Methods for Predict Memory Usage of Big Data Computing System: A Comparison Study

Xiaofan Wu\textsuperscript{1,2,3}, Ying Li\textsuperscript{1,2,3}, Zhijian Cheng\textsuperscript{1} and Yuan Wang\textsuperscript{2,3}

\textsuperscript{1} College of Computer Science, Zhejiang University, No. 38 Zheda Road, Hangzhou, China
\textsuperscript{2} Engineering Research Center of Augmented Reality and Intelligence Interaction of Zhejiang Province, No. 399 Wangshang Road, Hangzhou, China
\textsuperscript{3} Hangzhou Institute of Netease, No. 399 Wangshang Road, Hangzhou, China
Email: hzwuxiaofan@corp.netease.com

Abstract. Memory is a very important resource in big-data computing system. Reasonable allocation of memory is actually meaningful. But sometimes it is hard to achieve it. Mammoth Data Platform is a big data computing platform developed by NetEase. When we take a look at the jobs running on the data platform, we can find that the memory requested by the users will actually be much larger than the memory used on Mammoth Data Platform, which is not a good phenomenon. So this paper tries some machine learning methods to predict the actual memory usage so that we can allocate the memory reasonably. After trying these methods, we get a good prediction for memory used by the jobs on the platform ($R^2=0.95$), which will improve the memory utilization of the data platform.

1. Introduction
In recent years, big data computing system has play important role in many aspects such as personal recommendation, smart city, etc. Improving resource utilization is critical to achieving efficient and stable operation of big data computing systems. In past, resource such CPU, memory, network band width, were allocated based user applications. Arbitrary application of resource caused serious waste, reduced the resource utilization.

Mammoth Bigdata Platform is a big data computing platform developed by NetEase. Users can submit jobs on the platform to complete the corresponding calculation and data analysis tasks. The Mammoth platform allocated memory resource based on user application. The memory utilizations of more than 90% jobs are lower than 50% according our obvious. On the other hand, data analysis tasks may fail if insufficient memory resource was allocated.

It is important to predict the usage of data analysis tasks in order to ensure allocate sufficient resource and avoid wasting. In this paper, various machine learning methods to predict the actual memory usage and compared. The accuracy of these methods is tested based on real data which logged from Mammoth platform. We obtain a better fitting effect ($R^2=0.95$), which improves the memory utilization of the data platform.

2. Data Preprocessing
The data used in this paper is provided by NetEase. There are more than 50000 data in our dataset. Each piece of data has multiple attributes related to our task, including id, application id, application name, user name, start time, finish time, tracking url, job type, severity (aggregate severity of all the
heuristics), score (the application score which is the sum of heuristic scores), scheduler, resource_used, resource_wasted, total delay, insert time, query execute email and so on.

Above all, we need to calculate memory used and memory wasted of the tasks. Because in the dataset, resource_used means the resources used by the job in MB Seconds and resource_wasted means the resources wasted by the job in MB Seconds. So we get the running time of the task by the start time and finish time of the tasks. Then we get memory_used and memory_wasted of the tasks by resource_used, resource_wasted and running time. What’s more, for the quantitative variables in the dataset, we standardized them.

After analyzing the distribution of data, we can see that only about 5% of the memory_used are bigger than 1000000. As a matter of fact, the memory prediction task is a regression task. People always use mean square error as loss function in regression task. Mean square error is sensitive to outliers. So we remove this part of data.

Finally, we try to choose explanatory variables from our dataset. One thing we must know is that although some variables may be highly correlated with the final result we want to predict, these attributes are data that can only be obtained after the task actually runs, such as the running time of the task, total delay of the task, etc. So these attributes cannot be selected as explanatory variables into our model. In this paper, we select the following attributes as explanatory variables by drawing scatterplots. In the end, we select memory_wasted, score, severity, user_name and queue as explanatory variables. Figures 1-5 are the scatterplots:

Figure 1. memory_used and memory_wasted.

Figure 2. memory_used and score.

Figure 3. memory_used and severity.

Figure 4. memory_used and user_name.
variables randomly in a building two algorithms. We then select variables from L2 and Lasso regression as examples. This paper adopts the method of adding dummy variables in statistics to deal with, that is, if there are 4 types of qualitative variables, three dummy variables are added, using (1,0,0), (0,1,0), (0,0,1) and (0,0,0) to represent 4 different categories [1].

3. Methods
In this paper, we use some classic machine learning algorithms to predict memory usage, such as linear regression, ridge regression, Lasso, KNN, decision tree, random forest, bagging, etc. In addition, we also try the neural network method to make predictions. So in this part, we will give a brief introduction to these algorithms.

In statistics, linear regression is the most commonly used regression model, and it has a good effect on many problems [2]. However, linear regression sometimes has the problem of multicollinearity, which will lead to the bad performance of the model. Therefore, statisticians proposed ridge regression and Lasso regression to improve linear regression. Ridge regression can be understood as adding an L2 regular term based on the loss function of linear regression and Lasso regression can be understood as adding an L1 regular term based on the loss function of linear regression [3, 4]. By determining the value of $\lambda$, the model can be better than Linear regression:

$$\text{Loss}_\text{Ridge} = \text{Loss}_\text{Linear} + \lambda \Sigma \omega_j^2 \quad (1)$$

$$\text{Loss}_\text{Lasso} = \text{Loss}_\text{Linear} + \lambda \Sigma ||\omega_j||_1$$

KNN is a classic algorithm in the field of data mining [5]. The main idea of this algorithm is to use the nearest k points to represent this point. This algorithm was first widely used in classification jobs. We can get better results by adjusting parameter k. The only difference from the classification algorithm is that the regression algorithm is to use the average value of the nearest k neighbor points to predict the value of the point.

Decision tree algorithm is an efficient algorithm in practical application [6]. Decision tree algorithm can give good predictions for many practical problems. Unlike the classification tree, regression tree will always use the mean square error as the criterion to select the cut variable and cut point.

Both random forest and bagging method can be seen as an improvement of decision tree [7, 8]. The two algorithms are roughly similar. Both of them are trying to obtain better prediction results by building multiple decision trees. The only difference is that when the bagging method is used to build a tree, all variables are taken into account, while the random forest considers each split point, and randomly selects m variables from all p variables, which are used by the split points. Predictor variables can only be selected from these m variables. Therefore, the random forest is actually a
further improvement of the bagging method. Because the trees built by random forest method have smaller variance.

Neural network method is a very popular method in recent years. In order to get a better prediction, we also try to use neural network method to do the prediction. By studying related work before, we find that some similar problems are using fully connected neural network. In this paper, we also use the same approach. Neural network method performs very well on many problems. By adjusting the parameters, we can get a good model to solve our problem.

4. Results and Analysis
In this part, we will show the performance of various methods on our dataset. This paper mainly uses scikit-learn, Numpy and Pandas in python for experiments and comparisons. We also use Pytorch to build a network with dropout layers [9]. R² and mean square error are used to compare different methods. Before the experiment, we first divided the dataset into two parts, 70% for training, 30% for testing. Table 1 shows the results of the methods mentioned above. The results in the table are the best results obtained after sufficient parameter adjustments. For the neural network method, we choose a model with two hidden layers.

| Methods        | R²   | MSE  |
|----------------|------|------|
| Linear regression | 0.723| 0.280|
| Ridge          | 0.723| 0.280|
| LASSO          | 0.720| 0.283|
| KNN            | 0.925| 0.075|
| Decision Tree  | 0.946| 0.054|
| Random forest  | 0.956| 0.044|
| Bagging        | 0.956| 0.044|
| MLP            | 0.912| 0.088|
| DNN            | 0.887| 0.114|

Table 1. Results of the methods.

We draw scatterplots of predictive values and true values to intuitively compare the fitting effects of each method. Figures 6-14 are the scatterplots:

Figure 6. Linear regression.  
Figure 7. Ridge.  
Figure 8. Lasso.  
Figure 9. KNN.  
Figure 10. Decision Tree.  
Figure 11. Random Forest.
From the table and figures above, we can see that linear model has achieved good results on the memory usage prediction problem. The performances of Ridge regression and Lasso regression are almost the same as linear regression. So we can conclude that there is no strong multicollinearity among the variables we choose. The performance of the KNN algorithm is superior to the previous simple linear model and its improvements. What’s more, decision tree algorithm as a very efficient algorithm in practical applications, performs better than previous algorithms. For the random forest algorithm and the bagging method, we choose to build 100 decision trees. And we get the best performance on this problem ($R^2=0.95$). Last but not least, for the neural network method, we tried multiple sets of parameters. We find that neural network method is sensitive to parameters. And neural network without dropout layers performs better. But both of them do not perform as well as the best traditional algorithms.

At the beginning of this experiment, we removed some data from our dataset. So we do a comparative experiment on the whole dataset. Table 2 shows the results.

| Methods       | $R^2$ |
|---------------|-------|
| Linear regression | 0.949 |
| Ridge         | 0.949 |
| LASSO         | 0.949 |
| KNN           | 0.963 |
| Decision Tree | 0.989 |
| Random forest | 0.993 |
| Bagging       | 0.992 |
| MLP           | 0.968 |
| DNN           | 0.954 |

Compared with table 1, we can find that linear regression, ridge regression and lasso regression have the largest performance differences. The performance of other methods is only slightly improved. The reason for the increase in $R^2$ is that the addition of abnormal points increases the variance of the data. And we can draw a conclusion that these methods except simple linear models are robust.

5. Conclusion

In this paper, we try some common machine learning algorithms to solve the memory prediction problem in practical applications and get good results in the end. We find that in this problem, the traditional machine learning algorithms can get better results, and the difficulty of adjusting parameters is lower, which are more suitable for solving this problem. According to the relevant indicators in statistics, the bagging algorithm and the random forest algorithm can already fit the actual memory usage well, which has great significance for practical applications. But if we want to adapt to the actual situation better, there are still some issues that need further thinking. For instance, the actual memory prediction can be slightly larger, but too small will cause problems such as insufficient memory.
References

[1] Suits D B 1957 Use of dummy variables in regression equations Journal of the American Statistical Association 52 (280) 548-551.
[2] Seber George AF and Alan J Lee 2012 Linear regression analysis Wiley Series in Probability and Statistics Vol. 329 (Wiley).
[3] Marquardt D W and Snee R D 1975 Ridge regression in practice The American Statistician 29 (1) 3-20.
[4] Tibshirani R 1996 Regression shrinkage and selection via the lasso Journal of the Royal Statistical Society: Series B (Methodological) 58 (1) 267-288.
[5] Peterson L E 2009 K-nearest neighbor Scholarpedia 4 (2) 1883.
[6] Tso G K and Yau K K 2007 Predicting electricity energy consumption: A comparison of regression analysis, decision tree and neural networks Energy 32 (9) 1761-1768.
[7] Liaw A and Wiener M 2002 Classification and regression by random forest R News 2 (3) 18-22.
[8] Breiman L 1996 Bagging predictors Machine Learning 24 (2) 123-140.
[9] Baldi P and Sadowski P J 2013 Understanding dropout Advances in Neural Information Processing Systems pp 2814-2822.