

Identifying the Rush Strategies in the Game Logs of the Real-Time Strategy Game StarCraft-II

Hyunwoo Oh*¹ Teguh Budianto*¹ Yi Ding*¹ Zi Long*¹ Takehito Utsuro*²

*¹Grad. School of Systems and Information Engineering, University of Tsukuba

*²Faculty of Engineering, Information and Systems, University of Tsukuba

This paper studies strategies of StarCraft II and especially classifies them into rush and un-rush. More specifically, this paper proposes how to automatically classify the collection of StarCraft II game logs into those with / without rush strategies. In the proposed method, we evaluate the three types of features, namely, the upper bound of variance of time series numbers of workers, the upper bound of numbers of workers at a specific time, and the lower bound of the starting time of building the second base.

1. Introduction

Real-Time Strategy (RTS) game is one of the famous online computer game genres played by two sides of players who fight each other. Generally, a player is required to collect resources which make the player to be able to create building and train armies for battles. Unlike Go and Chess, RTS game have limited amount of information for the player to anticipate any possible occurrence in the game. The information in RTS game can change rapidly. Moreover, player must complete multiple tasks within a short time. Such characteristics make the task of playing the RTS game as well as developing AI in the RTS game difficult. Thus, so far, human players overwhelm the RTS AI.

Dealing with complex, dynamic, neither perfect nor complete information is one of the most important challenges for artificial intelligence. Road traffic, finance, and weather forecasts are the examples of such large and complex dynamic environments. The RTS game can be seen as a simplification of such real-life environments. In contrast, achieving complex decision making and future circumstances planning in a game with dynamic and limited information is definitely to influence the advancement of real-life artificial intelligence [Ontanón 13]. RTS game AI have been studied through various AI competitions such as “ORTS RTS Game AI Competition” (<https://skatgame.net/mburo/orts/#Competitions>), “AIIDE StarCraft AI Competition” (<http://www.cs.mun.ca/~dchurchill/starcraftaicom/>), and “CIG StarCraft RTS AI Competition” (http://cilab.sejong.ac.kr/sc_competition/).

StarCraft II is a well-known RTS game. The most important task in StarCraft II is to decide strategy which can damage the opponent player by using resources efficiently (see Figure 1). Due to an immature decision, it can happen that low-level players take an inappropriate strategy which can be a cause of a fatal misjudgement. Such a misjudgement can happen even on the game by high-level players.

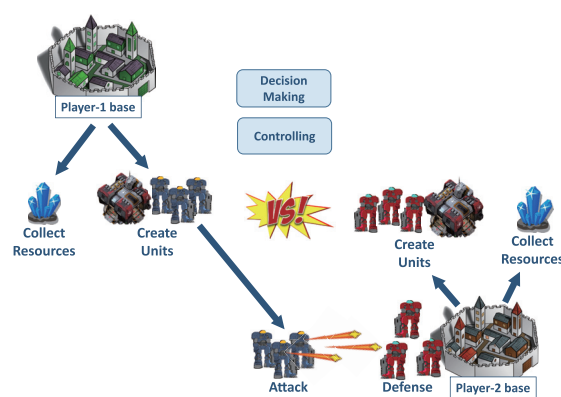


Figure 1: An Overview of StarCraft II

As in the case of well-known Korean professional gamers, human players use various strategies so as to increase the amount of information that contribute to making strategy decision, resulting in their own play style and winning rate. It is one of the important tasks of RTS game AI to efficiently learn the strategies that human players take for the purpose of increasing their winning rate.

Based on the discussion above, this paper studies strategies of StarCraft II and especially classifies them into rush and un-rush. More specifically, this paper proposes how to automatically classify the collection of StarCraft II game logs into those with / without rush strategies.

2. The Real-time Strategy Game: StarCraft II

2.1 Overview

StarCraft II is among one of the most well-known RTS games, which is developed by Blizzard Entertainment. StarCraft II is played by two sides of players, where the goal of the game is to win the battle. The most common match in StarCraft II is the one-versus-one match where the players play against each other in the game. Each player chooses one’s race out of the 3 races provided; Terran, Zerg, and Protoss. Because each race has different strength and weak-

連絡先: 宇津呂 武仁, 筑波大学システム情報系,
〒305-8573 茨城県つくば市天王台 1-1-1, 029-853-5427

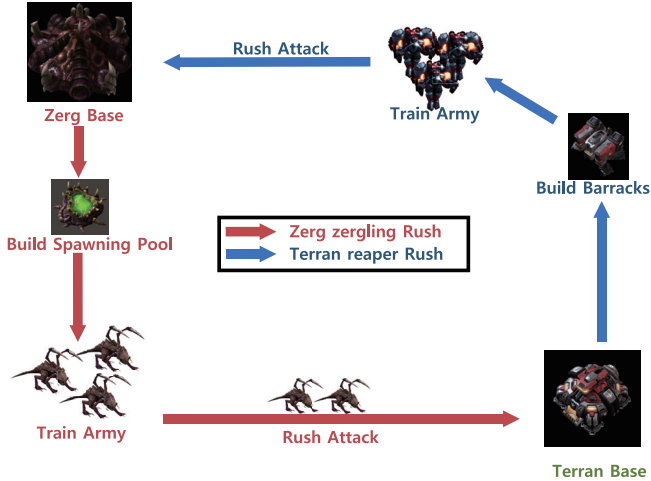


Figure 2: Rush Strategy in StarCraft II

ness, each race has a unique play style.

In StarCraft II, a player is required to collect resources which make the player to be able to create buildings and train armies for battles. In this paper, we divide the strategy elements of StarCraft II into “macro” and “micro” based on time and strategy scale. “Macro” means establishing the overall game strategy, while “micro” means detailed controlling of individual units. There are a lot of strategy elements in “macro”, but we divide it into the followings:

1. “Build Order” — the sequence of building construction over time,
2. “Units Combination” — most efficient combination of units, and
3. “Positioning” for efficient battles.

The “micro” strategy element is detailed controlling of individual units for efficient battles.

2.2 Rush Strategy in StarCraft II

Rush strategies are employed by aiming at defeating the opponent as early as possible starting from the beginning of the game. Figure 2 is an example of the game with a rush strategy. Rush strategies are employed by players of all the levels but their winning rates vary depending on “Build Order” timing and detailed controlling abilities in battles. In un-rush strategies, players spend resources mainly to create workers and buildings for training armies and technologies. The winning rate depends on success/failure of defense from opponent’s attack, the unit combination, and the efficiency of battles.

3. Resource and Data Set

3.1 Resource of Game Logs of StarCraft II

We collect 5,150 one-versus-one game replays of the StarCraft II from the site of <http://www.spawningtool.com>. We only use the game replays from the latest game update; *StarCraft II: Legacy of The Void (LOTV)* because the previous game versions have building and army characteristics that are different from those of the latest game update. All the reply files are converted into human-readable

Table 1: Statistics of Game Log Data of StarCraft II

Game type	Number of game logs
Terran vs. Terran	752
Zerg vs. Zerg	414
Protoss vs. Protoss	243
Terran vs. Zerg	1,567
Terran vs. Protoss	1,314
Zerg vs. Protoss	860
Total	5,150

Table 2: Data Set for Evaluation

Data Set	# rush logs	# un-rush logs
Training Set	70	306
Test Set	67	310

log files by SC2Reader (<https://github.com/GraylinKim/sc2reader>). The game logs are distributed in game matches of 6 types of the pairs of races as shown in Table 1.

3.2 Data Set

The StarCraft II divides their players into the leagues of Bronze, Silver, Gold, Platinum, Diamond, Master and Grandmaster according to the levels of the players, where Bronze is that of the lowest level, while Grandmaster is that of the highest. In this paper, we assume that the leagues in StarCraft II represent levels of the skills of the players. Since it is necessary for us to collect the data of successful strategies from those replays, this paper focuses only on the games of the leagues the higher levels, i.e., Diamond, Master, and Grandmaster. Then, as shown in Table 2, we randomly collected the training set of samples for inside evaluation, as well as the test set of other samples for outside evaluation. Each sample consists of a single player’s game log. Table 2 also shows the numbers of rush strategy games for both training and test sets.

4. Time Series Changes in the Number of Workers

The players who used the rush strategy tend not to spend their resources on the infrastructures like workers, upgrad-

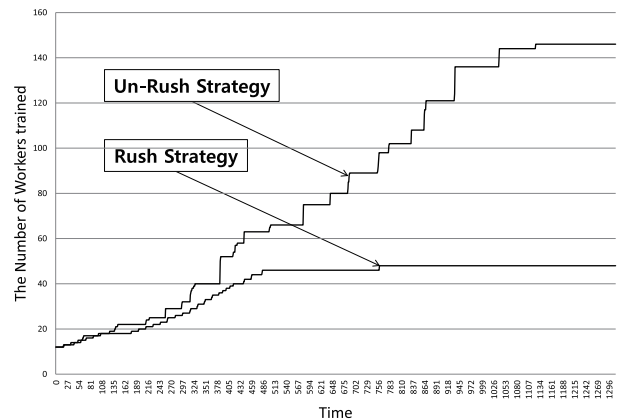


Figure 3: Time Series Changes in the Numbers of Workers of Rush and Un-rush Strategies

Table 3: Features of a Game Log x

Features	Variables of x	
Upper bound of variance of time series numbers of workers $f_{vw}(x; u_0, d_0, e_0) = (x.f_{vw}^v \leq u_0) \wedge (x.f_{vw}^d = d_0) \wedge (x.f_{vw}^e = e_0)$	$x.f_{vw}^v$	Variance of x
	$x.f_{vw}^d$	Time duration of calculating variance [s]
	$x.f_{vw}^e$	End time of calculating variance [s]
Upper bound of numbers of workers at a specific time $f_{nw}(x; t_0, n_0) = (x.f_{nw}^t = t_0) \wedge (x.f_{nw}^n \leq n_0)$	$x.f_{nw}^t$	Specific time [s]
	$x.f_{nw}^n$	Number of workers
Lower bound of the starting time of building the second base $f_b(x; t_0) = (x.f_b^t \geq t_0)$	$x.f_b^t$	Starting time of building the second base [s]

ing buildings, technologies, and resource extraction buildings. Based on this observation, in this paper, we propose several types of features which have close relation to the number of workers of each player. Figure 3 roughly compares typical cases of the time series changes of the numbers of workers between the rush and un-rush strategies. Typically, a player with the rush strategy trains a moderate number of workers, and after that, the player does not continue training more workers, which makes the time series change of the number of workers unchanged. A player with the un-rush strategy, on the other hand, continues training much more number of workers than the one with the rush strategy. Considering the typical situation of the comparison of rush and un-rush strategies, we employ features that are based on variance of the time series numbers of workers as well as the number of workers at a specific time. Here, note that we do not use the number of workers that a player has at a certain time, but we use the number of workers that a player has produced until a certain time. This is mainly because a player can control the number of workers he/she produces at a certain time, while he/she can not control the number of workers he/she possesses at a certain time.

5. Features of A Game Log

Let g be a game replay which consists of the game logs g_1 and g_2 of the player 1 and player 2, respectively. Here, we let x be either of the two game logs g_1 or g_2 :

$$g = \langle g_1, g_2 \rangle, x = g_1 \text{ or } g_2$$

For a game log x , Table 3 defines the three types of features, namely, the upper bound of variance of time series numbers of workers, the upper bound of numbers of workers at a specific time, and the lower bound of the starting time of building the second base.

5.1 Upper Bound of Variance of Time Series Numbers of Workers

As shown in Figure 4, the feature function $f_{vw}(x; u_0, d_0, e_0)$ of the upper bound of variance of time series numbers of workers of a game log x examines whether the variance of time series numbers of workers of time duration d_0 at the end time e_0 of variance calculation satisfies the upper bound u_0 as below:

$$f_{vw}(x; u_0, d_0, e_0) = (x.f_{vw}^v \leq u_0) \wedge (x.f_{vw}^d = d_0) \wedge (x.f_{vw}^e = e_0)$$

Table 4: Evaluation Results (%)

	Recall	Precision	F-measure
Optimal Precision	26.5	90.0	40.9
Optimal F-measure	73.5	64.1	68.5
Optimal Recall	92.7	31.5	47.0

Here, the variance of the time series numbers of workers is measured with time interval of 1 minute.

5.2 Upper Bound of Number of Workers at a Specific Time

The feature function $f_{nw}(x; n_0, t_0)$ of the upper bound of number of workers at a specific time of a game log x examines whether the number of workers at the time t_0 satisfies the upper bound n_0 as below:

$$f_{nw}(x; t_0, n_0) = (x.f_{nw}^t = t_0) \wedge (x.f_{nw}^n \leq n_0)$$

5.3 Lower Bound of the Starting Time to Build the Second Base

In addition to the two feature functions introduced in the previous two sections, which are closely related to the number of workers of each player, we further propose the third feature which is related to the starting time of building the second base.

In StarCraft II, buildings that collect resources, which are called “base”, are very important. If a player needs more resources, then he/she should build the second base as soon as possible. Especially, in the case of un-rush strategies, players need much more resources than those taking rush strategies, which requires those un-rush players to build the second base fast and safely. Conversely, players who use the rush strategy do not necessarily build the second base so quickly. Thus, it is expected that a player with rush strategy tends to build the second base over a certain time.

Based on the observation above, this section introduces the feature of the lower bound of the starting time of building the second base. The feature function $f_b(x; t_0)$ of the lower bound of the starting time to build the second base of a game log x examines whether the starting time to build the second base satisfies the lower bound t_0 as below:

$$f_b(x; t_0) = (x.f_b^t \geq t_0)$$

6. Evaluation

In the evaluation, for each of the three feature functions f_{vw} , f_{nw} , and f_b , combinations of parameters are exam-

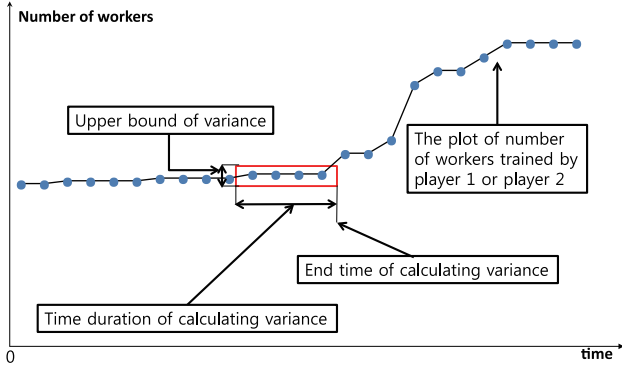


Figure 4: Calculating Variance of Time Series Numbers of Workers

ined. For f_{vw} , combinations of parameters are examined by changing v_0 from 0 to 2, d_0 from 60 to 300, and e_0 from 240 to 360. For f_{nw} , combinations of parameters are examined by changing t_0 from 300 to 600 and n_0 from 25 to 40. Finally, for f_b , parameters are examined by changing t_0 from 60 to 360. For each of the three feature functions f_{vw} , f_{nw} , and f_b , optimal combinations of parameters which give the maximum precision, recall, and f-measure against the training set, respectively, are identified.

Then, as the combinations of the three feature functions, we examine the following two types:

- $(f_{vw} \wedge f_{nw}) \vee f_b$
- $f_{vw} \wedge f_{nw} \wedge f_b$

As for the first type, the condition of $(f_{vw} \wedge f_{nw})$ is used because the small variance of the number of workers may coincide with the small number of workers in total. The disjunction with f_b , on the other hand, is based on an empirical observation that, even if a player with the rush strategy builds the second base, he/she can deceive the opponent and continue the rush strategy without training more workers from that time point. The second type is simply expected to achieve high precision.

For each type of the combinations, we examine 27 combinations of optimal parameters identified with the training set, out of which optimal combinations in terms of precision, recall, and f-measure are again identified with the training set. Here, the first combination type $(f_{vw} \wedge f_{nw}) \vee f_b$ is selected as optimal in terms of precision, recall, and f-measure. Then, with those combinations of parameters, final evaluation results against the test set are shown in Table 4. As can be seen from these results, although its optimal f-measure is around 70%, its optimal precision and recall are as high as 90%.

7. Related Work

There exist a large volume of studies that describe game predictions and analysis in StarCraft. [Avontuur 13] focused on player model prediction to distinguish the level of a player. From their model, it is found that the important features are visuospatial and motor skills of the players. It

is indicated that they can detect the level of the player in a early stage of a game. It can help AI or other human players to adapt with the level of their opponent. [Liu 13] investigates a player’s game style in StarCraft II by applying several machine learning techniques to predict player’s actions. It can help human players to judge what strategy being used by the opponent.

Studies about the prediction of strategies are also introduced [Weber 09, Park 12]. [Weber 09] used data mining techniques to make constructed opponent’s buildings data for predicting the opponent’s strategy. Their works indicate that the importance aspect of analyzing opponent’s buildings information can be a sign of discriminating types of strategies. [Park 12] use a scouting algorithm and several machine learning approaches in order to predict the opponent’s strategy. They apply this approaches to an AI bot that recognizes the constructed building (“Build Order”) of the opponent’s by sending a scout. [Ruíz-Granados 15] developed a model that can predict the winner of StarCraft match at the specific time using the replay information. [Budianto 17] aim at proposing certain actions at some particular conditions of the match using rush strategy in order to help the player to gain more winning opportunity in StarCraft II. But, those work do not provide how to collect a large amount of replay data that uses rush strategy. Our aim is to propose a method of identifying and collecting data for machine learning of human player’s strategies.

8. Conclusion

This paper proposed and developed a method of identifying the games with rush strategy within the RTS game replay log data. In the proposed method, we examine two types of the combinations of the three feature functions, i.e., the upper bound of variance of time series numbers of workers, the upper bound of numbers of workers at a specific time, and the lower bound of the starting time of building the second base.

References

- [Avontuur 13] Avontuur, T., Spronck, P., and Van Zaanen, M.: Player Skill Modeling in StarCraft II, in *Proc. the 9th AIIDE*, pp. 2–8 (2013)
- [Budianto 17] Budianto, T., Oh, H., Ding, Y., Long, Z., and Utsuro, T.: An Analysis on the Rush Strategies of the Real-Time Strategy Game StarCraft-II, in *Proc. 31st Annual Conf. JSAI* (2017)
- [Liu 13] Liu, S., Ballinger, C., and Louis, S. J.: Player Identification from RTS Game Replays, in *Proc. the 28th CATA*, pp. 313–317 (2013)
- [Ontonón 13] Ontonón, S., Synnaeve, G., Uriarte, A., Richoux, F., Churchill, D., and Preuss, M.: A Survey of Real-time Strategy Game AI Research and Competition in StarCraft, *IEEE Transactions on Computational Intelligence and AI in Games*, Vol. 5, No. 4, pp. 293–311 (2013)
- [Park 12] Park, H., Cho, H.-C., Lee, K. W., and Kim, K.-J.: Prediction of Early Stage Opponents Strategy for StarCraft AI using Scouting and Machine Learning, in *Proc. WASA*, pp. 7–12 (2012)
- [Ruíz-Granados 15] Ruíz-Granados, A. S.: Predicting the Winner in Two Player StarCraft Games, in *Proc. the 2nd CoSECVi*, pp. 24–35 (2015)
- [Weber 09] Weber, B. G. and Mateas, M.: A Data Mining Approach to Strategy Prediction, in *Proc. the 5th CIG*, pp. 140–147 (2009)