electronics

MDPI

Article

Feature Extraction for StarCraft II League Prediction †

Chan Min Lee and Chang Wook Ahn *

AI Graduate School, Gwangju Institute of Science and Technology, Gwangju 61005, Korea; mini0404@gm.gist.ac.kr

* Correspondence: cwan@gist.ac.kr

† This paper is an extended version of our paper presented at SMA'20 "Extracting Control Features to Predict a Player's League in StarCraft II".

Abstract: In a player-versus-player game such as StarCraft II, it is important to match players with others with similar skills. Studies modeling player skills were conducted, with 47.3% and 61.3% performance. In order to improve the performance, we collected 46,398 replays and compared features extracted from six sections of replays. Through the comparison of the six datasets we created, we propose a method for extracting features from a single replay. Two algorithms, k-Nearest Neighbors and Random Forest, which are most commonly used in related studies, are compared. Our research showed a outperforming accuracy of 75.3% compared to previous works. Although no direct comparison has been made with the current system, we conclude that our research can replace the placement games of five rounds.

Keywords: StarCraft II; league prediction; feature extraction

Citation: Lee, C.M.; Ahn, C.W. Feature Extraction for StarCraft II League Prediction. *Electronics* **2021**, *10*, 909. https://doi.org/10.3390/electronics10080909

Academic Editors: Jun Liu and Juan M. Corchado

Received: 27 February 2021
Accepted: 9 April 2021
Published: 11 April 2021

1. Introduction

Commercially successful games tend to provide environments big enough to hold a sufficient number of players to compete [1]. For competitive games such as StarCraft II, playing skills of each player should be taken into consideration as well as game materials [2]. If two competing players differ greatly in their degree of adeptness, one person will almost always end up with every match won. Then, the player will soon lose interest. It is essential to determine each player's skill accurately to match players with similar skills [3]. The main system used in the original game is based on ratings. Rating systems have been employed for the implementation of pairwise assessments of groups and for player-versus-player matchmaking. Each player has a rating that differs depending on the future results and is placed into one of the groups based on their rating [4].

StarCraft II is a real-time strategy game in which the players execute their actions in real time. Players employ a wide range of strategies in order to build their base, gather resources and units, and defeat opponents. Due to various maps, three races, and diverse buildings and units, there are an infinite number of possible situations in-game. Additionally, since players are provided with a large variety of action options at each moment, StarCraft II data have a high level of complexity. Therefore, it is a great place for researchers to try out machine learning algorithms [5].

In StarCraft II, players are distributed into seven different leagues according to their ratings. Regarding the distribution of the player's league, assigning each player a proper rating is essential. However, it is inaccurate with only a few rounds of play. The current ladder system allows each player to start with a default rating. With the results of the first five placement games, the system gives provisional rating and league to players. The placement games show greater variance in the rating than the other results. The weakness of this system is that even if a professional game player wins all of the placement games, he/she would not still be placed at the highest league right away. Beginners also suffer from being placed at the leagues that require higher adeptness than they currently

have [6]. Therefore, as a result, each player would need at least five rounds to be assigned to appropriate leagues.

This research proposes predicting each player's adequate league by extracting data from replays. We selected 14 different features and extracted from each replay. Since we use the average value of features, the way the extraction section of the feature is selected has an effect on performance. We generated datasets in six different sections and performed comparative testing on them. Two machine learning algorithms were applied to classify leagues with extracted data.

2. Related Work

The game replay records the game log and allows it to recheck past games. Through it, human data with higher complexity can be used instead of data from the simulation. StarCraft's game data are useful for testing several algorithms because they have a higher complexity than those of other games [7]. Studies exist that provide game data by constructing datasets with information extracted from replay [7–9].

The most representative task of using StarCraft data is strategy prediction [10–13]. Ben G. Weber and Michael Mateas illustrate a data mining methodology for modeling the opponent's strategy [13]. Their method takes the form of encoding game logs as a feature vector containing the production information of a unit or building. They developed a model to predict the opponent's behaviors by analyzing extracted features. If the opponent's strategy is predicted, it helps develop AI performance through the use of counter strategy accordingly. Similarly, some studies design mathematical models to predict the winner [14,15]. These studies have shown that they can help predict the outcome of the game.

There are also multiple studies focused on player modeling. Siming Liu et al. [16] recognized players through extracted features and Random Forest. T. Avontuur et al. [6] developed a model that predicts skills based on data collected during the early portions of the game. This work is the most similar study to ours, showing an accuracy of 47.3%. The key features of their model were related to the behavior and control of the players. We referred to this result when selecting features. Yinheng Chen et al. [17] showed better accuracy of 61.7%, using macro features related to the economy performance. The disparities in player skills among the leagues have been investigated by Thompson et al. [18]. They reported that there are differences in behavior depending on the player's league. Based on these research studies, we figured that control-related features could identify the leagues that players belong to.

3. Methodology

3.1. Data Collection

The first step was to collect game records of various players for each tier. StarCraft II provides records of previous games played through replays. Replays allow users to check all actions and results that have occurred in the game. We collected 46,398 replays from Spawning Tool, a StarCraft II online community. Since only one instance of each player is created in each replay, a large number of replays are required. This dataset has multiple types of replays including one-versus-one and some with AI. It excluded non-game replays between two human players and also excluded league differences of two or more levels. In case of large differences in skills, the game could be moved to a one-sided way, so features extracted may not include the player's characteristics. The details of the dataset for each league are shown in Table 1.

Table 1. Replay distribution by league.

League	Bronze	Silver	Gold	Platinum	Diamond	Master	Grandmaster
Number	1078	4651	6813	6380	40,607	13,467	9884

The parsing process is necessary to extract and use information from StarCraft II replay files. We used Sc2reader python library to parse replays and get game logs. Sc2reader offers the details about the players' actions and events in the game per frame. There are 14 types of features through Sc2reader, mostly associated with player controls. The first feature is camera switching. The player is only able to see a fraction of the entire map through the camera during the game. It is essential to move the camera in order to understand the game as it occurs simultaneously in several places. Orion Vinyals et al. [19] found out that the camera affects the performance of the agent. Therefore, we expect that the camera number has an impact on human skills and is a important feature. The second is action (APM), which is used as a key feature in a player modeling research [6,20]. Train and Build are the number of times the player has ordered to train units or construct buildings, which are related to the consumption of resources. A control group consists of 4 features: setting (ctrl + #), using (#), adding (shift + ctrl + #), and number. A command consists of 5 features: basic, targeted to unit, targeted to point, update unit, and update point. The last feature is the race of player, which changes the choice of units and buildings that players are provided with. The features other than the race are averaged per second during each section and scaled using minmaxscaler with a minimum value of 1 and a maximum of 10.

We extracted the features from 6 different kinds of section in the game. D1 is data during a combat. The results of combat have a impact on the flow and outcome of the game [15]. In order to extract data during a combat, it is necessary to define the start and end of a combat. In this paper, combat is defined as follows: First units that do not participate in combat, such as the workers and Larva, are excluded. If the difference between the times the units die is less than or equal to 3 s, the units are deemed killed in the same combat. Furthermore, any combat shorter than 10 s is excluded. The average combat length of whole replay is 18 s, so we create two datasets that are 18 s long. D2 and D3 are extracted for a duration of 18 s from the beginning and end of the game. In contrast, D4 and D5 are extracted for 5 min. Finally, D6 is created by extracting the entire section of the game. The description of each dataset is given in Table 2.

Table 2. Descriptions of six datasets.

Dataset	Description
D1	Combat
D2	18 s after the game starts
D3	18 s before the game is over
D4	5 min after the game starts
D5	5 min before the game is over
D6	Entire game

3.2. Evaluation

Two algorithms are used to classify with extracted data in this paper. The first algorithm is k-Nearest Neighbors (k-NN), which suggests that the test data is are with the k nearest training data in the feature space. We chose k-NN because it is commonly used in data mining due to its simple but high performance capabilities [21]. The parameter k is set to 100. The other algorithm is Random Forest, which is an ensemble method that utilizes multiple decision trees. Random Forest has shown outstanding performance in player identification research, so we use the same setting with 100 random trees.

To compare the performance of the algorithm in detail, we calculate accuracy, precision, recall, and F1-score. Each mathematical definition is as follows:

$$\text{Accuracy} = (TP + TN)/(TP + TN + FP + FN) \quad (1)$$

$$\text{Precision} = TP/(TP + FP) \quad (2)$$

$$\text{Recall} = TP/(TP + FN) \quad (3)$$

$$\text{F1-score} = 2 * \text{Precision} * \text{Recall}/(\text{Precision} + \text{Recall}) \quad (4)$$

TP means that the predicted and the actual class are true. TN means that the predicted and the actual are false. FP and FN mean that the predicted and the actual are not matched.

4. Results

We applied two algorithms to all datasets for the league prediction. Table 3 compares the six datasets performances for two algorithms. D6 shows the best performance in all evaluations. However, it can be observed that D4, D5, and D6 show similar results. A paired *t*-test is conducted for detailed analysis, and as a result, there is no significant difference with $p \gg 0.05$. We can also observe that D1, D2, and D3 have a similar performance. and the result of *t*-test is $p \gg 0.05$. Based on the evaluation, we grouped the datasets into two groups of the similar performances (D1 + D2 + D3, D4 + D5 + D6).

Table 3. Comparison of datasets for Accuracy, Precision, Recall, and F1-score.

		D1	D2	D3	D4	D5	D6
k-NN	Accuracy	0.62	0.62	0.64	0.71	0.71	0.73
	Precision	0.67	0.63	0.69	0.75	0.75	0.76
	Recall	0.59	0.58	0.62	0.68	0.68	0.70
	F1-score	0.63	0.61	0.65	0.72	0.72	0.73
Random Forest	Accuracy	0.63	0.64	0.62	0.72	0.73	0.75
	Precision	0.64	0.64	0.67	0.77	0.76	0.78
	Recall	0.59	0.59	0.62	0.72	0.70	0.73
	F1-score	0.62	0.62	0.64	0.75	0.73	0.75

Table 4 shows more detailed performance of two groups. The value of the second group is higher for all evaluations. In particular, the precision in Bronze is 0.25 higher, with an outstanding performance of 0.91. The *t*-test result also shows that their differences are significant with $p \ll 0.001$.

Table 4. Comparison of two groups. The first group contains D1, D2, and D3. The second group contains D4, D5, and D6.

		D1 + D2 + D3				D4 + D5 + D6			
		Accuracy	**Precision**	**Recall**	**F1-Score**	**Accuracy**	**Precision**	**Recall**	**F1-Score**
k-NN	Bronze	0.64	0.66	0.47	0.55	0.77	0.91	0.62	0.74
	Silver		0.61	0.57	0.59		0.76	0.78	0.77
	Gold		0.60	0.62	0.61		0.75	0.77	0.76
	Platinum		0.68	0.59	0.63		0.78	0.72	0.75
	Diamond		0.58	0.68	0.63		0.74	0.82	0.78
	Master		0.66	0.74	0.70		0.77	0.82	0.79
	Grandmaster		0.75	0.58	0.65		0.85	0.70	0.77
Random Forest	Bronze	0.64	0.67	0.46	0.55	0.77	0.91	0.64	0.75
	Silver		0.61	0.57	0.59		0.76	0.78	0.77
	Gold		0.60	0.61	0.60		0.76	0.77	0.76
	Platinum		0.68	0.58	0.63		0.78	0.71	0.74
	Diamond		0.58	0.67	0.62		0.73	0.82	0.77
	Master		0.66	0.74	0.70		0.77	0.82	0.79
	Grandmaster		0.75	0.58	0.65		0.85	0.71	0.77

According to these results, it can be observed that each group is a set of datasets with similar performance, and the performance difference between the two groups is significant. Compared to Table 2, we can see that the two groups can be distinguished by the length of the section. D1, D2, and D3 are extracted from different short sections of 18 s. On the other hand, D4, D5, and D6 are extracted from different longer sections. There is no difference in performance when compared between sections of similar length, but when compared between sections of different lengths, it is better if the length is longer. Therefore, we conclude, first, that the features we used are not influenced by the timing of the extraction section. We extracted and compared three short sections including combat, which are important moments in the game, but they showed the similar performances. Second, we concluded that the length of the extraction section should be longer to show better performance.

This experiment compares the performance of two algorithms using D6, which showed the best performance. Figures 1 and 2 show the confusion matrix of each algorithm. Both algorithms are properly classified in all leagues, and even if they are wrongly classified, we can see that they are close to the correct league. If one level of league error is allowed, all leagues except the Bronze show an accuracy of about 90%. This means that our results allow us to place players into appropriate leagues.

0.53	0.22	0.16	0.04	0.05	0.00	0.00
0.00	0.75	0.16	0.04	0.05	0.00	0.00
0.00	0.10	0.75	0.07	0.07	0.01	0.00
0.00	0.05	0.09	0.69	0.14	0.02	0.01
0.00	0.02	0.07	0.07	0.74	0.08	0.01
0.00	0.00	0.01	0.01	0.10	0.79	0.09
0.00	0.02	0.03	0.02	0.06	0.21	0.66

Figure 1. The confusion matrix for (k-Nearest Neighbor) k-NN.

0.63	0.20	0.10	0.05	0.03	0.00	0.00
0.01	0.77	0.13	0.04	0.05	0.00	0.00
0.00	0.09	0.74	0.07	0.08	0.01	0.00
0.00	0.04	0.08	0.70	0.15	0.03	0.00
0.00	0.02	0.04	0.06	0.79	0.08	0.01
0.00	0.00	0.00	0.01	0.10	0.81	0.08
0.00	0.01	0.03	0.02	0.05	0.21	0.68

Figure 2. The confusion matrix for Random Forest.

The two algorithms are compared in detail based on accuracy, precision, recall, and F1-score. As shown in Table 5 , k-NN outperforms in Bronze class with a precision of 96%. This indicates that if classified as Bronze using k-NN, there is a 96% likelihood that it is correct. Random Forest shows 90% precision in Bronze. In the other classes, Random Forest performs better precision than k-NN. Although k-NN shows outstanding precision in Bronze, Random Forest has better overall precision. In recall, Random Forest performs better recall than k-NN except for Gold class. In contrast to precision, two algorithms perform with low recall in Bronze. Both algorithms have high precision and low recall problem in Bronze, apparently due to its lower number of data compared to other leagues. Table 2 shows that Bronze differs by as little as four times and as much as 40 times from other leagues. F1-score of Random Forest is higher than k-NN in all leagues. Based on these evaluation metrics, Random Forest performs better than k-NN with our features.

Table 5. Comparison of two algorithms. Random Forest performed better than k-NN.

	k-NN				**Random Forest**			
	Accuracy	**Precision**	**Recall**	**F1-Score**	**Accuracy**	**Precision**	**Recall**	**F1-Score**
Bronze	0.73	0.96	0.53	0.68	0.75	0.90	0.63	0.74
Silver		0.68	0.75	0.71		0.73	0.77	0.75
Gold		0.68	0.75	0.71		0.74	0.75	0.74
Platinum		0.73	0.69	0.71		0.75	0.71	0.73
Diamond		0.71	0.75	0.73		0.72	0.79	0.76
Master		0.75	0.79	0.77		0.75	0.81	0.78
Grandmaster		0.82	0.66	0.73		0.83	0.68	0.75

5. Conclusions

In this paper, we propose a method to predict the player's skill with one game replay. As a result of an experiment with six sections of different timing and length, the data from the entire game showed the best performance. Through this result, we can observe that a steady control has a greater impact on the player's skill than an instantaneous control. Comparing the performance with two algorithms, k-NN achieves an accuracy of 72.7%, and Random Forest achieves 75.3%. Our results showed improvement compared to previous studies, namely 47.3% [6] and 61.7% [17]. Random Forest shows better performance in accuracy, precision, recall, and F1-score. We suggest using the entire section of the game to extract features, and we suggest using Random Forest for classification.

In order to use a large number of replays for learning, it was not conducted in detail, such as by country or by race. There are differences in the skills of each league depending on the country, and the strategy varies depending on the race of the match-up. Therefore, it is expected to show better performance considering detailed conditions. Since the results show that the length of the section and performance of the feature are related, we can prove the relationship between length and performance. We failed to compare directly with the system because there was a problem finding the results of the placement games. If the ladder system uses these results instead of the placement games, it can quickly and accurately place players with a single match. Furthermore, applying these results to AI's control will improve AI's performance and help make league-specific AI guidelines.

Author Contributions: Conceptualization, C.M.L. and C.W.A.; methodology, C.M.L.; formal analysis, C.W.A.; investigation, C.M.L.; resources, C.W.A.; data curation, C.M.L.; writing—original draft preparation, C.M.L.; writing—review and editing, C.M.L. and C.W.A.; funding acquisition, C.W.A. All authors have read and agreed to the published version of the manuscript.

Funding: This work was supported by IITP grant funded by the Korea government (MSIT)(No. 2019-0-01842, Artificial Intelligence Gradate School Program (GIST)), and the National Research Foundation of Korea(NRF) funded by the Ministry of Education (NRF-2021R1A2C3013687).

Conflicts of Interest: The authors declare no conflict of interest.

References

1. Lee, C.; Ahn, C.W. *Extracting Control Features to Predict a Player's League in StarCraft II*; SMA: Jeju, Korea, 2020.
2. Palero, F.; Ramirez-Atencia, C.; Camacho, D. Online gamers classification using k-means. In *Intelligent Distributed Computing VIII*; Springer: Berlin/Heidelberg, Germany, 2015; pp. 201–208.
3. Vicencio-Moreira, R.; Mandryk, R.L.; Gutwin, C. Now You Can Compete With Anyone: Balancing Players of Different Skill Levels in a First-Person Shooter Game. In Proceedings of the 33rd Annual ACM Conference on Human Factors in Computing Systems, CHI'15, Seoul, Korea, 18–23 April 2015; pp. 2255–2264. [CrossRef]
4. Sarkar, A.; Cooper, S. Level difficulty and player skill prediction in human computation games. In Proceedings of the AAAI Conference on Artificial Intelligence and Interactive Digital Entertainment, Snowbird, UT, USA, 5–9 October 2017; Volume 13.
5. Justesen, N.; Risi, S. Learning macromanagement in starcraft from replays using deep learning. In Proceedings of the 2017 IEEE Conference on Computational Intelligence and Games (CIG), New York, NY, USA, 22–25 August 2017; pp. 162–169.
6. Avontuur, T.; Spronck, P.; Van Zaanen, M. Player skill modeling in Starcraft II. In Proceedings of the AAAI Conference on Artificial Intelligence and Interactive Digital Entertainment 2013, Boston, MA, USA, 14–18 October 2013.
7. Lin, Z.; Gehring, J.; Khalidov, V.; Synnaeve, G. Stardata: A starcraft ai research dataset. *arXiv* **2017**, arXiv:1708.02139.
8. Robertson, G.; Watson, I. An improved dataset and extraction process for starcraft ai. In Proceedings of the The Twenty-Seventh International Flairs Conference, Pensacola Beach, FL, USA, 21–23 May 2014.
9. Synnaeve, G.; Bessiere, P. A Dataset for StarCraft AI & an Example of Armies Clustering. *arXiv* **2012**, arXiv:1211.4552.
10. Synnaeve, G.; Bessiere, P. A Bayesian model for opening prediction in RTS games with application to StarCraft. In Proceedings of the 2011 IEEE Conference on Computational Intelligence and Games (CIG'11), Seoul, Korea, 31 August–3 September 2011; pp. 281–288.
11. Hsieh, J.L.; Sun, C.T. Building a player strategy model by analyzing replays of real-time strategy games. In Proceedings of the 2008 IEEE International Joint Conference on Neural Networks (IEEE World Congress on Computational Intelligence), Hong Kong, China, 1–8 June 2008; pp. 3106–3111.
12. Schadd, F.; Bakkes, S.; Spronck, P. Opponent Modeling in Real-Time Strategy Games. In *GAMEON*; Citeseer: University Park, PA, USA, 2007; pp. 61–70.
13. Weber, B.G.; Mateas, M. A data mining approach to strategy prediction. In Proceedings of the 2009 IEEE Symposium on Computational Intelligence and Games, Milan, Italy, 7–10 September 2009; pp. 140–147. [CrossRef]
14. Ravari, Y.N.; Bakkes, S.; Spronck, P. Starcraft winner prediction. In Proceedings of the AAAI Conference on Artificial Intelligence and Interactive Digital Entertainment, Burlingame, CA, USA, 8–12 October 2016; Volume 12.
15. Stanescu, M.; Hernandez, S.P.; Erickson, G.; Greiner, R.; Buro, M. Predicting army combat outcomes in StarCraft. In Proceedings of the AAAI Conference on Artificial Intelligence and Interactive Digital Entertainment, Boston, MA USA, 14–18 October 2013; Volume 9.
16. Liu, S.; Ballinger, C.; Louis, S.J. Player identification from RTS game replays. In Proceedings of the 28th CATA, Honolulu, HI, USA, 4–6 March 2013; pp. 313–317.
17. Chen, Y.; Aitchison, M.; Sweetser, P. Improving StarCraft II Player League Prediction with Macro-Level Features. In Proceedings of the Australasian Joint Conference on Artificial Intelligence, Canberra, Australia, 29–30 November 2020; pp. 256–268.
18. Thompson, J.J.; Blair, M.R.; Chen, L.; Henrey, A.J. Video game telemetry as a critical tool in the study of complex skill learning. *PLoS ONE* **2013**, *8*, e75129. [CrossRef] [PubMed]
19. Vinyals, O.; Babuschkin, I.; Czarnecki, W.M.; Mathieu, M.; Dudzik, A.; Chung, J.; Choi, D.H.; Powell, R.; Ewalds, T.; Georgiev, P.; others. Grandmaster level in StarCraft II using multi-agent reinforcement learning. *Nature* **2019**, *575*, 350–354. [CrossRef] [PubMed]
20. Ballinger, C.; Liu, S.; Louis, S.J. Identifying Players and Predicting Actions from RTS Game Replays. In Proceedings of the 28th International Conference on Computer Applications in Industry and Engineering, San Diego, CA, USA, 12–14 October 2015
21. Zhang, S.; Li, X.; Zong, M.; Zhu, X.; Cheng, D. Learning k for knn classification. *ACM Trans. Intell. Syst. Technol.* **2017**, *8*, 43. [CrossRef]