Available online at www.ijpe-online.com

International Journal of Performability Engineering

vol. 15, no. 2, February 2019, pp. 710-718
DOI: 10.23940/ijpe.19.02.p35.710718

Database Repeat Record Detection based on Improved Quantum Particle Swarm Optimization Algorithm

Guangzhou Yu*

Educational Information Center, Guangdong Ocean University, Zhanjiang, 524008, China

Abstract

The detection of similar duplicate records was a key link in database data cleaning. In the process of detecting duplicate records in the same amount of data, the record attribute dimension was too high, which led to the problems of precision, recall and time efficiency. A database repeat recording detection method based on the IQPSO (Improved Quantum Particle Swarm Optimization) algorithm was proposed. The method constructed an entropy metric in terms of the similarity between objects, and evaluated the importance of each attribute in the original data set of the database, thereby removing unimportant or noise attributes. A subset of key attributes was preferred and attribute dimensions were reduced.Large data sets were divided, in the database, into multiple disjoint small data sets based on key attributes. Each small data set was used as an input to the support vector machine. The IQPSO algorithm was used to optimize the parameters of the support vector machine to obtain the optimal parameters of the support vector machine. The repeated record detection model was constructed according to the optimal parameter training classifier, and the model was used to perform similar repeated record detection. The experimental results indicated that the proposed method effectively improved the detection efficiency under the premise of ensuring the highest recall rate and precision. The proposed method also solved the problem of database similar duplicate record detection effectively.

Keywords: improved quantum particle swarm optimization algorithm; database; repeated record detection; support vector machine

(Submitted on November 10, 2018; Revised on December 11, 2018; Accepted on January 3, 2019)

1. Introduction

With fast development and wide application of database technology, the amount of data accumulated by various industries is increasing [1-2]. People have estimated that the amount of information in the world will double every 20 months, and the number and size of databases are growing at a faster rate. Faced with such a large amount of data, people hope to extract valuable information or knowledge from massive data to provide the reference for decision makers [3]. However, due to data entry errors, data fusion and migration of different representation methods, etc., the system inevitably has redundant data, missing data, uncertain data,and inconsistent data, etc. [4]. These data are collectively referred to as “dirty data”, which are derived from different channels leading to an increase in similar duplicate records, which seriously affects data utilization and decision quality. The detection and cleaning of similar duplicate records have become a hot research issue in the fields of data warehousing and data mining.

In recent years, scholars have proposed some new methods for database duplicate record detection. For example, Song et al. proposed a big data similar duplicate record detection algorithm based on the MapReduce model. The traditional SimHash algorithm was improved by using Chinese lexical analysis technology, Delphi method and word frequency-reverse file frequency algorithm. This method can solve the problems of the slow keyword extraction speed and low accuracy and the weight calculation accuracy [5]. The traditional SimHash algorithm was optimized by the inverted index algorithm to improve the matching efficiency of similar duplicate records. The Map function and the Reduce function were defined on the cloud platform by using the proposed MP-SYYT algorithm, and the parallel detection and direct output of big data similar repeated records were realized in the cloud environment by using the MapReduce model. Since the MP-SYYT

* Corresponding author.
E-mail address: wz20160401@163.com

algorithm had a great relationship with the Map function and the Reduce function setting, the actual recall rate and the precision rate were low and unstable [6]. An improved algorithm for high-dimensional similar duplicate record detection based on R-tree index was proposed. The R-tree was used to construct the index to preserve the high-dimensional space characteristics of the record, and the distance algorithm for measuring the similarity of the records was improved to avoid the influence of high-dimensional data sparsity. This method was more efficient for small-scale databases, but its detection efficiency was not ideal for large-scale databases.

Aiming at the above problems, a database repeat recording detection method based on IQPSO algorithm is proposed. The main research work is as follows.

(1) For the database duplicate record detection problem, by constructing an entropy measure in terms of the similarity between objects, the weight of each attribute in the database original data set is evaluated, and the key attribute set is selected. According to the key attributes, the data is divided into disjoint small data sets, and the similar repeated records are detected by the support vector machine method in each small data set.

(2) For the problem that the detection accuracy of the support vector machine algorithm is not high, the IQPSO algorithm is used to optimize the parameters of the support vector machine to obtain the optimal parameters of the support vector machine. The repeated record detection model is constructed by training the classifier according to the optimal parameters.

2. Database Repeat Record Detection Method based on IQPSO Algorithm

2.1. Similar Repeated Record Detection based on Entropy Feature Attribute Partitioning

By constructing an entropy metric in terms of the similarity between objects, the weight of each attribute in the original dataset of the database is evaluated, and the unimportant or noisy attributes are removed. The key attribute subsets are optimized, and the attribute dimension is reduced. According to the key attributes, the large data set in the database is divided into multiple disjoint small data sets, and the similar repeated records are detected by the support vector machine method in each small data set. The specific process is as follows:

Suppose there are n objects $x_i(x_{i1},\cdots,x_{im})^T$ in the m -dimensional feature space in relation R , and the distance between the two objects is measured by the norm distance [7]:

$$Lp(x_i,x_j)=\left\|x_i-x_j\right\|=(\sum_{k=1}^{m}\left\|x_{ik}-x_{jk}\right\|^p) \tag{1}$$

It can be seen from the formula that the distance L_1 is the absolute distance when $p=1$; when $p=2$, the distance L_2 is the Euclidean distance; the distance is the Chebyshev distance when $p\to\infty$ .Considering when $p\geq 2$, the distance between two objects x_{ik} , x_{jk} in the k -dimensional feature space converges to zero as the dimension increases. The distance between two objects is calculated using the L_1 distance, which is obtained by Equation (1).

$$L_1=(x_i-x_j)=\sum_{k=1}^{m}\left\|x_{ik}-x_{jk}\right\| \tag{2}$$

Assume that the database data feature set of the m-dimensional space is $F=\{X_1,\cdots,X_m\}$. In order to remove the influence of different dimensions on the distance, the standard deviation of the characteristic variable X_k is introduced $S_k=\sqrt{\frac{1}{m}\sum_{k=1}^{m}(X_k-\bar{X})^2}$, $\bar{X}$ is the average value of the characteristic variable X_k . In addition, the normalization coefficient λ is introducedin the consideration of dimensional normalization of the distance. The distance between two objects can be transformed by Equation (3).

$$d(x_i, x_j) = \sqrt{\frac{1}{\lambda}\sum_{k=1}^{m}\frac{|x_{ik} - x_{jk}|}{S_k}} \tag{3}$$

According to the distance metric given by Equation (3), the similarity between two objects can be described as Equation (4).

$$s(x_i, x_j) = e^{-\frac{d(x_i, x_j)}{\sigma}} \tag{4}$$

In Equation (4), $e^{(\cdot)}$ represents the approximation adjustment coefficient, and parameter σ is used to adjust the attenuation property of the similarity. The smaller the σ, the faster the similarity decays with distance.

Equation (4) shows that when $d(x_i, x_j) = 0$, $s(x_i, x_j) = 1$ indicates that the two objects are completely repeated; when $d(x_i, x_j) \to \infty$, $s(x_i, x_j) = 1$ indicates that the two objects are completely unrelated.

Assuming n objects $x_i(x_{i1}, \cdots, x_{im})^T$ in the m-dimensional feature space in relation R, the similarity matrix between objects is expressed as Equation (5).

$$S = (s_{ij})_{n\times n} = \begin{bmatrix} s_{11} & \cdots & s_{1n} \\ & \cdot & \\ & \cdot & \\ & \cdot & \\ s_{n1} & \cdots & s_{nn} \end{bmatrix} \tag{5}$$

In Equation (4), any two objects in relation $s_{ij} = s(x_i, x_j)$, the entropy metric between them is defined as Equation (6).

$$H_{ij} = \frac{1}{\ln 2}[s_{ij} \times \ln 2s_{ij} + (1 - s_{ij}) \times \ln 2(1 - s_{ij})] \tag{6}$$

Equation (6) shows that entropy measures the nature of the similarity between two objects.

(1) When $s(x_i, x_j) = 1$, $d(x_i, x_j) = 0$ or when $s(x_i, x_j) \to 0$, $d(x_i, x_j) \to \infty$, $H_{ij} = 0$, that is, the maximum or minimum similarity between two objects, the entropy is the smallest.

(2) When $s(x_i, x_j) = 0.5$, $d(x_i, x_j) = \sigma \ln 2$, $H_{ij} = 1$, the entropy is the largest. If all objects in the feature space are considered, the overall entropy can be expressed as Equation (7).

$$E_H = \frac{1}{n^2}\sum_{i=1}^{n}\sum_{j=1}^{n} H_{ij} \tag{7}$$

Where H_{ij} represents the attribute entropy of a feature. Let the attribute set of the m-dimensional space be $F = \{X_1, \cdots, X_m\}$, then the importance of the attribute X_k can be measured by the increase of the total entropy after deleting the attribute X_k in the feature set F.

$$\mathrm{Im}\, p(X_k) = E_H(F - X_k) - E_H(F) \tag{8}$$

Where $(F - X_k)$ indicates that the attribute X_k is deleted from the attribute set F, and $E_H(\cdot)$ indicates the total entropy measure value before and after the attribute X_k is deleted. In order to make the entropy in the attribute subspace of different dimensions comparable, the inter-record similarity should be calculated after the dimension normalization pre-processing.

After deleting attribute X_i, if the overall entropy is increased, $\mathrm{Im}\, p(X_k) \geq 0$, it indicates that the attribute is very distinguishable and helps to distinguish the cluster structure, and the attribute should be retained. Conversely, if attribute X_i is deleted, the overall entropy is reduced by $\mathrm{Im}\, p(X_k) \leq 0$. This indicates that the attribute is an unrelated attribute and can be deleted. When $\mathrm{Im}\, p(X_k) = 1$, the attribute X_i is deleted, which has the unique role of distinguishing clusters and should be retained. When $\mathrm{Im}\, p(X_k) = -1$, the deleted attribute X_i is completely confused, difficult to cluster, and should be deleted. After calculating $\mathrm{Im}\, p(X_k)$, the importance of attributes can be measured and sorted according to values, and important attributes are selected to remove unimportant attributes, thereby reducing the dimension of the attributes.

After the data is grouped, it is necessary to detect similar duplicate records of the small data sets of each group. Before cluster detection, the records in the data set are first mapped to points in the n'-dimensional space. Geometrically, if the two points are close together, they are two similar records in the data set. Calculation of the similarity between the two records evolves into the calculation of the distance between the two points. Considering the accuracy and time of detection, according to the idea based on q-gram algorithm [8], all records in the grouped small data set are mapped to a one-dimensional space after attribute reduction by 2-gram. A large number of records appear as different points in a n-dimensional space, and each point has n dimensions. Assume that relationship R contains all the records, and the number of attributes of R is recordedas $dom(R)$. The set of all strings of attribute j is recordedas $R[j]$. The j^{th} attribute value of each record recorded as $r.r$ is recorded as $r[j]$. $Gq[r]$ representsa multipleset of all attribute q-grams of record r.The N-gram hierarchical spatial similarity measure [9] is used to project the records into points in space. Then, after the two records r_1, r_2 are mapped into points in the n-dimensional Ω_q space, the similarity between the two can be calculated by Equation (9).

$$sim_q(r_1, r_2) = \frac{\sum_{i=1}^{n} \min\left(cood_i(r_1), cood_i(r_2)\right)}{\sum_{i=1}^{n} \max\left(cood_i(r_1), cood_i(r_2)\right)} \tag{9}$$

Among them, $cood_i(r_1)$ represents all attributes of record r_1, and $cood_i(r_2)$ represents all attributes of record r_2. $\max(\cdot)$ represents the maximum approximation, and $\min(\cdot)$ represents the minimum approximation.

The above process reduces the dimension of the spatial point, but the dimension is still high for the data in the database. For the characteristics of high-dimensional data, the support vector machine algorithm [10] is used in each small data set to detect similar duplicate records.

Assume that H represents a sample randomly selected in each group, and all training samples are divided into two categories by finding an optimal classification hyperplane.

$$y_i\left\{\left[\psi(x_i), \omega\right] + b\right\} \geq 1 \tag{10}$$

Where b represents the bias term, ω represents the adjustable weight vector, $\psi(x_i)$ represents the vector in the sample set, and y_i represents the algebraic distance. For a class hyperplane, parameter (ω, b) is not uniquely determined. There must be a pair of (ω, b) guarantees (10), and the minimum distance between $\psi(x_i)$ and the class hyperplaneis $1/\|\omega\|$, allowing some misclassified points to exist, so that Equation (10) becomes Equation (11).

$$y_i\left\{\left[\psi(x_i), \omega\right] + b\right\} \geq 1 - \zeta_i \tag{11}$$

Where ζ_i represents a negative relaxation variable when $\zeta_i = 0$ is satisfied. It indicates complete linear separability.

For a linear indivisible problem, you need to turn it into an optimization problem and then find the optimal classification hyperplane. By introducing a penalty factor of $C > 0$, Equation (12) is obtained.

$$\min \psi(\omega) = \frac{1}{2}\|\omega\|^2 + C\sum_{i=1}^{n}\zeta_i \qquad s.t.\left\{y_i\left\{\left[\psi(x_i),\omega\right]\right\}+b\right\} \geq 1-\zeta_i \tag{12}$$

Introducing the Lagrangianoperator α_i to convert the above equation to Equation (13).

$$\max W(\omega) = \sum_{i=1}^{n}\alpha_i - \frac{1}{2}\sum_{i=1}^{n}\alpha_i y_j \psi(x_i) \qquad s.t.\begin{cases} 0 \leq \alpha_i \leq C \\ \sum_{i=1}^{n}\alpha_i y_j = 0 \end{cases} \tag{13}$$

The classification discriminant function of the support vector machine is given by Equation (14).

$$f(x) = \omega\psi(x) + b = \sum_i \alpha_i K(x_i, x) + b \tag{14}$$

$K(x_i, x)$ represents a Gaussian kernel function.

The similar discrepancy record is detected using the classification discriminant function given by Equation (15).

$$J(\kappa) = \frac{sF(x_i, x_j)}{\zeta_i} f(x) \tag{15}$$

Where sF is the change category of the value of the duplicate record attribute.

2.2. IQPSO Algorithm for Supporting Similar Repetitive Record Detection of Support Vector Machines

When the values of the support vector machine parameters C and ω are different, the classification performance of the support vector machine is very different. This shows that the merits of the parameters C and ω directly determine the classification performance of the support vector machine, and the IQPSO algorithm optimizes the selection of the support vector machine parameters. According to the diversity of the population in the evolution of the population, the adaptive adjustment strategy [11] is used, taking into account the diversity and convergence speed of the population, and preventing the quantum particle swarm algorithm from falling into the local optimal value as much as possible. According to these methods, the purpose of improving the search capability and performance of the quantum particle swarm optimization algorithm is achieved. The measure of population diversity is calculated by calculating the sum of Euclidean distances between particles or the sum of Euclidean distances from all particles to the center. The following measurement methods are used.

$$D(S(t)) = \frac{1}{n_s L}\sum_{t=1}^{n_r}\sqrt{\sum_{j=1}^{n_x}\left(x_{i,j'}(t) - \overline{x_{j'}(t)}\right)^2} \tag{16}$$

Among them, $S(t)$ refers to the entire group. n_s is the size of the entire group, L is the longest diagonal length of the search space. n_x is the dimension of the problem solved, $x_{i,j'}(t)$ is the value of the j'^{th} dimension of the t^{th} particle. $\overline{x_{j'}(t)}$ is the j'^{th} average of all particles.

In general, populations that are randomly initialized will be relatively high during the initial stages of population evolution. As evolution progresses, particles will continue to aggregate and diversity will gradually decrease, which will result in enhanced local search capabilities of the algorithm and weaken global search capabilities. For the early and middle stages of evolution, the reduction in diversity is necessary for effective searching. However, in the middle and late stages,

especially in the later stages, when the diversity is reduced to a very small extent, the particles are concentrated in a small area, which may cause the whole population to fall into local extreme points. In order to overcome the loss of group diversity, a simple and effective method is to combine the advantages of other evolutionary algorithms with the quantum behavior particle swarm algorithm. Cross-operation and mutation operations [12] were introduced, and different adaptive adjustment strategies were adopted according to the diversity of groups in the evolution process.

When the population is more or less equal to the threshold d_l during evolution, the two particles with the lowest fitness value are generated by the current optimal position Pbt_w and the current optimal velocity of a randomly selected particle Pbt_τ is used to generate two new individuals.

$$\begin{cases} x_{new}^1 = \tau \cdot Pbt_w + (1-\tau) Pbt_\tau \\ x_{new}^2 = (1-\tau) \cdot Pbt_\tau + \tau Pbt_\tau \end{cases} \tag{17}$$

τ is a random number between $[0,1]$. Compare the fitness values of x_{new}^1 and x_{new}^2 and choose the better one and continue to compare with x_w. If the fitness value is better than x_w, replace the worst particle with the new one. This allows the group to move closer to the optimal solution and speed up the convergence.

When the group evolves, the diversity will continue to decrease. When 1 is less than a certain threshold 2, certain measures need to be taken to maintain the diversity of the group, and to avoid falling into the local optimum point as much as possible. Muting the particles to jump to a new location is a common way to improve the diversity of the population [13]. There are usually two kinds of variation: Cauchy variation and Gaussian variation. Because the two wings are almost parallel to the axis of abscissa, the Cauchy variation is more likely to generate random numbers farther from the origin, which makes it easier for the population to escape from the local optimum. Use the Cauchy variation to generate random numbers. The probability density function of the Cauchy mutation is defined as Equation (18).

$$f(x) = \frac{1}{\pi} \cdot \frac{a}{a^2 + x^2}, \quad -\infty < x < \infty \tag{18}$$

Where a is the scale parameter, and its distribution function is defined as Equation (19).

$$F(x) = \frac{1}{2} + \frac{1}{\pi} \arctan\left(\frac{x}{t}\right) \tag{19}$$

When $d(S) < d_1$, the individual optimal value of a particle has not improved in the past iterations. It can be assumed that it may be stuck in a local extremum. At this point, you need to jump to a new point, enhance its search ability, and perform a Cauchy variation on its position and its individual optimal value. Generate new individuals for each particle selected in the following way.

$$\begin{cases} x_i'' = x_i' + \eta C(0,1) \\ Pbt_i'' = Pbt_i' + \eta C(0,1) \end{cases} \tag{20}$$

$C(0,1)$ represents the Cauchy random number and η represents the scale parameter.

The fitness of individual particles plays a key role in the process of database repeat recording and detection. Follow these guidelines: The particles with the highest classification accuracy and less repeated records are selected as the quality individuals, then the adaptability is better. The fitness function can be designed as Equation (21).

$$fit = W_a \times SVM + (1-W_a) \times F\_s \tag{21}$$

SVM indicates the classification accuracy of the support vector machine. F_s indicates the number of duplicate records currently selected. The precision weights W_a and $(1-W_a)$ respectively indicate the importance of the classification

performance of the support vector machine and the size of the selected number of repeated records in the fitness. The optimal parameters of the support vector machine can be obtained by designing the fitness function.

$$TE = \sum_{i=1}^{n} fit \cdot F\_aW_a \tag{22}$$

Construct a duplicate record detection model based on the optimal parameter training classifier.

$$FA = \sum_{i=1}^{n} \frac{TE \cdot C(0,1)}{\eta} \tag{23}$$

η indicates the number of iterations.

3. Experimental simulation

In order to account for the comprehensive effectiveness of the database repeat record detection method based on the improved quantum particle swarm algorithm, simulation experiments were carried out in a CPU P42.8 GMHZ, RAM 2GB, and operating system Windows 2000 environment. Using SQL Server 2005 as the database software, the data source of a website's talent database, a total of 80,000 records each consisting of 4 segments was manually added with 500 duplicate records and the records were divided into training and test sets. Under the same experimental conditions, the big data similar repeated record detection algorithm based on the MapReduce model and the improved algorithm of high-dimensional similar repeated record detection based on the R-tree index were selected for comparison experiments. The evaluation index selects the commonly used precision and recall rate.

The Precision Ratio of the record indicates the proportion of the correct similar duplicate records detected in all detected similar duplicate records. The recall ratio (Recall Ratio) of the record represents the proportion of the actual duplicate similar records recorded in the database for the correct similar duplicate records. Let C denote the actual similar duplicate records in the database. F indicates the correct similar duplicate record detected. T indicates the similar repeated records detected, and the precision ratio R_P and the recall rate are calculated by Equations (24)-(25).

$$R_P = F / T \tag{24}$$

$$R_R = F / C \tag{25}$$

Figure 1 and Figure 2 show the change of the precision and recall rate of different detection methods with the number of records. In order to simplify the description,the big data similar repeated record detection algorithm based on the MapReduce model and the improved algorithm of high-dimensional similar repeated record detection based on R-tree index are respectively denoted as B, F and H.

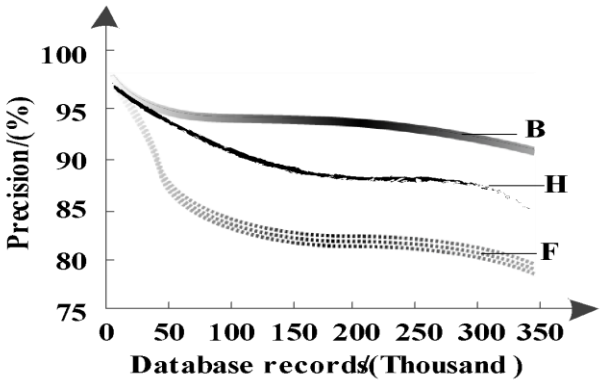

Figure 1. The variation of the precision of different methods with the number of records

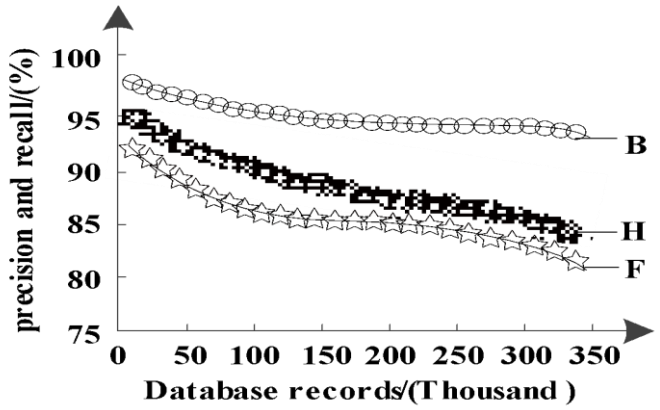

Figure 2. Changes in the recall rate of different methods with the number of records

As can be seen from Figure 1 and Figure 2, when the number of database records is 50,000, the precision of the proposed method is 95.5%. The precision of the big data similar duplicate record detection algorithm based on the MapReduce model is 93.2%. The accuracy of the improved algorithm for high-dimensional similar duplicate records detection based on the R-tree index is 87%. When the number of records is relatively small, the performance of the three detection methods is not very different. However, with the increase of records, the performance of the big data algorithm based on the MapReduce model and the improved algorithm of high-dimensional based on the R-tree index is much lower. The big data algorithm based on the MapReduce model has a great relationship with the Map function and Reduce function setting, resulting in a low actual recall rate and precision. In contrast, the proposed method is superior to the big data algorithm based on the MapReduce model and the high-dimensional algorithm based on the R-tree index. This is mainly because the IQPSO algorithm optimizes the parameters of the support vector machine, obtains the optimal parameters of the support vector machine, and builds the repeated record detection model based on the obtained optimal parameter training classifier, which improves the detection accuracy.

The proposed method, the big data similar algorithm based on the MapReduce model and the improved algorithm of high-dimensional based on the R-tree index are used for repeated record detection. The required running time (ms) comparison results are shown in Figure 3.

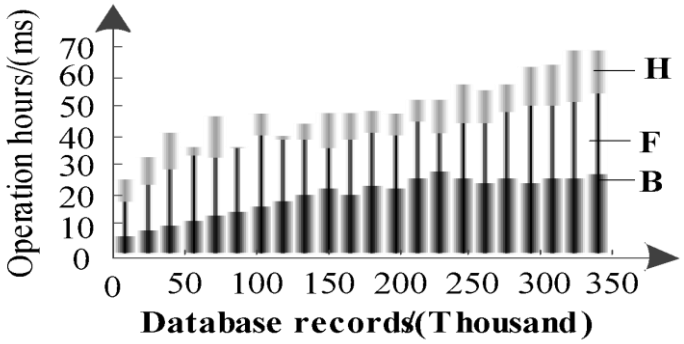

Figure 3. Comparison of running time of different methods

It can be seen from Fig. 3 that with the increase of the number of records, the proposed method, the big data similar repeated record detection algorithm based on the MapReduce model and the improved algorithm of high-dimensional similar repeated record detection algorithm based on R-tree index increase. However, the detection time of the proposed method is shorter. The comparison results show that the proposed method effectively improves the efficiency of database duplicate record detection, and satisfies the real-time requirements of large-scale database duplicate record detection.

4. Conclusions

The detection methods of similar duplicate records existing at present are studied, and the existing methods for detecting similar duplicate records were studied. The main work was summarized as follows.

(1) Focus on the detection methods of similar duplicate records. The method of similarity measurement between records was studied, and the related algorithms of similar repeated record detection were proposed. The method of grouping clustering similar duplicate records based on the entropy feature was proposed. By constructing an entropy metric in terms of the similarity between objects, the weight of each attribute in the original data set was evaluated, and the key feature set was selected. According to the key attributes, the data was divided into disjoint small data sets, and the similar repeated records were detected by the support vector machine algorithm in each small data set.

(2) The performance of the support vector machine classifier was affected by the repeated record data and parameter settings input during training, and two factors needed to be optimized at the same time. The quantum particle swarm optimization algorithm had good optimization ability and had the ability to synchronously optimize the repeated record data and parameter estimation. The IQPSO algorithm was used to optimize the parameters of the support vector machine, and the optimized support vector machine parameters were used in the process of repeated record detection, which effectively improved the accuracy of similar duplicate record detection.

The proposed method improved the quantum behavior particle swarm optimization algorithm. Compared with the original quantum behavior particle swarm optimization algorithm, the performance made great progress. However, there was still the possibility of falling into local optimums. Further improvements to the quantum behavior particle swarm optimization algorithm can be considered from other perspectives.

References

1. T. Papenbrock, A. Heise, and F. Naumann, "Progressive Duplicate Detection," *IEEE Transactions on Knowledge and Data Engineering*, Vol. 27, pp. 1316-1329, 2015
2. W. Liu and J. Zeng, "Duplicate Literature Detection for Cross-Library Search," *Cybernetics and Information Technologies*, Vol. 16, pp. 160-178, 2016
3. W. Xia, H. Jiang, D. Feng, and L. Tian, "DARE: A Deduplication-Aware Resemblance Detection and Elimination Scheme for Data Reduction with Low Overheads," *IEEE Transactions on Computers*, Vol. 65, pp. 1692-1705, 2016
4. K. Kreimeyer, D. Menschik, S. Winiecki, W. Paul, F. Barash, E. J. Woo, et al., "Using Probabilistic Record Linkage of Structured and Unstructured Data to Identify Duplicate Cases in Spontaneous Adverse Event Reporting Systems," *Drug Safety*, Vol. 40, pp. 571-582, 2017
5. B. Jia, S. Liu, and Y. Yang, "Fractal Cross-Layer Service with Integration and Interaction in Internet of Things," *International Journal of Distributed Sensor Networks*, Vol. 10, pp. 760248, 2014
6. Z. Pan, S. Liu, and W. Fu, "A Review of Visual Moving Target Tracking," *Multimedia Tools & Applications*, Vol. 76, pp. 16989-17018, 2017
7. V. López, S. del Río, J. M. Benítez, and F. Herrera, "Cost-Sensitive Linguistic Fuzzy Rule based Classification Systems under the MapReduce Framework for Imbalanced Big Data," *Fuzzy Sets and Systems*, Vol. 258, pp. 5-38, 2015
8. R. Singh, D. Rai, R. Prasad, and R. Singh, "Similarity Detection in Biological Sequences using Parameterized Matching and Q-gram," in *Proceedings of 2018 Recent Advances on Engineering, Technology and Computational Sciences*, pp. 1-6, 2018
9. Y. Liu, L. Jiao, and F. Shang, "An Efficient Matrix Factorization based Low-Rank Representation for Subspace Clustering," *Pattern Recognition*, Vol. 46, pp. 284-292, 2013
10. B. K. Mishra, A. Rath, N. R. Nayak, and S. Swain, "Far Efficient K-Means Clustering Algorithm," in *Proceedings of the International Conference on Advances in Computing, Communications and Informatics*, pp. 106-110, 2012
11. M. Bilenko and R. J. Mooney, "Adaptive Duplicate Detection using Learnable String Similarity Measures," in *Proceedings of the ninth ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, pp. 39-48, 2003
12. Y. Kwon, M. Lemieux, J. McTavish, and N. Wathen, "Identifying and Removing Duplicate Records from Systematic Review Searches," *Journal of the Medical Library Association*, Vol. 103, pp. 184, 2015
13. S. Liu, W. Fu, H. Deng, C. Lan, and J. Zhou, "Distributional Fractal Creating Algorithm in Parallel Environment," *International Journal of Distributed Sensor Networks*, Vol. 9, pp. 281707, 2013