Big Data Analytic on DotA2 to Analyzing Player Roles

Rosdyana Mangir Irawan Kusuma*, K. Robert Lai†
Department of Computer Science and Engineering, Yuan Ze University
Chung-Li, Taiwan 32026, R.O.C.
Email: *rosdyana.kusuma@gmail.com, †krlai@cs.yzu.edu.tw

***Abstract*—State-of-the-art, Computer games are big business, which is also reflected in the growing interest in competitive gaming. *Multi-player online battle arena* games are the most successful competitive gaming in computer games industry. Every player has own specific roles in this team based game. *Dota 2* is the most popular for this games. This paper analyzing player roles and their behavior using *supervised machine learning* and also investigates the applicability of feature normalization to enhance the accuracy of prediction result.We provide an in-depth discussion and novel approaches for constructing complex attributes from low-level data extracted from replay files. Using attribute evaluation techniques, we are able to reduce a larger set of candidate attributes down to a manageable number. *Random Forest* the decision trees algorithm is the most stable and best-performing classifier in our results. Again, *Random Forest* although not the best performing classifier it is proved to be very stable and well suited to this domain.**

I. Introduction

Computer games is a big business. Competitive gaming growing the interest in this business. The competitive gaming events, the so-called electronic sports (sPorts). State-of-the-art, multi-player online battle arena (MOBA) games are the most popular and successful games in this regard. Game tournaments always give significant awards with the big prize of money. Mostly, the professional player can make a living from the prize.

In this paper we analyze the player roles using supervised machine learning (ML) , investigate the applicability of feature normalization to enhance the accuracy of prediction results. We also compare our result with previous paper. Christoph Eggert on his paper said that their most stable and performing classifier is the logistic regression. We will use the same classification method with their work. The different is we will using ROpenDota to build the data set.

An approach to ML in computer games, in general, was proposed by Drachen et al.[1] . They suggest using unsupervised learning algorithms, specifically k-means and *Simplex Volume Maximization*, to cluster player behavior data.

Our investigation about applicability and performance of *supervised machine learning* (ML) to classify player roles based on behavior in *Dota 2*, the most popular MOBA game. Such information could be useful for the game designer to better understand how their game design influences emergent gameplay and player behavior but also for players, both casual and professional, who want to analyze their own performance or who want to learn from the others. It could also support casters and moderators in commentating and presenting matches. Furthermore, this research might hold implications for social and other research concerned with (human behavior in) games. While ML has been applied to games and traditional sports, most works are either interested in questions like spatial behavior, trying to predict the match outcome, or otherwise trying to correlate performance to certain events or behaviors. In contrast, we aim at building a classifier that is largely independent of an individual players performance and that is also not tied to the overall match outcome but that is able to identify a player's role in term of the non-formally defined roles established as common grounds within the *Dota 2* or MOBA community.

This paper contributes to the state of the art in several ways: We provide an in-depth discussion and novel approaches regarding the construction

of complex attributes from low-level data extracted from Dota 2 replay files, together with an evaluation of these attributes with respect to different classifiers. Based on the resulting reduced set of attributes, we compare and discuss the performance of a range of supervised classification algorithms, including logistic regression, random forest decision trees, support vector machines (in combination with Sequential Minimal Optimization), naive Bayes and Bayesian networks, classifying both with a newly established larger set of player roles, as well as with a reduced set inspired by related work [2].

II. Background

Dota 2 is team based game in *Multi-player Online Battle Arena* (MOBA). It is divided by two areas, the *Dire* and the *Radiant*. Each team has five players. Every player will choose their hero in pre-match games. Currently, there is 113 heroes, 166 current items. *Dota 2* game design has separated the heroes with three types. The first is agility heroes type, the second is strength heroes type and the last is intelligent heroes type. Every hero has own specialty and weakness. The items can be combined into a upgrade-able item which is will give more status in heroes abilities.

Hero selection is an important aspect of the game. The team needs to balance the heroes with different abilities that able to fulfill certain of roles and strategies. In the beginning of the game, each player will get 650 gold to buy starting items.

Dota 2 map layout split into three lanes. Each lanes has three defensive towers (see figure 1). the Dire lanes and the Radiant lanes are quite different. the top of Radiant it is called off-lane, meanwhile, in Dire, the top lane is called safe-lane. The bottom lane of Radiant is called safe-lane, while in Dire it is called off-lane. And the middle lane will remain same.

A. Player Roles

Player roles or play styles is the way how the player will choose and try to act as their roles or styles in the games. It is important to note that these roles describe a different facet of play than classic player type classifications (e.g. after Bartle [3]) which aim to classify expressed character traits of the players and were designed to match role-play style games. Comparison of videos, Online guides, and commentary of professional players and commentators are our sources for selection and characterization of the player roles. Player roles definitions and naming conventions will, in any case, differ slightly among the player base and shift over time as the game evolves, constituting another challenge for *Machine Learning* applications in this area. We isolated five player roles for the main *Machine Learning* task, which strike a balance between covering common play styles in great detail while leaving out some exotic styles which are rarely observed or are minor variations of the other styles. The isolated roles were: *Carry* - who are usually weak and need protection early on but are very strong in later stages, often deciding games. Carry typically end up with a high amount of last hits, gold per minute and overall kills, but they can get them in quite different ways. *Mid Laner* - this kind of playing style mostly will get middle lane position since early game. focusing on collecting more gold to obtain high percentage lane of gold in early ten minutes. *Offlaner* - safe lane as their common lane. *Dire* jungle is the safe lane for the Dire team, otherwise, the *Radiant* jungle is the safe lane for the Radiant team. Usually, have a balance between *networth* and efficiently lane gold in early ten minutes. *Roaming Support* - support heroes mostly weak, roaming support will have roaming attribute from replay file, this player role will buy support item such as: *Observer ward* and *Sentry ward* for vision. *Hard Support* - sometimes called the babysitter, because this playing style needs to do everything for supporting the Carry. Mostly, buy support items much more than roaming support.

III. Experiments

A. Data Collection

We are using ROpenDota by Rosdyana Kusuma to built our data set. ROpenDota is an API wrapper service for OpenDota in R. We labeling players from replay and the replay based on The International 2016 and Kiev Major, which contains 8419 labeled players. Using replay data from professional players will guarantee our data set from low-level data.

Fig. 1. *Dota 2* map designed with three lane. Top lane, middle lane and also bottom lane.

Fig. 2. *Dota 2* map jungle areas divided into two areas, *Dire* and *Radiant* jungle.

B. Attribute Construction and Evaluation

Attributes for evaluation is presented in table I. While some attributes correspond directly to summary data, attributes that capture positional information the fighting behavior require more complex processing of the replay data. Different attribute filters implemented in the Java library WEKA [4] to determine the best set of attributes. Our results with the *WrapperSubsetEval* class with best-first search are presented in table 1. We have chosen this algorithm for our final attribute selection because it resulted in the highest accuracy with our labeled data set. For classification, we selected all attributes that were present in at least four folds, excluding assists, as this resulted in the overall highest accuracy. In the following sections, we describe the algorithms and heuristics we developed to calculate the attributes that cannot be directly obtained from replay files. They can be grouped into five rough categories: space and movement, early ganks, team fights, support items, and damage types. Not all of the attributes we describe in this section were finally chosen to be used for classification within this work but they might prove valuable for future works.

TABLE I. Attributes evaluated for classification. Attributes marked with + are not directly available from replays. Attributes marked with * were finally selected for classification. Number of Folds shows in how many folds an attribute was selected by WEKAs WrapperSubsetEval class using 10-fold cross-validation for each classification.

Attribute	Number of Folds
KDA Ratio* = (Kills + Assists) / (Deaths + 1)	10
Last Hits*	10
Early Ganks*+	10
Number of Support Items*+	10
Damage to Neutral Creeps*+	10
Damage to Regular Creeps*+	10
Lane Partners*+	10
Kills*	9
Experience*	5
Deaths*	5
Assists	5
Team Fight Participation*+	4
Early Movement (Visited Cells)*+	4
Damage to Heroes*+	4
Solo Lane+	2
Damage to Towers+	1
Chosen Hero	0
Gold	0

Player Lane Most of the player roles depend on the lane (see figure 1) which is the most active in the early game. Player lane information also provides the constructor for the other attributes, e.g., the *lane partners*. In the early game, players commonly have three main positions: *top lane*, *mid lane*, or *bottom lane*. In additional, the *jungle* areas also can be used (see figure 2). Or, players can have a *roaming* position, it means that they will move around the map instead of focusing in a specific lane.

Lane Partners and Solo Lane To calculate the lane attribute from lane partners and solo lane can be determined by comparing the number of players. The player in *jungle* position or *roaming* are always assigned with zero lane partners, while players who are sharing the other lane, such as top, mid or bottom lane are corresponding with a number of their *teammates* in the current lane. Solo lane attribute included in our attributes as an alternative to the lane partners. The value of this attribute will be true if a player is assigned to one of the tree main lanes without any teammate and otherwise false.

Support Items Player who considering playing as support mostly must buy support items. Many items in *Dota 2* are useful for support player. List of common suppor items: *Courier*, *Flying Courier*, *Observer Ward* and *Sentry Ward*.

Damage Types We categorized damage type based on player roles and requirement of other attributes: damage to the tower, damage to heroes, damage to regular creeps, damage to neutral creeps and damage to ancient creeps.

C. Feature Normalization

An approach of *Feature Normalization* to enhance the accuracy result proved by Singh [5]. We applied feature normalization on this paper to cross check if this methodology can help to tweak accuracy of classification results. Various normalization techniques in this study are:

Z-Score Normalization, A very common techniques to normalize the feature to zero mean and unit variance is Z-score normalization [6]. It is a linear technique which initially mean ($\bar{x}$) and standard deviation (σ) of the specific feature values are computed using:

$$\bar{x} = \frac{1}{N}\sum_{i-1}^{N} x_i \tag{1}$$

and,

$$\sigma^2 = \frac{1}{N-1}\sum_{i-1}^{N}(x_i - \bar{x})^2 \tag{2}$$

the normalized feature is then given by:

$$\bar{x}_i = \frac{x_i - \bar{x}}{\sigma} \tag{3}$$

Min-Max Normalization, This technique performs a linear transformation on the original data. For mapping a value, of an attribute x_i from range [min(x_i), max(x_i)] to a new range [min $x_n ew$,,max $x_n ew$, the normalized feature is given by:

$$\hat{x}_i = \frac{x_i - min(x_i)}{max(x_i) - min(x_i)}(max\, x_{new} - min\, x_{new}) + min\, x_{new} \tag{4}$$

The advantage of Min-Max normalization is that it preserves the relationship among the original data values [7]. In this study $max\, x_{new} = 1$ and $min\, x_{new} = -1$ is used.

Linar Scaling to Unit Range, This is also a linear transformation technique to normalize data in range[0,1]. Given a lower bound min(x_i) and upper bound max(x_i) of an attribute x_i, the normalized value is given by:

$$\hat{x}_i = \frac{x_i - min(x_i)}{max(x_i) - min(x_i)} \tag{5}$$

Linear scaling to unit range is special case of min-max normalization in which max x_{new}=1 and min x_{new}=0.

Softmax Scaling, In addition to linear scaling, nonlinear normalization techniques may be used in cases where data are not evenly distributed around the mean [8]. In such cases, the transformations based on nonlinear (i.e., exponential, logarithmic, sigmoid etc) functions can be used to map the data within specified intervals. One such popular technique is so called softmax scaling which squashes the data value non linearly in the interval [0,1]. the normalized feature is given by:

$$\hat{x}_i = \frac{1}{1 + e^{-y}} \tag{6}$$

where $y = \frac{x_i - \bar{x}}{r\sigma}$ and *r* is a user defined parameter. in the above equation that for small values of y i.e., for values of x_i closer to mean, y is an approximately linear function. Values away from mean are squashed exponentially.

IV. Results and Conclusion

A. Result

We trained and evaluated several different classifiers using 10-fold cross-validation and independent test on our data set. The classifiers were based on existing works and complemented by commonly used classification approaches. It included: *Random forest decision trees*, *logistic regression*, *support vector machines with sequential minimal optimization*, *naive bayes classifier*, and *bayesian networks*. We are using WEKA library and tool for the technical platform. The classifiers were evaluated according

TABLE II. Summary of 10-fold cross -validation and independent test accuracies with feature normalization data set.

Full Classes	**10-Folds Cross Validation**		**Independent Test**	
Original	Random forest	89.37%	Random forest	88.54%
Linear Scale	Random forest	89.77%	Logistic Regression	81.65%
Min-Max Normalization	Random forest	89.53%	Logistic Regression	81.65%
Softmax Scaling	Random forest	80.13%	Random forest	79.81%
Z-Score Normalization	Random forest	89.68%	Random forest	86.52%
Reduced Classes	**10-Folds Cross Validation**		**Independent Test**	
Original	Random forest	94.52%	Random forest	93.53%
Linear Scale	Random forest	94.33%	Random forest	88.48%
Min-Max Normalization	Random forest	94.39%	Random forest	88.78%
Softmax Scaling	Random forest	92.59%	Random forest	92.34%
Z-Score Normalization	Random forest	89.40%	Random forest	92.10%

TABLE III. Comparison with previous works. Value with (*) is the previous work results, and marked by (**) is our work results.

Classifier	**Accuracy**		**Mean Absolute Error**		**Wgt. Avg. AUC**	
Full set of classes	*	**	*	**	*	**
Random Forest	76.27%	89.37%	0.0905	0.0691	0.943	0.984
Logistic Regression	75.85%	82.64%	0.0826	0.1013	0.947	0.962
SMO	75.28%	81.65%	0.1753	0.2503	0.926	0.91
Bayesian Networks	72.03%	70.54%	0.0801	0.122	0.933	0.927
Naive Bayes	70.76%	74.43%	0.0769	0.1055	0.933	0.95
Reduced set of classes	*	**	*	**	*	**
Bayesian Networks	96.58%	92.61%	0.0322	0.0919	0.995	0.973
SMO	96.15%	88.15%	0.2308	0.1185	0.975	0.875
Logistic Regression	96.15%	88.15%	0.0381	0.1662	0.993	0.953
Naive Bayes	95.58%	89.84%	0.0383	0.2319	0.994	0.89
Random Forest	91.17%	94.52%	0.1182	0.0916	0.985	0.987

to several established performance metric (accuracy, mean absolute error (MAE) [9], and area under ROC (AUC) [10].

Table II list the highest accuracy from 10-fold cross-validation and independent test of our works. Random forest is the most stable with good performance from our works. The accuracy result also reasonable. it means the good accuracy is not overfitting after checking the cross-validation with the independent test results.

Table III, the comparison result of our works with previous works. Which is our works is quite better in full set classes. Random forest still dominating for the accuracy result.

Classify the player roles is not easy. We asking *Dota 2* experts to manually classify the player roles. The responses were highly divergent among them. This illustrates that the classification tasks is difficult, even for human experts. Our data set from the professional tournament (Kiev Major 2017 and *The International 2016*) with 8419 data set, We separated into 70% for training data and 30% the rest for testing data. We also classified our data set with the same attributes but with a reduced set of classes (*carry*, *support*) inspired by Gao et al. [11], classification with our attributes achieved a higher accuracy for our data set. A direct comparison is not possible since the data sets differ. Still, the result indicate that classification with a reduced set of classes works well and could already be employed for many applications. Summarizing, we can state that the full set of classes shows promising results but also highlights that, although these classes are accepted by *Dota 2* experts, some are ambiguous even to humans manually labeling the data.

B. Conclusion and Future Work

In this paper, we presented and discussed an approach to apply machine learning techniques for the classification of player roles in *Dota 2*. We believe that our approach should be applicable to another MOBA games, which is share many similar key game mechanics. With 89.37% in Random forest for 10 fold cross validation and 88.54% for our independent test result shows that it is still promising and reasonable result. Applying Feature normalization on our classification has proved that Linear scaling with 89.77%, but sadly, this accuracy result is not really enhanced significantly. Reducing our full classes was very successful with an accuracy of 94.52% on Random forest, which is already suitable for many applications. Again, Random forest proved to be very stable and well suited to this domain. In the future we plan to look more closely at the mentioned problem of identifying the game phase and transitions more reliably. It might also be beneficial to detect roles not for a whole match but rather for phases or sections to account for role changes during the game.

V. Acknowledgment

The authors would like to thank to Muhammad Ramadhan as an expert Dota 2 professional player, my advisor K. Robert Lai, and my Ho Thi Trang.

REFERENCES

[1] A. Drachen, M. Yancey, J. Maguire, D. Chu, I. Y. Wang, T. Mahlmann, M. Schubert, and D. Klabajan, "Skill-based differences in spatio-temporal team behaviour in defence of the ancients 2 (dota 2)," in *Games media entertainment (GEM), 2014 IEEE*. IEEE, 2014, pp. 1–8.

[2] C. Eggert, M. Herrlich, J. Smeddinck, and R. Malaka, "Classification of player roles in the team-based multi-player game dota 2," in *International Conference on Entertainment Computing*. Springer, 2015, pp. 112–125.

[3] R. Bartle, "Hearts, clubs, diamonds, spades: Players who suit muds," *Journal of MUD research*, vol. 1, no. 1, p. 19, 1996.

[4] M. Hall, E. Frank, G. Holmes, B. Pfahringer, P. Reutemann, and I. H. Witten, "The weka data mining software: an update," *ACM SIGKDD explorations newsletter*, vol. 11, no. 1, pp. 10–18, 2009.

[5] B. K. Singh, K. Verma, and A. Thoke, "Investigations on impact of feature normalization techniques on classifier's performance in breast tumor classification," *International Journal of Computer Applications*, vol. 116, no. 19, 2015.

[6] T. Jayalakshmi and A. Santhakumaran, "Statistical normalization and back propagationfor classification," *International Journal of Computer Theory and Engineering*, vol. 3, no. 1, p. 89, 2011.

[7] G. Manikandan, N. Sairam, S. Sharmili, and S. Venkatakrishnan, "Achieving privacy in data mining using normalization," *Indian Journal of Science and Technology*, vol. 6, no. 4, pp. 4268–4272, 2013.

[8] S. Theodoridis, A. Pikrakis, K. Koutroumbas, and D. Cavouras, *Introduction to pattern recognition: a matlab approach*. Academic Press, 2010.

[9] C. J. Willmott and K. Matsuura, "Advantages of the mean absolute error (mae) over the root mean square error (rmse) in assessing average model performance," *Climate research*, vol. 30, no. 1, pp. 79–82, 2005.

[10] J. Huang and C. X. Ling, "Using auc and accuracy in evaluating learning algorithms," *IEEE Transactions on knowledge and Data Engineering*, vol. 17, no. 3, pp. 299–310, 2005.

[11] L. Gao, J. Judd, D. Wong, and J. Lowder, "Classifying dota 2 hero characters based on play style and performance," *Univ. of Utah Course on ML*, 2013.