Postprint: Fault Sequence Pattern Mining Algorithm Based on Text Analysis

Authors: Chang Wenbing, Yuan Xinglong, Shenghan Zhou, Li Lei

Date: 2018-05-24T00:00:00+00:00

Abstract

Aiming at text data with low structural consistency and diverse expression forms, a fault sequence pattern mining algorithm based on text information is proposed to discover temporal relationships among faults. To mine fault patterns from text-recorded fault information, the text information is first vectorized, similarity measurement is performed on fault text information, and faults expressing identical meanings are grouped into the same category. Building upon this foundation and according to fault characteristics, the concepts of maximum window threshold and minimum co-occurrence degree threshold are introduced to construct a fault sequence pattern mining algorithm framework. Finally, by applying sequence pattern mining to text fault information of a certain aircraft type, correct fault sequence relationships are identified. Case studies verify that the proposed algorithm is both correct and effective.

Full Text

Preamble

Failures Sequence Pattern Mining Algorithm Based on Text Analysis

Chang Wenbing, Yuan Xinglong, Zhou Shenghan†, Li Lei

School of Reliability & System Engineering, Beihang University, Beijing 100191, China

Abstract: For textual data with poor structural consistency and diverse expression forms, this paper proposes a failure sequence pattern mining algorithm based on text information to discover temporal relationships between failures. To mine failure patterns from text records, the text information is first vectorized to measure similarity among failure descriptions, grouping failures with identical meanings into the same category. Based on failure characteristics,

the concepts of maximum window threshold and minimum concurrence threshold are introduced to construct a failure sequence pattern mining framework. Finally, sequence pattern mining is performed on textual failure data from a certain aircraft type, successfully identifying correct failure sequence relationships. The case study validates the correctness and effectiveness of the proposed algorithm.

Keywords: sequence pattern; data mining; text similarity; aircraft failure; text mining

---

0 Introduction

Aircraft, as large-scale complex equipment systems with long service cycles and harsh operating environments, experience frequent failures with complex root causes. Extensive textual failure information accumulated during long-term maintenance support processes holds significant value for failure analysis and maintenance decision-making, yet remains to be explored at deeper levels. Currently, no research exists on applying sequence pattern mining to text information for identifying temporal relationships between failures. This paper first addresses text similarity measurement to group semantically equivalent failure descriptions, then builds a sequence pattern mining framework tailored for textual failure data.

1 Research Methods

1.1 Related Concepts and Definitions

The following concepts and symbols are defined for precise description of the methodology:

a) Event. An event is denoted as e_i, satisfying $e_i \in p$ where p is the event set. Each event has a timestamp indicating its temporal order.

b) Event Sequence. A sequence is denoted as $e_i \rightarrow e_j$, where e_i and e_j represent events i and j. Events in a sequence have temporal precedence, with e_i occurring before e_j.

c) Event Window. An event window is denoted as win_{ij}, representing the number of events between sequence $e_i \rightarrow e_j$.

d) Event Similarity. Event similarity is denoted as X_{ij}, representing the similarity degree between events i and j.

e) Similar Event Set. A similar event set is denoted as $SES_k = [e_1^k, e_2^k, \dots, e_n^k]$, where e_i^k represents the i-th event in similar event set SES_k. Any two events in the set satisfy the minimum similarity threshold min_sim, and all events in SES_k are considered the same type of event.

f) Similar Event Set Frequency. Event frequency is denoted as $freq(k)$, representing the number of events in similar event set SES_k.

g) Frequent Event Set. When the number of events in similar event set SES_k, $freq(k)$, is greater than or equal to min_freq, SES_k is identified as a frequent event set, denoted as $FES_k = [e_1^k, e_2^k, ..., e_n^k]$, where min_freq represents the minimum frequency threshold.

h) Sequence Support. If there exists an event sequence $e_i^p \rightarrow e_j^q$ where e_i^p and e_j^q represent events i and j belonging to frequent event sets FES_p and FES_q respectively, and win_{ij} is less than or equal to the maximum event window threshold max_win, then the support of sequence pattern $p \rightarrow q$ is incremented by one. The sequence pattern support is denoted as $sup(p \rightarrow q)$.

i) Sequence Concurrence. The concurrence of a sequence $p \rightarrow q$ refers to the number of distinct products in which the sequence appears, denoted as $occ(p \rightarrow q)$.

j) Sequential Pattern. A sequence $p \rightarrow q$ is a sequential pattern if and only if it satisfies three constraints: (a) both p and q are FESs meeting min_freq; (b) $sup(p \rightarrow q) \geq min\_sup$; and (c) $occ(p \rightarrow q) \geq min\_occ$, where min_sup is the minimum support threshold and min_occ is the minimum concurrence threshold.

The sequence patterns mined in this study are repetitive gapped sequential patterns, where gaps mean that two events need not be consecutive. This is more realistic since causal relationships may exist within certain intervals rather than only between consecutive failures. For example, in [Figure 1: see original paper], assuming a maximum gap of 2, for sequence A → B, using element positions to represent sequence relationships, in S1 we have 1→3, 2→3, 6→7, and 6→8 satisfying the conditions (recorded 4 times), and in S2 we have 1→2 (recorded 1 time), totaling 5 occurrences across both sequences.

1.2 Text Similarity Measurement Model

Due to natural language characteristics, different individuals may describe the same event differently, making identical textual descriptions rare and failure sequence patterns difficult to identify. Measuring similarity among failure text descriptions facilitates more effective sequence pattern mining.

1.2.1 Text Preprocessing Natural language processing using language models is word-based. Since Chinese lacks explicit word delimiters, fault texts must first be segmented before further processing. The segmentation results contain prepositions, conjunctions, punctuation marks, and other low-discriminative elements. To better measure text similarity, stop-word removal is performed on the segmented results.

1.2.2 Text Vectorization Distributed word representation refers to a dense low-dimensional real-valued vector, such as [0.792, −0.177, −0.107, 0.109, −0.542, ···]. The Doc2Vec model maps each segmented sentence to an independent vector, capturing semantic relationships between words while considering word order, thus effectively vectorizing text. The Doc2Vec training model used in this paper is:

```
model = Doc2Vec(sentences, size, window, min_count, workers, min_alpha)
```

where *sentences* is the sentence corpus, *size* is the feature vector dimension, *window* is the maximum distance between the target word and context words used for prediction, *alpha* is the initial learning rate, *min_count* ignores words with total frequency below this value, and *workers* specifies the number of CPU threads for training.

1.2.3 Similarity Calculation For vectorized texts, cosine similarity is used to calculate inter-text similarity by measuring the cosine of the angle between two vectors:

$$similarity = \frac{\sum(A_i \times B_i)}{\sqrt{\sum(A_i)^2} \times \sqrt{\sum(B_i)^2}}$$

where A_i and B_i represent vector components of A and B respectively. Due to the nature of similarity measurement, the similarity matrix is symmetric with diagonal elements equal to 1.

The algorithm design flow is shown in [Figure 2: see original paper]. The text similarity matrix is presented in , where X_{ij} represents the similarity between events i and j (obviously $X_{ij} = 1$ when i=j). Given a minimum similarity threshold min_sim, similar event sets are identified using:

$$X_{ij} \geq min\_sim, X_{ij} = 1 X_{ij} < min\_sim, X_{ij} = 0$$

This transformation yields the similar event set matrix shown in , where matrix values of 1 indicate similar events belonging to the same event set, and 0 indicates dissimilar events, thus identifying similar event sets SES_k.

The frequency of similar event sets is calculated as:

$$freq_p = \sum X_{ij}, \quad i, p = 1, 2, \ldots, n$$

where X_{ij} represents event similarity (1 or 0), and $freq_p$ denotes the number of events in similar event set SES_p. Given a minimum frequency threshold min_freq, event sets with $freq_p \geq min\_freq$ are retained as frequent event sets FES_k, while others are removed.

1.3 Algorithm Flow

The algorithm design flow comprises two main components: establishing a fault text similarity measurement model and designing a failure sequence mining algorithm. Fault text information must first be processed before similarity measurement, followed by failure sequence pattern mining, and finally validated through case studies. The design flow is illustrated in [Figure 2: see original paper].

2 Algorithm Construction

The algorithm consists of two parts: (1) mining frequent event sets based on text similarity measurement, and (2) mining failure sequence patterns from frequent event sets.

2.1 Frequent Event Set Mining

2.1.1 Algorithm Description The frequent event set mining algorithm first obtains a similarity matrix through the fault text similarity measurement model. It then identifies similar event sets and calculates their frequencies. Given min_freq, event sets meeting the threshold are retained as frequent event sets FES_k.

2.1.2 Algorithm Flow The frequent event set mining process is shown in [Figure 3: see original paper]. The algorithm proceeds as follows: (a) obtain the similarity matrix from the fault text similarity measurement model; (b) identify similar event sets; (c) calculate frequencies and compare against min_freq to extract frequent event sets.

2.2 Sequence Pattern Mining

2.2.1 Algorithm Description Sequence pattern mining involves four steps: (a) partition all events in frequent event sets FES_k by aircraft ID, as shown in ; (b) mine fault sequence patterns for individual aircraft by calculating support and concurrence for sequences between frequent event sets p and q, given maximum window threshold max_win; (c) iterate across all aircraft, accumulating support and concurrence for each sequence pattern; (d) verify whether each sequence pattern $p \rightarrow q$ satisfies both min_sup and min_occ thresholds according to:

$$sup(p \rightarrow q) \geq min\_supocc(p \rightarrow q) \geq min\_occ$$

If satisfied, sequence pattern $p \rightarrow q$ is considered valid.

2.2.2 Algorithm Flow The sequence pattern mining algorithm flow is illustrated in [Figure 4: see original paper].

3 Instance Verification

Textual failure data from three aircraft (20 failure records total) were used for validation. Aircraft 1 and 2 each had 7 failure text descriptions, while Aircraft 3 had 6. The objective was to identify failure sequence patterns. Product IDs and failure sequence numbers are listed in , where e_{ij}^{p} represents the j-th failure event on aircraft i belonging to frequent event set p.

3.1 Text Preprocessing

Fault text examples are shown in . Using Python' s jieba segmentation package, the texts were segmented and stop words removed, yielding the preprocessed results in .

3.2 Fault Text Similarity Calculation

The Doc2Vec model was applied for text representation with parameters: `model=Doc2Vec(sentences, size=10, window=3, min_count=3, workers=4, min_alpha=0.002)`. Cosine similarity was used for measurement. With $min\_sim = 0.8$, the similarity matrix was converted to a 20×20 binary matrix for frequency calculation. The frequency of each fault text description was: [3, 3, 1, 2, 3, 1, 4, 3, 1, 4, 3, 2, 3, 1, 4, 3, 6, 1, 3, 1].

3.3 Frequent Event Set Mining

Setting $min\_freq = 3$, the pseudo-code program identified frequent event set text indices as [1, 2, 5, 7, 8, 10, 11, 13, 15, 16, 17, 19]. The similar frequent event sets are shown in .

3.4 Sequence Pattern Mining

With parameters $max\_win = 4$, $min\_sup = 4$, and $min\_occ = 2$, the pseudo-code program mined failure sequence patterns, yielding the results in . The valid sequence pattern was { "No. 4 engine oil radiator honeycomb structure oil leakage" → "No. 2 engine oil radiator honeycomb hole oil seepage" }, occurring 4 times across 2 aircraft. Validation against the textual data confirms this failure sequence relationship objectively exists. The results suggest that during maintenance support, if an engine oil radiator exhibits oil leakage or seepage, all engine oil radiators should be inspected for preventive maintenance.

4 Conclusion

Textual failure information exhibits diverse expression forms and poor structural consistency, making direct application of existing sequence pattern mining algorithms infeasible. This paper proposes a failure sequence pattern mining algorithm based on text similarity measurement with two key distinctions: (a) it effectively groups semantically equivalent failure texts into categories, integrating poorly structured textual data before sequence pattern mining; and (b)

it introduces the concepts of minimum frequency for text data, and maximum event window and minimum concurrence for sequence pattern mining, ensuring discovered patterns are universally applicable.

The case study demonstrates the algorithm' s correctness and effectiveness, providing support for identifying failure temporal relationships, failure prediction, and maintenance decision-making.

References

[1] Wu Yixiao. Research on Chinese word segmentation algorithm [J]. Technology and Economic Guide, 2018 (2): 122-123.

[2] Zheng Wenchao, Xu Peng. Research on Chinese word clustering with word2vec [J]. Software, 2013, 34 (12): 160-162.

[3] Zhang Zhichang, Zhou Huixia, Yao Rendong, et al. Recognition of Chinese lexical entailment relation based on word vector [J]. Computer Engineering, 2016, 42 (2): 169-174.

[4] Zhou Lian. Exploration of the working principle and application of word2vec [J]. Sci-tech Information Development and Economy, 2015, 25 (2): 145-148.

[5] Tang Ming, Zhu Lei, Zou Xianchun. Document vector representation based on word2vec [J]. Computer Science, 2016, 43 (6): 214-217.

[6] Zhang Dongwen, Xu Hua, Su Zencai, et al. Chinese comments sentiment classification based on word2vec and SVM perf [J]. Expert Systems with Applications, 2015, 42 (4): 1857-1863.

[7] Yin Yaoming, Zhang Dongzhan. Sentence similarity computing based on relation vector model [J]. Computer Engineering and Applications, 2014, 50 (2): 198-203.

[8] Tian Junfeng, Lan Man, Wu Yuanbin, et al. An adversarial joint learning model for low-resource language semantic textual similarity [C]// Advances in Information Retrieval. 2018: 89-101.

[9] Miao Xuelian. Comparative study of sequential pattern mining with gap constraints [J]. Network Security Technology and Application, 2017 (2): 66-67.

[10] Krishna B. PrefixSpan: Mining Sequential Patterns by Prefix-Projected Pattern [J]. International Journal of Computer Science & Engineering Survey, 2012, 2 (4): 111-122.

[11] Le B, Hai D, Truong T, et al. FCloSM, FGenSM: two efficient algorithms for mining frequent closed and generator sequences using the local pruning strategy [J]. Knowledge & Information Systems, 2017, 55 (3): 1-37.

[12] Yan Xiaowu, Zhang Jifu, Xun Yaling, et al. A parallel algorithm for mining constrained frequent patterns using MapReduce [J]. Soft Computing, 2017, 21 (9): 2237-2249.

[13] Wu Youxi, Zhou Kun, Liu Jingyu, et al. Mining Sequential Pattern with Periodic General Gap Constraints [J]. Chinese Journal of Computers, 2017, 40 (6): 1338-1352.

[14] Mooney C, Roddick J. Sequential pattern mining: approaches and algorithms [J]. ACM Computing Surveys, 2013, 45 (2): 1-39.

[15] Aloysius G, Binu D. An approach to products placement in supermarkets using PrefixSpan algorithm [J]. Journal of King Saud University of Computer & Information Sciences, 2013, 25 (1): 77-87.

[16] Wright A, Wright T, Mccoy A, et al. The use of sequential pattern mining to predict next prescribed medications [J]. Biomedical Informatics, 2015, 53 (C): 73.

Note: Figure translations are in progress. See original paper for figures.