Journal of Process Control 56 (2017) 69–78

Contents lists available at ScienceDirect

Journal of Process Control

journal homepage: www.elsevier.com/locate/jprocont

Online pattern matching and prediction of incoming alarm floods☆

Shiqi Lai [a,*], Fan Yang [b], Tongwen Chen [a]

[a] Department of Electrical and Computer Engineering, University of Alberta, Edmonton T6G 2V4, Canada
[b] Tsinghua Laboratory for Information Science and Technology, Department of Automation, Tsinghua University, Beijing 100084, China

ARTICLE INFO

Article history:
Received 30 September 2015
Received in revised form 24 October 2016
Accepted 10 January 2017

Keywords:
Time-stamped sequences
Online sequence matching
Alarm prediction
Alarm flood analysis
Industrial alarm monitoring

ABSTRACT

Alarm floods are serious hazards for industrial process monitoring. In this paper, we propose an online algorithm to provide early prediction of an incoming alarm flood by matching an online alarm sequence with a pattern database and conducting similarity calculation. It overcomes three main challenges in online time-stamped pattern matching: the partial information (future events are unknown in the online sequence), the high computational efficiency requirement, and the segmentation of the online alarm sequence. New elements introduced in this algorithm include chattering and sequence window filters, modified time distance and similarity measurements, a modified gap penalty, and an incremental dynamic programming strategy. Potentially, the proposed algorithm could serve as the state identification stage in applications of predictive alarming, online alarm attribute modification, and online alarm suppression. A dataset from a real chemical plant is used to test both efficiency and accuracy of the proposed algorithm.

1. Introduction

The effectiveness of industrial process monitoring depends heavily on alarm systems. As new sensor and network technologies appear, it has become much easier to configure alarms with no hardware cost in industrial facilities. However, if alarm configurations are not rationally designed, the problem of excessive alarm messages would impact negatively the efficiency or even the safety of plant operations due to distracted information provided to operators. An alarm flood is an extreme case of this problem, during which the operator efficiency of handling important alarms is usually reduced significantly because of the overwhelming workload created by the numerous alarm messages. In order to identify this problem, both ISA [1] and the EEMUA [2] standards have suggested to set the alarm rate benchmark for alarm floods to be 10 alarms per 10 min per operator. Based on this benchmark, the alarm system undergoes an alarm flood when the alarm rate is higher than 10 alarm per 10 min per operator. For offline studies, an alarm flood sequence is formed by the alarm messages raised during the period when the alarm rate is higher than the benchmark.

There are two steps to reduce alarm floods. The first is to remove univariate alarms by applying techniques such as delay timers and dead-bands. Design techniques for delay timers and dead-bands have been introduced in [3–5]. The second step to reduce alarm floods is to study consequence alarms (alarms that have correlation with each other) because they form another important part of an alarm flood. For example, when the compressor trips, alarms for low speed, low oil pressure, high suction pressure, low discharge pressure, and low amps are usually raised successively as a pattern. This type of consequence alarms are triggered by one or more certain faults and appear frequently in the alarm and event logs. Since the consequence alarms are annunciated successively, they can easily raise the alarm rate and trigger alarm floods. Pattern mining algorithms are commonly used among the offline methods to find pattern sequences in alarm floods. Then the patterns can be used in root cause analysis (the compressor trip as the root cause in the example), locating bad configurations in alarm systems (e.g., redundant measurement on the same process), and operator trainings (train operators to deal with the consequence alarms triggered by frequent faults). Advanced methods include predictive alarming to warn the operator of oncoming alarms, online alarm attribute modification, and online alarm suppression, as mentioned but not fully developed in the EEMUA standard [2]. These techniques, however, all depend on matching the online alarm sequence with patterns previously found from historical alarm data.

☆ This work was supported by NSERC CRD and NSFC (No. 61433001) projects. The authors are grateful to Suncor for providing the data in the case study.
* Corresponding author.
E-mail addresses: slai3@ualberta.ca (S. Lai), yangfan@tsinghua.edu.cn (F. Yang), tchen@ualberta.ca (T. Chen).

http://dx.doi.org/10.1016/j.jprocont.2017.01.003
0959-1524/

1.1. State of art concerning offline alarm flood pattern matching

Expert consultation and operator's experience, by far, are still the two approaches that have been used by most industrial companies when dealing with consequence alarms in an alarm flood. Expert consultation provides good and accurate results; however, without doubts, its efficiency is extremely low because of the involvement of a relatively large amount of process knowledge. The approach based on the operators' experience is usually faster, but its accuracy is not guaranteed since it depends heavily on human judgments.

Many pattern mining algorithms have been developed to facilitate the study of consequence alarms in alarm floods. The authors in [6] proposed a context-based segmentation algorithm to extract patterns from alarm sequences; they made use of the status (alarm and return to normal) of a target tag together with several filters to determine the extraction. A pattern matching algorithm based on dynamic time warping (DTW) was proposed in [7] to find the global alignment of two alarm flood sequences; an alignment score of the two sequences was used to measure their similarity thereafter. In the same paper [7], the authors also applied hidden Markov models to alarm flood sequences and used the Euclidian distances between the transition matrices to measure the similarities. Another alignment algorithm based on dynamic programming was proposed in [8]; compared to the one based on DTW [7], this algorithm has two major improvements: (1) searching for the optimal local alignment rather than the global one, and (2) taking the time information into consideration while aligning the sequences. A multiple sequence alignment method [9] was also developed to extend the algorithm in [8] to the case of aligning multiple alarm floods. The ideas of frequent pattern mining have also been adopted for the pattern matching of alarm floods. In [10], the authors developed an algorithm to find patterns in alarm flood sequences in a pattern-growth manner; however, as they remarked, the method was not robust enough to disturbance alarms (alarms appear in a pattern sequence but not related to the cause of the pattern). In order to improve robustness to the disturbance alarms and take the time information into consideration, the method proposed in [11] was applied to discover frequent sequences in alarm floods. In [12], a dissimilarity based method was proposed to extract alarm sequence templates of given faults and a Needleman-Wansch based algorithm was developed to isolate alarm sequences caused by certain known faults in [13]. In [14], the authors proposed a method to re-order alarms during alarm floods by assigning their priorities values and designed an ecological interface for a better operator support during flood scenarios.

1.2. Current status of online sequence matching

The algorithms introduced in the previous subsection are all offline methods, as the full sequences need to be available before matchings are carried out. However, for the online matching of alarm floods, the online sequence only contains the current and past alarm messages, as future alarms are unknown. Moreover, each time when there is a new alarm raised and added to the online alarm sequence, the matching between the online sequence and the patterns in the database need to be re-conducted, which significantly increases the requirement for computational efficiency. Segmentation of the online sequence is another obstacle for online alarm flood matching since as new alarms continue to be annunciated, the online sequence grows longer and longer; it should be segmented before matchings are carried out. These are the three main reasons why the offline pattern mining methods cannot be applied to online matching directly. To the best of our knowledge, there is no algorithm till now capable of matching alarm sequences online.

However, some online algorithms have already been developed to match text sequences. In [15], the authors introduced a consecutive suffix alignment problem: given the matching result of the longest common subsequence (LCS) or the edit distance between two sequences A and B, incrementally compute the answer for A and bB, and the answer for A and Bb, where "b" is an added item. In the same paper, the authors proposed an algorithm to handle this problem that runs in $\mathcal{O}(kn)$ time and uses $\mathcal{O}(m+n+k^2)$ memory space, where m and n are the lengths of sequences A and B, k is the tolerated difference in edit distance, by taking advantage of the properties of dynamic programming matrices. The big O is the notation that characterizes the growth rates of the occupied space and computational complexity. In [16], the authors simplified the complicated properties of dynamic programming matrices and proposed an algorithm that runs in $\mathcal{O}((m+n)n)$ time and uses $\mathcal{O}(mn)$ memory space. The authors in [17] proposed another algorithm that improved space occupation to $\mathcal{O}(m+n)$. In [18], an algorithm with a preprocessing stage was introduced that runs in $\mathcal{O}(nL+n\log L)$ time and uses $\mathcal{O}(mn)$ space, where L was the length of the LCS between A and B.

Unfortunately, since text data does not contain time information and the LCS alignment problem is much simpler than the standard alignment problem (with gap and mismatch penalties), the dynamic programming matrix used for finding the LCS between text sequences has many essential monotonicity properties with respect to diagonals [15] that are not compatible to the problem of aligning alarm sequences. Thus the algorithms introduced above cannot be used for the online alarm sequence matching problem directly.

1.3. Alarm flood analysis procedures

As the online sequence matching serves as one step in the alarm flood analysis, we briefly introduce the procedures of alarm flood analysis in this subsection. There are mainly six steps:

(1) remove univariate alarms in alarm message log by applying techniques such as delay timers;
(2) extract alarm floods based on the suggested alarm rate threshold, e.g., 10 alarms per 10 min per operator;
(3) apply offline pattern matching algorithms to compute the pairwise similarity scores between the extracted alarm floods;
(4) cluster the alarm flood sequences into similar groups based on their similarity scores;
(5) carry out pattern mining to find a common pattern for each of the cluster and form a pattern database with the discovered patterns;
(6) use the proposed algorithm to match the online alarm sequence with the pattern database and apply advanced alarm management techniques such as predictive alarming, online alarm attribute modification, and online alarm suppression.

1.4. Contribution of our work

In this paper, we propose an algorithm that can monitor the online alarm systems for certain alarm sequence patterns, which potentially can be used to provide operator warnings of oncoming alarms and their root causes to give operators more time to take actions (predictive alarming), to automatically suppress oncoming alarms in the pattern sequence that do not provide valuable information for operator decision making (online alarm suppression), and to automatically modify alarm attributes when the pattern is operating mode related (online alarm attributes modification). Our main contributions are: