Computers, Materials & Continua
DOI:10.32604/cmc.2021.016188
Article

Tech Science Press

A Data-Semantic-Conflict-Based Multi-Truth Discovery Algorithm for a Programming Site

Haitao Xu[1], Haiwang Zhang[1], Qianqian Li[1], Tao Qin[2,*] and Zhen Zhang[3]

[1]University of Science and Technology Beijing, Beijing, 100083, China
[2]National Computer Network Emergency Response Technical Team/Coordination Center of China, Beijing, 100000, China
[3]Audio Analytic, 2 Quayside, Cambridge, UK
[*]Corresponding Author: Tao Qin. Email: qintao@cert.org.cn
Received: 26 December 2020; Accepted: 17 February 2021

Abstract: With the extensive application of software collaborative development technology, the processing of code data generated in programming scenes has become a research hotspot. In the collaborative programming process, different users can submit code in a distributed way. The consistency of code grammar can be achieved by syntax constraints. However, when different users work on the same code in semantic development programming practices, the development factors of different users will inevitably lead to the problem of data semantic conflict. In this paper, the characteristics of code segment data in a programming scene are considered. The code sequence can be obtained by disassembling the code segment using lexical analysis technology. Combined with a traditional solution of a data conflict problem, the code sequence can be taken as the declared value object in the data conflict resolution problem. Through the similarity analysis of code sequence objects, the concept of the deviation degree between the declared value object and the truth value object is proposed. A multi-truth discovery algorithm, called the multiple truth discovery algorithm based on deviation (MTDD), is proposed. The basic methods, such as Conflict Resolution on Heterogeneous Data, Voting-K, and MTRuths_Greedy, are compared to verify the performance and precision of the proposed MTDD algorithm.

Keywords: Data semantic conflict; multi-truth discovery; programming site

1 Introduction

With the increase in complexity and scale in software development, a conflict between high demand and low efficiency arises. The application of real-time collaborative programming technology and various collaborative programming technologies can enable multiple users to develop and upload software based on their respective collaborative sites [1], which greatly improves the efficiency of software development. However, the programming habits and design ideas of different users are inconsistent, which will inevitably lead to conflicts in data syntax and semantics at a programming site [2–4]. How to find the best code from the conflicting code has become an urgent

need for real-time collaborative programming technology. The problem can be called the "truth discovery problem" in the programming scene.

In this paper, the data semantic conflict problem of multiple users in the programming scene in the function realization of the same code segment is mainly studied. Fig. 1 is taken as an example, in which the implementation of the stacking function in the data structure, in the case of standardized naming of variables, methods, and interfaces in the program, is considered. The program segments submitted by different users have certain differences in code standardization, code robustness, and functional realization, which can be manifested in different recurrences of the functional core code, such as the top pointer self-increment operation in this scenario. The program segments submitted by different users are very different. Based on this, it should be weighed, and the excellent programmers are chosen to submit the high-quality program segment as the standard result. Therefore, the research works focusing on this problem can be considered to be those obtaining high-quality core function codes through truth discovery technology. One must first divide the program segments through lexical analysis technology to obtain the code sequences. Then, the code is treated as an object, and the sequence of the code segment is the object with multiple possible sets of truth values, based on the truth value discovery algorithm. Finally, the truth value discovery results in the problem are determined as a set of high-quality core code sequences.

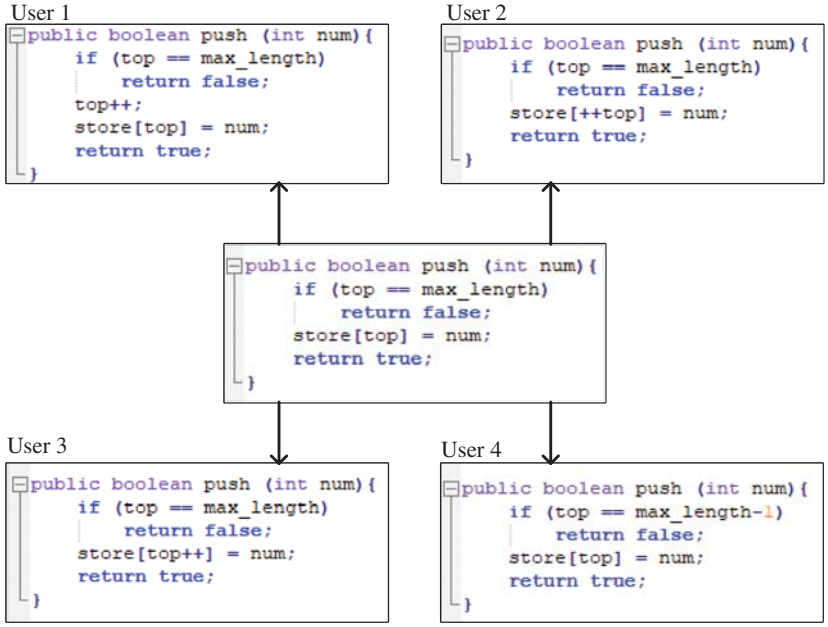

Figure 1: Code semantic conflict scenarios in a programming scene

Recently, numerous research works have appeared that are focused on the truth discovery technology of the data cleaning field in both industry and academia [5–9]. Yin et al. [5] proposed the concept of truth discovery when considering the quality of joint data sources and the object truth value. Dong et al. [6] considered the copy relationships between data sources and proposed a Bayesian method for determining the dependence relationship. Kao et al. [7] considered the authoritative factors of the data source and the reliability of the joint data source. Then, the

accuracy of the data source was optimized by using different probability voting methods. Galland et al. [8] considered the difficulty of object judgment and optimized the definition of the data source credibility. Blanco et al. [9] optimized the quality evaluation problem considering the weight of the data source. Zhao et al. [10] proposed the multi-truth discovery problem, which was based on the probability graph model combining the quality of the data source and the credibility of the declared value, to construct an optimization problem model. Regarding the multi-truth discovery problem, Wang et al. [11] considered the mapping relationships between the data source and value set to construct a multi-truth discovery problem model. Ma et al. [12] proposed an optimization model and a greedy algorithm. Among the above truth discovery algorithms, the calculation models proposed in [5–9] are all aimed at the single truth model and are not suitable for multiple truth discovery problems. In the field of multi-truth discovery research, the algorithm proposed in [10] was based on the assumption that the dataset obeys the beta distribution, and the multiple truth value discovery algorithm was proposed based on optimization. Although the MTRuths algorithm in [12] was proposed to deal with the multi-truth discovery problem based on optimization, the problem model must consider the support between the declared values [13–15].

In conclusion, most of the existing research works are not suitable for the problem of data conflict problems in the programming field. The current research on the multi-truth discovery technology only considers the reliability of the data source and that of the declared value, in which the factor of the support of the declared value is not considered. In fact, different users have different code ages, and the quality of the submitted code is hierarchical. In addition, the programming habits of different users would also lead to differences in code length and fragments. Therefore, the support of the declared value is a factor that cannot be ignored in the multi-truth discovery. In this paper, an attempt is made to solve the problem of multi-truth discovery in a programming scene considering the support of the declared value. The main contributions of this paper are the following.

(1) The characteristics of multi-source code data are combined to construct a multi-truth discovery problem model, and the corresponding optimization problems are proposed.
(2) The deviation degree between the claims based on the support of the claim and the quality of the data source is defined, and the convergence rate of the function is optimized.

The rest of this paper is organized as follows. In Section 2, the multi-truth discovery problems are proposed. Experiments and results are presented in Section 3. Conclusions are presented in Section 4.

2 Multi-Truth Discovery

2.1 Notions and Notations

First, the relevant definitions involved in the multi-truth discovery problem are defined.

Definition 1: **Claim.** The description value of a certain entity attribute from different data sources.

Definition 2: **Data source quality.** The authority of the data source; the higher the quality of the data source, the closer the claim is to the truth.

Definition 3: **Claim deviation.** A measure of the degree of deviation between the claim and the truth.

Definition 4: **Claim support.** When one claim is true, the probability that the other claim is true.

Definition 5: **Multi-truth discovery.** A process of finding multiple truth sets of entity objects from datasets provided by multiple data sources.

All notations used in this paper are listed in Tab. 1.

Table 1: Notation

Notation	Meaning
O	Collection of objects
S	Collection of code data sources
W	Collection of data source quality
A	Collection of declared values
w_n	Quality of W_n
$A_{*,m}$	Collection of declared values of O_m
T_m	Collection of truth of O_m
$A_{n,m}$	Claim of S_n with respect to O_m

2.2 Problem Definition

The problem of multi-truth discovery through the definition of the multi-truth situation can be formulated as follows:

$$\min_{A_{*,m}} F = \sum_{n=1}^{N} w_n \sum_{m=1}^{M} \varphi\left(A_{n,m}\right) \quad s.t.\ T_m \subseteq A_{*,m} \tag{1}$$

The objective function is the weighted sum of the deviation between the declared value of the data source and the standard true value. When the deviation between the obtained true value and standard value of the conflicting dataset reaches the minimum, the obtained truth vector is closest to the standard true value.

In the process of truth discovery, it is generally assumed that if the quality of the data source is high, the probability that the provided claim is true would be high [16]. Then, the quality of the data source providing the claim would be high. In fact, under normal circumstances, the claims provided by multiple data sources are as close as possible to the truth. Based on the existing Conflict Resolution on Heterogeneous Data (CRH) algorithm, the support between heterogeneous claims in the multi-truth problem is considered, and the weighted sum of the deviation of the claims is minimized to find the truth of the entity description.

2.2.1 Data Source Quality

If the similarity distance between the claim of the object provided by a data source and the truth is high, the quality of the data source would be low. Otherwise, one can have a higher quality of the data source. The following formula is used to calculate the data source quality:

$$w_n = -\log\left(\sum_{m=1}^{M} dis\left(A_{n,m}\right) / \left|\mathrm{A}\left(w_n\right)\right|\right). \tag{2}$$

It can be found that the weight of the data source is inversely proportional to the distance between the claim and the truth, the value of which can be calculated by the above logarithmic function.

2.2.2 Deviation

(a) Loss function

In the multi-truth discovery problem for a programming site, first, the data characteristics of the code block are considered, and then the loss function is determined. The declaration values of the data source are collected, and the difference in the length of the declaration values provided by different data sources is considered. A formula is then defined to calculate the offset distance as follows:

$$dis\left(A_{n,m}, T_m\right) = \left| \frac{A_{n,m} \cap T_m}{T_m} \right|. \tag{3}$$

(b) Claim support

In the process of collaborative programming, the code data submitted by different users are different in code quantity and quality. Then, it is necessary to use the asymmetric support calculation method to calculate the support of the declaration value in the multi-truth case, as given in the following equation:

$$\sup(A_{n,m}, A_{n',m}) = \frac{1 - dis(A_{n,m}, A_{n',m})}{len A_{n,m}}. \tag{4}$$

(c) Claim deviation

In the collaborative programming environment, it is necessary to combine the claim support and loss function to calculate the deviation, and the formula is given as follows:

$$\varphi\left(A_{n,m}\right) = \frac{\sum_{A_{i,m} \in A_{*,m}, i \neq n}^{N} dis\left(A_{n,m}, T_m\right) / sup\left(A_{i,m}, A_{n,m}\right)}{N - 1}. \tag{5}$$

2.3 Multi-Truth Discovery

Assuming that the concept of high cohesion and low coupling is strictly followed in the process of software collaborative programming, the different code segments are independent of each other. Then, the objective function corresponding to each object can be converted as follows:

$$\min_{T_m} \ \psi(m) = \sum_{n=1}^{N} w_n \cdot \varphi\left(A_{n,m}\right). \tag{6}$$

In the multi-truth discovery problem, the quality of the data source is determined by the deviation of the claim provided by it. The degree of support between the claims in the definition of the degree of deviation is fixed. Then, the key to solve the optimization model is to refer to the truth. In this paper, the strategy of reference truth selection is based on the enumeration method. When the possible set of objects exceeds a certain threshold, the enumeration method will not meet the needs of real-time truth discovery. Therefore, in the iterative process, the declared value that minimizes the value is selected as the reference true value for subsequent iterations.

Algorithm 1: Multiple truth discovery algorithm based on deviation (MTDD)

Input: $A=\left\{A_{\mathrm{n},m}\right\}_{n=1\ m=1}^{N\ \ M}$, $S=\{S_n\}_{n=1}^{N}$, $P=\{P_m\}_{m=1}^{M}$.
Output: Set of object truth $T=\{T_m\}_{m=1}^{M}$.
1. Initializes the data source quality $W_n(0), n=1,2,\ldots,N$;
2. For each $A_{n,m}$ **do**
3. To calculate $sup\left(A_{n,m},A_{i,m}\right)$ according to formula (4);
4. **end for**
5. **do**
6. Obtain a possible set of truth values through the greedy algorithm;
7. **for** each P_m **do**
8. **for** each $A_{n,m}$ **do**
9. To calculate $\varphi\left(A_{n,m}\right)$ according to formula (5);
10. **end for**
11. To calculate $\psi(m)$ according to formula (6);
12. **end for**
13. Returns the claim that minimizes $\psi(m)$ as T_m
14. **end if**
15. For each $S_n \in S$ **do**
16. To calculate w_n according to formula (2);
17. **end for**
18. **until** Convergence
19. **return** $T=\{T_m\}_{m=1}^{M}$

3 Experiments

In this section, the proposed method is compared with the existing multi-truth methods from the following three aspects [17].

- Precision: Ratio of the truth set returned by the algorithm to the standard set:

$$P=\frac{|T\cap t|}{|T|}\times 100\%. \tag{7}$$

- Recall rate: Ratio of correct truth values in the standard set to truth values returned by the algorithm:

$$R=\frac{|T\cap t|}{|t|}\times 100\%. \tag{8}$$

- F-score: Harmonic average of the precision and recall rate:

$$F_1=\frac{2PR}{P+R}, \tag{9}$$

where T is the standard truth value set and t is the predicted truth value set.

3.1 Experimental Setup

3.1.1 Baselines

- Voting-K: For the multi-truth case, Voting-K selects the declared value as the true value when the voting proportion exceeds the K value.
- CRH: For the multi-truth case, the CRH algorithm is an algorithm based on the probability distribution.
- MTRuths_Greedy: The MTRuths algorithm is a truth discovery algorithm calculated through greedy.
- MTDD: A multi-truth discovery based on the deviation degree; see Section 2 for details.

3.1.2 Datasets

BOOK: Taking the data characteristics in the programming site into account, the BOOK dataset [2] is used as the experimental dataset. In the BOOK dataset, each object contains the book title, ISBN, author list, and data source. After cleaning the dataset, a dataset containing 877 book websites, 1,254 books, and 24,221 author name records is obtained. Each author is used as a declared value object, and the author name record is used as a multi-truth set. The truth set provided in the literature is used as the standard set, with the cleaned book author dataset as the test set.

MOVIE: The collected data of 2,000 movies are used as a dataset, sourced from 10 different video sites, including Tencent Video, iQiyi Video, and Douban. the MOVIE dataset contains 23,968 different director names and 11,365 movie entities. The dataset is processed in the same way as the BOOK dataset, and the processed dataset is used as the test set, and 100 sample instances are randomly selected and labeled as the standard set.

3.1.3 Environment

All the experiments are implemented in an environment with a Intel® Core™i5-7300HQ CPU@2.50 GHz processor, with 12 GB of RAM running the Windows 10 operating system. All methods in this paper are implemented in Python, with Python 3.6 as the development environment and MySQL 5.6.42 as the database.

3.2 Effectiveness

3.2.1 Accuracy Assessment

For the multi-truth problem, the corresponding adjustments are made to the baseline methods. For the Voting-K method, the threshold K is set, and all attribute values in which the voting proportion exceeds the K value are considered the true values. In this experiment, the precision, recall rate, and F1-score of Voting-K, MTRuths_Greedy, CRH, and MTDD are compared for the virtual dataset. The analysis results are shown in Tab. 2.

Fig. 2 shows that the Voting algorithm judges the true value according to the proportion of data sources, which provide the declared value of the object. As K increases, its accuracy rate increases and the recall rate decreases. The CRH and MTRuths_Greedy algorithms have better F1 scores. Among them, the CRH algorithm solves the multi-truth problem based on the probability distribution of the declared value, which is greatly affected by the distribution of the declared value in the dataset. The MTRuths_Greedy algorithm uses a weighted voting method to calculate the probability of the initial declared value and solves the optimal truth set by the greedy algorithm. However, it is easy for the results to fall into the local optimum, which would cause the algorithm to terminate prematurely and obtain incomplete results. Different from these

methods, the proposed MTDD algorithm considers the long-tail characteristics of conflicting data in a multi-source environment, uses an asymmetric distance measurement function, and introduces support between declared values to define deviation variables. The proposed algorithm is not easily affected by the local optima. The misleading of low-quality data sources has a high accuracy and recall rate.

Table 2: Accuracy for the BOOK dataset

Methods	Performance		
	Precision	Recall	F1
MTRuths_Greedy	0.8548	0.8333	0.8439
CRH	0.8441	0.8172	0.8304
Voting-50%	0.8602	0.6452	0.7373
Voting-70%	0.9247	0.4032	0.5615
MTDD	0.8565	0.8217	0.8387

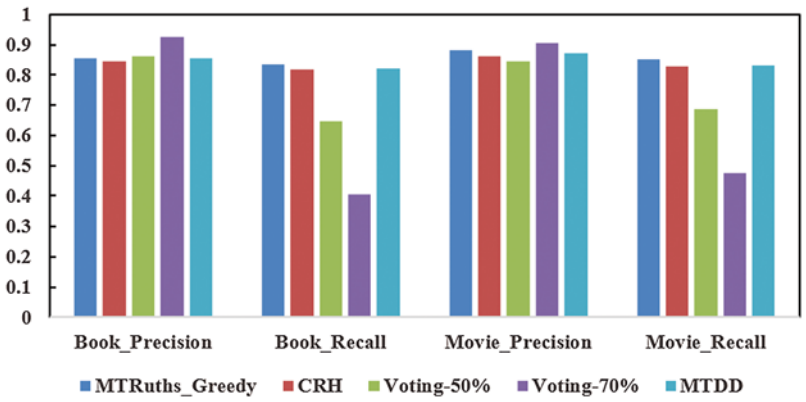

Figure 2: Accuracy for the BOOK and MOVIE datasets

3.2.2 Efficiency Evaluation

The algorithm time of the Voting-50%, MTRuths_Greedy, CRH, and MTDD algorithms was compared under the same dataset scale, as given in Tab. 3.

As shown in Fig. 3, the Voting algorithm uses a voting mechanism to select the truth value, which does not require iteration and has the least time complexity. It can be seen that the runtime of the Voting algorithm is the shortest. MTRuths_Greedy uses the greedy algorithm for truth selection, with a lower time complexity and shorter runtime. The proposed MTDD algorithm adopts the enumeration method to select the true value, which has the highest time complexity and a relatively long runtime.

The convergence conditions of the algorithms are the following: the quality vector cosine similarity of the data source is obtained from the second iteration, which is used to measure the change in the results of the second iteration. If the similarity is higher, the change would be smaller. When the change reaches a certain threshold, the iteration stops.

Table 3: Runtime

Method	Runtime (s)
MTruths_Greedy	5.83
CRH	6.12
Voting-50%	1.47
MTDD	13.54

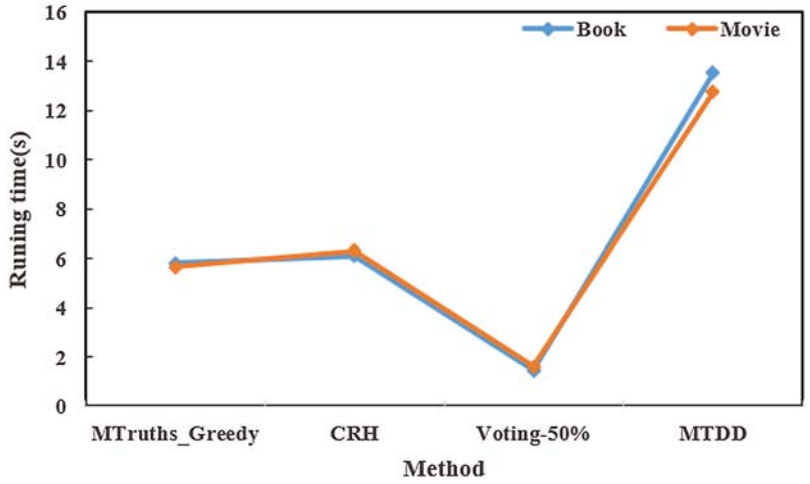

Figure 3: Runtime

It can be seen from Fig. 4 that the proposed MTDD algorithm converges quickly for both datasets; that is, the convergence condition can be satisfied after five iterations.

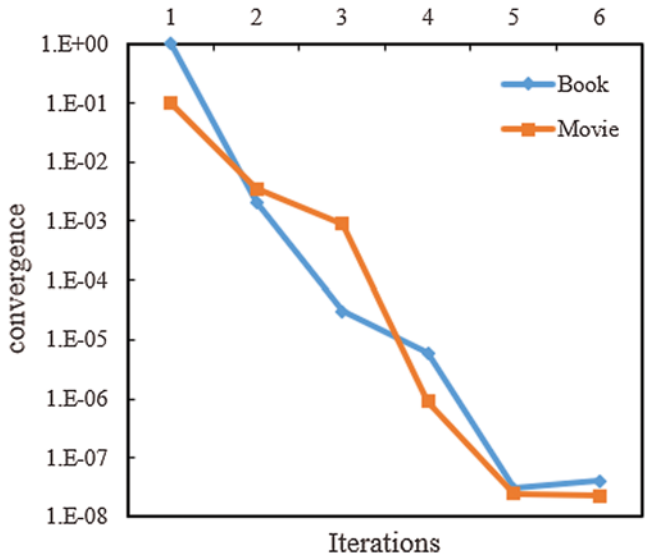

Figure 4: Convergence rate of iterations

4 Conclusions

In the process of software online collaborative development, several challenges must be solved that are brought about by the large-scale code data of a programming site. The code data submitted by different users will have semantic inconsistencies; that is, data semantic conflicts. According to the data characteristics of the code segment, the problem is defined as a multi-truth discovery problem. The MTDD algorithm is then proposed to convert the multi-truth discovery problem into an optimization problem. The truth value set obtained should minimize the weighted deviation from different object sets. The support between different declared values and data is considered in the process of calculating the truth value. The optimal solution of the truth value is obtained through an optimized method. This method is slightly better than the existing multi-truth discovery methods in terms of accuracy and has good performance in convergence.

Funding Statement: This work is supported by the National Key R&D Program of China (No. 2018YFB1003905) and the National Natural Science Foundation of China under Grant (No. 61971032), Fundamental Research Funds for the Central Universities (No. FRF-TP-18-008A3).

Conflicts of Interest: The authors declare that they have no conflicts of interest to report regarding the present study.

References

[1] H. F. Fan and C. Z. Sun, "Supporting semantic conflict prevention in real-time collaborative programming environments," *ACM Sigapp Applied Computing Review*, vol. 12, no. 2, pp. 39–52, 2012.

[2] A. Alhroob, W. Alzyadat, A. T. Imam and G. M. Jaradat, "The genetic algorithm and binary search technique in the program path coverage for improving software testing using big data," *Intelligent Automation & Soft Computing*, vol. 26, no. 4, pp. 725–733, 2020.

[3] A. Mirarab, S. L. Mirtaheri and S. A. Asghari, "A model to create organizational value with big data analytics," *Computer Systems Science and Engineering*, vol. 35, no. 2, pp. 69–79, 2020.

[4] B. Abdullah, H. Daowd and S. Mallappa, "Semantic analysis techniques using twitter datasets on big data: Comparative analysis study," *Computer Systems Science and Engineering*, vol. 35, no. 6, pp. 495–512, 2020.

[5] X. X. Yin, J. W. Han and P. S. Yu, "Truth discovery with multiple conflicting information providers on the web," *IEEE Transactions on Knowledge and Data Engineering*, vol. 20, no. 6, pp. 796–808, 2008.

[6] L. D. Xin, L. Berti-Equille and D. Srivastava, "Integrating conflicting data: The role of source dependence," *Proc. of the VLDB Endowment*, vol. 2, no. 1, pp. 550–561, 2009.

[7] M. J. Kao, W. Zhang and H. Gao, "Truth discovery algorithm in conflicting data," *Journal of Computer Research and Development*, vol. 47, pp. 188–192, 2010.

[8] A. Galland, S. Abiteboul, A. Marian and P. Senellart, "Corroborating information from disagreeing views," in *Proc. of the 3rd ACM Int. Conf. on Web Search and Data Mining*, New York, USA, ACM PUB27, pp. 131–140, 2010.

[9] L. Blanco, V. Crescenzi, P. Merialdo and P. Papotti, "Probabilistic models to reconcile complex data from inaccurate data sources," in *Proc. of the 22nd Int. Conf. on Advanced Information Systems Engineering*, Berlin, Springer-Verlag, pp. 83–97, 2010.

[10] Z. Zhao, J. Cheng and W. Ng, "Truth discovery in data streams: A single-pass probabilistic approach," in *CIKM'14 Proc. of the 23rd ACM Int. Conf. on Conf. on Information and Knowledge Management*, pp. 1589–1598, 2014.

[11] X. Z. Wang, Q. Z. Sheng, X. S. Fang, L. N. Yao, X. F. Xu *et al.,* "An integrated bayesian approach for effective multi-truth discovery," in *CIKM '15 Proc. of the 24th ACM Int. on Conf. on Information and Knowledge Management*, pp. 493–502, 2015.

[12] R. X. Ma, W. Wang, X. F. Meng and Y. J. Shi, "MTruths: Multi-true value discovery method for web information," *Journal of Computer Research and Development*, vol. 52, no. 12, pp. 2858–2866, 2016.

[13] H. L. Li, C. C. Zhou, H. T. Xu, X. Lv and Z. Han, "Joint optimization strategy of computation offloading and resource allocation in multi-access edge computing environment," *IEEE Transactions on Vehicular Technology*, vol. 69, no. 9, pp. 10214–10226, 2020.

[14] C. Caifeng, X. Sun, L. Deshu and T. Yiliu, "Research on efficient seismic data acquisition methods based on sparsity constraint," *Computers, Materials & Continua*, vol. 64, no. 1, pp. 651–664, 2020.

[15] C. Wu, V. Lee and M. E. McMurtrey, "Knowledge composition and its influence on new product development performance in the big data environment," *Computers, Materials & Continua*, vol. 60, no. 1, pp. 365–378, 2019.

[16] D. Yao and Y. Chen, "Design and implementation of log data analysis management system based on hadoop," *Journal of Information Hiding and Privacy Protection*, vol. 2, no. 2, pp. 1–7, 2020.

[17] F. Bingxu, "Design and analysis of a rural accurate poverty alleviation platform based on big data," *Intelligent Automation & Soft Computing*, vol. 26, no. 3, pp. 549–555, 2020.