Random Forest Algorithm in Gene Expression Programming for Dynamic Data

[1]Dr.P.Srimanchari, [2]Dr.G.Anandharaj

[1]Assistant Professor and Head, Department of Computer Applications, Erode Arts and Science College (Autonomous), Erode, India. *srimanchari@gmail.com*

[2]Associate Professor and Head, Department of Computer Science, Adhiparasakthi College of Arts and Science (Autonomous), Kalavai, Vellore, India. *younganand@gmail.com*

Abstract - Random forests or random decision forests are an ensemble learning method for classification, regression and other tasks, that operate by constructing a multitude of decision trees at training time and outputting the class that is the mode of the classes (classification) or mean prediction (regression) of the individual trees. Random decision forests correct for decision trees' habit of over fitting to their training set. We use F-Measure and G-mean to evaluate the performance of the algorithm. Experiment result shows that the ensemble random forest algorithm outperformed SVM and other classification algorithms in both performance and accuracy within the imbalanced data, and it is useful for improving the accuracy of product marketing compared to the traditional artificial approach.

Keywords: Classification, Regression, Decision tree, F-Measure,

I. INTRODUCTION

This marketing way has achieved good results in the past, which maintained the company sales performance for a long time through widespread sales. With the gradual opening of the insurance industry, a large number of private insurance companies enter the market, which forms a healthy competitive environment and constantly promote the reform of the insurance industry. On the other hand, people's willingness to purchase insurance gradually increased, the potential insurance customers are rapidly expanding.

According to statistics, the success rate of the traditional telephone sale is less than one thousandth, and the insurance sales rate of senior insurance salesmen can reach about two percent, but this is obviously very inefficient. Therefore, how to better accurately understand the users' purchase intention has become a very urgent need for the insurance company. Gene expression programming (GEP) is a data driven evolutionary technique that well suits for correlation mining. Parallel GEPs are proposed to speed up the evolution process using a cluster of computers or a computer with multiple CPU cores. However, the generation structure of chromosomes and the size of input data are two issues that tend to be neglected when speeding up GEP in evolution.

To fill the research gap, this paper proposes three guiding principles to elaborate the computation nature of GEP in evolution based on an analysis of GEP schema theory. As a result, a novel data engineered GEP is developed which follows closely the generation structure of chromosomes in parallelization and considers the input data size in segmentation. Via pushing data as a kind of commodity into a digital market, the data owners and consumers are able to connect with each other, sharing and further increasing the utility of data. Nonetheless, to enable such an effective market for data trading, several challenges need to be addressed, such as determining proper pricing for the data to be sold or purchased, designing a trading platform and schemes to enable the maximization of social welfare of trading participants with efficiency and privacy preservation, and protecting the traded data from being resold to maintain the value of the data.

To address the aforementioned issues, in this paper we conduct a comprehensive survey of big data trading to assist newcomers and provide a general understanding of this complex discipline and emergent research area. Our contributions are listed as follows:

- The review existing research related to big data, and identifies the big data lifecycle for data trading, including data collection, data analytics, data pricing, data trading, and data protection. It is worth noting that, because a significant volume of research has been devoted to data collection and data analytics, our survey focuses on data pricing, data trading, and data protection, which have not been well explored.

 DOI : 10.18231/2454-9150.2018.1097

- The existing research related to big data pricing. We first illustrate the principles of data pricing and explain the reasons why this process is important.
- Categorize the popular market structures, data pricing strategies, and data pricing models, and list the advantages and limitations of each category.
- Investigate the data trading process, and summarize data trading issues and the solutions for handling those issues.
- Systematically investigate the auction, as one popular trading strategy, and detail different auction schemes, related platforms, and issues with respect to efficiency, security, and privacy protection, big data lifecycle: data protection.

II. RELATED WORKS

Usually, the classification problem for unbalanced data sets has two pretreatment methods.

(1) Over sampling, this generates data of the minority class to balance the proportion of data through some specific algorithms

(2) Under sampling, this reduces the proportion of the majority class through some sampling algorithm, so as to balance the number of positive and negative cases of training set [7].

Over sampling method is mainly divided into two types,

(1) A non-heuristic sampling method which increases minority class samples by random replication, is easy to cause the over fitting of decision boundary

(2) A heuristic sampling method, which is represented by SMOTE algorithm [1], balances the category distribution of original data sets by adding some virtual samples.

In recent years, many improved algorithms are proposed base on SMOTE, such as SMOTE-RSB [2] algorithm combined with the theory of RST, which filter the samples from the final sampling result when their similarity is greater than the given threshold; SMOTE-IPF [3] algorithm uses multi noise filter to resample synthetic sample data; SMOTE-FRST[4] algorithm is also combined with the theory of RST, which remove the synthetic samples that are less than the distance threshold; Borderline- SMOTE [10] focuses on the samples of minority class in the decision boundary when the law in the sample of a few samples on the boundary of decision during the sampling. Sampling method is also divided into non-heuristic method and heuristic method.

(1) Non-heuristic method randomly remove the samples of majority class, in order to reduce the degree of imbalance, but this method will usually remove some key samples, which results in under fitting on the boundary of decision because of the lack of key characteristics

(2) The heuristic sampling method usually distinguishes samples based on the recent neighbor algorithm, which divides the samples into safe points, dangerous points and noise points [5]. Representative algorithms include Tomek links [6], compressed nearest neighbor algorithm CNN [8], nearest neighbor removal algorithm NCL [9], and so on. In addition, some research scholars also proposed some novel algorithms such as boosting, bagging and other combination algorithm [11], [12], reverse random under sampling [7] and so on.

III. RANDOM ALGORITHM

3.1 Preliminaries: Decision tree learning

Decision trees are a popular method for various machine learning tasks. Tree learning "come[s] closest to meeting the requirements for serving as an off-the-shelf procedure for data mining", say Hastie *et al.*, "because it is invariant under scaling and various other transformations of feature values, is robust to inclusion of irrelevant features, and produces inspect able models. However, they are seldom accurate. [8]

In particular, trees that are grown very deep tend to learn highly irregular patterns: they over fit their training sets, i.e. low bias, but very high variance. Random forests are a way of averaging multiple deep decision trees, trained on different parts of the same training set, with the goal of reducing the variance [9]. This comes at the expense of a small increase in the bias and some loss of interpretability, but generally greatly boosts the performance in the final model.

Bagging

The training algorithm for random forests applies the general technique of bootstrap aggregating, or bagging, to tree learners. Given a training set $X = x_1, \ldots, x_n$ with responses $Y = y_1, \ldots, y_n$, bagging repeatedly (B times) selects a random sample with replacement of the training set and fits trees to these samples:

For $b = 1, \ldots, B$:

1. Sample, with replacement, n training examples from X, Y; call these X_b, Y_b.
2. Train a classification or regression tree f_b on X_b, Y_b.

$$\hat{f} = \frac{1}{B}\sum_{b=1}^{B} f_b(x')$$

After training, predictions for unseen samples x' can be made by averaging the predictions from all the individual regression trees on x' :

or by taking the majority vote in the case of classification trees or by taking the majority vote in the case of classification trees [10].

 DOI : 10.18231/2454-9150.2018.1097

$$\sigma = \sqrt{\frac{\sum_{b=1}^{B}(f_b(x') - \hat{f})^2}{B-1}}.$$

This bootstrapping procedure leads to better model performance because it decreases the variance of the model, without increasing the bias. This means that while the predictions of a single tree are highly sensitive to noise in its training set, the average of many trees is not, as long as the trees are not correlated. Simply training many trees on a single training set would give strongly correlated trees (or even the same tree many times, if the training algorithm is deterministic); bootstrap sampling is a way of de-correlating the trees by showing them different training sets. Additionally, an estimate of the uncertainty of the prediction can be made as the standard deviation of the predictions from all the individual regression trees on x':

The number of samples/trees, B, is a free parameter. Typically, a few hundred to several thousand trees are used, depending on the size and nature of the training set [11-14]. An optimal number of trees B can be found using cross-validation, or by observing the *out-of-bag error*: the mean prediction error on each training sample x_i, using only the trees that did not have x_i in their bootstrap sample The training and test error tend to level off after some number of trees have been fit.

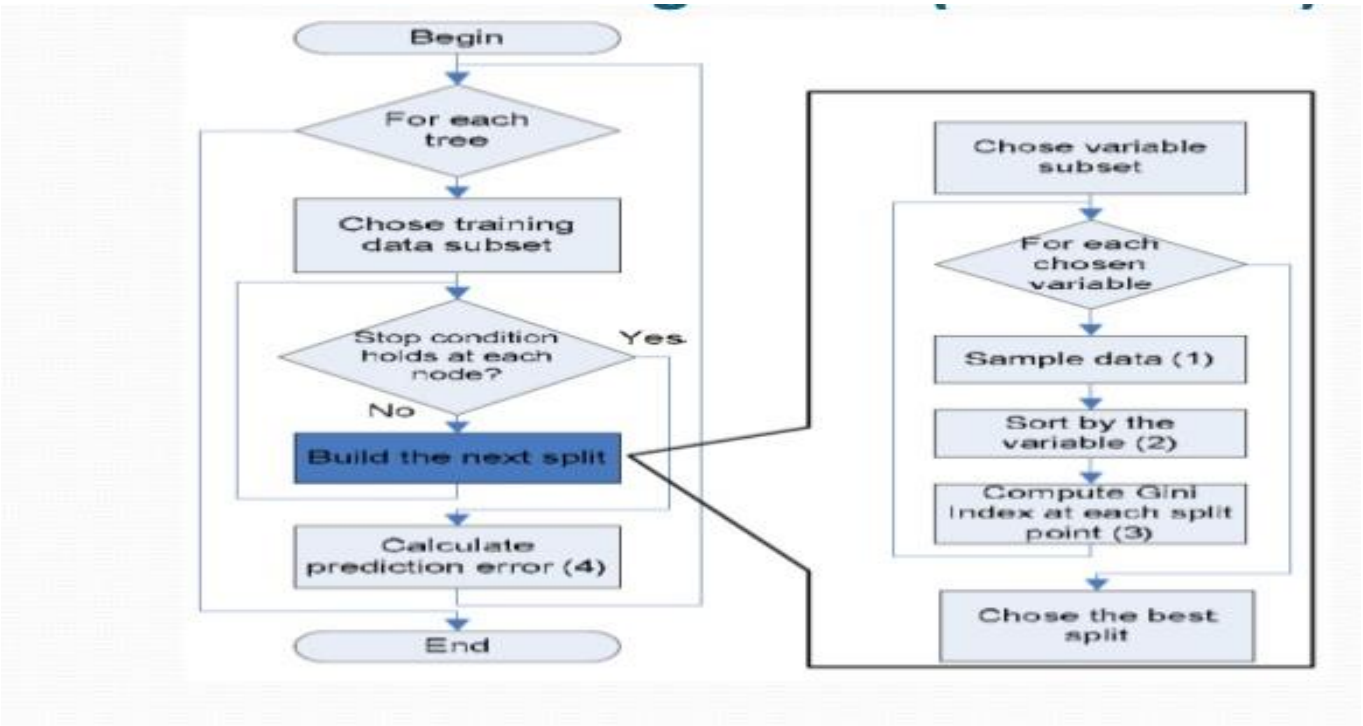


Figure 3.1 Flowcharts for Random Algorithm

For a lot of classification models, the distribution balance and correlation of features directly affect the forecast results. Due to the imbalance distribution of the insurance business data feature and the independence between each other, directly using SVM or other strong classifier algorithms such as decision tree etc. will make a serious deviation result. So in handling such kind of problem, we prefer to use boosting algorithms or ensemble algorithms [15]. The primary problem of imbalance feature dataset classification is data resampling, there are two main types of paradigm through the previous research which called oversampling and under-sampling. In this article, we used a heuristic under-sampling method which using the bootstrap under-sampling with replace for several times on the original dataset, for each sampling batch we used k-nearest neighbor algorithm to preprocess[16-19] the candidate data sample set, and then calculated the classification result of each sampling batch. We draw lessons from the method of ensemble learning in the sampling process, used homomorphism integrated learning method to categorize the user feature dataset, unified with the random forest algorithm as the classification kernel model for each data batch.

3.2 Ensemble Algorithm

The Random Forest is made up of several decision trees, each decision tree will be full growth, it do not need to cut processing, the more tree it has the more accurate the result will be, and it will not over _tting. The random forest algorithm will do the overall estimate, and it has the advantage of automatic feature selection etc. So we have the following main problems to be solved.

1) Design a bootstrap sampling with replace algorithm for minor class, and then find the k nearest major class Neighbor of the minority class samples, this could be parallel processed by taking the advantage of Spark[20].
2) Build the random forest classifier for each sampling batch, each classifier only process the training dataset

 DOI : 10.18231/2454-9150.2018.1097

that dispatched on the same node, then using the model to predict the testing dataset in parallel, at last collect the prediction results by Spark Driver. To solve the above problems we need to design a global partition algorithm, put the data items which have the same key into same partition, and put the partitions that have to rely on into the same node. The ensemble random forest algorithm has to sampling[21-24] the origin dataset several times, and building random forest is a processing of iteration to promoting for user feature information gain, so we choose to implement the whole algorithm over Apache Spark platform, take the advantage of the Spark's efficient parallel computing to accelerate the calculation of the algorithm.

IV. EXPRESSION PROGRAMMING

4.1 Segmentation

Segmentation is employed to segment the original input data set into a number of smaller data chunks of an equal size. The size of a data chunk is determined by a predefined segmentation ratio. A data chunk consists of a number of data samples. Two segmentation approaches are employed, which are random selection and cutting in sequence. Following the approach presented in [25] which provides a good sampling performance in data coverage, random selection is developed to select data samples from the original input data set and generate a data chunk. Each chromosome in a generation is processed with the same data chunk during the evolution of GEP.

4.2 Overlapping

While segmentation reduces the computation complexity of GEP, processing individual data chunks instead of the whole data set normally degrades the accuracy level of GEP [26]. This is especially true when the data samples have strong correlations.

4.3 GEP Parallelization

The data engineered GEP presented in Section IV-C is further parallelized with an aim to speed up the computation process when dealing with potential big data. The parallel GEP maintains the generation structure in such a way that it processes the chromosomes on a generation basis using a number of CPU cores simultaneously of which each CPU core has two threads.

This fitness evaluator has two versions, one is designed for the *random selection* segmentation scheme without overlapping, whereas the other is designed[27] for the *cutting in sequence* segmentation scheme with overlapping. In the case of *random selection*, the quality of a chromosome is assessed considering the best local fitness value.

The two versions of the P-GEP are significantly faster than the NICHE work. This is mainly due to the fact that P-GEP follows closely the generation structure of GEP leading to an efficient evolution. In addition, processing segmented data chunks further speeds up the computation. P-GEP-*random* is even faster than P-GEP-*overlap* because the less computation overhead incurred in accessing the multiple data chunks.

Model	RMSE at t =				
	1	5	10	60	600
Training Phase					
Single random forest	0.22*	0.28*	0.29	0.33	0.43
Linear regression ensemble	0.18*	0.22*	0.26*	0.34*	0.49*
Neural Network ensemble	0.13*	0.19*	0.23*	0.31*	0.41*
SVR ensemble	0.13*	0.18*	0.20*	0.31*	0.40*
Random forest ensemble	0.15	0.20	0.26	0.32	0.40
CV Phase					
Single random forest	0.23*	0.28*	0.31*	0.35	0.44
Linear regression ensemble	0.21*	0.26*	0.32*	0.38*	0.50*
Neural network ensemble	0.22*	0.29*	0.31*	0.40*	0.45*
SVR ensemble	0.24*	0.29*	0.35*	0.36*	0.42*
Random forest ensemble	0.15	0.22	0.25	0.33	0.40
Test Phase					
Single random forest	0.19*	0.30*	0.31*	0.35	0.46*
Linear regression ensemble	0.24*	0.27*	0.31*	0.42*	0.59*
Neural network ensemble	0.24*	0.28*	0.31*	0.37*	0.50*

Figure 4.1 Phase of Training, CV and Test

For Data Owners: Big data is the foundation for the next wave of productivity resolution: Data Technology (DT). Data owners, such as Facebook, Google, Amazon, Tencent, and Alibaba, collect massive data via the services that they provide [84]. Obviously, via the advancements in big data analytics supported by machine learning and data mining techniques, those datasets produce huge value for those companies. For instance, with the assistance of machine learning and data mining techniques, e-commerce companies are able to push commodities on consumers' wish lists or browsing[28] history. The location-based service providers are able to distinguish the home or work locations for a customer, and provide the best route at the appropriate time. Nonetheless, not all the companies have the ability to collect the demanding data, since collecting huge and comprehensive datasets requires a significant infrastructure investment and long-term efforts. In terms of providing services, stimulating productivity, and maximizing [29] the value of data, the data owners have strong aspirations to trade their own datasets with others.

 DOI : 10.18231/2454-9150.2018.1097

Indicator	Parameters
Price features	
Exponential moving average of the last n observations of best prices	$n = 16$
Bollinger bands of the last n observations of best prices	$n = 32$
Momentum of the best prices over the last n observations	$n = 12,$ and 24
Acceleration of the best prices over the last n observations	$n = 18$
The rate of change of the best prices over the last n observations	$n = 22$
The MACD of the best prices	$f = 12, s = 24$
The relative strength index of the best prices over the last n observations	$n = 20$ and 32
The fast stochastic K of the best prices over the last n observations	$n = 12$ and 18
The Chaikin volatility of the best prices over the last n observations	$n = 10$
The accumulation/distribution line	-
The Chaikin oscillator	$n_1 = 3$, and $n_2 = 10$
Spread features	
Exponential moving average of the last n observations of spread	$n = 10$
Momentum of the spread over the last n observations	18
The rate of change of the spread over the last n observations	$n = 10,$ 16 and 22
The MACD of the spread	$f = 12, s = 30$
The relative strength index of the spread over the last n observations	$n = 14$
The fast stochastic K of the spread over the last n observations	$n = 12$ and 24
Liquidity features	
Exponential moving average of bid/ask book volume over the last n observations	$n = 22$
Exponential moving average of volume at best bid/ask price over the last n observations	$n = 12$ and 36
*Momentum*of bid/ask book volume over the last n observations	$n = 12,$ 24 and 36
Number of price improvements in the last n observations	$n = 25,$ and 50
Number of trades in the last n observations	$n = 50$
Number of bid/ask quotes arrived in the last n observations	$n = 50$
Number of bid/ask cancellations in the last n observations	$n = 50$
Current modal bid/ask price relative to best bid/ask price	-

Figure 4.2 Indicator parameter for all features

For Data Consumers: In high-competition environments, information is the key for a company to discover new business opportunities, values, and customers. Nonetheless, a big challenge is where the consumers can obtain the necessary datasets, since they have no ability to collect the data by themselves. To this end, the data consumers have a strong desire to purchase data [30] from the market, and use those valuable datasets to improve their services or products. As an example, based on sufficient information, manufacturers are able to maximally match the requirements for many different consumers with product differentiation, and service providers are able to re_ne their service plans [10] to improve and target their services to their customers. Thus, data trading is one viable approach to satisfy those needs.

V. SIMULATION

We separately used the SMOTE based algorithms and ensemble random forest algorithm to analysis the business data, then we choose traditional SVM, Logistic Regression and Random Forest algorithms to compare with the ensemble random forest algorithm.

Our experiment was performed on a node Linux cluster. Each node has two 1.35GHz Intel core CPU, 8 GB memory, with Centos 6.5 operation system installed and Apache Spark 1.5 deployed. We extracted more than 500,000 customer purchase behavior data from Company over the past three years. The positive cases are 20787 only accounted for 4.1% of the total data, and the features of the data is highly out of balance. We choose 16 available user features by computing the information gain, and preprocessing the business data by setting segmentation points for discrete values or setting the threshold value for sequence values. We separately used the SMOTE based algorithms and ensemble random forest algorithm to analysis the business data, then we choose traditional SVM, Logistic Regression and Random Forest algorithms to compare with the ensemble random forest algorithm.

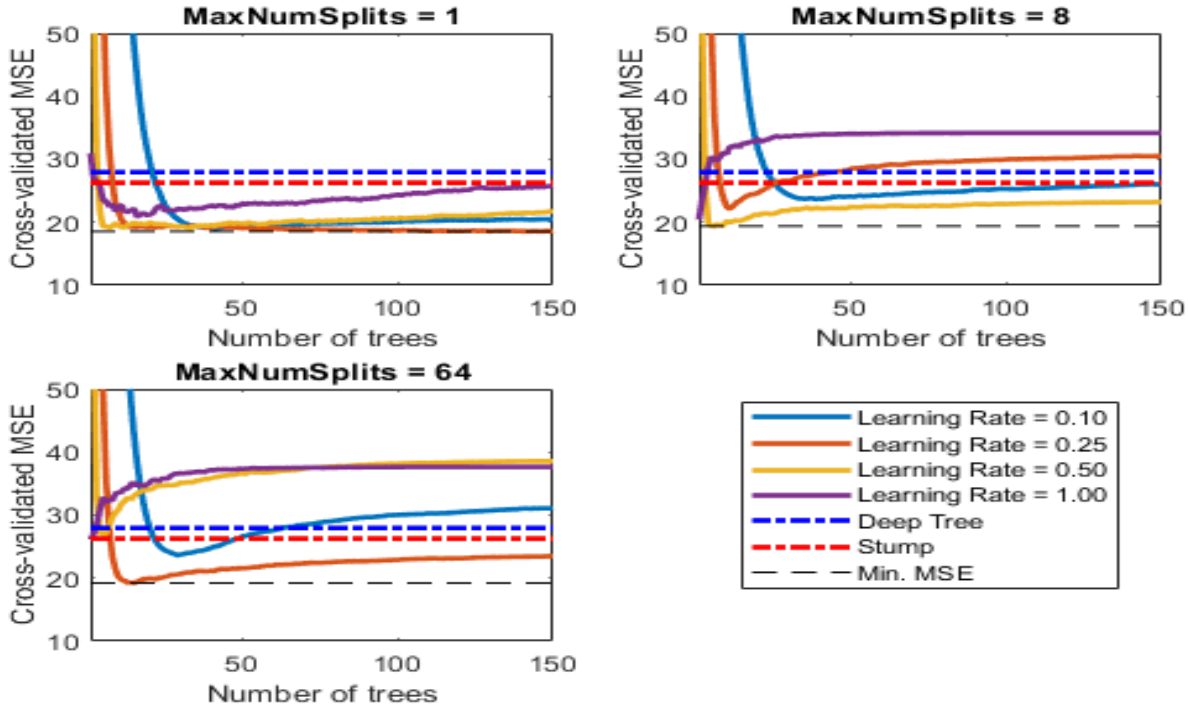
Their running times and evaluated results are shown in table 2, which shows that the ensemble random forest algorithm has higher operation efficiency than most of the strong classification algorithms, and it is suitable for imbalanced classification model. And the table 2 also shows that, strong classifiers to find the decision boundary due with the imbalance distribution of user features, the SMOTE sampling preprocessing is useful but it will spend a longer time to build the correct model on the large number of virtual samples, the ensemble random forest take fewer running time to get more accurate results because the sampling batch is parallel processed by executors, and it's Finally we used the ensemble random forest algorithm to analysis the customer feature data derived from Company, the experiment was performed to choose the potential user by ensemble random forest which compared with the traditional artificial approach, then compared with the real sales data, the error analysis was shows that the recall of traditional approach is only 4%, and it distinguish the purchase ability of the potential users; the ensemble random forest algorithm predicted 60831 potential customers and 24% of them buy the product indeed, the recall reached up to 30.9% within the range of prediction interval between 60% and 90%, the error analysis shows that the ensemble

 DOI : 10.18231/2454-9150.2018.1097

random forest could find the purchase feature of potential users effectively and improve the accuracy of product marketing.

Table 5.1 Construction Techniques for different type of convergence

Construction techniques	Applications	Type of Convergence
First order Taylor expansion	reweighted ℓ_1-norm minimization [15]	stationary point
	robust covariance estimation [22]–[24], [33]	global minimum
	variance component model [34], [35]	objective value
	optimization with projection forms [36]	stationary point
	maximization of a convex objective over a compact set [37]	first order optimal
	sparse eigenvector problem with ℓ_0-norm constraint [19]/penalty [21], [37]	objective value
	edge-preserving regularization in image processing [7], [8], [38]	stationary point (nonconvex objective) global minimum (convex objective)
	ℓ_p-norm minimization [21], [39]–[43]	same as above
Second order Taylor expansion	(sparse) logistic regression [44], [45]	global minimum
	rank constrained matrix quadratic form minimization [46]	objective value
	quartic form minimization [32], [47]	objective value
	sparse linear regression [14], [17], [18], [29], [48], [49]	ℓ_1-norm: global minimum concave regularization: stationary point ℓ_0-norm: local minimum
	nonnegative least squares [50]–[52]	global minimum
Convexity inequality	robust covariance estimation [24]	global minimum
Special inequalities	signomial proramming, complementary GP [53], [54] (arithmetic-geometric mean inequality)	stationary point
	nonnegative least squares [55] (arithmetic-geometric mean inequality plus first order Taylor expansion)	global minimum
	phase retrieval [47], [56], [57] (Cauchy-Schwartz inequality)	stationary point
	sensor network localization [58]–[61] (Cauchy-Schwartz inequality)	stationary point
	variance component model [25], [62], [63] (Schur complement/convexity inequality)	stationary point



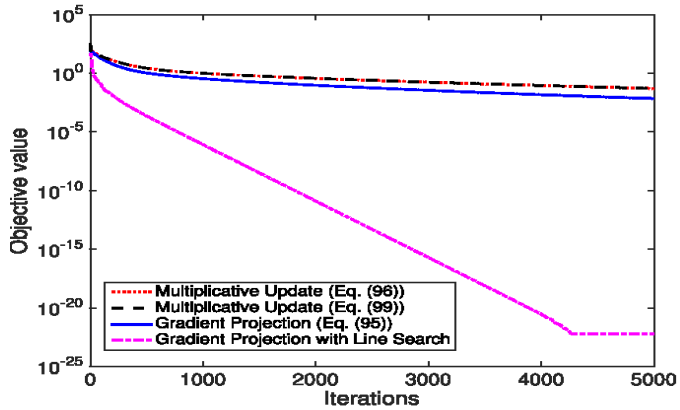


Figure 5.1 Computational scalability for objective value with Iterations

 DOI : 10.18231/2454-9150.2018.1097

We further evaluated the computation scalability of the data engineered GEP in dealing with varied numbers of CPU threads. The execution time of the data engineered GEP decrease when processing 1 TB particle physics data with an increasing number of CPU threads up to 1000. It can be observed that the speedup of parallelization is high when the number of CPUs is less than 100 due to the fact that CPU threads themselves can also cause an additional computation overhead. Parallelization on particle physics data for parallelization on power system data. Fluctuations in performance gain via parallelization. Data samples in the power system data set have a simpler Structure than the data samples in the particle physics data set. As a result, the performance gain achieved via parallelization in processing one unit of power system data using a number of CPU threads is less than the case of processing one unit of particle physics data. When the structure of a data set like the power system data is simple, the performance gain of parallelization can be easily offset by the computation overhead incurred in maintaining these CPU threads. This can be observed the execution time of the data engineered GEP decreases sharply with an increasing number of CPU threads up to 23.

The data engineered GEP reaches the lowest estimated execution time of 5.66E + 013 s when 23 CPU threads participate in the computation. After this point, the execution time goes up due to a high ratio of the overhead incurred in maintaining these CPU threads to the performance gain achieved through parallelization. The fluctuations in performance gain via parallelization can be further observed in a segmentation ratio of 5% was used on the two original data sets. Overall the data engineered GEP achieves a high scalability in dealing with potential big data using a large number of CPU threads. The different types of digital content management, namely software-based DRM, multimedia-based DRM, and unstructured data-based DRM, have been well explored. As we can see, digital management techniques serve as the key method to protecting big data from being stolen and copied. Nonetheless, with the rapid increase of digital content and the trade properties of big data, the feasibility of existing data protection schemes and more advanced techniques should be further investigated.

VI. CONCLUSION

This paper analyzed the imbalance distribution of business data, concluded the preprocessing algorithms of imbalance dataset, proposed an ensemble random forest algorithm based on Apache Spark which can be used in the large scaled imbalanced classification of insurance business data, the experiment result showed that the ensemble random forest algorithm is more suitable in the insurance product recommendation or potential customer analysis than traditional strong classifier like SVM and Logistic Regression etc. The proposed bootstrap under-sampling algorithm combined with the KNN could be used into preprocessing of imbalanced classification algorithms. The ensemble learning algorithms combined with bootstrap sampling preprocessing could reduce the learning process further, and it also has a good reference to other imbalanced data mining algorithms. Although the proposed ensemble random forest algorithm is used to analyze insurance big data in this paper, it can also be applied to big data analytics for Internet of things and mobile Internet. It should be pointed out that for data sets with a high volume in size but a low complexity in data structure, purely increasing

the number of CPU threads could lead to slow executions due to the fact that the overhead incurred in maintaining these CPU threads is higher than the performance gain to be achieved through parallelization. The data engineered GEP can further benefit from the schema theory proposed in our previous work [23] which introduces the concept of building blocks in GEP evolution. A GEP building block is a segment shared by high quality chromosomes in a population which can be discovered during the evolutionary process. Building blocks can be used to replace the corresponding segments of low quality chromosomes for computation speedup in evolution. Therefore, a future work will research how the data engineered GEP can be integrated with building blocks. Then, we reviewed existing works related to big data pricing. With regard to data pricing, we clarified its importance, categorized different market structures, data pricing strategies, and data pricing models, and then listed the advantages and limitations of each category. For the data trading process, we outlined key issues associated with data trading and their possible solutions. We further investigated auction strategies and detailed different schemes, trading platforms, and related issues. Finally, we investigated data protection as the last stage of the big data lifecycle. We categorized existing copyright protection schemes and outlined the challenges of big data copyright protection. Notice that the main purpose of this survey is to provide a clear and deep understanding of big data trading.

REFERENCE

[1] K. W. Bowyer, L. O. Hall, and W. P. Kegelmeye, ``SMOTE: Synthetic minority over-sampling technique," J. Artif. Intell. Res., vol. 16, no. 6, pp. 321_357, 2002

[2] E. Ramentol, Y. Caballero, R. Bello, and F. Herrera, ``SMOTE-RSB: A hybrid preprocessing approach based on oversampling and undersampling for high imbalanced data-sets using SMOTE and rough sets theory," Knowl. Inf. Syst., vol. 33, no. 2, pp. 245_265, 2012.

[3] J. A. Sáez, J. Luengo, J. Stefanowski, and F. Herrera, ``SMOTE-IPF: Addressing the noisy and borderline examples problem in imbalanced classification by a re-sampling method with _ltering," Inf. Sci., vol. 291, pp. 184_203, Jan. 2015.

 DOI : 10.18231/2454-9150.2018.1097

[4] E. Ramentol, N. Verbiest, R. Bello, Y. Caballero, C. Cornelis, and F. Herrera, ``SMOTE-FRST: A new resampling method using fuzzy rough set theory," in Proc. 10th Int. FLINS Conf. Uncertainty Modelling Knowl. Eng. Decision Making, pp. 800_805, 2012.

[5] G. Ping and O. Y. Yuan-You, ``Classification research for unbalanced data based on mixed-sampling," Appl. Res. Comput., vol. 32, no. 2, pp. 379_381, 2015.

[6] Tomek, ``Two modi_cations of CNN," IEEE Trans. Syst., Man Cybern., vol. 6, no. 11, pp. 769_772, Nov. 1976.

[7] M. A. Tahir, J. Kittler, and F. Yan, ``Inverse random under sampling for class imbalance problem and its application to multi-label classi_cation," Pattern Recognit., vol. 45, no. 10, pp. 3738_3750, 2012.

[8] C. Zhou, P. C. Nelson, W. Xiao, and T. M. Tirpak, "Discovery of classification rules by using gene expression programming," in Proc. Int. Conf. Artif. Intell., Las Vegas, NV, USA, Jun. 2002, pp. 1355–1361.

[9] C. Zhou, W. Xiao, T. M. Tirpak, and P. C. Nelson, "Evolving accurate and compact classification rules with gene expression programming," IEEE Trans. Evol. Comput., vol. 7, no. 6, pp. 519–531, Dec. 2003.

[10] M. H. Marghny and I. E. El-Semman, "Extracting logical classification rules with gene expression programming: Microarray case study," in Proc. Int. Conf. Artif. Intell. Mach. Learn. (AIML), Cairo, Egypt, 2005, pp. 11–16.

[11] J. Zuo, C.-J. Tang, C. Li, C.-A. Yuan, and A.-L. Chen, "Time series prediction based on gene expression programming," in Proc. 5th Int. Conf. Adv. Web Age Inf. Manag. (WAIM), vol. 3129. Dalian, China, 2004, pp. 55–64.

[12] V. I. Litvinenko, P. I. Bidyuk, J. N. Bardachov, V. G. Sherstjuk, and A. A. Fefelov, "Combining clonal selection algorithm and gene expression programming for time series prediction," in Proc. 3rd Workshop IEEE Intell. Data Acquisition Adv. Comput. Syst. Technol. Appl., Sofia, Bulgaria, 2005, pp. 133–138.

[13] H. S. Lopes and W. R. Weinert, "A gene expression programming system for time series modeling," in Proc. 25th Iberian Latin Amer. Congr. Comput. Methods Eng. (CILAMCE), Recife, Brazil, 2004, pp. 1–13.

[14] H. S. Lopes and W. R. Weinert, "An enhanced gene expression programming approach for symbolic regression problems," Int. J. Appl. Math.Comput. Sci., vol. 14, no. 3, pp. 375–384, 2004.

[15] Z. Cai, Q. Li, S. Jiang, and L. Zhu, "Symbolic regression based on GEP and its application in predicting amount of gas emitted from coal face," in Proc. Int. Symp. Safety Sci. Technol., 2004, pp. 637–641.

[16] E. Bautu, A. Bautu, and H. Luchian, "Symbolic regression on noisy data with genetic and gene expression programming," in Proc. 7th Int.Symp. Symbolic Numer. Algorithms Sci. Comput. (SYNASC), Timi¸soara, Romania, 2005, pp. 321–324.

[17] L. Teodorescu and Z. Huang, "Enhanced gene expression programming for signal-background discrimination in particle physics," in Proc. 12th Adv. Comput. Anal. Tech. Phys. Res., 2008, pp. 1–11.

[18] L. Teodorescu, "Gene expression programming approach to event selection in high energy physics," IEEE Trans. Nucl. Sci., vol. 53, no. 4, pp. 2221–2227, Aug. 2006.

[19] L. Teodorescu, "High energy physics data analysis with gene expression programming," in Proc. IEEE Nucl. Sci. Symp. Conf. Rec., vol. 1. Fajardo, Puerto Rico, 2005, pp. 143–147.

[20] X. Li, C. Zhou, W. Xiao, and P. C. Nelson, "Prefix gene expression programming," in Proc. Genet. Evol. Comput. Conf. (GECCO), Washington, DC, USA, 2005, pp. 25–31, 2005.

[21] L. Huo, J. Yin, L. Guo, J. Hu, and X. Fan, "Short-term load forecasting based on improved gene expression programming," in Proc. IEEE Int. Conf. Circuits Syst. Commun., Shanghai, China, 2008, pp. 5647–5650.

[22] S. F. Mekhamer, A. Y. Abdelaziz, H. M. Khattab, and M. A. L. Badr, "Gene expression programming for power system static security assessment," Int. J. Eng. Sci. Technol., vol. 4, no. 2, pp. 77–88, 2012.

[23] Z. Huang, "Schema theory for gene expression programming," Ph.D. dissertation, Department of Electronic and Computer Engineering, Brunel Univ. at London, London, U.K., 2014.

[24] C. Zhihua, J. Siwei, Z. Li, and G. Yuanyuan, "A novel algorithm of gene expression programming based on simulated annealing," in Proc. Int. Symp. Intell. Comput. Appl., 2005, pp. 605–610.

[25] M. Snir, S. Otto, S. Huss-Lederman, D. W. Walker, and J. Dongarra, MPI: The Complete Reference, vol. 2. Cambridge, MA, USA: MIT Press, 1996.

[26] X. Du, L. Ding, and L. Jia, "Asynchronous distributed parallel gene expression programming based on estimation of distribution algorithm," in Proc. 4th Int. Conf. Nat. Comput., Jinan, China, 2008, pp. 433–437.

[27] J. Wu et al., "Parallel NICHE gene expression programming based on general multi-core processor," in Proc. Int. Conf. Artif. Intell. Comput. Intell. (AICI), vol. 3. Sanya, China, 2010, pp. 75–79.

[28] J. R. Koza, "Genetic programming as a means for programming computers by natural selection," Stat. Comput., vol. 4, no. 2, pp. 87–112, 1994.

[29] H. Cheng and J. Xue, "The research on evolution schema theorem on gene expression programming," in Emerging Computation and Information Technologies for Education, E. Mao, L. Xu, and W. Tian, Eds. Heidelberg, Germany: Springer, 2012, pp. 399–406.

[30] R. Poli and W. B. Langdon, "Schema theory for genetic programming with one-point crossover and point mutation," Evol. Comput., vol. 6, no. 3, pp. 231–252, 1998.

 DOI : 10.18231/2454-9150.2018.1097