The Pseudo-DF Approach for Learning Huge-scale Data

ChangHui Deng¹, XiaoJun Wang¹,², Jun Gu¹ and Wei Wang¹

¹College of Information Engineering, Dalian Ocean University, No.52 Heishijiao St., Shahekou District, Dalian, 116023, China
²School of Management Science and Engineering, Dongbei University of Finance and Economics, No.217 Jianshan St., Shahekou District, Dalian, 116025, China
Email: wxjessicaj0903@126.com

Abstract. As the advent of the big data era, huge-scale data continuously appears in various fields of science, commerce, industry and society. More algorithms/methods/approaches are urgently required to learn huge-scale data collected from different applications/backgrounds. Therefore, the Pseudo Data Flow (Pseudo-DF) approach with ensemble ReOS-ELMs is proposed in this paper. The Pseudo-DF approach randomly divides a huge-scale data set into K (K>1) non-overlapping data chunks, and a Pseudo-DF is constructed by these data chunks. The computation of a huge-scale data is changed into that of a Pseudo-DF with smaller-scale chunks, the computational burden will be much reduced. Then, the ensemble Regularized OS-ELMs (ReOS-ELMs) based on Different random Hidden-node Parameters (DiffHPs) is presented to learn a Pseudo-DF, which is a recursive learning algorithm possessing the advantages of low computational burden, high accuracy, well generalization and stability, and strong robustness. Lastly, experiments are performed to validate the effectiveness of the proposed approach.

1. Introduction

As the advent of the big data era, the scale of data enlarges rapidly and dramatically with the advancement of data acquisition and storage techniques. Large-scale data becomes commonly and huge-scale data continuously appears in various fields of science, commerce, industry and society[1]. Huge-scale data contains abundant information which is helpful to explain phenomena, discove laws, predict trends, and so on. By now, many algorithms/methods/approaches aiming at large-scale data has emerged [2], and reversely those oriente to huge-scale data is much fewer. Moreover, huge-scale data collected from different applications/backgrounds inevitably contain unlike features. More algorithms/methods/approaches are urgently required to mine and learn huge-scale data. Therefore, this paper focuses on the study of learning approach for huge-scale data.

A natural idea to handle a huge-scale data set is to divide it into many smaller-scale chunks (sets), and then the computational burden of learning a huge-scale data set will be much reduced by learing these smaller-scale chunks. However, how to guarantee and even boost the utilization ratio of the information hidden in huge-scale data? That is, how to guarantee and even improve the generalization of the model learned on smaller-scale chunks is the most basic challenge. In order to meet this challenge, two key steps should be well considered, which are the division criterion and the information fusion of smaller-scale chunks.

On the basis of this idea, this paper proposes the Pseudo Data Flow (Pseudo-DF) approach to learn huge-scale data. In Pseudo-DF approach, a huge-scale data set is randomly divided into K (K>1) non-overlapping data chunks, and a Pseudo-DF is constructed by these data chunks. Then, a recursive
leaning algorithm is adopted to learn the Pseudo-DF, by which the information of data chunks can be transferred from the original generation of model into the (K-1)-th generation of model via updating. The last generation of model is the final model learning from a Pseudo-DF of a huge-scale data set.

Because a Pseudo-DF has the time-insensitive characteristic, and recursive leaning algorithms without forgetting mechanism should be adopted to learn it. Online Sequential Extreme Learning Machine (OS-ELM) [3] is firstly considere. With the advantages of low computational burden and strong nonlinear-fitting ability, OS-ELM can fast and accurately learn large-scale data flows [4,5]. However, OS-ELM is perplexed by the instability from random Hidden-node Parameters (HPs), becomes failed by singular and ill-posed problems, and yields bad generalization when learning on data flows with noise. In order to improve the stability of OS-ELM from random HPs, Lan et al. presented Ensemble OS-ELMs (EOS-ELMs) [6] which builds several OS-ELMs with the ensemble mode of Different random Hidden-node Parameters (DiffHPs) and averages/votes their outputs to obtain ensemble outputs. Liu et al. indicates that ensemble learning also can boost the robustness of OS-ELM from noise, and they combine particle swarm optimization selective ensemble method with OS-ELM [4]. Moreover, many scholars hold that the weak robustness of OS-ELM learning on noisy data flows is derived from its basic criterion of empirical risk, and then they added the consideration of structural risk to OS-ELM [7]. Regularized OS-ELM (ReOS-ELM) is a typical exemplar, which minimizes both the norm of the matrix of empirical error and the norm of the matrix of hidden-layer weights [7]. ReOS-ELM can also avoid singular and ill-posed problems.

Therefore, an ensemble ReOS-ELMs method is presented to learn the Pseudo-DF of a huge-scale data set. The ensemble mode of DiffHPs is applied to build diverse ReOS-ELM sub-models, and a weight average (or a vote) method is used to integrate sub-models. To validate the effectiveness (i.e., generalization and interpretability) of the proposed Pseudo-DF approach with ensemble ReOS-ELMs, experiments are carried out on one huge-scale data sets from UCI machine learning repository, and it related to a regression task.

2. ReOS-ELM

Suppose that there are \( N \) training samples \( (x_n, t_n), n = 1,...,N \), where \( t_n \) is the target corresponding to the input vector \( x_n \), and \( x_n=[x_1,x_2,...,x_M] \). \( M \) is the dimension of input attributes. The mathematical description of an ELM model is as follows [8],

\[
f(x) = h\beta, \tag{1}
\]

where \( \beta \) is the matrix of hidden-layer weights, and \( \beta=[\beta_1, \beta_2,..., \beta_J]^T, h=[h_1, h_2,..., h_J], J \) is the number of hidden nodes. \( h \) is the vector of hidden-layer outputs, \( h_j=G(a_j,b_j,x) \), \( j=1,...,J \) where \( G(\cdot) \) is the activation function, and \( (a_j,b_j) \) are randomly generated Hidden-node Parameters (HPs). Given the activation function (e.g., ‘RBF’, ‘Sigmoid’, or ‘Sine’), and random HPs (i.e., \( a \) and \( b \)), the goal of training is to estimate \( \beta \). The estimation of \( \beta \) from ReOS-ELM is the solution to minimize \( \|H\beta-T\| + \lambda \|eta\| \), where \( H \) is the matrix of hidden-layer outputs, and \( T=[t_1,t_2,..., t_N]^T \). \( \lambda \) is the Regularization Factor (RF) and \( \lambda \geq 0 \). The solution for \( \beta \) is given by \( \beta=(H^TH+\lambda I)^{-1}H^TT \), where \( I \) is an unit matrix with size \( J \times J \).

Given an initial chunk of data \( \Omega_0 \). The solution to minimizing \( \|H_0\beta_0-T_0\| + \lambda \|eta_0\| \) is given by

\[
\hat{\beta}_0 = K_0^{-1}H_0^TT_0, \tag{2}
\]

where

\[
K_0 = H_0^TH_0 + \lambda I . \tag{3}
\]

When the \( k \)-th chunk of data \( \Omega_k \) is received, we have

\[
\hat{\beta}_k = \hat{\beta}_{k-1} + (K_k)^{-1}H_k^T(T_k - H_k\hat{\beta}_{k-1}), \tag{4}
\]

where
\[ K_i^* = K_{i-1}^* + H_i^T H_i. \] (5)

3. Text The Pseudo-DF Approach with Ensemble ReOS-ELMs

The mathematical description of a regression ensemble model is given by (6),

\[ f^{\text{Ens}}(x) = \sum_{i=1}^{p} \alpha_i \cdot f_i(x), \] (6)

where \( f_i(x), i=1,2,\ldots,P, \) represents \( P \) members of ensemble, i.e., \( P \) sub-models, and \( \alpha_i \) is the weight corresponding to \( f_i(x) \). The mathematical description of a classification ensemble model is given by (7),

\[ f^{\text{Enc}}(x) = \arg \max_{i} \sum_{i=1}^{p} \text{sign}(f_i(x)) \] (7)

where \( \text{sign}(.) \) is the sign function.

Algorithm 1 The Ensemble ReOS-ELMs Algorithm

- \( \lambda (\geq 0) \) is the RF of ensemble ReOS-ELMs.
- \( P \) is the number of ReOS-ELMs sub-models and \( K \) is the number of data chunks.
- \( \Omega_0, \Omega_1, \ldots, \Omega_{K-1} \) are \( K \) data chunks.
- \( f_{\omega_p}(\cdot) \) is the \( p \)-th sub-model updated on the \( t \)-th data chunk, \( p=1,\ldots,P \), and \( t=0,\ldots,K-1 \).
- \( \alpha_{\omega_p} \) is the weight of \( f_{\omega_p}(\cdot) \).
- \( f^{\text{Ens}}_k(\cdot) \) is the \( k \)-th generation of ensemble ReOS-ELM model.
- \( t_i = [t_{i1}, t_{i2}, \ldots, t_{in}]^T \) is the targets in \( \Omega_t \), and \( \hat{T}_k \) is the predictions of \( t_i \) from \( f_{\omega_p}(\cdot), f_{\omega_{p2}}(\cdot), \ldots, f_{\omega_{pn}}(\cdot) \), where \( \hat{T}_k = [\hat{t}_{i1}, \hat{t}_{i2}, \ldots, \hat{t}_{in}] \).

1. Initialising
   - The value of \( \lambda \) is given.
   - For \( k=0 \)
     - For \( p=1:1:P \)
       - Random HPs \( (\alpha_{\omega_p}, b_{\omega_p}) \) are generated for sub-model \( f_{\omega_p}(\cdot) \).
       - \( \hat{\beta}_{\omega_p} \) is obtained from (2).
       - \( f_{\omega_p}(\cdot) \) learnt on \( \Omega_t \) are obtained from \( H_{\omega_p} \) and \( \hat{\beta}_{\omega_p} \) in terms of (1).
     - End
   - \( \alpha_{\omega_0}, \alpha_{\omega_2}, \ldots, \alpha_{\omega_p} \) are obtained on the basis of \( t_i \) and \( \hat{T}_0 \).
   - \( f^{\text{Ens}}_k(\cdot) \) is established by combining \( f_{\omega_0}(\cdot), f_{\omega_2}(\cdot), \ldots, f_{\omega_p}(\cdot) \) in terms of (6) or (7).
   - End
2. Updating (Recursive Learning)
   - For \( k=1:1:K \)
     - For \( p=1:1:P \)
       - \( \hat{\beta}_{\omega_p} \) is updated from \( \hat{\beta}_{\omega_p(k-1)} \) by the recursive function of (4).
       - \( f_{\omega_p}(\cdot) \) learnt on \( \Omega_t \) are obtained from \( H_{\omega_p} \) and \( \hat{\beta}_{\omega_p} \) in terms of (1).
     - End
   - \( \alpha_{\omega_0}, \alpha_{\omega_2}, \ldots, \alpha_{\omega_p} \) are obtained on the basis of \( t_i \) and \( \hat{T}_0 \).
   - \( f^{\text{Enc}}_k(\cdot) \) is established by combining \( f_{\omega_0}(\cdot), f_{\omega_2}(\cdot), \ldots, f_{\omega_p}(\cdot) \) in terms of (6) or (7).
   - End

End
**Ω** is a huge-scale data set with \( N \) samples. The \( N \) samples are randomly ordered and then divided into \( K \) (\( K>1 \)) non-overlapping data chunks \( \Omega_1, \Omega_2, \ldots, \Omega_K \), which constructed a Pseudo-DF. \( N_0, N_1, \ldots, N_K \) denote the number of samples in \( \Omega_0, \Omega_1, \ldots, \Omega_K \), respectively. An ensemble ReOS-ELMs model will be built on a Pseudo-DF, the algorithm of which is presented in Algorithm 1. In line 5, the ensemble mode of DiffHPs is applied to build diverse ReOS-ELM sub-models, which enable ensemble ReOS-ELMs with parallel structure and sub-models to learn and update on different computers. That is, line 5~7 (and line 15~16) can be performed independently. In line 9 and line 18, the combination rule defined by Perrone & Cooper [9] is used to calculated the weighs of sub-models, and the calculation process of it is rewrite in [10], pages 771-772.

4. Experiments

The main goal of experiments was to validate the generalization and interpretability of the proposed Pseudo-DF approach with ensemble ReOS-ELMs for the learning of huge-scale data. Experiments involved two parts. One was to investigate if the model learning on a huge-scale data set by the Pseudo-DF approach was with high generalization and interpretability. The other was to investigate if ensemble ReOS-ELMs outperformed OS-ELM, ReOS-ELM and EOS-ELMs when learning a Pseudo-DF of huge data.

![Figure 1. The curves of the RMSEs of ensemble ReOS-ELMs models on the testing set.](image)

SGEMM GPU kernel performance data set related to a regression task was used in this sub-section [11]. This huge-scale data set has 241600 samples with 10 independent input attributes and 1 target attributes. There were no missing values. The input attributes of the data set were in integer values. The range of target values was between 13.25ms and 3397.08ms, and a target value was transformed into its logarithm form. A training set was random divided into 120 data chunks which constructed a Pseudo-DF. The huge-scale data set would be randomly divided into a training set and a testing set with the proportion of 9:1. A training set was to construct a Pseudo-DF and to learn a model, and a testing set was to validate the model. The data would be normalized in the interval [0,1] before entering a model, and the outputs would be reversed after existing the model. The “Sigmoid” activation function was used by all of four recursive leaning algorithm. The evaluating criterion of performance is Root Mean Square Error (RMSE). The random division of the huge-scale data set into a training set and a testing set was repeated 10 times and the reported RMSE was the averages over the 10 iterations, i.e., statistical results.

In any ReOS-ELM model, \( J \) was also set as 1300. To select a suitable value of \( \lambda \) for an ensemble ReOS-ELMs model, five nominated values of \( \lambda \) were investigated. The curves of the testing RMSEs of ensemble ReOS-ELMs models when \( \lambda = 0, 0.1, 0.5, 1 \) or 2 in each ReOS-ELM sub-model were illustrated in Fig.1. The first point of each curve showed the testing RMSE of a single ReOS-ELM model. As the number of ReOS-ELM sub-models increasing, the testing RMSE gradually reduced and converged to a fixed RMSE. It was indicated that the generalization of an ensemble ReOS-ELMs model were higher than those of a single ReOS-ELM model.

The curves showed that when \( \lambda = 0.5 \), the ensemble ReOS-ELM model obtained the lowest RMSE. Thus, it is useful to set the value of \( \lambda \) as 0.5 for an EOS-ELM model of the Pseudo-DF derived from SGEMM GPU kernel performance data set. Moreover, EOS-ELMs was a special case of ensemble
ReOS-ELMs when $\lambda=0$. It was indicated that ensemble ReOS-ELMs outperform EOS-ELMs for building a model of the Pseudo-DF derived from a huge-scale data set. The comparisons of ensemble ReOS-ELMs, OS-ELM, ReOS-ELM, EOS-ELMs and on the testing set of SGEMM GPU kernel performance data set were displayed in Table 1. The results show that the model of a huge-scale data set learning by the Pseudo-DF approach had high generalization and interpretability. Among the four recursive learning algorithms, the testing RMSE of the ensemble ReOS-ELMs model were lowest. The performance of an ensemble (i.e., EOS-ELMs or ensemble ReOS-ELMs) model outperformed that of single models (i.e., ReOS-ELM and OS-ELM).

Table 1. The comparisons of ensemble ReOS-ELMs, OS-ELM, EOS-ELMs and ReOS-ELM on the testing set.

| recursive leaning algorithm | $\lambda$ | $P$ | RMSE(ms) |
|-----------------------------|-----------|-----|----------|
| ensemble ReOS-ELMs          | 0.5       | 20  | 0.1708   |
| EOS-ELMs                    | 0         | 20  | 0.1721   |
| single ReOS-ELM             | 0.5       | 1   | 0.1875   |
| single OS-ELM               | 0         | 1   | 0.1892   |

5. Conclusion
The main contribution of this paper is that the Pseudo-DF approach with ensemble ReOS-ELMs for learning huge-scale data is proposed. The random and non-overlapping division criterion is applied to a huge-scale data set for generating a Pseudo-DF. As a more stable and robustness recursive leaning algorithm, ensemble ReOS-ELMs is presented to learn the Pseudo-DF, which can realize the information fusion of smaller-scale chunks via updating models, that is, the information of data chunks can be transferred from the original generation of model into the last generation of model. The results showed that the model of a huge-scale data set learning by the Pseudo-DF approach had high generalization and interpretability, which was satisfactory. The results also showed that the generalization of ensemble ReOS-ELMs outperforms that of EOS-ELMs, ReOS-ELM and OS-ELM for a regression.

6. Acknowledgements
The authors would like to thank all creators and donators of the data sets used in this paper for their kindly sharing the data. The authors also would like to acknowledge UCI Machine Learning Repository [http://archive.ics.uci.edu/ml]. This work is supported by the National Natural Science Foundation of China (No. 61702070) and the Research Projects of Liaoning Marine Fisheries Office (No. 201512).

7. References
[1] Wang X-J, Liu Y, Yuan P, Zhou C-J, Zhang L (2019). *IEEE Access*, 7: 7368-7380.
[2] Wang X, You M, Mao Z, Yuan P (2017). *Adv Eng Inform*, 30:368-375.
[3] Liang N-Y, Huang G-B, Saratchandran P, Sundararajan N (2006). *IEEE T Neural Networ*, 17(6):1411-1423, 2006.
[4] Liu Y, He B, Dong D, Shen Y, Yan T, Nian R, Lendasse A (2014). *Proceedings of ELM-2014*. 1:325-344.
[5] Ding S, Mirza B, Lin Z, Cao J, Sepulveda J (2018). *Neurocomputing*, 277:139-148.
[6] Lan Y, Soh Y C, Huang G-B (2009). *Neurocomputing*, 72:3391-3395, 2009.
[7] Huynh H T and Won, Y (2011). *Pattern Recognition Letters*, 32:1930-1935.
[8] Huang G B, Chen L, Siew C K (2006). *IEEE T Neural Networ*, 17(4):879-892.
[9] Perrone M P and Cooper L N. Artificial Neural Networks for Speech and Vision. London: Chapman & Hall., 126-142, 2001.
[10] Wang X.(2017). *IEEE/CAA J Automatica Sinica*, 4(4):770-774.
[11] Ballester-Ripoll R, Paredes E G., Pajarola R (2017). Sobol Tensor Trains for Global Sensitivity Analysis. In arXiv Computer Science/Numerical Analysis e-prints.