

Clash Royale Challenge: How to Select Training Decks for Win-rate Prediction

Andrzej Janusz^{*†}, Łukasz Grad^{*†}, Marek Grzegorowski^{*†}

^{*}Institute of Informatics, University of Warsaw, Poland

[†]Esensei Sp. z o.o., Poland

Contact Email: janusza@mimuw.edu.pl

Abstract—We summarize the sixth data mining competition organized at the Knowledge Pit platform in association with the Federated Conference on Computer Science and Information Systems series, titled Clash Royale Challenge: How to Select Training Decks for Win-rate Prediction. We outline the scope of this challenge and briefly present its results. We also discuss the problem of acquiring knowledge about new notions from video games through an active learning cycle. We explain how this task is related to the problem considered in the challenge and share results of experiments that we conducted to demonstrate usefulness of the active learning approach in practice.

Keywords—Data Mining Contest; Training Subset Selection; Win-rates Prediction; Active Learning; Clash Royale

I. INTRODUCTION

Video games, and especially mobile games, are considered as one of the domains in which a huge amount of data is generated by players on a daily basis. Utilization of such data in practical applications requires complex analysis towards a proper understanding of hidden concepts and time-consuming data preparation process. In particular, it is often necessary to provide labels of data records that we want to use for model training. Even though it is very laborious, this process is necessary to train intelligent models that could provide value to end-users. Due to limited time and budget, it is usually possible to label only a small amount of data. The “as-is” market standard is to manually label data records. To handle this, a number of corporations utilize crowd-computing services to outsource data-labeling capability. However, it seems that the labeling process could be optimized using approaches related to active learning (AL) [1]. An alternative way could be, so-called weak supervision, where less reliable labels are generated using simple heuristics using domain knowledge [2].

In this research, we use as an example a popular mobile collectible card video game – Clash Royale – which combines elements of collectible card game and tower defense genres (<https://clashroyale.com/>). In this game, players build decks consisting of 8 cards that represent playable troops, buildings, and spells, which they use to attack opponent’s towers and defend against their cards. Using good decks is one of the critical abilities of successful Clash Royale players. We describe a challenge in which we take on a problem of measuring and predicting the deck effectiveness in 1v1 ladder games. In particular, we would like to find out whether it is possible to train an efficient win-rate prediction model on a relatively

small subset of decks, whose win-rates were estimated in the past. Such a task can also be considered in the context of active learning, as a selection of a data batch that should be labeled and used for training a win-rate prediction model.

The remaining of the paper is organized as follows: In Section II, we discuss a context for the competition, i.e. the problem of active learning from video game data. In Section III, we briefly describe the competition and summarize its results. In Section IV, we present a framework for predicting win-rates of Clash Royale decks. In Section V, we conclude the paper and draw some directions for future research.

II. ACTIVE LEARNING FROM VIDEO GAME DATA

Active learning is a domain within the field of machine learning, in which the learning algorithm can interactively query an oracle about labels (or more generally, target attribute values) of some limited number of training records [3]. Its applications are particularly suitable when the availability of labeled data is limited. In such cases, to train reliable prediction models, it is often necessary to perform a laborious and costly process of manual data labeling. Through the use of AL, it is possible to facilitate this process by allowing the algorithm to choose records which seem the most beneficial for learning [4]. Such a selection of training examples is performed based on results of a model constructed in a previous iteration of the AL cycle (Figure 1). The importance of unlabeled examples is determined by the confidence of their classification or by the expected model change after including the instances to the training data [5]. To deal with the cold-start problem, the first training batch is typically selected at random or by using some clustering technique to find a diverse yet representative set of initial examples for labeling [6]. Then, in subsequent iterations of the AL cycle, additional examples are selected and the prediction model is continuously improved [7].

In practice, the data is usually labeled by a committee of experts and the oracle is implemented as a voting system. Since humans are prone to errors, several experts assign labels to each data record, and the final labeling is determined by voting [1]. Research in the AL field focus mainly on algorithms for selecting a single data record for labeling in each iteration of the AL cycle. However, when there are many available experts, it is more efficient to choose larger batches. In this way, experts who label faster do not have to wait until others finish their tasks and the prediction model is updated.

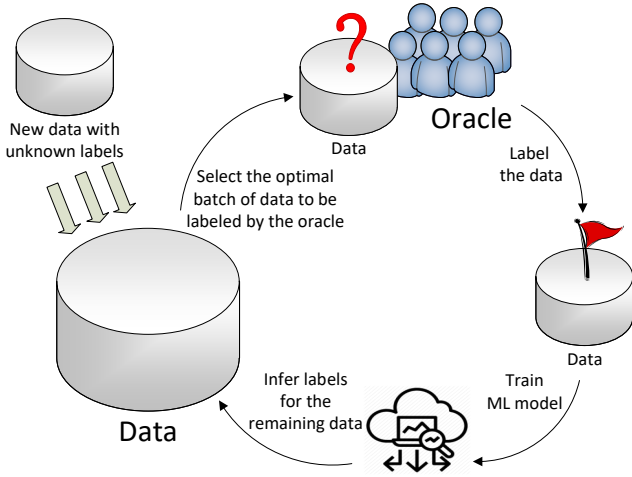


Fig. 1. An active learning cycle. The oracle is interactively queried about labels of records which are selected as the most beneficial for learning by the algorithm.

In a context of video game data, the role of experts can be assumed by the community of players. As a consequence, the committee which assigns labels can be quite large and diverse. This fact impacts the active learning setup in two main aspects:

- 1) In order to optimize the efficiency of the AL cycle and avoid lags in the labeling process, algorithms need to select many data records for labeling at a time.
- 2) Each of selected records should be shown to a subset of available labelers. The voting algorithm should take into account the diversity of labelers and remain robust, even in a presence of a large number of noisy labels.

The system governing the AL cycle should be able to guarantee that whenever there is an available labeler, it can provide a new example for labeling. Moreover, the oracle should be able to find a consensus among contradicting assignments of labels and be able to discard those whose quality is likely to be low. This can be done through the estimation of labelers' expertise, combined with a weighted voting schema [8].

III. CLASH ROYALE CHALLENGE

The task in Clash Royale Challenge was related to the first of the two problems mentioned in Section II, namely, the selection of a data subset that allows to construct an efficient model for predicting win-rates of Clash Royale decks. The competition took place between April 24, 2019 and June 12, 2019, under the auspices of 14th Federated Conference on Computer Science and Information Systems. It was organized on the KnowledgePit platform which underwent a significant lift-up shortly before the start of the challenge (<https://knowledgepit.ml/clash-royale-challenge/>).

The competition's task could also be viewed as a continuation of the topic started in the previous year, i.e. the prediction of win-rates of decks from collectible card video games [9]. The ability to assess quality of decks in a continuously

evolving game is one of core features of an advisory system for players, called SENSEI, which is being developed by one of the competition's sponsors [10].

Data in this challenge consisted of 100.000 Clash Royale decks that were most commonly used by players during three consecutive league seasons in 1v1 ladder games. They were provided in a tabular format. Each row the training set corresponded to a Clash Royale deck and was described by four columns. The first one listed eight cards that constitute the deck. The second and third column showed the number of games played with the deck, and the number of players that were using it, respectively. These values were computed based on over 160.000.000 game results obtained using the RoyaleAPI service (<https://royaleapi.com/>) and SENSEI's data acquisition module. The last column indicated estimations of win-rates of the decks, that were calculated based on games played in the given time window. Participants were asked to indicate ten subsets of those decks, with sizes fixed to 600, 700, ..., 1500. These subsets were ought to allow training efficient support vector regression models (SVR) with radial kernels [11] for a purpose of win-rate prediction (one model for each training data subset). Competitors could also tune hyper-parameters of the models.

A. Evaluation of results and participation in the challenge

The quality of solutions was assessed by measuring the prediction performance of the models trained on data subsets indicated by the participants. This evaluation step was conducted on a separate set of decks. This test data consisted of decks that were popular during the three game seasons after the training data period. This set was not revealed to participants before the end of the challenge. However, a small subset of decks from the test period (a validation data set) was given to participants. It is also worth noticing that the same decks could appear in both the training and evaluation data, but they were likely to have different win-rates. The cause of those differences is the fact that the game evolves in time, players adapt to new strategies, and the balance of individual cards (and their popularity) changes from one season to another.

During the competition, submitted solutions were evaluated online, and the preliminary results were published on Leaderboard. The preliminary score was computed on a randomly selected set of 2000 test records, fixed for all participants. The final evaluation was performed after completion of the competition using the remaining part of the test data. Each teams was oblige to submit a report describing their approach before the end of the challenge.

The measure chosen for the assessment of solutions was the R-squared. If we denote a prediction for a test instance i as f_i , and its reference win-rate as y_i , the R-squared metric is:

$$R^2 = 1 - \frac{RSS}{TSS} \quad (1)$$

where RSS and TSS are the residual and total sum of squares, respectively:

$$RSS = \sum_i (y_i - f_i)^2, \quad TSS = \sum_i (y_i - \bar{y})^2$$

TABLE I. FINAL R-SQUARED VALUES AND NUMBER OF SUBMISSIONS FROM TOP-RANKED TEAMS. THE LAST ROW SHOWS THE RESULT OBTAINED BY THE BASELINE SOLUTION.

team name	rank	number of submissions	final result
Dymitr	1	144	0.2552
amy	2	123	0.2530
ru	3	25	0.2257
ms	4	51	0.2241
~	5	30	0.2215
...
baseline	14	1	0.1564

and $\bar{y} = \frac{1}{N} \sum_i y_i$.

A value of this metric was computed independently for predictions made by SVR models trained on each of the subsets indicated in the submitted solutions. The final score was an average of the obtained results.

B. Summary of the competition results

The scores obtained by top-ranked teams are presented in Table I. The baseline in this challenge was obtained using a simple algorithm that utilizes basic properties of the SVR model, i.e., only records which correspond to the support vectors have any impact on the model. A ν -regression SVR was trained on a subset of the most popular training decks with the parameter values set such that the number of selected support vectors corresponded the the desired sizes of target sets. These vectors were taken as the baseline solution.

Participants of the challenge were able to significantly improve over the baseline score. Unfortunately, no team from the top 10 was using an approach that could be applied to the considered problem in practice. The winners were using a greedy search heuristic to limit the candidate decks. Then, they fine-tuned the final sets using exhaustive search. In both cases, the quality of fit was computed as the R-squared value obtained on the validation data. In practice, such data would not be available. Thus any supervised search heuristic would not be feasible. More detailed description of the winning approach can be found in [12]. In Section IV, we propose an alternative method which solves the competition problem without a need for a validation sample. It uses an approach inspired by active learning and can be utilized in a way similar to the AL cycle to continuously adapt to a changing game.

IV. ESTIMATION OF WIN-RATES USING LIMITED DATA

We approach the problem of win-rate estimation in Clash Royale using limited training data from a pool-based active learning perspective. Specifically, we propose a solution based on density weighted batch uncertainty sampling. For uncertainty sampling, we provide an informativeness function tailored to the case of known, but noisy labels. Such an approach is viable in the context of win-rate prediction because they change in time due to balance changes in the game. Even though we can always estimate win-rates using historical data (e.g. data from a previous game season), such estimates are likely to be invalid for new game seasons.

A. An informativeness measure

Formally, given a training data set T consisting of records $(x_i, y_i)_{i=1}^N$ with known label noise $\text{Var}[y_i] = \sigma_i^2$ and a model M , we search for a training subset of given size K , such that the model trained on this subset achieves the lowest generalization error, i.e.:

$$A^* = \underset{A: |A|=K}{\operatorname{argmin}} \mathbb{E}_{(X,Y)} [l(Y, f_A^M(X))] \quad (2)$$

where f_A^M is the mapping induced by the model M trained on subset A and l is the mean squared error loss function.

In our method, we begin by choosing an initial training subset A_0 of size m at random. Then, at each step, we greedily select a sample that maximizes the importance:

$$x^* = \underset{x:T}{\operatorname{argmax}} [\phi(x)^\alpha \times \text{Sim}(x)^\beta \times \text{Dis}(x)^\gamma] \quad (3)$$

where ϕ measures the informativeness of samples, $\text{Sim}(x) = (\frac{1}{u} \sum_{i=1}^u \text{sim}(x, x_i))$ is a measure of a representativeness, and $\text{Dis}(x) = (\frac{1}{b} \sum_{i=1}^b \text{dis}(x, x_i^B))$ measures the dissimilarity in the current batch, assuming that we have already chosen samples (x_0^B, \dots, x_b^B) . Parameters α, β, γ control the relative importance of each factor. In this work, we set each of the parameters to 1. Similarity measure used in all our experiments was the Jaccard index: $\text{sim}(x_1, x_2) = \frac{|x_1 \cap x_2|}{|x_1 \cup x_2|}$ and $\text{dis}(x_1, x_2) = 1 - \text{sim}(x_1, x_2)$.

To derive the measure of informativeness ϕ , we assume normality of the response Y . Given a trained model M and the posterior distribution of the response $g_M(X_i) \sim \mathcal{N}(\mu_i, \tau_i^2)$, we obtain the posterior predictive distribution $\hat{Y}_i \sim \mathcal{N}(\mu_i, \tau_i^2 + \sigma_i^2)$. This is valid, since noise process and the posterior distribution are independent Gaussians. Now, given a sample (x_i, y_i) we define the informativeness as

$$\begin{aligned} \phi(x_i) &= 1 - \mathbb{P}(|\hat{Y}_i - \mu_i| > |d_i|) \\ &= 1 - \mathbb{P}((\hat{Y}_i - \mu_i) > |d_i|) - \mathbb{P}((\hat{Y}_i - \mu_i) < -|d_i|) \\ &= 1 - \mathbb{P}\left(Z_i > \frac{|d_i|}{\sqrt{\sigma_i^2 + \tau_i^2}}\right) - \mathbb{P}\left(Z_i < \frac{-|d_i|}{\sqrt{\sigma_i^2 + \tau_i^2}}\right) \\ &= 1 - (1 - \Phi(|d_i|)) - \Phi(-|d_i|) \\ &= 1 - 2\Phi(-|d_i|) \end{aligned}$$

where $d_i = y_i - \mu_i$, Z_i denotes a standard normal variable and Φ is a standard normal CDF.

In our experiments, we used Gaussian Process Regression model [13], along with absolute exponential covariance kernel:

$$K_{\kappa^2, \lambda}(x, x^*) = \kappa^2 \exp\left(-\frac{|x - x^*|}{\lambda}\right) \quad (4)$$

where κ^2 and λ are kernel hyper-parameters optimized during model fitting.

B. Experimental results

We compared the results obtained using our approach to the best solution from the winners of the challenge [12]. Instead of computing the R-squared metric, we simply measured the root mean squared error (RMSE) of SVR models trained on

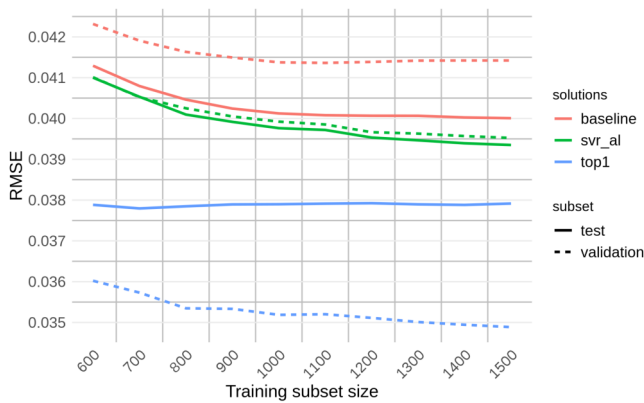


Fig. 2. Results of the compared training subset selection methods. The dotted lines indicate RMSE values obtained on the validation data, whereas the solid lines correspond to the final test set.

each data subset from the winner's solution, and on subsets of corresponding sizes found using our method. We checked errors of the trained models on the final test set from the challenge, as well as on the validation set which was fully available to competing teams during the competition. Figure 2 shows those results. To provide a better reference, we also computed RMSE values achieved by the baseline method.

Even though our method achieved lower scores than the winner's for all subset sizes, it is important to notice that in practice, when the validation set is not available, it would be much more useful. For the largest subset size the difference between is lower than 0.002, which seems negligible considering the variance of predicted win-rates. Furthermore, the AL-inspired method is always better than the baseline. The plot also shows that our method is not over-fitted to any particular data subset, whereas the winners achieved much better results on the validation set than on the final test set. Interestingly, RMSE of the winning solution on the final set does not decrease with the growing size of training data subset (it is even slightly higher). This could be regarded as another argument in favour of our method.

V. SUMMARY

In this paper, we described Clash Royale Challenge organized at the KnowledgePit platform, whose scope was on finding an optimal data subset for training win-rate prediction models. Our competition attracted 115 teams from 18 countries. Among the participating teams, 68 submitted at least one solution file which was ranked on the public Leaderboard. More than 40 of those teams decided to disclose their approach by uploading short reports.

A dominating approach used by competitors was based on a greedy search heuristic with a fit function that used an additional validation set. As an alternative, we proposed a method inspired by active learning, which is more suitable to solve the considered problem in practical applications. In the presented experiments, we showed that it can be effective. In the future, we will focus on extending our approach in the

context of large batch sampling, e.g. by effectively utilizing a predictive covariance matrix for computing the informativeness function utilized by our method.

ACKNOWLEDGMENTS

This research was co-funded by Smart Growth Operational Programme 2014-2020, financed by European Regional Development Fund under GameINN project POIR.01.02.00-00-0184/17, operated by National Centre for Research and Development in Poland.

REFERENCES

- [1] S. Yan, K. Chaudhuri, and T. Javidi, "Active learning from imperfect labelers," in *Proceedings of the 30th International Conference on Neural Information Processing Systems*, ser. NIPS'16. USA: Curran Associates Inc., 2016, pp. 2136–2144. [Online]. Available: <http://dl.acm.org/citation.cfm?id=3157096.3157335>
- [2] S. H. Bach, D. Rodriguez, Y. Liu, C. Luo, H. Shao, C. Xia, S. Sen, A. Ratner, B. Hancock, H. Alborzi, R. Kuchhal, C. Ré, and R. Malkin, "Snorkel drybell: A case study in deploying weak supervision at industrial scale," in *SIGMOD Conference*. ACM, 2019, pp. 362–375.
- [3] B. Settles, *Active Learning*. Morgan & Claypool Publishers, 2012.
- [4] E. Lughofer, "Hybrid active learning for reducing the annotation effort of operators in classification systems," *Pattern Recogn.*, vol. 45, no. 2, pp. 884–896, Feb. 2012. [Online]. Available: <http://dx.doi.org/10.1016/j.patcog.2011.08.009>
- [5] W. Cai, Y. Zhang, Y. Zhang, S. Zhou, W. Wang, Z. Chen, and C. Ding, "Active learning for classification with maximum model change," *ACM Trans. Inf. Syst.*, vol. 36, no. 2, pp. 15:1–15:28, Aug. 2017. [Online]. Available: <http://doi.acm.org/10.1145/3086820>
- [6] H. T. Nguyen and A. Smeulders, "Active learning using pre-clustering," in *Proceedings of the Twenty-first International Conference on Machine Learning*, ser. ICML '04. New York, NY, USA: ACM, 2004, pp. 79–. [Online]. Available: <http://doi.acm.org/10.1145/1015330.1015349>
- [7] K. Konyushkova, R. Sznitman, and P. Fua, "Learning active learning from real and synthetic data," *CoRR*, vol. abs/1703.03365, 2017. [Online]. Available: <http://arxiv.org/abs/1703.03365>
- [8] C. Zhang and K. Chaudhuri, "Active learning from weak and strong labelers," in *Proceedings of the 28th International Conference on Neural Information Processing Systems - Volume 1*, ser. NIPS'15. Cambridge, MA, USA: MIT Press, 2015, pp. 703–711. [Online]. Available: <http://dl.acm.org/citation.cfm?id=2969239.2969318>
- [9] A. Janusz, T. Tajmayer, M. Świechowski, Ł. Grad, J. Puczniewski, and D. Ślęzak, "Toward an intelligent HS deck advisor: Lessons learned from aaia'18 data mining competition," in *Proceedings of the 2018 Federated Conference on Computer Science and Information Systems, FedCSIS 2018, Poznań, Poland, September 9-12, 2018.*, M. Ganzha, L. A. Maciaszek, and M. Paprzycki, Eds., 2018, pp. 189–192. [Online]. Available: <https://doi.org/10.15439/2018F386>
- [10] A. Janusz, D. Ślęzak, S. Stawicki, and K. Stencel, "SENSEI: an intelligent advisory system for the esports community and casual players," in *2018 IEEE/WIC/ACM International Conference on Web Intelligence, WI 2018, Santiago, Chile, December 3-6, 2018*. IEEE Computer Society, 2018, pp. 754–757. [Online]. Available: <https://doi.org/10.1109/WI.2018.00010>
- [11] A. J. Smola and B. Schölkopf, "A Tutorial on Support Vector Regression," *Statistics and Computing*, vol. 14, no. 3, pp. 199–222, 2004.
- [12] D. Ruta, L. Cen, and Q. H. Vu, "Greedy incremental support vector regression," in *Proceedings of the 2019 Federated Conference on Computer Science and Information Systems, FedCSIS 2019, Leipzig, Germany, September 1-4, 2019.*, M. Ganzha, L. A. Maciaszek, and M. Paprzycki, Eds., 2019.
- [13] C. E. Rasmussen and C. K. I. Williams, *Gaussian Processes for Machine Learning*. MIT Press, 2006.