DotA 2 Bots Win Prediction Using Naive Bayes Based on Adaboost Algorithm

Pulung Nurtantio Andono, Nanang Budi Kurniawan, Catur Supriyanto
Faculty of Computer Science, Universitas Dian Nuswantoro, Indonesia
pulung@research.dinus.ac.id, nb.kurniawan@gmail.com, catur.supriyanto@dsn.dinus.ac.id

ABSTRACT
DotA 2 is a multiplayer game that is widely played today. In DotA 2, the players are divided into two teams, i.e., *radiant* and *dire* to against each other. Each team consists of five heroes. One hero is played by human and four heroes are controlled by artificial intelligence (AI). In our prediction, we only collect the statistical data of AI heroes. Each team has the main headquarters, which needs to be protected; the headquarters is a called ancient. When the ancient of a team is destroyed, then the game is over. The features of prediction are collected from the statistical value in each bot. Therefore, the player knows which team (*radiant* or *dire*) is going to be the winning team. To predict the winning team, we use Naive Bayes (NB) as a classifier in data mining algorithm. NB is the appropriate algorithm. Since NB works on probability, it can predict the winning team, not only the winning heroes. However, NB has a shortcoming, e.g., imbalance data, this study proposes to implement NB+Adaboost. This study evaluates some approaches of NB, i.e., discretization and Gaussian distribution kernel function. Both are used to treat the numerical attribute. The results of the experiment show that the highest accuracy of the win prediction by using NB+Adaboost with Gaussian distribution kernel achieves 80%.

CCS Concepts
Information systems → Data mining • Computing methodologies → Machine learning • Computing methodologies → Boosting • Applied computing → Computer games.

Keywords
DotA 2 game; win prediction; naive bayes classifier; adaboost.

1. INTRODUCTION
Related to the study of a game, artificial intelligence (AI) has been proposed to be included in a game, i.e., in strategy management or can be the bots as the enemy or machine against the human. Checkers and Othello are the AI games which are played in individually, and nowadays, many studies developed AI games to be played in a team.

DotA 2 is a multiplayer online game which is a sequel of Defense of the Ancient (DotA). DotA is played by two teams in which each team may have five heroes. Some approaches have been developed to predict the victory of the team or hero. Win prediction is important to recommend the heroes for the players [1].

The aim of this study is to evaluate Naive Bayes (NB) classifier for DotA 2 bots win prediction. NB has been implemented in many cases, such as lyric classification [2], sentiment analysis [3], spam detection [4], etc. NB is appropriate to be used for our experiments because our attributes are nominal and numerical. NB is also capable of predicting the winning team, not only the winning heroes. Since, NB works on probability. Moreover, NB is easy to be implemented compared to another classifier, such as Support Vector Machine, Neural Network, and Decision Tree.

However, the performance of NB is not as good as these classifiers. Therefore, we proposed to apply Adaboost (Adaptive Boosting) algorithm to improve the performance of NB classifier. Adaboost is the popular boosting technique which shows the good performance [5]. The using of NB and Adaboost has been implemented to anomaly intrusion detection [6]. In this study, we try to implement NB and Adaboost to evaluate this approach is applicable for win prediction or not. Indeed, our dataset consists of nominal and numerical attributes which is more challenging to solve with NB. Both discretization and Gaussian distribution techniques are used to treat the numerical attributes in this study.

The remainder of this paper is structured as follows. In Section 2, we describe some related works. Section 3 presents the proposed framework for win prediction. Section 4 describes our experiment and result. In the last section, we give a conclusion on DotA 2 bots win prediction using Naïve Bayes and Adaboost algorithm.

2. RELATED WORKS
Some studies have been published related to game win prediction. In a football match prediction, there are some researchers published their study about football match prediction. Huang and Chang [7] used 2006 World Cup Football Game for the dataset. They used match's attributes such as goals for, shots, shots on goal, corner kicks, direct free kicks to goal, indirect free kicks to goal, ball possession, and fouls suffered. Meanwhile, Shin and Gasparyan [8] and Prasetyo and Harlili [9] used FIFA 2015 football game for the dataset. The win prediction of a match used the player's attributes, such as the score of heading and passing of a player. They stated that by using their model and their data, the football win prediction helps to predict the real soccer match in the real words.

Cho et al. [10] proposed a method to predict the rival's strategy, therefore a player has a chance to win the game. They collected the dataset from StarCraft, a real-time strategy game which constructs the building. The game needs a strategy to destroy the opponent's building. Feature expanded decision tree has been proposed to model the dataset. They stated that machine learning success to predict the strategy of the enemy.

ICCIP 2017, November 24-26, 2017, Tokyo, Japan.

ACM ISBN 978-1-4503-5365-6/17/11…$15.00
DOI: https://doi.org/10.1145/3162957.3162981

Kinkade and Lim [1] proposed to assess Logistic Regression and Random Forest Classifier to predict the victory of a team. In their experiments, by using individual hero performance, the accuracy of prediction achieved 73%. The prior research used numerical attributes whereas our study attempts to evaluate categorical attributes for the prediction.

3. THE FRAMEWORK OF DOTA 2 WIN PREDICTION

In this study, we employ Adaboost algorithm due to the ability to strengthen the weak classifier. Some machine learning algorithms have been used as a weak learner, such as Artificial Neural Network (ANN) and Naïve Bayes [11]. The algorithm of Adaboost is shown below.

Given the training data $(x_1, y_1), \ldots, (x_n, y_n)$, where x_i is the instance of the i-th example. Initialize the number of weak classifier T.

For $t = 1, \ldots, T$

1. Assign the equal initial weight $1/m$ to the all training instances.
2. Train a weak classifier h_t using $w_{t,i}$
3. Get the error of weak classifier

$$\varepsilon_t = \sum_{i=0}^{n} w_{t,i} \left\| h_t(x_i) - y_i \right\|^2 \tag{1}$$

4. Compute α_t

$$\alpha_t = \frac{1}{2} \ln\left(\frac{1-\varepsilon_t}{\varepsilon_t} \right) \tag{2}$$

5. Update the weight

$$w_{t,i} = w_{t,i} \times \begin{cases} e^{-\alpha_t}, h_t(x_i) = y_i \\ e^{\alpha_t}, h_t(x_i) \neq y_i \end{cases} \tag{3}$$

6. Normalize the weight

$$w_{t+1,i} = \frac{w_{t,i}}{\sum_{i=1}^{n} w_{t,i}} \tag{4}$$

where T is the number of iteration or weak learner, α_t is the importance factor of how weak learning contribute to the prediction, ε_t is the error of prediction of weak learner. The weak learner is important when it has large α_t and small ε_t.

As mention in Introduction, we use NB as a weak classifier. NB is a simple classifier, which uses probability to estimates the label of the data. The probability of the data belongs to a category can be computed as follows:

$$P(C_i \mid D) = P(C_i) \prod_{k=1}^{n} P(A_k \mid C_i) \tag{5}$$

where C_i is the set of i classes, D is a test instance which will be estimated its class membership probabilities, A_k is the attribute value of D, n is the number of attributes.

In Naïve Bayes classifier, there are some approaches to treat continues attributes. In this study, we used some approaches, i.e., discretization and Gaussian kernel function. First, we use discretization to transform the continues value into the nominal value. In our case, we use mean value to split the attributes value, i.e., $x < 56$ and $x \geq 56$. Discretization is used for quick way to process the numeric attributes. Although, the discretization is sometimes subjective, i.e., the selection of how many categories discretize the numeric attributes. The second approach, we use kernel function. In this study, we adopt Gaussian distribution. The Gaussian distribution uses mean and standard deviation to process the numerical attributes. The following equations show Naïve Bayes treats the numerical attributes.

$$P(X = x \mid C = c) = f(x, \mu, \sigma) \tag{6}$$

$$f(x, \mu, \sigma) = \frac{1}{\sqrt{2\pi\sigma}} e^{-\frac{(x-\mu)^2}{2\sigma^2}} \tag{7}$$

where f is a Gaussian function kernel, x is the value of attribute of testing instance, μ and σ are mean and standard deviation of attribute in the training data.

Naïve Bayes classifies the data in several steps:

Training step:

1. Compute the probability of each class.
2. Our data contains three attributes (numerical attribute and two categorical attributes). For the numerical attribute, i.e., stat attribute, we perform two approaches: discretization and the kernel function. We need to specify the split value if we perform discretization. However, calculate mean and standard deviation for the numerical attribute if we use kernel function.

Testing Step:

1. Given unknown instance, compute the probability of attribute value, then assign the instance which class has the larger probability.

4. EXPERIMENTS AND RESULTS

This study collects the data from the matches: hero statistic, hero types (melee and range), and hero attributes (strength, agility, and intelligence) to predict the victory of the team (win or lose). Table 1 shows the sample of our dataset. Table 2 and Table 3 show the description of our attributes.

In DotA 2, the game has to be played in two laptops or PCs. In a game, each team has five heroes. One hero is played by human and four heroes are controlled by artificial intelligence (AI). In our dataset, we only collect the data for heroes which are controlled by AI.

Table 1. Sample Dataset

Matches	Teams	Heroes	Stat	Hero Types	Hero Attributes	Prediction
1	Radiant	Juggernaut	68	Melee	Agility	Win
		Zeus	50	Range	Intelligence	
		Vengefull Spirit	58	Range	Agility	
		Bristleback	54	Melle	Strength	
	Dire	Lion	28	Range	Intelligence	Lose
		Draw Ranger	58	Range	Agility	
		Dragon Knight	52	Melle	Strength	
		Walock	56	Range	Intelligence	
2	Radiant	Phantom Assasin	58	Melle	Agility	Win
		Windranger	54	Range	Intelligence	
		Chaos Knight	50	Melle	Strength	
		Sven	60	Melle	Strength	
	Dire	Pudge	53	Melle	Strength	Lose
		Lich	51	Range	Intelligence	
		Luna	49	Range	Agility	
		Jakiro	63	Range	Intelligence	

Table 2. Description of Dataset

		Number of Instance (Heroes)	
Attributes	**Value**	**Win**	**Lose**
Stat	≥56	61	68
	<56	59	52
Hero Types	MELEE	50	58
	RANGE	70	62
Hero Attributes	STR	34	45
	AGI	40	34
	INT	46	41

Table 3. Description of Stat Attribute

Label	Mean μ	Variance σ^2
Win	56,11	26,34
Lose	56,77	26,12

The games were held 45 times. Each game takes 30-40 minutes to collect the result (win or lose). In this study, data collection was conducted in December 2016. There are 45 matches for the prediction. For the experiments, the matches are divided into thirty matches for training and fifteen matches for testing. The prediction model is evaluated by the accuracy as shown in Eq. 8.

$$accuracy = \frac{correctly\ classified\ matches}{total\ number\ of\ matches} \times 100\% \qquad (8)$$

In the prediction, we need three steps to predict the victory of the team: compute the probability of each class, the probability of each hero and the probability of each team.

An example of victory prediction is shown below. The example tries to predict the winning team of matches 1 (Table 1).

Step 1: We need to calculate the probability of each class (probability of win and probability of lose). In our training dataset, there are 30 matches, each match contains two teams, and each team has four heroes. Therefore, it should be 120 heroes win and 120 heroes lose the game.

$$P(Win) = 120/240 = 0.5$$

$$P(Lose) = 120/240 = 0.5$$

Step 2: Calculate the probability of each hero, the sample below show the probability of *Juggernaut* and *Vengefull* Spirit in team *radiant*.

Hero: Juggernaut

$$P(Stat \geq 56|Y = Win) = 61/120 = 0.51$$

$$P(Stat \geq 56|Y = Lose) = 68/120 = 0.56$$

$$P(Hero\ Types = Melle|Y = Win) = 50/120 = 0.42$$

$$P(Hero\ Types = Melle|Y = Lose) = 58/120 = 0.48$$

$$P(Hero\ Attributes = Agility|Y = Win) = 40/120 = 0.33$$

$$P(Hero\ Attributes = Agility|Y = Lose) = 34/120 = 0.28$$

$$P(Win|Juggernaut) = 0.5 \times 0.51 \times 0.42 \times 0.33 = 0.035$$

$$P(Lose|Juggernaut) = 0.5 \times 0.56 \times 0.48 \times 0.28 = 0.038$$

Hero: Vengefull Spirit

$P(Stat \geq 56 | Y = Win) = 61/120 = 0.51$

$P(Stat \geq 56 | Y = Lose) = 68/120 = 0.56$

$P(Hero\ Types = Range | Y = Win) = 70/120 = 0.58$

$P(Hero\ Types = Range | Y = Lose) = 62/120 = 0.52$

$P(Hero\ Attributes = Agility | Y = Win) = 40/120 = 0.33$

$P(Hero\ Attributes = Agility | Y = Lose) = 34/120 = 0.28$

$P(Win | Vengefull\ Spirit) = 0.5 \times 0.51 \times 0.58 \times 0.33 = 0.048$

$P(Lose | Vengefull\ Spirit) = 0.5 \times 0.56 \times 0.52 \times 0.28 = 0.041$

Step 3: Compute the probability of each team, as shown in the following example.

$$P(Win | Radiant) = Juggernaut \times Zeus \times Vengefull\ Spirit \times Bristleback$$

$$P(Lose | Radiant) = Lion \times Draw\ Ranger \times Dragon\ Knight \times Warlock$$

Team: Radiant

$$P(Win | Radiant) = 0.035 \times 0.051 \times 0.048 \times 0.029 = 0.00000248$$

$$P(Lose | Radiant) = 0.038 \times 0.038 \times 0.041 \times 0.039 = 0.00000231$$

Team: Dire

$$P(Win | Dire) = 0.051 \times 0.049 \times 0.029 \times 0.057 = 0.00000413$$

$$P(Lose | Dire) = 0.038 \times 0.041 \times 0.039 \times 0.071 = 0.00000431$$

Based on the sample computation above, the probability of win of team *radiant* is bigger than the probability of lose, and the probability of lose of team *dire* is bigger than the probability of win. Therefore, team *radiant* wins the competition.

In fact, some matches show that two teams may have the same condition in which the probability of win is higher than the probability of lose or vice versa. Therefore, we cannot decide which team is winning or losing. To solve this problem, we compare the probability of win and lose. If the probability of win is higher than the probability of lose, we decide the winning of team based on the probability of win. For instance, the probability of win and lose of team *radiant* is 0.7 and 0.4, the probability of win and lose of team *dire* is 0.9 and 0.1. Both team *radiant* and team *dire*, the probability of win is higher than the probability of lose. Since the probability of win of team *dire* is higher than team *radiant*, the team *dire* will be the winning team and team *radiant* will be the losing team.

Table 4. Accuracy for Win Prediction (%)

Approaches	NB	NB+Adaboost
Discretization	46.6	53
Gaussian Distribution	53.3	80

In our experiment, the accuracy of each approach is shown in Table 4. NB with discretization method gives the lowest accuracy rate. Meanwhile, the higher accuracy rate is achieved by NB+Adaboost. Also, the using of Gaussian distribution kernel yield higher accuracy than discretization method. It is shown in the original NB and NB+Adaboost. In our case, not all classification methods can be applied to predict, such as C4.5 and K-Nearest Neighbor. Since, these algorithms can be applied only for hero win prediction, not for team win prediction.

5. CONCLUSIONS

Artificial intelligence has been included into video games. One of the approaches is machine learning to predict the winning of a team or player. In this study, we demonstrate Adaboosted NB to predict the winning of a team in DotA video game. The accuracy of the DotA 2 win prediction is 80%. Highest accuracy is achieved by NB+Adaboost. The comparison of NB approaches shows that Gaussian distribution kernel outperforms NB with discretization method. Low accuracy of the original NB means that there are many opportunities to produce desirable results in the future study. In the next, an optimization technique can be proposed to improve the accuracy of this study.

6. REFERENCES

[1] Kinkade, N. and Lim, K. y. K.. 2015. DOTA 2 Win Prediction. University of California San Diego.

[2] An, Y., Sun, S. and Wang, S. 2017. Naive Bayes Classifiers for Music Emotion Classification Based on Lyrics. in *16th International Conference on Computer and Information Science (ICIS)*.

[3] Goel, A., Gautam, J. and Kumar, S. 2016. Real time sentiment analysis of tweets using Naive Bayes. in *2nd International Conference on Next Generation Computing Technologies*.

[4] Arifin, D. D., Shaufiah and Bijaksana, M. A. 2016. Enhancing Spam Detection on Mobile Phone Short Message Service (SMS) Performance using FP-Growth and Naive Bayes Classifier. in *IEEE Asia Pacific Conference on Wireless and Mobile (APWiMob)*.

[5] Liu, M. 2010. Fingerprint classification based on Adaboost learning from singularity features. *Pattern Recognition*, vol. 43, pp. 1062-1070.

[6] Li, W. and Li, Q. 2010. Using naive Bayes with AdaBoost to enhance network anomaly intrusion detection. in *Third International Conference on Intelligent Networks and Intelligent Systems*.

[7] Huang, K.-Y. and Chang, W.-L. 2010. A Neural Network Method for Prediction of 2006 World Cup Football Game. in *The 2010 International Joint Conference on Neural Networks (IJCNN)*.

[8] Shin, J. and Gasparyan, R. 2014. A novel way to Soccer Match Prediction, Stanford University: Department of Computer Science.

[9] Prasetio, D. and Harlili. 2016. Predicting Football Match Results with Logistic Regression. in *th IGNITE Conference and 2016 International Conference on Advanced Informatics: Concepts, Theory and Application*.

[10] Cho, H.-C., Kim, K.-J. and Cho, S.-B. 2013. Replay-based strategy prediction and build order adaptation for StarCraft AI bots. in *IEEE Conference on Computatonal Intelligence and Games*.

[11] Zhou, M. and Wei, H. 2009. Constructing weak learner and performance evaluation in AdaBoost. in *International Conference on Computational Intelligence and Software Engineering, CiSE*.