Proceedings of the Federated Conference on Computer Science and Information Systems pp. 11–14
DOI: 10.15439/2019F363
ISSN 2300-5963 ACSIS, Vol. 18

Training Subset Selection for Support Vector Regression

Cenru Liu
Ngee Ann Polytechnic, Singapore
liucenru@gmail.com

Jiahao Cen
Nanyang Polytechnic, Singapore
cenjiahao456@gmail.com

***Abstract*—As more and more data are available, training a machine learning model can be extremely intractable, especially for complex models like Support Vector Regression (SVR) training of which requires solving a large quadratic programming optimization problem. Selecting a small data subset that can effectively represent the characteristic features of training data and preserve their distribution is an efficient way to solve this problem. This paper proposes a systematic approach to select the best representative data for SVR training. The distributions of both predictor and response variables are preserved in the selected subset via a 2-layer data clustering strategy. A 2-layer step-wise greedy algorithm is introduced to select best data points for constructing a reduced training set. The proposed method has been applied for predicting deck's win rates in the Clash Royale Challenge, in which 10 subsets containing hundreds of data examples were selected from 100k for training 10 SVR models to maximize their prediction performance evaluated using R-squared metric. Our final submission having a R^2 score of 0.225682 won the 3^{rd} place among over 1200 solutions submitted by 115 teams.**

***Index Terms*—Clash Royal, Support Vector Regression (SVR), R-squared metric (R^2), Radial Basis Function kernel (RBF), k-means clustering**

I. Introduction

NOWADAYS with the growth of the Internet of Things (IoT), 2.5 quintillion bytes of data are produced every day at our current speed [1]. As 2 sides of a coin, a large amount of available data help to build complex and robust machine learning models, while data processing and model training can be rather intractable. Among all data collected, some of them are irrelevant to targets, inter-dependent, and noisy with outliers, leading to inefficient or even intractable training procedure, and more seriously, poor generalization capability.

Support Vector machine (SVM), developed at AT & T Bell Laboratories by Vladimir Vapnik and his co-workers [2], [3], [4], [5], [6], [7] based on the statistical learning theory (or VC theory) [8], [9], [10]. The SVM has shown competitive generalization over many existing machine learning models in various fields, e.g. optical character recognition (OCR), object recognition, time series prediction, etc. [6], [11], [12], [13], [14], as well as in regression, denoted as Support Vector Regression (SVR) [15], [16], [17], [18]. As we know, training a SVR model needs to solve a large quadratic programming optimization problem, which becomes computation intractable on large datasets.

To overcome this disadvantage, it is useful to identify a representative and discriminative data subset from full training data, which is the intention of the Clash Royale Challenge 2019. Clash Royale is a popular video game which combines elements of collectible card game and tower defense genres. In the game, players build decks having 8 cards representing playable troops, buildings, and spells to attack opponent's towers and defend against their cards. Wining a game is highly dependent on decks. The task of the challenge is to select small data subsets from a large training dataset, on which SVR models can be trained to predict win rates of decks.

To address this problem, a systematic approach is proposed in this paper. The major advantages of our proposed method can be summarized as follows:

1) Selecting data points on the clustered space of response variables helps to preserve response distribution, allow parallel implementation, and reduce computational cost.
2) Selecting data points from cluster centers of predictor variables can largely speedy up search procedure by removing most of training examples from the selection candidates pool, meanwhile reserving predictors' distribution and their characteristic features.
3) Although no guarantee of global optimality, the systematic approach can deterministically find near-optimal solutions.

By using our method in the challenge, 10 subsets containing only hundreds of examples were selected from 100k data points, on which 10 SVR models were trained to predict win rates of decks. The average R-squared metric of the 10 models on unknown testing data is 0.225682, wining 3^{rd} place among over 1200 solutions submitted by 115 teams.

This paper is organized as follows. The challenge is described in Section II. The details of the proposed method are presented in Section III. Section IV discusses the experiment results. Conclusions are given in Section V.

II. Clash Royale Challenge

A. Challenge task

The intention of the Clash Royale Challenge is to find a small subset from a large training dataset, on which a SVR model with Radial basis function (RBF) kernel can be efficiently trained for predicting win rates of decks. Specifically, competition participants are required to submit 10 subsets of

decks, including 600, 700, 800, 900, 1000, 1100, 1200, 1300, 1400, and 1500 decks, respectively, each of which allows training an efficient SVR based win rate prediction model, and the hyper-parameters of the SVR trained on the these subsets, i.e. ϵ, C, and γ.

B. Database

The data used in the challenge are divided into training, validation, and testing sets. The training data consist of 100k Clash Royale decks that were most commonly used by players during 3 consecutive league seasons in 1v1 ladder games. The decks in the validation and testing data were popular during the three next game seasons after the training data period. The validation dataset consists of 6k decks, which was provided to competitors for self-evaluation of their solutions, while the test set was not revealed to participants. The win rates of decks were also provided in the training and validation datasets. Since the decks in the 2 sets were collected from different game seasons, the same decks in different sets may have different win rates.

C. Solution evaluation

The quality of solutions is assessed using prediction performance measured in the R-squared metric of the models trained on the indicated subsets and the associated hyper-parameters. The R-squared metric is defined as

$$R^2 = 1 - \frac{RSS}{TSS}, \tag{1}$$

where RSS is the residual sum of squares and TSS is the total sum of squares, which can be expressed as

$$RSS = \sum_i (y_i - f_i)^2, \tag{2}$$

and

$$TSS = \sum_i (y_i - \frac{1}{N}\sum_i y_i)^2, \tag{3}$$

where y_i and f_i are the ground truth label of the i^{th} data example and its prediction, respectively, and N is the number of data records in the dataset. The score of a solution is the average R^2 metric of the 10 SVR models.

Leaderboard scores were provided in the preliminary stage of the challenge, which were calculated based on a small subset of the testing data fixed to all participants. The final scores of the 2 best solutions submitted by a competitor evaluated on the full testing set were provided at the end of the challenge.

III. Method for Subset Selection

A. Method overview

A systemic method is proposed to select a small subset of data for training an efficient SVR model, which consists of 5 parts concluded as follows, as shown in Fig. 1.

1) Dividing training data into k_y groups according to the response variable, denoted as y, e.g. win rates in the challenge.

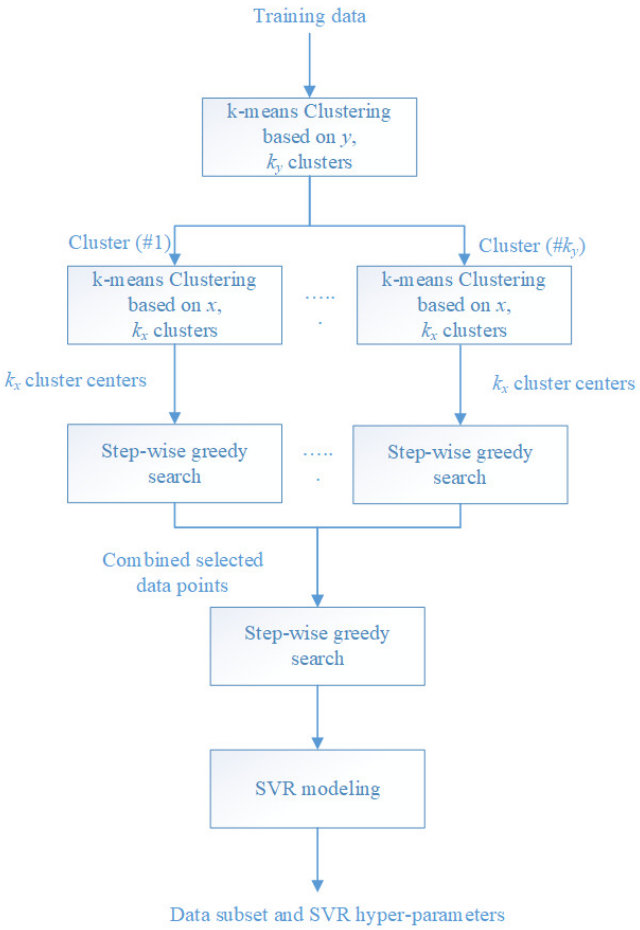

Figure 1. Flowchart of the proposed method.

2) Dividing each of k_y groups into k_x clusters according to predictor variables, denoted as x, e.g. decks in the challenge, and constructing k_y sets of cluster centers.
3) Selecting a specific number of data points individually from each of k_y center-sets by step-wise greedy search. The number is dependent on the sizes of the full dataset, center-set and the subset to be constructed, which will be discussed later. Note that the total number of selected points should be much more than the desired size of the subset.
4) Combining all points selected from the k_y center-sets and selecting exact number of points to construct the required subset by applying again the step-wise greedy algorithm.
5) Obtaining the settings of hyper-parameters (ϵ, C, and γ) for the SVM model trained on the selected subset.

B. Data representation

A data example is represented using a binary vector with a length of 90 representing 90 unique cards. Each value in the vector indicates whether or not a card is in the deck, i.e.

- 1- the associated card is used in the deck,

- 0- the associated card is not used in the deck.

The training data containing 100k examples are represented using a matrix with a dimension of 100000×90 and the validation data having 6000 examples are represented using a matrix with a dimension of 6000×90. The response variable of the training data, i.e. win rates, is represented using a vector with a length of 100000, and similarly, the win rates of the validation set are represented using a vector with a length of 6000.

It has been mentioned in Section II that the decks in training and validation sets were extracted from different game seasons. Although the same decks may exist in both sets, their win rates are likely different because the game evolves in time, players adapt to new strategies, and the balance of individual cards and their popularity changes slightly from one season to another. Removing the training examples having the same decks as the validation set but with different win rates can avoid uncertainty of such gap, which, however, cannot yield significant improvement on prediction accuracy. This indicates, from a certain of view, the robustness of our selection method.

C. Two-layer clustering analysis

Clustering analysis is firstly applied to guild data selection. Specifically, a 2-layer clustering strategy inspired by the work presented in [19] is employed to divide training data into groups, as illustrated in Fig. 1. In our method, data clustering is performed by using the K-means clustering algorithm that is a classical and popular unsupervised machine learning algorithm [20]. The aim of clustering analysis here is to preserve the distribution of the full training dataset and reflect their characteristic features in a reduced dataset.

Clustering analysis is performed independent on predictor and response variables, e.g. decks and win rates in the challenge.

1) The training dataset is firstly separated into k_y clusters according to the response variable. The value of k_y can be set empirically based on the distribution of y, e.g. $k_y = 2$ in win rate prediction. In this way, the distribution of y can be preserved, and meanwhile the subsequent steps can be implemented in parallel.
2) Each of y_k groups are then further divided into k_x clusters according to the predictor variables. The value of k_x is empirically determined according to the distribution of x as well as the sizes of training dataset and the subset to be selected.

We can finally obtain k_y groups, each having k_x cluster centers, via the 2-level clustering strategy. Similarly, the validation dataset can be divided into groups using the same cluster centers as the training data.

D. Two-layer step-wise greedy search

The data subset is selected to feed to SVR training to maximize the prediction performance of the model via a 2-layer step-wise greedy search strategy .

1) First, a specific number of data points are independently selected from each of k_y center-sets by step-wise greedy search that follows below procedure, where X denotes the full training set containing N data points, S represents the subset to be built and $R(S)$ is its R^2 score.
   - Step 1. The search procedure starts with a full training set of X and an empty subset of S.
   - Step 2. Adding the data point, denoted as p, selected from X to S, which gives the highest score among all points in X.
   - Step 3. Removing the p^{th} point from X, and $N = N - 1$.
   - Step 4. Going to Step 2 until S is fully filled.

   The score of a SVR model is the R-squared metric given in (1) calculated on the validation dataset.
   Let N_i be the number of data points selected from the i^{th} center-set, which is set as:
   $$N_i = N_{all} \times (\frac{c_{ti}}{N_t} + \frac{c_{vi}}{N_v})/2, \tag{4}$$
   for $i \in [1, 2, ..., k_y]$, where
   - N_{all} is the approximate total number of data points to be selected from all of k_y clusters, which can be empirically set to be twice as the desired size of the data subset under selection;
   - c_{ti} and c_{vi} are the sizes of the i^{th} center-sets of the training and validation sets, respectively;
   - N_t and N_v are the sizes of the full training and validation sets, respectively.
2) After the data points are selected from each of k_y center-sets, they are combined to construct a bigger set, on which the step-wise greedy search is applied again to select best data points based on the same selection criteria as the first layer of greedy search.

E. SVR hyper-parameters

The hyper-parameters of the non-linear SVR model with a Gaussian radial basis function kernel, including ϵ, C, and γ, are optimized for each selected subset using a heuristic grid search with a range around the seeds and a grid of 0.00001. The seeds of the hyper-parameters are set as follows.

1) ϵ in the ϵ-insensitive loss function controls the smoothness of the SVR model and the number of support vectors, which can largely affect model complexity and its generalization capability. ϵ is set to be an estimate of a tenth of the standard deviation using the inter-quartile range of the response variable y, expressed as:
   $$\epsilon = iqr(y)/13.49, \tag{5}$$
   where $iqr(y)$ is the inter-quartile range of y.
2) The parameter C controls the trade off between training error and model complexity, i.e. margin maximization, e.g. $C = \infty$ yielding a hard margin SVR model. In our method, C is set to be an estimate of the standard deviation of the response variable, expressed as:
   $$C = iqr(y)/1.349. \tag{6}$$

3) γ is a free parameter used in the radial kernel. The radial basis function kernel, or RBF kernel on two samples x_i and x_j is defined as

$$K(x_i, x_j) = exp(-\gamma||x_i - x_j||^2). \quad (7)$$

The value of γ is optimized by the heuristic procedure using sub-sampling [21].

IV. Experiment Results

The numbers of clusters in the 2-layer clustering analysis were set to be:

$$k_y = 2, \quad (8)$$

i.e. the data were divided into 2 clusters according to win rates, and

$$k_x = 5000, \quad (9)$$

i.e. the data in each of the 2 groups were divided into 5000 clusters. The full training dataset containing 100k examples were reduced into 10k cluster centers from 2-layer clustering analysis, among which 10 relative small subsets containing the required numbers of data examples were selected by using the 2-layer step-wise greedy search strategy.

The best solution that we submitted to the competition as the final solution has a preliminary R-squared metric of 0.2352 evaluated on a subset of testing data and a final score of 0.225682 evaluated on the full testing set, which was scored the 3^{rd} place in the challenge among over 1200 solutions submitted by 115 teams.

Although the current version of the proposed method was designed to select a best data subset for SVR model training, our method can be easily extended for other machine learning methods without many modifications. The search procedure followed in our method adding data points in a recursive way cannot guarantee global-optimal performance. Improvement can be expected with suitable implementation of global search.

V. Conclusions

It is useful to select a subset from full labeled data for efficiently training machine learning models, in order to maximize prediction performance at a small number of data examples. This cannot only reduce computational cost but also lead to better generalization capability. To address this, a systematic approach is proposed for data selection, the performance of which has been shown in the Clash Royale Challenge, in which 100k data points were reduced to 600-1500 inputted to train Support Vector Regression (SVR) based win rate prediction models, wining the 3^{rd} place in the challenge. This method, although developed for data selection in SVR training, can be easily modified for other machine learning methods. Future work will also improve the search procedure by introducing global optimization methods like evolutionary algorithms.

References

[1] B. Marr, "How Much Data Do We Create Every Day? The Mind-Blowing Stats Everyone Should Read," https://www.forbes.com/sites/bernardmarr/2018/05/21/how-much-data-do-we-create-every-day-the-mind-blowing-stats-everyone-should-read/#ae77e2560ba9, 2018.

[2] B.E. Boser, I.M. Guyon, V. Vapnik, "A training algorithm for optimal margin classifiers," *Proceedings of the Annual Conference on Computational Learning Theory, ACM*, pp. 144–152, Pittsburgh, PA 1992.

[3] I. Guyon, B. Boser, and V. Vapnik, "Automatic capacity tuning of very large VC-dimension classifiers," *Advances in Neural Information Processing Systems 5*, pp. 147–155, Morgan Kaufmann Publishers, 1993.

[4] C. Cortes, and V. Vapnik, Support vector networks, Machine Learning, vol. 20, pp. 273–297, 1995.

[5] B. Schölkopf, C. Burges, and V. Vapnik, "Extracting support data for a given task," *Proceedings of First International Conference on Knowledge Discovery and Data Mining, AAAI Press*, 1995.

[6] B. Schölkopf, C. Burges, and V. Vapnik, "Incorporating invariances in support vector learning machines," *Artificial Neural Networks, Springer Lecture Notes in Computer Science*, Vol. 1112, pp. 47–52, Berlin, 1996.

[7] V. Vapnik, S. Golowich and A. Smola, "Support Vector Method for Function Approximation, Regression Estimation, and Signal Processing," in M. Mozer, M. Jordan, and T. Petsche (eds.), Neural Information Processing Systems, vol. 9, MIT Press, Cambridge, MA., 1997.

[8] V. Vapnik and A. Chervonenkis, "Theory of Pattern Recognition" (in Russian), Nauka, 1974.

[9] V. Vapnik, "Estimation of dependences based on empirical data," Springer Verlag.

[10] V. Vapnik, "The Nature of Statistical Learning Theory," Springer, New York.

[11] B. Schölkopf, P. Simard, A. Smola, and V. Vapnik, "Prior knowledge in support vector kernels," In: M.I. Jordan, M.J. Kearns, and S.A. Solla (Eds.), Advances in Neural Information Processing Systems 10, MIT Press, Cambridge, MA, pp. 640–646, 1998.

[12] V. Blanz, B. Schölkopf, H. Bulthoff, C. Burges, V. Vapnik, and T. Vetter, "Comparison of view-based object recognition algorithms using realistic 3D models," *Artificial Neural Networks, Springer Lecture Notes in Computer Science*, vol. 1112, pp. 251–256, Berlin, 1996.

[13] B. Schölkopf, K. Sung, C. Burges, F. Girosi, P. Niyogi, T. Poggio, and V. Vapnik, "Comparing support vector machines with Gaussian kernels to radial basis function classifiers," *IEEE Transactions on Signal Processing*, vol. 45, pp. 2758–2765, 1997.

[14] K.R. Muller, A. Smola, G. Ratsch, B. Schölkopf, J. Kohlmorgen, and V. Vapnik, "Predicting time series with support vector machines," *Artificial Neural Networks, Springer Lecture Notes in Computer Science*, vol. 1327, pp. 999–1004, Berlin, 1997.

[15] H. Drucker, C.J.C. Burges, L. Kaufman, A. Smola, and V. Vapnik, "Support vector regression machines," *Advances in Neural Information Processing Systems 9*, pp. 155–161, MIT Press, Cambridge, MA, 1997.

[16] M. Stitson, A. Gammerman, V. Vapnik, V. Vovk, C. Watkins, and J. Weston, "Support vector regression with ANOVA decomposition kernels," *Advances in Kernel Methods—Support Vector Learning*, MIT Press Cambridge, MA, pp. 285–292, 1999.

[17] A. Smola, and B. Schölkopf, "A Tutorial on Support Vector Regression," *STATISTICS AND COMPUTING*, vol. 14, pp. 199-222, 2003.

[18] D. Basak, S. Pal, and D. Patranabis, "Support Vector Regression," *Neural Information Processing – Letters and Reviews*, vol. 11, Non. 10, pp. 203-224, October 2007.

[19] X. Xia,M. Lyu, T. Lok, G. Huang, "Methods of Decreasing the Number of Support Vectors via k-Mean Clustering," *Proc. International Conference on Intelligent Computing, Lecture Notes in Computer Science book series (LNCS)*,vol. 3644 pp. 717-726, 2005.

[20] J. Hartigan, and M. Wong, "Algorithm AS 136: A k-Means Clustering Algorithm," *Journal of the Royal Statistical Society*, Series C, vol. 28, no. 1, pp. 100–108, 1979.

[21] fitrsvm: Fit a support vector machine regression mode, https://www.mathworks.com/help/stats/fitrsvm.html.