Classification of Player Roles in the Team-based Multi-player Game Dota 2

Christoph Eggert, Marc Herrlich, Jan Smeddinck, and Rainer Malaka

Digital Media Lab, TZI, University of Bremen, Germany

Abstract. Computer games are big business, which is also reflected in the growing interest in competitive gaming, the so-called *electronic sports*. *Multi-player online battle arena* games are among the most successful games in this regard. In order to execute complex team-based strategies, players take on very specific roles within a team. This paper investigates the applicability of *supervised machine learning* to classifying player behavior in terms of specific and commonly accepted but not formally well-defined roles within a team of players of the game *Dota 2*. 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. Based on this resulting set of attributes, we compare and discuss the performance of a variety of supervised classification algorithms. Our results with a data set of 708 labeled players see *logistic regression* as the overall most stable and best performing classifier.

Keywords: multi-player games, player roles, classification

1 Introduction

Digital games have become an important social, cultural, and economical factor. Online multi-player games attract especially large player bases and big audiences. Computer games have also matched many traditional media in terms of total revenue [14]. This is also reflected in the growing interest in competitive gaming, the so-called *electronic sports* (eSports). Stemming from it's early roots in the 1990s, it has only been in recent years that *eSports* has been showing signs of becoming a mainstream phenomenon. Parallel developments, e.g. the success of game-related online videos in the form of so-called *Let's Plays* and live broadcasting, also play an important role as an indicator and multiplier for societal impact. Game tournaments award significant prize money and there are players and teams that can make a living from playing games. *Multi-player online battle arena* (MOBA) games are among the most popular and successful games in this regard. Due to their popularity, their competitive nature, as well as their complex team-based strategies and tactics, they share many similarities with traditional physical team sports, and akin to the recent rise of data analysis in physical sports, data analysis and machine learning begin to play an important role for the development and analysis of digital games.

In this paper we investigate the applicability and performance of *supervised machine learning* (ML) to classify player behavior in terms of specific roles within a team of players of the game *Dota 2*, a popular contemporary MOBA game. Such information could be useful for game designers 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 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 individual player performance and that is also not tied to the overall match outcome but that is able to identify a player's role in terms 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 [5].

2 Related Work

We restrict the discussion to three main areas: traditional physical sports, comparable works that focus on different games or genres, and works that also focus on *Dota 2*, yet have different classification goals.

Recognizing behaviors in traditional sports typically requires some form of image processing or other recognition techniques to extract usable data. While there is no need for image processing in our case since positional information is directly available, there are similarities that might be applicable to MOBA games. Tovinkere et al. [16] make use of the trajectory of the ball in soccer games to detect events. Combined with player positions and a rule-based-system, which was built with domain-specific knowledge, this leads to a large number of detectable events. A very similar approach is presented by Li et al. [10] for ice hockey games. A notable domain difference compared to works for soccer is that not only the position of the goal is considered, but also the moment when the blue line is crossed. In basketball games, as presented by Fu et al. [4], the actual tracking of the ball is less important for certain tactics. In order to detect offensive strategies they make use of the fact that defenders are closer to their basket than the offensive team to predict ball possession. The strategy is then recognized by comparing player positions relative to each other with expected patterns. While in MOBA games there is no ball or puck that could be tracked, using relative player positions might be applied to certain events (e.g. team fights; see section 6). In addition to positional information, Zhu et al. [21] also utilize information like score boards and game time that are typically on display during TV broadcasts to improve their predictions for soccer games. *Dota 2* replays (see section 5) contain similar information, e.g. amounts of damage, kills, healing etc. that could be combined with positional information.

An approach to ML in computer games in general was proposed by Drachen et al. [2]. They suggest using unsupervised learning algorithms, specifically k-means and *Simplex Volume Maximization*, to cluster player behavioral data. They use two very different games for a proof of concept. Knowledge of the game design is used with both titles to define the attributes used for the algorithms. This also means that the chosen attributes strongly differ. In contrast to our work, their goal is specifically to aid developers in terms of general game design, for example by finding underused mechanics. Because of that a few game specific attributes are selected and unsupervised learning is applicable. Our approach targets behavior that is not tied as directly to just one or two mechanics. Therefore we expect that we need a larger set of attributes and employ supervised learning methods. Other notable works are based on the real-time strategy game series *Starcraft*. Liu et al. [11] target the identification of a specific player from *Starcraft 2* replays by his or her personal play style. In contrast, we are looking for players behaving according to a certain common role, which can be seen as trying to remove personal play style and performance as noise. Synnaeve et al. [15] present a method for an adaptive artificial intelligence (AI) in *Starcraft: Brood War*, which uses similar features. Instead of units, their method collects data on the produced buildings to recognize build orders. The prediction is then made with a Bayesian model. Their work in turn is partly based on the works of Weber et al. [17], who use produced buildings, units and upgrades as attributes. In MOBA games we cannot rely on such attributes alone. The most similar attributes to the production in *Starcraft* would be the items players are buying for their hero. However, unlike *Starcraft*, players almost never have a fixed income, which has a big influence on the items players are buying. Items are also often more connected to specific heroes than to player roles.

A notable work on *Dota 2* is presented by Gao et al. [5]. They target the identification of both the heroes that players are playing, and the role they are taking. They define a basic model with three roles a player can fulfill that are predicted with an accuracy of about 74%. For comparison in addition to our more complex set of classes, we also applied our attributes to the reduced set of roles by Gao et al. (see section 7) with signification improvements in terms of accuracy over their results. It must be noted, though, that we had no access to their test data. Other works about *Dota 2* are mostly based on skill-related questions or social studies. For example, Pobiedina et al. [13] come to the conclusion that the national diversity of players as well as the number of friends playing together has a significant influence on team success. Nuangjumnonga et al. [12] research correlations between the leadership behavior (such as authoritarian and democratic) and the roles the players are fulfilling in the game. Notably they use the roles *Carry*, *Support* and *Ganker*, which we will also cover in section 6. Yang et al. [20] identify combat patterns to predict game outcomes with an accuracy of 80%. More recently, another contribution by Drachen et al. [3] investigated skill-based differences in the spatio-temporal team behavior of *Dota 2* matches. They find higher-skilled players to move more actively and closer to their teammates around the map. For collecting positional information they make use of a spatial division of the *Dota 2* map into zones, looking at zone changes, which shares some similarities with our method to detect early game movement (see section 6) that focuses on the number of entered zones.

3 Background: Dota 2

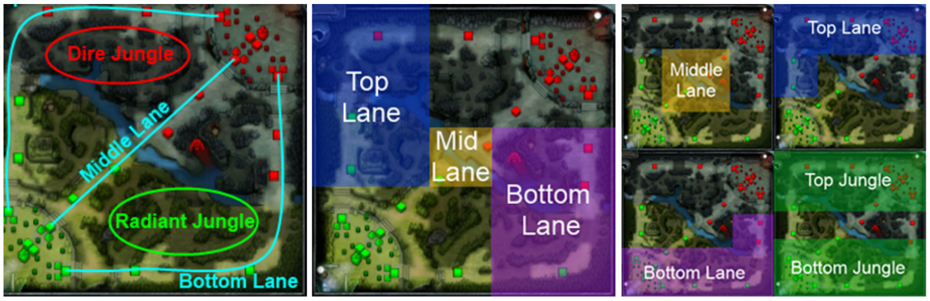

Fig. 1. Left: *Dota 2* map layout with lane annotations Mid: Areas used for determining the player lane Right: Area masks to reduce false positives for early gank detection

Based on the popular modification *DotA* (Defense of the Ancients) for the game *Warcraft 3*, *Dota 2* is is a typical example of the MOBA genre. Most popular MOBAs (including e.g. League of Legends, Heroes of Newerth, Smite, etc.) are identical in terms of the basic gameplay but differ in specific details, e.g. heroes, skills, additional mechanics, graphics, maps, etc.. *Dota 2* is played in teams of five. Each player controls a *hero* character with specific strengths and weaknesses, abilities, matching items and so on that is picked from a large pool at the beginning of each round. The choice of a specific hero is an important aspect of the game. Teams need a balance of heroes with different abilities that are able to fulfill certain roles with respect to the team tactics and strategies, resembling traditional team-based sports. Although the core setup appears simple, it can lead to a large variety of complex team-based behaviors, roles, and strategies. Heroes develop their abilities in a heterogeneous manner and become stronger throughout the game by collecting experience points and gold, which the players can invest into items that support the heros abilities or provide other advantages to the team. As shown in figure 1 (left), the map is split into three lanes. Each lane has three defensive towers (green / red squares in figure 1) (e.g. left) that constantly attack enemies within their range. The goal of the game is to destroy the enemys main building (the *Ancient*) after destroying all towers leading to it on a lane. Along each of the lanes, a wave of *non-player characters* (NPC), called *creeps*, runs from the base of each team to the base of the enemy. Creeps are important sources for gold and experience and can be utilized for attacks. Finishing blows to them are called last hits and are often used as an efficiency benchmark. Additionally, there are camps of neutral creeps (hostile towards both factions), in the map areas marked as *Dire Jungle* and *Radiant Jungle* in figure 1 (left). Another possibility to earn gold and experience is killing enemy heroes. Players lose some gold with each death and have to wait a certain amount of time until their hero is revived at their base. Surprise attacks on enemy players from behind while they are dealing with creeps are a common tactic. In *Dota 2* this behavior is called *ganking.* Each match is separated into *three phases*, called the *early game*, *mid game* and *late game.* There are no exact time thresholds for these phases and transitions (based on the behavior of the players and the strength of their heroes) can be subtle. The *early game* typically lasts for about 10-15 minutes. In this phase players mostly stay on their side of the map and collect experience points and gold by killing creeps. The *mid game* is the game phase that differs the most in every match. It depends heavily on the heroes chosen by each team. At some point in the game, heroes get so strong that even towers are no real threat to them anymore and heroes will be fully developed and equipped to their maximum abilities. This phase is called the *late game.*

4 Player Roles

There are recurring *roles* (or play styles) that players choose and try to follow for a specific game. These roles are not formally defined but have established themselves informally among players. It is important to note that these roles describe a different facet of play than classic player type classifications (e.g. after Bartle [1]) which aim to classify expressed character traits of the players and were designed to match role-play style games. Our selection and characterization of roles is based on a comparison of online guides, videos and commentary of professional players and commentators. Definitions and naming conventions of player roles will in any case differ slightly among the player base and shift over time as the game evolves, constituting another challenge for ML applications in this area. We do not view this mutability as a limiting factor but rather as a realistic constraint. We provide characterizations and in-depth explanations on the attribute calculation and selection (section 6) in order to facilitate reproducibility and comparability with other classification schemes.

In the end, we isolated nine player roles for the main ML 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 other styles. Additionally, we employ a set containing only three rather general roles that have been used in other work for comparison. In general, we tried to avoid performance-based characterizations or attributes as much as possible as we were not interested in distinguishing bad from good players. The isolated roles were: *Carries* - who are usually weak and need protection early on, but are very strong in later stages, often deciding games. Carries 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. Therefore, we define two kinds of carries called *active carries* and *farming carries.* Active carries engage enemy players and participate in team fights to gain experience and gold, while farming carries focus on utilizing enemy or neutral creeps for character development. *Gankers* try to waylay enemy heroes with surprise attacks, sometimes very early in the game. *Support* players in different ways try to help other players, sometimes even sacrificing themselves. We define three kinds of support players to cover different strategies. *Babysitter support* players protect teammates (usually a carry), staying very close to them. In contrast, *roaming supports* are active around the map and even waylay other players similar to gankers. However, they still let other team members take the greater share. *Farming supports* also take their share of experience and gold. However, they spend their gold on support items and they avoid interfering with the carry. *Pushers* continuously try to clear out enemy towers, thereby pushing their lane. *Feeders* are players that somehow get taken advantage of or show very bad performance during the whole game. This is a special role we added to be able to separate such players that do not show any useful observable behavior. *Inactive* players represent another special class of players that - due to technical difficulties or other reasons - do not actively participate.

5 Data Collection

Dota 2 games are stored as replay files (*replays*). Replays contain all low-level game events that occurred during a game and allow the engine to re-simulate whole games. This approach has different advantages and disadvantages. It is very flexible because watching a replay is not limited to watching only one player's perspective or watching at the same speed as the original game. However, this comes with the cost that it is not simply playing back a recording, but the full game logic simulation has to be processed. For our goal this also means that, while low-level events and some additional data needed for attribute construction can be read directly from the replays, constructing some attributes will require significant additional processing (see section 6). We built our attribute construction processing on top of the Java-based replay parser *Clarity* by Martin Schrodt.

Adequately labeling replays requires watching the whole game, sometimes several times for different players. This is a time-consuming task. Therefore we designed and implemented a tool to crowd-source the labeling (of the play style) to the *Dota 2* player community. The tool allows anyone to quickly upload labeled match summaries, based on local replay files, to an online database. The tool was advertised by calls to the player community through established community websites and available for download from our website. Players were free to label whatever games they liked, which could be either their own games or games taken from other sources like online replay archives. For the second community call we also asked users to label a specific, seemingly problematic game (in terms of ambiguous player roles) to gain more insight into the issues as discussed in section 7. The labeled replays contain a large variety of players from different skill levels. In addition, we manually labeled a set of replays from the tournament *The International 2014*, which contains only replays of highly professional players. Overall our final data set contained 708 labeled players.

6 Attribute Construction and Evaluation

The full set of attributes we considered for evaluation is presented in table 1. While some attributes correspond directly to low-level events or summary data, attributes that capture positional information and fighting behavior require more complex processing of the replay data. We experimented with different attribute filters implemented in the Java library WEKA [7] to determine the best set of attributes using our labeled examples. These include algorithm-independent subset selections, for example the *CfsSubsetEval* class based on the works of Mark Hall [6]; algorithm-dependent subset selections, most notably the *WrapperSubsetEval* class based on works from Kohavi et al. [9]; and several classes, such as *InfoGainAttributeEval*, as presented in the works of Witten et al. [19, p. 487-492]. 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.

6.1 Space and Movement

Player Lane Many roles depend on the *lane* (see figure 1) a player is *most active* in during the *early game*. This information also provides the foundation for other attributes, e.g., the *lane partners*. Players typically have three main positions at the beginning of the game: *top lane*, *mid lane*, or *bottom lane*. Additionally, the *jungle* areas can be used, or players can have a *roaming* position, meaning that they move around the map instead of staying in a certain area. The typical areas in which players are positioned most of the time for each lane according to our

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

Table 1. 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 WEKA's WrapperSubsetEval class using 10-fold cross-validation for logistic regression.

observations can be seen in figure 1 (middle). During the very early game (0:30-6:00) the position of each player is checked every two seconds and for each player a region counter is increased if the player is present. Players that are positioned in one of the three lanes at least half of the time are assigned to the *corresponding lane*. If this is not the case, they are flagged as *roaming*. Unfortunately, trying to detect the *jungle* position with the same approach does not work as these areas would necessarily overlap with the areas of the lanes. Instead, the damage that players do to *neutral creeps* is tracked (see *damage types* below). If an empirically determined damage threshold of 6000 is surpassed, the position is set to *jungle* regardless of any other positional information.

Lane Partners and Solo Lane The number of lane partners is determined by comparing the *lane* attribute calculated as described above between players. Players in the *roaming* or *jungle* position are always assigned *zero* lane partners, while players sharing the top, mid or bottom lane are assigned the corresponding *number of* their *teammates* in the same lane. We also included a *solo lane* attribute as an alternative to the lane partners. This attribute is always true if a player is assigned to one of the three main lanes without any teammate and otherwise false.

Early Movement Some roles are characterized by how active players are on the map during the early game. Although this attribute might depend on skill as is indicated by Drachen et al. [3], this could also be caused by active play styles being more prevalent in higher-skilled matches. Unfortunately, the movement activity cannot simply be determined by tracking the total movement of players, as all characters are usually running back and forth even when they do not change their general position. For this reason, we make use of the structure of positional information in *Dota 2* replays. Positions in replays are specified by a 128x128 grid and additional offsets. As we do not need the accurate coordinates, which might also introduce a lot of noise into the classification, we divide grid positions by ten, effectively resulting in a grid consisting of 13x13 cells. During the early game, we count the *total number of cells* that each player visits and assign it to the respective attribute.

6.2 Early Ganks

Early ganks are a key indicator for aggressive player roles, such as *roaming supports* and *gankers*. We collect every fight that is not considered a *team fight* (see below) first. Players are considered to be within the fighting area if they are within 10% of the map size. We use a time threshold of 5 seconds to find the end of a fight and an empirically determined damage threshold of 100 to avoid false positives. Within the collection of fights we detect *early gankers* based on the *lane* attribute described in section 6.1. Based on the assigned lane, we define extended areas in which players are expected to fight, which can be seen in figure 1 (right). If a player is participating in a fight that is not taking place in the expected area for their lane, we increase the corresponding *early gank* attribute by one. For players in the *roaming* position, every fight participation therefore increases the attribute. Due to expected false positives this attribute does not necessarily reflect the exact number of early ganks but a large value should reliably provide a strong indication for them.

6.3 Team Fights

Team fights are typically considered to involve most players of both teams and often decide the outcome of a match. In a standard match there is usually at least one player of each team assigned to each of the three main lanes. If both teams decide to assign both of the remaining two players to the same lane, this leads to a situation where fights involving six players might happen early on. However, these types of fights should not be considered team fights according to the description above. Therefore, we define a minimum of seven players to participate in a fight to label it as a team fight. In addition, we use empirically determined *spatial*, *damage* and *time thresholds* to extract team fights from the attack events contained in the replays files. For each such event a fight entry is instantiated that at first contains only the two players directly involved in the event. Additional players are added to the fight if they either take or receive damage within a radius of 20% of the map size (the team fight zone). Fights are ended if no corresponding attack events occur for 5 seconds. The total damage dealt or received within the team fight must surpass a threshold of 2000 to further reduce false positives. After all team fights have been counted, each player is assigned the percentage of team fights they were involved in as the corresponding attribute.

6.4 Support Items

The right items are key to *support roles*. However, many items that are useful for support players also have their uses for other roles. Therefore, based on available game guides we manually compiled a list of items that are exclusively used by support players: *Courier*, *Flying Courier*, *Observer Ward*, and *Sentry Ward*. Item purchases are not directly reflected in the available data from the replay files. Therefore, we periodically check the inventory of each player and increment a counter for support items if a new item from our list is found. The resulting count is finally assigned to a corresponding attribute for each player.

6.5 Damage Types

We determined the following damage categories based on player roles and requirements of other attributes: damage to heroes, damage to towers, damage to neutral creeps, and damage to regular creeps. We extract this information from the replay files by utilizing a categorized database of all units in the game and comparing identifiers of attacker and victim for each damage event.

7 Classification Results and Discussion

Based on the attribute selection described in section 6, several different classifiers were trained and evaluated using 10-fold cross-validation on our data set. The choice of candidate classifiers was based on existing works and complemented by commonly used classification approaches. It included: *Logistic regression* (LR), *random forest decision trees* (RF), *support vector machines* with *sequential minimal optimization* (SMO), *naive Bayes* classifiers (NB), and *Bayesian networks*. The WEKA library and tools provided the technical platform. The classifiers were evaluated according to several established performance metrics (accuracy, mean absolute error (MAE) [18], and area under ROC (AUC) [8]) and we also analyzed the confusion matrix. All performance metrics were calculated by using the WEKA default implementations, which are described in the reference documents [19], with optimized parameters.

Table 2 lists the accuracies and MAEs of all classification approaches, as well as the weighted AUC averages. Table 3 presents the confusion matrix of the LR classifier. With accuracies of around 75%, MAEs around 0.08 and AUC values around 0.95 the results are not perfect but quite promising with respect to the complex classification task and the limited data set. Our analysis revealed neuralgic points that might be good starting points to improve the results in

Classifier	**Accuracy**	**Mean Absolute Error**	**Wgt. Avg. AUC**
Full set of classes			
Random Forest	76.27%	0.0905	0.943
Logistic Regression	75.85%	0.0826	0.947
SMO	75.28%	0.1753	0.926
Bayesian Networks	72.03%	0.0801	0.933
NaiveBayes	70.76%	0.0769	0.933
Reduced set of classes			
Bayesian Networks	96.58%	0.0322	0.995
SMO	96.15%	0.2308	0.975
Logistic Regression	96.15%	0.0381	0.993
Naive Bayes	95.58%	0.0383	0.994
Random Forest	91.17%	0.1162	0.985

Table 2. Summary of 10-fold cross-validation accuracies, mean absolute errors and weighted averages of the AUC for the full set and for a reduced set of classes

a	**b**	**c**	**d**	**e**	**f**	**g**	**h**	**i**	**classified as**
163	6	13	11	1	0	2	3	0	a = Carry - Active
12	**101**	0	0	0	1	9	0	0	b = Carry - Farming
27	3	**28**	0	3	0	0	0	0	c = Ganker
10	0	0	**113**	5	0	0	5	0	d = Support - Babysitter
1	0	2	8	**57**	0	0	3	0	e = Support - Roaming
5	0	0	0	2	**8**	1	0	0	f = Support - Farming
7	14	0	1	0	0	**30**	0	0	g = Pusher
6	0	3	7	0	0	0	**31**	0	h = Feeder
0	0	0	0	0	0	0	0	**6**	i = Inactive

Table 3. Confusion Matrix for logistic regression using 10-fold cross-validation.

future work. For further analysis we limit ourselves to the LR case as the overall most stable and best performing classifier of our selection. Taking a closer look at the confusion matrix (table 3) reveals that there are two frequent cases of misclassification: *Active carry* versus *ganker* and farming carry versus *pusher*. In order to gain better insight we manually analyzed a number of problematic games and players and we could observe that in these games the *early gank* detection could be problematic because of the unclear transition between game phases (esp. early to mid game). As mentioned in section 6 we empirically determined a fixed time threshold for early game detection that – on average – worked well but not for exceptional cases that were present in some games; e.g. players leaving their lanes either extremely early or extremely late. A second factor that we noticed originated from roles looking very similar even to the human eye, although for apparently different reasons. For example, the *farming carry* wants as much gold and experience as possible and the *pusher* on the other hand wants to destroy towers as fast as possible but these goals in many cases can be achieved by the same actions. The *active carry* and the *ganker* are both involved in many player versus player fights. As we based our initial role definitions on established information sources for the game, our results highlight that some of the accepted roles do in fact bear overlaps because even though they define a certain player behavior, the distinction in some cases seems to be made solely by the players' intentions, which cannot be directly observed. A third factor that might play an important role is what we call performance noise in cases where players are not able to demonstrate a certain role clear enough. Although the special roles *feeder* and *inactive* were added to reduce performance effects, borderline cases may not be detected. A fourth factor we identified are dynamic role changes that are demanded by the situation or enforced by the competing team. This is problematic because our attributes and classification are currently based on large time spans or even the whole game. A solution might be detecting roles for smaller time spans within the game, however, this would also necessarily reduce the amount of game data available due to the shorter time span and dividing the game into meaningful phases is in itself a very challenging task.

We conducted a small study on the issue of ambiguous roles, asking three *Dota 2* experts to manually classify ten players in one of the problematic games. The responses were highly divergent with up to four different labels being provided for some players. This illustrates that the classification task is difficult, even for human experts, which is an interesting insight for the MOBA community and game designers. We classified the professional tournament data set (203 players) to look deeper into the possible influence of performance noise and achieved an accuracy of 81% percent. Although based on a limited data set, this indicates that performance might indeed be an issue. Yet, limiting the original data set to only the players of the winning teams did not result in any notable differences in classification performance (accuracy 75.62%, MAE 0.0842, weighted AUC 0.936). This suggests that individual performance noise might be an issue but not the team performance as a whole. We also classified our data set with the same attributes but with a reduced set of classes (*carry*, *support*, and *solo lane*) inspired by Gao et al. [5] to compare the power of our attributes and to assess the influence of the number of classes. Before classification with the same attributes as before we relabeled our data according to the following rules: all types of *carries*, *gankers*, and *pusher* were relabeled as *carry* and all types of *supports* were relabeled as just support. Entries labeled as *inactive* were completely removed before training and classification (cf. [5]). Players labeled as *feeders* were manually looked up and relabeled by a human expert to the best fitting class. The results of a ten-fold cross-validation are also presented in table 2. Again, LR proved to be among the best performing and most stable classifiers. However, for the reduced set of classes, Bayesian networks also performed well. While the results overall improved compared to the full set of classes, there are still some residual errors. Compared to the results of Gao et al. [5], 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 results 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 game experts, some are ambiguous even to humans manually labeling the data. Furthermore, we identified the reliable distinction of game phases as a major challenge. Additionally, looking at global game roles per match has proven to have limits, due to dynamic roles switching within shorter time spans. Lastly, performance noise exists, although of the mentioned factors, this one is comparatively well controlled by our attributes (in accordance with the design goals, as differences between winners and losers or even compared to professionals were not significantly large).

8 Conclusion and Future Work

We presented and discussed an approach to apply machine learning techniques for the classification of player roles in the MOBA game *Dota 2*. Since most MOBA games share many key game mechanics, our approach should be applicable to other games of the same genre or even to similar team-based games of other genres with slight modifications. Investigating a larger set of attributes, we isolated a manageable set and employed that reduced set to estimate the applicability of a range of classifiers according to established performance metrics. While the classification accuracy for the whole set of classes is limited, with approx. 75%, it is still promising and for our data set logistic regression was clearly the overall most stable and best performing classifier. Classification for a reduced set of classes was very successful with an accuracy of 96%, which is already suitable for many applications. Again, LR – although not the best performing classifier – proved to be very stable and well suited to this domain.

Looking at the limitations of our results highlights several important challenges. First, the definition of classes seems to be primarily intention-defined rather than behavior-defined for some classes, which makes them very difficult to detect. Still, our approach could be useful as a pre-processing step for a tool that allows game designers, players, or casters to look at games and player roles, e.g, by highlighting problematic games. Second, the distinction of specific games phases is an important issue that is not trivial to solve. It affects the classification of certain roles that are characterized by behavior related to certain phases in the game and further complications arise from the fact that roles may change during a game. Third, performance noise is a factor, e.g., if players are not able to act out their intended role. However, we presented indications that our attributes are insensitive to performance up to a certain degree by comparing the results of limiting classification just to the winning team or just professional players. 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.

References

1. Bartle, R.: Hearts, clubs, diamonds, spades: Players who suit muds. Journal of MUD research 1(1), 19 (1996)
2. Drachen, A., Sifa, R., Bauckhage, C., Thurau, C.: Guns, swords and data: Clustering of player behavior in computer games in the wild. In: CIG. pp. 163–170. IEEE (2012)
3. Drachen, A., Yancey, M., Maguire, J., et al.: Skill-Based Differences in Spatio-Temporal Team Behaviour in Defence of The Ancients 2. In: Proc. of IEEE Games, Entertainment, and Media (GEM). IEEE (2014)
4. Fu, T.S., Chen, H.T., Chou, C.L., Tsai, W.J., Lee, S.Y.: Screen-strategy analysis in broadcast basketball video using player tracking. In: Yang, J.F., Hang, H.M., Tanimoto, M., Chen, T. (eds.) VCIP. pp. 1–4. IEEE (2011)
5. Gao, L., Judd, J., Wong, D., Lowder, J.: Classifying dota 2 hero characters based on play style and performance. In: Univ. of Utah Course on ML (2013)
6. Hall, M.A.: Correlation-based Feature Subset Selection for Machine Learning. Ph.D. thesis, University of Waikato, Hamilton, New Zealand (1998)
7. Hall, M., Frank, E., Holmes, G., Pfahringer, B., Reutemann, P., Witten, I.H.: The weka data mining software: An update. SIGKDD Explor. Newsl. 11(1), 10–18 (2009)
8. Huang, J., Ling, C.X.: Using auc and accuracy in evaluating learning algorithms. IEEE Transactions on Knowledge and Data Engineering 17, 299–310 (2005)
9. Kohavi, R., John, G.H.: Wrappers for feature subset selection. Artificial Intelligence 97(1-2), 273–324 (1997), special issue on relevance
10. Li, F., Woodham, R.J.: Analysis of player actions in selected hockey game situations. In: CRV. pp. 152–159. IEEE (2005)
11. Liu, S., Ballinger, C., Louis, S.: Player identification from rts game replays. In: 28th Int. Conf. on Computers and Their Applications (CATA) (2013)
12. Nuangjumnonga, T., Mitomo, H.: Leadership development through online gaming. In: 19th ITS Biennial Conference. ITS, Bangkok (2012)
13. Pobiedina, N., Neidhardt, J., del Carmen Calatrava Moreno, M., Werthner, H.: Ranking factors of team success. In: WWW (Companion Volume). pp. 1185–1194. WWW Conf. Steering Committee, ACM (2013)
14. Statista: Video games revenue worldwide from 2012 to 2015. `http://statista.com/statistics/278181/`, access: 2015-05-08
15. Synnaeve, G., Bessire, P.: A Bayesian Model for Plan Recognition in RTS Games Applied to StarCraft. In: AIIDE. AAAI Press (2011)
16. Tovinkere, V., Qian, R.J.: Detecting Semantic Events in Soccer Games: Towards A Complete Solution. In: Proc. of IEEE Int. Conf. on Multim. & Expo (2001)
17. Weber, B.G., Mateas, M.: A data mining approach to strategy prediction. In: CIG. pp. 140–147. IEEE, Piscataway, USA (2009)
18. Willmott, C.J., Matsuura, K.: Advantages of the mean absolute error (MAE) over the root mean square error (RMSE) in assessing average model performance. Climate Research 30, 79–82 (2005)
19. Witten, I.H., Frank, E., Hall, M.A.: Data Mining: Practical Machine Learning Tools and Techniques. Morgan Kaufmann Publ. Inc., SF, CA, USA, 3rd edn. (2011)
20. Yang, P., Harrison, B., Roberts, D.L.: Identifying patterns in combat that are predictive of success in moba games. Proc. of Foundations of Digital Games (2014)
21. Zhu, G., Huang, Q., Xu, C., Rui, Y., Jiang, S., Gao, W., Yao, H.: Trajectory Based Event Tactics Analysis in Broadcast Sports Video. In: Proc. of the 15th Int. Conf. on Multimedia. pp. 58–67. ACM, New York, USA (2007)