores r_A, r_B are combined according to equation Eq. (12) resulting to the vector r. The players' common utility is defined as the correlation of the vectors r and q. The method for calculating correlation may differ in applications. In each round of the game the weights are renewed, α decreases and β increases by a constant c. Then the score vector r is recalculated with the updated weights. At the end of each iteration the new utility u is calculated. The game ends when the utility decreases or when $\beta = 1$ and $\alpha = 0$.

5.2. Equilibrium in Genetic Algorithm

Collaborative filtering and demographic filtering identify the top-k most relevant users to the target user. These users can be represented as vectors every element of which corresponds to a product and the value of the element is the user rating for this product. These vectors can be used as the initial population for a genetic algorithm. Each user is an individual of the initial population and each element of the vector that describes them is a chromosome. The genetic algorithm is performed iteratively and in each iteration, the fittest individuals, i.e. those who are more relevant to the target user in terms of their ratings, pass on to the next generation or reproduce and their offspring pass on to the next generation. The fitness function may differ depending on the characteristics of each application. The stage of selection is modelled as a cooperative game.

Definition 3. *(Game of Individual Selection)*
The game is defined as follows:

– *Players: A, B are the parents, who will participate in the reproduction*
– *Strategies: s_A and s_B are the individuals whose chromosomes will be combined to produce an offspring*
– *Utility function: u is common to both players and they try to maximize it. Each pair of strategies has a chance to be chosen, which is proportional to the utility it offers.*

More specifically, let D be the population of the genetic algorithm and d_i, d_j two individuals of the population. If the fitness of the individuals is based on their correlation to the target user's profile q, calculated by a function f, namely $fitness(d_i) = f(d_i, q)$ and $fitness(d_j) = f(d_j, q)$ then the shared utility of the players for choosing this pair of individuals is calculated as follows:

$$u(d_i, d_j) = (fitness(d_i) + fitness(d_j) * f(d_i, d_j)) \tag{13}$$

In each generation the utility for every combination of individuals is calculated and the probability of a combination being selected for reproduction is the utility of this combination to the sum of the utilities of all combinations. The goal of the algorithm is to create a population with a higher fitness than the initial, so the algorithm terminates when the sum of the fitness of the new generation is lower than the previous one or when a predetermined number of generations is exceeded. The recommendations derived from the final population, as described above for the collaborative filtering algorithm.

The following simple example is presented to illustrate the algorithm. Let a target user u_T and three users u_1, u_2 and u_3 who represent the choices of the two players A and B i.e. parents in the crossover stage of the genetic algorithm. The parents aim to select the most dominant users, those who, when reproduced will result in an offspring that is closest to the target user. Every user is represented by six chromosomes c_i for $i = 1, \ldots, 6$ which are their rating to a corresponding movie. In this example we will use euclidean distance to calculate the fitness of every user, so this is now a simple minimization problem.

Table 1. Choice by Game Example.

	c_1	c_2	c_3	c_4	c_5	c_6
u_T	0	3	2	4	2	1
u_1	1	3	2	2	4	2
u_2	2	2	3	4	3	1
u_3	1	5	1	5	2	1

Chromosomes of Individuals

		Player B		
		u_1	u_2	u_3
	u_1	-	17.42	25.31
Player A	u_2	17.42	-	21.16
	u_3	25.31	21.16	-

Costs of Strategies

The chromosomes of every individual are presented in the table on the left. Respectively, the cost of the players' choices, calculated using equation 13, are presented in the table on the right. The players aim to minimize their cost i.e. the distance to the target user. The lower the cost the higher the probability of a user combination to be chosen for the crossover stage. Of course the element of randomness exist in every stage of a genetic algorithm, however for the sake of this example we suppose that the most dominant pairs are chosen for the crossover stage. In this example the most dominant pair of users are (u_1, u_2) and (u_2, u_3) with approximate costs of 17.42 and 21.16 respectively, so these two pair of choices will have a higher probability to be chosen and their offspring will proceed to the next generation. The crossover and mutation methods as well as the probabilities thereof depend on every problem's characteristics.

6. Implementation

The evaluation of the algorithms was based on two sets of experiments, for which the MovieLens 100k and MovieLens 25M datasets were used [12].

6.1. MovieLens 100k

The MovieLens 100k dataset contains:

- 100,000 ratings from 943 users for 1,682 movies,
- demographic data for each user which includes age, gender, occupation, and zip code

Pre-processing The collaborative filtering and demographic algorithms were implemented and evaluated for this dataset using user ratings and demographic data, respectively. Preprocessing is performed so that the data takes the appropriate form to run the algorithms.

For the collaborative filtering, user profiles are created, which are represented by vectors of ratings. Users have not rated all the movies in the set, so the missing ratings are initialized with 0. Next step in the preprocessing is the test-train Split, which is done in a ratio of 1 to 3, i.e. 75% is the set of users from which the most relevant users derived and the remaining 25% of users are used to evaluate the algorithms. From each user of the test set, we keep the 75% of the ratings as its history and the remaining 25% for evaluating the recommendations of each algorithm. For the demographic filtering, a different procedure is followed. The data is already in the form of user profiles, so the first step is omitted, however another pre-processing is required. Some of the demographics are categorical variables, so they are converted to numeric. In addition, unlike ratings, each feature in demographic profiles has a different scale, so we normalize the values of each of the four features on a scale of 0 to 1. Then apply train-test split with a ratio of 1 to 3 as described above.

Experiments For every one of the collaborative filtering, demographic filtering, Hybrid1.1 and Hybrid1.2 algorithms we distinguish the three cases ConvQ, ConvM and Game, which have been described above. For each user of the test set we execute the algorithms and their cases to compare the results. For the Hybrid3 and Hybrid4 algorithms the cases differ. More specifically, for the linear combination algorithm, Hybrid3, we compare the case in which the weights of the combined algorithms result from a cooperative game with the case where the weights, α and β, are constant. Respectively for the Hybrid4 genetic algorithm we consider two cases, in which the method of selecting parents for reproduction differs. We compare the selection method Choice by Game with the existing method Choice by Roulette.

Collaborative Filtering: after selecting the target user, the three cases ConvQ, ConvM and Game are executed sequentially. For the ConvQ case, we first calculate the relevance of the users in the train set to the target user. We consider top-k users as relevant and the rest as non-relevant. The parameter k has been given the value 100, since after experiments it showed the best results. The user profile is then reformulated according to equation Eq. (8). The learning rate η, is set to 0.1. This process is repeated until the profit converges to a maximum, using the threshold 10^{-4}. In most cases, the utility function converges before 50 iterations, so the profile is updated 50 times. The learning rate, the ratio of convergence and the iterations are based on precedents experiments [36]. In the case of ConvM the same procedure is followed except that the retrieval model is updated according to equation by Eq. (10). Finally, for the Game, the profile and the retrieval model are reformulated simultaneously in order to maximize their common utility. In each iteration, player Q chooses the best strategy in response to the strategy of M, who does the same. When neither player has an incentive to change their strategy, i.e. to reformulate the profile or retrieval model respectively, equilibrium occurs, and the profit function converges to a maximum.

Demographic Filtering: The same procedure as collaborative filtering is followed, except that instead of rating profiles, the top-100 users are calculated based on demographic profiles. The iterations for the convergence of the utility function with the gradient ascent method are reduced to 25 from 50. The demographic profiles consist of 4 characteristics versus the $1,682$ characteristics of the rating profiles, therefore faster convergence can be achieved. At the end of each case the demographic profiles are replaced with the corresponding rating profiles to calculate the predicted ratings.

Hybrid1.1 & Hybrid1.2: The Hybrid1.1 algorithm combines collaborative filtering and demographic filtering methods to perform double filtering. More specifically, collaborative filtering is initially executed to identify the top-200 most relevant users, who are then filtered again using the demographic filtering method resulting to the top-100 final users. The Hybrid1.2 algorithm is similar to Hybrid1.1 except that the demographic filtering precedes the collaborative filtering.

Hybrid3: The Hybrid3 algorithm is a linear combination of collaborative filtering and demographic filtering. The individual algorithms are executed, and the predicted ratings are calculated. Then, for each movie, the weighted average of the two ratings is calculated using as weights the α, β, which have the initial values 1 and 0, respectively. The utility function is calculated with the inverse hyperbolic sine of the inner product of the predicted ratings and the target user's profile q

$$u = sinh^{-1}(r^\top q) \tag{14}$$

The next step is to renew the weights, subtracting 0.01 from α and adding it to β, i.e. $\alpha = 0.99$ and $\beta = 0.01$. This process is repeated until the utility is reduced or β has a value of 1 and α 0. To evaluate the above algorithm, we also implement a simple linear combination, where the weights have a constant value of 0.5, i.e. the ratings are derived from the average of the individual scores.

Hybrd4: The Hybrid4 algorithm combines the results of collaborative filtering and demographic filtering. Once the two algorithms are executed, their top-100 users are used as the initial population of the genetic algorithm. For each individual, there is a possibility of mutation in which a chromosome randomly takes on a value. The probability of reproduction and mutation is 0.7 and 0.05 respectively. During reproduction, the two-point crossover method is used. The algorithm terminates when the generation limit is exceeded, which is set to 100, or when the sum of the fitness of the individuals of the current generation is lower than that of the previous one. The fitness of an individual d_i is the inverse hyperbolic sine of the inner product of d_i and the target user's profile q multiplied by the correlation of q and d_i calculated by the Spearman's ρ correlation

$$fitness(d_i) = sinh^{-1}(q^\top d_i) * \rho(d_i, q) \tag{15}$$

To evaluate the efficiency of the Choice by Game method, the same algorithm is implemented with the Choice by Roulette method and their results are compared.

6.2. MovieLens 25M

The implementation of the algorithms for the MovieLens 25M data set is similar to that for the MovieLens 100k dataset. The main difference is due to the size of the second dataset, which makes it more difficult to handle, while the methods used for the first dataset are prohibitive, due to time complexity. In addition, the MovieLens 25M dataset does not contain demographic characteristics but movie-tag correlations, which describe the content of the movies. Therefore, content based filtering is implemented instead of demographic filtering. The MovieLens 25M set contains:

- 25,000,095 ratings from 162,541 users for 62,423 movies
- 1,093,360 movie correlations with 1,128 tags

Pre-processing As mentioned above, this data set is significantly larger than the previous one, so the same preprocessing methods cannot be used. To address this, instead of representing profiles as vectors, an inverted index is used. The same procedure is followed for the train-test split as well as for the user ratings in the test set. The preprocessing required for content based filtering is to create profiles for the movies based on the movie-tag correlations. In addition, a corresponding profile should be created that presents the preferences of the target user.

Experiments For the collaborative filtering, content based filtering, Hybrid2.1 and Hybrid2.2 algorithms the three cases ConvQ, ConvM and Game are examined while for the Hybrid3 and Hybrid4 the same cases described for the MovieLens 100k are used.

Collaborative Filtering: The collaborative filtering algorithm is the same as in the MovieLens 100k dataset. For every target user, the set of available users from which the top-100 are retrieved, is limited to the 1,000 most relevant users, instead of using the whole train set, for time efficiency.

Content Based Filtering: this algorithm follows the same procedure as collaborative filtering and top-750 most relevant movies are retrieved. The predicted ratings derived from their degree of relevance to the target user.

Hybrid2.1: Firstly, the collaborative filtering algorithm is executed, and the top-100 most relevant users are retrieved. From each of these users the 10 movies with the highest rating are selected to create a set of 1,000 movies. Content based filtering is applied to this set to retrieve the top-750 movies for the user.

Hybrid2.2: The top-750 most relevant movies for the target user are retrieved from the content based filtering. Then, the user's profiles are modified adding a duplicate element for each of the top-750 movies. Doing so, these movies affect the top-100 user retrieved from the collaborative filtering.

Hybrid3: For the MovieLens 25M, collaborative filtering and content based filtering are combined linearly. The utility function based on which the players of the game renew their weights is the inverse hyperbolic sine of the inner product of the predicted ratings r and the target user's profile q, multiplied by the Spearman's ρ of r and q.

Hybrid4: For the Hybrid4 algorithm, in this dataset we use as initial population the top-100 users from the collaborative filtering and Hybrid2.2 algorithms. The same comparison is made between the parent selection methods, i.e. selection by game and selection by roulette. As a fitness function, which determines the individuals of each generation, the inverse hyperbolic sine of the inner product of the vectors describing the target user and the individual of the population, is used. The probability of reproduction and mutation and the maximum number of generations remain the same.

7. Results

After the algorithms are executed, we evaluate the results [13]. The metrics used for the evaluation are Normalized Discounted Cumulative Gain and Precision for the first 10 and the first 30 recommendations, Mean Average Precision and Mean Reciprocal Rank i.e. NDCG@10, NDCG@30, P@10, P@30, MAP and MRR [28]. For the dataset MovieLens 100k experiments are run for the entire test set, while in the MovieLens 25M dataset the experiments for the "Game of Recommendation Process" algorithms were very time consuming so experiments could not be performed for the entire test set. However, the results are reliable, because we focus on the comparison of the algorithms and not their actual scores on the metrics. The comparisons are confirmed with t-test for each algorithm [15]. For the collaborative filtering, demographic filtering, content based filtering, Hybrid1 and Hybrid2 algorithms we compare the cases ConvQ, ConvM and Game. For Hybrid3 we compare the cases Game and Simple, i.e. the determination of weights for the linear combination with game and the simple linear combination respectively as well as the results of the Game with the individual algorithms that are combined. Respectively, for the Hybrid4 we comp-selecting parents for reproduction, we also compare the Game with the individual algorithms as for Hybrid3. The best value of each algorithm for the respective metric of evaluation is written in bold.

It should be noted that, the Content Based Filtering algorithm tested on the MovieLens 25M dataset returns 750 movies out of 62,423 movies as recommendations for the user. Having predicted ratings for only 750 movies, it is difficult to evaluate the algorithm as it is very likely that the target user has rated only a few of them or even none. This is a common problem in recommendation systems address by many elaborate solving methods [14,32]. However, we resolve in a simplified method, during the evaluation we delete the movies that the user has not rated to easier evaluate the algorithms. This affects the evaluation metrics greatly, however the absolute values of the metrics used in the evaluation are not relevant. Rather we focus on the comparison of the algorithms to determine which is more efficient and since this method is used in every algorithm their relative efficiency is not altered.

7.1. MovieLens 100k

Table 2. Collaborative filtering results for the MovieLens 100k dataset.

	NDCG@10	NDCG@30	P@10	P@10	MAP	MRR
ConvQ	0.80551	0.86006	0.54746	0.44802	**0.43742**	0.25880
ConvM	0.80566	0.86012	0.54788	0.44689	0.43296	0.26080
Game	**0.80815**	**0.86205**	**0.55000**	**0.44845**	0.43737	**0.26085**

The Game case of collaborative filtering shows better results than ConvQ and ConvM in five out of six metrics, while the ConvQ case shows better performance only based on the metric MAP. Overall, the Game case is the best in the collaborative filtering algorithm. These results are in accordance to previous experiments [36].

Table 3. Demographic filtering results for the MovieLens 100k dataset.

	NDCG@10	NDCG@30	P@10	P@10	MAP	MRR
ConvQ	0.77903	0.83975	**0.53771**	**0.44562**	0.41965	0.26056
ConvM	0.76436	0.83055	0.53432	0.44251	0.41149	0.26079
Game	**0.77918**	**0.84131**	0.53602	0.44477	**0.42429**	**0.26353**

In the demographic filtering algorithm, the Game case performs better on all metrics except Precision, for the first 10 and 30 results. So, overall, the Game case outperforms the other cases.

Table 4. Hybrid1.1 filtering results for the MovieLens 100k dataset.

	NDCG@10	NDCG@30	P@10	P@10	MAP	MRR
ConvQ	**0.77903**	0.83975	0.53771	0.44562	**0.41965**	0.26056
ConvM	0.77795	0.84034	0.53898	0.44605	0.42010	**0.26070**
Game	0.77880	**0.84036**	**0.53898**	**0.44675**	0.41920	0.25979

In the Hybrid1.1 algorithm, the Game case performs better for the NDCG@30, P@10 and P@30 metrics, the ConvQ case for the NDCG@10 and MAP metrics, while the ConvM case outperforms only the metric MRR. So, we can assume that the Game case is the best of the three.

Table 5. Hybrid1.2 filtering results for the MovieLens 100k dataset.

	NDCG@10	NDCG@30	P@10	P@10	MAP	MRR
ConvQ	0.78641	0.84782	**0.54110**	**0.44718**	0.42750	**0.26964**
ConvM	0.78395	0.84637	0.53898	0.44647	0.42691	0.26478
Game	**0.78676**	**0.84794**	0.54068	0.44703	**0.42839**	0.26854

In Hybrid1.2 algorithm, Game shows better results on the metric NDCG@10, NDCG@30 and MAP, while ConvQ on P@10, P@30 and MRR. So, these are the two best cases, but we cannot say that one is outperforms the other.

Below, in Table 6, we compare the results of the Game case of the Hybrid3 algorithm, with the results of the individual algorithms that are combined, i.e. the Game cases of Collaborative Filtering (CF) and Demographic Filtering (DF). We also compare the two cases, Game and Simple, of Hybrid3. The aim is to determine whether the linear combination yields better results than the two algorithms and whether the combination using a game is more efficient than a simple combination. The results show that the Collaborative Filtering method performs better according to the metrics NDCG@10 and NDCG@30, the Demographic Filtering in the metric MRR, while the linear combination of the two shows better results in the metrics P@10, P@30 and MAP, therefore we can conclude that the Hybrid3 algorithm excels the two individual algorithms combined. In addition,

Table 6. Hybrid3 results for the MovieLens 100k dataset.

	NDCG@10	NDCG@30	P@10	P@30	MAP	MRR
CF	**0.80815**	**0.86205**	0.55000	0.44845	0.43737	0.26085
DF	0.77918	0.84131	0.53602	0.44477	0.42429	**0.26353**
H3 (Game)	0.80795	0.86159	**0.55042**	**0.44859**	**0.43741**	0.26183

	NDCG@10	NDCG@30	P@10	P@30	MAP	MRR
H3 (Simple)	0.79729	0.85371	0.54576	0.44788	0.43226	**0.27150**
H3 (Game)	**0.80795**	**0.86159**	**0.55042**	**0.44859**	**0.43741**	0.26183

the linear combination using game is more efficient than the simple linear combination, as it shows better results in all metrics except MRR.

Table 7. Hybrid4 results for the MovieLens 100k dataset.

	NDCG@10	NDCG@30	P@10	P@30	MAP	MRR
CF	0.80815	0.86205	0.55000	0.44845	0.43737	0.26085
DF	0.77918	0.84131	0.53602	0.44477	0.42429	**0.26353**
H4 (Game)	**0.81118**	**0.86233**	**0.55127**	**0.44972**	**0.43739**	0.26190

	NDCG@10	NDCG@30	P@10	P@30	MAP	MRR
H4 (Roulette)	0.77347	0.83472	0.54195	0.44421	0.42848	0.25773
H4 (Game)	**0.81118**	**0.86233**	**0.55127**	**0.44972**	**0.43739**	**0.26190**

To evaluate Hybrid4 we make the same comparison as for Hybrid3. As shown above, on Table 7, when comparing the genetic algorithm, Hybrid4, with the individual collaborative and demographic filtering algorithms, which are combined, the genetic algorithm excels in all metrics, except MRR. Similarly, in the comparison of the different parent selection methods, the Choice by Game method show a consistent performance improvement over Choice by Roulette.

7.2. MovieLens 25M

Table 8. Collaborative filtering results for the MovieLens 25M dataset.

	NDCG@10	NDCG@30	P@10	P@10	MAP	MRR
ConvQ	**0.79442**	**0.85507**	0.58550	**0.51083**	**0.43954**	**0.23508**
ConvM	0.79285	0.85353	**0.58700**	0.51017	0.43932	0.23011
Game	0.79292	0.85436	0.58400	0.51067	0.43782	0.23203

As shown above (Table 8), for the collaborative filtering, the Game case does not show better results in any metric, unlike ConvQ, which is the most efficient on all metrics except P@10. Therefore, the ConvQ case excels the other two cases.

Table 9. Content based filtering results for the MovieLens 25M dataset.

	NDCG@10	NDCG@30	P@10	P@10	MAP	MRR
ConvQ	0.75590	0.77239	0.36500	0.16383	0.42211	0.39969
ConvM	0.76139	**0.78382**	**0.37450**	**0.18650**	**0.44950**	0.40544
Game	**0.76612**	0.78302	0.35600	0.16033	0.42157	**0.41650**

Based on the results of Content Based Filtering as presented in Table 9, the ConvM case excels, since it shows better results in four out of six metrics while the Game case in only two.

Table 10. Hybrid2.1 filtering results for the MovieLens 25M dataset.

	NDCG@10	NDCG@30	P@10	P@10	MAP	MRR
ConvQ	0.73204	0.76125	0.34800	0.19433	0.41748	0.37929
ConvM	0.72591	0.75460	**0.36950**	**0.19950**	**0.42829**	0.39349
Game	**0.73524**	**0.76375**	0.33900	0.18933	0.40660	**0.39910**

Table 11. Hybrid2.2 filtering results for the MovieLens 25M dataset.

	NDCG@10	NDCG@30	P@10	P@10	MAP	MRR
ConvQ	0.74198	0.81873	0.54400	0.48967	0.40090	0.22376
ConvM	**0.74414**	0.81919	**0.54450**	0.49117	**0.40493**	0.21798
Game	0.74195	**0.81995**	0.54350	**0.49217**	0.40316	**0.22553**

As shown in Tables 10 and 11, where the results of Hybrid2.1 and Hybrid2.2 are presented respectively, the Game case yields better results in three out of six metrics and the same applies for the ConvM case. Therefore, we cannot consider that one of the two cases prevails.

Table 12. Hybrid3 results for the MovieLens 25M dataset.

	NDCG@10	NDCG@30	P@10	P@30	MAP	MRR
CF	0.79292	0.85436	**0.58400**	0.51067	0.43782	0.23203
CB	0.76612	0.78302	0.35600	0.16033	0.42157	**0.41650**
H3 (Game)	**0.79365**	**0.85527**	0.58350	**0.51100**	**0.43846**	0.23117

	NDCG@10	NDCG@30	P@10	P@30	MAP	MRR
H3 (Simple)	0.77583	0.84160	0.56850	0.50350	0.42228	0.23116
H3 (Game)	**0.79365**	**0.85527**	**0.58350**	**0.51100**	**0.43846**	**0.23117**

For the Hybrid3 algorithm, we compare the results as for the experiments in the MovieLens-100k dataset. As shown in Table 12, the Game case of the Hybrid3 outperforms the two individual algorithms combined, as it shows better results in all metrics except MRR and P@10. Also, the Game case is a more efficient than the simple linear combination, based on all metrics.

Table 13. Hybrid4 results for the MovieLens 25M dataset.

	NDCG@10	NDCG@30	P@10	P@30	MAP	MRR
CF	0.79292	0.85436	0.58400	**0.51067**	0.43782	0.23203
H2.2	0.74195	0.81995	0.54350	0.49217	0.40316	0.22553
H4 (Game)	**0.80882**	**0.86029**	**0.59350**	0.51000	**0.45030**	**0.23711**

	NDCG@10	NDCG@30	P@10	P@30	MAP	MRR
H4 (Roulette)	0.79782	0.84916	0.58850	0.50600	0.44753	**0.23860**
H4 (Game)	**0.80882**	**0.86029**	**0.59350**	**0.51000**	**0.45030**	0.23711

As shown in Table 13, the Hybrid4 algorithm outperforms the algorithms collaborative filtering and Hybrid2.2 filtering from which the initial population is derived, as it achieves better performance based on all metrics except P@30. In addition, the method Choice by Game leads to better overall results compared to Choice by Roulette, as it is a more efficient based on all metrics except MRR.

Summing up, to draw conclusions, we categorize the algorithms based on the way they were modelled as games, thus distinguishing three categories. The first category includes the algorithms in which the ”Game of Recommendation Process” is used, i.e. the algorithms collaborative filtering, demographic filtering, content based filtering and their combinations, Hybrid1 and Hybrid2. The second category concerns the Hybrid3 algorithm, which uses the ”Game of Recommendation Methods Combination” and finally, the third category, which includes the Hybrid4 genetic algorithm where the ”Game of Individual Selection” is used, where each parent is considered player of a cooperative game. For the first category, the initial experiments in the MovieLens-100k dataset lead to the conclusion that the recommendation process can be improved if modelled as a game, since in all algorithms except Hybrid1.2, the Game case provides better results than other cases. This conclusion is overturned by the experiments in the MovieLens-25M dataset since the Game case does not outperform in any algorithm. For the experiments in the MovieLens 25M dataset, the same parameters as for MovieLens 100k were used instead of being experimentally optimized, because the experiments were very time consuming. Presumably, the fact that the same parameters were used is responsible for the reduced performance of the Game case. For the second category, i.e. for the Hybrid3 algorithm, although different combinations where performed for MovieLens 100k and MovieLens 25M, in both cases the weighted combination using game outperformed the individual algorithms combined and the simple weighted combination as well. Finally, for the third category, the Hybrid4 algorithm, experiments in the MovieLens 100k dataset show that the use of the top-k users of the collaborative filtering and demographic filtering algorithms as the initial population of the genetic algorithm results in better recommendations than the two algorithms, if the Choice by Game method is used, while the same does not apply for the Choice by Roulette method. This conclusion is confirmed and reinforced by the experiments in the MovieLens-25M dataset since the results are the same although different algorithms were combined.

8. Conclusions & Future Work

As mentioned above, the purpose of the present study is to develop recommendation systems using game theoretic techniques to further inquire and confirm the claim that the recommendation process can be improved by utilizing game theory[36]. After a series of experiments on different datasets for recommendation algorithms developed using game theoretic techniques, we conclude that a recommendation system can become more efficient when game theory is integrated to it. The algorithms developed do not only concern the recommendation systems, since they can be applied to information retrieval systems as well as in machine learning in general. An important finding is the method Choice by Game as it brings about a significant performance improvement over the existing method Choice by Roulette.

For the future work, better integration of Nash equilibrium theory into algorithms may lead to improved results. In particular, the execution of the Hybrid3 and Hybrid4 algorithms is interrupted when an equilibrium state is found, but this state is probably a local maximum of the utility function and not a total one. So, algorithms need to be further developed to find the total maximum. Regarding the Hybrid3 algorithm, it would be interesting to adapt to more players, i.e. in a linear combination of more than two methods of recommendations [9]. Finally, it is important to further research and apply techniques of game theory in more methods and scenarios of recommendations and Machine Learning in general, since it is possible that new optimization techniques will emerge for this field.

References

1. Azadjalal, M.M., Moradi, P., Abdollahpouri, A.: Application of game theory techniques for improving trust based recommender systems in social networks. In: 2014 4th International Conference on Computer and Knowledge Engineering (ICCKE). pp. 261–266 (2014)
2. Azam, N., Yao, J.: Game-theoretic rough sets for recommender systems. Knowledge-Based Systems 72 (12 2014)
3. Bashiri, P.: Recommender systems: Survey and possible extensions (08 2018)
4. Ben-Porat, O., Tennenholtz, M.: A game-theoretic approach to recommendation systems with strategic content providers. arXiv (Nips) (2018)
5. Bobadilla, J., Ortega, F., Hernando, A., Gutiérrez, A.: Recommender systems survey. Knowledge-Based Systems 46, 109–132 (2013), https://www.sciencedirect.com/science/article/pii/S0950705113001044
6. Branzei, R., Dimitrov, D., Tijs, S., Ebooks Corporation.: Models in cooperative game theory : Crisp, fuzzy, and multi-choice games p. 137 (2005)
7. Burke, R.: Hybrid web recommender systems. Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics) 4321 LNCS, 377–408 (2007)
8. Çano, E., Morisio, M.: Hybrid recommender systems: A systematic literature review. Intelligent Data Analysis 21(6), 1487–1524 (2017)
9. Davis, M., Maschler, M.: The kernel of a cooperative game (1965)
10. Halkidi, M., Koutsopoulos, I.: A game theoretic framework for data privacy preservation in recommender systems. pp. 629–644 (09 2011)
11. Harman, D.: Relevance feedback revisited. In: Proceedings of the 15th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval. p. 1–10.

SIGIR '92, Association for Computing Machinery, New York, NY, USA (1992), https://doi.org/10.1145/133160.133167
12. Harper, F.M., Konstan, J.A.: The movielens datasets: History and context. ACM Transactions on Interactive Intelligent Systems 5(4) (2015)
13. Hernández del Olmo, F., Gaudioso, E.: Evaluation of recommender systems: A new approach. Expert Systems with Applications 35(3), 790–804 (2008)
14. Hwang, W.S., Li, S., Kim, S.W., Lee, K.: Data imputation using a trust network for recommendation via matrix factorization. Computer Science and Information Systems 15(2), 347–368 (2018)
15. Kim, T.K.: T test as a parametric statistic. Korean J Anesthesiol 68(6), 540–546 (2015), http://ekja.org/journal/view.php?number=8123
16. Lavrenko, V., Croft, W.B.: Relevance models in information retrieval. Language Modeling for Information Retrieval pp. 11–56 (2003)
17. Lü, L., Medo, M., Yeung, C.H., Zhang, Y.C., Zhang, Z.K., Zhou, T.: Recommender systems. Physics Reports 519(1), 1–49 (2012)
18. Myerson, R.B.: Game Theory: Analysis of Conflict. Harvard University Press (1991), http://www.jstor.org/stable/j.ctvjsf522
19. Nisan, N., Roughgarden, T., Tardos, É., Vazirani, V.V.: Algorithmic game theory, vol. 9780521872829. Cambridge University Press, Cambridge (2007)
20. Nussbaumer, A., Dahrendorf, D., Schmitz, H.C., Kravčík, M., Berthold, M., Albert, D.: Recommender and guidance strategies for creating personal mashup learning environments. Computer Science and Information Systems 11, 321–342 (1 2014)
21. Quadrana, M., Karatzoglou, A., Hidasi, B., Cremonesi, P.: Personalizing session-based recommendations with hierarchical recurrent neural networks. RecSys 2017 - Proceedings of the 11th ACM Conference on Recommender Systems pp. 130–137 (2017)
22. Rendle, S., Zhang, L., Koren, Y.: On the difficulty of evaluating baselines: A Study on Recommender Systems. arXiv pp. 1–19 (2019)
23. Resnick, P., Varian, H.R.: Recommender systems. Commun. ACM 40(3), 56–58 (Mar 1997), https://doi.org/10.1145/245108.245121
24. Robertson, S.E., Jones, K.S.: Relevance weighting of search terms. Journal of the American Society for Information Science 27(3), 129–146 (1976), https://asistdl.onlinelibrary.wiley.com/doi/abs/10.1002/asi.4630270302
25. Robertson, S., Zaragoza, H.: The probabilistic relevance framework: Bm25 and beyond. Foundations and Trends in Information Retrieval 3, 333–389 (01 2009)
26. Rocchio, J.J.: Relevance feedback in information retrieval. In: Salton, G. (ed.) The Smart retrieval system - experiments in automatic document processing, pp. 313–323. Englewood Cliffs, NJ: Prentice-Hall (1971)
27. Ross, P., Corne, D.: Applications of genetic algorithms. In: ON TRANSCOMPUTER BASED PARALLEL PROCESSING SYSTEMS," LECTURE. University of Edinburgh (1995)
28. Shani, G., Gunawardana, A.: Evaluating Recommendation Systems, vol. 12, pp. 257–297 (01 2011)
29. Ter Hofstede, A.H., Proper, H.A., Van Der Weide, T.H.: Query formulation as an information retrieval problem. Computer Journal 39(4) (1996)
30. Wang, J., de Vries, A.P., Reinders, M.J.T.: Unifying user-based and item-based collaborative filtering approaches by similarity fusion. In: Proceedings of the 29th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval. p. 501–508. SIGIR '06, Association for Computing Machinery, New York, NY, USA (2006), https://doi.org/10.1145/1148170.1148257
31. Xu, J., Croft, W.B.: Quary expansion using local and global document analysis. SIGIR Forum 51(2), 168–175 (Aug 2017), https://doi.org/10.1145/3130348.3130364
32. Xu, Z., Jiang, H., Kong, X., Kang, J., Wang, W., Xia, F.: Cross-domain item recommendation based on user similarity. Computer Science and Information Systems 13, 359–373 (6 2016)

33. Zhai, C.: Towards a game-theoretic framework for text data retrieval. IEEE Data Eng. Bull. 39, 51–62 (2016)
34. Zhai, C., Lafferty, J.: A study of smoothing methods for language models applied to information retrieval. ACM Trans. Inf. Syst. 22(2), 179–214 (Apr 2004), https://doi.org/10.1145/984321.984322
35. Zhang, S., Yao, L., Sun, A., Tay, Y.: Deep learning based recommender system: A survey and new perspectives. ACM Comput. Surv. 52(1) (Feb 2019), https://doi.org/10.1145/3285029
36. Zou, S., Tao, G., Wang, J., Zhang, W., Zhang, D.: On the equilibrium of query reformulation and document retrieval. ICTIR 2018 - Proceedings of the 2018 ACM SIGIR International Conference on the Theory of Information Retrieval pp. 43–50 (2018)

Evangelos Sofikitis is currently working as a Junior Software Engineer at Oxla, a highly efficient, analytical, distributed Database. As of 2021, he is a graduate of the Department of Computer Engineering and Informatics, University of Patras. His main field of interest during his studies was Machine Learning and Recommendation Systems.

Christos Makris is an Associate Professor in the University of Patras, Department of Computer Engineering and Informatics, from September 2017. Since 2004 he served as an Assistant Professor in CEID, UoP, tenured in that position from 2008. His research interests include Data Structures, Information Retrieval, Data Mining, String Processing Algorithms, Computational Geometry, Internet Technologies, Bioinformatics, and Multimedia Databases. He has published over 100 papers in refereed scientific journals and conferences and has more than 600 citations excluding self-citations (h-index: 18).

Received: September 30, 2021; Accepted: May 10, 2022.