ijecs open access www.ijecs.in
International Journal Of Engineering And Computer Science ISSN: 2319-7242
Volume 4 Issue 12 Dec 2015, Page No. 15142-15147

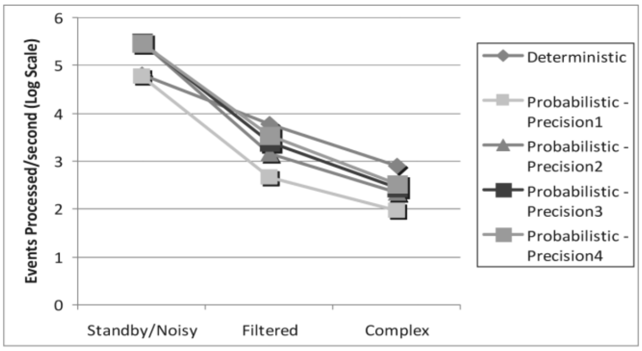
Vibrant Rule Base Erection and Continuance plot for Ailment Profecy

Ms. M. Mythili , Mr. A.P.V.Raghavendra
M.E., AP/CSE
mythilimurugan7@gmail.com
Department of Computer Science and Engineering
V.S.B Engineering College
Karur,India

M.Tech (Ph.D), AP/CSE
raghu221084@gmail.com
Department of Computer Science and engineering
V.S.B Engineering College
Karur,India

Abstract:-
Business and healthcare application are tuned to automatically detect and react events generated from local are remote sources. Event detection refers to an action taken to an activity. The association rule mining techniques are used to detect activities from data sets. Events are divided into 2 types' external event and internal event. External events are generated under the remote machines and deliver data across distributed systems. Internal events are delivered and derived by the system itself. The gap between the actual event and event notification should be minimized. Event derivation should also scale for a large number of complex rules. Attacks and its severity are identified from event derivation systems. Transactional databases and external data sources are used in the event detection process. The new event discovery process is designed to support uncertain data environment. Uncertain derivation of events is performed on uncertain data values.

***Index Terms*-Association, selectability, derivation, sampling, approximation**



1. INTRODUCTION

Medical data mining has great potential for exploring the hidden patterns in the data sets of the medical domain. These patterns can be utilized for clinical diagnosis. However, the available raw medical data are widely distributed, heterogeneous in nature, and voluminous specialty and moreover there is a shortage of resource persons at certain places. Therefore, an automatic medical diagnosis system would probably be exceedingly beneficial by bringing all of them together. Appropriate computer-based information and/or decision support systems can aid in achieving clinical tests at a reduced cost. Efficient and accurate implementation of automated system needs a comparative study of various techniquesavailable.

information system. Data mining technology provides a user oriented approach to novel and hidden patterns in the data. Healthcare has been one of the top demands of this generation. In the advent of technology, the provision of healthcare continues to improve. One of the top priorities is the provision of diagnosis, and the one responsible for this is the physician. Physician's diagnosis is the most relevant factor that leads to the acquisition of proper health guides. As technology soars, there are changing medical requirements and solutions, and sometimes physicians are not updated with these upgrades, that they need to surf the internet for more information.

Medical diagnosis is regarded as an important yet complicated task that needs to be executed accurately and efficiently. The automation of this system would be extremely advantageous. Regrettably all doctors do not possess expertise in every sub.

One of the solutions that could aid in the physician's diagnosis is the Clinical Diagnostics Decision Support System (CDSS). The CDSS is generally defined as any computer program designed to help health professionals make clinical decisions. Clinical Decision Support Systems supports case-specific advice which addresses to the aiding of physician's diagnosis via computer-based facility. Since physicians sometimes encounter complicated ailments, a CDSS can make decision making simpler by providing relevant pre-diagnosis. With the use of CDSS, a prediagnosis could be extracted which can strongly support the physician's decision. Hence, lesser time could only be consumed in the provision of diagnosis to the patient. Along with this CDSS is a logic which provides probabilistic conditions that leads to a specific outcome, which is the Fuzzy Logic; another one is the method to be taken, which is the rule-based method.

II. RELATED WORK

Complex event processing is supported by systems from various domains. These include ODE, Snoop and others for active databases and the Situation Manager Rule Language, a general purpose event language. Event management was also introduced in the area of business process management [3] and service-based systems [7]. An excellent introductory book to complex event processing is also available. A recent book introduces principles and applications of distributed event based systems [6]. Architectures for complex event processing were proposed, both generic and by extending middleware.

The majority of existing models do not support event uncertainty. Therefore, solutions adopted in the active database literature, such as the Rete network algorithms fail to provide an adequate solution to the problem, since they cannot estimate probabilities. An initial rule-based approach for managing uncertain events was proposed in [14]. This work was followed by a probabilistic event language for supporting RFID readings. Our proposed model shares some commonalities with, namely uncertainty specification at both the event occurrence and the rule level, and providing an algorithm for uncertain event derivation. However, unlike [4], our framework does not limit the type of temporal expressions it can handle nor does it limit attribute derivation types. We extend the work by providing algorithms for deriving uncertain events and empirical evidence to the scalability of the approach.

Recent works on event stream modeling propose formal languages to support situations such as event negations. Various aspects of event-oriented computing were discussed at CIDR 2007 (e.g., [2]). Our focus in this work is on modeling and efficiently managing uncertainty in complex event systems, which was not handled by these works. Modeling probabilistic data and events was suggested in [8]. This work extends those models by proposing a model and a probability space representation of rule uncertainty. A common mechanism for handling uncertainty reasoning is a Bayesian network, a method for graphically representing a probability space, using probabilistic independencies to enable a relatively sparse representation of the probability space. Qualitative knowledge of variable interrelationships is represented graphically, while quantitative knowledge of specific probabilities is represented as Conditional Probability Tables (CPTs). The network is, in most applications, manually constructed for the problem at hand. In our framework, we automatically construct a Bayesian network from a set of events and rules, following the Knowledge Based Model Construction (KBMC) paradigm. This paradigm separates uncertain knowledge representation from inference, which is usually carried out by transforming the knowledge into a Bayesian network that can model knowledge at the propositional logic level knowledge. The work also follows the KBMC paradigm. In this work, the quantitative knowledge, as well as the quantitative deterministic knowledge, are represented as a set of Horn clauses and the qualitative probabilistic knowledge is captured as a set of CPTs. Our framework, however, in that our framework need not be restricted to first order knowledge. For example, second order knowledge, or knowledge that can be expressed using any procedural language, can also be captured. In addition, in our framework, probabilities may have a functional dependency on the events themselves. For example, the probability assigned to a flu outbreak could be specified to be min (90 + (X - 350)=10, 100). In addition, our framework enables the uncertainty associated with such event derivation to be captured by different probability spaces at different points in time.

Existing works [9] for processing complex events in specific domains tailor probabilistic models or direct statistical models to the application. No general framework was defined there to derive uncertain events. Some details of our framework were presented in [13] with two significant extensions in this work.

DOI: 10.18535/Ijecs/v4i12.9

We discuss in details efficiency aspects of our algorithms, and in particular event selectability. We also significantly extended our empirical analysis to include more cases.

III. EVENT DRIVEN SYSTEMS (EDS)

In recent years, there has been a growing need for event driven systems, i.e., systems that react automatically to events. The earliest event-driven systems in the database realm impacted both industry and academia. New applications in areas such as Business Process Management (BPM) [5]; sensor networks [11]; security applications; engineering applications; and scientific applications all require sophisticated mechanisms to manage and react to events.

Some events are generated externally and deliver data across distributed systems, while other events and their related data need to be derived by the system itself, based on other events and some derivation mechanism. In many cases, such derivation is carried out based on a set of rules. Carrying out such event derivation is hampered by the gap between the actual occurrences of events, to which the system must respond, and the ability of event-driven systems to accurately generate events. This gap results in uncertainty and may be attributed to unreliable event sources, an unreliable network, or the inability to determine with certainty whether a phenomenon has actually occurred given the available information sources. Therefore, a clear trade-off exists between deriving events with certainty, using full and complete information, and the need to provide a quick notification of newly revealed events. Both responding to a threat without sufficient evidence and waiting too long to respond may have undesirable consequences.

In this work, we present a generic framework for representing events and rules with uncertainty. We present a mechanism to construct the probability space that captures the semantics and defines the probabilities of possible worlds using an abstraction based on a Bayesian network. In order to improve derivation efficiency we employ two mechanisms: The first mechanism, which we term selectability, limits the scope of impact of events to only those rules to which they are relevant, and enables a more efficient calculation of the exact probability space. The second mechanism we employ is one of approximating the probability space by employing a sampling technique over a set of rules. We show that the approximations this mechanism provides truly represent the probabilities defined by the original probability space. We validate the approximation solution using a simulation environment, simulating external event generation, and derive events using our proposed sampling algorithm.

IV. CHALLENGES IN EDS

A generic model is used for representing the derivation of new events under uncertainty. Uncertain derivations of events are performed on uncertain data values. Relevance estimation is a more challenging task under uncertain event analysis. Selectability and sampling mechanism are used to improve the derivation accuracy. Selectability filters events that are irrelevant to derivation by some rules. Selectability algorithm is applied to extract new event derivation. A Bayesian network representation is used to derive new events given the arrival of an uncertain event and to compute its probability. A sampling algorithm is used for efficient approximation of new event derivation. The following drawbacks are identified from the existing system.

- Static rule base model
- Limited accuracy in rule probability estimation
- Manages limited incoming events only
- Rule class and structure are not generalized

V. EVENT DERIVATION MODELS

The challenges associated with event derivation, note first that such events only suggest a high probability of a flu outbreak, which does not necessarily mean that such an event should be derived. We term this type of uncertainty derivation uncertainty, as it stems from the inability to derive such events with certainty from the available information. In addition, there is uncertainty regarding the data itself, because the provided data are rounded to the nearest ten. For example, the increase from November 30 to December 4 is of 350 units, yet rounding may also suggest a smaller increase of 341 units. This is considered uncertainty at the source, resulting from inaccurate information provided by the event source. Additional details regarding the different types of uncertainty can be found in [1].

EID	Date	Daily Sales (Rounded)
113001	Nov 30	600
120101	Dec 1	700
120201	Dec 2	800
120301	Dec 3	900
120401	Dec 4	950
120501	Dec 5	930

TABLE 1: Over-the-Counter Sales Relation

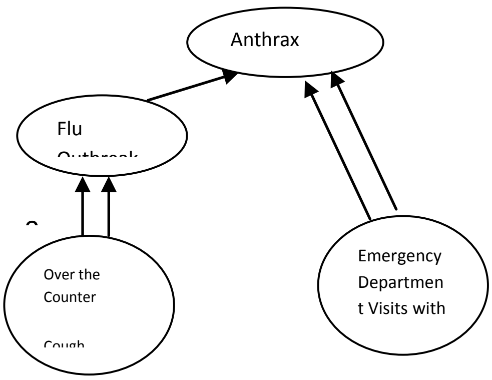


Fig. 1. Anthrax rule.

To highlight the complexity of the processing uncertain events and rules we note that the rule presented above is a simplified version of rules expected to exist in the real world. For example, instead of just specifying a single certainty level of 90 percent, one could compute the probability of an outbreak as a function of the amount of increase. Note that while such a rule may be sufficient to capture in some settings the derivation uncertainty, it does not capture uncertainty at the source.

We also note that real-life applications need to process multiple rules with multiple data sources [10]. To illustrate, consider Fig. 1, where we need to consider the possibility of an anthrax attack, in addition to the flu outbreak. In this setting, data must be combined from events that come from different data sources. A lower probability should be assigned to an anthrax attack whenever a flu outbreak is recognized and a higher probability otherwise. As a result, a flu outbreak event is not derived; a significant increase of hospital emergency department visits with respiratory complaints is enough to assign a probability of 60 percent to the occurrence of an anthrax attack event. Other examples of more complex settings may involve correlating data between pharmacies or across regions. Therefore, it must be possible to specify a set of rules that captures this complexity.

VI. PROBABILISTIC EVENT MODEL

A.Event Model

An event is an actual occurrence or happening that is significant and atomic. Examples of events include termination of workflow activity, daily OtCCMS and a person entering a certain geographical area. We differentiate between two types of events. Explicit events are signaled by external event sources. Derived events are events for which no direct signal exists, but rather need to be derived based on other events, e.g., Flu Outbreak and Anthrax Attack events. Data can be associated with an event occurrence. Some data types are common to all events, while others are specific. The data items associated with an event are termed attributes.

B. Derivation Model

Derived events in our model are inferred using rules. For ease of exposition, we refrain from presenting complete rule language syntax. Rather, we represent a rule by a quintuple, r = $\langle s, p, q, m, p_r \rangle$ defining the necessary conditions for the derivation of new events. Such a quintuple can be implemented in a variety of ways, such as a set of XQuery statements, Horn clauses, and CPTs such as in [12], or as a set of procedural statements. We detail next each of the rule elements, illustrating them with the rule r_1: “If there is an increase in OtCCMS for four sequential days to a total increase of 350, then the probability of a flu outbreak is 90 percent.” Recall that OtCCMS events contain the volume of the daily sales.

C. Event Detection Algorithms

Selectability, as defined by function s_r in a rule specification, plays an important role in event derivation, in both the deterministic and the uncertain settings. In our setting, the relevance of an EID to derivation according to rule r is determined by the function s_r. s_r is a function $s_r : h \rightarrow h_r$ that receives an event history as input, and returns a subset $h_r \subseteq h$. As an example of a function s_r consider the selection function corresponding. Given an event history h, it returns a set consisting of all possible events e that indicate an increase in OtCCMS in two consecutive days. Formally, this would be defined as all possible events e such that e may be one of a pair of events $\{e_1, e_2\}$ in h such that both e_1 and e_2 are OtCCMS events, e_2 occurred one day after e_1, and the number of cough medication sold as indicated by e_2 is greater than the number of cough medication sold as indicated by e_1. It is the role of the selection expression to filter out events that are irrelevant for derivation according to r. An event $e \in h$ to be relevant to derivation according to rule r iff $e \in s_r(h)$.We also require that for every pair of event histories h, h’ such that $h \supseteq h0 \supseteq s_r(h)$ it must hold that $s_r(h) = s_r(h')$. (1)

Note that from (1) we have the following special case: $s_r(h) = s_r(s_r(h))$. (2)

C. Selectability Algorithm

As in the uncertain setting derivation is carried out on EIDs, algorithms are required to compute which EIDs, from a given system event history H, are selectable. Deciding whether an EID E is selectable by rule r may, by itself, incur significant computational effort. This is because, selectability depends on the possible event histories in which the event corresponding to E participates.

We propose an efficient algorithm based on a decomposition of s_r. We decompose s_r into two functions as follows: Let = $\{e_1, \ldots, e_n\}$, then s_r may be described by $cs_r(es_r(e_1)) \cup es_r(e_2) \cup \ldots \cup es_r(e_n))$, such that $es_r : e \rightarrow \{\{\}, \{e\}$.$es_r$ receives a single event as input, and returns as output either the empty set if it is clear from the attributes of this event that this event is not selectable by the rule, or the set containing the single event received as input otherwise. This representation enables distinguishing between a selection that can be carried out only by looking at an individual event and its attributes using the function e s_r, and filtering that requires looking at combinations of events in h, which is carried out by function c s_r. Note that such decomposition into e s_r and c s_r can always be carried out, since e s_r can, in the worst case, return the set containing the input event. In such a case, $s_r \underline{=} cs_r$.

DOI: 10.18535/Ijecs/v4i12.9

Based on this decomposition we provide an algorithm that, for each rule r, provides the subset of EIDs in a system event history that is selectable by r. In this algorithm the function es_r is first used to test selectability at the individual EID level. A set of EIDs will be constructed such that EID E is in the set iff one of the possible states of E corresponds to an event e that es_r returns as selectable. Following this, only the set of event histories defined by this subset of EIDs is checked for selectability, by using the c s_r function. The pseudocode of this algorithm—function calculateSelectableEIDs—appears in Algorithm 1.

Algorithm 1. calculateSelectableEIDs(H,r)

```
selectableEIDs ← ϕ
esEIDs ← ϕ,
for all E ∈ H
        for all e ∈ E
                if es_r (e) = {{e}}
                        esEIDs esEIDs ∪ E
                end if
        end for
end for
for all h ∈ esEIDs
        for all e ∈ s_r (h)
                E ← getCorrespondingEID(e)
                selectableEIDs ← selectableEIDs ∪ E
        end for
end for
return selectableEIDs
```

E. Sampling Algorithm

The algorithm described in the generates a Bayesian network from which the exact probability of each event can be computed. Given an existing Bayesian network, it is also efficiently possible to approximate the probability of an event occurrence using a sampling algorithm, as follows: Given a Bayesian network with nodes $E_1, \ldots, E_n$, we calculate an approximation for the probability that E_i = {occurred} by first generating m independent samples using a Bayesian network sampling algorithm. Then, $P_r(E_i = \{occurred\})$ is approximated by $\frac{\#(E_i = \{occ\})}{m}$, where $\#(E_i = \{occurred\})$ is the number of samples in which E_i has received the value occurred.

Algorithm 2. RuleSamp, triggered by a new event arrival

```
h ← ϕ
for all E ∈ H_0
        e ← probSampling(E)
        h ← h ∪ e
end for
order ← obtainTriggeringOrder
while order ≠ ϕ
        r ← deleteNextRule(order)
        h' ← ϕ
        selEvents ← s_r (h)
        if p_r(selEvents) = true
                assocT uples ← a_r (selEvents)
                for all tuple ∈ assocT uples
                        {s_1, . . . , s_n} ← m_r (tuple)
                        prob ← pr_r (tuple, m_r (tuple))
                        probSamp ←
sampleBernoulli(prob)
                        if probSamp = 1
                          e ← {occurred, s_1,. , s_n}
                        else
                          e ← {notOccurred}
                        end if
                        h ← h' ∪ {e}
                end for
        end if
        h ← h ∪ h'
end while
return h
```

VII. DISEASE PREDICTION WITH HYBRID RULE BASE MODEL

The event derivation system is enhanced to map dynamic rules on uncertain data environment. The rule base construction and maintenance operations are handled by the system. Rule probability estimation is carried out using the apriori algorithm. The rule derivation process is optimized for domain specific model. The system is designed to detect events on uncertain data environment. Dynamic rule base model is used in the system. The system integrates the rule base update process. The system is divided into six major modules. They are patient diagnosis, rule base management, sampling process, selection process, event detection and rule base update.

A. Patient Diagnosis

The system uses TB patient diagnosis information. Patient diagnosis data can be imported from external databases. The user can also update new patient diagnosis details. Diagnosis list shows the patient symptom levels.

B. Rule Base Management

Rule base is used to manage the rules and event names. Rule base details are collected from domain experts. The rules are composed with attribute combination and values. Three types of rules are used in the system. Static, dynamic and hybrid rules are maintained under the rule base.

C. Sampling Process

The sampling process is performed to select event derivation models. The attribute values are sampled from uncertain data collections. The sampling algorithm is used to select sample data derivations. Sampled data values are passed to rule analysis process.

D. Selection Process

The selection process is applied to filter irrelevant events. Item combinations are selected under the selection process. The selectability algorithm is used in the selection process. Event derivations are used in the rule analysis process.

E. Event Detection

The rule base analysis is performed with user data collections. The bayesian network is used in the event detection process. The attribute combinations are compared with rule base information. Event detection is carried out with rank and priority information.

F. Rule Base Update

The dynamic rules are derived from user data and static rule base information. The dynamic rules are updated with event name and priority values. Rule ranking is performed with frequency information. Rules are updated with priority details.

VIII. CONCLUSION

Rule bases are build to manage activities and their class information. Event derivation is carried out with rule bases. Dynamic rule base update model introduced to improve the event detection process. Selective and sampling algorithms are enhanced to derive events under dynamic and uncertain domain environments. The system integrates the experts knowledge and local domain information for rule base construction. Rule base generation and maintenance operations are done using machine learning models. Event classes and its structures are generalized. Association rule mining methods are used to extract rule patterns.

References

[1] L. Girod, R. Newton, S. Rost, A. Thiagarajan, H. Balakrishnan, and S. Madden, "The Case for a Signal-Oriented Data Stream Management System," Proc. Conf. Innovative Data Systems Research, 2007.

[2] R. Ammon, C. Emmersberger, T. Greiner, A. Paschke, F. Springer, and C. Wolff, "Event-Driven Business Process Management," Proc. Second Int'l Conf. Distributed Event-Based Systems, July 2008.

[3] M. Balazinska, N. Khoussainova, and D. Suciu, "PEEX: Extracting Probabilistic Events from Rfid Data," Proc. Int'l Conf. Data Eng. (ICDE), 2008.

[4] C. Beeri, T. Milo, and A. Pilberg, "Query-Based Monitoring of BPEL Business Processes," Proc. ACM SIGMOD Int'l Conf., 2007.

[5] Principles and Applications of Distributed Event-Based Systems, A. Hinze and A. Buchmann, eds. IGI Global, 2010.

[6] H. Chen and J. Dong, "Jtang Synergy: A Service Oriented Architecture for Enterprise Application Integration," Proc. 11th Int'l Conf. Computer Supported Cooperative Work in Design, Apr. 2007.

[7] A. Demers, M. Riedewald, V. Sharma, and W. White, "Cayuga a General Purpose Event Monitoring System," Proc. Third Biennial Conf. Innovative Data Systems Research, pp. 412-422, 2007.

Author Index:

Ms.M.Mythili - Received the Bachelor Degree in Computer Science and Engineering from Park College Of Engineering And Technology in 2010. She had completed Master of Engineering in Computer Science and Engineering at PGP college of Engineering and Technology in 2013. Presently she is working as an assistant professor in VSB Engineering College, CSE department from June 2013 onwards. She had attended so many Workshops , Conferences and FDP. Her research interests include Data Mining and Network security.

Mr.A.P.V.Raghavendra is persuing Ph.D. from the MANONMANIUM SUNDARANAR UNIVERSITY, Tirunelveli, India from 2013 onwards and Completed M.Tech(CSE) degree from Bharath University, Chennai, India in 2009. He is a Member in ISTE New Delhi, India, IAENG, Hong Kong. He has the experience in Teaching of 6.5 Years and in Industry 2 Years. Now he is currently working as an Assistant Professor in Computer Science and Engineering in V.S.B Engineering College, Karur, Tamil Nadu, and India. His research interests include Data Mining, Data Bases, Artificial Intelligence, Software Engineering Networking etc.,He had published 3 Annexture – II Scopus indexed and 27 international publications.