

Player Modelling in Target-based Games

Yogeshwar Mutneja

Dept. of Computer Science, ITM University
Rajouri Garden, New Delhi, India

ABSTRACT

With people's attention span dropping and computational capacity increasing, artificial intelligence and machine learning are making inroads into the area of gaming. One such technique to make games more interesting, i.e. player modeling has become popular off late. However, most of the work on player and opponent modeling involves strategy games. The aim of the paper is to model the style, and more importantly, even the skill of a player in an entirely different genre, the target based aiming games. It was found that traditional player modeling approaches fail in this genre as real valued continuous output needs to be modelled. The research took a turn to regression methods to model this genre and explore and pick from a number of algorithms. These were next applied to a practical game and evaluated against the humans, on which they were modelled, to judge their performance. The paper concludes by talking about how these methods can also be extended to other similar genres like first person shooters.

General Terms

Image processing, gui automation, opponent modeling, game AI advances, feature selection, model selection.

Keywords

Target-based aiming, player modeling, regression, adaptive game AI.

1. INTRODUCTION

The aim of this paper is to model the game-playing style and skill of a person for a target-based game and hence evaluate the predictive outcomes against a real player to verify its performance. Of late, player modeling has become an important development in the area of gaming. A lot of the previous work on player modelling has been either theoretical or focused on strategy based games. This paper, thus, focusses on a practical application of predictive player modelling in a different game genre.

Games embody one of the major revenue sources for the digital sector of the entertainment industry, every day attracting a large number of new customers and surprising them with tremendous advancements. One of the most important metrics is immersion, that generally relates to how absorbing and engaging a game is. [1-3] Two common approaches to achieve immersion are the use of stunning graphics and the development of a good AI system. For a long time, the game industry has put much of its efforts on the graphics. Unfortunately, the same level of advancement did not happen in AI, which commands the virtual opponent of a player. The games currently in the market still rely on decision trees with almost deterministic sequences of actions and reactions. Difficulty degrees are, in general, offered to provide players with harder virtual opponents. Yet, these degrees are generally associated with just deeper levels that the AI can access in the decision tree. Part of the fun in playing against another actual human opponent depends on the fact that she/he can uncover possible winning tactics and adapt to them, thus the challenge and the fun gets prolonged

considerably. But people may not be available all the time, therefore a strategy to learn a game-playing style of a person becomes of increased value. Player modelling concerns the capturing of characteristic features of a player in a model. Such features may encompass player actions, behaviours, preferences, style, goals, attitudes, personality, and motivations. This has been called preference modelling in the nomenclature defined by Spronck and den Teuling [4]. Player models can also be used to let the game adapt automatically, to be able to achieve its goals better with respect to the player.

Various proposed theoretical methods were evaluated and the most suited one practically applied to an actual game-playing scenario. As the game is a target based one, the outputs that need to be predicted are continuous and real valued. Thus it is the regression algorithms and not the classification ones like in most cases that was used to model the skill and style of the player. Once a suitable learning method was picked, the data was collected and this labelled data was used to train the algorithm. It was then evaluated how the method fared by making both the algorithm and the human player play the game again and comparing the scores in both cases. The game that was chosen for the experiment, Pocket Tanks, is a target-based game which takes power and the angle as an input and calculates the trajectory accordingly to hit the other tank with the chosen weapon. Graphical user interface automation was used to simulate the game using the algorithm for the purpose of score calculation.

The paper follows the following structure. The next section begins by surveying the literature to note previous work done in the field of player modeling and follow it up with the objectives of the game chosen. The following section briefly describes the various regression methods that have been proposed in the literature and chooses the most suitable one for the game. Section 4 describes how the chosen algorithm has been applied to the game and its working. It covers both, data set acquisition and the running of the simulation by gui automation and the trained algorithm. Section 5 observes the outcome of the experiment and compares it with the human player. The final section concludes by talking about future enhancements and work possible in this area.

2. BACKGROUND

2.1 Literature Survey

As mentioned before, great advancements have been made in the field of artificial intelligence for gaming. Many upcoming areas were mentioned by Johannes Furnkranz in his paper [5] such as pattern discovery, learning search control, and player modeling etc. Player modeling was formally accepted as a term when its taxonomy was defined [6]. Under player modeling, Furnkranz talked about opponent modeling in which the computer AI adapts and adjusts to the gaming style studied in this experiment. He further talked about new improvements in opponent modeling for strategy based games like poker. Aiolli and Palazzi also talked about opponent modeling and went on to demonstrate the same for another strategy based game [7]. Ricciardi et al in their paper also talked about opponent modeling but for a fighting game [8].

HY Ong et al [9] did the same for a strategy based game but their approach was quite novel. Their framework used unsupervised learning to find behavior clusters which were, in turn, used with classification algorithms to learn the outcome predictor.

Unlike all the work done in the past, the experiment attempted to model the player instead of the opponent adapting to the style. The aim of this was to capture the skill and style of the player so that one can play against him even without his actual physical presence. Also, unlike choosing a strategy based game which have previously been modeled, modelling a target based aiming game was tried in this paper while also evaluating its performance.

2.2 Game Description

Pocket Tanks is a single-player/multiplayer target based aiming game. It features a turn by turn tank duel. The aim of the game is to strike the enemy tank with the provided weapons. The model was able to aim and thus control the trajectory of the weapon by providing just two inputs: angle and power. The scoring is based on the impact of the shot which in turn depends upon where and how it hits the target tank. The game is played on multiple terrains and at different wind speeds which affect the trajectory of the shot. For simplicity, the weapon was fixed and the wind speed was set to zero. A total of ten volleys is given to each player and the score is calculated in the end.

3. LEARNING ALGORITHM

The results that the experiment is trying to predict are the power and the angle of the shot. Since these are continuous, real valued outputs, it is looking at a machine learning regression problem. There are a number of proposed methods for regression found in the literature. However, each comes with its own pros and cons. These methods are briefly described and the paper decides on the one which would be used for the problem statement.

3.1 Regression Algorithms

3.1.1 Neural Network Regression

Neural Network (NN) is a well-known information processing paradigm that is inspired by the way biological nervous systems, like the brain, can process information. The key element of this paradigm is the structure of the information processing system which is novel. It is made of a large number of intricately interconnected processing elements that work in unison to solve specific problems. Apart from performing classification, they can also be used for regression and are extremely flexible in the different types of data they support. [10] NNs do a decent job at learning the important features from any data structure, without requiring to manually derive features. There are, however, alternatives that are faster, simpler, easy to train, and provide much better performance. Multi-layer neural networks are usually quite hard to train, and require fine tuning of lots of parameters. It is a computationally expensive method, and used only when the number of features are huge.

3.1.2 Support Vector Regression

Support vector machines (SVM) are a kind of supervised learning methods which is applied to classification or regression. SVMs can be used for regression problems by the introduction of an alternate loss function, [11]. The 'loss function' must be altered to include a distance measure. The following figure illustrates four possible loss functions.

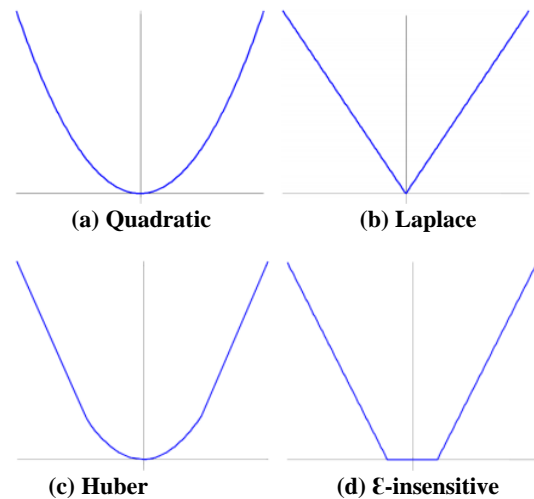


Figure 1: Loss Functions

The loss function in Figure 1(a) depicts the 'least-squares error' criterion which is conventional. The loss function in Figure 1(b) is a Laplacian loss function which is much less sensitive towards outliers than the quadratic loss function. Huber proposed that the loss function in Figure 1(c) would be a robust loss function that has optimal properties when underlying distribution of the data is not known. These three loss functions will produce no sparseness in any support vectors. For addressing this issue, Figure 1(d) had been proposed as an approximation to Huber's loss function which enables a varied set of support vectors to be obtained.

The advantage of support vector regressions is that it requires fewer hyper-parameters [12]. Generally, support vector machines require less grid searching to get an accurate model. SVM with an RBF kernel generally performs quite well. Also, global optimum is guaranteed. However, for most purposes, support vector machines are black boxes. There is some research on interpreting SVMs, but if it will ever be as intuitive as generalized linear models (GLMs) is doubtful. This is a serious problem in some problem domains.

3.1.3 Linear/Polynomial Regression

A linear regression model $y = X\beta + \epsilon$ is a general model used for fitting any relationship that is linear in the parameter β which is unknown [13]. On the other hand, a polynomial term is a quadratic (squared) or cubic (cubed) term which turns a linear regression model into a curve. Since it is X that is squared or cubed, and not the β -coefficient, it still qualifies as linear model [14]. This makes it a simple and a straightforward way to model curves without requiring to model complicated non-linear models. It is computationally cheap for less number of features but as features go up it starts to get really complex.

3.1.4 Regression Trees

Linear regression is known as a global model, where there is a single predictive formula that takes in consideration the entire data-space. When the data has a large number of features which interact in complicated, non-linear ways, assembling only one global model can be quite difficult, and even more confusing when you do succeed. An alternative approach to non-linear regression is to divide, or partition, the space into smaller regions, where the interactions between the data are much more manageable. The divisions are then partitioned again — this is known as recursive partitioning — until finally it results in chunks of the space which are tame enough

that can be fitted into simple models. The global model therefore has two parts: one is only the recursive partition, the other is a simple model for each cell of the partition. Regression trees use it to represent the recursive partition [15]. Each of the terminal nodes, also known as leaves, of the tree represents a cell of the partition, and has attached to it quite a simple model which applies only in that cell. This is a piecewise-constant model and there are several advantages to this. Making predictions is fast and is easy to understand what variables are important in making the prediction. There are fast, reliable algorithms to learn these trees. Furthermore, if some data is missing, it may not be possible to go all the way to a leaf, but can still be predicted by averaging all the leaves in the sub-tree that can be reached. Regression trees have a few limitations as well, such as a lot of effort is required for pruning the trees and the trees can get complicated really fast. The tree formation also requires extra computational efforts.

3.2 Regression Algorithm chosen

The machine learning regression problem this paper is looking at has a very small number of input features. The game is based on a simple physics based simulation environment. Thus, the relation between the features should not be too complex. Hence, it makes no sense to go for a more computationally expensive solution that requires a huge number of training data as well. It also runs the risk of overfitting with some of them, given the small amount of the training data set. Thus, for these mentioned reasons it can be believed that polynomial regression would do justice as the learning algorithm [16].

4. FINAL IMPLEMENTATION

The problem at hand required us to predict two output variables, i.e., angle and power. Due to the obstacle mountain being a hindrance between the tanks, the angle was clearly needed to be set before the power as it is almost always possible to adjust power accordingly. Thus, two separate polynomial regression algorithms were trained and modelled for angle and power and the output of one to the input of the other was connected, apart from the other common features.

Every machine learning problem can be broken down into two parts - collecting the data and, training and implementing the algorithm. Before delving into those tasks, the input features for the chosen regression algorithm were first decided upon.

4.1 Feature selection

The following features were chosen: horizontal distance between the two tanks, vertical distance between the two tanks, and height of the obstacle mountain between the two tanks. These features were used to predict the angle. Once the angle had been predicted, it was also used apart from the above mentioned features as an input feature to predict the power. The problem was modelled as a degree four polynomial and used forward search as a feature selection method [17]. Figure 4 depicts the above features.

The next consideration was the task of collecting the dataset for the given problem. Since the game is not open source,

video processing was used for computing the dataset on which the algorithm would operate for which Python's open CV library was used.

4.2 Dataset acquisition

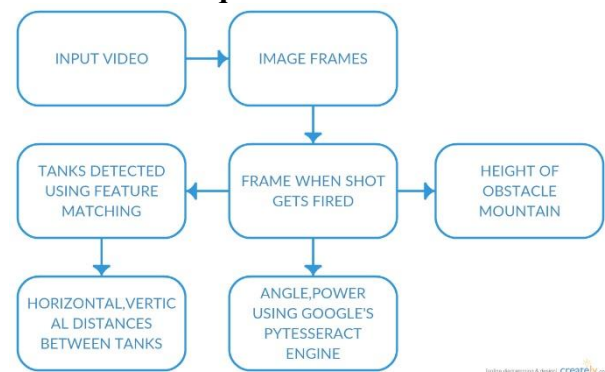


Figure 2: Data Collection workflow

The first task was to break the video into images and analyze every image for the frame when the shot was fired. The shot was detected by matching the specific colour of the shot in the frame. Once the frame was found, the tanks in the image were detected using feature matching using ORB [18] and then differentiated the two tanks by their colours. After detecting the tanks, their vertical and horizontal distances were easily found by subtracting their respective co-ordinate values. As the colour of the terrain is green, the height of the obstacle mountain between the tanks could be calculated by distinguishing between the colours in the frame. Google's tesseract engine was used for optical character recognition [19] to read the angle and power value from the image using the pytesseract wrapper class. The data was collected for two players, one a seasoned player and the other a beginner. Data was collected for 10 games in total, each game having 10 volleys each.



Figure 3: Tank detection



Figure 4: Input features

4.3 Implementing the algorithm

Both these models were trained separately using gradient descent [20]. One third of the data was used for cross validation [21]. They were trained, in turn, by the data collected from observing the seasoned player and then by the beginner player's data on separate occasions. Once the algorithms were trained, they were used to predict and thus run the games independently without human intervention. This was done for 5 games using python's gui automation library: pyautogui. The players were asked to do the same. The algorithm was thus evaluated by comparing the scores from both these runs for both these players. Following is the outcome of the experiment and analysis of how the algorithm fared by comparing the final score to the one scored by the human player on whom the player skill and style modelling was done.

5. EXPERIMENTAL RESULTS & DISCUSSION

Two players played the game - one was a seasoned player while the other was relatively new to the game. Player modelling was done for both the players separately. Five games were played by each and data was collected accordingly. The models of each individual player was then pitted against the computer AI individually. The models had learnt to predict the power and angle for every shot fired. Following are the results of the experiment in a tabulated form:

Table 1: Sample labelled data

Vertical distance	Horizontal distance	Obstacle height	Angle	Power
53	840	399	60	67
93	851	399	77	100
-25	639	314	60	59
-35	625	312	80	100
118	742	354	60	61

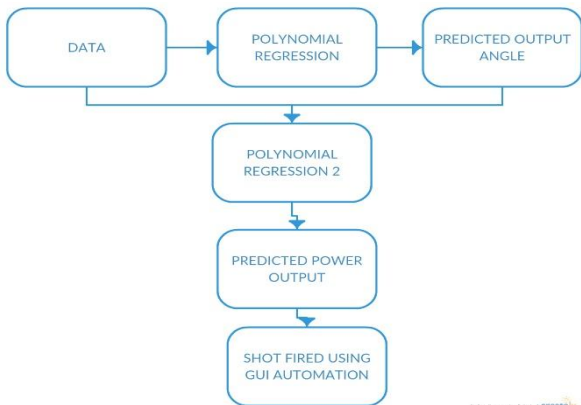


Figure 5: Output Production workflow

Table 2: Seasoned player model predictions

Vertical distance	Horizontal distance	Obstacle height	Angle	Power
112	466	334	76	64
71	953	394	53	64
141	733	340	62	62
132	889	389	59	63

Table 3: Beginner player model predictions

Vertical distance	Horizontal distance	Obstacle height	Angle	Power
86	675	299	81	100
221	510	395	63	50
152	823	369	73	81
150	775	336	73	75

Table 1 shows a sample of the data that was collected from the game. Since this is supervised learning, labelled data was collected. The angle was measured in the clockwise direction. The vertical distance was positive when the first tank is elevated than the second tank. Similarly, a positive vertical

distance implies that the second tank was to the right of the first one. Table 2 and 3 were the values predicted by both the player models when they were pitted against the AI. The players were then asked to play the game again and the final scores from the games played by the player models were compared to that by the actual human players. This is represented in Figure 5 by plotting the line chart. Thus, it can be clearly seen from the data above that the algorithm modelled the players' skills quite accurately. It is also noted that apart from the skills, the players' styles were also modelled, for example, it can be seen from the data that the experienced player relied more on angle manipulations than power manipulations to achieve the same distance with better results.

6. CONCLUSION & FUTURE WORK

It is thus shown how regression can effectively build a player model for target based aiming games. In the future, the aim is on extending this player model that is built for target based aiming games and make the opponent adapt to it as well. Thus opponent modeling [22] for this genre could build on the work done here in the paper. Another improvement that is hoped to achieve in the future is to continue on this idea and move from a static target based game to a more dynamic one like a first person shooter.

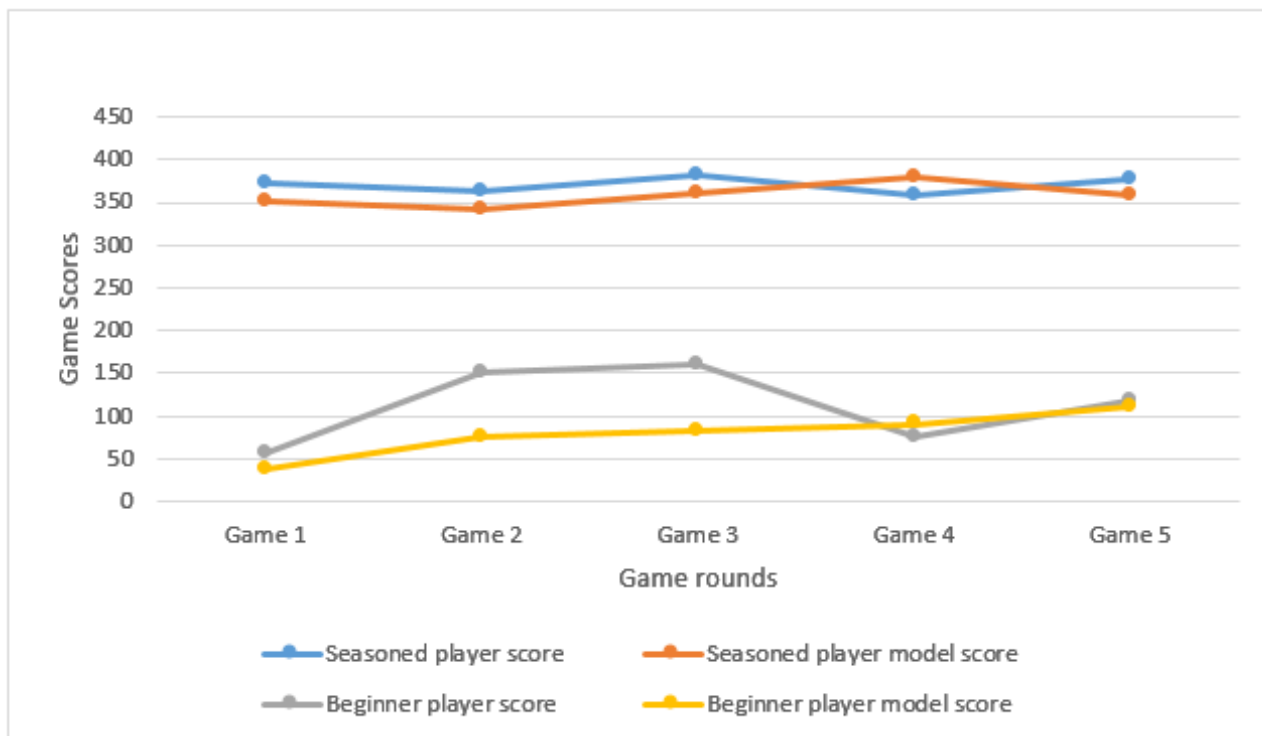


Figure 5: Score comparisons between actual and predicted

7. ACKNOWLEDGEMENTS

I would like to thank Dr. Sangeet Srivastava, Asst. Professor at The Northcap University, for his constant support and immeasurable guidance throughout the duration of the project, and beyond.

8. REFERENCES

- [1] Lev Manovich, "The Language of New Media", The MIT Press, ISBN: 9780262332781, February 2002.
- [2] Taylor, Laurie N. Video games: Perspective, point-of-view, and immersion. Diss. University of Florida, 2002.
- [3] Bakkes, Sander, Pieter Spronck, and Jaap Van den Herik. "Rapid and reliable adaptation of video game AI." *Computational Intelligence and AI in Games, IEEE Transactions on* 1.2 (2009): 93-104.
- [4] Spronck, Pieter, and Freek den Teuling. "Player Modeling in Civilization IV." *AIIDE*. 2010.
- [5] Fürnkranz, Johannes. "Recent advances in machine learning and game playing." *ÖGAI Journal* 26.2 (2007): 19-28.
- [6] Smith, Adam M., et al. "An inclusive taxonomy of player modeling." University of California, Santa Cruz, Tech. Rep. UCSC-SOE-11-13 (2011).
- [7] Aiolli, Fabio, and Claudio E. Palazzi. "Enhancing artificial intelligence on a real mobile game." *International Journal of Computer Games Technology* 2009 (2009): 1.
- [8] Ricciardi, Antonio, and Patrick Thill. "Adaptive AI for Fighting Games." December 12.200 (2008): 8.
- [9] Ong, Hao Yi, Sunil Deolalikar, and Mark Peng. "Player Behavior and Optimal Team Composition for Online Multiplayer Games." arXiv preprint arXiv: 1503.02230 (2015).
- [10] Specht, Donald F. "A general regression neural network." *Neural Networks, IEEE Transactions on* 2.6 (1991): 568-576.
- [11] Smola, Alex, and Vladimir Vapnik. "Support vector regression machines." *Advances in neural information processing systems* 9 (1997): 155-161.
- [12] Welling, Max. "Support vector regression." Department of Computer Science, University of Toronto, Toronto (Kanada) (2004).
- [13] Montgomery, Douglas C., Elizabeth A. Peck, and G. Geoffrey Vining. *Introduction to linear regression analysis*. Vol. 821. John Wiley & Sons, 2012.
- [14] Brandt, Siegmund. "Linear and Polynomial Regression." *Data Analysis*. Springer New York, 1999. 413-426.
- [15] De'ath, Glenn, and Katharina E. Fabricius. "Classification and regression trees: a powerful yet simple technique for ecological data analysis." *Ecology* 81.11 (2000): 3178-3192.
- [16] Kleinbaum, David, et al. *Applied regression analysis and other multivariable methods*. Cengage Learning, 2013.
- [17] Guyon, Isabelle, and André Elisseeff. "An introduction to variable and feature selection." *The Journal of Machine Learning Research* 3 (2003): 1157-1182.
- [18] Rublee, Ethan, et al. "ORB: an efficient alternative to SIFT or SURF." *Computer Vision (ICCV), 2011 IEEE International Conference on*. IEEE, 2011.
- [19] Smith, Ray. "An overview of the Tesseract OCR engine." *icdar*. IEEE, 2007.
- [20] Kivinen, Jyrki, and Manfred K. Warmuth. "Exponentiated gradient versus gradient descent for linear predictors." *Information and Computation* 132.1 (1997): 1-63.
- [21] Picard, Richard R., and R. Dennis Cook. "Cross-validation of regression models." *Journal of the American Statistical Association* 79.387 (1984): 575-583.
- [22] Bakkes, Sander CJ, Pieter HM Spronck, and H. Jaap Van Den Herik. "Opponent modelling for case-based adaptive game AI." *Entertainment Computing* 1.1 (2009): 27-37.