INTERNATIONAL JOURNAL OF MATHEMATICAL MODELS AND METHODS IN APPLIED SCIENCES
DOI: 10.46300/9101.2022.16.19 Volume 16, 2022

A Long Command Subsequence Algorithm for Manufacturing Industry Recommendation System with Similarity Connection Technology

Siyu Huang[1,2,3], Xueyan Huang[4], Taisheng Zeng[1,2,3], Danlin Cai[1,2,3], Daxin Zhu[1,2,3*]

[1] School of Mathematics and Computer Science, Quanzhou Normal University, Quanzhou 362000, China

[2] Fujian Provincial Key Laboratory of Data Intensive Computing, Quanzhou 362000, China

[3] Key Laboratory of Intelligent Computing and Information Processing, Fujian Province University, Quanzhou 362000, China

[4] School of Educational Science, Quanzhou Normal University, Quanzhou 362000, China

*Corresponding authors:Daxin Zhu

***Abstract*—Manufacturing industry requires a unique recommendation system to suggest products and raw materials, but its performance is often poor in massive data environment. In order to solve the similarity connection problem of large-scale real-time data, the optimized incremental similarity connection method which is used to deal with streaming data can concisely obtain the longest common additive sequence of two given input sequences. This paper, on the basis of the recursion equation, applies a very simple linear space algorithm to solve this problem and adopts new states to carry out similarity connection of incremental data. The experimental results demonstrate that this method can not only ensure the accuracy of real-time recommendation system but also greatly reduce the computed amount**.

***Keywords*—Long command subsequence, Similarity connection, Recommendation system, Real-time, Manufacturing industry**

I. INTRODUCTION

With the rapid development and popularization of Internet technology, increasing volume of information resources begin to appear on the Internet, because of which, users find it more and more difficult to achieve specific information they need. Information resources can not be achieved by users who need them in time and some of them can not be found, which waste a lot of information resources. At the same time, users have to invest much time and energy to search for information resources they need. Such phenomenon with booming information resources but low utilization rate of information resources is known as 'information overload'. This problem can be traditionally solved by using search engines, but search engines do not take users' personalized demands into consideration. Different people search for information through the same keywords but they only achieve the same results. Therefore, the problem of information overload is still not well solved. In order to solve this problem, this paper puts forward the method personalized service technology which aims to provide different information resources for different users. By counting users' own information, this technology can find differences between users and their own characteristics. Besides, it can provide different users with targeted, differentiated services according to the above features.

At present, many scholars have done a lot of researches on personalized services whose core point is personalized recommendation system. Therefore, recommendation system is an important part of the research work of personalized service technology. Recommendation algorithm plays the most important role in the recommendation system [1] and the advantages and disadvantages of algorithm affect the recommendation performance of the system.

Some existing recommendation systems can be divided as follows: recommendation system [2] based on knowledge contents, collaborative filtering recommendation system and mixed recommendation system. Many manufacturing industry system have fully appreciated the benefits of the recommendation system, but the recommendation system is also facing severe security problems. Due to the openness of recommendation system itself and its sensitivity to user information, people begin to pay more and more attention to the trusted recommendation mechanism of collaborative filtration recommendation system, making it gradually a hot research topic.

Similarity connection is used to find all data groups whose similar values are greater than the threshold given by users from a given data set under the specified similarity function measure. Similarity connection has been widely used in many fields. For example, in geography, similarity connection can be used to detect the collision or proximity of geographical features, such

as landmarks, houses and roads. In the application of medical imaging, similarity connection can be used to detect whether the direct distances of some cancer cells are less than a certain threshold. In the recommendation system, similarity connection can be used to compare the similarity of recommended items. However, if the similarity connection is used to unordered and unindexed data sets, it will cost a lot in computation.

Because the scale of data grows very fast in the recent years, similarity connection is a bottleneck which blocks its development to a larger scale. The inquiry done through similarity connection of mass data means that all groups of similar objects can be searched from the mass data sources. This is a basic operation in similarity connection's dealing with mass data, so it attracts wide attention.

Similarity connection query of data can be divided into the following three main categories according to different classification standards: (1) In terms of different definitions of "similarity", it can be divided into threshold join query and Top-k join query; (2) In term of the difference of the number of data sources, it can be divided into single-source (self-join)query, double-source query and multi-source join query. Single-source join query means that all objects of similar two-tuples found come from the same data source. (3) In term of the difference of data types, it can be divided into set, character string, vector, graph and other join queries. Join queries of dataset aim to deal with data sets. Similarity connection queries of different categories, such as threshold connection query, can be interlocked. It can be single source connection query or dual source connection query.

There are still many problems on new characteristics of data development and application development that have not been studied yet. For example: (1) high-dimensional data similarity connection technology. Similarity connection query of high-dimensional data faces great challenges due to the existence of dimension disasters. The traditional query algorithm with index structure as the basis does not work now. Similarity connection query of high-dimensional data will be basic operations of many data mining and machine learning tasks. (2) Massive online real-time similarity connection technology. At present, Map-Reduce framework featured by good extensibility, fault tolerance and usability is used to perform the similarity connection of mass data due. However, as a batch processing model, Map-Reduce is not suitable for real-time data processing. The online real-time processing of similarity connection of mass data needs to be further studied. In order to solve these two problems, we should think about whether we can (1) use piece-wise cumulative approximation method to reduce the high-dimensional data to a low-dimensional space so as to overcome the shortcoming of the rapid performance decline caused by the increase of dimensions and carry out effective filtering at a lower cost. Parallel join query algorithm can be used to process large-scale high-dimensional data? (2) Whether data processed through dimension reduction filtration can be further divided into several subspaces according to certain rules to and prepare for streaming data processing in memory in the next step? (3) Can the reverse index list of character strings be regarded as states to iteratively process the incremental character string of streaming data? Can the new states be used to carry out similarity connection for the incremental data and to process online data in real time rapidly within limited memories?.

II. Progress of Related Works

By calculating the unilateral function of multiple sets and their conjunctive functions. In order to calculate the unilateral function, we only need toA lot of innovative achievements have been made in terms of similarity connection query technology. However, there is no unified definition of similarity connection query in domestic and foreign literature. For example, Shim et al. define spatial similarity connection query as follows (1) Self connection: Given a set consisting of high-dimensional points and a distance metric, you should find all point pairs whose distances do not exceed the threshold from this set. (2) Non-self-connection: Given two sets of points consisting of higher dimensional points and a distance metric, you should find all pairs of points that meet the following conditions: two points come from each of these sets and the distance does not exceed the threshold[4]. Xiao et al. put define Top-k similarity connection query: Given two sets and similarity functions composed of records, you should find the most similar record pairs from these two sets[5].

There are also some research results on the similarity connection of massive data. Ma et al. put forward a method based on Map-Reduce and filtering technology, that is, set Jaccard-Similarity coefficient is used to complete single-source or double-source record join query based on set similarity[6]. In the paper, Map-Reduce framework is used for the first time to solve the similarity connection query problem of massive set data and the following two operations are done to improve the algorithm efficiency: (1) Prefix filtering technology is used in Map phase and length, position and suffix filtering technologies are used in Reduce phase, which reduces the number of copies and computing times. (2)All collection elements are sorted in ascending order according to word frequency, which can not only enhance the effect of prefix filtering, but also ensure the load balance of Reduce to a certain extent. Rafiei et al. proposes several general algorithms based on Map-Reduce[7]. That is to say, the set Jaccard-Similarity coefficient, hamming distance function or character string editing distance function are used to complete the similarity connection queries of single source collection, bit character string or character string. Original data are divided into several groups by means of replicas; then some results obtained in each group will be processed in parallel and all results will be aggregated to get the final result. At the same time, there shall be as few copies as possible and shall be no duplicate calculation and no duplicate output results. Vernica et al. proposes an iterative division method based on Map-Reduce and completes the similarity connection query of single-source or double source metric space[8].

This paper carries out iterative division of original data sets so that the data sets obtained can be as small as possible. Then, an efficient memory similarity connection algorithm is

used to process them so as to obtain partial results. Finally, partial results are summarized to obtain the final result. Albeanu et al. put forward that a two-step method based on Map-Reduce and filtering technology shall be used to complete the similarity connection query of single source multiple collections, collections, character strings or vectors[9]. The basic idea is that the similarity calculation of multiple set pairs can be obtained b scan multiple sets and in order to calculate conjunctive function, we need to scan the intersection. Therefore, we need to calculate the unilateral function of multiple sets and their conjunctive function before we calculate their similarity and obtain the final result. Silva et al. put forward the Map-Reduce algorithm, which uses Euclidean distance function to complete the query of single-source vector Top-k similarity connection[10].

The fundamental concept is as follows: copies shall be used to divide the original data sets into different groups. The Top-k similar vector pairs of each group will be then counted. After that, the results of all groups shall be aggregated and ranked to obtain top-k similar vector pairs of the original data sets.

III. METHODOLOGY

A Related definitions

There are many types of similarity connection. According to data types of objects connected, it can be divided into such types as similarity connection of character strings, set similarity connection and graph similarity connection. In recommendation system experiments studied in this paper, recommendation and ranking are mainly implemented according to the character string matching method.

In order to simplify them, this paper ignores the consistency between contents and topics and adopts the character string matching size of the keywords entered by the user and the index keywords.

Definition 1: Definition of editing distance of character string $S1$ and character string $S2$: the minimum number of single-character editing operations needed to convert $S1$ to $S2$ is denoted as $ED(S1,S2)$.

Definition 2: Standardized editing distance:

$$\frac{ED(S1,S2)}{MAX\left(|S1|,|S2|\right)} \tag{1}$$

Definition 3: Editing similarity

$$1-\frac{ED(S1,S2)}{MAX\left(|S1|,|S2|\right)} \tag{2}$$

Two data sets, $D1$ and $D2$, which are respectively composed of character strings, use the editing similarity based on editing distance to measure the character string similarity. The similarity connection query of $D1$ and $D2$ aims to find out all character string pairs which can meet the following conditions: character strings come from $D1$ and $D2$ respectively and their similarity is not less than the threshold E.

We often use filter-verification framework in practice. However, this framework has several shortcomings: (1) It can not efficiently process short character strings (character strings with an average length of less than 30); (2)the algorithm has low processing efficiency if the data sets are dynamically updated; (3) many indexing operations need to be done [11].

B Problems and Bottlenecks

Due to the retrieval difficulties caused by the continuous growth of similarity connection of mass data and the high retrieval consumption caused by the disordered and unindexed data sets, researches on similarity connection have become more and more important. Although traditional methods can effectively carry out the similarity connection of character strings, the continuously updated state need to be preserved during the execution, which takes a lot of storage space in practical applications, especially in the age of big data.

Various enterprises in the manufacturing industry will spend more and more time and energy using this method as time goes by. In addition to the rapid growth of data size, data models become increasingly complex and dense in practical applications. What is worse, the improvement of data dimensions also makes calculation more complex. In addition, existing studies mainly focus on the disk-based similarity connection algorithm, which lacks effectiveness and scalability in memory connection calculation. Because of the urgent demand for processing real-time similarity connection of mass data, the similarity connection is still a bottleneck in many scientific applications.

In order to meet the growing application demands and to solve problems effectively, two problems still exist in the effective similarity connection in the high-dimensional, massive and real-time data.

If the m-dimensional mapping distance of D-dimensional vectors v_1 and v_2 is greater than $K\varepsilon$, the probability that the Euclidean distance of v_1 and v_2 is greater than ε is greater than $1-p$. Based on these research findings, the high-dimensional vectors v_1 and v_2 can be mapped to the low-dimensional space and be filtered by calculating the distance of low-dimensional space. In this way, the calculation cost can be reduced. This problem can be solved by the following two steps[12,13,14].

A new dimension reduction method based on the piecewise cumulative approximation method in time series can be used to reduce the high-dimensional data to the low-dimensional space dimension. Besides, the distance of the low-dimensional space is the lower bound of the original distance.

On the basis of the previous research, this paper finds an effective distance space to calculate the distance between two vectors in the m-dimensional space. If $\Delta m(v1,v2) > k\varepsilon$, $(v1,v2)$ will be filtered. In this way, we can effectively filter

them at a relatively low cost, which greatly reduce the computational cost[15,16].

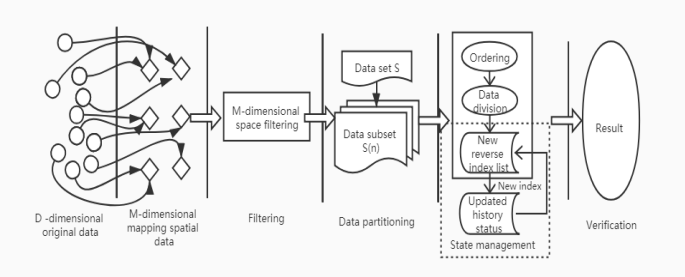

Figure 1: Data dimension reduction and filtering

As shown in Figure 1, dimension reduction can be carried out first. m random vectors $a_1, a_2, ...a_m$ are used to do dot product operation with the original D-dimension vector Y so as to reduce the vector V from D-dimension space to M-dimension space. Filter processing will be then done after that. In this stage, the distance between two vectors is calculated in m dimensional space. If $\Delta m(v_1, v_2) > k\varepsilon$, (v_1, v_2) will be filtered. Finally, the distance of the candidate pairs will be calculated so as to make final judgment accordingly.

With the rapid development of Internet and mobile technology, there have been more and more online application systems. Big data brings challenges to existing application systems and streaming data make it urgent to improve the batch processing methods and technologies. However, the existing character string similarity connection algorithms are all based on the space of limited memory, which requires that the data must be read into the memory at one time. In this era of big data, this method is not feasible.

This paper will adopt an incremental character string similarity connection method based on memory computation to process streaming data. It takes the reverse index list of character strings as the state to iteratively update states obtained in our processing historical data. Besides, new states are also used to carry out similarity connection for incremental data. The state consists of $n+1$ substrings of a character string and the character strings that contain the current substring saved after each substring[12,17,18].

IV. LCS-K SIMILARITY CONNECTION TECHNOLOGY

This paper adopts a space efficient algorithm for the longest common subsequence in k -length substrings. The longest common subsequence (LCS) problem is a classic problem in computer science. Given two sequences A and B, the longest common subsequence (LCS) problem is to find a subsequence of A and B whose length is the longest among all common subsequences of the two given sequences. The problem has numerous applications in many apparently unrelated fields ranging from file comparison, pattern matching, computational biology, etc.[12,13,19,20].

Definition 4. Given two sequences $A = a_1a_2...a_n$ and $B = b_1b_2...b_m$, and an integer k, the LCS-k problem is to find the maximal length l such that there are l substrings, $a_{i1}...a_{i1+k-1}, \cdots, a_{il}...a_{il+k-1}$, identical to $b_{j1}...b_{j1+k-1}, \cdots, b_{jl}...b_{jl+k-1}$ where $\{a_{it}\}$ and $\{b_{jt}\}$ are in increasing order for $1 \le t \le l$ and any two k-length substrings in the same sequence, do not overlap[21,22].

A similar problem is the LCS at least k problem (LCS$\ge k$). In this problem, the demand of matching substrings of length exactly k is relaxed to the length of the matched substrings to be at least k. The length of the common substrings is further limited by $2k-1$ since a longer common substring contains two substrings each of length k or more.

LCS-k can be solved by using a dynamic programming algorithm. Let $d(i, j)$ denote the length of the longest match between the prefixes of $A[1:i] = a_1a_2...a_i$ and $B[1:j] = b_1b_2...b_j$. Then, $d(i, j)$ can be computed recursively as follows.

$$d(i, j) = \begin{cases} 1 + d(i-1, j-1) & if\ a_i - b_j \\ 0 & otherwise \end{cases} \quad (3)$$

Let f (i, j) denote the number of k matchings in the longest common subsequence, consisting of k matchings in the prefixes A[1:i] and B[1:j]. Then, f (i, j) can be computed recursively as follows.

$$f(i, j) = \max \begin{cases} f(i-1, j) \\ f(i, j-1) \\ f(i-k, j-k) + \delta(d(i, j)) \end{cases} \quad (4)$$

Where, δ is a simple piecewise linear function defined by:

$$\delta(i) = \begin{cases} 1 & \text{if } i \ge k \\ 0 & otherwise \end{cases} \quad (5)$$

Based on the Equation (2), the table f(i, j) for the given input sequences $A = a_1a_2...a_n$ and $B = b_1b_2...b_m$ of size n and m respectively, can be computed in $O(mn)$ time and $O(mn)$ space by a standard dynamic programming algorithm.

Algorithm 1: LCS-k

Input: A, B

Output: f (i, j), the number of k matchings in the longest common subsequence of A and B

```
for i=1 to n do
  for j=1 to m do
if ai = bj then d(i, j) ← 1 + d(i−1, j−1);
f (i, j) ← max{f (i−1, j), f (i, j−1), f (i − k, j − k) + δ(d(i, j))};
end
end
```

return f (n,m)

By adopting the optimized incremental similarity connection method based on memory computation optimized above to process streaming data, we can obtain the longest common additive sequence of two given input sequences easily. On the basis of the recursive equation, this paper uses a very simple linear space algorithm to solve this problem and adopts a new state to carry out similarity connection of incremental data.

V. Performance Analysis and Results

The experiment uses the Fabric Data Sets collected in large textile factories, and such data can be used for fabric recommendation. There are 972 kinds of classification in the data set, 486 kinds of training set are divided semi randomly, and the remaining 486 kinds are used as the test set. This paper compared the content-based recommendation method BMR, the recommendation method BFR based on collaborative filtering, the improved similarity connection technology recommendation method TBRR[13] and the LCS similarity connection technology recommendation method LSCR proposed in this paper.

Besides, the Fabric Data Sets also compared the three indexes of Recall rate (RECALL), precision (PRECISION), mean average error (MAE), these are the recommended systems to measure the accuracy of the forecast.

Equation 4 is the recalculation method of RECALL.

$$RECALL = \frac{\sum_{u \in U} |R(u) \bigcap T(u)|}{\sum_{u \in U} |T(u)|} \tag{6}$$

Equation 5 is the calculation method of PRECISION.

$$PRECISION = \frac{\sum_{u \in U} |R(u) \bigcap T(u)|}{\sum_{u \in U} |R(u)|} \tag{7}$$

Equation 6 is the calculation method of F1-MEASURE.

$$F1 = 2 \times \frac{RECALL \times PRECISION}{RECALL + RPECISION} \tag{8}$$

Here R (u) denotes the recommended list of user u, T (u) denotes the list of behavior records of user u in the test set, and the larger the values of these two indexes are, the better the algorithm is.

Equation 7 is the computation method of MAE

$$MAE = \frac{\sum_{i=1}^{N} |p_i - q_i|}{N} \tag{9}$$

Here, p_i is the predicted score set, N is the predicted number of items, q_i is the actual score set, which reflects the recommended accuracy by calculating the deviation between the predicted and actual values. The smaller the MAE value is, the closer the score is to the true score, the more accurate the forecast is and the better the final recommendation is.

As shown in Figure 2,3,4,5, we compare the content-based recommendation method BMR, the collaborative filtering-based recommendation method BFR, the time-related composite filtering recommendation TBRR[13] and the recall (RECALL), precision (PRECISION), f1 (F1-MEASURE) and mean absolute error (MAE) of the LCS similarity connection technology recommendation method LSCR proposed by us in the Fabric Data Sets with recommended given k=1,5,10,40,50.

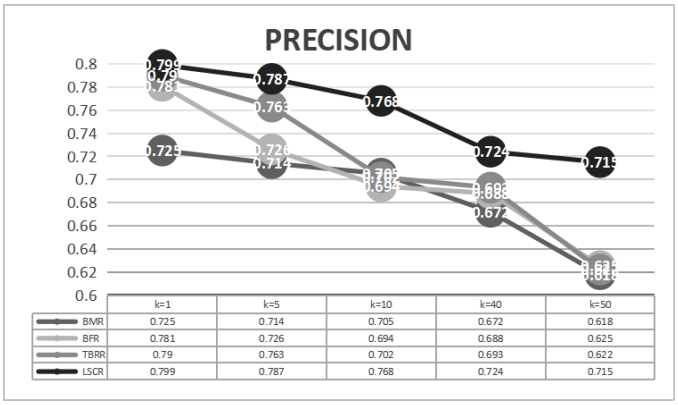

Figure 2: Precision of BMR, BFR, TBRR and LSCR

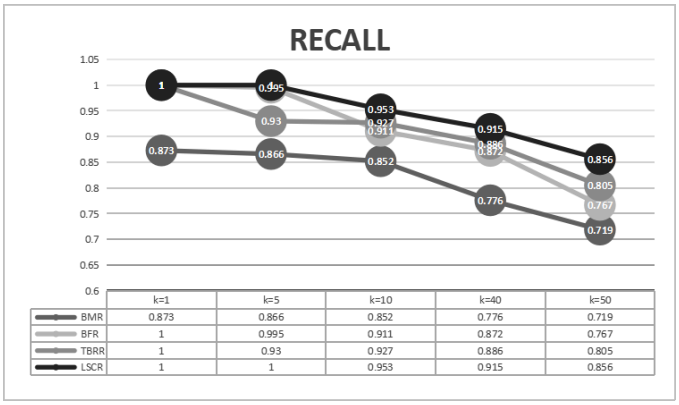

Figure 3: Recall of BMR, BFR, TBRR and LSCR

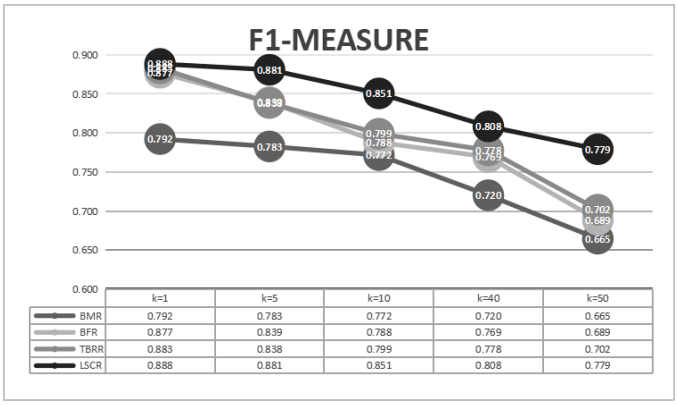

Figure 4: F1-MEASURE of BMR, BFR, TBRR and LSCR

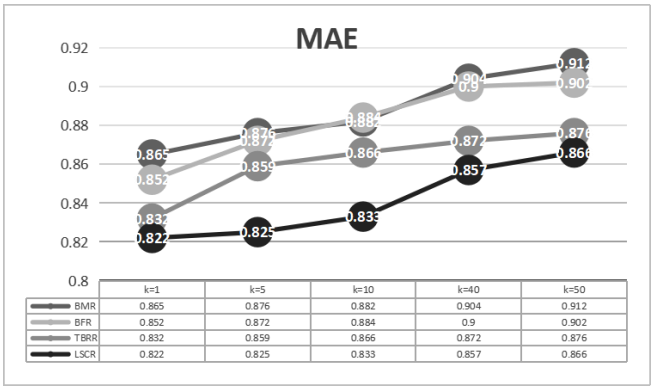

Figure 5: MAE of BMR, BFR, TBRR and LSCR

It can be seen that the LSCR is proposed in this paper combines the advantages of the optimized incremental similarity connection method and the longest common additive sequence of two given input sequences, and the related technical indicators are superior to the traditional BMR, BFR and our previous algorithm TBRR.

VI. Conclusion

In order to solve the timeliness problem of recommendation system, this paper extracts time series of data according to the time distribution of data and adopts incremental processing method to greatly reduce the calculation amount on the premise that the precision of real-time recommendation system is ensured. For Fabric Data Sets, the technology used in this paper is better than the traditional BMR, BFR and TBRR algorithms in Recall, Precision and MAE. This method, which is applied in manufacturing industry recommendation system, can respond to changing behaviors of users and adjust the ranking of recommendation results in real time, which can continuously improve users' experience in the recommendation system.

Acknowledgments

The study is supported by Natural Science Foundation of Fujian Provincial Science and Technology Department (No.2018J01558, No.2021H6037, No. 2020HZ02014), Key Projects of Fujian Provincial Education Department (No.JAT190509), 2020 Fujian Provincial new engineering research and reform practice project (No.33), Quanzhou Science and Technology Project (No.2021C0008R, No.2021GZ1) , 2018 Fujian provincial undergraduate teaching team project, Fujian Provincial Big Data Research Institute of Intelligent Manufacturing.

References

[1] Gedikli F, Jannach D . Improving Recommendation Accuracy Based on Item-Specific Tag Preferences. ACM Transactions on Intelligent Systems and Technology (TIST), 2013, 4(1):1-19.

[2] Tan Z, Jamdagni A, He X, et al. A System for Denial-of-Service Attack Detection Based on Multivariate Correlation Analysis. IEEE Transactions on Parallel & Distributed Systems, 2014, 25(2):447-456.

[3] Xiaodong, Wang, Lei, et al. An Efficient Dynamic Programming Algorithm for a New Generalized LCS Problem. IAENG Internaitonal journal of computer science, 2016, 43(2):204-211.

[4] Shim, K. , R. Srikant , and R. Agrawal . "High-dimensional similarity joins." IEEE 1997:156-171.

[5] Xiao, C. , et al. "Top-k Set Similarity Joins." IEEE International Conference on Data Engineering IEEE Computer Society, 2009.

[6] Ma Y, Zhang R, Jia S, et al. An efficient similarity join approach on large - scale high - dimensional data using random projection. Concurrency and Computation: Practice and Experience, 2019, 31(11):e5303.

[7] Rafiei, D. , and F Deng. "Similarity Join and Similarity Self-Join Size Estimation in a Streaming Environment." IEEE Transactions on Knowledge and Data Engineering (2019):1-1.

[8] Vernica, R. , M. J. Carey , and L. Chen . "Efficient Parallel Set-Similarity Joins Using MapReduce." (2010).

[9] Albeanu G . Fuzzy joins using MapReduce. Computing reviews, 2013, 54(8):504-505.

[10] Silva Y N, Reed J M . Exploiting MapReduce-based similarity join. Proceedings of SIGMOD.2013.

[11] Pang J, Yu G U, Jia X U . Research Advance on Similarity Join Queries. Journal of Frontiers of Computer ence & Technology, 2013.

[12] Daxin, Zhu, Lei, et al. A space efficient algorithm for the longest common subsequence in k-length substrings. Theoretical Computer Science, 2017.

[13] Danlin Cai, Daxin Zhu, Junjie liu,A Time-Related Composite Filtering Recommendation Method, International Journal of Recent Trends in Engineering & Research, 2017(11)

[14] Metwally A, Faloutsos C . V-SMART-join. Proceedings of the VLDB Endowment, 2012, 5(8):704-715.

[15] Armstrong K . Big Data: A Revolution That Will Transform How We Live, Work, and Think. Mathematics & Computer Education, 2014, 47(10):181-183.

[16] Shim K, Srikant, Ramakrishnan, et al. High-Dimensional Similarity Joins. IEEE Transactions on Knowledge & Data Engineering, 2002.

[17] Kim Y, Shim K . Parallel Top-K Similarity Join Algorithms Using MapReduce. IEEE Computer Society, 2012.

[18] Ge, S. , et al. "Solutions for Processing K Nearest Neighbor Joins for Massive Data on MapReduce." Proceedings of the 23rd International Conference on Parallel, Distributed and Network-based Processing IEEE, 2015.

[19] Rong C, Wei L, Wang X, et al. Efficient and Scalable Processing of String Similarity Join. IEEE Transactions on Knowledge and Data Engineering, 2013.

[20] Jang M, Chang J W . Grid-Based Parallel Algorithms of Join Queries for Analyzing Multi-Dimensional Data on MapReduce. Transactions on Information & Systems, 2018, 101(4):964-976.

[21] Yi L, Luo C, Ning J, et al. Parallel Top-k Spatial Join Query Processing on Massive Spatial Data. Journal of Computer Research and Development, 2011.

[22] Z. Tang, G. Zhao, T. Ouyang, Two-phase deep learning model for short-term wind direction forecasting, Renewable Energy, 173 (2021) 1005-1016.

Siyu Huang was born in Quanzhou, China , in 1976.She received the M.E. degree from the School of software engineering, Sichuan University in 2012. She is currently with the School of Mathematics and Computer Science, Quanzhou Normal University, as a experimentalist. Her research interests include mobile Internet development and algorithms.

Xueyan Huang was born in Quanzhou, China , in 1983.She received a master's degree in management from the School of Public Administration of overseas Chinese University in 2014. She currently serves as the director of the teaching and research office of Quanzhou normal University, assisting the head of the college to carry out the management work of the degree site construction, qualification and special evaluation of the college.

Taisheng Zeng was born in Quanzhou(Fujian),china,in 1975.He received the BS and MS degrees from the HuaQiao University, China, in 1998 and 2005. His main research interests include image engineering(image processing , image analysis, image understanding and technique application),and big data technology.

Danlin Cai was born in Quanzhou,China, in 1977.She received the M.E. degree from the School of Computer Science, Huaqiao University ,Fujian, China, in 2007. From 2015 to 2016, she worked as a visiting scholar in University of Maryland , College Park, US. She is currently with the School of Mathematics and Computer Science, Quanzhou Normal University, as a Professor. Her research interests include data intensive computing, network model, and information model.

Corresponding authors:Daxin Zhu was born in 1976. He received the M.E. degree from the School of Computer Science, Huaqiao University, Fujian, China. He was a Visiting Scholar with the Department of Civil and Environmental Engineering, University of Maryland, College Park (UMD) ,USA, from 2015 to 2016, and the Department of Computer Science, Aberystwyth University, UK, from 2019 to 2020. He is currently a Professor with the School of Mathematics and Computer Science, Quanzhou Normal University, China. His current research interests include data intensive computing, network model, and information security.

Creative Commons Attribution License 4.0 (Attribution 4.0 International, CC BY 4.0)