[Wang* 4(3): March, 2017] ISSN 2349-4506
Impact Factor: 2.785

Global Journal of Engineering Science and Research Management

AN UNCERTAIN COMPLEX EVENT PROCESSING METHOD BASED ON MNFA

Pengcheng Cai, Yongheng Wang*
* College of Information Science and Engineering, Hunan University, Changsha 410082, China

DOI: 10.5281/zenodo.376577

KEYWORDS: Complex Event Processing, Uncertain Event Streams, NFA, Pattern Recognize

ABSTRACT
A plethora of data streams are generated every day due to the rapid development of the Internet. In the era of big data, real-time data stream processing becomes more and more popular in data mining. However, traditional techniques can no longer satisfy our needs, because of noise, sensor errors and other natural factors within data streams. In this paper, on the basis of complex event processing (CEP) technology, we propose a method combining NFA with match buffer (mNFA) using probability theory to address the issue. Our method can not only process massive uncertain event streams efficiently, but also support probabilistic event streams query processing. Furthermore, we also apply dynamic probability calculation algorithm and filtering event early to optimization.

INTRODUCTION
The size of data generated every day grows exponentially as a result of the rapid development of technology. We can utilize complex event processing (CEP) [1] to address this issue. Internet of Things (IOT) system consists of a wide range of interconnected devices, like cell phones, cars, sensors and cameras. These devices produce a large number of signals, and the primitive events generated by these raw signals are received by the sensor. However, high-level system and users can't process raw events directly. With middleware or event-driven system based on the original event preprocessing, users can have a sense of the high-level events — complex events. However, dirty reads and loopholes make the original data generated by the internet with uncertainty that results in a misreading in the transmission or identification process. The data can be categorized into two groups: unreliable data and uncertainty data [2]. Unreliable data result from errors and incompleteness read by RFID, and can be cleaned up using specific constraint rules or removed directly. Uncertainty data are generated because the limitations of sensor's precision or some natural factors make the appearances of data source attributes are with probabilities or fall into a certain interval range. Thus uncertainty data can't be eliminated by defined determinate rules. In this paper, we analyze the second kind of uncertain event, and convert them into complex probability event streams. As probabilistic event stream is prevalent in real-world applications, it should be meaningful to deal with uncertain event streams in academic and industrial fields using complex events.

However, some existing complex event processing systems, such as Esper [3], SASE [4] and Cayuga [5], assume that data are precise and certain, or are cleansed before processing. They therefore failed to consider the uncertainty and detect probabilistic complex event. Recently, Lahar [6] and Cascadia [7] systems are proposed to process uncertainty within RFID data. Since they focus mainly on the probabilistic-based data model and related queries, in-depth studies are required to detect interest event in uncertainty event streams and probability event streams. Two challenges of complex event detection from probabilistic event streams are the detection of complex events from real-time updating probabilistic event streams and calculating the probability of the complex event generated from relevant uncertain events [8]. In this paper, we propose a method based on mNFA and probability theory to deal with uncertainty event streams. Our method can not only process massive uncertain event streams efficiently and real-time, but also support probabilistic event streams query processing. We also use dynamic probability calculation algorithm to optimize our results.

RELATED WORK
In recent years, CEP for uncertain data becomes more and more popular. Correia [9] combines various scenarios to establish a complex event processing network and monitor uncertain events in real time. Currently, many complex event processing systems based on RFID middleware take uncertainties into consideration. Kimelfeld [10] uses a statistical model and Markov sequences to represent uncertain RFID data streams, and match them

using Markov sequences. Segev [11] proposed a rule-based complex event processing framework, using the probability theory to represent uncertain complex event rules, and then calculating probability of complex events using Bayesian network and sampling. In order to support pattern evaluation in an uncertainty event streams (event streams with imprecise timestamp), Zhang H [12] propose a temporal uncertainty of event model to analyze temporal uncertainty of event occurrence. Formal semantics of schema evaluation is presented in this model, the calculation is for point-based and event-based framework, respectively, and the corresponding frameworks are optimized accordingly. H. Kawashima [13] extends NFAb to DFAb on the basis of SASE+, and uses matching binary tree to calculate the complex event probability. Wang [14] proposed an efficient method to deal with events in a distributed probability event streams by combining a partition active instance stack (PAIS) with an indefinite finite state machine (NFA).

Although there are a variety of existing techniques being applied to complex event processing framework and the probability of complex event processing, few of them focus on the detection of uncertain complex events. Thus there is still much room for the optimization of probability calculation algorithm of uncertain events.

EVENT QUERIES

In this paper, we use the pattern matching query language in SASE, and extend the uncertainty event query UCEQL based on it.

Table 1. Query describe in UCEQL.

Query1:	Query2:
PATTERN SEQ (A a , B b , D d) WHERE skip_till_next_match(a,b,d) AND a.type = ‘A’ AND b.type = ‘B’ AND c.type = ‘D’ AND c.location > 12 CONF 0.5 WITHIN 100s	**PATTERN SEQ (A a , B+ b[] , ! D d) WHERE skip_till_any_match(a,b[]) AND a.type = ‘A’ AND b.type = ‘B’ AND a.location > 12 AND b[i].volocity <= min(b[..i-1].volocity) CONF 0.5 WITHIN 1hour**

Table 1 show two different query, PATTERN field declaration query uses the SEQ constructor. This sequence constructor contains three components, A, B and D three types of events. WHERE field declares the event selection strategy as skip_till_next_match in SASE+, at the same time, the event instances of each component are also required to satisfy the corresponding predicate expression in AND. Only the event instances that satisfy these predicate expressions are selected by the SEQ constructor. CONF field indicates that the probability value of a complex event that matches the PATTERN can’t be less than 50%. WITHIN field declares the sliding window of the PATTERN, only the event instances within the window are valid. We can use this SEQ pattern to describe a car passing through A, B, D three junctions, and in the intersection away from the intersection of D greater than 12,that indicating that the section of traffic smoothly.

Compared to Query 1, the event selection policy of Query 2 is used skip_till_any_match, and the predicate contains the aggregation operation min. This mode can indicate that a vehicle passing through the intersection of A, after crossing near the intersection of B to detect the number of vehicles, the speed is less than the original minimum speed, D intersection without vehicles, indicating junction B traffic jams or traffic accidents of uncertain complex events.

CEP UNDER UNCERTAINTY

CEP system is the core part of the pattern of event matching, as mentioned in the previous section, these match patterns include SEQ, ANY, Negative, and so on. In order to facilitate the analysis, this paper mainly studies the

ISSN 2349-4506
Impact Factor: 2.785

Global Journal of Engineering Science and Research Management

uncertain event matching in SED Pattern. In addition, the method and model of this paper are realized under the premise that the event is orderly.

The traditional NFA-based CEP has two drawbacks: First, for the treatment of massive event streams, efficiency will become very inefficient. Second, can't deal with uncertain event streams. Therefore, this paper presents a method based on mNFA and probability theory to deal with the event streams, needn't to maintain AIS, thereby enhancing performance. mNFA is a combination of match buffer and NFA data structure, each mNFA instance is composed of a subset that satisfies PATTERN. When fully meet the conditions will reach the final state, the output event, otherwise it discarded. Fig 1 shows the basic process of SEQ event processing in mNFA.

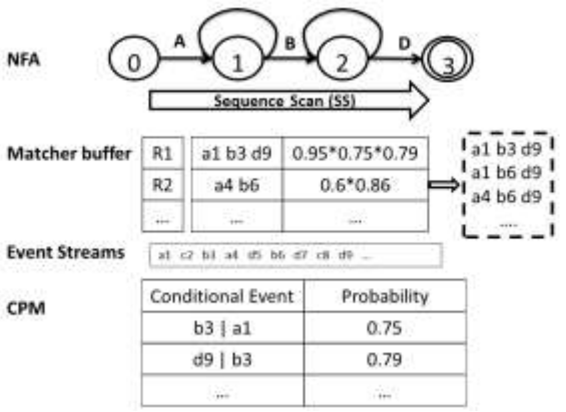

Figure 1. The basic process of SEQ event processing based on mNFA.

Considering that the input event stream is uncertain, complex events detected after CEP processing may be uncertain. To better represent the probabilistic properties of complex events, we propose a probabilistic computational model.

Unrelated and Markovian events exist in the probability event stream, we classify the input event stream of CEp = SEQ (e1, e2, ... e3) into two sets of events: A set of M represents one or more Markov chains, and a set of N represents the independence of the original events. So CEp calculation can be expressed as:

$$P(CEp) = \prod_{e_i \in N} Pro(e_i) * \prod_{m_i \in M} \left(Pr(e_{i1}) \prod_{j=1}^{|m_i|-1} Pr(e_{j+1}|e_j)\right) \quad (1)$$

For events with Markov properties, the conditional probability can be managed by the conditional probability matrix (CPM) matrix, CPM is derived from historical statistics, of course, can also be learned through the machine and regularly updated.

In order to detect complex events in probabilistic event streams, we extend the NFA model and propose a probabilistic detection algorithm based on mNFA. This work takes the existing complex event processing system SASE as the prototype. The basic idea of our algorithm is to use the mNFA model to detect uncertainty events that satisfy matching complex events and store them through instances of mNFA. When an event is selected and added to the instance of mNFA, we are able to compute the probability of the current instance, and when the instance is in an acceptable state, a match is output. The algorithm is as Fig 2:

Algorithm 1. Probabilistic event detection algorithm

Input: Uncertain event stream s, complex event query plan q
Output: Complex event set CE
Method:

1. initialize NFA and build NFA through q;
2. initialize run instance collection:activeRuns;

3. for all instance e in the event stream s of uncertainty do
4. if e == NFA.state(start) then
5. create a new run instance r, and then add the event e to r,
according to e.probability initialize the probability of r,
add r into activeRuns;
6. else if e == NFA.state(transition) then
7. for all run r in activeRuns do
8. check the transition predicate;
9. if checkResult == true then
10. add an event to the current active instance r;
11. query CPM Calculates the probability of the current r;
12. else discard event e
13. if r == NFA.state(accepted) then
14. check the time window of r ;
15. if r.timespan < time window then
16. add d to CE, and output the probability of r corresponding;
17. remove r from the activeRuns collection;
18. else discard e
19. end

Figure 2. Probabilistic event detection algorithm.

For a better understanding of Algorithm 1, we will show the processing of Query 1 according to figure 1. When the time a1 arrives, it triggers the start state of the NFA structure generated from Query 1, from "0" to "1". We add a1 to a new run instance R1, and calculate the probability dimension of R1 of 0.95. Because c2 does not satisfy the NFA start-up state and transition state, it is discarded. The arrival of the b3 event triggers the transition state of the NFA from "1" to "2" and adds b3 to R1, and according to the conditional probability table query the corresponding probability value to calculate, the probability of R1 at this time is 0.95 * 0.75. The arrival of a4 will trigger the start state, so a new instance R2 will be created. Events d5 and b6 are as above. The arrival of d7 triggers an NFA transition from "2" to "3". Since d7 only satisfies R1, only d7 is added to R1 and the R1 probability is calculated as 0.95 * 0.75 * 0.79. At this point the NFA has reached the final state, output complex event matching (a1, b3, d7) probability value of 0.56, activieRuns delete R1, at the moment a complex event processing is complete.

Careful study of the process of algorithm 1, will find that c2, c8 are independent of the incident, in the course of the incident and no practical significance. So we can filter out the unrelated events in advance through a filter, can improve the processing efficiency. Reference to this idea can further optimize the algorithm. Since the probability values are between 0 and 1, the probability of the partial match must be less than or equal to the probability of an exact match. First, the part of the running instance whose probability value is less than the threshold is filtered out and can further enhance the efficiency, this algorithm 2, as Fig 3:

Algorithm 2. Probability event detection algorithm optimization

Input: Uncertain event stream s, complex event query plan q, threshold k
Output: Complex event set CE
Method:
1. initialize NFA and build NFA through q;
2. initialize a run instance collection:activeRuns;
3. for all instance e in the event stream s of uncertainty do
4. if e ∉ NFA. eventModel then

ISSN 2349-4506
Impact Factor: 2.785

Global Journal of Engineering Science and Research Management

```
     discard e;
   else if update run with Algorithm 1 then
     if r.probability <= k || r.timespan > time window then
       remove running instance from r activeRuns;
     else if r == NFA.state(accepted) && r.timespan < time window then
         add r to CE, and the output of the corresponding probability of r;
         remove r from the activeRuns collection;
end
```

Figure 3. Probability event detection algorithm optimization.

EXPERIMENTAL ANALYSIS

Our experiments are taken to compare the performance of the proposed uncertainty complex event process system (UCEP) with SASE. We first analyze the effect of stream size on the experimental results.

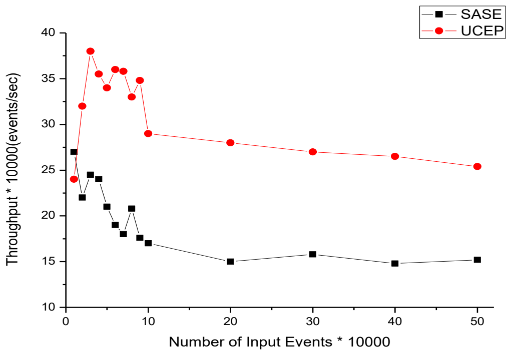

Figure 4. Throughput over different sizes of input event streams.

As shown in Fig 4, the performance of the improved algorithm as a whole is higher than that of the prototype system. SASE is higher than UCEP when the number of input events is less than 20000 because UCEP had a better performance in large event stream. After 30000, UCEP reach the maximum capacity of the system. Under the same event type and time window, the processing efficiency of the prototype system is obviously reduced and the UCEP is relatively less affected with the increase of the event stream.

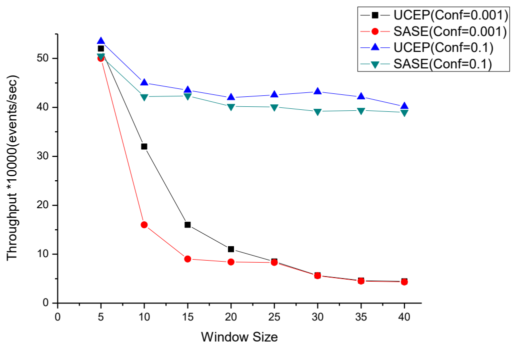

Figure 5. The impact of Conf and time window on throughput.

As shown in Fig 5 can be seen in the same event source confidence threshold decreased, the system processing efficiency decreased significantly. On the contrary, the larger the confidence threshold, the higher the efficiency of system processing, Figure shows a gentle curve, indicating that the design of high confidence values can improve system efficiency and smoothness. Because when a confidence threshold high that it can discard partial matches whose Conf less than threshold to improve the performance. For example, our optimized proposal can process about 42000 events per seconds with the 0.1 confidence threshold. Besides, as the length growing of time window, the throughput decreases rapidly when their size is less than 10 and then in a relatively smooth. This is because the time window is much larger than the length of PATTERN which leads to longer time to detect event in input stream.

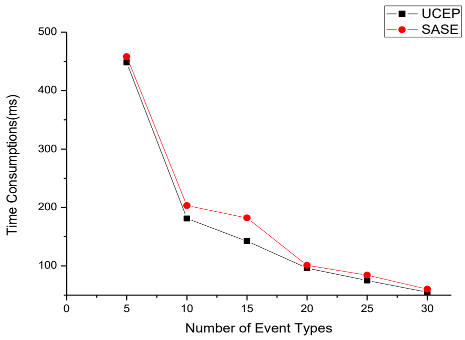

Figure 6. Time consumption over different sizes of event types.

See Fig 6 shows that the time consumption of SASE and UCEP has a little different when the number of event types is less than 5. Because our PATTERN only has three valid components and our optimized method can filter a few events. When the event types increases, optimized method gets a better performance that it filters irrelevant events for matches.

CONCLUSION
We analyze a problem of complex event processing system, and propose to use a mNFA-based probabilistic event stream processing algorithm to optimize it. Since incomplete knowledge domain could cause another kind of uncertainty. We plan to adopt a general uncertainty reasoning model using Bayesian network and Markov-logic network. Our method is efficient in uncertainty event stream processing.

ACKNOWLEDGEMENTS
This paper is based on the National Natural Science Foundation of China, "Research on Active Complex Event Processing Technology for Large-scale Internet of Things" (No.61371116).

REFERENCES
1. D.C. Luckham, The Power of Events: An Introduction to Complex Event Processing in Distributed Enterprise Systems. Boston: USA. 2001.
2. Y. Nie, Z. Li and Q. Chen, “Complex Event Processing over unreliable RFID Data Streams”, in 13th Asia-Pacific Web Conference (APWeb 2011), pp 278-289, April 2011.
3. Esper Tech [Online]. Available: http://www.espertech.com.
4. D. Gyllstrom, E Wu, HJ Chae, Y Diao. “SASE: Complex Event Processing over Streams”. Rro of CIDR, pp.407-411, Dec 2007.
5. L. Brerma, A. Demers and J. Gehrke. “Cayuga: a high-performance event Processing engine”, in proeeedings of the 2007 ACM SIGMOD international conference on Management of data. pp.1100-1102, June 2007.
6. C. Ré, J. Letchner and M. Balazinksa. “Event Querieson Correlated Probability Streams”. Proc of the SIGMOD Conf. pp.715-72, June. 2008.
7. E. Welbourne, N. Khoussainova. “Cascadia: a system for specifying, detecting, and managing RFID events”, in International Conference on Mobile Systems, Applications, and Services. 2008:281-294.
8. C. Xu, S. Lin and W. Lei. “Complex Event Detection in Probabilistic Stream”, in Proceedings of the 12th International Asia-Pacific Web Conference (APWeb 2010). pp. 361-363, April.2010.
9. I. Correia, F. Fournier and I. Skarbovsky. “The uncertain case of credit card fraud detection”, in the, ACM International Conference on Distributed Event-Based Systems. 2015:181-192.
10. B. Kimelfeld, C. Re. “Transducing markov sequences”, in Proceedings of the twenty-ninth ACM SIGMOD-SIGACTSIGART symposium on Principles of database systems of data. pp.15-26, June.2010.

11. S. Wasserkrug, A. Gal and O. Etzion. “Efficient processing of uncertain events in rule-based systems”, IEEE Transactions on Knowledge and Data Engineering, TKDE(2010):45-58.
12. H. Zhang, Y. Diao and N. Immerman. “Recognizing Patterns in Streams with Imprecise Timestamps”,in Proceedings of the Vldb Endowment, 2013, 38(8):1187-1211.
13. H. Kawashima, H. Kitagawa and X. Li. “Complex Event Processing over Uncertain Data Streams”, in Proceedings of the fifth international conference on P2P, parallel, grid, cloud and internet computing.pp.521- 526, Nov,2010.
14. Y. H. Wang, K. Cao and X. M. Zhang. “Complex Event Processing over Distributed Probabilistic Event Streams”., in 2012 9th International Conference on Fuzzy Systems and Knowledge Discovery,pp.1489-1493,May. 2012.