The effective interaction of the models of classification with the usage of auction heuristics

V B Barakhnin, O Yu Kozhemyakina and I S Pastushkov

Federal Research Center for Information and Computational Technologies, Lavrentiev av. 6, Novosibirsk, 630090, Russian Federation

e-mail: bar@ict.nsc.ru, olgakozhemyakina@mail.ru, pas2shkov.ilya@gmail.com

Abstract. The article proposes the approach that allows to make multi-model systems of classification systems more suitable for solving the practical problems in the field of information technologies. To achieve this, it was necessary to provide a stable system response time to support SLA (service-level agreement) and, on the other hand, to minimize the downtime of server hardware. The first requirement would allow to give the requests correctly and quickly to the client, and the second one would make the cost of renting servers more justified. The main idea of the proposed approach is to obtain the pre-dictions with the highest possible accuracy in a fixe time. We considered three auction models: Dutch auction (rate decreasing), English auction (rate increasing) and the adapted version of Vickrey auction (highest rate switched off). In all cases, we used the highest probability of the mark as a rate, and we set a time parameter, after which the prediction is recognized as final. The obtained results and their comparison with the methods of ensembling and balancing allow us to conclude that the proposed approaches can be use-ful in the development of multi-model systems of classification.

1. Introduction

1.1. A problem statement

There are many approaches which provide the multi-model interaction, also called ensemble methods, among them are: boosting [1] – the consecutive learning of the same models where each model learns on mistakes of previous model; stacking [2] – the usage of meta-model which operates the data of initial model outputs as an input; voting [3] where the decision depends from the majority predictions; etc. However, each one of them has the essential disadvantage: we should wait for all of models responses to obtain the result. It is worth noting that scikit-learn module [4] for Python has consecutive model execution although it with the R language is in fact the industry standard. However, even in case of any kind of model isolation or if we emulate the parallel computing, we have the growth of hardware requirements and, on the other hand, the ensemble response time will not be less than response time of slowest model. In case of stacking, we should add the meta-model response time too. The fact that even in the parallel ensemble models work as one consecutive model which performs one request in the moment is also important. The article proposes the approach that allows to speed up sufficiently the work of ensemble of models with acceptable quality loss.
1.1.1. Service-level agreement. One of base requirements for multi-model interaction in IT, and in this work in particular, is the existence of service-level agreement or SLA [5] as we will call it in further. This agreement is done between the backend development and backend users for the purpose to provide the fast content transfer and to perform the different scenarios when for some reason the backend is unable to give response to the users and to process the various scenarios when, for one reason or another (the user accesses the models not directly, but via the API, web or mobile app), the backend cannot respond in the time determined by the application's operating conditions.

The server's response speed is also directly proportional to the number of requests which it can process, what reduces downtime and hardware rental costs. This paper presents the results for the most common balancing methods used in the IT industry. In them, we evaluate the effectiveness of methods based on two criteria: the response time and the quality. It should be clarified that in this case, the quality is the probability of a response, which many classification models, excluding the SVM support vector method, return by default.

In our task of classifying of the poetic texts by genre and style, there are no time limits, however, for the stable operation of the interface [1], we must have a priori set agreement about the server response time for this tool. So, on the one hand, we guarantee a response in a user-friendly time and the correct processing of the response by the interface, on the other hand, we can enter the more complex models into the set without reducing the overall response time for each user: some people are willing to wait to get an accurate answer, while others are satisfied with a less accurate one, but ad hoc. Moreover, we can give the user the ability to choose among which models to set the auction for their requests, or to exclude from the set of models those models that were least used for their requests.

1.1.2. First free choice. In case of default of heuristic the request to the set of models is going to the first free of them, if there are more free models it goes to random chosen one [6]. All the experiments mentioned here were performed on set contains three base models: the logistic regression, the multi-layer perceptron implementations from scikit learn [4] and XGBoost [7] classifier from separate module on 20 newsgroups dataset [8]. This dataset contains about 20 000 text which are partitioned almost on 20 different newsgroups. On Figure 1 the quality and response time distribution histograms are represented.

![Figure 1](image)

**Figure 1.** Plots of distribution of response time and quality on x axis for first free.

1.1.3. Round-robin. Round-robin [6] is rather simple heuristic on which the each follow request to server sets to next in prior declared queue. After the queue ends, the last one request goes to first etc. This
approach is widely using in load balancing, in particular on DNS-servers. The quality and response time
distribution histograms are represented on Figure 2.

![Histograms of response time and quality distribution](image)

**Figure 2.** Plots of distribution of response time and quality on x axis for round-robin.

2. **Auction heuristics**

Next three approaches are based on the original method of usage of auction performing algorithms [9] as the ensembling methods which is proposed by I.S. Pastushkov. Notably, the e-auctions were widely used in cloud computing [10], but there are several sufficient differences in these three approaches. The first one proposes the two-factor optimization between quality and time usage, only the best price is choosing in cloud computing. The second one works with almost constant models and the cloud-computing providers can change their bids sufficiently. At last, the proposed approaches are easier and there are no information of their usage in text classification or in any other ML task.

We can consider our task as one of scheduling tasks, in particular, as a single-stage task with directive deadlines on multiple machines [11]. It means that we also have a flow of tasks that have to be performed on any machine. As far as every task is performed only once, it’s a single stage task, so the delays between the stages of one task haven’t been taken into account. The cost function of such tasks is:

\[ L_{max} = \max_{i=1,n} (c_i - d_i) \rightarrow \min \]

Where \(c_i\) is the time of task performing and \(d_i\) is the directive deadline which is equal to SLA multiplied by the number of tasks. The cost function means that we minimize the SLA contradictions for average task.

The tasks with directive deadlines are solved by building a matrix of work completion times and of the selection of the best variant. However, we do not have a priori knowledge of the execution time and we must also consider the quality. We use the quality as the machine's priority within the framework of schedule theory. An important note is that if a set of machines consists of machines with similar quality, then we always get the best result on the available machines. So, if there are three machines in the set with a quality of 90, 91, 92%, we will actually get the result from the machine with the best quality, but provided that there are no other requests to the cluster and this machine is free. But in this case, there is no point in the other machines in the set. The auction heuristics solve the problem when the quite high cluster load and allow to follow the SLA.

2.1. **Classic of English auction**

There is the classic auction model with price that is increasing with each bid from the first one and becoming the final in case if there are no increasing during predefined time range.

The confidence in the prediction (the quality of the response of each model) is considered as a price, while there is no apriori initial price – this is the degree of confidence of the fastest model – it is assumed that the speed of the model is inversely proportional to its degree of confidence (quality). We suppose that response time is proportional with quality (easy models is faster than complex, but have worse
quality) [9]. In the course of experiments with the considered models, this was confirmed empirically. The interaction model is the following: if after the predictions of one of the models the none of the following models has not responded, the prediction of that model is final, otherwise we expect the time from the next prediction.

In Figure 3, which shows the time and quality distributions, you can see that the response time of the described heuristic is much faster than that of a single model, and the quality is more evenly distributed in the direction of improvement.

![Figure 3. Plots of distribution of response time and quality on x axis for English auction](image_url)

2.2. Vickrey auction or second-price auction

Vickrey auction is the anonymous bid collecting and the final price is the second price on descending order [12]. In real auctions the winner is one who suggest the highest price. In the case of interaction between models the Vickrey auction is implemented as follows:

1. The preliminary calculation for the first n iterations works similarly to the English auction, with the only difference that after the end of the work, the predictions of the remaining models are read regardless of the time of their operation.
2. The data on prices (i.e. the quality of the predictions) and response time of each model are stored.
3. Based on the average of the last m quality predictions, the estimated price for each is calculated, and the model with the highest price is eliminated at this iteration.
4. Figure 4 shows that despite the obvious gain in time, the quality losses are unacceptable.

![Figure 4. Plots of distribution of response time and quality on x axis for Vickrey auction](image_url)
2.3. Dutch auction

In our task, max price is 1, e.g. 100% quality, a minimal one is some predefined threshold, in experiments we have use 0.6. Max time is max from precomputed response times; a minimal is minimum from them. Pre-calculation is performing a part of dataset it way as the Vickrey auction. Supposing the fact that max quality correspond the max time, we reach minimal time during price (quality) decreasing.

The Dutch auction, originally used for the sale of tulips, is arranged as follows: the price decreases over time and stops when someone agrees with the price [13]. The maximum price in our case is 100% quality, that is, the degree of confidence is equal to 1, the minimum is a certain a priori set threshold, in our case 0.6. The maximum time is taken as the maximum from the time of preparation of applications by models, the minimum time is determined similarly. The preliminary count is a small part of the training sample that is processed in the same way as step 1 of the Vickrey auction implementation. Based on the assumption that the maximum quality corresponds to the maximum time, in the process of reducing the price (prediction quality), we approach the minimum time.

Figure 5 shows that we have similar situation with Vickrey auction: faster but worse.

![Figure 5](image_url)

Figure 5. Plots of distribution of response time and quality on x axis for Dutch auction

3. Experiments results

3.1. Results on 20 newsgroups dataset

Tables 1 and 2 represent median (e.g. 50th percentile), 80th, 90th and 95th percentile and also the standard deviation for all interaction models mentioned in article. As the baseline, we use the fastest model from set.

| Method            | 50th | 80th | 90th | 95th | Std. deviation |
|-------------------|------|------|------|------|----------------|
| Baseline          | 0.36 | 0.44 | 0.47 | 0.49 | 0.08           |
| First free        | 0.2  | 0.31 | 0.41 | 0.46 | 0.14           |
| Round-robin       | 0.08 | 0.21 | 0.28 | 0.37 | 0.11           |
| English auction   | 0.2  | 0.3  | 0.35 | 0.38 | 0.11           |
| Vickrey auction   | 0.07 | 0.21 | 0.25 | 0.29 | 0.1            |
| Dutch auction     | 0.1  | 0.19 | 0.24 | 0.26 | 0.08           |

As represented in tables 1 and 2, Vickrey and Dutch auctions are faster than baseline with almost the same quality. The English auction model has the time advantage from round-robin and comparable time with first free, so the quality of an English auction model is same with round-robin and better than first free.
Table 2. Table represents experiment quality data percentiles.

| Method          | 50th  | 80th  | 90th  | 95th  | Std. deviation |
|-----------------|-------|-------|-------|-------|----------------|
| Baseline        | 0.81  | 0.82  | 0.83  | 0.84  | 0.02           |
| First free      | 0.81  | 0.85  | 0.89  | 0.9   | 0.08           |
| Round-robin     | 0.78  | 0.84  | 0.88  | 0.9   | 0.1            |
| English auction | 0.81  | 0.85  | 0.88  | 0.9   | 0.07           |
| Vickrey auction | 0.76  | 0.81  | 0.82  | 0.83  | 0.09           |
| Dutch auction   | 0.75  | 0.81  | 0.82  | 0.83  | 0.09           |

3.2. The results on poetic texts data with comparison

In [14] we have already presented several approaches which are adapted for poetry texts classification task with its problems among the class imbalance and small corpus size. Both of these problems were successfully solved by using different data generation methods. Previously, we did not have a multi-user performance task and SLA, so the time optimization was not a top priority. First of all, as noted earlier, all methods of ensembling require the time at least equal to the slowest representative of the ensemble. This means that the usage of ensemble methods as a separate classifier is pointless, since it does not change the response time or quality in any way. The best quality, for example, of stacking, is offset by the fact that the other algorithms will be too fast. It is the same for time gain, since the stacking will not get the requests due to the slow response speed, so it makes no sense to apply any detailed measurements. It is enough to mention that classifiers with voting are 2–3 times slower and stacking is 4–5 times slower. We conducted the experiment on a similar [14] corpus of A. S. Pushkin's early lyrics for a more correct comparison. In order to model a highly loaded system with a large number of requests per second, we used the augmentation. This approach is widely used in image processing to extend the corpus with copies of images with minor changes such as resizing or cropping. For our text data, almost the same thing was done to make the random differences with each other, for greater clarity of comparison. We randomly deleted a line from a poem in an incoming request. Table 3 contains the data for the experiments.

Table 3. Table represents experiment quality data percentiles for poetry corpus.

| Method          | 50th  | 80th  | 90th  | 95th  | Std. deviation |
|-----------------|-------|-------|-------|-------|----------------|
| Baseline        | 0.76  | 0.79  | 0.8   | 0.8   | 0.03           |
| First free      | 0.75  | 0.83  | 0.87  | 0.87  | 0.09           |
| Round-robin     | 0.73  | 0.83  | 0.88  | 0.88  | 0.12           |
| English auction | 0.76  | 0.83  | 0.87  | 0.87  | 0.09           |
| Vickrey auction | 0.71  | 0.81  | 0.85  | 0.85  | 0.11           |
| Dutch auction   | 0.7   | 0.79  | 0.83  | 0.83  | 0.1            |

As we see in table 3, the proposed approach with no special modifications could be applied for task of poetry classification analysis and could be included in interface of system [15] as were declared above.

4. Conclusion

We plan also apply these approaches to extract any other text characteristics. The suggested method shall provide the stable work for user applications without sufficient quality loss.

5. Acknowledgments

The work was supported by Russian Science Foundation grant No. 19-18-00466.
References

[1] Freund Y, Schapire R and Abe N 1999 A short introduction to boosting Journal of Japanese Society for Artificial Intelligence 14 (5) pp 771–780
[2] Wolpert D H 1992 Stacked generalization Neural networks 5 (2) pp 241–259
[3] Breiman L 1999 Pasting small votes for classification in large databases and on-line Machine Learning 36 (1–2) pp 85–103
[4] Scikit-learn Machine Learning in Python https://scikit-learn.org
[5] Kearney K T and Torelli F 2011 The SLA model Service Level Agreements for Cloud Computing (New York: Springer) pp 43–67
[6] Silberschatz A, Galvin P B and Gagne G 2006 (John Wiley & Sons)
[7] Chen T and Guestrin C 2016 Xgboost: A scalable tree boosting system Proc. of the 22nd ACM Sigkdd Int. Conf. on Knowledge Discovery and Data Mining pp 785–794
[8] Home Page for 20 Newsgroups Data Set. http://qwone.com/~jason/20Newsgroups/
[9] McAfée R P and McMillan J 1987 Auctions and bidding Journal of economic literature 25 (2) pp 699–738
[10] Chard K, Caton S, Rana O and Bubendorfer K 2010 Social cloud: Cloud computing in social networks 2010 IEEE 3rd International Conference on Cloud Computing pp 99–106
[11] Tanaev V S, Gordon V S and Shafransky Ya M 1984 Schedule theory, single-stage systems (Springer)
[12] Vickrey W 1961 Counterspeculation, auctions, and competitive sealed tenders The Journal of finance 16 (1) pp 8–37
[13] Katok E and Kwasnica A M 2008 Time is money: The effect of clock speed on seller’s revenue in Dutch auctions Experimental Economics 11 (4) pp 344–357
[14] Barakhnin V B, Kozhemyakina O Yu and Pastushkov I S 2017 Comparative analysis of methods of automated classification of poetic texts based on lexical signs CEUR Workshop Proceedings vol 2022 pp 252–257 (in Russ.)
[15] Barakhnin V B, Kozhemyakina O Yu and Borzilova Yu S 2019 Designing of poetry texts complex analysis representation’s informational system Novosibirsk State University bulletin. Series: Informational technologies 17 (1) pp 5–17 (in Russ.)