Data Mining for Item Recommendation in MOBA Games

Vladimir Araujo*
Pontificia Universidad Católica de Chile
Santiago, Chile
vgaraujo@uc.cl

Felipe Rios
Pontificia Universidad Católica de Chile
Santiago, Chile
farios@uc.cl

Denis Parra*
Pontificia Universidad Católica de Chile
Santiago, Chile
dparra@ing.puc.cl

ABSTRACT

E-Sports has been positioned as an important activity within MOBA (Multiplayer Online Battle Arena) games in recent years. There is existing research on recommender systems in this topic, but most of it focuses on the character recommendation problem. However, the recommendation of items is also challenging because of its contextual nature, depending on the other characters. We have developed a framework that suggests items for a character based on the match context. The system aims to help players who have recently started the game as well as frequent players to take strategic advantage during a match and to improve their purchasing decision making. By analyzing a dataset of ranked matches through data mining techniques, we can capture purchase dynamic of experienced players to use it to generate recommendations. The results show that our proposed solution yields up to 80% of mAP, suggesting that the method leverages context information successfully. These results, together with open issues we mention in the paper, call for further research in the area.

CCS CONCEPTS

• **Information systems → Recommender systems**; *Data mining*.

KEYWORDS

Item Recommendation, Data Mining, MOBA Games

ACM Reference Format:
Vladimir Araujo, Felipe Rios, and Denis Parra. 2019. Data Mining for Item Recommendation in MOBA Games. In *Thirteenth ACM Conference on Recommender Systems (RecSys '19), September 16–20, 2019, Copenhagen, Denmark.* ACM, New York, NY, USA, 5 pages. https://doi.org/10.1145/3298689.3346986

1 INTRODUCTION

Competitive video gaming has gained popularity, thanks to e-Sports. An analysis by Seo and Jung [18] presented several reasons why people watch e-Sports according to their motivation: from pure entertainment to gaining a better understanding of the game. Recent studies [20] show that an estimated 335 million spectators watched e-Sports in 2018, generating a revenue of more than $900M. Although encouraging, the economic impact is not the only reason that makes this phenomena worth of research. Recently, the Olympic Council of Asia decided to include a medal for the e-Sports in their Hangzhou Games in 2022, as the British newspaper *The Guardian*, reported in mid-April of 2017. Certainly, MOBA games have contributed to this initiative.

Several Game AI studies related to MOBA games have been carried out, for instance, intelligent agents [8], prediction of match outcome [13], and also recommender systems [7]. MOBA presents two interesting recommendation problems to researchers: character recommendation, which has been studied in recent years [6, 12] with different approaches, and item recommendation [15], which is still an open issue. This work focuses on the second one.

Recommender systems have been very successful since the Netflix Prize [2]. Classical approaches to multimedia recommendation focus on past user behavior, items similar to those already consumed or similar users to suggest items for only one user [17]. Unlike canonical recommendation problems, in-game recommendation presents further challenges. First, each MOBA match is unique, giving a different combination of characters depending on their role and abilities. Second, when choosing an adequate team of characters, players must think of the most suitable set of items for that game considering their allies and enemies. Therefore, context is essential for a relevant recommendation for this problem.

Players use their expertise to acknowledge what item is convenient for a particular situation. Newcomer players need to do some research on the game dynamics and purchase strategy to understand what they often find with most popular recommendations by the community. Those recommendations may not be relevant to the context that the player faces. An adequate recommendation can help improve not only performance on a particular game but also smooth the learning curve. On the other hand, professional players are more experienced, but they could take advantage of recommendations in one-off matches. In other words, a well-trained recommender system can help them to find out combinations of relevant items for particular cases and contexts.

The remainder of this paper is organized as follows: Section 2 shows a brief overview of League of Legends and the recommendation problem; Section 3 shows the review of the related work; Section 4 explains the methodology; Section 5 presents the results and discussion; and Section 6 synthesizes the main conclusions and future work.

2 LEAGUE OF LEGENDS: OVERVIEW

League of Legends (LoL) is a popular game belonging to the MOBA genre. The game consists of two teams (red and blue) of five players each, that compete to be the first to destroy the enemy base. Each

*Also with Millennium Institute Foundational Research on Data, IMFD.

RecSys '19, September 16–20, 2019, Copenhagen, Denmark

ACM ISBN 978-1-4503-6243-6/19/09...$15.00
https://doi.org/10.1145/3298689.3346986

Figure 1: For this case with the current *champion* selection for both teams (red and blue), one *champion* is selected for item recommendation. We concentrated on the enemy team as context to reduce the search space.

player controls one character (*champion*) that interacts with the rest through combats, which are carried out in a particular arena.

Each *champion* has an exclusive set of abilities used to fight and that evolve during the progress of the game. Players receive gold for capturing positions and killing their opponents or little bots called *minions*. The resources acquired allow *champions* to extend their abilities by purchasing items. Appropriately managing the resources could generate an advantage over the rival team [9].

This game presents at least two recommendation problems: *champion* and item recommendation, because there are more than 140 characters and around 240 items (taking into account basic, advanced, and finished items). The second one is difficult because the items combinations can be of significant magnitude if we consider the *champion* role, ally or enemy team composition. For the scope of this paper, we focus on the finished item recommendation.

Figure 1 shows a typical example of *matchup*[1], in which a player has to choose a combination of six finished items according to the strategy, and it has to be done mainly considering the enemy abilities. To address this problem, we focus on the recommendation of a set of elements given a *champion* as a user and an enemy team as the context for the recommendation.

3 RELATED WORK

There are several works related to recommender systems in video games, especially in massively multiplayer online games, because of the large amount of data that can be gathered and the significant impact that this market has.

The consumption of content has been a main motivation for the study of recommendation systems, but maintaining a similar approach to the general market [7, 11, 21]. On the other hand, in-game recommendations have also received interest, where most works focused on collaborative filtering for the recommendation of items for role-playing games [3, 14, 19].

The MOBA Games genre has been studied mostly from the *champion* prediction paradigm. Recent work [12] exploited the use of association rules and ANN (artificial neural networks) to predict the winning team given a character selection. The system was able to recommend a character based on the choice phase. The ANN then predicts if the actual selection made by the players will be the winner.

Two of the most known MOBA games, LoL and DotA2 have draft pick[2] mode. The work by Chen et al. [6] predicts the most suitable *champion* selection in that mode, taking turns with an open selection (where each participant is aware of the selection of her mates and opponents). Given the current partial state of the *champion* selection, a Monte Carlo Tree Search will predict the most efficient *champion* selection.

Item recommendation is a paradigm that few researchers have explored. Some studies [15] have taken into consideration a sequential solution to solve this problem. Using association rules, the method predicts the next purchase of an item given a state of purchases of a specific *champion*. Some role or style associated with the *champion* has been taken into consideration.

Item recommendation is a paradigm that few researchers have explored. Some studies [15] have taken into consideration a sequential solution to solve this problem. The method predicts the next purchase of an item given a state of purchases of a specific *champion* using association rules. Some role or style associated with the *champion* has been taken into consideration.

4 METHODOLOGY

We have tested several techniques for item recommendation for LoL. Figure 2 shows the proposed framework, which is based on classical data mining methodology [1], but including slight changes to use it for the recommendation task. This methodology allows generating recommendations using knowledge learned from the actions carried out by the users in previous matches.

The first step is preprocessing the dataset, which consists of filtering, reordering, and encoding. We then apply several techniques of data mining, both descriptive (see Section 4.2) and predictive (see Section 4.3). From each method, we produce a ranking interpretation to perform the recommendation.

4.1 Dataset and Preprocessing

The dataset used in this study is provided by [4], consisting of 184,070 ranked matches from the regions of Europe and North America. It was selected because we hypothesized that information was collected from high-level players who know how to choose the items based on the current context. The raw dataset includes several files that present information about a match, such as the *champions* used, the items purchased, and the resulting statistics. It does not include information about the users' account.

Because the purpose is a recommendation for a specific champion, we decided to reorganize the data, including the most relevant information for this work. The final dataset has transactional structure[3], where each instance includes the identifiers of a selected champion during a match, the items used, and the enemies involved. We filtered only the winning matches of the season 7 because each season involves big changes in the metagame that affect the purchase strategy [10].

[1]The team composition of the opponents to face off during a match.

[2]Mode where players ban and select *champions* in competitive matchs.

[3]Typically used in data mining, each record captures a transaction, such as a customer's purchase, a flight booking, etc.

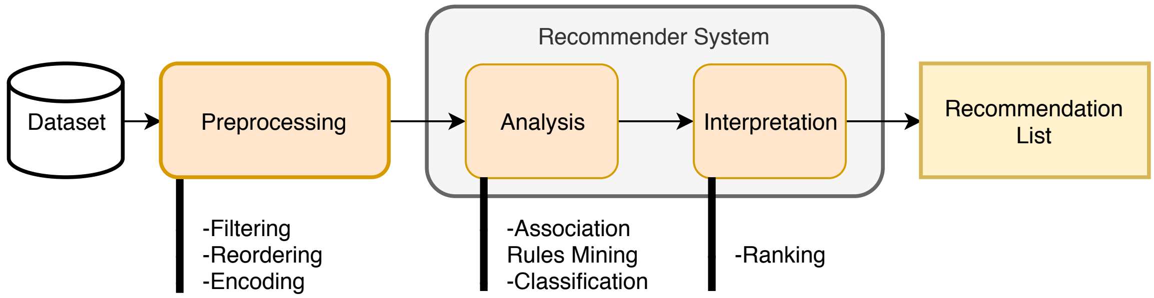

Figure 2: Recommendation framework based on data mining process.

On the one hand, all new *champions* released in the game after season 7 were removed, leaving 136. On the other hand, only finished items were used, which means that basic, advanced, and consumable items were eliminated, resulting in 93 items. This final file was divided into $631,590$ instances for training and $70,405$ instances for testing.

4.2 Recommender System Based on Association Rules Mining

The association rules have already been used for the recommendation of items for DotA 2 [15], but our approach is slightly different. We filtered the dataset for each champion of the game for rules mining (including both the enemies confronted and the items used as a transaction). We did this because item purchasing strategies tend to be champion-specific [5] and also so that the item recommendation will be context-aware.

This system consists of an association rules algorithm module that extracts and stores the frequent itemset of each champion. We used the Apriori and Eclat algorithms for this task. Also, a search engine module is responsible for generating a recommendation through the search in the files stored by the previous module.

Through $X, Y = \phi_{AR}(D \mid supp_{min}, conf_{min})$, the algorithm generates rules with the form $X \rightarrow Y$, where Y is an enemy *champion* and X is a subset of items. D is the transactional data of a picked *champion* composed of the enemy team and the items used.

On the one hand, *support* is defined as the perceptual frequency of observation of the same item in all the transactions. On the other hand, *confidence* is defined as the probability $P(Y \mid X)$. For both it is important to set a minimum value $supp_{min}$ and $conf_{min}$. Both Apriori and Eclat were set up with minimum support of 4 and minimum confidence of 80%.

In this way, it is possible to find a frequent itemset $I_f = X$. However, because it is quite likely that I_f could contain a subset of frequent itemset J_f, we use the maximal itemset that is defined as $I_m = I_f \mid I_f \wedge \nexists J_f \supset I_f$. The itemset I_m retrieved with the highest confidence were used to generate the recommendation.

4.3 Recommender System Based on Classifiers

Classifiers have been used frequently for mining tasks [16]. For this work, we applied a supervised approach for multi-label classification to obtain a mapping between features space (*champions*) and label space (items). On the one hand, the input, which represents a match played, was coded with one-hot, where the selected *champion* was represented with 1, the enemies with -1, and the rest with 0. On the other hand, the output presents the items used with 1 and 0 for the others. Though we evaluated three types of classifiers, the task can be carried out with other models.

4.3.1 Decision Trees. This technique generates a classifier in the form of a tree structure based on the features of the inputs. In this work, each decision node evaluates the *champion* selected, and the enemies presented, while leaf nodes indicate the item used in that context. We have implemented a decision tree with the Gini impurity function to measure the quality of a split because it is good at dealing with classification errors and is not computationally exhausting. Also, we do not use any maximum depth value for the tree to capture special cases of the context.

4.3.2 Logistic Regression. Logistic regression is a linear model used to predict the probability that an input belongs to one class or another. Typically, this model is trained with gradient descent, but due to the amount of this dataset, we decided to use stochastic gradient [22]. Besides, to use this model as a multi-label classifier, we adopted the one-vs-all strategy, which consists of fitting one classifier per class. The class is then fitted against all the other classes for each classifier.

4.3.3 Artificial Neural Network. Multi-Layered Perceptron (MLP) is frequently used to approximate nonlinear relationships existing between an input and the corresponding output. We used a fully connected architecture with 2 hidden layers of 150 neurons each, the input layer corresponds to the 136 *champions* and the output layer corresponding to the 93 items. Because this is a multi-label classification, we used binary cross-entropy loss function and sigmoid activation function to evaluate each class independently.

4.4 Interpretation by ranking

In a general view, in order to generate the itemset recommendation for the user, we use a sort function over the output of the recommender systems based on its probabilities $RecList = sort(Output_{RecSys}, P)$.

On the one hand, association rules methods generate several frequent itemsets with a confidence value associated. We organized these itemsets from highest to lowest, and then we take the first one. If the highest confidence itemset does not have a high enough amount for generating the recommendation required, the next itemset is merged.

Table 1: Results for @N-itemset

Method	Precision	Recall	F1-Score	MAP	MRR
Apriori@1	0.23	0.11	0.18	0.43	0.43
Eclat@1	0.44	0.12	0.18	0.44	0.44
D Tree@1	0.65	0.17	0.27	0.64	0.64
Logit@1	0.67	0.18	0.28	0.67	0.67
ANN@1	**0.71**	**0.19**	**0.30**	**0.71**	**0.71**
Apriori@3	0.42	0.33	0.36	0.59	0.60
Eclat@3	0.42	0.33	0.36	0.60	0.61
D Tree@3	0.47	0.37	0.41	0.71	0.73
Logit@3	0.53	0.42	0.46	0.74	0.76
ANN@3	**0.60**	**0.48**	**0.52**	**0.78**	**0.79**
Apriori@6	0.40	0.62	0.48	0.58	0.63
Eclat@6	0.41	0.62	0.48	0.59	0.64
D Tree@6	0.32	0.50	0.38	0.69	0.75
Logit@6	0.37	0.59	0.43	0.71	0.78
ANN@6	**0.44**	**0.69**	**0.53**	**0.74**	**0.81**

On the other hand, because of the implemented classifiers have a multi-label output, we apply sort function over the list of probabilities. This function returns the most probable item for the context at the beginning, and so on.

5 RESULTS AND DISCUSSION

In order to measure the quality of our systems, we adopt an evaluation per itemset size using metrics like precision, recall, and F1-score.

It must be taken into account that it is possible that a *champion* has not completed its items-build during a match. Because of that, the first evaluation was done by generating a fixed itemset size recommendation for each instance in the test set.

Table 1 shows the results, and we have also included the mean average precision (MAP), which evaluates relevant items from 1 to k, and the mean reciprocal rank (MRR), which evaluates the rank of the first correct item. Typical behavior occurs in all models: precision decays and recall rises while the list of recommendations increases. Therefore, F1 grows, transmitting the balance between precision and recall.

As we expected ANN model outperforms the other approaches [1], achieving F1 of 53%, and MAP of 74%, while the worst was Apriori approach reaching F1 of 48%, and MAP of 58%, showing a difference of $\sim$ 15% between them. Also, ANN obtained 71%, 79%, and 81% of MRR, which might indicate that this model recommends the first best situational-item.

For the second evaluation, we take into account the dimension of the label vector. We divide the dataset of testing into six groups, according to the size of the label vector, and each group is evaluated with itemset recommendations of the corresponding size. Figure 3 shows the results. In this evaluation, we also add Random and *champion*-based Most Popular recommendation because there is not a previous baseline. It is possible to see that the ANN recommender has the highest performance once again, reaching 80% of MAP@6.

In contrast, as we expected, the Random recommendation achieved lower performance with almost 15% of MAP@6. Remarkably, Most

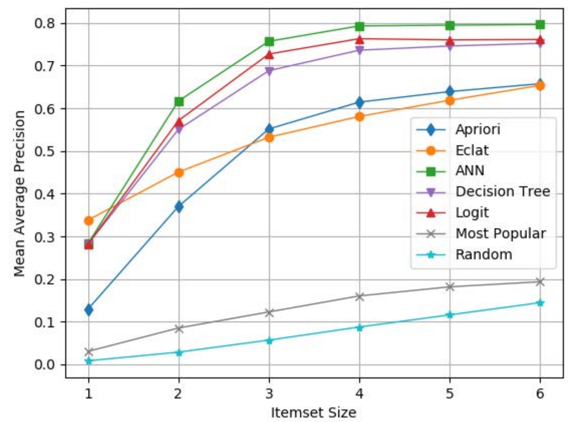

Figure 3: Evaluation with MAP by itemset size for the second evaluation.

Popular recommendation achieved only 20% in spite of being a *champion*-based version. This result may suggest that items used during a match are dependent on the current situation.

As seen in the previous evaluation, ANN, Logit, and Decision trees outperform Apriori and Eclat algorithms in almost all itemset sizes. There was an unexpected case with the Eclat algorithm, which could predict the first best situational-item (best performance of MAP@1). However, we assume that it was a coincidence because it then shows the lowest performance than the rest.

These results are promising. However, we only use the finished items. The inclusion of the remaining items could reduce the performance of our models, so more sophisticated systems that take into account more information from the dataset may be necessary.

6 CONCLUSIONS AND FUTURE WORK

In this paper, we presented a simple framework for the recommendation of items for LoL. It is based on a data mining methodology using five different methods. The evaluation showed promising results with the best model based on ANN, which seems to be able to make recommendations according to the context (enemy team).

This work shows that the recommendation systems in the MOBA games have relevant challenges to face, encouraging to explore this field to improve these types of systems.

Qualitative evaluation by an expert would be desirable to analyze the list of recommendations in an experimental setting. It might ensure that the suggested items are reliable and that they include context information. We will also continue with the context approach, but use additional information such as the allied team, characteristics of the items, and the role of the characters.

It would be desirable to have a system that can run during a match. Future work will focus on assert which technique could run in real-time without losing accuracy.

ACKNOWLEDGMENTS

This work has been partially funded by Millennium Institute for Foundational Research on Data (IMFD).

REFERENCES

[1] Xavier Amatriain and Josep M. Pujol. 2015. Data Mining Methods for Recommender Systems. In *Recommender Systems Handbook*. Springer US, 227–262. https://doi.org/10.1007/978-1-4899-7637-6_7

[2] James Bennett, Stan Lanning, et al. 2007. The netflix prize. In *Proceedings of KDD cup and workshop*, Vol. 2007. New York, NY, USA, 35.

[3] Paul Bertens, Anna Guitart, Pei Pei Chen, and Africa Perianez. 2018. A Machine-Learning Item Recommendation System for Video Games. In *2018 IEEE Conference on Computational Intelligence and Games (CIG)*. IEEE. https://doi.org/10.1109/cig.2018.8490456

[4] Paolo Campanelli. 2017. League of Legends Ranked Matches. data retrieved from Kaggle, https://www.kaggle.com/paololol/league-of-legends-ranked-matches.

[5] O. Cavadenti, V. Codocedo, J. Boulicaut, and M. Kaytoue. 2016. What Did I Do Wrong in My MOBA Game? Mining Patterns Discriminating Deviant Behaviours. In *2016 IEEE International Conference on Data Science and Advanced Analytics (DSAA)*. 662–671. https://doi.org/10.1109/DSAA.2016.75

[6] Zhengxing Chen, Truong-Huy D Nguyen, Yuyu Xu, Christopher Amato, Seth Cooper, Yizhou Sun, and Magy Seif El-Nasr. 2018. The art of drafting: A Team-Oriented Hero Recommendation System for Multiplayer Online Battle Arena Games. In *Proceedings of the 12th ACM Conference on Recommender Systems - RecSys '18*. ACM Press. https://doi.org/10.1145/3240323.3240345

[7] Germán Cheuque, José Guzmán, and Denis Parra. 2019. Recommender Systems for Online Video Game Platforms: The Case of STEAM. In *Companion Proceedings of The 2019 World Wide Web Conference (WWW '19)*. ACM, New York, NY, USA, 763–771. https://doi.org/10.1145/3308560.3316457

[8] V. d. N. Silva and L. Chaimowicz. 2015. On the Development of Intelligent Agents for MOBA Games. In *2015 14th Brazilian Symposium on Computer Games and Digital Entertainment (SBGames)*. 142–151. https://doi.org/10.1109/SBGames.2015.33

[9] Victor do Nascimento Silva and Luiz Chaimowicz. 2017. MOBA: a New Arena for Game AI. *CoRR* abs/1705.10443 (2017). arXiv:1705.10443 http://arxiv.org/abs/1705.10443

[10] Scott Donaldson. 2015. Mechanics and Metagame. *Games and Culture* 12, 5 (jun 2015), 426–444. https://doi.org/10.1177/1555412015590063

[11] Jinyoung Han, Daejin Choi, Taejoong Chung, Chen-Nee Chuah, Hyun-chul Kim, and Ted Taekyoung Kwon. 2019. Predicting content consumption from content-to-content relationships. *Journal of Network and Computer Applications* 132 (2019), 1–9.

[12] Lucas Hanke and Luiz Chaimowicz. 2017. A Recommender System for Hero Line-Ups in MOBA Games. (2017). https://aaai.org/ocs/index.php/AIIDE/AIIDE17/paper/view/15902

[13] Victoria J. Hodge, Sam Devlin, Nick Sephton, Florian Block, Anders Drachen, and Peter I. Cowling. 2017. Win Prediction in Esports: Mixed-Rank Match Prediction in Multi-player Online Battle Arena Games. *CoRR* abs/1711.06498 (2017). arXiv:1711.06498 http://arxiv.org/abs/1711.06498

[14] S.G. Li and L. Shi. 2013. The recommender system for virtual items in MMORPGs based on a novel collaborative filtering approach. *International Journal of Systems Science* 45, 10 (jan 2013), 2100–2115. https://doi.org/10.1080/00207721.2012.762560

[15] W. Looi, M. Dhaliwal, R. Alhajj, and J. Rokne. 2018. Recommender System for Items in Dota 2. *IEEE Transactions on Games* (2018), 1–1. https://doi.org/10.1109/TG.2018.2844121

[16] Maryam Khanian Najafabadi, Azlinah Hj. Mohamed, and Mohd Naz'ri Mahrin. 2017. A survey on data mining techniques in recommender systems. *Soft Computing* 23, 2 (nov 2017), 627–654. https://doi.org/10.1007/s00500-017-2918-7

[17] Denis Parra and Shaghayegh Sahebi. 2012. Recommender Systems: Sources of Knowledge and Evaluation Metrics. In *Advanced Techniques in Web Intelligence-2*. Springer Berlin/Heidelberg, 149–175.

[18] Yuri Seo and Sang-Uk Jung. 2016. Beyond solitary play in computer games: The social practices of eSports. *Journal of Consumer Culture* 16, 3 (2016), 635–655.

[19] Rafet Sifa, Eric Pawlakos, Kevin Zhai, Sai Haran, Rohan Jha, Diego Klabjan, and Anders Drachen. 2018. Controlling the crucible. In *Proceedings of the Australasian Computer Science Week Multiconference on - ACSW '18*. ACM Press. https://doi.org/10.1145/3167918.3167926

[20] M Sjöblom, J Hamari, H Jylhä, J Macey, and M Törhönen. 2019. Esports: Final Report. *Tampere University* (2019).

[21] Z. Tao, M. Cheung, J. She, and R. Lam. 2014. Item Recommendation Using Collaborative Filtering in Mobile Social Games: A Case Study. In *2014 IEEE Fourth International Conference on Big Data and Cloud Computing*. 293–297. https://doi.org/10.1109/BDCloud.2014.73

[22] Y. Wang, D. Feng, D. Li, X. Chen, Y. Zhao, and X. Niu. 2016. A mobile recommendation system based on logistic regression and Gradient Boosting Decision Trees. In *2016 International Joint Conference on Neural Networks (IJCNN)*. 1896–1902. https://doi.org/10.1109/IJCNN.2016.7727431