FIFARecs: A Recommender System for FIFA18

Jaka Klancar[1], Karsten Paulussen[2], and Kostas Stefanidis[3]

[1] University of Ljubljana, Slovenia
jk7808@student.uni-lj.si
[2] Eindhoven University of Technology, The Netherlands
k.paulussen@student.tue.nl
[3] Tampere University, Finland
konstantinos.stefanidis@tuni.fi

Abstract. One of the most popular features in the FIFA18 game is the career mode, where the target of the users is to improve their teams and win as much competitions as possible. Usually, it is hard for the users to decide which players to select to buy to maximally improve their team by taking into account all different players' attributes. In this paper, we introduce an approach towards helping the users to both determine the worst player in a specific team and recommend a list of players seen as possible replacement to improve the team.

Keywords: Recommender Systems; Recommendations; Group Recommendations; FIFA18 Game

1 Introduction

FIFA18 is a football simulation video game developed by Electronic Arts Sports (shortly, EA Sports[4]). It is the newest game in a series of football video games going back to 1994, with more than six million sales around the globe. EA Sports was the first label that had an official license from FIFA, namely the International Federation of Association Football. The license gave them rights to use names and presentations of most players and competitions around the world.

In FIFA18, users can play with more than 700 teams from 30 competitions. The game contains different gameplays. One of the most popular is the career mode, on which we focused in this paper. In career mode, the users start with a self-chosen team. They can decide and change which formation they want their team to play with, and then choose players for the chosen formation. The users play matches with their team in different competitions. To improve the team, they can buy new players during the transfer periods, which are two times during the season. Improving a team by buying and selling players is an important part of the career. Users in the game can build their team through the career-mode gameplay, by exploiting a big number of attributes/characteristics regarding the players, which can increase or decrease over the years. Due to the large number

[4] https://www.ea.com

of players in the game and their different attributes that have be taken into account, it is typically difficult for the user to find the right player for the right position.

Users first have to decide which player is the worst in their team. One option is just to look at the overall score describing the player's performance, but this overall score can change based on the age and the potential of the players. Furthermore, it can also be that a player whose overall score is higher, but his transfer price and wage are also higher, can be rated as worse. After deciding which player is the worst the user has to find the right player to take his place. Again the age, overall and potential, together with some other attributes have to be taken into account.

It is hard for the users to decide which players to buy to maximally improve their team by taking into account all different attributes. Therefore, in this work, we provide an application that helps the users by both determining which is the worst player in their team and by recommending a list of players that could replace that player and improve the team. So far, only the Xbox game console has its own recommender system that proposes games based on the games played earlier [5]. The first application for recommender systems to digital games was proposed in 2014 [10]. Two different Top-L recommender systems were proposed based on archetypal analysis. However, in all existing approaches the concept of suggesting the video games option to improve user's experience and performance is missing. To the best of our knowledge, our approach is the first in the field that explorers how to provide in-game recommendations.

We have evaluated our approach along two perspectives: usability and performance. Our usability experiments consider both the easiness of the approach and the satisfaction of the users from the quality of the results. We used a real dataset, containing 17980 football players with more than 70 attributes, describing both personal and performance characteristics. Our performance experiments focus on evaluating the efficiency of the similarity measures used, namely Pearson correlation, cosine similarity and Minkowski similarity, versus the quality of the results thus achieved.

In a nutshell, in this paper:

- We introduce a recommenders-like approach for assisting users form their teams in FIFA18: We initially suggest the first eleven players, as well as the worst player in the team, and then recommend a list of players that can replace the latter.
- We propose an effective presentation solution for the recommended players which builds upon real needs. Our solution provides a ranked overview of the players enriched with summarized information, and can facilitate user browsing.
- We implement a proof of concept prototype for players recommendations and evaluate our approach in terms of usability and performance. Our experiments show that using FIFARecs can achieve fast and useful suggestions for the users.

The rest of this paper is structured as follows. In Section 2, we present related work, and in Section 3, we introduce our approach for producing recommendations in FIFARecs. In Section 4, we describe the dataset that is associated with FIFA18, while in Section 5, we focus on the effective presentation of FIFARecs. Section 6 presents our usability and performance evaluation results, and finally, Section 7 concludes the paper with a summary of our contributions.

2 Related Work

Recommender systems aim at providing suggestions to users or groups of users by estimating their item preferences and recommending those items featuring the maximal predicted preference. Typically, depending on the type of the input data, i.e., user behavior, contextual information, item/user similarity, recommendation approaches are classified as content-based [8], collaborative filtering [9], knowledge-based [2], hybrid [1], or even social ones [12]. Nowadays, recommendations have more broad applications, beyond products, like links (friends) recommendations [15], social-based recommendations [12], query recommendations [3], health-related recommendations [13, 14], open source software recommendations [6], diverse venue recommendations [4], recommendations for groups [7], or even recommendations for evolution measures [11].

Clearly, recommender systems can be also useful in the domain of video games. Due to the large number of game releases every year, gamers can have hard time finding games fitting their interests. Therefore, the Xbox game console has its own recommender system that proposes games based on the games played earlier [5]. The first application for recommender systems to digital games was proposed in 2014 [10]. Two different Top-L recommender systems were proposed based on archetypal analysis. However, in all existing academic literature of recommender systems the concept of suggesting the video games option to improve user's experience and performance is missing. To the best of our knowledge, our approach is the first in the field that explorers how to provide in-game recommendations.

However, in all existing recommender systems the concept of suggesting video games is completely missing. To the best of our knowledge, our approach is the only one in this field that explores how to provide games recommendations.

3 Recommendations in FIFA18

In this section, we introduce the main steps of our approach. First, we explain how the first eleven football players are chosen and how we determine which of these players is the worst. Then, a joint profile for the team players is created, and finally, we describe how the recommendations for the players to replace the worst one are made. Each player has more than 70 attributes divided into personal attributes (e.g., age, nationality and value), performance attributes (e.g., overall, potential and aggression), and overall score for each position. More details on the FIFA18 dataset appear in Section 4.

```
def sort_first_eleven(players):
    true_overall = overall if age > 23 else
        (overall+potential)/2
    players_diff = true_overall - avg_overall
    players_diff = players_diff * penalty
    players_diff = normalize(players_diff)
    wage_diff = normalize(wage)
    players_diff = 0.7*players_diff + 0.3*(-1 * players_wage)
    return sort(players_diff)
```

Fig. 1: The pseudo-code for sorting the players in a team.

3.1 Selection of the First Eleven and the Worst Player

Every team in FIFA 18 contains more than 25 players in its selection. This makes the process of selecting the formation and the players fitting the formation hard. FIFARecs introduces a method that helps users make valuable selection in an easy way. The user can choose between different formations, and the best eleven players for that formation will be provided automatically. For each position in the formation, the best player who has this position as a preferred position is taken. The best player for a position is based on the positional score, i.e., the attribute players have for that position. If none of the players in the team have the specific position as preferred, then the most similar position is taken. Similar positions are positions, which should a player also be able to play. Usually football players can play a variety of positions, but only some of them are their preferred positions. For example, a central defensive midfielder can also play as a center back. After the first eleven players were selected, the application will determine the worst player in the team. Figure 1 shows the algorithm used to rank and sort the eleven players. Based on a group of experts experience with the game, we decide that age, overall and potential score, as well as player's salary are important attributes to evaluate players.

At first, a new overall score (true overall score) is computed based on the age, overall, and potential score of the players. The first score is given to each player (player diff) based on the difference between his true overall score and the average score of the team. This score is multiplied by a penalty, between 0 and 1, and is based on the age of the player. Older players get higher penalty, because most of the users prefer younger players. The score is normalized and multiplied with the normalized wage. Here, more weight is given to the combination of overall score and age than to the salary of the players. The weights are decided purely by the experience of a group of experts with the game: overall, experts support that the player's ability is more important than his wage.

3.2 Joint Profile of the Team

After the selection of the worst player of the team, a joint profile for the team is formed. This joint profile will be used to evaluate if the quality of the players

that are recommended fit with the general quality of the team. For doing so, the average and standard deviations of the following attributes are computed: Age, Aggression, Ball control, Composure, Positioning, Reactions, Short passing, Sprint speed, Stamina, and Strength. These attributes are important for all field players, regardless of their positions. Average values were given weights – the more dense an attribute is, the higher weight it gets. The weights were given, because the more dense attributes appear to be more important for the selected team.

3.3 Collaborative Filtering Recommendations

The list of potential recommendations is first filtered based on self-determined constraints. First of all, a player that is recommended needs to have the same position as the player we want to replace. The score for this position has to be higher than the score of the worst player of this position, except when the player is younger than 23 years old. Then, the application take into account the potential overall score instead of the score for the specific positions. The potential score is used, because the player can still develop and become a better player in future. Furthermore, the recommended players cannot have a wage that is more than twice as high than that of the worst player – we assume that the users would not be able to afford that player for their team. In addition to the self-determined constraints, the user of the application can also choose some constraints, including age, wage, transfer price, overall.

After the list of players is filtered, a collaborative filtering algorithm is implemented in order to identify the recommendations. For doing so, we actually compute the similarity of each player in the filtered list with the joint profile we created for the first eleven team players. Such joint profile represents an artificial player. Thus, the top-k most similar players to the artificial one, represent the candidates for replacing the worst player.

Similar players are located via a *similarity function* $simP(p, p')$ that evaluates the proximity between two players p and p', by considering their shared dimensions. In our approach, we use three different similarity measures, namely: Pearson correlation, cosine similarity, and Minkowski distance. Pearson is widely used for computing similarities between users, players in our case. It actually measures the linear dependence between two players p and p': it returns a value between $+1$ and -1, where $+1$ is total positive linear correlation, 0 is no linear correlation and -1 is total negative linear correlation.

$$Pearson(p, p') = \frac{\sum_{i \in I}(r(p,i) - \mu_p)(r(p',i) - \mu_{p'})}{\sqrt{\sum_{i \in I}(r(p,i) - \mu_p)^2}\sqrt{\sum_{i \in I}(r(p',i) - \mu_{p'})^2}} \quad (1)$$

where $r(p, i)$ (resp., $r(p', i)$) is the score that the player p (resp., p') has for attribute i, μ_p (resp., $\mu_{p'}$) is the mean of p's (p''s) scores, and I is the set of attributes for which both p and p' have scores.

In a similar way, the cosine similarity of two players can be derived by using the Euclidean dot product formula, as follows:

$$cosine(p, p') = \frac{\sum_{i \in I} r(p,i) r(p',i)}{\sqrt{\sum_{i \in I} r(p,i)^2} \sqrt{\sum_{i \in I} r(p',i)^2}} \tag{2}$$

The resulting similarity ranges from -1 meaning exactly opposite, to $+1$ meaning exactly the same, with 0 indicating orthogonality, and in-between values indicating intermediate similarity or dissimilarity.

Alternatively, via Minkowski, we can measure the numerical difference for the corresponding attributes of p and p':

$$Minkowski(p, p') = \sqrt[n]{\sum_{i \in I} |r(p,i) - r(p',i)|^n} \tag{3}$$

Intuitively, Minkowski measures the numerical difference for each corresponding attributes of player p and player p'. Then it combines the square of differences in each attribute into an overall distance.

4 Data in FIFA18

The dataset for the FIFA18 game is available on Kaggle[5]. It contains 17980 cases, where each case relates to one football player. Each football player has more than 70 attributes. These attributes can be divided into personal attributes (e.g., age, nationality and value), performance attributes (e.g., overall, potential and aggression), and overall score for each position. Those three groups, with attributes and their data types, can be seen in Table 1. Note that Table 1 shows only the attributes that are used in our framework. The rest (attributes like club logo, flag and photo) are dropped.

Table 1: Player attributes in FIFA18 dataset.

Personal attribute	**Performance attributes**	**Position scores**
Name (object)	Acceleration (int64)	CAM (oat64)
Age (int64)	Aggression (int64)	CB (oat64)
Nationality (object)	Agility (int64)	CDM (oat64)
Overall (int64)	Vision (int64)	ST (oat64)
Potential (int64)	Volleys (int64)	GK (oat64)
Club (object)		
Value (oat64)		
Wage (oat64)		

Most of the players have multiple preferred positions. Those positions are presented as a string and positions are separated by space. We transform this

[5] https://www.kaggle.com/thec03u5/fifa-18-demo-player-dataset

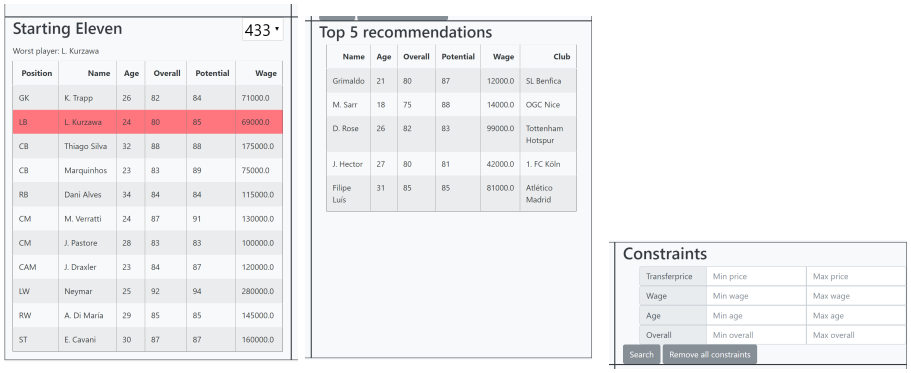

Fig. 2: The first eleven and the worst player (left), the top 5 recommendations for replacing the worst player (middle), and constraints specification (right).

string to a list of positions for every player. That makes it easier to use during development. The Wage and Value attributes are transformed to numeric values. Both attributes are structured the same (e.g., €123M and €280K). To transform them to numeric values, we needed to get rid of the € symbol and transform the last letter to corresponding number of zeros. Every player has score for every position, expect position of goalkeeper (GK). We created a new attribute GK, which has a score for goalkeeper position. To get that score, we simply took the overall of the player, and check if the player has GK as a preferred position, otherwise his GK score is 0. Some players were in the dataset twice, with different IDs. Those duplicated cases where dropped. Finally, we transformed all performance attributes to numeric values.

5 Presentation of Recommendations

To facilitate the players selection towards the formation of a team, we propose to organize the user interactions in a compact yet intuitive and representative way. To this end, to demonstrate the feasibility of our approach, we have developed a research prototype for FIFARecs that employs particular constraints on specific attributes related to the football players. Let's assume that the user in question is a fan of Paris Saint-Germain. After choosing his favorite team, using a dropdown element, the application will automatically create the first eleven players of the team by following the default formation, namely 4-3-3. The user can easily change the team's formation by using the drop-down select element in the top right field of the section "Starting Eleven" (Figure 2, left). The first eleven players are depicted also in Figure 2 (left). The worst player of the top eleven players is marked in red. In this example, this player, L. Kurzawa, has the lowest overall score, but note that the worst player is not only determined by the overall or potential score. The top five recommendations for replacing the worst player are given in Figure 2 (middle). The recommendation list shows the name of

the player, his age, his overall score, potential score, his wage, and his current club in the game. By default, the list of players is only filtered based on the systems self-determined constraints. The users can specify their own constraints in Figure 2 (right). In particular, the users first have to fill constraints they prefer before clicking on the search button. The constraints can be easily cleared by the remove button.

6 Experimental Evaluation

In this section, we evaluate usability and performance of our approach. Our usability experiments consider the easiness of the approach and the satisfaction of the users from the quality of recommendations. Our performance experiments focus on evaluating the efficiency of the similarity measures used, namely Pearson correlation, cosine similarity and Minkowski similarity.

For both cases, in our experiments, we used a real dataset (Section 3), containing 17980 football players, each one related to more than 70 attributes, describing personal and performance characteristics[6].

6.1 Usability Evaluation

The goal of our usability study is to justify the use of FIFARecs. The objective here is to show that users get interesting and useful results in an automatic way, without spending time for manually selecting the football players. We conducted an empirical evaluation of our approach with 20 users. For all users, it was the first time that they used the system. We evaluated our approach along two lines: overhead of understanding and using FIFARecs, and quality of results.

Easiness of Approach. For counting the easiness of our approach, we consider cases in which FIFARecs is either used or not. For both cases, we counted how long (in mins) it took users to specify their teams. Specifically, when FIFARecs was not involved in the process, we counted the total time needed by the users to manually select their teams. We repeated the same with FIFARecs.

For the FIFARecs case, since for all users this was their first experience with the system, the reported time includes the time it took the user to get accustomed with the system. These results are reported in Table 2; the last line of the table summarizes the resulting time for all users. Our general impression is that FIFARecs save much time in selecting football players. Using our system makes it easier for someone to understand several ideas behind the game, since the whole process acts as example. With regards to time, there was deviation among the time users spent on selecting teams: some users were more meticulous than others, spending more time in adjusting their players.

[6] Data and codes can be found here: https://github.com/klancar16/FIFA18_recommender

Table 2: Easiness of approach: Time (in minutes) for selecting teams manually and with FIFARecs.

	Selection time - Manually	**Selection time - FIFARecs**
User 1	7	2
User 2	7	3
User 3	8	2
User 4	8	3
User 5	12	4
User 6	13	4
User 7	5	3
User 8	5	2
User 9	5	2
User 10	7	2
User 11	9	5
User 12	8	4
User 13	8	3
User 14	9	3
User 15	11	3
User 16	14	3
User 17	9	2
User 18	13	4
User 19	11	2
User 20	11	3
Average time	9	2mins & 57secs

Quality of Results. In this set of experiments, we asked the users to evaluate the quality of the FIFARecs results. First, users were asked to evaluate the quality of the 11 players appearing in the result. For characterizing the quality of the players, users marked each of them with 1 or 0, indicating whether they considered that the player should belong to the best 11 ones or not. The number of 1s corresponds to the precision of the top-11 players, or $p(11)$. Users were also asked to give a numerical interest score between 1 and 10 to each of the 11 players. This score was in the range [1, 5], if the previous relevance indicator was 0 and in the range [6, 10], otherwise. We reported the number of players that were rated highly (interest score ≥ 7), namely the Highly Preferred Players (HPP). Finally, users were asked to provide an Overall Score (OS) in the range [1, 10] to indicate their overall satisfaction.

Next, we asked users to evaluate the result presenting the worst player. Users marked the worst player (WP) with 1 or 0, indicating whether they considered that the reported player should be the worst or not. Also, users were asked to evaluate the suggestions for replacing the worst player. Users were presented with three alternatives, each containing 5 player names, computed using three different measures, namely: Pearson (P), cosine (C), and Minkowski (M), and selected the one that best matches their needs.

Table 3: Quality of results.

	p(11)	**HPP**	**OS**	**WP**	**P**	**C**	**M**
User 1	82%	9	9	1			✓
User 2	91%	10	10	1	✓		
User 3	91%	10	10	1	✓		
User 4	91%	9	9	0		✓	
User 5	64%	7	8	1		✓	
User 6	82%	9	8	1			✓
User 7	82%	8	8	0		✓	
User 8	91%	10	10	1	✓		
User 9	82%	9	8	1			✓
User 10	91%	10	9	1	✓		
User 11	82%	9	9	0	✓		
User 12	100%	10	10	1	✓		
User 13	91%	10	8	0		✓	
User 14	91%	9	9	1	✓		
User 15	100%	10	9	1			✓
User 16	91%	10	9	1	✓		
User 17	82%	9	8	1	✓		
User 18	64%	7	7	1	✓		
User 19	91%	10	8	1			✓
User 20	91%	10	9	1			✓
Avg	86,5%	9,25	8,75	80%	50%	20%	30%

Table 3 depicts the detailed per user scores. Our results show that using FIFARecs, we can achieve recommendations of high quality. In almost all cases, FIFARecs is able to detect the worst players, while the overall satisfaction when considering the reported suggestions is high. With regards to the employed similarity functions, users seem to prefer the results computed via the Pearson correlation. Minkowski was selected by the 30% of the users, while only the 20% of the users selected cosine. Overall, given that users many times were not aware of the whole set of available players, since they were just presented with the top 11 players, they left room in their choices for better results that could be lying in the dataset that was not presented to them. For this reason, precision can be higher than the computed one.

6.2 Performance Evaluation

In this set of experiments, we evaluate the performance of our approach in terms of the execution time. This depends on whether we use Pearson (P), cosine (C) or Minkowski (M) for locating the candidate players for replacing the worst player. Thus, we study these three cases separately. We looped through all the teams in our database, and produced the top-5 suggestions, i.e., players' names, for replacing the worst player, for all three measures. The results appear in Table 4. Given that the reported time refers to not a single prediction but to

Table 4: Efficiency of the FIFARecs approach.

	P	**C**	**M**
Time (in secs)	236.40	6.15	64.44

predictions for all teams, it is easy for someone to claim that for single users, and thus recommendations for single teams, the system's response time is very good for the gaming conditions. Cosine reports the best time, while Pearson needs the highest amount of time. However, according to our usability evaluation, the detailed computations of Pearson offer the best recommendations.

7 Conclusions

FIFARecs introduces a method useful for the FIFA18 users. The goal of our recommender is to automatically pick the best first eleven players, so as to form a team. Furthermore, the recommender targets at helping users to decide which player of their team they have to replace and by whom, based on the formation and the constraints selected by the user.

Our future work focuses on how to enhance FIFARecs with online discussions between the users of FIFA18. Specifically, we focus on how users express their opinions, how opinions related to each other and how discussions around a game are formed. This process involves techniques such as text mining, topic modeling, and sentiment analysis. Moreover, we target at supporting services for retrieving opinions and providing overviews of opinions sets in a way that avoids information overload, so as to help users understand strategies of other users, as well as discover useful information about players and teams. User information needs will be expressed through keywords, and results will incorporate ranking and diversification, according to user preferences and contextual relevance. Finally, to be able to explore connections between opinions, we study navigation means and appropriate representation models.

Acknowledgement

This work has been partially supported by the Virpa D project funded by Business Finland.

References

1. Balabanovic, M., Shoham, Y.: Content-based, collaborative recommendation. Commun. ACM **40**(3), 66–72 (1997)
2. Bridge, D.G., Göker, M.H., McGinty, L., Smyth, B.: Case-based recommender systems. Knowledge Eng. Review **20**(3), 315–320 (2005)
3. Eirinaki, M., Abraham, S., Polyzotis, N., Shaikh, N.: Querie: Collaborative database exploration. IEEE Trans. Knowl. Data Eng. **26**(7), 1778–1790 (2014)

4. Ge, X., Chrysanthis, P.K., Pelechrinis, K.: MPG: not so random exploration of a city. In: MDM (2016)
5. Koenigstein, N., Nice, N., Paquet, U., Schleyen, N.: The xbox recommender system. In: Proceedings of the Sixth ACM Conference on Recommender Systems. pp. 281–284. RecSys '12 (2012)
6. Koskela, M., Simola, I., Stefanidis, K.: Open source software recommendations using github. In: TPDL (2018)
7. Ntoutsi, E., Stefanidis, K., Nørvåg, K., Kriegel, H.: Fast group recommendations by applying user clustering. In: ER (2012)
8. Pazzani, M.J., Billsus, D.: Content-based recommendation systems. In: The Adaptive Web, Methods and Strategies of Web Personalization (2007)
9. Sandvig, J.J., Mobasher, B., Burke, R.D.: A survey of collaborative recommendation and the robustness of model-based algorithms. IEEE Data Eng. Bull. **31**(2), 3–13 (2008)
10. Sifa, R., Bauckhage, C., Drachen, A.: Archetypal game recommender systems **1226**, 45–56 (01 2014)
11. Stefanidis, K., Kondylakis, H., Troullinou, G.: On recommending evolution measures: A human-aware approach. In: 33rd IEEE International Conference on Data Engineering, ICDE 2017, San Diego, CA, USA, April 19-22, 2017. pp. 1579–1581 (2017)
12. Stefanidis, K., Ntoutsi, E., Kondylakis, H., Velegrakis, Y.: Social-Based Collaborative Filtering, pp. 1–9. Springer New York, New York, NY (2017)
13. Stratigi, M., Kondylakis, H., Stefanidis, K.: Fairness in group recommendations in the health domain. In: ICDE (2017)
14. Stratigi, M., Kondylakis, H., Stefanidis, K.: Fairgrecs: Fair group recommendations by exploiting personal health information. In: DEXA (2018)
15. Yin, Z., Gupta, M., Weninger, T., Han, J.: LINKREC: a unified framework for link recommendation with user attributes and graph structure. In: WWW (2010)