Player Identification from RTS Game Replays

Siming Liu, Christopher Ballinger, Sushil J. Louis
Dept. of Computer Science and Engineering
University of Nevada, Reno
Reno, 1664 N. Virginia Street, 89557
{simingl, caballinger, sushil}@cse.unr.edu

Abstract

This paper investigates the problem of identifying an RTS game player from their playing style. More specifically, we use machine learning algorithms in the WEKA toolkit to learn how to identify a StarCraft II player from features extracted from game replays. Results reveal that using AdaBoost on a decision tree and Random Forest decision trees perform best on identifying a player from replay data. For a particular player, the results also help us identify the most frequently used strategy against different opponent types and provide some insight into the player's strengths and weaknesses. We believe that these results will help us design better RTS game AI.

1 Introduction

Models of player behavior in real-time strategy games are important for the AI community. If we can learn to recognize a player from the way the player plays the game, we are learning a player model and we can use this model to devise counter-strategies that beat the player. On the other hand, especially if the player is good, we can learn strong winning strategies from analyzing the player's gameplay. Either way, player modeling helps us develop better gameplay strategies. Using board and card games for AI research has a long history since the early fifties when Samuel worked on developing a checkers players [1]. In this paper, we focus on using machine learning techniques to identify a professional StarCraft II player from a database of game replays. We expect this work to help us build better artificial RTS game players.

StarCraft II is one of the most popular multiplayer real-time strategy games with professional player leagues and large databases of professional and amateur replays available on the Internet [2]. Figure 1 shows a screenshot from StarCraft II. In South Korea, there are twelve professional StarCraft II teams with top players making six-figure salaries and the average professional gamer making more than the average Korean.

Figure 1: Screenshot from StarCraft II

RTS games involve spatial reasoning, resource management, and strategic and tactical thinking. A player has to build up an economy to obtain enough resources to generate and support a strong military that will defeat the opponent. Any advances in AI approaches in designing RTS game players will have industrial, military, and social applications. Our overall research goal is to develop competent RTS game players and this paper represents initial research in this direction. We are interested in analyzing professional games to see if professional players share common play characteristics (styles). That is, can we model a specific StarCraft II player from the player's replays? How does this player's style compare with another? What are a player's strengths and weaknesses? In our research, we explore the use of supervised machine learning techniques to identify a StarCraft II player from game replays. Specifically, we apply machine learning algorithms from the WEKA toolkit to features extracted from StarCraft II replays to learn to identify a specific professional StarCraft II player. Preliminary results show that our prediction accuracy on a testing set can be as high as 87 percent using a random forest with one hundred trees. Other analysis with an entropy minimizing decision tree indicates that the player always tries to maximize economic resources early in the game. These results indicate that such analysis is useful both in devising artificial RTS game players and in helping novices learn to be better players.

The remainder of this paper is organized as follows. Section 2 describes related work in RTS games and in player modeling. The next section describes our methodology and features used for player identification. Section 4 contains the results from our research work and the last section provides conclusions and discusses future work.

2 Related Work

StarCraft II was released in 2010 and being a relatively new game, has not been used much for scientific research. Michael Whidby implemented a Python game for studying scouting efficiency in different leagues from one-versus-one games in StarCraft II [3]. His results, for a specific kind of scouting, shows that players in higher leagues scout more than players in lower leagues.

However, StarCraft: Brood Wars, the predecessor to StarCraft II, has been used often for research in the AI community. Ji-Lung Hsieh and Chuen-Tsai Sun applied a case-based reasoning approach for the purpose of training their system to learn and predict player strategies [4]. Ben G. Weber and Michael Mateas present a data mining approach to opponent modeling in StarCraft [5]. They applied various machine learning algorithms to detecting an opponent's strategy in game replays. Then they used the learned models to predict the opponent's strategic actions and timing. If you can predict what your opponent is doing, it is usually fairly straightforward to find a good counter strategy and defeat the opponent. Note that this is the main reason that industrial game AI for RTS games is easy to beat. Predictability is often a fatal weakness in these games.

There is much player identification research in other games. Jan Van Looy and Cedric Courtois studied player identification in online games [6]. They were focused on massively multiplayer online games (MMOGs) and their research did not use game data, rather, it was based on a group of game volunteers, who gathered data on the preferences of their avatar's appearance using survey questions.

Some work has been done in extracting features from replay files. SC2Gears provides a StarCraft II replays parsing service to convert a binary replay file to an XML structured file which we can easily understand [7]. Gabriel Synnaeve and Pierre Bessiere worked on extracting the complete game state from a recorded StarCraft replay file by rerunning the replay file and recording the complete game state through the Brood War API (BWAPI) framework [8]. This approach enables access to the complete game state for every frame in the game. However, StarCraft II does not have such an interface yet so we cannot access its complete game state. We therefore only use the data from player actions in StarCraft II replay files as parsed by the SC2Gear parsing service.

3 Methodology

3.1 Data Collection

One of the challenges of identifying a player is to gather enough game replays from one specific player versus other players. A StarCraft II replay is a file which saves all user action events in a game. These user actions reflect the players' thinking and decision making at every stage during a game and we therefore believe that we can infer the play style and find useful strategic patterns from replays. There are many websites that collect and share user and team contributed game replays and many of these replays are from professional tournaments including MLG, IPL, GSL and other pro-leagues [9, 10, 11]. Therefore, it is possible to collect a representative set of replays for specific professional players. In this early work, we only focus on one-versus-one type of games, because this is the most popular game type for professional matches. There are three different races (Terran, Protoss and Zerg) that a player can choose, each race is significantly different from others in terms of structures and units and thus play styles. In this research, we select a Protoss player because typically Protoss players have very different strategies versus each of the other races. Most Protoss players prefer early attacks when their opponent is Protoss, a more balanced game against Zerg, and a much more economically focused game versus Terran.

We gathered more than 450 replays from SC2Rep.com and GameReplays.org [12, 13]. Half of the games have our specific player, Player 1, and half do not. Since the sources of these replays are from big fans of this game, the data can be noisy. For example, there are Zerg versus Zerg replays mislabeled as Protoss versus Zerg. We had to manually clean up this noisy data and finally ended up with 397 game replay files. The breakdown of these games among the races are shown in Table 1. The first row represents our target player, the second row represents all other players, and each column represents Protoss (P) versus (V) different races (Terran (T) and Zerg (Z)).

Table 1: Game Distribution in Replay Files

	PVP	PVT	PVZ
Player 1	87	64	52
Other players	66	74	54

Table 2: Replay Logs

Frame(Time)	Player	Action	Object
3296	Player 1	Build	Pylon
3588	Player 2	Build	Supply Depot
3625	Player 1	Train	Probe
4804	Player 2	Train	SCV
5638	Player 1	Select	Hotkey 1
6208	Player 2	Build	Barracks
7543	Player 1	Attack	Target position

StarCraft II replay files are stored in a binary format by the game. We need to parse it to a format that we can understand and use. There are several websites that provide parsing services. We used SC2Gears to convert StarCraft II replays to XML structured files [7]. We get all user interface actions from the parsed replay files. Note however, that although we get every mouse click, the complete game state information is not available because some state information is generated by the game engine and not saved in replay files. BWAPI for StarCraft Broodwar could get complete game state, but StarCraft II does not have this interface yet. Table 2 shows a subset of an example game log for a Protoss player parsed from one replay file.

3.2 Feature Extraction

The goal of our representation is to maximize the capture of game information, hopefully including any unique aspects of one specific player compared to other players, so that we can identify the player from others based on these unique characteristics. We create and use a feature vector made of three parts. The first part is general game information which includes game length, winner, and actions per minute (APM). The second part represents the changing state of the game and covers how many units or structures are built in each three minutes time slice. The last part records the time (as a frame number) for the first build or use of each building, unit, and unit ability. It turns out the first use features indicate an important part of the strategy the player used.

Formally, our features are represented by Equation 1 where x is units, structures, upgrades, or abilities, and t is the index of a three minutes time slice with $t \leq 10$.

$$\overrightarrow{F} = \{G, S_x^t, O_x\} \tag{1}$$

Here, $\overrightarrow{F}$, the feature vector is made up of three parts. First, G represents general game information and contains game length, winner, map name, game version, etc. Second, S_x^t represents the number of x produced in time slice t. And third, O_x is given by Equation 2.

$$O_x = \left\{ \begin{array}{ll} f, & \text{frame that } x \text{ was first produced} \\ 0, & \text{if } x \text{ was never produced} \end{array} \right\} \tag{2}$$

In our early experiments, we noticed one very simple distinguishing feature for players. The number of Action Per Minutes (APM) serves to reliably identify specific players with an accuracy of up to 95%. However, this dominating feature only shows how fast this player is clicking the keyboard and mouse, but tells us nothing about how they played in the game and what strategies were used. Therefore, we did not include APM in the rest of our work. In the end, we collected 230 features from each replay file.

3.3 Evaluation

Given our overall research aim, our approach was trying to achieve maximum classification performance from the training set and testing set. We apply various classification and prediction algorithms using the WEKA toolkit from the University of Waikato to explore multiple machine learning approaches to identifying Player 1 [14]. WEKA is a powerful machine learning software that includes many prediction and classification algorithms, as well as preprocessing and regression techniques and toolkits from statistics. We applied the following techniques:

- J48 - C4.5 Decision Tree
- ANN - Artificial Neural Networks
- AdaBoost - Adaptive Boosting
- Random Forest - Ensemble classifier that consists of many decision trees

J48 and ANN use default parameters provided by WEKA. AdaBoost was configured to use the J48 decision tree with default settings. Random Forest was configured to use 100 random trees with four random features.

Table 3: Best Features in PVP

1	Workers built in first 3 minutes
2	Time for first use of chrono boost
3	Blink count in 6 to 9 minutes
4	First Gateway build time
5	First Pylon build time
6	Zealots count in first 3 minutes
7	Observers count in 9 to 12 minutes

Different features represent different play styles and player preferences. Some players prefer to expand quickly, and the number of workers built in the first three minutes turns out to be more important than the time of first appearance of a Stalker (a military unit). Some features may not be used at all in some strategies. For example, Player 1 never used Phoenixes (a flying unit) when playing against Terran, which means that all features related to Phoenix contribute nothing to identifying this player against Terran opponents. Clearly some features are more important than others.

To identify these features and to reduces the size of the decision trees, we decided to reduce the set of features being fed to our machine learners. In addition, discarding noisy and obfuscatory features increases classification accuracy. We therefore used WEKA's attribute evaluator to choose the features with the best predictive ability to help us find the important features for identifying our specific player. Table 3 shows the resulting best seven attributes for Protoss versus Protoss games for Player 1.

4 Results

Several machine learning algorithms were applied to get high identification performance with the WEKA toolkit. We used ten fold cross-validation and all results are on the test-set.

4.1 Player Identification Performance

Our first experiment uses all 230 features with the machine learning algorithms in Subsection 3.3. J48 gets 79.9% accuracy on PVP, 71% on PVT and 61% on PVZ. The reason that we get better results from PVP is that the PVP dataset is the biggest as shown in Table 1, PVZ has the smallest dataset. Since PVP data coverage is better, our classifiers learn better and get better results.

ANN does better than the decision tree. It gets 84.6% on PVP, 81.2% on PVT and 62% on PVZ. We next applied Boosting and Voting to improve performance. AdaBoost and Random Forest get us even better results and the best performance on PVP is 86.6% from AdaBoost. The best accuracy on PVT is 81.2% from ANN and the best performance on PVZ is 71.7% from Random Forest. Figure 2 shows these results for J48, ANN, Random Forest and AdaBoosted J48 with all 230 features. We can explain these results

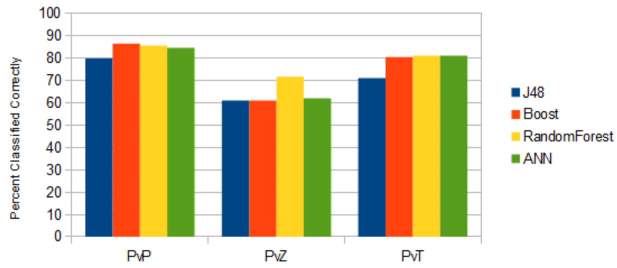

Figure 2: Results of All Attributes

by noting that different features play a different role in different strategies and for different machine learning algorithms. The reason for these differences stem from the fact that a small number of features turns out to be good features (near the root) for building a decision tree to identify Player 1. However, some strategies that Player 1 used only three or four times may not be represented at all in the tree. Therefore, for decision trees, AdaBoost and Random Forest get better results than J48. For ANN, experimentation resulted in our finding the right number of hidden nodes for the default three layer network, so the more attributes we input, the better result we will generally get.

We conducted another set of experiments with the best attributes from the WEKA toolkit's attribute selection and get the seven best attributes for PVP (See Table 3). From the table, we can see that the number of workers built in the first three minutes is very important in identifying this player, and the first time use of Chrono Boost is also a big difference from other professional players. Figure 3 shows the performance of our learning algorithms when using only the best features. The results indicate that Random Forest gets the best results along all three game types with these seven best features. We get 87.6% percent on PVP, 83.3% on PVT and 77.3% on PVZ. This performance is a consequence of how random forest work and our random forest algorithm selects four features randomly. Randomly selecting four from the seven best features should lead to better performance than randomly selecting four from the full 230 feature set.

Figure 4 compares the accuracy of using all at-

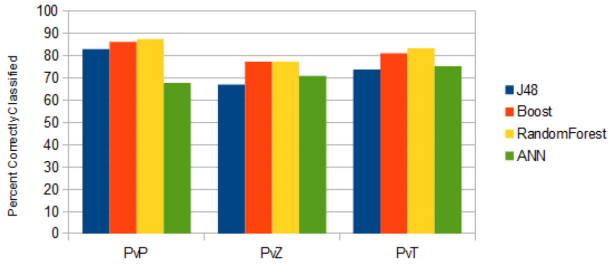

Figure 3: Results of Best Attributes

tributes and using only the best attributes for each algorithm and race. Neural Network have a higher accuracy when using all attributes, while other algorithms have a higher accuracy when given the best attributes. The PVZ bars show that using only the best attributes results in better performance when the dataset is limited.

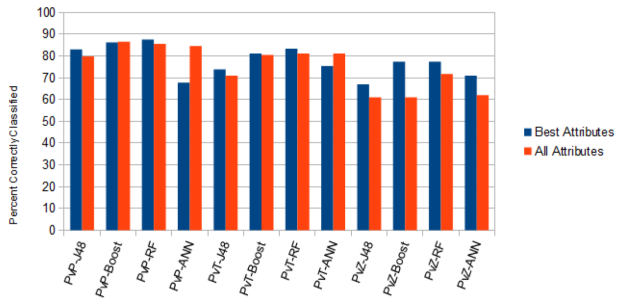

Figure 4: All Attributes vs Best Attributes

There are at least two reasons that we cannot get 100 percent identification accuracy. Our approach uses game replays to extract features that could help us to identify a player's play style. Two players may use very similar strategies and the machine learning algorithms cannot tell the difference. This does happen among top pro players because they actively look to copy the latest and best strategies that they see. Another reason is that the player sometimes chooses an uncommon tactic that is used only once or twice, there are too few cases that machine learning algorithms could use to learn and recognize that player. For example, we found a false negative case where Player 1 used a strategy called a "Cannon rush" in a tournament match and the game ends very quickly in four minutes[1]. The build order and strategic structure are very uncommon for the player and all the algorithms fail on this test replay case.

[1]Most games are about twenty minutes

4.2 Rules and Strategies

The decision tree algorithm used by J48 is not only fast, but allows easy conversion from a tree to a set of rules; Rules which may be better understood by people. On the other hand, the representations used by ANNs, AdaBoosted trees, and Random Forests are harder to understand. We thus analyzed rules generated by high performing decision trees generated by J48. Below is an example of a single rule from PVP games generated by the decision tree. If a game's features matched this rule, then we know it is Player 1, otherwise, it is not Player 1.

- Worker count built in first 3 minutes greater than 24, and
- First Gateway was built before frame 6320, and
- Blink was used less than 12 times between 6 to 9 minutes, and
- First Chrono Boost was used after frame 4692

From the rule, we can see this player always built more workers than others in the first three minutes, always built his Gateway building before frame 6320, and he used the Blink ability fewer than twelve times between six to nine minutes. AI players can be informed by this rule and we can use this rule to uniquely identify Player 1.

We can also get Player 1's strategy preferences against different opponents from the set of rules induced by the decision tree. From these rules we see that strategy versus Protoss is different from strategy versus Terran and Zerg. For example, the Chrono Boost's first time use in PVZ games is later than in PVP games because Player 1 prefers early pressure when his opponent is Zerg and saved Chrono Boost for more quickly pumping out military units - and not for workers (economy). For PVT games, the decision tree is more distributed and has more leaves than PVP and PVZ games. This indicates that the player seems to have more choice of strategy to play against Terran and play is not dominated by one or two strategies as it is in PVP and PVZ games.

5 Conclusions and Future Work

In this paper we described our approach to identifying a specific player from game replay history and achieved good performance. This reveals that we can reliably identify a professional player among other professional player based only on his actions during the game. Our results also indicate that a professional gamer seems to have a unique playing style. The features extracted from replays contain a player's unique characteristics and also the difference between him and other pro gamers.

Results also help reveal important features for a good player. We know that maximizing economic resources is important in the early game, and our baseline should be to build 25 workers in the first three minutes. Therefore, we can improve our understanding to design a better AI player that plays like a pro player, and in future work design game AI that finds a counter strategy to defeat the pro player.

References

[1] A. L. Samuel, "Some studies in machine learning using the game of checkers," *IBM Journal of Research and Development*, vol. 44, no. 1.2, pp. 206 –226, jan. 2000.

[2] (2012) Blizzard entertainment. [Online]. Available: http://www.starcraft2.com

[3] M. Whidby, "Sacovie - zerg scouting game for starcraft ii," *CHI*, may 2012.

[4] J.-L. Hsieh and C.-T. Sun, "Building a player strategy model by analyzing replays of real-time strategy games," in *Neural Networks, 2008. IJCNN 2008. (IEEE World Congress on Computational Intelligence). IEEE International Joint Conference on*, june 2008, pp. 3106 –3111.

[5] B. Weber and M. Mateas, "A data mining approach to strategy prediction," in *Computational Intelligence and Games, 2009. CIG 2009. IEEE Symposium on*, sept. 2009, pp. 140 –147.

[6] J. Van Looy, C. Courtois, M. De Vocht, and L. De Marez, "Player identification in online games: Validation of a scale for measuring identification in mmogs," *Media Psychology*, vol. 15, no. 2, pp. 197–221, 2012.

[7] A. Belicza. (2012) The SC2Gear website. [Online]. Available: https://sites.google.com/site/sc2gears/

[8] M. Buro, "Real-time strategy games: A new ai research challenge," *Proceedings of the 18th International Joint Conference on Artificial Intelligence. International Joint Conferences on Artificial Intelligence*, pp. 1534–1535, 2003.

[9] (2012) Major league gaming. [Online]. Available: http://www.majorleaguegaming.com

[10] (2012) Ign entertainment, inc. [Online]. Available: http://www.ign.com/ipl

[11] (2012) The GOMTV website. [Online]. Available: http://www.gomtv.net

[12] (2010) The SC2Rep website. [Online]. Available: http://www.sc2rep.com/

[13] (2012) The GameReplays website. [Online]. Available: http://www.gamereplays.org /starcraft2/replays.php?game=33

[14] M. Hall, E. Frank, G. Holmes, B. Pfahringer, P. Reutemann, and I. H. Witten, "The weka data mining software: an update," *SIGKDD Explor. Newsl.*, vol. 11, no. 1, pp. 10–18, Nov. 2009. [Online]. Available: http://doi.acm.org/10.1145/1656274.1656278