Proceedings of the Federated Conference on Computer Science and Information Systems pp. 193–196
DOI: 10.15439/2018F362
ISSN 2300-5963 ACSIS, Vol. 15

Predicting winrate of Hearthstone decks using their archetypes

Jan Betley zweryfikujfirme.pl
Email: jan.betley@zweryfikujfirme.pl
Anna Sztyber Warsaw University of Technology
Email: sztyber.anna@mchtr.pw.edu.pl
Adam Witkowski University of Warsaw
Email: adam.witkowski@mimuw.edu.pl

***Abstract*—This paper describes our solution for the AAIA'18 Data Mining Challenge: Predicting Win-rates of Hearthstone Decks. Train and test decks were clustered by DBSCAN algorithm with precomputed distance matrix dependent on the number of common cards. We observed that each cluster can be represented by an archetype deck - one of popular decks used by human players. For each deck we created features describing cards quality and types. Additionally we used differences of these features with respect to archetype decks. Finally we used XGBoost to build a model predicting outcome of a game played between two decks.**

I. Introduction

Hearthstone is the most popular online collectible card video game[1]. Despite simple rules, game requires high strategic skills, of two separate types:

- Ability to create a high quality deck (set of cards)
- Ability to use given deck as well as possible

Article describes our solution to AAIA'18 Data Mining Challenge, where teams were asked to predict winrates of different decks, played by different bots. Full data about deck composition was easily available, while bots strategies could be only guessed from training games history. Because of that, of the two factors (deck variability and player variability), corresponding to the two skills mentioned, we decided to focus on the first, using bot type only as a control variable dividing training/testing data into 16 subsets (4 bots playing against each other).

In Hearthstone, every deck has a strategy - a way to beat the opponent. The three main strategy types are aggro, control and combo. An aggro deck tries to kill the opponent as fast as possible. A control deck tries to destroy minions played by the opponent and finish the game with strong, high-cost minons. A combo deck uses a special combination of cards that work very well together to gain great advantage or even kill the opponent in one turn.

[1]While describing Hearthstone mechanics is clearly out of the scope of this work, some basic rules are enough to make it understandable. There are two players, each of them starts with deck of 30 cards and a 'hero' card. They play alternating rounds, in each round player may play any number of cards, limited by their costs and the resource called 'mana'. Cards have different types, most important being spells and minions. Played cards more or less directly contribute to dealing damage to the other player, and the only goal is to be the first player to deal 30 damage.

Although there is virtually infinite set of possible hearthstone decks, only few of them are good enough to be played on at least semi-professional level. One of the most popular webpages with hearthstone statistics, https://hsreplay.net/decks/, currently defines 48 deck "archetypes" (exact number varies in time), such as "Aggro Hunter" or "Cube Warlock". Archetypes are based on existence of certain key cards, cleverly matched to other key cards. Those connections build deck strength, and with some of those cards missing deck would become unplayable.

Archetypes usually define the only one correct strategy. Without going too much into detail, strategy is about maximizing played card value by playing them in the right moment. E.g. strategy for Aggro Hunter is to deal as much damage as possible as fast as possible, while Cube Warlock defends until he has enough mana to play very strong cards cheaply. Each strategy has a counter strategy that might be more or less available to the opponent, so most decks - even top quality ones - do much better against some certain decks, and much worse against other.

II. Problem statement

The goal of the competition was to predict winrates (percentage of games won) of 200 test decks played by bots[2]. There were 4 different bots (denoted by $A1, A2, B1, B2$). A priori nothing was known about the bots, in particular it was not known what algorithms were used by the bots (a bot could just use a set of simple heuristics to play the game, or it can be a deep neural network, like AlphaGo) The training set consisted of results of 299680 games played between 400 training decks. For each game, we had a tuple (bot1, deck1, bot2, deck2) and the result of the game. This tuple denoted that bot1 played deck1 against bot2 using deck2.

The winrates that we had to predict were calculated based on a large number of games played between test decks and the training decks. The score was calculated as RMSE between the actual winrate and the predicted winrate for each bot, deck pair (so there were 800 numbers to predict).

[2]bot is a computer program that plays a game, here Hearthstone

III. Bot winrates

While we knew nothing about specific bot strategies[3], they were certainly different. Overall winrates for each bot are presented in Table I.

TABLE I
Bot winrates - overall, as first/second player, vs others

bot	overall	1st	2nd	vs A1	vs A2	vs B1	vs B2
A1	0.45	0.51	0.39	0.50	0.44	0.38	0.40
A2	0.52	0.57	0.47	0.56	0.50	0.45	0.43
B1	0.54	0.58	0.50	0.62	0.55	0.50	0.46
B2	0.56	0.61	0.51	0.60	0.57	0.54	0.50

It is obvious that every reasonable predicting model must include information about bots playing, and - when predicting single game result - also about starting deck/bot. Having stated that, later in this article we won't be explicitly referring to bots and starting positions - they are present in every model, but our solution is based entirely on differences between decks.

IV. Deck archetypes via clustering

A natural question is: are the given decks just random collections of available cards, or are they created from some archetypes? We used clustering to answer this question. We defined a distance between decks as

$$d = \frac{30 - n_c}{30} \tag{1}$$

where n_c is the number of common cards (counting with repetitions) in both decks.

Then we used DBSCAN [1] algorithm from scikit-learn library [2] with this metric. The parameters of the algorithm were eps = 0.4 and min_samples = 5. The algorithm found 11 clusters:

- 2 different clusters for Paladin and Warlock heroes;
- 1 cluster for each other hero

There were also 25 decks that did not have any cluster assigned. Each cluster had 47 or 48 decks, except for the Hunter cluster which had 96 decks.

This result looked promising — if the decks were random, because of the big number of possible cards, there would be no meaningful clusters.

A. Deck archetypes

For each cluster we calculated the most frequent cards used in the decks from this cluster. In each cluster there were from 5 to 10 cards that appeared in almost every deck of the cluster. For example, in one of the Paladin clusters cards Vilefin Inquisitor, Murloc Tidecaller, Bluegill Warrior, Grimscale Chum, Murloc Warleader and Rockpool Hunter that are all murloc minions were among top 10 most frequent cards (see Table II). This fact combined with the knowledge that there exists a popular Murloc Paladin deck allows to easily classify decks from this cluster as being of this archetype.

Based on the most frequent cards and domain knowledge (one of the authors used to play a lot of Hearthstone) we determined the archetype of the decks in every cluster. The archetypes were: dead man's hand warrior, inner fire priest, jade shaman, aggro hunter, jade druid, zoo warlock, cube warlock, tempo rogue, secret mage, murloc paladin, and dude paladin.

[3]Competition data included full training games courses, so those strategies could be somehow extracted, but we did not use them

TABLE II
Top 10 most frequent cards in the Murloc Paladin cluster.

card name	frequency
Vilefin Inquisitor	97.92 %
Murloc Tidecaller	95.83 %
Righteous Protector	95.83 %
Bluegill Warrior	93.75 %
Corridor Creeper	93.75 %
Grimscale Chum	93.75 %
Call to Arms	91.67 %
Murloc Warleader	91.67 %
Rockpool Hunter	91.67 %
Unidentified Maul	91.67 %

B. Model decks

We had decks clustered and we knew their archetypes. We were interested in how much the given decks differ from decks of those archetypes played by professional human players. Ideally, we would prefer to compare provided decks with 'optimal' decklists but there is no such thing as a "perfect" deck — optimal decklist depends on the decks played by the opponents. For example, there is a card Golakka Crawler that is very effective against decks that use pirates. If the decks with pirates are popular, then the decks with the Golakka Crawler will be more successful than those without it. On the other hand, if no one uses pirates, then the card is useless. [4]

For each cluster (except of 'other') we chose one model deck from the website Tempostorm (https://tempostorm.com/hearthstone/meta-snapshot/standard/2018-01-08) that creates reports about popular/strong decks. One problem with this approach was that the used decklists change every week and we did not know the date from which we should take the decklists. We used the report from the beginning of January 2018, based on the following observations:

1) the decks contained cards from the Kobolds & Catacombs expansion, released in December 2017;
2) the decks did not have any cards from The Witchwood expansion, released in April 2018; and
3) high percentage of the decks contained cards Corridor Creeper (46%) and Patches the Pirate (28%) which were changed in February 2018 and lost a lot of popularity as an effect [5]

In Table IV (column average distance) we give the average distance of decks from each cluster to the model decks. The distance function is the same as the one used in the clustering.

Note the huge discrepancies in the average distances between clusters. This can be simply a matter of the prepared decks: maybe the Shaman decks were generated differently

[4]For human players it is therefore very important to know "the meta" — that is, which decks are strong and which are popular.

[5]changing a card to a weaker version is called a nerf in the Hearthstone terminology.

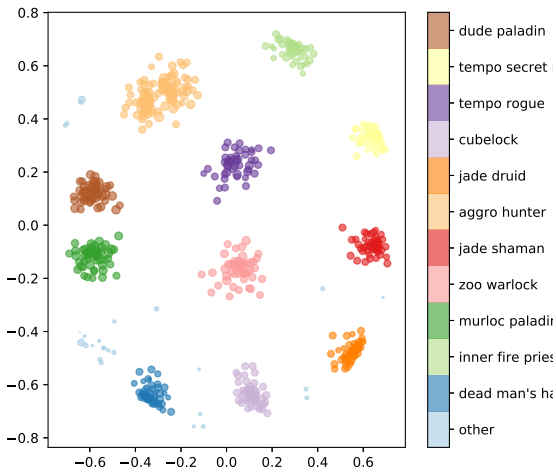

Fig. 1. 2-D decks embedding based on distance matrix using t-SNE ($perplexity = 2$, $angle = 0.9$)

than the Aggro Paladin ones. Of course this can be also a matter of poor choice of the model decks for some archetypes. We did not try to find other model decks.

To better visualize the clusters, we created a 2-D embedding using the t-SNE algorithm [4] with precomputed distance matrix. The clusters are visualised in Figure 1, Decks are coloured according to clusters found by DBSCAN. The size of dots is proportional to win-rates (with 50% substituted for test decks). In the figure we can see that clusters are clearly visible with except of the 'other' cluster. The only one unexpected fact is the division of the Hunter cluster into two groups. The number of decks in "aggro hunter" cluster is roughly two times greater than the number of decks in other clusters. Therefore it is possible that this cluster should be further divided into two clusters. However, neither DBSCAN nor multidimensional scaling (MDS) visualization confirm this division.

C. Simple cluster-based model

With established clusters, we tried the simplest possible model: for each deck D, predict the average winrate of the training decks from D's cluster as its winrate. For example, if in the training set there are 3 "murloc paladin" decks with winrates $65\%, 56\%$ and 55% respectively, each murloc paladin deck from test set gets winrate $\frac{(55+56+65)}{3} = 58.66\%$. Averaged winrates per clusters are shown in Table IV (column winrates). We can observe significant differences between clusters and poor performance of decks classified as "other". This model ignored any differences between decks in the same cluster, so it could not achieve a good score.

Another approach we tried was to use the metric given by Equation (1) directly, take the 10 training decks closest to the test deck and predict the average winrate of those 10 decks, but this gave much worse results. We decided to use more standard predictive models, improving them with features based on the clusters and model decks.

V. Predicting the winrates

A. Basic deck features

For each deck we generated a number of features that tried to capture the 'goodness' of the deck. Those included e.g.

- Average card cost
- Number of cards with cost of 0/1/2/3 or more
- Number of free/common/rare/epic/legendary cards[6]
- Number of neutral/single-hero cards[7]
- Number of minions/spells/weapons/other cards[8]
- Average overall card winrates[9]
- Number of special cards such as murlocs, beasts, minons with divine shield, taunt minions, etc.

We also gathered data from few webpages with Hearthstone statistics:

- "Card value" - overall card value when playing in arena mode[10] [http://www.heartharena.com/tierlist]
- "Card played winrate" - chance of winning the game, under condition that given card was played [https://hsreplay.net/cards]
- 2821 most popular decks [https://hsreplay.net/decks]

First and second datasets were averaged into deck "mean card value" and "mean card winrate" features. Most popular decks were used to estimate "how well cards in deck are connected". For each card pair we calculated:

- How often they appear in external decks
- How often they appear together in external decks
- "Card pair connection strength" as quotient of the above values

Deck feature "card connection strength" was calculated as a mean of connection strength between all card pairs in deck.

B. Differences with the model decks

In addition to deck-only based features, we created a set of features describing how different the deck is from the model deck of the same archetype, such as:

- General distance (as described in 1),
- How many 1-mana cards were added/removed
- Difference between added/removed card's arena value

This way we wanted to approximate how big is the "real" impact of the differences between decks and their archetypes. E.g. if deck uses many cheap minions, replacing some of them with other cheap minions is a small difference, while replacing them with expensive cards would be a drastic change.

[6]Every Hearthstone card fits into one of those categories, they approximately describe card strength

[7]Neutral card may be played by any hero, in the contrast to cards that may be played only by specific hero

[8]Every Hearthstone card has its type, "minions", "spells" and "weapons" are the most popular

[9]Taken from https://hsreplay.net/cards/

[10]Arena mode is a specific hearthstone variant, played with more or less random decks

TABLE III
FEATURES IMPORTANCES

feature	f-score
deck2 winrate	1198
mean card value p2	470
mean card value p1	442
mean minion health p2	436
mean card winrate p2	409
mean minion attack p2	387
who plays first	366
mean minion attack p1	363
mean minion health p1	352
mean card winrate p1	349
diff mean minion health p2	337
diff mean minion attack p2	334
card connection strength p1	328
diff arena value p2	327
diff arena value p1	320
card connection strength p2	320
mean minion cost p1	315
diff mean minion health p1	314
diff mean minion attack p1	308
mean minion cost p2	289

C. Final model

We considered two approaches for generating final predictions:

(a) train a regression model predicting winrate of each deck,
(b) train a classification model predicting result (win, lose) of a game between the deck and a particular opponent.

We tested both approaches and we decided on (b) due to larger training set available (300K training games versus 400 train decks) and more promising preliminary results. Since the test decks were evaluated based on the games against training decks, we added the feature "opponent's winrate" which was by far the most important one (see Table III).

Classification model was trained using XGBoost library [3], which is an implementation of gradient boosting algorithm. After training we predicted results of 4 million random games between train and test decks. Final winrates were averages of these games results.

Using XGBoost model we analysed features importances given by f-score, which is a measure of how often given variable was used to split node of a decision tree. Table III shows twenty most important features with respect to f-score. Each deck feature was repeated for both players, p1 and p2 denote player1 and player2 respectively. Features with diff prefix are differences between deck and its archetype.

VI. RESULT PER CLUSTER

After preparing the final model we tested how did the model work on particular clusters. Since we did not have the ground truth, the test was done taking 100 random training decks as the validation decks and training the model on the games played with the remaining 300 decks. We then calculated for each cluster the mean absolute error for the validation decks. The results are shown in Table IV (column mae).

We expected that we will not do so well on the 'other' cluster, since for those decks we did not have the model decks. One possible explanation is that the 'other' decks were quite weak (27% average winrate) and quite different from the other decks and therefore easier to predict.

TABLE IV
RESULTS FOR EACH CLUSTER SEPARATELY: AVERAGE WINRATE PER CLUSTER, AVERAGE DISTANCE FROM EACH CLUSTER TO THE MODEL DECK AND MEAN ABSOLUTE ERROR

cluster	winrate	average distance	mae
dude paladin	0.6482	5.33	3.90
murloc paladin	0.6141	11.98	3.13
zoo warlock	0.5871	6.46	5.08
aggro hunter	0.5860	8.13	5.54
cubelock	0.5074	5.83	4.70
tempo rogue	0.5013	6.62	5.79
jade shaman	0.4680	14.51	3.53
jade druid	0.4510	9.28	3.93
dead man's hand warrior	0.4174	12.04	5.24
tempo secret mage	0.4084	8.85	3.38
inner fire priest	0.3888	11.56	5.78
other	0.2651	-	3.01

VII. CONCLUSIONS

Our idea was to explore if the decks were generated randomly or followed the pattern of human players decks. Clustering revealed existence of groups. Each group can be represented by an archetype deck selected from decks of successful human players. Our final solution was generated by XGBoost model using features describing differences between each deck and its archetype. These distance features improved model results significantly with comparison to the model using only basic features. Our model achieved the RMSE score 6.349 which gave us 10th place in the competition.

The solution could be further improved by:

- building ensemble of different models,
- exploring other clustering algorithms,
- XGBoost hyper-parameter tuning.

Final result of our model evaluated on all test decks was slightly worse than on a fraction of test decks available for early evaluation of submissions, which can be caused by overfitting of the solution to the test data available. It would probably be beneficial to leave part of training decks for validation and comparison of different solutions.

REFERENCES

[1] M. Ester, H.-P. Kriegel, J. Sander, and X. Xu, "A density-based algorithm for discovering clusters a density-based algorithm for discovering clusters in large spatial databases with noise," in *Proceedings of the Second International Conference on Knowledge Discovery and Data Mining*, ser. KDD'96. AAAI Press, 1996, pp. 226–231. [Online]. Available: http://dl.acm.org/citation.cfm?id=3001460.3001507

[2] F. Pedregosa, G. Varoquaux, A. Gramfort, V. Michel, B. Thirion, O. Grisel, M. Blondel, P. Prettenhofer, R. Weiss, V. Dubourg, J. Vanderplas, A. Passos, D. Cournapeau, M. Brucher, M. Perrot, and E. Duchesnay, "Scikit-learn: Machine learning in Python," *Journal of Machine Learning Research*, vol. 12, pp. 2825–2830, 2011.

[3] T. Chen and C. Guestrin, "XGBoost: A scalable tree boosting system," in *Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, ser. KDD '16. New York, NY, USA: ACM, 2016. doi: 10.1145/2939672.2939785. ISBN 978-1-4503-4232-2 pp. 785–794. [Online]. Available: http://doi.acm.org/10.1145/2939672.2939785

[4] L. van der Maaten and G. Hinton, "Visualizing data using t-SNE," *Journal of Machine Learning Research*, vol. 9, pp. 2579–2605, 2008. [Online]. Available: http://www.jmlr.org/papers/v9/vandermaaten08a.html