International Journal of Software and Informatics, ISSN 1673-7288
http://www.ijsi.org, ijsi@iscas.ac.cn, +86-10-62661048
IJSI, 2021, 11(1): 31−57, doi: 10.21655/ijsi.1673-7288.00239

International Journal of Software and Informatics

Research Article

Time Series Data Cleaning under Multi-speed Constraints

Fei Gao (高菲) [1], Shaoxu Song (宋韶旭) [1,2,3], Jianmin Wang (王建民) [1,2,3]

[1] (School of Software, Tsinghua University, Beijing 100084, China)

[2] (National Engineering Laboratory for Big Data Software, Beijing 100084, China)

[3] (Beijing National Research Center for Information Science and Technology (Tsinghua University), Beijing 100084, China)

Corresponding author: Shaoxu Song, sxsong@tsinghua.edu.cn

Abstract As the basis of data management and analysis, data quality issues have increasingly become a research hotspot in related fields, which contributes to optimization of big data and artificial intelligence technology. Generally, physical failures or technical defects in data collectors and recorders will cause anomalies in collected data. These anomalies will strongly impact on subsequent data analysis and artificial intelligence processes; thus, data should be processed and cleaned accordingly before application. Existing repairing methods based on smoothing will cause a large number of originally correct data points being over-repaired into wrong values. And the constraint-based methods such as sequential dependency and SCREEN cannot accurately repair data under complex conditions since the constraints are relatively simple. A time series data repairing method under multi-speed constraints is further proposed based on the principle of minimum repairing. Then, dynamic programming is used to solve the problem of data anomalies with optimal repairing. Specifically, multiple speed intervals are set to constrain time series data, and a series of candidate repairing points are formed for each data point according to the speed constraints. Next, the optimal repair solution is selected from these candidates based on the dynamic programming method. With regard to the feasibility study of this method, an artificial dataset, two real datasets, and another real dataset with real anomalies are employed for experiments in case of different rates of anomalies and data sizes. Experimental results demonstrate that, compared with the existing methods based on smoothing or constraints, the proposed method has better performance in terms of RMS errors and time cost. In addition, the investigation of clustering and classification accuracy with several datasets reveals the impact of data quality on subsequent data analysis and artificial intelligence. The proposed method can improve the quality of data analysis and artificial intelligence results.

Keywords time series; multi-speed constraints; data cleaning; dynamic programming

Citation Gao F, Song SX, Wang JM. Time series data cleaning under multi-speed constraints. International

This is the English version of the Chinese article “多区间速度约束下的时序数据清洗方法. 软件学报, 2021, 32(3): 689−711. doi: 10.13328/j.cnki.jos.006176”.

Foundation item: National Natural Science Foundation of China (61806044, U2001211, 61932004, 61732003); Beijing Institute of Technology Research Fund Program for Young Scholars (3070012222010)

Received 2020-07-03; Revised 2020-09-02; Accepted 2020-11-06; IJSI published online 2021-03-31

Journal of Software and Informatics, 2021, 11(1): 31−57. http://www.ijsi.org/1673-7288/00239.htm

1 Introduction

Amid the development and popularization of information technology, massive data have been accumulated in all walks of life through corresponding information systems, providing basic data support for big data and artificial intelligence technology. Data management and analysis technology is indispensable as a basic support to give full play to the advantages and improve the efficiency and application of the big data and artificial intelligence technology. As is known to all, sensors, terminal recorders and other devices will be affected by subjective and objective factors during acquisition, transmission and recording of data. Within physical and technical constraints, the final data quality will be impaired. Then it cannot accurately characterize the real world, failing to promote the optimization of artificial intelligence technology. Removing anomalies for higher data quality can further optimize artificial intelligence in data link through data management and analysis, advancing the development in the artificial intelligence field.

1.1 Background

At the moment, the cost and risk caused by poor data quality should not be underestimated, and it has remained as an important subject in the field of data management to effectively identify and repair anomalies in data. With the progress in technology, the cost of data storage and transmission has dropped sharply. Meanwhile, the development of big data and artificial intelligence technology enlightens people as to constantly tap the enormous potential of data. Human society, especially in the industrial field, tends to store the data records that can be generated as much as possible. Usually, they are mostly time series data, namely a series of data points including time stamps.

Time series data generally exists in people’s daily life and industrial fields, such as driving routes, temperature changes, and stock trends. Since the points in most time series data vary with time, Zhang *et al.*[1] put forward the SCREEN method based on speed constraints to repair anomalies in time series data. In this method, the speed constraint range [$s^{\min}$, $s^{\max}$] ([minimum speed, maximum speed]) restrains the change speed of data, and the data points beyond the speed constraint range are regarded as anomalies and then repaired. However, this method is only applicable to a single-speed constraint, namely that the range of speed is between a maximum and a minimum. In actual data, the speed change of data can be subject to multiple constraint intervals. For instance, the speed is likely to change in the interval of $[s_1^{\min}, s_1^{\max}]$ or $[s_2^{\min}, s_2^{\max}]$ ($s_1^{\max} < s_2^{\min}$). If the change speed is only constrained to $[s_1^{\min}, s_1^{\max}]$, many normal points constrained by $[s_2^{\min}, s_2^{\max}]$ will be treated as anomalies for repairing. Similarly, if the change speed is only restrained to $[s_2^{\min}, s_2^{\max}]$, the normal points constrained by $[s_1^{\min}, s_1^{\max}]$ will also be regarded as anomalies, resulting in excessive cleaning. In addition, if the change speed is limited to $[s_1^{\min}, s_2^{\max}]$, the anomalies between ($s_1^{\min}, s_2^{\max}$) will be regarded as normal points free of repair. In either case, the quality of data repair will be considerably reduced.

Example 1: The company supervises the fuel consumption data of vehicles for analysis of the consumption situation, encouraging safe drive and reducing the cost. In this time series data, two behaviors, fuel consumption and refueling, are included, as shown in Figure 1. Since the vehicle consumes fuel during driving, the oil level in the fuel tank presents a downward trend as a whole. Nevertheless, it is inevitable that the vehicle will bump during driving, and the measurement of oil level in the oil tank will oscillate within a small range. Therefore, in the process of fuel consumption, the oil level maintains a dynamic downward trend. The specific speed constraint is [−1, 1], namely that the change in the oil level per unit time fluctuates by 1 cm up and down (according to the real data, the unit time is set to 5 s in this example). On the contrary, in the course of refueling, the oil level grows sharply. The specific speed constraint is [20,70], namely that the oil level increases by 20–70 cm per unit time during refueling. The speed of data change will be abnormal if that between two data points in a given time window is outside the ranges of [−1,1] and [20,70], and there must be abnormal data in these two data points. As shown in Figure 1, blue data indicate anomalies.

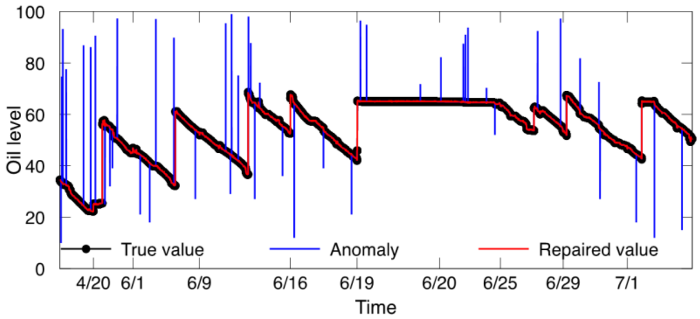


Figure 1. Example of fuel consumption data

To fully illustrate the difference between multi-speed constraints and single-speed constraint, this paper selects 100 data points from 34 220 in the above-mentioned fuel consumption dataset for repair. In this data, if the speed constraint is set as [−1,1] without careful consideration, the refueling behavior is treated as abnormal data to be repaired, as shown in Figure 2. Due to the refueling behavior at 15:09, many normal values are misjudged as abnormal for over-repair; similarly, if the speed constraint is [20,70], substantial fuel consumption behaviors will be regarded as abnormal data, also leading to over-repair. If the speed constraint [−1, 70] is adopted, most data is regarded as normal; however, the anomalies within the speed constraint of (1,20) cannot be accurately identified and repaired. As shown in Figure 2, owing to an anomaly at 15:12, subsequent correct values are falsely repaired.

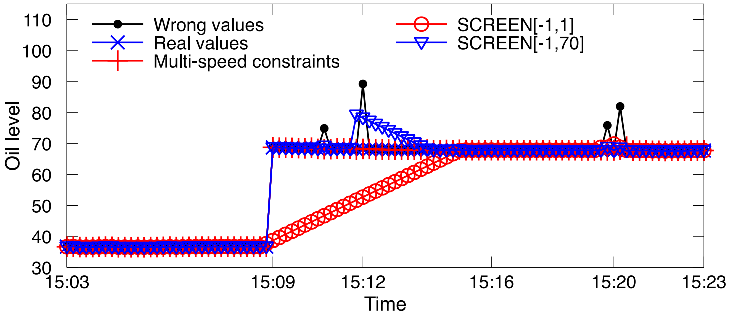


Figure 2. Repairing with SCREEN and multi-speed constraints

Consequently, this paper mainly studies the quality of time series data and repairs the anomalies in time series data through multi-speed constraints.

1.2 Major contributions

The primary contributions of this paper are as follows:

(1) Multi-speed constraints and repaired results are defined. The speed constraint mentioned in this paper is not limited to a single range between the maximum and the minimum; instead, it has multiple intervals of speed, making the constraints more specific and precise. The repaired results are defined specific to a given window, which can be flexibly extended to the whole time

series in application, thus making the repair method more widely applicable.

(2) A time series data repairing method under multi-speed constraints is proposed, and the corresponding algorithm is given. Within multi-speed constraints, the repairing range and candidate repairing points of the objects are given. Besides, with regard to each candidate point, a more specific repairing range is generated for the subsequent time in the given window, and then the subsequent candidate points are further screened to determine the final candidate repairing points. Finally, the candidate points determined at each time in the given window are stored based on the graph, so as to be repaired with the dynamic programming method.

(3) The theory and proof are put forward for the dynamic programming-based repair with multi-speed constraints, providing a theoretical support for the repairing method. Theory 1 demonstrates that an optimal repairing path can be identified at the candidate repairing point, minimizing the sum of repairing distance in the window. In addition, when the candidate repairing point generated by the subsequent point relative to the current point is not in the final repairing range, Theory 2 also reveals that the corresponding constraint point can be selected as the new candidate repairing point to ensure the minimum repairing distance.

(4) The dynamic programming-based repair is compared with SCREEN in a special case of single-speed constraint. In this case, the results from the method under multi-speed constraints proposed in this paper are consistent with those obtained by SCREEN in determining the constraint range and candidate repairing points. Not worthily, the repair method proposed in this paper will be equivalent to or better than SCREEN, because the method under multi-speed constraints identifies the optimal repairing path according to dynamic programming in the given candidate points.

(5) Experiments are performed with an artificial dataset, two real datasets and a dataset with real anomalies in terms of RMS errors, clustering and classification accuracy, and time cost, and the proposed method is compared with existing methods including SCREEN. The results demonstrate that the method under multi-speed constraints in this paper performs the best in repairing, with a good trade-off between the effect and time cost. Additionally, this paper verifies the classification accuracy by multiple time series datasets with classification labels. From the results of accuracy, the method under multi-speed constraints can provide a solid data support for subsequent research and processing including data analysis and artificial intelligence.

1.3 Structure

Section 2 of this paper provides the basic definitions related to this study. It explains the time series, multi-speed constraints and repaired results, and offers the formal definition of the problem, namely repairing conditions and goals. In Section 3, a dynamic programming-based repair method under multi-speed constraints is developed, including specific description, theoretical proof, special case analysis and specific algorithms. In Section 4, the proposed method is compared with the existing methods regarding repair effect, time cost, and clustering and classification accuracy through an artificial dataset, two real datasets and a dataset with real anomalies. In particular, the classification accuracy is compared with multiple datasets. The related work is introduced in Section 5, and finally, the work of this paper is analyzed and summarized in Section 6.

2 Fundamental Definition

As the data support for big data and artificial intelligence technology, industrial big data are becoming a hot spot in data-related research fields, and this paper mainly studies the industrial time series data. For convenience of description, this section provides definitions of time series, time stamp, speed constraints, multi-speed constraints and repaired results.

2.1 Time series

A time series refers to a series of data points containing time stamps. To be specific, in a data series $x = x[1], x[2], \dots$, the data point $x[i]$ indicates the i^{th} data point, and each data point $x[i]$ has a time stamp $t[i]$. $x[i]$ is abbreviated as x_i and $t[i]$ as t_i for simplicity.

2.2 Multi-speed constraints

In a constraint interval of speed $s_r = [s_r^{\min}, s_r^{\max}]$, $s_r^{\min}$ is the minimum speed, while $s_r^{\max}$ is the maximum speed.

The multi-speed constraint S refers to a set of constraint intervals s_r (r=1, 2, …, m), namely S={s_1, s_2, …, s_m}. In a given time window w, if the time series x follows the multi-speed constraint S, any data points x_i and x_j satisfy S. And if x_i and x_j satisfy S, any data points x_i and x_j in the window w satisfy a certain-speed constraint s_r among the multi-speed constraints, namely

$$0 < t_j - t_i \le w, s_r^{\min} \le \frac{x_j - x_i}{t_j - t_i} \le s_r^{\max}$$

In Figure 3, in a given window w, the constraint interval S includes two sub-intervals $s_1 = [s_1^{\min}, s_1^{\max}]$ and $s_2 = [s_2^{\min}, s_2^{\max}]$, and a data point pair (x_1, x_2) is within the constraint interval $s_1 = [s_1^{\min}, s_1^{\max}]$, namely

$$s_1^{\min} \le \frac{x_2 - x_1}{t_2 - t_1} \le s_1^{\max}$$

Similarly, (x_1, x_3) is within the constraint interval $s_2 = [s_2^{\min}, s_2^{\max}]$. As such, the above two pairs of data points are all under the multi-speed constraint S, but (x_1, x_4) neither satisfies s_1 nor s_2, namely that the data pair (x_1, x_4) does not satisfy S.

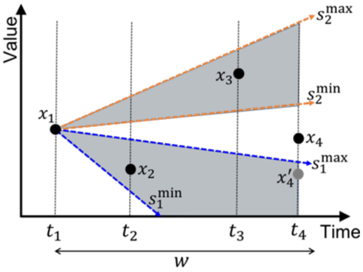


Figure 3. Example of multi-speed constraints

2.3 Repaired results

The repaired result x' means that within a given window w, the data point x_i on the time stamp t_i is repaired into x_i', and the time stamp remains unchanged after repair, i.e., $t_i' = t_i$. According to the principle of minimum repairing, the final repairing distance can be defined as

$$\Delta(x, x') = \sum_{i=k}^{k+\omega} |x_i - x_i'|, \quad i = k, k+1, \cdots, k+\omega, \quad 0 \le t_k \le t_i \le t_k + w$$

From Figure 3, x_4 is repaired into x_4', and the time stamp is still t_4. The final repairing distance in this window is $\Delta(x, x') = \sum_{i=1}^{4} |x_i - x_i'| = |x_4 - x_4'|$.

2.4 Problems

Given a time series x, the time window is w with ω+1 data points and the starting point x_k. The speed constraint range of each x_j in the window and the multi-speed constraint are illustrated as S={s_1, s_2, …, s_m}, $s_r = [s_r^{\min}, s_r^{\max}]$, where r=1, 2,…, m. The repair under multi-speed constraints refers to finding a repair result x' in the window w. Then each point in the repaired window meets the multi-speed constraints and the repairing distance is the minimum, namely

$$\min \sum_{j=k}^{k+\omega} |x_j - x'_j|, \quad j = k, k+1, \cdots, k+\omega, \quad 0 \le t_k \le t_j \le t_k + w$$

where the speed constraint range of each x_j in the window is obtained by multi-speed constraints on other points before x_k and in the same window of x_j. If no data point before x_k shares the same window with x_j, then the constraint range of x_j is not set. Specific methods are detailed in Section 3.1.

3 Dynamic Programming-based Repairing

According to the above definitions, this section further elaborates the proposed repairing method. Overall, if the time series data is repaired under speed constraints, the range of multi-speed constraints on data should be first determined. In addition, the candidate repairing points of each data point can be given for repair selection. At last, the final repairing point is selected by the dynamic programming method to complete the repair.

3.1 Speed constraint range

Parameters are given as the multi-speed constraint S, the window w, the starting data point x_k in the window and each point in the window, x_k, x_{k+1}, x_{k+2}, …, $x_{k+\omega}$, with $0<t_{k+\omega}-t_k\le w$. This paper determines the corresponding speed constraint range to figure out whether each data point in the window x_j ($t_k\le t_j\le t_k+w$) and the aforementioned data points x_i ($t_i<t_j\le t_j-w$) in the same window as them meet the multi-speed constraints in Section 2.2.

To illustrate multi-speed constraints more distinctly, this section first studies the data point x_j. The constraint range of x_j is solved by other data points x_i ($t_j-w\le t_i<t_k \le t_k + w$) in its same window but not in the given window w. Then the speed constraint range of each x_j in the window mentioned in Section 2.4 is solved as the following.

For convenience of description, the definitions of x_{k-n_j} are given in this paper. When $x_j(t_k\le t_j\le t_k+w)$ is the last point in a window, x_{k-n_j} is the first point in the window.

x'_i ($i=k-1$, $k-2$, …, $k-n_k$) is the previous point of x_k in the same window(when x_k is the last point in a window, x_{k-n_k} is the first point in the window, $0 < t_k - t_i \le t_k - t_{k-n_k} \le w$). According to Formulas (1)–(2), a speed constraint range $[x^{\min}_{i,k,r}, x^{\max}_{i,k,r}]$ is generated by x'_i for x_k. Then with Formula (3), the speed constraint ranges for each x'_i are intersected to generate the constraint range $X^{\text{const}}_{j,r}$, and finally with Formula (4), the multi-speed constraint ranges of x_k are combined into the set X^{const}_k. It is indicated by the gray area in Figure 4.

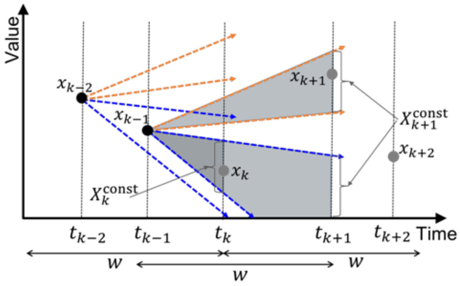


Figure 4. Range of multi-speed constraints

Similarly, x'_i ($i=k-1,k-2,\ldots,k-n_k$) is the previous point ofx_{k+1}(except for the points in the given window w) in the same window (when x_{k+1} is the last point in a window, $x_{k-n_{k+1}}$ is the first point in this window, $0 < t_{k+1} - t_i \le t_{k+1} - t_{k-n_{k+1}} \le w$). Then the constraint range $X_{k=1}^{\text{const}}$ will be generated by x_i' for x_{k+1}, indicated by the blue area of Figure 4.

x_i' (i=k−1, k−2, …, k−n_{k+2}, $0 < t_{k+2} - t_i \le t_{k+2} - t_{k-n_{k+2}} \le w$) is the previous point of x_{k+2} (except for the points in the given window w) in the same window. Then the constraint range X_{k+2}^{const} will be generated by x_i' for x_{k+2}. In the example of Figure 4, the aforementioned points in the same window with x_{k+2} are all in the given window w, so the speed constraint range of x_{k+2} is not generated at this time.

In the same way, the speed constraint range of all points (x_k, x_{k+1}, x_{k+2}, …, $x_{k+\omega}$) in the window w is determined:

$$x_{i,j,r}^{\min} = x_i' + s_r^{\min}(t_k - t_i) \tag{1}$$

$$x_{i,j,r}^{\max} = x_i' + s_r^{\max}(t_k - t_i) \tag{2}$$

$$X_{j,r}^{\,const} = \bigcap_{i=k-n_k}^{k-1} [x_{i,j,r}^{\min}, x_{i,j,r}^{\max}] \tag{3}$$

$$X_j^{\text{const}} = \bigcup_{r=1}^{m} X_{j,r}^{\text{const}} \tag{4}$$

where $0 < t_j - t_i \le t_j - t_{k-n_j} \le w, t_i < t_k \le t_j \le t_k + w, k - n_j \le i \le k-1 < k \le j \le k+\omega$.

3.2 Candidate repairing points

The relationship between data points and speed constraints reveals that the speed constraint S can be combined with the data point x_i (t_k≤t_i<t_k+w) to constrain the range of subsequent points x_j (t_k≤ t_j<t_k+w). Similarly, the speed constraint S can also be integrated with x_j (t_k≤ t_j<t_k+w) to reversely deduce the approximate range of the aforementioned data points x_i (t_k≤ t_i<t_k + w) in the same window, namely the candidate repairing points of x_i. Next, this paper gives a specific method to determine the candidate repairing points.

x_i(x_{k+1},x_{k+2},…,$x_{k+\omega}$), t_k<t_i≤$t_{k+\omega}$≤t_k+w indicates the subsequent points of x_k in the given window w, and the candidate point set $X_k^{\min} \cup X_k^{\max} \cup \{x_k\}$ of x_k can be generated according to Formulas (5)–(6):

$$X_i^{\min} = \{x_j + s_r^{\min}(t_i - t_j) \mid t_k \le t_i < t_j \le t_k + w, k \le i < j \le k+\omega, 1 \le r \le m\} \tag{5}$$

$$X_i^{\max} = \{x_j + s_r^{\max}(t_i - t_j) \mid t_k \le t_i < t_j \le t_k + w, k \le i < j \le k+\omega, 1 \le r \le m\} \tag{6}$$

In the same way, the candidate point set $X_{k+1}^{\min} \cup X_{k+1}^{\max} \cup \{x_{k+1}\}$ of x_{k+1} in this window can be generated on the basis of its subsequent points (x_{k+2}, …, $x_{k+\omega}$). As such, the candidate repairing point set X_i of each data point x_i(i =k, k+1, …, k+ω) in the window w can be obtained, and the candidate repairing point set $X_{k+\omega}$ of $x_{k+\omega}$ only includes the point itself, as indicated in Figure 5.

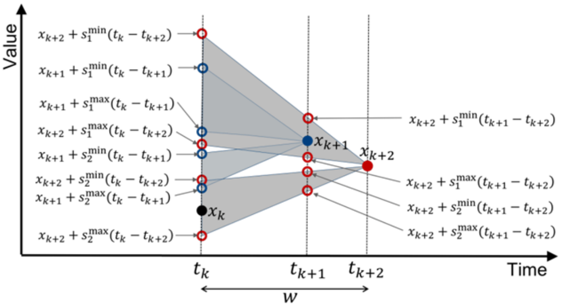

Value
$x_{k+2} + s_1^{\min}(t_k - t_{k+2})$
$x_{k+1} + s_1^{\min}(t_k - t_{k+1})$
$x_{k+1} + s_1^{\max}(t_k - t_{k+1})$
$x_{k+2} + s_1^{\max}(t_k - t_{k+2})$
$x_{k+1} + s_2^{\min}(t_k - t_{k+1})$
$x_{k+2} + s_2^{\min}(t_k - t_{k+2})$
$x_{k+1} + s_2^{\max}(t_k - t_{k+1})$
x_k
$x_{k+2} + s_2^{\max}(t_k - t_{k+2})$
x_{k+1}
x_{k+2}
$x_{k+2} + s_1^{\min}(t_{k+1} - t_{k+2})$
$x_{k+2} + s_1^{\max}(t_{k+1} - t_{k+2})$
$x_{k+2} + s_2^{\min}(t_{k+1} - t_{k+2})$
$x_{k+2} + s_2^{\max}(t_{k+1} - t_{k+2})$
t_k
t_{k+1}
t_{k+2}
Time
w

Figure 5. Capture of candidate repairing points in a given window

Then, according to the speed constraint range X_i^{const} in Section 3.1, the candidate repairing points are re-screened by Formula (7), and the candidate repairing point set X_i within the speed constraint range is finally determined:

$$X_i = \{x_i' \mid x_i' \in X_i^{\min} \cup X_i^{\max} \cup \{x_i\}, x_i' \in X_i^{\mathrm{const}}\} \tag{7}$$

In Figure 6, the empty circles represent the candidate points determined by the above method, while the solid circles depict the original data points. The gray points are invalid candidate points beyond the constraint range, and other points are valid.

Theory 1. Under multi-speed constraints, among x_k and its candidate repairing points, it is certain to find the optimal repairing solution x_k'.

Proof: $\omega=|\{i|t_k<t_i \le t_k + w, 1\le i \le n\}|$ is set as the total number of data points except x_k in the window starting from x_k, and there are $2\times\omega\times r+1$ points in the candidate point set X_k of x_k. These points are sorted to get $c_1, \ldots, c_{2*\omega*r+1}$, and $c_j \le c_{j+1}, j=1, \ldots, 2\times\omega\times r+1$.

If x_k' is not a candidate point, namely $x_k' \notin X_k$, then another repair method can be developed on the basis of candidate points x_k'' to minimize the repairing distance. For convenience, this paper gives the following definition:

$d_{i,j}^{\max} = c_j + s_{r_{\max}}^{\max}(t_i - t_k), d_{i,j}^{\min} = c_j + s_{r_{\min}}^{\min}(t_i - t_k), x_{i,k}^{\max} = x_k' + s_{r_{\max}}^{\max}(t_i - t_k), x_{i,k}^{\min} = x_k' + s_{r_{\min}}^{\min}(t_i - t_k)$, where $s_{r_{\max}}^{\max}$ is the maximum speed in the range of multi-speed constraints; $s_{r_{\min}}^{\min}$ is the minimum speed in the range, $t_k<t_i \le t_k+w$.

Firstly, it is proved that the repaired result x_k' must be within the candidate set, namely $c_1 \le x_k' \le c_{2*\omega*r+1}$. From Figure 7, it is certain that $x_i \ge d_{i,1}^{\max}$; otherwise, other candidate points will be introduced between x_k' and c_1. For any $x_k' < c_1$, other x_i points in the window will be repaired to $x_{i,k}^{\max}$, and the repairing distance is obviously longer than that to $d_{i,1}^{\max}$, compromising the principle of minimum repairing. When $x_k' > c_{2*\omega*r+1}$, similar results will be obtained.

Second, it is assumed that $c_j \le x_k' \le c_{j+1}, j \in [1, 2*\omega*r+1]$, namely that x_k' is between two continuous candidate repairing points, with specific cases as follows:

(1) If x_i is not changed after repairing, namely $x_i' = x_i$, the repairing distance of x_i is 0 at this time, as shown in Figure 8.

(2) If it is repaired as $x_i' = x_{i,k}^{\max}$, then $x_i \ge d_{i,j+1}^{\max}$; otherwise, new candidate points will be introduced betweenc_j and c_{j+1}.It can also be repaired as $x_k'' = c_j$ or $x_k'' = c_{j+1}$, and then $x_i'' = d_{i,j}^{\max}$ or $x_i'' = d_{i,j+1}^{\max}$, with the corresponding repairing distance as $|x_i'' - x_i| = |x_i' - x_i| - c_j + x_k'$, or $|x_i'' - x_i| = |x_i' - x_i| - c_{j+1} + x_k'$, as shown in Figure 9.

(3) Similarly, if it is repaired as $x_i' = x_{i,k}^{\min}$, then $x_i \le d_{i,j+1}^{\min}$, or new candidate points will be introduced betweenc_j and c_{j+1}. It can also be repaired as $x_k'' = c_j$ or $x_k'' = c_{j+1}$, and then $x_i'' = d_{i,j}^{\min}$ or $x_i'' = d_{i,j+1}^{\min}$, with the corresponding repairing distance as $|x_i'' - x_i| = |x_i' - x_i| + c_j - x_k'$, or $|x_i'' - x_i| = |x_i' - x_i| + c_{j+1} - x_k'$, as shown in Figure 10.

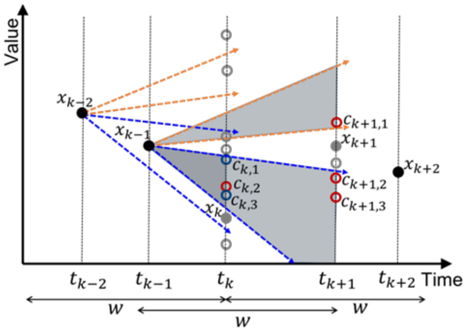


Figure 6. Generation of candidate point set X_i according to range X_i^{const}

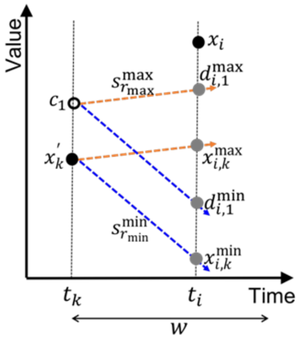


Figure 7. Impossible case of x'_k smaller than the minimum candidate c_1, $x'_k \le c_1$

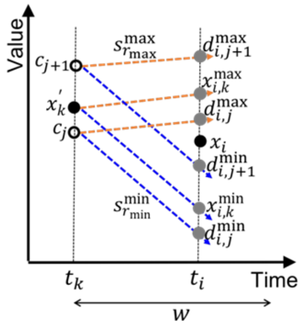


Figure 8. x'_k between two continuous candidates, $c_j \le x'_k \le c_{j+1}$, without repairing x_i

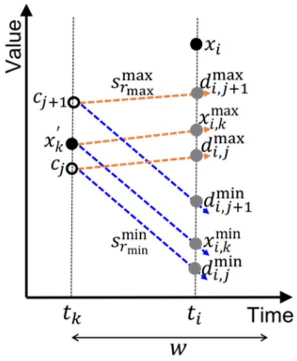


Figure9. x'_k between two continuous candidates, $c_j \le x'_k \le c_{j+1}, x_i \ge d_{i,j+1}^{\max}$

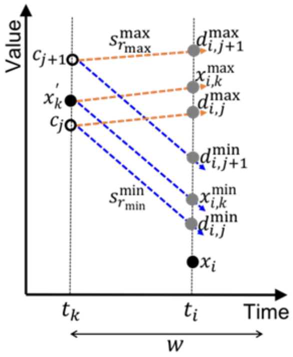


Figure 10. x'_k between two continuous candidates, $c_j \le x'_k \le c_{j+1}, x_i \le d_{i,j}^{\min}$

x_i in the above Cases (2) and (3) can be counted to calculate the total repairing distance in the window.

(a) When the number of x_i in Case (2) and Case (3) is the same, either of c_j and c_{j+1}, which is closer to x_k, is selected as x''_k . When c_j is chosen, the total repairing distance is $\Delta(x,x'')=\Delta(x,x')-(x'_k-c_j)<\Delta(x,x')$; as c_{j+1} is chosen, the total repairing distance is

$$\Delta(x,x'')=\Delta(x,x')-(c_{j+1}-x'_k)<\Delta(x,x');$$

(b) When the number of x_i in Case (2) is greater than that in Case (3), c_{j+1} serves as x''_k. Then the total repairing distance is $\Delta(x,x'')\le\Delta(x,x')-c_{j+1}+x'_k+|c_{j+1}-x'_k|<\Delta(x,x')$;

(c) When the number of x_i in Case (2) is smaller than that in Case (3),c_j serves as x''_k. Then the total repairing distance is $\Delta(x,x'')\le\Delta(x,x')+c_j-x'_k+|c_j-x'_k|<\Delta(x,x')$.

In summary, $x''_k=c_j$ or $x''_k=c_{j+1}$ can be employed to obtain the repaired result x'', making $\Delta(x,x'')\le\Delta(x,x')$.

Similar results will be obtained for other points $x_i(x_{k+1}, x_{k+2}, \ldots, x_{k+\omega})$, $t_k<t_i\le t_{k+\omega}\le t_k+w$ in the window.

3.3 Repairing path

Previous discussion reveals that in the window w starting from x_k, each data point x_i has a set of candidate repairing points within a given speed constraint range. The number of candidate points in this set is as η_i.

As in Section 3.1, parameters are given as the multi-speed constraint S, the window w, the starting data point x_k in the window, and each point $x_j(t_k \le t_j \le t_k + w)$ in the window has a relation of multi-speed constraint with the previous point $x_i(t_i<t_j \le t_j - w)$ in the same window; the candidate repairing points of each data point can also place a speed constraint range for its subsequent points in the window w, as described below.

Each candidate repairing point c_{i,d_i} of x_i can generate a speed constraint range $[x^{\min}_{d_i,j,r}, x^{\max}_{d_i,j,r}]$ for the subsequent point x_j in the same window w according to Formulas (8)–(9):

$$x^{\min}_{d_i,j,r}=c_{i,d_i}+s^{\min}_r(t_j-t_i) \tag{8}$$

$$x^{\max}_{d_i,j,r}=c_{i,d_i}+s^{\max}_r(t_j-t_i) \tag{9}$$

where $t_k \le t_i<t_j \le t_k + w$, $k\le i <j\le k+\omega$,$1\le d_i \le \eta_i$.

According to $X^{\text{const}}_{j,r}$ in Formula (3), a new speed constraint range X^{const}_{j,d_i} for x_j is determined according to Formulas (10)–(12):

$$X^{\text{const}}_{j,d_{i-1},r}=X^{\text{const}}_{j,r}, i=k \tag{10}$$

$$X^{\text{const}}_{j,d_i,r}=[x^{\min}_{d_i,j,r}, x^{\max}_{d_i,j,r}]\cap X^{\text{const}}_{j,d_{i-1},r} \tag{11}$$

$$X^{\text{const}}_{j,d_i}=\bigcup\nolimits_{r=1}^{m} X^{\text{const}}_{j,d_i,r} \tag{12}$$

where $t_k \le t_i<t_j \le t_k + w$, $k \le i <j \le k + \omega$, $1 \le d_i \le \eta_i$.

Theory 2. In the window w with x_k as the starting point, if the data point x_k has no candidate repairing point in the constraint range $X^{\text{const}}_{j,d_{j-1}}$, the constraint point closest to the original x_j can be designated as the candidate. The speed constraint point refers to the boundary point of the constraint range determined for x_j in the above $X^{\text{const}}_{j,d_{j-1}}$, and the repairing distance is the smallest. The candidate repairing point set is updated as $X^{\text{cand}}_{j,d_{j-1}}$:

$$X^{\text{cand}}_{j,d_{j-1}}=\begin{cases} X_j, & X_j\ne\varnothing \\ \{x'_j\}, & X_j=\varnothing \end{cases} \tag{13}$$

where $x'_j \in X^{\text{const}}_{j,d_{j-1}}$, and the distance from x'_j to x_j is the smallest, namely

$$\Delta(x_j,x'_j)=\min\{\Delta(x_j,x'_j)\mid x'_j\in X^{\text{const}}_{j,d_{j-1}}\},\ t_k\le t_{j-1}<t_j\le t_k+w,\ k\le j-1<j\le k+\omega,\ 1\le d_{j-1}\le \eta_{j-1}$$

Proof: Theory 1 demonstrates that there must be an optimal repairing solution when the candidate repairing points of x_j are within the range $X^{\text{const}}_{j,d_{j-1}}$.

When candidate repairing points of x_j are beyond the range $X^{\text{const}}_{j,d_{j-1}}$, the boundary point x'_j of the speed constraint range, which is closest to x_j, can be selected according to Formula (13), namely $\Delta(x_j,x'_j)=\min\{\Delta(x_j,x'_j)\mid x'_j\in X^{\text{const}}_j\}$. It is evident that within the range $X^{\text{const}}_{j,d_{j-1}}$, the distance from other points to original x_j is greater than $\Delta(x_j,x'_j)$, so in the repairing path with the previous repairing points determined, the selected x'_j is the closest to the original data point under the constraints. At this time, the repairing distance is the minimum.

If the repairing range generated by this candidate point for subsequent points includes the existing candidate points, the optimal repairing path can still be selected according to this candidate point; on the contrary, if the existing candidate points are not within the repairing range, the candidate repairing points of the subsequent points can be generated according to Theory 2, and the repairing distance of this path is the smallest.

In summary, there is an optimal repairing path among the candidate repairing points determined by this theory, which minimizes the repairing distance.

In Figure 11, from moment t_k, for data point x_i ($t_k \le t_i < t_k + w$), any of its candidate repairing point c_{i,d_i} generates a repairing range of $[x^{\min}_{d_i,j,r}, x^{\max}_{d_i,j,r}]$ for subsequent x_j ($t_k \le t_i < t_j \le t_k + w$) according to Formulas (8)–(9). This range will intersect with the above X^{const}_j according to Formulas (10)–(12) to determine a new candidate repairing range X^{const}_{j,d_i}. If one or more candidate repairing points $c_{i+1,d_{i+1}}$ fall in this range at the next moment t_{i+1}, then one or more candidate points are connected with the candidate points c_{i,d_i} selected by t_i into one or more edges. The weight of the edge is the distance between the candidate point $c_{i+1,d_{i+1}}$ and the original data point x_{i+1}, i.e., $\Delta(x_{i+1},c_{i+1,d_{i+1}})$. If there is no candidate repairing point in this range in the next moment, then the speed constraint point closest to the original x_i is designated as the candidate repairing point $c_{i+1,d_{i+1}}$ according to Theory 2. To be specific, the speed constraint point refers to the boundary point of the constraint range determined for x_i above, and c_{i,d_i} is connected with $c_{i+1,d_{i+1}}$ to form an edge. Similarly, the weight of the edge is equivalent to the distance between the candidate point $c_{i+1,d_{i+1}}$ and the original data point x_{i+1}, namely $\Delta(x_{i+1},c_{i+1,d_{i+1}})$, as shown in Figure 12.

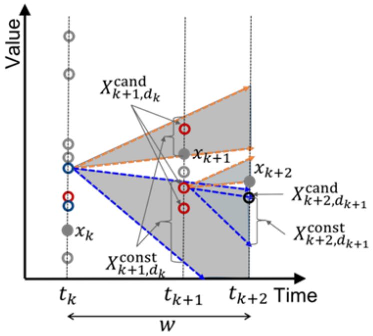


Figure 11. Obtain candidate sets $X^{\text{cand}}_{k+1,d_k}, X^{\text{cand}}_{k+2,d_{k+1}}$ according to ranges $X^{\text{const}}_{k+1,d_k}, X^{\text{const}}_{k+2,d_{k+1}}$

Furthermore, the above one or more candidate points continue to generate candidate repairing ranges for the subsequent x_j ($t_i+1<t_j\le t_k+w$), which intersect with X^{const}_{j,d_i} to form the new candidate range $X^{\text{const}}_{j,d_{i+1}}$. Then at the next moment t_i+2, candidate repairing points are selected within the range $X^{\text{const}}_{i+2,d_{i+1}}$ to generate new repairing edges. In this way, a map of repairing paths is finally shaped, as

shown in Figure 13. Solid lines indicate the path edges with weight, while dotted lines depict the completion of repairing. To obtain the minimum repairing path, this paper adopts the method of dynamic programming [3] to select the repairing path.

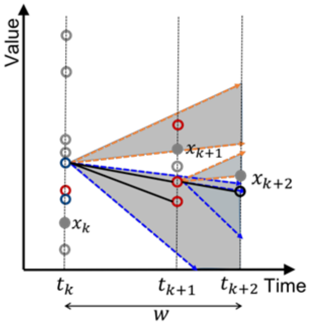


Figure 12. Generate edges for repairing path

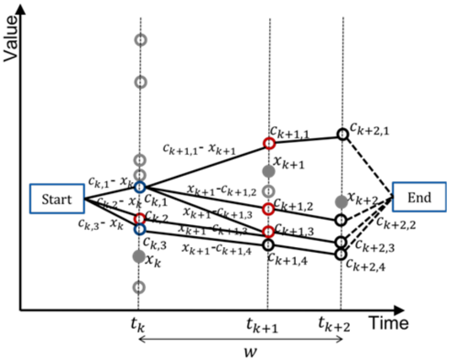


Figure 13. Repairing paths

3.4 Special case: single-speed constraint

This section will illustrate a special case, single-speed constraint, with only one constraint interval $[s_r^{\min}, s_r^{\max}]$, where r=1. It will be compared with SCREEN which is also based on speed constraints.

3.4.1 Speed constraint range

As there is only one speed constraint interval, as described in Section 3.1, the previous points x_i' (i=k− 1, k − 2,…, k − n_k) in the same window of x_k will constrain the speed ranges for x_k and all its subsequent points x_j (x_{k+1}, x_{k+2}, …, $x_{k+\omega}$) in the same window w, as shown in Figure 14. The speed constraint range is the same as that generated by point x_{k-1}' in the SCREEN method.

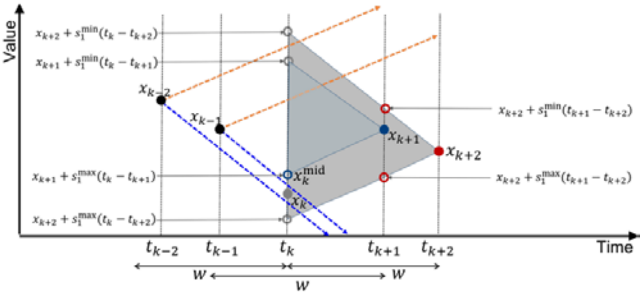

Value
$x_{k+2} + s_1^{\min}(t_k - t_{k+2})$
$x_{k+1} + s_1^{\min}(t_k - t_{k+1})$
x_{k-2}
x_{k-1}
$x_{k+1} + s_1^{\max}(t_k - t_{k+1})$
$x_{k+2} + s_1^{\max}(t_k - t_{k+2})$
x_k^{mid}
x_k
x_{k+1}
x_{k+2}
$x_{k+2} + s_1^{\min}(t_{k+1} - t_{k+2})$
$x_{k+2} + s_1^{\max}(t_{k+1} - t_{k+2})$
t_{k-2}
t_{k-1}
t_k
t_{k+1}
t_{k+2}
Time
w
w
w

Figure 14. Special case of single-speed constraint

3.4.2 Candidate repairing point

Same as the method of multi-speed constraints, in a single speed interval, the subsequent points x_i (x_{k+1}, x_{k+2}, …, $x_{k+\omega}$, $t_k<t_i\leq t_{k+\omega}\leq t_k+w$) of x_k in a given window w will generate candidate points of x_k, and the candidate point of x_{k+1} in this window can be generated by its subsequent points (x_{k+2}, …, $x_{k+\omega}$). Then, the candidate repairing points of the data points in window w are determined, and the candidate repairing point of $x_{k+\omega}$ is the point itself. From Figure 14, the above candidate repairing points, in a given window w, are consistent with those obtained by the SCREEN method.

3.4.3 Repairing path

As the same case in multi-speed constraints, in a single speed interval, each data point xi, in the window w, can obtain a series of candidate repairing points within a given speed constraint range. Each candidate repairing point will constrain the speed range of the subsequent points in the same window, and the range is the same as that constrained in the SCREEN method. Within this speed constraint range, a repairing path can be formed by selecting candidate points at each time in the window w. Since the constraint range is the same, the candidate repairing points are also the same as those in the original version of SCREEN. Therefore, among the repairing paths dependent on the above candidate repairing points, one must be the same as the repairing path in the original SCREEN version. In this paper, the shortest repairing path is chosen according to the dynamic programming method, which must be better than or equivalent to the repairing path in the SCREEN method.

3.5 Algorithm

Algorithm 1 is established according to the above repairing method.

Algorithm 1. Dynamic programming based on multi-speed constraints

Input: sequential time series x, time window w, starting data point x_k, and multi-speed constraints S;
Output: repaired time series x'.

```
for $i\leftarrow k$ to $n$ do        // Preliminarily determine candidate repairing points
  if $t_i>t_k+w$ then
    break;
  end if;
  for $j\leftarrow 1$ to $k$ do
    if $t_j<t_k-w$ then
      break;
    end if;
    From $x_j$, generate the speed constraint range of $x_i$, $X_i^{\text{const}}$, as shown in Formulas (1)–(4);
  end for;
  for $j\leftarrow i$ to $n$ do
    if $t_j>t_k+w$ then
      break;
    endif;
      Generate the candidate repairing point of $x_i$ from $x_j$, as shown in Formulas (5)–(6);
      Generate the set of candidate points $X_i$ based on $X_i^{\text{const}}$ , as shown in Formula (7);
  endfor;
endfor;
for $i\leftarrow k$ to $n$ do      // Determine repairing path
```

```
20:     if $t_i>t_k+w$ then
21:       break;
22:     endif;
23:     for each candidate $c_{i,d_i}$ of $x_i$ do
24:       for $j \leftarrow i+1$ to $n$ do
25:         if $t_j>t_k+w$ then
26:           break;
27:         endif;
28:           Generate the speed constraint range of $x_j$ from $c_{i,d_i}$, as shown in Formulas (8)–(9);
29:           Determine the speed constraint range of $x_j$, $X_{j,d_i}^{\text{const}}$, as shown in Formulas (10)–(12);
30:         if $j \leftarrow i+1$ then
31:           Determine the candidate repairing range of $x_j$, $X_{j,d_i}^{\text{cand}}$, as shown in Formula (13);
32:           Generate the repairing edge between the candidate $c_{i,d_i}$ of $x_i$ and $c_{j,d_j}$ in the candidate set
    $X_{j,d_i}^{\text{cand}}$ of $x_j$, and make $\Delta(x_j, c_{j,d_j})$ as the weight;
33:         endif;
34:       endfor;
35:     endfor;
36:   endfor;
37:   Use dynamic programming to determine the optimal repairing path;
38:   return $x'$;
```

As described above, in Algorithm 1, Lines 1–18 preliminarily determine the candidate repairing points and the speed constraint range. Among them, Lines 5–10 generate the corresponding range set X_i^{const} of multi-speed constraints for each data point x_i in the given time window w through the aforementioned x_j in the same window but outside the given window w according to Formulas (1)–(4); from Formulas (5)–(7), Lines 11–17 generate the set X_i of candidate repairing points for each x_i in a given time window w through the subsequent x_j in this window and the above constraint range set X_i^{const}.

Lines 19–36 determine the repairing path. To be specific, Lines 28–29 generate the speed constraint range of each subsequent time point in the window w though candidate time points according to Formulas (8)–(9), and then determine a new constraint range based on the above set of constraint ranges; Lines 30–33 screen out the candidate repairing points at the subsequent time according to the new constraint range. Then the corresponding repairing path edges are formed and given weights at the same time. At last, the optimal repairing path is solved by the dynamic programming method in Line 37.

The above definition reveals the maximum number of data points in the window is w, and Lines 1–18 require time of $O(w^2)$ for preliminarily determining candidate repairing points. The total number of candidate repairing points generated in the window is $(w-1)\times r+1$ at most, where r is the number of speed constraint intervals. As such, Lines 19–36 demand time of $O(w^2\times((w-1)\times r+1))$ for determining the repairing path. Considering the time for dynamic programming $O(((w-1)\times r+1)^2)$, the time complexity of the whole algorithm is $O(w^3\times r)$. As a locally defined algorithm, the multi-speed constraints proposed in this paper can quickly repair big data, providing a data basis for subsequent data analysis and artificial intelligence research.

4 Experiment

To verify the multi-speed constraints proposed in this paper, this section selects multiple datasets for experimental evaluation according to corresponding criteria and then compare the results with those of other existing methods. Specific experimental conditions, datasets, evaluation criteria, existing comparison methods and results are illustrated as follows.

4.1 Experimental setup

4.1.1 Conditions

In this paper, Java language is adopted to implement each part in the following conditions: 3.1 GHz Intel Core i5 processor and 16 GB 2133MHz LPDDR3 memory.

4.1.2 Data

An artificial dataset and two real datasets are used for experiments in this paper. The artificial dataset contains 30 000 data points, with main speed constraint intervals between [−10, −8] and [0,2]. Real dataset 1, with 34 220 data points, indicates fuel consumption of vehicles. As Example 1 in Section 1.1, the data mainly reflects fuel consumption and refueling: Considering the vibration of the fuel tank during the running of a vehicle, the speed range of fuel consumption is set as [−1,1]; refueling will make the oil level rise sharply, and the speed range of refueling can be set as [10, 70]. Real dataset 2, with a total of 7 962 data points, includes GPS track data, mainly collecting people’s walking track and vehicles’ running track. In light of the actual situation and collected data, the people walk in a speed range of [0, 10], while vehicles run in a speed range of [30, 100]. The above three datasets are all composed of error-free data points. In this paper, the method proposed in Ref.[4] is adopted to randomly generate new values as errors to replace the original true values into anomalies. As shown in the following experimental results, the anomaly rate of 0.1 indicates that 10% of data points are randomly replaced into anomalies. In the real dataset of fuel consumption, the input data value, which may be less than 0, represents abnormal data, because the lowest oil level in the fuel tank is 0 and cannot be negative. Similarly, the input data value may be greater than 70. According to the actual data value, the highest oil level in the oil tank is 70, so the data beyond this value can be regarded as anomalies. When the input data value is within the range of [0, 70], it can still be regarded as an anomaly, since the randomly input data value cannot form a data change speed, within a given range of threshold, with other points in the same time window in most cases. Also, in the GPS dataset and the artificial dataset, randomly input data values can be regarded as anomalies. Then the effect of multi-speed constraints on repairing real anomalies is verified on the basis of a dataset of altitudes, which is collected during the running of vehicles underground and on the ground light rail. The collected real data reveals many anomalies, and the altitude change at some data points may even be as high as 14 m within 1 s. Statistics prove 1 398 data points in this dataset, of which 218 are abnormal. After analyzing the overall data distribution and rationality, this paper selects [−2, 1.61] and [1.9, 2] as the speed constraint range of the data. In addition, to further validate the support provided by this method for subsequent data analysis and artificial intelligence research, this paper selects five datasets from UCR Time Series Classification Archive (http://www.cs.ucr.edu/~eamonn/ time_series_data/), including Car, Coffee, BeetleFly, Fish and InlineSkate, to verify the classification accuracy of repaired results.

4.2 Evaluation criteria

The RMS error [5] serves as the criteria for evaluating repaired results. x_{truth} is taken as the true value of time series, and x_{repair} as the repaired time series data. $\Delta(x_{\text{truth}},x_{\text{repair}})$ is taken as the distance between x_{truth} and x_{repair} to evaluate the similarity between the repaired result and the true value. Smaller $\Delta(x_{\text{truth}}, x_{\text{repair}})$ indicates the shorter distance between the repaired result and the true value, namely the more accurate repaired result.

Additionally, considering that the repaired data will play a basic supporting role in the subsequent data analysis and artificial intelligence research, this paper also presents the clustering and classification results of the repaired datasets. In this paper, the DBSCAN[6] method is adopted to cluster the repaired results, and the KNN method[7] to further classify them. Besides, *k*-fold cross-validation[8] is adopted. The accuracy of clustering and classification in this paper[9] is shown by the following formula:

$$\text{Accuracy} = \frac{\text{Data points classified correctly}}{\text{Total data points}}$$

4.3 Existing methods

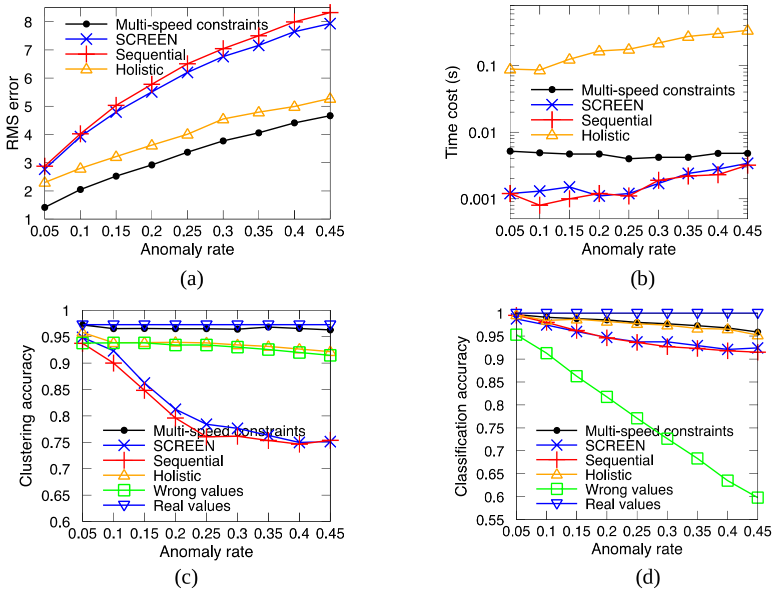
The multi-speed constraints proposed in this paper are compared with existing repairing methods, including the SCREEN method based on the single-speed constraint, the repairing method based on Sequential Dependency[10], and the Holistic[11] repairing method based on negative constraints.

4.4 Experimental results

Three datasets are selected to verify the repairing methods, and the RMS errors, time cost, and clustering and classification accuracy of repairing methods are provided for each dataset in case of different anomaly rates (anomalies/total data points) and data sizes. Moreover, the altitude dataset with real anomalies is adopted, and RMS errors, time cost, and clustering and classification accuracy are repaired by multiple methods. In addition, the classification accuracy of these repairing methods is verified by five datasets in UCR time series data. The specific results are as follows:

(1) Artificial dataset

From Figure 15(a), the repairing method based on multi-speed constraints proposed in this paper performs better than other methods at all rates of anomalies. The trend in results is similar to that of the Holistic method, but greatly better than it.



(a) (b)

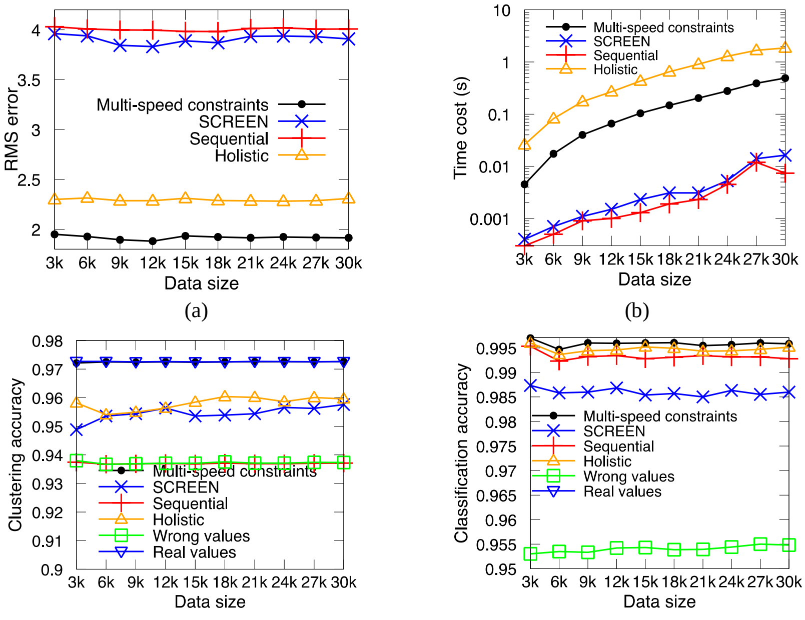
(c) (d)

Figure 15. Repairing in the artificial dataset at different anomaly rates

Figure 15(c) and 15(d) illustrate the clustering and classification accuracy to verify the impact of the proposed method on data analysis and artificial intelligence research. The figures reveal that the clustering effects of the SCREEN method and the Sequential method decrease significantly with the increase in the anomaly rate, even far lower than the clustering result of unrepaired wrong data; the proposed method, however, obtains the optimal clustering result, which is close to the real result and obviously higher than that of the Holistic method. With regard to classification results, similar to the case for RMS errors, the repairing method based on multi-speed constraints and the Holistic method are far superior to the SCREEN method and the Sequential method with the higher anomaly rate; meanwhile, the method based on multi-speed constraints shows the optimal repairing effect as a whole. The experimental results demonstrate that, compared with other methods, the repairing method based on multi-speed constraints proposed in this paper provides more accurate data basis for data analysis and artificial intelligence research.

Figure 15(b) reveals the proposed method requires far lower time cost than the Holistic method, and the time cost of the proposed method is relatively stable at different anomaly rates. Although the time cost of the SCREEN method and the Sequential method is low, the RMS errors of the proposed method are much smaller than those of the two methods at each anomaly rate. Further observation finds that with the increase in the anomaly rate, the repairing method based on multi-speed constraints proposed in this paper, compared with the SCREEN method and the Sequential method, has more outstanding advantages regarding RMS errors and accuracy of clustering and classification.

Form Figure 16, similar to the repairing results at different anomaly rates, those of the method based on multi-speed constraints are optimal for RMS errors in case of different data sizes. Besides, the repairing method based on multi-speed constraints and the Holistic method have much smaller RMS errors than the other two.



(a)

(b)

(c)

(d)

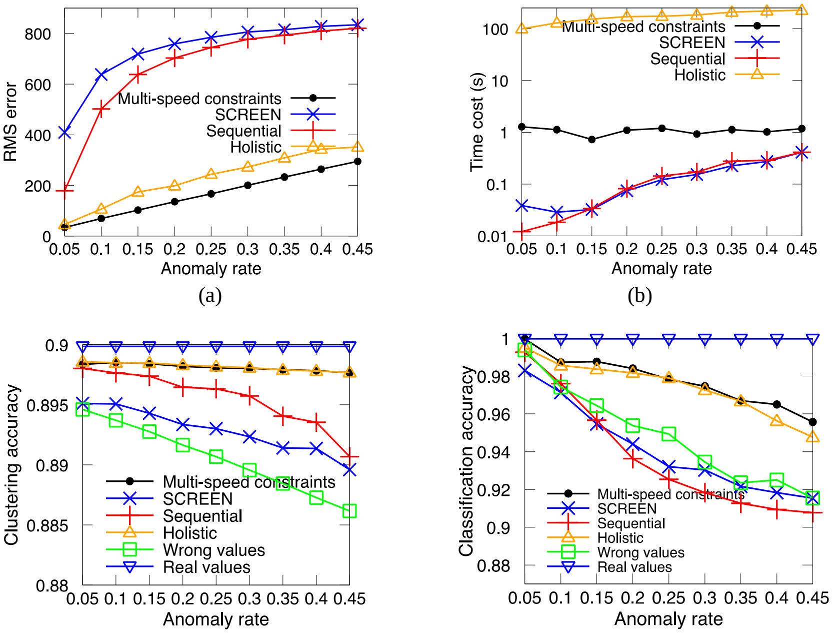
Figure 16. Repairing in the artificial dataset with different data sizes

In regard to clustering and classification accuracy, the data with an anomaly rate of 0.05 is selected in this experiment. In terms of clustering accuracy, the Sequential method performs worst in repairing results with different data sizes; the method based on multi-speed constraints, however, shows the highest clustering accuracy, with the results highly close to those of all correct values.

With regard to classification accuracy, the SCREEN method shows the lowest results, while the method based on multi-speed constraints also has the highest accuracy. At the same time, the results of this extended experiment also demonstrate this method can provide better data support in different data scales, even big data. Moreover, the method based on multi-speed constraints in this paper demands lower time cost than the Holistic method, with a good trade-off between repairing effect and time cost.

(2) Fuel consumption dataset

The experiment on the real fuel consumption dataset at different anomaly rates reveals the results indicated in Figure 17(a). Similar to the artificial dataset, the repairing method based on multi-speed constraints proposed in this paper performs better than other repairing methods at different anomaly rates. Especially at the anomaly rate greater than 0.1, it is far superior to the SCREEN method and Sequential method. According to RMS errors and time cost in Figure 17(b), the method based on multi-speed constraints proposed in this paper achieves a proper balance between repairing effect and time cost. At a time cost much lower than that of the Holistic method, the proposed method performs better in repairing.



(a)

(b)

(c)

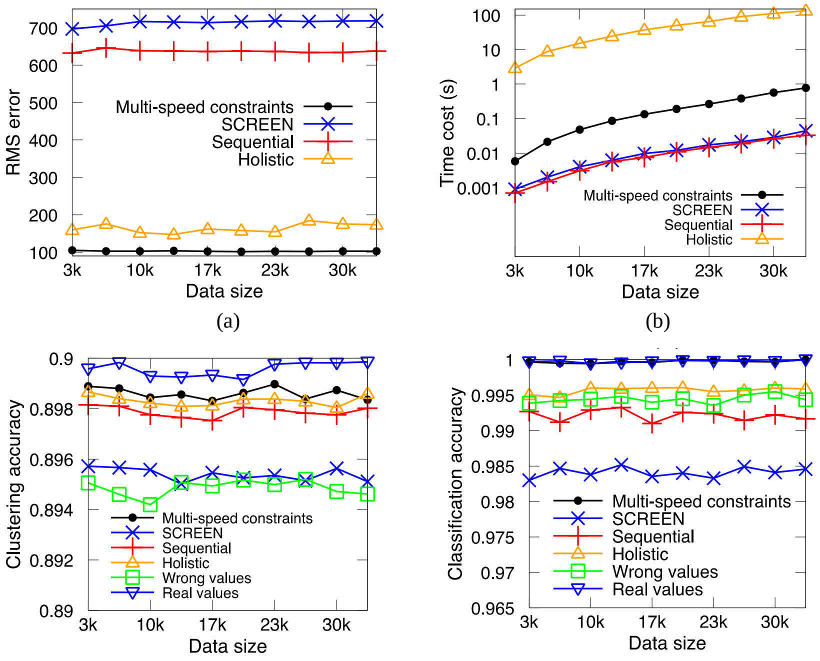
(d)

Figure 17. Repairing in the fuel consumption dataset at different anomaly rates

Furthermore, in terms of clustering accuracy, the repairing method based on multi-speed constraints proposed in this paper obtains the clustering results far higher than those of unrepaired wrong data. With the increase in the anomaly rate, its clustering accuracy is similar to that of the Holistic method, and much higher than that obtained by the SCREEN method and the Sequential method. It is worth mentioning that regarding classification accuracy, the repaired results of the SCREEN method and the Sequential method are even lower than those of unrepaired wrong data. However, the proposed method based on multi-speed constraints and the Holistic method achieve the results much higher than those of the unrepaired wrong data; especially when the anomaly rate is 0.05, they are quite close to the repairing result of real values.

Similar to the experimental results for the artificial dataset, the time cost of the proposed method is relatively stable at different anomaly rates; the time cost of the SCREEN method and the Sequential method presents a marked increase with the higher anomaly rate. On the whole, the repairing method based on multi-speed constraints proposed in this paper is more suitable for subsequent data analysis and artificial intelligence research. It can clean the time series efficiently in a short time to improve data quality, thus providing a more accurate data basis.

In addition to the experiments based on the anomaly rate, this paper provides the following experiments on real fuel consumption datasets with different data sizes. From Figure 18(a), consistent with the experimental results based on the anomaly rate, the repairing method based on multi-speed constraints shows the optimal repairing results with regard to different data sizes. Specifically, from the clustering accuracy in Figure 18(c), compared with other methods, the proposed method achieves the highest accuracy with different data sizes; the repairing results are stable, with good extendibility. From the classification accuracy indicated in Figure 18(d), at the anomaly rate of 0.05, the proposed method in this paper shows excellent performance on classification, and the repairing results are highly similar to the real values and close to 1 in case of different data sizes. Figure 18(b) shows that the RMS errors of this method are much lower than those in the SCREEN method and the Sequential method at a time cost slightly higher than those of the two. Moreover, with RMS errors lower than those of the Holistic method, the proposed method greatly reduces the time cost, with the overall difference by two orders of magnitude.



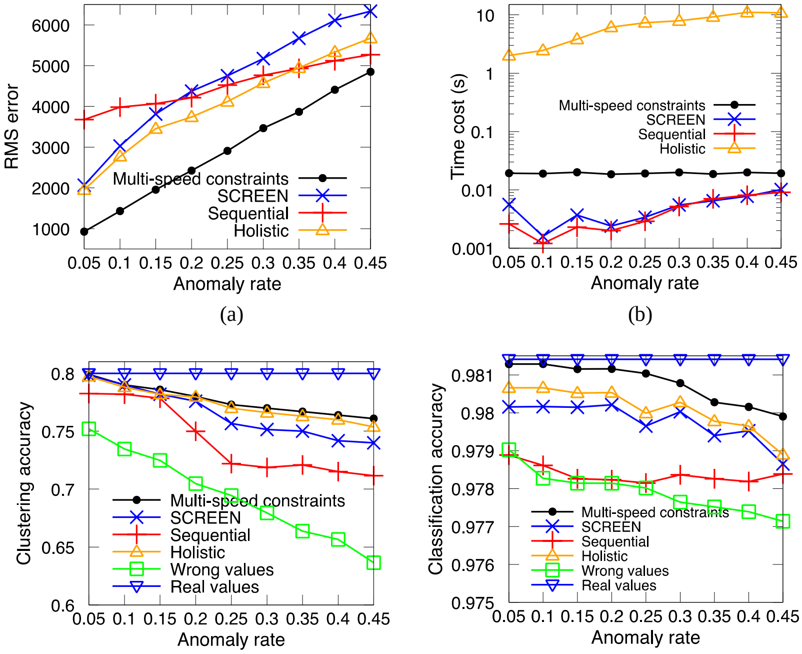
(a) (b)

(c) (d)

Figure 18. Repairing in the fuel consumption dataset with different data sizes

(3) GPS dataset

In Figure 19, the experimental results on the real GPS dataset at different anomaly rates are similar to those on the artificial dataset and the real fuel consumption dataset in terms of time cost. With regard to the experimental results of RMS errors, Figure 19(a) shows that other existing methods obtain similar repairing results for this dataset, and the method based on multi-speed constraints is distinctly superior to others. It is worth noting that, in the artificial dataset and the real fuel consumption dataset, the data series are spaced at an equal time interval, and the Sequential method considers the numerical distance constraint between two consecutive data points. As such, the Sequential method and the SCREEN speed constraint method have similar results and trends. In this experiment on the GPS dataset, however, the data series have unequal time intervals, so the experimental results of the above two methods are markedly different.



(a) (b)

(c) (d)

Figure 19. Repairing in the GPS dataset at different anomaly rates

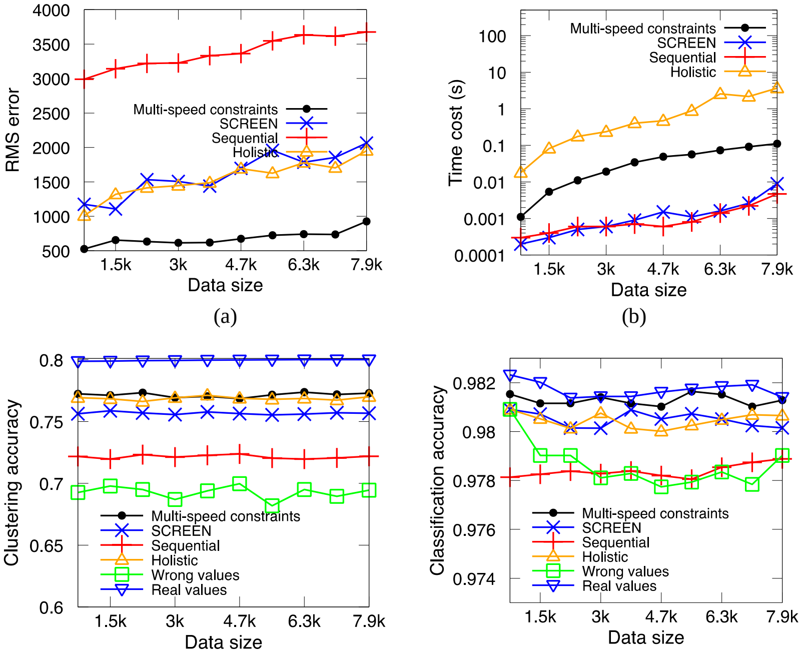
From Figure 19(c), with regard to clustering accuracy, the clustering results of the proposed method are much better than those of the SCREEN method, the Sequential method and unrepaired wrong data with the higher anomaly rate, which are also superior to those of the Holistic method. Figure 19(d) also clearly displays that the proposed method based on multi-speed constraints presents the optimal repairing effect regarding classification accuracy. When the anomaly rate is lower than 0.25, the classification accuracy is close to that of real time series data and far higher than that of other repairing methods, especially the Sequential method.

To better reflect the performance of each method, this extended experiment selects the GPS dataset at the anomaly rate of 0.05 and provides results regarding different data sizes. From Figure 20(a), with the larger data sizes, the RMS error of each method has increased more or less, and the proposed method based on multi-speed constraints has the smallest increment, proving its optimal extendibility.

In terms of classification accuracy, the repairing method based on multi-speed constraints proposed in this paper also achieves the highest result. At the same time, the Sequential method has the lowest accuracy, and even with some data sizes (such as less than 2.3k), its classification accuracy is far lower than that of the unrepaired wrong data, consistent with the repairing results at different anomaly rates and results of RMS errors with different data sizes. As the clustering effect on the repairing results of multiple methods is relatively close at the anomaly rate of 0.05, this paper selects the data at the anomaly rate of 0.25 for the extended experiment on the clustering results, so that the clustering accuracy of each method is clearly presented. From the figures, the clustering accuracy of the proposed method, at a lower time cost, is slightly higher than that of the

Holistic method, while far higher than that of the Sequential method and the unrepaired wrong data.

With regard to the time cost, Figure 20(b) illustrates that similar to the experiment on the real fuel consumption dataset, the time cost of the proposed method, with the smallest RMS errors, is within an acceptable range, which is one order of magnitude lower than that of the Holistic method as a whole.



(a)

(b)

(c)

(d)

Figure 20. Repairing in the GPS dataset with different data sizes

(4) Altitude dataset

To further prove the practicability of the proposed method based on multi-speed constraints in real datasets, this paper adopts a variety of repairing methods for the altitude dataset with real anomalies, with the final results of RMS errors, clustering and classification accuracy, and time cost in Table 1. As indicated in the table, in terms of RMS errors, the repairing method based on multi-speed constraints proposed in this paper obtains the most accurate repairing results, with smaller RMS errors than those of the Holistic method, the SCREEN method and the SWAB method, which is also greatly superior to the Sequential method. Moreover, the proposed method behaves better than others regarding clustering accuracy; its classification accuracy is closer to that based on correct values; it requires the lowest time cost, far lower than that of the Holistic method with relatively good repairing results.

Table 1 Repairing in the altitude dataset with real anomalies

Repairing method	RMS error	Time cost (ms)	Clustering accuracy	Classification accuracy
Multi-speed constraints	1.07	2.3	0.70	0.76
SCREEN	1.34	3.3	0.69	0.75
Sequential	2.33	4.0	0.63	0.74
Holistic	1.27	55.2	0.65	0.74
Wrong values	□	□	0.59	0.71
Real values	□	□	0.75	0.80

(5) UCR dataset

Further verifying the performance of each repairing method in classification can provide a data support for applications including data analysis and artificial intelligence. Then this paper selects multiple time series datasets with classification labels to verify the effect of the proposed method based on multi-speed constraints on subsequent data applications in a more comprehensive manner. Figure 21 shows that in different datasets, the proposed method has a desired classification result, which is much better than that based on unrepaired wrong data and close to the classification accuracy of real values. There are some differences in the number of classes for real data among the five datasets in the figure, among which the Car dataset has four classification labels, while Coffee and BeetleFly datasets have two. As such, the overall classification accuracy is high in these two datasets. On the contrary, there are seven classification labels in Fish and InlineSkate datasets, with the overall low accuracy of classification. However, in these two datasets, the classification results repaired by each method are much higher than those without repairing, demonstrating the important role of data cleaning and repairing in the early stage of data applications in the subsequent data analysis and artificial intelligence processes.

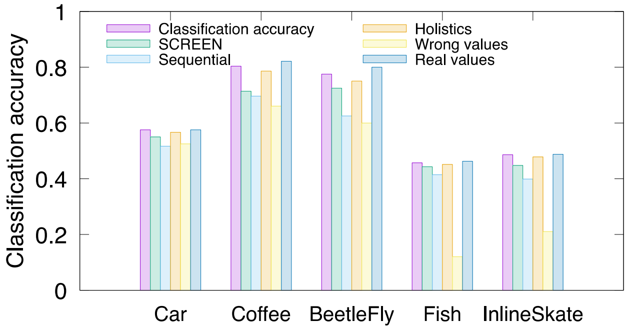


Figure 21. Classification accuracy in UCR datasets

5 Related work

Amid the impressive progress in big data and artificial intelligence technology, data management and analysis technology, as the technical support and foundation, has increasingly become a hot issue in related fields. Data quality has attracted increasing attention since it is pivotal to promoting data management and analysis technology, optimize the output of big data and artificial intelligence, and reduce the restrictions on subsequent analysis and research caused by data problems. Ji *et al*.[12] put forward the query and fault-tolerance mechanism, but this mechanism must compromise on fault-tolerance, performance and implementation cost, without dealing with the data usage outside the scope of data query. In addition, many researchers have introduced a variety of detection and repairing techniques for data anomalies.

5.1 Repairing methods based on smoothing

Smoothing-based repairing methods depend on data smoothing technology to reduce anomalies. They can reduce noise points and make the data change smoother. Intuitively, smoothed data has fewer anomalies. The SWAB smoothing algorithm[13] cleans time series data based on linear interpolation[14] and regression[15]. As this algorithm can segment time series, it can support online cleaning of time series data. Additionally, the moving average method is often used to smooth and repair time series data. Simple Moving Average (SMA) is the unweighted average of the last k data, which is applied by the algorithm to repair the next data point. The Weighted Moving Average (WMA) adds weights to data points at different positions in the window; for example, the point farther away from the target data points has the lower data weight. Accordingly, Exponentially Weighted Moving Average (EWMA)[16] adds exponentially decreasing weights with the increase in time and distance, which is suitable for unsteady time series[17-19].

The repairing method based on smoothing has serious limitations. To ensure the smoothness of time data series, smoothing-based repairing method will over-repair many original correct true data near anomalies into wrong errors, and anomalies will greatly compromise the accuracy of repairing results.

5.2 Repairing methods based on constraints

Most data has a relationship of constraint and dependence between points. Many cleaning algorithms based on integrity constraints are available in the field of relational databases. Some rules of data constraints, such as functional dependency (FD) [20,21], can be applied to data cleaning and repairing. Such methods clean the data by solving the minimum repairing, with the corresponding results conforming to the given FD constraint rules. However, because the constraint relationship is applicable to any pair of tuples, repairing with minimal modifications is usually considered as a NP-hard problem[22]. Considering this problem, Beskales *et al*. [23] proposed a data cleaning method based on sampling, the basic idea of which was to extract some samples from candidate repairing datasets for data cleaning. In addition, some real datasets cannot be subject to the absolute FD constraints. Then, Bohannon *et al.* [24] introduced the concept of conditions in cleaning and repairing based on FD, namely using conditional functional dependency (CFD)[25,26] as the constraint to clean data. However, the results of the algorithm are not ideal for time cost, because the constraint rules required by this method are massive and complicated. Against the background of big data and artificial intelligence, this method fails to provide high-quality data support for subsequent technical operations in a quick and accurate manner.

Generally, in time series data, most of the data values are concrete, and FD, CFD and other data constraint rules need to follow strict equality relations. As a result, the repairing method based on the above constraints is hard to produce good cleaning and repairing results in time series data. Then Fan *et al.* [27] put forward matching dependency (MD), which further relaxed the strict equality relations into similarity relations, namely that similarity measure was introduced on the left side of constraint rules. Song et al. [28] proposed differential dependency (DD), which introduced similarity measure to both left and right sides of constraint rules, thus relaxing the equality relations between both sides into similarity relations. Furthermore, Lopatenko *et al.* [29] proposed a rule based on denial constraint (DC) and then studied the data cleaning and repairing method based on DC. Chu *et al.* [11] also proposed a DC-based Holistic algorithm.

However, as a technology supporting speed constraints, the Holistic method can only repair general table data, failing to serve for online cleaning of stream data. Nowadays, data analysis and artificial intelligence technology needs to process massive data, and the Holistic method cannot provide corresponding technical support. In this paper, a local repairing method based on multi-speed constraints in a given window is proposed to support online cleaning. This local method is more conducive to data cleaning in big data environment, which is more convenient for subsequent data management and analysis and artificial intelligence research. The experiments demonstrate compared with the overall cleaning, the proposed method can reduce the time cost by two orders of magnitude at most. Moreover, the sequential dependency method cannot accurately express speed constraints. The Sequential method mainly focuses on the difference between two consecutive data points in a series, but the given dependence is not accurate when the time interval between data points is different. The speed constraint-based SCREEN method only considers the constraint in a single range. When the speed constraint involves multiple ranges, the repairing method will fail to achieve desired results due to insufficient or excessive detection and repairing.

The multi-speed constraints proposed in this paper consider more specific constraint intervals for more accurate repairing results. As Example 1 in Section 1.1, the change in the oil level of vehicles is due to two behaviors: fuel consumption and refueling. Considering the vibrations of vehicles and the fuel tank, the change speed of the oil level will be within the ranges of [−1,1] and [10,70]. The single-speed constraint cannot express such an accurate constraint condition, resulting in large errors in repairing results; multi-speed constraints will accurately constrain the data, so it can be applied reasonably and widely. As illustrated in Figure 22, this paper uses various constraint methods to repair 100 data points in the fuel consumption dataset. It can be observed that the Sequential method is similar to the SCREEN method. However, the Sequential method only constrains the data distance between two consecutive points, while the SCREEN method also considers the speed between the two points (namely the relationship between data values and their time stamps), so the latter is more accurate. Nevertheless, since the SCREEN method only sets a single-speed constraint, some normal points and anomalies cannot be correctly detected and distinguished. When the interval is set to [−1,70], due to an anomaly at 15:12, subsequent correct values are corrected by mistake; when it is set to [−1,1], due to refueling at 15:09, most of the subsequent normal values will be misjudged as anomalies for over-repair; similarly, if it is set to [10,70], almost all the points will be over-repaired, resulting in more serious damage to the data. To sum up, the proposed repairing method based on multi-speed constraints can satisfy the multi-speed-threshold constraints, with accurate repairing results. Furthermore, the experimental results of the above datasets reveal that the proposed method produces better repairing results with much lower time cost than the Holistic method.

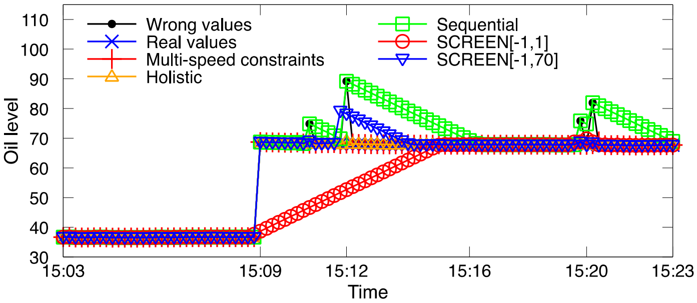


Figure 22. Repairing by constraint-based methods

6 Conclusion

Considering the strong correlation between time stamps and corresponding data values in time series, this paper introduces multi-speed constraints based on the speed of data change. In this method, the speed constraint interval of each data point in a given window of time series can be obtained by multi-speed constraints, and then the time series data can be constrained to detect the anomalies. Meanwhile, the multi-speed constraints can generate candidate repairing points for each data point in the window through its subsequent points. Then this paper relies on dynamic programming to select the optimal repairing path from the above candidate repairing points, with the repairing result following the principle of minimum repairing. To verify the above-mentioned repairing methods, this paper tests this method and other existing methods through an artificial dataset, two real datasets (fuel consumption dataset and GPS dataset) and a dataset with real anomalies (altitude dataset). The experimental results demonstrate that, compared with other existing repairing methods, the proposed dynamic programming method based on multi-speed constraints follows the principle of minimum repairing. Besides, it can cope with complicated data, thus having the smallest RMS errors in case of different anomaly rates and data sizes, with the optimal repairing effect. At the same time, since data quality is crucial to the subsequent data analysis and artificial intelligence technology, this paper applies multiple datasets to verifying each method with regard to clustering and classification accuracy. In the experimental results, the repairing method based on multi-speed constraints in this paper still shows the optimal repairing results, with the highest classification accuracy among all methods. It also obtains ideal results in running performance and time cost, which are markedly superior to those of the constraint-based Holistic method.

References

[1] Song SX, Zhang AQ, Wang JM, Yu PS. SCREEN: Stream data cleaning under speed constraints. Proc. of the 2015 ACM SIGMOD Int'l Conf. on Management of Data. 2015. 827−841.

[2] Bohannon P, Flaster M, Fan W, Rastogi R. A cost-based model and effective heuristic for repairing constraints by value modification. Proc. of the ACM SIGMOD Int'l Conf. on Management of Data. 2005. 143−154.

[3] Xu GG, Zhao HS, Huang ZY. Optimization Method with MATLAB Implementation. Beijing: Beijing University of Aeronautics and Astronautics Press, 2018. 91−102 (in Chinese).

[4] Brillinger DR. Time series—Data analysis and theory, volume 36 of Classics in applied mathematics. Proc. of the SIAM. 2001.

[5] Jeffery SR, Garofalakis MN, Franklin MJ. Adaptive cleaning for RFID data streams. Proc. of the 32nd Int'l Conf. on Very Large Data Bases. 2006. 163−174.

[6] Gan JH, Tao YF. DBSCAN revisited: Mis-claim, un-fixability, and approximation. Proc. of the 2015 ACM SIGMOD Int'l Conf. on Management of Data. 2015. 519−530.

[7] Zhang SC, Li XL, Zong M, Zhu XF, Cheng DB. Learning *k* for *k*NN classification. ACM TIST, 2017, 8(3): 43:1−43:19.

[8] Wong TT, Yang NY. Dependency analysis of accuracy estimates in *k*-fold cross validation. IEEE Trans. on Knowledge and Data Engineering, 2017, 29(11): 2417−2427.

[9] Accuracy. In Encyclopedia of Machine Learning and Data Mining. 2017.

[10] Golab L, Karloff HJ, Korn F, Saha A, Srivastava D. Sequential dependencies. Proc. of the VLDB Endowment, 2009, 2(1): 574−585.

[11] Chu X, Ilyas IF, Papotti P. Holistic data cleaning: Putting violations into context. Proc. of the 29th IEEE Int'l Conf. on Data Engineering (ICDE 2013). 2013. 458−469.

[12] Ji YH, Chai YP, Zhou X, Ren LP, Qin YJ, *et al*. Smart intra-query fault tolerance for massive parallel processing databases. Data Ence and Engineering, 2020, 5(1): 65−79.

[13] Keogh EJ, Chu S, Hart DM, *et al*. An online algorithm for segmenting time series. Proc. of the 2001 IEEE Int'l Conf. on Data Mining. 2001. 289−296.

[14] Wiener N. Extrapolation, Interpolation, and Smoothing of Stationary Time Series: Volume 7. Cambridge: MIT Press, 1949.

[15] Shatkay H, Zdonik SB. Approximate queries and representations for large data sequences. Proc. of the 12th Int'l Conf. on Data Engineering. 1996. 536−545. https://doi.org/10.1109/ICDE.1996.492204

[16] Gardner Jr ES. Exponential smoothing: The state of the art—Part II. Int'l Journal of Forecasting, 2006, 22(4): 637−666.

[17] Holt CC. Forecasting seasonals and trends by exponentially weighted moving averages. Int'l Journal of Forecasting, 2004, 20(1): 5−10.

[18] Winters PR. Forecasting sales by exponentially weighted moving averages. Management Science, 1960, 6(3): 324–342.

[19] Brown RG. Smoothing, forecasting and prediction of discrete time series. Journal of the American Statistical Association, 1964, 59(307): 973.

[20] Huhtala Y, Kärkkäinen J, Porkka P, *et al*. Efficient discovery of functional and approximate dependencies using partitions. Proc. of the 14th Int'l Conf. on Data Engineering. 1998. 392–401. https: //doi.org/10.1109/ICDE.1998.655802

[21] Jin CQ, Liu HP, Zhou AY. Functional dependency and conditional constraint based data repair. Ruan Jian Xue Bao/Journal of Software, 2016, 27(7): 1671–1684 (in Chinese with English abstract). http://www.jos.org.cn/1000-9825/5037.htm [doi: 10.13328/j. cnki.jos.005037]

[22] Kolahi S, Lakshmanan LVS. On approximating optimum repairs for functional dependency violations. In: Proc. of the 12th Int'l Conf. on Database Theory (ICDT 2009). 2009. 53–62.

[23] Beskales G, Ilyas IF, Golab L. Sampling the repairs of functional dependency violations under hard constraints. Proc. of the VLDB Endowment, 2010, 3(1): 197–207. http://www.comp.nus.edu.sg/~vldb2010/proceedings/files/papers/R17.pdf

[24] Bohannon P, Fan W, Geerts F, *et al*. Conditional functional dependencies for data cleaning. Proc. of the 23rd Int'l Conf. on Data Engineering (ICDE 2007). 2007. 746–755. https://doi.org/10.1109/ICDE. 2007. 367920

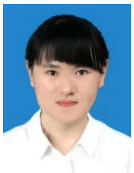
[25] Fan WF, Geerts F, Jia X, *et al*. Conditional functional dependencies for capturing data inconsistencies. ACM Trans. on Database Systems, 2008, 33(2): 6:1–6:48. http://doi.acm.org/10.1145/1366102.1366103

[26] Fan WF, Geerts F, Lakshmanan LVS, *et al*. Discovering conditional functional dependencies. Proc. of the 25th Int'l Conf. on Data Engineering (ICDE 2009). 2009. 1231–1234. [doi: 10.1109/ICDE.2009. 208]

[27] Fan WF, Jia X, Li J, *et al*. Reasoning about record matching rules. Proc. of the VLDB Endowment, 2009, 2(1): 407–418. http://www.vldb.org/pvldb/2/vldb09-654.pdf

[28] Song SX, Chen L. Differential dependencies: Reasoning and discovery. ACM Trans. on Database Systems, 2011, 36(3): 16:1–16:41. http://doi.acm.org/10.1145/2000824.2000826

[29] Lopatenko A, Bravo L. Efficient approximation algorithms for repairing inconsistent databases. Proc. of the 23rd Int'l Conf. on Data Engineering (ICDE 2007). 2007. 216–225.



GAO Fei (1993–), PhD, is mainly engaged in the research on data cleaning.

SONG Shao-Xu (1981–), PhD, associate professor, PhD supervisor, CCF professional member, is mainly engaged in the research on database, data quality, time series data cleaning, and big data integration.

WANG Jian-Min (1968–), PhD, professor, PhD supervisor, CCF senior member, is mainly engaged in the research on database, workflow, big data and knowledge engineering (unstructured data management, business process and product lifecycle management, digital copyright and system security technology, and database testing technology).