Two Steps Genetic Programming for Big Data

– Perspective of Distributed and High-Dimensional Data

Jih-Jeng Huang
Department of Computer Science & Information Management,
SooChow University, Taipei, Taiwan.
e-mail: jjhuang@scu.edu.tw

***Abstract*—The term big data has been the most popular topic in recent years in practice, academe and government for realizing the value of data. Then, many information technologies and software are proposed to deal with big data, such as Hadoop, NoSQL databases, and cloud computing. However, these tools can only help us to store, manage, search, and control data rather than extract knowledge from big data. The only way to mine the nugget from big data is to have the ability to analyze them. The characteristics of complexity of big data, e.g., volume and variety make traditional data mining algorithms invalid. In this paper, we deal with big data by solving distributed and high-dimensional problems. We propose a novel algorithm to effectively extract knowledge from big data. According to the empirical study, the propose method can handle big data soundly.**

Keywords-big data; knowledge extraction; distributed problems; high-dimensional problems; genetic programming.

I. INTRODUCTION

The term big data has been a popular topic recently in practice, academe, and government to reflect the needs of using the huge data. Big data refer to data sets which are so large and complex that is beyond the ability of typical software tools to capture, store, manage, and analyze it within a tolerable elapsed time [33]. The purpose of collecting big data is similar to tradition data mining to solve the key problems of society, business and science. However, the massive volume of data makes it very difficult to perform effective analysis using the existing traditional techniques [23]. In addition, other characteristics like velocity, variety, variability, value and complexity [13-14] put forward the big data issue more challenge.

To deal with the complexity of big data, many information technologies and software have been proposed, e.g. Hadoop, NoSQL, and cloud computing. These solutions are usually technological orientation rather from the perspective of theory. However, there are still many challenges to uncover the nugget in theory. As shown from the results of a 2012 survey (TM Forum, 2012), the top four big data challenges are data integration, data volume, skill availability, and solution cost. Among these issues, data integration and skill availability cannot simply be solved by information technology.

The problem of data integration comes from the property of variety in big data. With such variety, a challenge is how to combine the distributed and massive meaningful features for analysis. Although it is convenient to combine all features across tables, it may suffer the curse of dimensionality [30] and problems of feature selection [18, 40]. On the other hand, the problem of skills availability is the fact that the traditional data mining methods cannot deal with big data due to these data are stored distributed [28].

In the field of machine learning, high dimensional data analysis and distributed data mining (DDM) algorithms are newly developing topics and received much attention recently. Although these issues are clearly related to big data, they are not well-integrated and should be overcome appropriately.

The value of big data is indubitable. However, how to transform big data to big value is the main issue. Although there are plenty of tools and architectures, such as Hadoop, MapReduce, NoSQL database etc., to search, manage, store, and control huge volume of data, the analysis of big data can truly derive the nugget of big data. Therefore, the purpose of this paper is to propose algorithms and procedures to solve the above problems from the perspective of machine learning in classification problems.

II. HIGH-DIMENSIONAL AND DISTRIBUTED PROBLEMS

High-dimensional data analysis presents an exponential difficulty to model and optimize data set with features [24, 32, 39]. The reason for considering the high-dimensional problem is the curse of dimensionality [2] to indicate that the number of samples needed to estimate an arbitrary function with a given level of accuracy grows exponentially with the number of dimensions that it comprises. The goal of high-dimensional data analysis has been reported by [3] to accurately predict the future observations and gain insight into the relationship between the features and response for scientific purposes. There are many algorithms have been proposed to analyze high-dimensional data. Generally, we can divide them as the dimension reduction techniques and feature selection algorithms [17, 43]. [41] proposed the concept of predominant correlation to identify relevant features as well as redundancy among relevant features without pairwise correlation analysis.

Massively distributed storage is a fundamental change in dealing with the property of huge volume in big data. A distributed database system means the analytic tables are located in different database. However, we cannot simply merge all tables for analysis due to the restrictions of the computing power and memory of a computer. Classical methods are not designed to cope with this kind of problem. Hence, big data arise additional problems of high-dimensional data analysis to understand heterogeneity and commonality across different subpopulations [15].

On the other hand, distributed data mining (DDM) [16, 34] is concerned with the data mining in a distributed computing environment. The first step of DDM is to analyze the local database at each distributed site to obtain the local

978-1-4673-7278-7/15 $31.00 © 2015 IEEE
DOI 10.1109/BigDataCongress.2015.125

model. Then, with the integration of the distributed local models is performed, we can obtain the global model.

Distributed databases can be divided into horizontally distributed (homogeneous) and horizontal distributed (heterogeneous) databases. In the former case, all databases across distributed data sites contain the same set of features. On the other hand, in heterogeneous situation, the features differ among the distributed databases. There are many algorithms which are based on statistical techniques, meta-learning and ensemble generation algorithms have been proposed to deal with homogeneous data sites, e.g. [4-9]. However, as stated previously, the heterogeneous case approaches the realistic situation in big data. Hence, we focus only on the existing literature for heterogeneous DDM.

The issues in mining from heterogeneous data are discussed in [36] according to the perspective of inductive bias. Their method used the concept of inter-site correlations to handle distributed data problem. Later, [1] employed activation spreading approach to deal with heterogeneous data by computing cardinal distribution of the feature values in the individual data sets. However, the above two methods are too restricted because they need to ask the relationship among databases and only based on the first order statistical approximation of the distribution. These inter-database pattern cannot be captured by the aggregation of heterogeneous local classifiers [34].

[20,21] proposed collective data mining (CDM) for predictive data modeling in heterogeneous environments. CDM is a foundation in which any function can be represented in a distributed fashion using an appropriate set of basis functions. When the basis functions are orthonormal, the local analysis produce correct and useful results that can be directly used as a component of the global model without loss of accuracy.

Usually, the performance of CDM depends on the quality of estimated cross-terms. Many algorithms have been proposed based on the concept of CDM. For example, collective principal component analysis [20-21], distributed clustering algorithm [21], collective multivariate regression [20-21], and distributed decision tree construction [35].

There are a few papers have been proposed recently to handle the distributed high-dimensional data. For example, [26] proposed a fast outlier detection strategy for distributed high-dimensional data sets with mixed features; [27] used suitable distance function between the feature vector approximations to deal with distributed high-dimensional problems. However, the former have to restrict the mixed features and the latter can only suit in homogeneous distributed databases. A more flexible and integrated strategy or algorithm should be proposed in this file. In this proposal, we hope to cope with this issue appropriately.

III. TWO STEPS GENETIC PROGRAMMING (2SGP)

A. ICA and Genetic Programming

Feature extraction involves reducing the amount of the dimensions of a data set and is a common way to deal with the high-dimensional data. The most well-known and popular feature extraction method is principle component analysis (PCA). Although several papers successfully used PCA for feature extraction in many applications, e.g. [31,37], the limitations of PCA has been reported as only using two-order statistics and a linear technique [6,10].

In practice, the transform defined by second-order methods like PCA is not useful for many purposes where optimal reduction of dimension in the mean-square sense is not needed. This is because PCA neglects such aspects of non-Gaussian data and independence of the components (which, for non-Gaussian data, is not the same as uncorrelatedness). Hence, high-order technique may be constructed for non-Gaussian data.

Independent component analysis (ICA) is one of the statistical tool to extract the independent component (IC) from an observed multivariate series according to high-order statistics. The major distinguish of ICA from other methods is that it looks for components that are both statistically independent and non-Gaussian [6]. ICA has been proposed to deal with many real-world applications such as signal processing, magnetoencephalography (MEG), and image analysis. The concepts of ICA can be described as follows. Let a signal vector be $\boldsymbol{x}_t = \{x_1, x_2, \cdots, x_n\}$, the ICA model can be formulated as

$$\boldsymbol{x}_t = \boldsymbol{A}\boldsymbol{s}_t \tag{1}$$

where A denotes the unknown mixing matrix and s denotes the sources.

The problem of ICA is to extract the IC vector, $\boldsymbol{y}_t$, from the signal vector, $\boldsymbol{s}_t$. We can depict the problem above as shown in Figure 2.

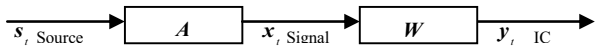

Figure 1. The concept of ICA

In order to derive the ICs, we can calculate the demixing matrix, $\boldsymbol{W}$, such that

$$\boldsymbol{y}_t = \boldsymbol{W}\boldsymbol{x}_t = \boldsymbol{W}\boldsymbol{A}\boldsymbol{s}_t. \tag{2}$$

Therefore, if we can find $\boldsymbol{W} = \boldsymbol{A}^{-1}$, then $\boldsymbol{y}_t = \boldsymbol{s}_t$, and the perfect separation occurs.

In traditional ICA, an n by n sequence x should be extract *n* independent components (ICs). However, if the number of sources is unavailable and the purpose is to reduce the dimensionality of the data, we need to derive the rectangular unmixing matrix. Hence, if we want to obtain the subset of the original single s, we should transform our problem to recover the following problem:

$$\boldsymbol{y}_t = \boldsymbol{W}\boldsymbol{x}_t = \boldsymbol{W}\boldsymbol{A}\boldsymbol{s}_t. \tag{2}$$

where K< N. Usually, we can use some criteria, e.g., entropy or mutual information, to achieve the purpose.

In this paper, the FastICA algorithm [25] is used to reduce the dimensionality of the distributed data tables. The algorithm minimizes the mutual information to derive ICs based on the fixed-point iteration schema. The FastICA has been widely and successfully used for many applications to reduce the dimensionality of the problems. In addition, the nonlinear FastICA algorithm have been proposed to consider

nonlinear relationships between ICs [19]. It can be also used here to obtain the subset of each data table.

On the other hand, genetic programming (GP) was proposed by [29] to automatically extract intangible relationships in a system and has been used in many applications, such as symbolic regression [38] and classification [42]. The preparatory steps of GP contains to determine the terminals, functions, fitness function, parameters, and termination criterion. All these preparatory steps are problem dependent. The representation of GP can be viewed as a tree-based structure composed of the function set and terminal set.

Once we initialize a population of the GP tree, the following procedures are similar to genetic algorithms (GAs) including defining the fitness function, genetic operators, such as crossover, mutation and reproduction, and the termination criterion. The fitness function can be considered as the way we define the problem to GP. For example, the fitness function can be classification accuracy or error rate.

Similar to GAs, GP uses the mutation operator in order to avoid falling into the local optima. The mutation operator is used to randomly choose a node in a subtree and replace it with a new created subtree randomly. Finally, a new generation can be reproduced from two parents using the reproduction operator to represent a better solution. Readers can refer to [29] for understating the contents of GP in detail. Here, we focus on the purpose here to highlight the feature selection in GP as follows.

The initial GP-trees are randomly generated and not all features must appear within a GP-tree. Hence, from the description of GP, it can be seen that one of the unique characteristic of GP is its built-in mechanism to select the features that are related to the problem via the operators of evolution. In this way, a nonlinear variable selection can be used for dimensional reduction. It should be highlighted that the concept of feature selection between ICA and GP are different. In ICA, new features are generated to replace the original features. On the other hand, GP derives the significant features from the original features.

B. 2SGP

In this paper, we follow the concept of DMM to first derive the local models from distributed databases and then calculate the global model. However, since high-dimension features may destroy the data structure of the problem and decrease the efficiency of algorithms, we should first reduce our features by some dimension reduction techniques. In this paper, we first consider ICA as the tools for reduce the features of local databases. Then, we will use GP to derive the local models from each distributed databases. Finally, we will integrated all local models into the global model by using genetic programming again. The genetic trees of the local models will be the initial population of the global genetic programming. In addition, we will sample the validation set from the distributed databased to determine the final global model by the cross-validation method.

IV. EMPIRICAL STUDY

Here, we use SUSY Data Set as the empirical study. The problem of SUSY Data Set is to distinguish between a signal process which produces supersymmetric particles and a background process which does not. It contains 18 features. The first 8 features are kinematic properties measured by the particle detectors in the accelerator. The last ten features are functions of the first 8 features; these are high-level features derived by physicists to help discriminate between the two classes. There is an interest in using deep learning methods to obviate the need for physicists to manually develop such features.

To test the proposed algorithm, we divide the data set into three local data set. Three local data sets contain 4.5 million random records with different 6 features for training the local models. The test data set contains the rest 0.5 million records used in the global model. Then, we use the ICA for each local data set to extract and reduce features. The eigenvalues of three local data sets can be depicted as shown in Figure 2.

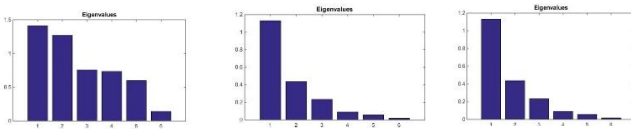

Figure 2. Eigenvalues of three local data sets

Here, we retrieve first four eigenvalues for the first local data set, three eigenvalues for the second data set, and two eigenvalues for the third data set and the retained information of eigenvalues is 66.98%, 84.58%, and 79.87%, respectively. Then, we calculate the independent components (ICs) for the three local data sets and obtain the results as shown in Figure 3:

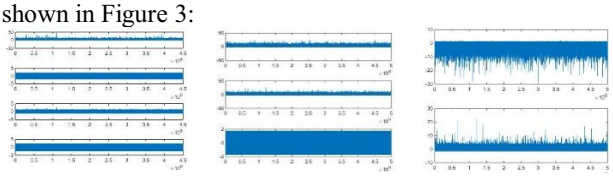

Figure 3. Independent components of three local data sets

Next, we regard the ICs of each local data set as the independent features to run genetic programing to obtain the local model. We run each local data set ten times, respectively, to obtain 30 genetic trees. These genetic trees are the initial population of training global model. In this case, we merge three local data sets and randomly select 2 million records to be the training set. Then, we run genetic programming to obtain the final global model and used to classify the test data set. Here, we run genetic programming five times to calculate the accuracy rates of data, as shown in Table II.

TABLE I. COMPARISON RESULTS

Accuracy ratio		Run 1	Run 2	Run 3	Run 4	Run 5	Average
2SGP	Test	70.8766%	72.0436%	70.8047%	70.8318%	69.3050%	70.7723%
	Training	70.9277%	72.0702%	70.8342%	70.8934%	69.3677%	70.8186%
GP	Test	68.6981%	67.5822%	70.7860%	70.6515%	69.3182%	69.4072%
	Training	68.7535%	67.5216%	70.8327%	70.7242%	69.2915%	69.4247%

From Table 2, it can be seen that the accuracy ratio of the proposed method is slightly higher than the conventional method. However, the proposed method only deals with six

features. In contrast, conventional GP uses 18 features to generate the final model. Hence, it is asserted that the proposed method is more suitable for handling the problem of big data.

V. CONCLUSION

In this paper, we proposed an integrated algorithm to deal with the distributed classification problem of big data. First, we use ICA to reduce the dimensionality of local data sets and retain the importance information of features. Then, we adopt GP to generate genetic trees as the local model and initial population of the global model. Finally, we run GP again to obtain the final global model. According to the results of the empirical study, the proposed method outperforms than the conventional GP with respect to the accuracy ratio. In addition, the proposed method can be considered as a way to deal with the classification of big data.

REFERENCES

[1] Aronis, J., Kolluri, V., Provost, F., & Buchanan, B. (1997). The WoRLD: Knowledge discovery from multiple distributed databases. In Proceedings of 10th international Florida AI research symposium, 337-341.

[2] Bellman, R. (1957). Dynamic Programming. Princeton, NJ: Princeton University Press.

[3] Bickel, P., Li, B., & Bengtsson, T. (2008). Sharp failure rates for the bootstrap particle filter in high dimensions. In Pushing the limits of contemporary statistics: Contributions in honor of Jayanta K. Ghosh (pp. 318-329). Institute of Mathematical Statistics.

[4] Breiman, L. (1996). Bagging predictors. Machine learning, 24(2), 123-140.

[5] Breiman, L. (1999). Prediction games and arcing algorithms. Neural computation, 11(7), 1493-1517.

[6] Cardoso, J. F. (1999). High-order contrasts for independent component analysis. Neural computation, 11(1), 157-192.

[7] Chan, P. K., & Stolfo, S. J. (1993). Experiments on multistrategy learning by meta-learning. In Proceedings of the second international conference on information and knowledge management, 314-323.

[8] Chan, P. K., & Stolfo, S. J. (1998). Toward Scalable Learning with Non-Uniform Class and Cost Distributions: A Case Study in Credit Card Fraud Detection. In KDD (Vol. 1998, 164-168).

[9] Cheung, D. W., Han, J., Ng, V. T., & Wong, C. Y. (1996). Maintenance of discovered association rules in large databases: An incremental updating technique. In Data Engineering, 1996. Proceedings of the Twelfth International Conference on, 106-114.

[10] Du, Q., & Kopriva, I. (2008). Automated target detection and discrimination using constrained kurtosis maximization. Geoscience and Remote Sensing Letters, IEEE, 5(1), 38-42.

[11] Hubert, M., Rousseeuw, P. J., & Vanden Branden, K. (2005). ROBPCA: a new approach to robust principal component analysis. Technometrics, 47(1), 64-79.

[12] Chen, D., & Yang, L. (2006). Exploiting high dimensional video features using layered Gaussian mixture models. In Pattern Recognition, 2006. ICPR 2006. 18th International Conference on, 2, 1078-1081.

[13] Demchenko, Y., Ngo, C., de Laat, C., Membrey, P., & Gordijenko, D. (2014). Big Security for Big Data: Addressing Security Challenges for the Big Data Infrastructure. In Secure Data Management (pp. 76-94). Springer International Publishing.

[14] Fan, W., & Bifet, A. (2013). Mining big data: current status, and forecast to the future. ACM SIGKDD Explorations Newsletter, 14(2), 1-5.

[15] Fan, J., Han, F., & Liu, H. (2014). Challenges of Big Data analysis. National Science Review, nwt032.

[16] Fu, Y. (2001). Distributed data mining: An overview. Newsletter of the IEEE Technical Committee on Distributed Processing, 5-9.

[17] Hesterberg, T., Choi, N. H., Meier, L., & Fraley, C. (2008). Least angle and ℓ1 penalized regression: A review. Statistics Surveys, 2, 61-93.

[18] Hoi, S. C., Wang, J., Zhao, P., & Jin, R. (2012). Online feature selection for mining big data. In Proceedings of the 1st International Workshop on Big Data, Streams and Heterogeneous Source Mining: Algorithms, Systems, Programming Models and Applications, 93-100.

[19] Lappalainen, H., & Honkela, A. (2000). Bayesian non-linear independent component analysis by multi-layer perceptrons. In Advances in independent component analysis, 93-121.

[20] Kargupta, H., Byung-Hoon, D. H., & Johnson, E. (1999). Collective data mining: A new perspective toward distributed data analysis. In Advances in Distributed and Parallel Knowledge Discovery.

[21] Kargupta, H., Sivakumar, K., Huang, W., Ayyagari, R., Chen, R., Park, B. H., & Johnson, E. (2001). Towards ubiquitous mining of distributed data. In Data Mining for Scientific and Engineering Applications (pp. 281-306). Springer US.

[22] Hershberger, D. E., & Kargupta, H. (2001). Distributed multivariate regression using wavelet-based collective data mining. Journal of Parallel and Distributed Computing, 61(3), 372-400.

[23] Katal, A., Wazid, M., & Goudar, R. H. (2013). Big data: Issues, challenges, tools and Good practices. In Contemporary Computing (IC3), 2013 Sixth International Conference on, 404-409.

[24] Koch, P. N., Simpson, T. W., Allen, J. K., & Mistree, F. (1999). Statistical approximations for multidisciplinary design optimization: the problem of size. Journal of Aircraft, 36(1), 275-286.

[25] Koldovsky, Z., Tichavsky, P., & Oja, E. (2006). Efficient Variant of Algorithm FastICA for Independent Component Analysis Attaining CramÉr-Rao Lower Bound. Neural Networks, IEEE Transactions on, 17(5), 1265-1277.

[26] Koufakou, A., & Georgiopoulos, M. (2010). A fast outlier detection strategy for distributed high-dimensional data sets with mixed attributes. Data Mining and Knowledge Discovery, 20(2), 259-289.

[27] Kriegel, H. P., Kunath, P., Pfeifle, M., & Renz, M. (2005). Approximated clustering of distributed high-dimensional data. In Advances in Knowledge Discovery and Data Mining (pp. 432-441). Springer Berlin Heidelberg.

[28] Kumar, P., & Pandey, K. (2013). Big Data and Distributed Data Mining: An Example of Future Networks. International Journal, 2, 36-39.

[29] Koza, J. R. (1992). Genetic programming: on the programming of computers by means of natural selection (Vol. 1). MIT press.

[30] Lange, K., Papp, J. C., Sinsheimer, J. S., & Sobel, E. M. (2014). Next-generation statistical genetics: Modeling, penalization, and optimization in high-dimensional data. Annual Review of Statistics and Its Application, 1, 279-300.

[31] Li, X. R., Jiang, T., & Zhang, K. (2006). Efficient and robust feature extraction by maximum margin criterion. Neural Networks, IEEE Transactions on, 17(1), 157-165.

[32] Li, G., Rosenthal, C., & Rabitz, H. (2001). High dimensional model representations. The Journal of Physical Chemistry A, 105(33), 7765-7777.

[33] McKinsey Global Institute, Big data: The next frontier for innovation, competition, and productivity, 2013.

[34] Park, B. H., & Kargupta, H. (2002). Distributed data mining: Algorithms, systems, and applications.

[35] Park, B., Kargupta, H., Johnson, E., Sanseverino, E., Hershberger, D., & Silvestre, L. (2001). Distributed, collaborative data analysis from heterogeneous sites using a scalable evolutionary technique. Applied Intelligence, 16(1), 19-42.

[36] Provost, F. J., & Buchanan, B. G. (1995). Inductive policy: The pragmatics of bias selection. Machine Learning, 20(1-2), 35-61.

[37] Rosipal, R., Girolami, M., Trejo, L. J., & Cichocki, A. (2001). Kernel PCA for feature extraction and de-noising in nonlinear regression. Neural Computing & Applications, 10(3), 231-243.

[38] Searson, D. P., Leahy, D. E., & Willis, M. J. (2010, March). GPTIPS: an open source genetic programming toolbox for multigene symbolic regression. In Proceedings of the International multiconference of engineers and computer scientists (Vol. 1, pp. 77-80).

[39] Shorter, J. A., Ip, P. C., & Rabitz, H. A. (1999). An efficient chemical kinetics solver using high dimensional model representation. The Journal of Physical Chemistry A, 103(36), 7192-7198.

[40] Wu, X., Zhu, X., Wu, G. Q., & Ding, W. (2014). Data mining with big data. Knowledge and Data Engineering, IEEE Transactions on, 26(1), 97-107.

[41] Yu, L., & Liu, H. (2004). Efficient feature selection via analysis of relevance and redundancy. The Journal of Machine Learning Research, 5, 1205-1224.

[42] Zhang, Y., & Bhattacharyya, S. (2004). Genetic programming in classifying large-scale data: an ensemble method. Information Sciences, 163(1), 85-101.

[43] Zuber, V., & Strimmer, K. (2011). High-dimensional regression and variable selection using CAR scores. Statistical Applications in Genetics and Molecular Biology, 10(1), 1-27.