Hindawi
Mathematical Problems in Engineering
Volume 2022, Article ID 5234230, 8 pages
https://doi.org/10.1155/2022/5234230

Research Article

A Case Retrieval Strategy for Traffic Congestion Based on Cluster Analysis

Hao Zhang[1] **and Jing Yang**[2]

[1] *College of Mathematics and Computer Science, Tongling University, Tongling 244000, China*
[2] *Department of Information Engineering, Anhui Industry Polytechnic, Tongling 244000, China*

Correspondence should be addressed to Hao Zhang; 027510@tlu.edu.cn

Received 13 June 2022; Accepted 5 July 2022; Published 10 August 2022

Academic Editor: Gengxin Sun

In order to improve the retrieval efficiency, this paper uses case-based reasoning (CBR) in the retrieval of traffic congestion cases and tries to adopt the strategy of clustering case databases before retrieval so as to narrow the scope of case retrieval. In terms of case clustering, the *k*-means algorithm, with excellent performance in text clustering, is selected to cluster traffic congestion edge cases. At the same time, considering that there is a certain similarity among the descriptions of traffic congestion, the *K*-means algorithm is optimized to generate an accurate clustering. Those edge cases are clustered into microcase clusters of traffic congestion and then divided into different traffic congestion categories according to the distance of cluster center. Experimental results show that the clustered case base is divided into several microcase bases, which improves the accuracy and shortens the retrieval time in the process of retrieval and provides a new idea for the retrieval method in the process of case-based reasoning.

1. Introduction

A segmentation method for retrieval is proposed in Reference [1]. First, similar case groups of different levels are formed according to the importance of events, and then the degree of similarity is calculated according to the new event levels and related case groups. The method of clustering associated cases is used to improve the success of case retrieval to a certain extent in Reference [2]. An intracase crossover algorithm is proposed to improve the processing effect of parallel data and the efficiency of case retrieval in Reference [3]. A cleaning algorithm for regression filtering is put forward in Reference [4], which shortens the time for case retrieval. In Reference [5], the optimization method of the GRNN neural network is used to improve the efficiency of CBR retrieval, realize the self-learning and self-growth of field problem diagnosis, and effectively avoid the problems of low matching degree and slow convergence speed of traditional CBR algorithms. A sememe-based set similarity matching algorithm (CMSBS) is proposed in Reference [6], which is used to analyze cases with high similarity to the current case. Experiments show that the algorithm has better performance in terms of matching cases and matching accuracy. In Reference [7], the case similarity calculation methods of 5 different attributes are analyzed, and a mode of combining subjective weights and objective weights is put forward. A combination of local and global similarity calculation methods for different types of traffic congestion is adopted in Reference [8]. At the same time, the updating and preserving mode of the traffic congestion case database is proposed. In Reference [9], a traffic emergency decision-making method is designed. At the same time, the case database for traffic-aided decision-making is established, the calculation method of similarity in global-local features is designed, and a case retrieval strategy is given. The weighted information degree to model the traffic route horizontally is used, and a new method for sampling the weighted competition value for a single demand level is proposed in Reference [10]. In Reference [11], a microsimulation to characterize the flow interaction is created by using the toolchain sumo–jade, ensuring that the emergency vehicles arrive as quickly as possible. In Reference [12], a hierarchical structure for representing historical cases is developed. Reference [13] evaluates the strategy of optimizing the performance of the road network by combining real-time traffic information with predicted traffic information and adopting a heuristic dynamic traffic assignment (DTA) model combined with case-based reasoning technology for instance detection. Reference [14]uses case-based reasoning to calculate the shortest path of traffic and get the optimal solution.

Case-based reasoning, used in traffic safety, has also been widely studied but mainly focused on rail transit or large road networks. Therefore, the combination model of rule-based reasoning and case-based reasoning is mostly used in those research studies for regulation of data analysis. Most of them related to urban road congestion are about congestion prediction, and there are relatively few research studies on the timely dredging of congestion and even fewer on the decision support system of urban road traffic congestion dredging by using case-based reasoning. Especially for the application of case retrieval, the methods are often complicated. In this paper, a retrieval strategy based on text clustering is used to improve the retrieval link of case-based reasoning. Experiments also show that this method has a certain superiority and feasibility.

2. Calculation Method for the Attributes of Traffic Congestion

2.1. Enumeration Property Calculation. Enumeration data are unstructured data and mainly perform Boolean calculations. The value can be 0 or 1, where 1 means being the same and 0 means being different. Let the k attribute of C_i and C_j be an enumeration attribute, so

$$\mathrm{sim}\left(C_{ik}, C_{jk}\right) = \begin{cases} 1, & C_{ik} = C_{jk}, \\ 0, & C_{ik} \neq C_{jk}. \end{cases} \tag{1}$$

2.2. Numerical Attribute Calculation. The distance between two different cases in the traffic congestion database is reflected by the difference of the same numerical attribute in the two cases. The similarity calculation is as follows:

$$\mathrm{sim}\left(C_{ik}, C_{jk}\right) = \frac{\left|C_{ik} - C_{jk}\right|}{\left|\max_k - \min_k\right|}. \tag{2}$$

In the formula, $\max_k$ and $\min_k$ represent k's maximum value and minimum value, respectively, in the case.

2.3. Attribute Calculation of Numerical Interval Type. Numerical interval data could be considered as the fuzzy interval. Suppose G_{ik} as the K attribute of C_i, which is a numerical interval type, then G_{ik} is represented as the number of fuzzy intervals $[G_{ik}^-, G_{ik}^+]$, where G_{ik}^- and G_{ik}^+ are the lower and upper limits of the intervals, respectively. Similarly, if the number of fuzzy intervals of the K attribute G_{jk} of C_j is $[G_{jk}^-, G_{jk}^+]$, then the similarity calculation of the K attribute of C_i and C_j is as follows:

$$\mathrm{sim}\left(C_{ik}, C_{jk}\right) = 1 - D\left(C_{ik} - C_{jk}\right) = 1 - \sqrt{\frac{1}{2}\left[\left(G_{ik}^- - G_{ik}^+\right)^2 + \left(G_{jk}^- - G_{jk}^+\right)^2\right]}. \tag{3}$$

In this formula, $D\,(C_{ih} - C_{jk})$ expresses the K attribute of C_i and C_j, the average Euclidean distance:, $D\,(C_{ik} - C_{jk}) \in [0, 1]$.

The similarity calculation has been divided into two steps: when the attributes of different data types are completed, the calculation of similarity among cases is considered the following step. Firstly, the improved algorithm of cluster analysis is used to cluster more than 660 cases in the database. Clustering was carried out according to the traffic congestion cause index of the attribute value.

3. Case Clustering Based on Min-Cluster Distributed Clustering Algorithm

3.1. Selection of Case Library Samples. In a CBR (case-based reasoning) system, the case library, as an important component of the system, is represented in the form of a set. On the assumption that case library $C = (c1, c2, c3, \ldots, cn)$ is a nonempty finite set, which is composed of n cases and $\exists c_i\,(1 \leq i \leq n)$ represents 1 case of the case set. The case library can be classified into m grid units, and $\forall c$ is regarded as 1 grid unit; each grid unit is of the same size, and there is no critical case between grids. But, when starting to classify these grid units, the critical cases within every case are not taken into consideration, and the cases are only classified generally. Thus, inaccuracy is caused by clustering afterwards. Taking this point into account, this paper adopts a new K-means algorithm to cluster, namely, introduce Min-cluster into the critical case and reclassify and cluster for feature value of the critical case, that is, classify the critical case into m grid clustering after the 2nd time clustering, in order to make the target case better find cases that are of more similarity, and perform case treatment to obtain a case optimal solution.

During case retrieval, set the target case, and select the case that is most similar by retrieving the matching degree with elements in Set-C, thus ascertaining the answering case. Meanwhile, store the target case in the case library. The more similarity between ci in case library C and the target case, the better ci answers. Hence, users need to try their best to find among the source cases the most similar case to the target one. A user can calculate the weight of a case in the case library according to the user's feedback on cases and ascertain the best solution based on weight. For selection of the case clustering initial value, on the premise of grid division, put cases of higher weight into the same grid cell pi

($1 \leq i \leq m$), and perform 2^{nd} time refining and clustering through the improved K-means algorithm, thus obtaining cases of higher weight, and classify pi to obtain pi', then store it in the cases of higher weight after 2^{nd} time clustering.

Definition 1. Some of the source cases in the case library are of higher similarity to the target case, which is > the specified threshold value sim, which means they are the very source cases similar to the target case.

Adopt a quadruple to represent the cases: $A =$ (case, area case, tackle case, sim case).

Here, the case represents any one of the cases in Case-C; the area case is the set of all elements in the cluster which takes case as a sample case. Hence, the case set which is similar to the target case can be regarded as the area, where the distance between the target case ≤ sim case; tackle case is the set of answering cases, and the element in a tackle case is represented through two-tuples, $T = (t$ case, count), among which t case is the answering case, while count is the frequency of t case being answered; sim is the case set which is in conformity with the definition. Since the similar cases whose output needs to meet the definition, the similarity between the target case and the answering case is ensured, and the weight of case ci is represented as follows:

$$W(\mathrm{c}_i) = \frac{\text{Count (case)}}{\text{Count (globalcase)}}. \qquad (4)$$

Among which, count (case) is the sum-up of the count for all the elements in the tackle case; count (global case) is the sum-up of the frequency in all the answering cases in Set-C. Finally, calculate out the weight value q of ci.

The purpose of case clustering is to divide cases into several grids and store cases of similarity in each grid. When a target case is mapped to a certain area, and its similarity is found to be relatively higher, it is the very case cluster for target case solution generating.

First, calculate the similarity between all elements in case library C, generally using the Euclidean distance formula as follows:

$$D_{(i,j)} = \sqrt{\left(c_{(i,1)} - c_{(j,1)}\right)^2 + \left(c_{(i,2)} - c_{(j,2)}\right)^2 + \cdots + \left(c_{(i,n)} - c_{(j,n)}\right)^2}. \qquad (5)$$

Represent the similarity of all elements in C through the following similarity matrix:

$$s = \begin{vmatrix} \text{sim}_{11} & \text{sim}_{12} & \cdots & \text{sim}_{1n} \\ \text{sim}_{21} & \text{sim}_{22} & \cdots & \text{sim}_{2n} \\ \vdots & \vdots & \vdots & \vdots \\ \text{sim}_{n1} & \text{sim}_{n2} & \cdots & \text{sim}_{nn} \end{vmatrix}. \qquad (6)$$

Among which, $0 \leq \text{simij} \leq 1$, when $i = j$, $\text{simij} = 1$, and when $i \neq j$, $\text{simij} < 1$. In the matrix, the i th row or i th column is all the similarities between *ci* and other cases.

If perform retrieval and matching use the target case for each case in case library C, more time will be cost, hence, it needs to be performed that, clustering of similarity for the cases in case library, and classifying cases of more similarity into one grid, with the following rules to be followed.

Rule 1: Combine the two cases, if the similarity between cases is greater than the specified threshold value.

If the two cases *sci* and *scj* exceed the specified threshold value sim, the two cases are regarded as the same, and combine *sci* and *scj* to be one case.

Rule 2: There is no need to store if the case density in grid unit is greater than the specified threshold value.

Set the density threshold value as P. P is the maximum quantity of the stored cases in the area, namely, the density of the cases in the grid is controlled by P; if the density of the cases in the grid is saturated, newly added cases will not be stored, thus ensuring the case quantity and misrepresentations inside the grid.

Rule 3: If the quantity ratio of the noise case in the case library exceeds S, start clustering; if there is no intersection between cases, stop clustering. The effect of the clustering inside the grid unit is shown in Figure 1.

3.2. Source Case 2nd Time Clustering

3.2.1. Improved K-Means Algorithm. The traditional K-means algorithm can be described as follows: randomly select K elements from the set to be clustered as the initial sample according to the given clustering quantity K, through continuous iteration adjust centroid, thus completing clustering. But, because there are some common features between each case in the case library, namely, the limit between cases is relatively fuzzy, hence, the result of retrieval is strongly dependent on the target case. Therefore, the effect of case clustering in a case library should be inclusive of the elements, which are relevant to the target case as much as possible, so as to improve the success rate of retrieval results.

The K-means algorithm transfers data between each station and occupies abundant network resources, so the limit between data cluster is not clear. Meanwhile, there will still be internal data breach during data transferring. Thus, based on K-means, introducing Min-cluster can not only tremendously improve the efficiency of data clustering but also reduce the possibility of data breach.

The improved algorithm interprets the system framework of the K-means algorithm from another perspective, regards the main station as the central point, and the center points of the k clusters, which have already been classified as margin point, and such system framework is regarded as the center-margin structure. In this system framework, each marginalized node only deals with the partial data near this node and analyzes the data, which has already been treated, and then directly submits the analysis result to the center point, which performs the 2^{nd} time treatment and analysis at the center point, finally, obtaining the result of data clustering. The system framework is shown in Figure 2. Because there is no data interaction between each marginalized node in this system framework, each marginalized node only communicates with the center point. There is no meta data transferring in the whole system. Thus, the tremendous loss of meta data during transferring

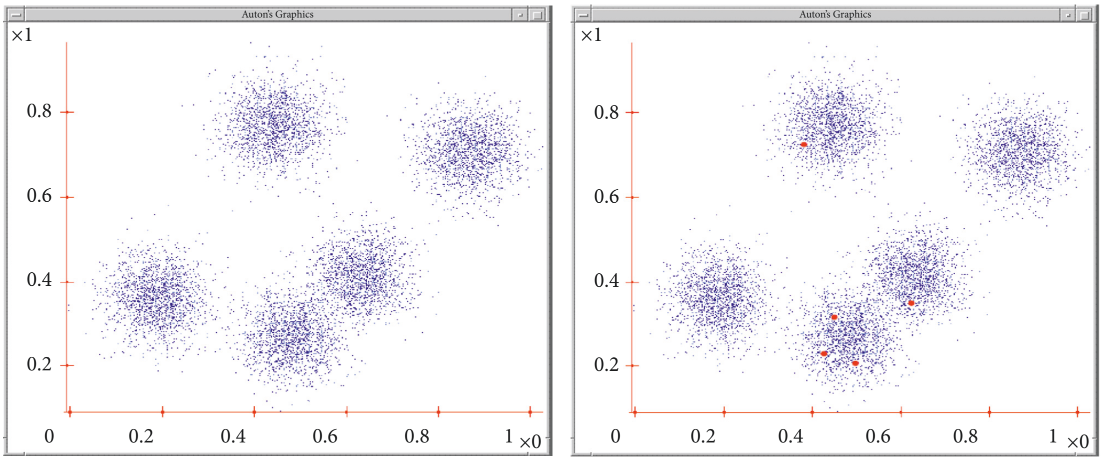

Figure 1: Effect of 1st time grid clustering.

is reduced. Meanwhile, the breach of the meta data during transferring is prevented. Hence, data clustering efficiency is greatly improved.

The case library has already been divided into several grids by the grid clustering at the 1st time in the case library, and cases of more similarity are stored in each grid. But there are still some marginal cases between each grid, which cannot be classified into the corresponding case library. Then, perform 2nd time clustering for the marginal case in grids using the improved K-means algorithm, and again reduce noise of case in grid.

3.2.2. The 2nd Time Clustering Algorithm. In a distributed clustering environment, considering the difference between each node, generally there is a time difference in original data clustering, adopt the K-means algorithm to perform data clustering generally. But, the smaller the quantity of the nodes selected in the K-means algorithm is, the more unstable the result of clustering will be, and the accumulative effect of this clustering instability exists at each marginalized node, finally, it will lead to inaccuracy of data, which is transferred to the center node. Then, to avoid this situation, introduce Min-cluster at the marginalized node and cluster the original data.

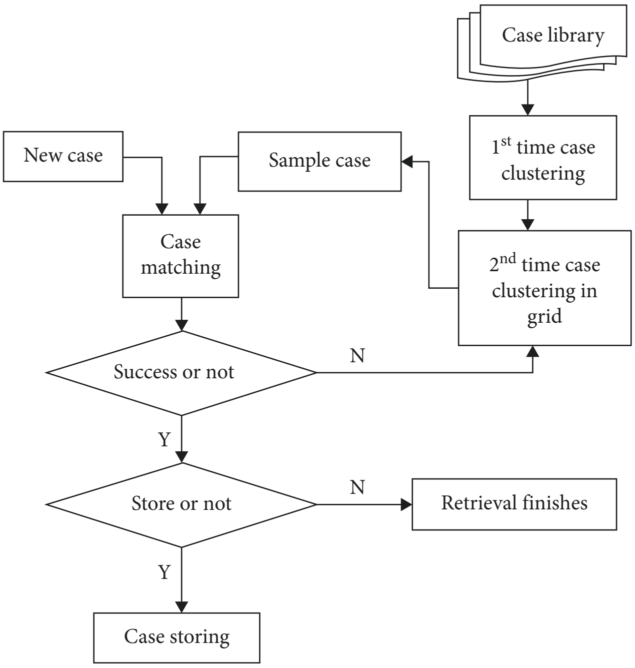

Figure 2: Case retrieval strategy process.

Theorem 1. *Min-cluster created from clustering is the subset of the source case.*

Use reduction to absurdity, and assume that there are n cases: $C1$, $C2, \ldots, Cn$ in 1 grid, this classification is adjacent to the C' case. All the original data points in $C1$, $C2, \ldots, Cn$ are relatively far away from the C' case. While, the process of clustering of original data in $C1$, $C2, \ldots, Cn$ meets the definition of Min-cluster; hence, these Min-clusters are relatively far away from the centroid of the C' case, thus, the 2nd time clustering process of Min-cluster will not be taken into the C' case. Under the same principle, the other cases can be proved, the theorem is proved.

Take the marginal case Sn as an example. Assume that the original data set in this case is N.

(1) Step 1: Select k random data as the initial center point among N data, and based on the established center point, naturally form k Min-clusters; each Min-cluster is $1d + 3$ dimensional vector with the form as $(\overline{CF1^x}, n, \text{class\_id})$.

(2) Step 2: Calculate the distance of all data points to k center points, select the cluster and add it, which is the shortest distance, thus forming Min-cluster.

(3) Step 3: After all data points are added to the cluster, based on the change in the data set, readjust the center $\overline{CF1^x}$ of Min-cluster and the included node quantity n.

(4) Step 4: When $\overline{CF1^x}$ and n do not change, output all Min-clusters; otherwise, return to Step 2.

The Min-clusters which are formed at marginal cases, will finally be transferred to the case center nodes, which have already been formed for fusion. There are different weight values between Min-clusters, namely, the more nodes included in Min-cluster cases, the higher the weight value, and the bigger the possibility to be a classification center. Furthermore, there may be superposition in Min-cluster cases; hence, each Min-cluster case is not equal, and calculation cannot be performed using a general cluster algorithm. Hence, considering that the data clustering for case center point adopt the K-means clustering algorithm based on weight value, take the centroid of Min-cluster as the data object of center node, and distribute different weights according to the n value of each microcluster. The algorithm steps are as follows:

Input: the Min-cluster set $\{C1, C2, \ldots, Cm\}$ from m cases, among which each Min-cluster set Cj includes k' Min-clusters $\{cj1, cj2, \ldots, cjm\}$ after the clustering of this node.

Output: the result of clustering of the whole case set.

(1) Step 1: make treatment for $m * k'$ Min-cluster, select the Min-clusters, which are equal at center, and adjust the value of n;

(2) Step 2: select k clusters, which are of high weight and of relatively large distance between each other, as the initial clustering center; the distribution of data clustering is not even; if centroids A, B of nodes $C1$, $C2$ do not superpose but are very near, then the weight of $C1$, $C2$ is equivalent. If the selected initial centroid is according to the method of weight sequencing, it will cause a cluster with A and B as centroids, and the result of clustering will not be accurate. Hence, this paper sets threshold value and ensures proper centerfold on the premise of performing weight sequencing.

(3) First, for $m * k'$ Min-clusters, perform weight sequencing; second, calculate the average value of the distance between any of the 2 Min-clusters.

$$\delta = \overline{d} = \frac{\sum \mathrm{d}_{ij}}{C_{mk}^{2}}. \qquad (7)$$

(4) Third, perform sequencing according to weight and take the Min-cluster, which ranks 1 as the 1st initial centroid; then, compare the calculated new Min-cluster with the set threshold value. The selected Min-cluster can only be regarded as the initial centroid when their distance is > threshold value

$$d_{ij} > \delta. \qquad (8)$$

(5) Finally, select k clusters, which are of high weight value and with large distance between each other, as the initial clustering center.

(6) Step 3: distribute Min-cluster to the newest cases according to distance, and update the quantity of the original data in case center and case center.

(7) Because Min-cluster itself is a small-scale data set and is different from the data source, which was included in the case previously, gather the Min-cluster as "original data," and calculate its geometric mean to ascertain the center of the classification, instead of only calculating the average value of data points. According to $\overline{CF1^{x}} * n$, which is Min-cluster center point multiplied by the data quantity included in Min-cluster, record the data quantity, obtain the result, and average it to be the center after case updating. There is

$$\overline{CF1} = \sum_{j} \frac{n_j}{\sum n_i} \overline{CF1_j}. \qquad (9)$$

(8) Step 4: If the final case center does not change, proceed to Step 5; otherwise, return back to Step 3.

(9) Step 5: output clustering results.

After the clustering inside the grid for the 2nd time, the cases in the grid are made accurate further. Compare the similarity between cases by clustering marginal cases, reclassify the cases inside the grid, ascertain the center of cases again, enrich the conditions for case retrieval, and compare the target case better as shown in Figure 3.

From Figure 2, it can be detained that marginal cases after 2nd time clustering reduced tremendously, and source cases which are more complete and independent to each other are formed inside the grid generally.

4. Case Retrieval Strategy

The success rate of case solving of CBR intelligent system depends on the quantity and similarity matching of cases in the case library to a large extent. Based on the clustering for 2 times in the case library (as shown above), map the target case to one of the grid units, then, to the utmost, retrieve the matched case target in the grid, based on the result of the matching of cases in sample set S. The case retrieval process is shown in Figure 3.

The case retrieval process is as follows:

(1) The newly created target case matches with the elements in the sample case set S sequentially, and calculates the similarity between cases $\mathrm{sim}_1, \mathrm{sim}_2, \ldots, \mathrm{sim}_n$.

(2) Compare $\mathrm{sim}_1, \mathrm{sim}_2, \ldots, \mathrm{sim}_n$ with the similar threshold value, extract the set S', which is up to the sample case. If S' is empty, the target case is stored and is marked as noise case; if S' is not empty, extract the sample case s', which is the most similar among $\mathrm{sim}_1, \mathrm{sim}_2, \ldots, \mathrm{sim}_n$.

(3) Store s' to temp list and extract elements of s' to match with the target case, acquire the most similar solution set simcase', and sequence according to degree of similarity, and then output.

(4) Judge and ascertain whether or not the recommendation is successful according to users' feedback information. If successful, judge the selected cases in the

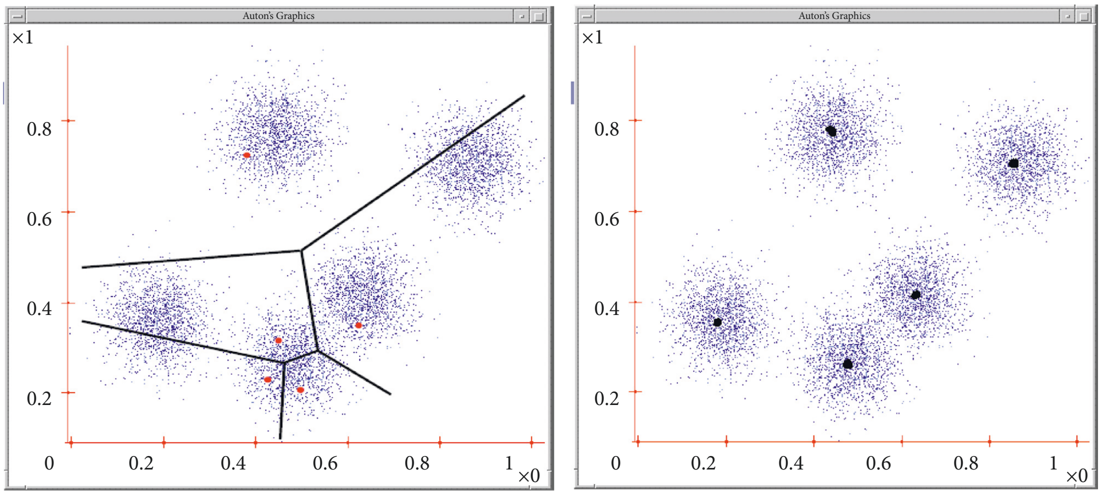

Figure 3: 2nd time clustering inside the grid.

target case and temp list and ascertain whether or not storage conditions are met. If Rule-1 and Rule-2 are met, store the target case; if failed, store the target case on the premise that Rule-1 is met; otherwise, do not store.

(5) If there is no case stored in grid cell after 2nd time clustering, finish retrieval; otherwise, judge whether or not Rule-3 is met, finish if met; otherwise, again cluster, return to 3.

5. Experimental Analysis

The weight calculation of traffic congestion feature attributes can be applied to the retrieval idea of web search engines. The traditional method has been abandoned. This study tried to take text classification as an example, mainly taking the spatial vector model (SVM) as the representation of text.

Firstly, the text is divided into morphemes (word segmentation), and then the selection of eigenvalues and the calculation of the weight of eigenvalues are carried out. Finally, a set of multidimensional traffic congestion feature attribute vectors could be formed.

Second, a table of the attributes of traffic congestion cases is established to integrate the attributes of various traffic congestion cases and is divided into different options. Table 1 is formed by analyzing the traffic congestion text data, which was collected by the research team members from an economic development zone of a city.

All the cases go through data prepossessing from the database, then the indicators are integrated and decomposed. The table of characteristic statistics of traffic congestion cases has been established (shown in Tables 2–4). In this table, attributes are presented as multiple contents, which are diversified (for example, plane intersections show different shapes of intersections) or visibility on hazy days, as shown in Table 5.

There are 70 feature items, which were decomposed from the cases. The computer used in the experiment is configured with a 3.5 GHz Pentium IV CPU, 4 G memory, 250 G and 7200 to IDE hard disk.

According to the attribute content of traffic congestion, these contents can be divided into seven categories, which can be represented as $F = (S_a, S_b, S_c, S_d, S_e, S_f, S_g, S_h)$, where each element represents the following attributes, and the data type of each category is shown in Table 4.

The causes and types of traffic congestion have been expounded in detail. In the database of traffic congestion cases, the causes of traffic congestion can be taken as the focus of the first clustering, and then frequent congestion and occasional congestion are taken as the secondary clustering focus. After the cluster simplification, according to the data types given by attributes, we adopt the method of combining local similarity calculation with global similarity calculation. In the calculation of local similarity, different methods are adopted for different data types, considering different data types.

In the expression of case knowledge, an eigenvector has been established for the eigenvalue attribute of each case, calculating the angle between the two eigenvectors by using the law of cosines. All the weight of the feature value is positive, so the two feature vectors between the cosine values are between 0 and 1. If the cosine value between two feature vectors is close to 1, namely, the two vectors' angle is smaller, the two eigenvectors represented the closer feature value. Conversely, if the cosine value is close to zero, the angle is greater, and the correlation between the two cases is smaller.

Through the previous elaboration, a presentational feature vector has been established for each case, and the angle between the two feature vectors can be calculated by the law of cosines. The formula is as follows:

$$\cos(\theta) = \frac{\sum_{i=1}^{n} (a_i \times b_i)}{\sqrt{\sum_{i=1}^{n} (a_i)^2} \times \sqrt{\sum_{i=1}^{n} (b_i)^2}} = \frac{a^T \bullet b}{\|a\| \times \|b\|}. \qquad (10)$$

The experiment compared the cases of unclustered system 1 and clustered system 2. All cases were divided into 8 sets, and each set was clustered according to $K = 4$. Each set was arranged from low to high according to the number of clustered cases. A judgment analysis was made on the

Table 1: Attributes of traffic congestion cases.

No.	Congestion time	Congestion location	Causes of congestion	Congestion type	Congestion range	Weather	Congestion degree
1.	Morning peak	Trunk road	Normal congestion	Initial congestion	Line	Fine day	Serious
2.	Peak peace	Y-crossing	Sudden congestion	Subsequent congestion	Plane	Heavy rain	Deadlock
3.	Evening peak	T-crossing	Normal congestion	Initial congestion	Point	Fine day	Congestion
……	……	……	……	……	……	……	……
n.	Morning peak	Collector road	Special events	Initial congestion	Line	Light rain	Serious

Table 2: System 2 search results.

Test set	Number of cases	Retrieval time (s)	Number of successful cases retrieved	Retrieval success rate (%)
Test 1	148	0.442	144	97.29
Test 2	154	0.480	149	96.75
Test 3	162	0.486	155	95.56
Test 4	175	0.499	166	94.85
Test 5	179	0.501	169	94.44
Test 6	186	0.587	175	94.08
Test 7	189	0.588	177	93.65
Test 8	197	0.597	186	94.41

Table 3: System 1 search results.

Test set	Number of cases	Retrieval time (s)	Number of successful cases retrieved	Retrieval success rate (%)
Test 1	148	0.445	142	95.94
Test 2	154	0.478	148	96.1
Test 3	162	0.479	154	95.06
Test 4	175	0.497	165	94.28
Test 5	179	0.501	167	93.29
Test 6	186	0.588	172	92.47
Test 7	189	0.589	174	92.06
Test 8	197	0.598	181	91.88

Table 4: Case features and values of case library.

Case attributes	Classification of feature attributes	Types of feature attribute
Sa	Congestion time	Enumeration
Sb	Congestion location	Enumeration
Sc	Congestion causes	Enumeration, numerical type
Sd	Congestion types	Enumeration
Se	Congestion range	Enumeration, numerical type
Sf	Weather	Numerical interval type
Sg	Congestion level	Enumeration

Table 5: Characteristic statistics of traffic congestion cases.

No.	Index
1	Sudden congestion
2	Early peak
……	……
6	Normal congestion
7	Latte peak
……	……
60	Special events
61	Visibility between 100 and 200 m
……	……

retrieval time and success rate, respectively. The average retrieval time is taken 10 times for each collection. The retrieval results are shown in Tables 3 and 2.

By comparing the above charts, it can be found that the retrieval test is carried out on the 8 cases to be tested and is only selected from the test case base. The case from system 2 (clustered) shows a linear and slow increase in the retrieval time as the number of retrieved cases increases. In addition, the retrieval time of system 1 (unclustered) is almost the same as that of the system with clustering, and the success rate of the system with clustering has always been higher and more stable. The retrieval success rate of system 1 is not only lower than system 2 but also less stable than System_2.

6. Conclusion

This paper proposes a traffic congestion case retrieval strategy based on cluster analysis. Through the research on the relevant algorithms of clustering analysis, the shortcomings of the *K*-means algorithm in clustering are improved. Then introduce the concept of Min-cluster, and regard marginal cases as Min-cluster, perform clustering at the margin, select neighboring cases based on the clustering effect, take cases with more matching similarity as the new center point, directly transfer data to the new case, and then adjust the centerfold of the new case. Thus, the quantity of cases at case margins inside the grid is tremendously reduced, so the chances of success in the target case retrieval are greatly improved, and it has been proved through a test that the success rate of the case library retrieval after 2nd time clustering is also greatly improved. It improves the success rate of target case retrieval, expands the scope of case solutions in the decision-making system, and enhances the reliability and flexibility of decision-making selection. The next step is to further optimize the case set structure and the relevant parameters and to improve the learning ability of the system.

Colleagues and authors try to apply the optimized algorithm to the daily management of traffic congestion relief. Experiments show that the clustering traffic congestion case set has improved the retrieval accuracy and time.

Data Availability

The data used to support the findings of this study are available from the corresponding author upon request.

Conflicts of Interest

The authors declare that they have no conflicts of interest.

Acknowledgments

This research was supported by the National Natural Science Foundation of Anhui (Grant no. KJ2021ZD0126 and SK2020A0930).

References

[1] W. Bannour, M. Maalel, H. Ben Ghezala, and B. Ghezala, “Case-based reasoning for crisis response: case representation and case retrieval,” *Procedia Computer Science*, vol. 176, pp. 1063–1072, 2020.

[2] Z. Zhai, J. F. Martínez Ortega, B. V, and N. Lucas Martínez, “An associated representation method for defining agricultural cases in a case-based reasoning system for fast case retrieval,” *Sensors*, vol. 19, no. 23, p. 5118, 2019.

[3] Y. Guo and K. Wu, “Research on case retrieval of Bayesian network under big data,” *Data & Knowledge Engineering*, vol. 118, no. 12, pp. 1–13, 2018.

[4] D. C. Corrales, A. Ledezma, and J. Carlos Corrales, “A case-based reasoning system for recommendation of data cleaning algorithms in classification and regression tasks,” *Applied Soft Computing*, vol. 90, Article ID 106180, 2018.

[5] Xi Lin, W. Zhang, H. Qiu, and B. Peng, “Research on auxiliary diagnosis of power communication field operation and maintenance based on CBR,” *Telecommunications and Radio Engineering*, vol. 79, no. 19, pp. 1761–1771, 2020.

[6] Y. Peng, J. Wang, and L. Jiao, “A novel text retrieval algorithm for public crisis cases,” *Chinese Journal of Electronics*, vol. 28, no. 4, pp. 712–717, 2019.

[7] H. Wang, B. Sun, and X. Shen, “Hybrid similarity measure for retrieval in case-based reasoning systems and its applications for computer numerical control turret design,” *Proceedings of the Institution of Mechanical Engineers - Part B: Journal of Engineering Manufacture*, vol. 232, no. 5, pp. 918–927, 2018.

[8] H. Zhang and G. L. Dai, “The strategy of traffic congestion management based on case-based reasoning,” *International Journal of System Assurance Engineering and Management*, vol. 10, no. 1, pp. 142–147, 2019.

[9] H. Zhang and G. L. Dai, “Research on traffic decision making method based on image analysis case based reasoning,” *Optik*, vol. 158, pp. 908–914, 2018.

[10] S. Stephen and L. Du, “Optimal information perturbation for traffic congestion mitigation: Gaussian process regression and optimization,” *Transportation Research Part C: Emerging Technologies*, vol. 138, 2022.

[11] H. Nizar, K. Ali, and F. Zeinab, “Intelligent transportation systems to mitigate road traffic congestion,” *Intelligenza Artificiale*, vol. 15, no. 2, pp. 91–104, 2022.

[12] A. Khattak and A. Kanafani, “Case-based reasoning: a planning tool for intelligent transportation systems,” *Transportation Research Part C: Emerging Technologies*, vol. 5, pp. 267–288, 2014.

[13] A. W. Sadek, M. J. Demetsky, and B. L. Smith, “Case-based reasoning for real-time traffic flow management,” *Computer-Aided Civil and Infrastructure Engineering*, vol. 14, no. 5, pp. 347–356, 2014.

[14] A. J. Whitsitt and L. E. Travis, “Traffic route generation and adaptation using case-based reasoning,” *ITS Journal - Intelligent Transportation Systems Journal*, vol. 3, no. 3, pp. 181–204, 1996.