# Influencer events in episode rules: a way to impact the occurrence of events

Lina Fahed, Armelle Brun, Anne Boyer

Available online at www.sciencedirect.com

ScienceDirect

Procedia Computer Science 60 (2015) 527 – 536

Procedia Computer Science

19th International Conference on Knowledge Based and Intelligent Information and Engineering Systems

# Influencer events in episode rules: a way to impact the occurrence of events

Lina Fahed, Armelle Brun, Anne Boyer

*Lorraine University - LORIA Campus scientifique, BP 239 54506, Vandoeuvre-lès-Nancy Cedex, France*

**Abstract**

Episode rules are event patterns mined from a single event sequence. They are mainly used to predict the occurrence of events (the consequent of the rule), once the antecedent has occurred. The occurrence of the consequent of a rule may however be disturbed by the occurrence of another event in the sequence (that does not belong to the antecedent). We refer such an event to as an influencer event. To the best of our knowledge, the identification of such events in the context of episode rules has never been studied. However, identifying influencer events is of the highest importance as these events can be viewed as a way to act to impact the occurrence of events, here the consequent of rules. We propose to identify three types of influencer events: distance influencer events, confidence influencer events and disappearance events. To identify these influencer events, we propose to rely on the set of episode rules discovered by mining algorithms. The proposed approach for discovering influencer events is evaluated on an event sequence of social networks messages. Experiments measure the execution time efficiency according to the adopted episode rules mining algorithm. In addition, they show that some events do actually highly influence the consequent of some rules, that influencer events may not only influence several consequents, but also influence several characteristics of rules.



Peer-review under responsibility of KES International

*Keywords:* Data mining, Knowledge Discovery, Episode rules, Influencer events discovery.

## 1. Introduction

Episodes mining is an important research topic in temporal data mining, which refers to the discovery of temporal patterns made up of relatively close, partially ordered, items (or events) that appear often throughout a single data sequence or in a part of it[1]. Episode rules mining relies most of the time on the extraction of episodes (similarly to association rules mining, which relies on the extraction of itemsets).

Episode rules are generally used to predict events[2]. Let $R : ant \rightarrow p$ be an episode rule. The exploitation of $R$ allows to predict the occurrence of $p$ as yet as $ant$ has occurred. However, the occurrence of an event $e$, after $ant$ has occurred, may impact the predicted occurrence of the consequent $p$. For example, it may prevent the occurrence of $p$.

In this work, we are interested in the identification of such events, which will be referred to as *influencer events*. An event $e$ is considered as an influencer event associated with a rule $R : ant \rightarrow p$ if some characteristics (such as the

*E-mail address:* {lina.fahed, armelle.brun, anne.boyer}@loria.fr



Peer-review under responsibility of KES International

doi:10.1016/j.procs.2015.08.174

confidence or the support) of the rule $R' : ant, e \rightarrow p$ are significantly different from those of the rule $R$. Obviously, an event is influencer only in the specific context of the rule it impacts, as it may have no influence on other rules. To the best of our knowledge, influencer events have never been studied in the framework of episode rules. Their discovery constitutes not only a new challenge, but it is also of high importance, specifically in the context of event prediction.

To understand the importance of identifying influencer events, let us consider an example. Suppose the influencer events mining process has discovered an influencer event $e$ associated with a rule $R : ant \rightarrow p$. If the influence of $e$ is the increase of the confidence of $R$, this means that if $e$ occurs after $ant$, the probability that $p$ occurs is increased. If the goal is to ensure that $p$ will actually occur in the sequence, one thus has to force the occurrence of $e$ once $ant$ has occurred. At the opposite, if $e$ decreases the confidence of $R$, one has to guarantee that $e$ does not occur.

Many application domains can take advantage of the identification of influencer events. In the frame of social networks, companies (a bank for example) aim to increase or at least to maintain their e-reputation. So, they are highly interested in identifying events that positively impact their e-reputation. Suppose the following rule: *R: (complain interest rates in the bank), (threaten to leave the bank)* $\rightarrow$ *(bad e-reputation of the bank)*. An influencer event associated with this rule is *e: (marketing messages about interest rates)*, which decreases the probability of the occurrence of the consequent of $R$. This means that, if both events from the antecedent occur, the e-reputation of the bank will be negatively impacted. To save its e-reputation, the bank should spread marketing messages (event $e$). Notice that spreading marketing messages in another context may have the opposite effect. In the context of epidemic handling, the databases of the emergency department of several hospitals are employed. These databases contain information about the patients (admission time, symptoms, proposed treatment, zip code, etc.). An example of a rule mined can be *R:* $(symptom_A$*, humid* $area_1)$ , $(symptom_B$*, humid* $area_1) \rightarrow (symptom_A$*,* $area_2)$. This rule represents the propagation of $symptom_A$ to another area $area_2$ when $symptom_A$ and $symptom_B$ are present in a *humid* $area_1$. In this context, an influencer event $e$ should make the consequent disappear, *i.e.* stop the propagation of the epidemic to other areas. Concretely, the influencer event $e$ may be a treatment dedicated to these symptoms and this infected area. A treatment only dedicated to these symptoms may have the opposite effect as the characteristics of the area are not considered.

When predicting events, it is well known that the earlier a prediction can be made, the more useful it is[3]. More importantly, performing early prediction is primordial in some cases, as it allows to have a maximal time to react once the consequent event is predicted. To early predict events, traditional approaches rely on mining rules with an antecedent as small as possible. Mining such rules may either rely on mining a complete set of rules then keeping only rules with an antecedent as small as possible through a post-processing step, or rely on a dedicated algorithm[4,5].

As rules with an antecedent as small as possible are the most useful rules to perform early prediction, we consider them as a good basis to discover influencer events. We thus aim at discovering influencer events associated with such episode rules. However, the approach we propose is applicable whatever the characteristics of the episode rules are.

We are interested in the discovery of three types of influencer events: (i) distance influencer events, which influence the distance between the consequent and the antecedent of a rule, (ii) confidence influencer events, which influence the confidence of a rule, (iii) disappearance events, which influence the support of a rule.

The contributions of our paper are three-fold. **First**, we introduce a purely new concept: influencer events in episode rules. **Second**, we define three new measures related to influencer events: distance risk measure, confidence risk measure and disappearance measure, in order to discover respectively distance influencer events, confidence influencer events, and disappearance events. **Third**, we propose an influencer events discovery algorithm, and propose two ways for integrating this algorithm within an episode rules mining algorithm.

The rest of this paper is organized as follows: in section 2 we discuss the most relevant related works. Our contributions are detailed in section 3, followed by experimental results in section 4. We conclude in section 5.

## 2. Related works

We start by introducing few concepts related to episodes mining. Let $I_t$ be the set of items that occur at a timestamp $t$, referred to as an **event**. A **serial episode** is an ordered list of events that occurs throughout an event sequence $S$. Its support $supp(P)$ represents the number of occurrences of $P$. $P$ is said to be a **frequent episode** if $supp(P) \geq minsupp$ where $minsupp$ is a predefined threshold. Let $P$ and $Q$ be two episodes. An **episode rule** $R : P \rightarrow Q$ means that $Q$ appears after $P$. The **confidence** of $R$ is the probability to find $Q$ after $P$: $conf(P \rightarrow Q) = supp(P \cdot Q)/supp(P)$. $R$ is said to be confident if its confidence exceeds a predefined threshold $minconf$.

In the literature, the rules mining task is usually decomposed into two sub-problems: (i) mining frequent itemsets or episodes, (ii) constructing confident rules from those frequent itemsets or episodes[6]. A rule is constructed by considering the last items in the episode (or some items in the itemset) as the consequent of the rule, and the rest of the items as its antecedent. Since the second sub-problem is quite straightforward, most of the researches focus on the first one: frequent episodes or itemsets mining.

*Winepi* and *Minepi* are seminal episodes mining algorithms[1]. They start by forming episodes made up of one item, then iteratively extend them by merging items on their right side. When the order of items is total, the episode is serial, and when the order is not considered, the episode is parallel. This approach is still used by recent algorithms[7,8].

Influencer events, that can be considered as important events, have not been studied in the literature. However, the importance of events or patterns has been the focus of several works. It is represented by odds ratio, relative risk, utility or specific characteristics. Works related to odds ratio and risk patterns[9,10] have the drawback of requiring multiple transactional datasets, so have limited application domains. Most of works about high utility patterns mining represent the utility of a pattern as its importance or weight and focus on mining transactional databases[11,12]. The integration of utility into sequential patterns mining[13] and episodes mining[14] has been recently explored. In episodes mining, the recent tendency is to mine only significant episodes with predefined characteristics such as maximal episodes[15], closed episodes[16], or strict episodes[17]. All these works focus on entire patterns (episodes), not on a specific event or sub-pattern, and none of them study the evolution of these patterns. Thus, they do not fulfil our goal.

As previously mentioned, we propose to rely on a set of episode rules mining to discover influencer events. The rules mined by traditional algorithms, from our point of view, are not perfectly adequate for this purpose. Recall that they first mine episodes, then form rules by identifying the consequent. So, as the consequent is not known when an event is appended to an episode, the study of the influence of this event on the characteristics of the rule cannot be made. The only way to identify influencer events is to first mine all rules, then perform a post-processing, *i.e.* for each rule, study the influence of events, by comparing the characteristics of the rule without the event being a part of its antecedent and the characteristics of the rule with the event being a part of its antecedent, which is very costly. In addition, such an approach cannot identify events that significantly decrease the support of the rule. Indeed, the rules with a low support will not be mined by traditional algorithms. To identify such events, these algorithms have to also mine the large set of non frequent rules, which is not efficient. However, it is known that it is more efficient to incorporate the required characteristics within the mining algorithms, compared to running a post-processing step[18].

Recall that we aim at discovering influencer events associated with rules in which the antecedent is as small as possible, *i.e.* rules that allow to perform early prediction. When focusing on algorithms dedicated to the mining of such rules[4,5], we remark that they construct each rule by fixing the consequent at an early stage of the mining algorithm. Then, the antecedent of the rule (associated with the already known consequent) grows up iteratively. We propose to take advantage of this characteristic to discover influencer events. Indeed, as the consequent is known when appending an event, we can study the influence of this event on the characteristics of the rule, within the mining process. This way will have the advantage of not increasing the computational cost of the mining algorithm as no post-processing is required. The algorithm of Rahal[4] mines association rules with an antecedent as small as possible, whereas the algorithm of Fahed[5] mines episode rules with the same characteristic. As we are concerned with episode rules, we choose to take advantage of the last one.

## 3. Discovering influencer events

Following, we present the way we propose to discover influencer events, the three types of influencer events we focus on, as well as the three evaluation measures we propose.

### *3.1. Principle*

To identify influencer events, the algorithm we design (see Algorithm 1) relies on a set of episode rules *ER* resulting from an episode rules mining algorithm. Our algorithm requires that each of the episode rules is associated with three values: (i) the support of the rule, (ii) the confidence of the rule and (iii) the median distance between its antecedent and its consequent. The latter is not classically managed by episode rules mining algorithms, but can be easily computed.

Let $R : ant \rightarrow p$ be an episode rule, resulting from an episode rules mining algorithm. Let us consider also a second rule $R' : ant, e \rightarrow p$ that differs from $R$ in the presence of the event $e$. The principle of discovering influencer events is to compare the characteristics of each two rules $R$ and $R'$ (Algorithm 1 line 1) by asking the following questions: how much the characteristics of $R$ are different from those of $R'$? is the occurrence of the event $e$ in $R'$ changes significantly one/some of its characteristics comparing to those of $R$? If so, $e$ is denoted by an influencer event.

### *3.2. Definition of influencer events and their associated evaluation measures*

We present in details the three proposed types of influencer events and the measures we designed to identify them.

#### *3.2.1. Distance influencer events*

Given the rule $R : ant \rightarrow p$, and its associated median distance between its antecedent and its consequent $median(R)$. Let us consider also the rule $R' : ant, e \rightarrow p$ (it differs from $R$ in one event $e$ ). We study here the influence carried by $e$ on $R$, under the condition that both $R$ and $R'$ are frequent (their support exceed a predefined threshold). When comparing $R$ and $R'$, if the median distance between the antecedent and the consequent of $R'$ is significantly different from that of $R$, we can conclude that $e$ is a distance influencer event associated with $R$ (see Algorithm 1 lines 2 and 3). It is evaluated by the distance risk measure:

**Definition 1.** *The distance influence carried by an event e on a rule R is evaluated by the* **distance risk measure***, which represents the increase/decrease rate of the median value of the distance of the consequent to the antecedent in R relatively to that in R′, with the condition that both rules are frequent:*

$$Risk_{dist}(R, e) = \begin{cases} \frac{median(R,e) - median(R)}{median(R)} & : if\ R \wedge R'\ are\ frequent \\ 0 & : else \end{cases} \quad (1)$$

The distance risk measure may have a positive or a negative value, depending on if $e$ moves respectively away or closer the consequent. It varies from $-1$ to $\infty$. Let $\theta_{R_{dist}}$ be a threshold used to determine whether the event $e$ is a distance influencer event or not. When $\theta_{R_{dist}} > 0$, this means that influencer events are those with $Risk_{dist}(R, e) \geq \theta_{R_{dist}}$ (see Algorithm 1 lines 3 and 4). At the opposite, when $\theta_{R_{dist}} < 0$, we consider cases where $Risk_{dist}(R, e) \leq \theta_{R_{dist}}$, this means that the occurrence of the event $e$ brings the consequent closer to the antecedent.

#### *3.2.2. Confidence influencer events*

We are now interested in discovering the influence of events on the confidence of a rule by comparing the confidences of $R$ and $R'$, with the condition that both rules are frequent (Algorithm 1 lines 2 and 5), as follows:

**Definition 2.** *The influence carried by an event e on the confidence of a rule R is calculated by the* **confidence risk measure***, which represents the increase/decrease rate of the confidence of R (conf(R)) when appending e to its antecedent (conf(R, e)):*

$$Risk_{conf}(R, e) = \begin{cases} \frac{conf(R,e) - conf(R)}{conf(R)} & : if\ R \wedge R'\ are\ frequent \\ 0 & : else \end{cases} \quad (2)$$

We propose to use $\theta_{R_{conf}}$ as a threshold to determine whether $e$ is a confidence influencer event or not. When $\theta_{R_{conf}} > 0$, events where $Risk_{conf}(R, e) \geq \theta_{R_{conf}}$ are considered as confidence influencer events. When $\theta_{R_{conf}} < 0$, events where $Risk_{conf}(R, e) \leq \theta_{R_{conf}}$ (see Algorithm 1 line 5 and 6) are confidence influencer events. In this last case, the occurrence of the event $e$ decreases significantly the confidence of the rule and hence the probability of occurrence of the consequent (it breaks down the association between the antecedent and the consequent of the rule $R$ under study).

#### *3.2.3. Disappearance events*

Let us consider that the rule $R$ is frequent. The rule $R' : ant, e \rightarrow p$ may be not frequent. This case (Algorithm 1 line 7) represents the disappearance of the consequent $p$ after the new antecedent $ant, e$ occurs in the context of the rule. We are interested in discovering the reason of this rare/null occurrences of the rule. We identify two sub-cases. First, the support of $ant, e$ is low, which is the reason why support of $ant, e \rightarrow p$ is low. In this case, $e$ is considered

as totally out of context: there is no link between *ant* and $e$. So, $e$ is not a disappearance event related to $R$. Second, $e$ occurs frequently after the antecedent *ant* (the support of $ant, e$ is high), but the support of $ant, e \rightarrow p$ is low. In that case, $e$ can be considered as the cause of the disappearance of the consequent and is referred to as a disappearance event (see Algorithm 1 lines 8 and 9). Following, we formalize the conditions of identifying disappearance events:

**Definition 3.** *An event e is said to be a disappearance event of the rule* $R : ant \rightarrow p$ *if the three following conditions are met, represented by the* **disappearance measure***:*

$$Disapp(R,e): \begin{cases} supp(R) \geq minsupp & ; R \text{ is frequent} \\ conf(R) \geq minconf & ; R \text{ is confident} \\ supp(R') < minsupp & ; R' \text{ is not frequent} \\ supp(ant,e) \geq minsupp & ; (ant,e) \text{ is frequent} \end{cases} \quad (3)$$

We would like to mention that despite the use of thresholds to determine the most significant influencer events, an expert analysis phase is required to refine the final result, as the significance of events is application dependent. A pseudo code of the algorithm of discovering influencer events is presented in Algorithm 1.

**Algorithm 1:** Discovering influencer events

```
Data: set of episode rules ER, θ_{R_dist}, θ_{R_conf}
Result: influencer events
1 foreach (R : ant → p) ∈ ER ∧ (R' : ant, e → p) ∈ ER do
2     if R, R' are frequent then
3         if |Risk_dist(R, e)| ≥ |θ_{R_dist}| then
4             e is a distance influencer event
5         if |Risk_conf(R, e)| ≥ |θ_{R_conf}| then
6             e is a confidence influencer event
7     else
8         if R is frequent ∧ R is confident ∧ R' is not
          frequent ∧ (ant, e) is frequent then
9             e is a disappearance event
```

### *3.3. How to integrate influencer events discovery in an episode rules mining algorithm?*

We propose to discover influencer events by starting with the set of episode rules *ER* mined by an episode rules mining algorithm. Each episode rule has to be associated with three values: its support, its confidence and its median distance between the antecedent and the consequent. As previously mentioned, it is preferable that these rules have an antecedent as small as possible, to perform early prediction. We discuss here two propositions of how to integrate the influencer events discovery algorithm (Algorithm 1) according to the adopted episode rules mining algorithm and we discuss the expected performance of each proposition.

**Relying on a traditional algorithm:** Traditional episode rules mining algorithms usually start by forming episodes from left to right, then form rules by identifying the consequent. Discovering influencer events with a traditional episode rules mining algorithm can only be performed once the rules are formed. Thus, post-pocosseing is indispensable and can be performed by two steps: (i) Identifying, in the set *ER* resulting from the episode rules mining algorithm, rules with an antecedent as small as possible (necessary for early prediction). They will represent rules $R$ in Algorithm 1. (ii) Identifying all the couples of rules $R, R'$ so that $R'$ differs from $R$ in the presence of one additional event $e$. These two steps require a high computational time. In addition, a traditional algorithm results only frequent episode rules. So, rules with a low support are not mined. Thus, disappearance events (section 3.2.3) cannot be discovered as rules $R'$ are not in the set *ER*. To identify such events, traditional algorithms can be modified in order to mine also the large set of non frequent rules, which is extremely costly.

**Relying on a dedicated algorithm (that mines rules with an antecedent as small as possible):** In the previous section, we have shown that algorithms dedicated for mining rules with an antecedent as small as possible[4,5], construct each rule by fixing the consequent at an early stage in the mining process. The resulting rules represent the rules $R$ on which rely the Algorithm 1 to discover influencer events (recall that traditional algorithms have to identify them during a costly post-processing step). We propose to take advantage of the algorithm proposed by Fahed[5] that mines episode rules. It works as follows: first, it fixes the prefix (the first event) of the rule being mined. Second, it determines the consequent. Third, it iteratively completes the antecedent by appending events. Completing the antecedent stops once the rule is frequent and confident, which represents a rule with an antecedent as small as possible.

A way to take advantage of this algorithm to discover influencer events is by running the algorithm one step further: once the rule is frequent and confident (which represent a rule of type $R$), we propose to append one other event $e$ to the antecedent, to form a rule $R'$. It results in two advantages: (i) Rules $R'$ are formed immediately after mining rules $R$.

So, the couples $R, R'$ are identified during the mining process, and the algorithm of discovering influencer events can be applied directly. Identifying $R'$ does not require a significant additional computational cost, contrary to traditional algorithms where rules $R$ and $R'$ should be searched in the entire set of rules during a post-processing. (ii) The rule $R'$ may be not frequent, so the influence carried by the event $e$ on the rule $R'$ can be studied, and disappearance events can be discovered with a small increase in the computational time. Recall this was not possible with traditional algorithms, except by decreasing the minimal support threshold, which dramatically increases the computational time.

We have proposed to integrate the influencer events discovery algorithm in a traditional episode rules mining algorithm and in an algorithm dedicated to the mining of rules with an antecedent as small as possible. It is important to precise that whatever the episode rules mining algorithm used, the resulting set of influencer events is the same, but the computational time importantly varies depending on the adopted algorithm.

## 4. Experimental results

In this section, we first study the execution time of the influencer events discovery according to the adopted episode rules mining algorithms. Second, we focus on the characteristics of the influencer events discovered and the rules they are associated with. We would like to mention that, to the best of our knowledge, no work has been dedicated to the discovery of influencer events in episode rules, so this work is not compared with any other algorithm. In this work, the dataset used is made up of 27, 612 messages extracted from blogs about finance and banks. Messages are annotated using the $Temis$[1] software and are represented by their annotations. Messages are annotated with 4.8 items on average. There are about 4, 000 distinct items, with an average frequency of 88.5. Recall that for discovering influencer events, we require a set of episode rules mined by an episode rules mining algorithms. Therefore, we propose to rely on the traditional $Minepi$[1] algorithm for mining episode rules, and on the dedicated algorithm proposed by Fahed[5] for mining rules with an antecedent as small as possible, which are both run with $minsupp = 20$, $minconf = 0.4$, $w = 100$ (maximal search window size).

### 4.1. Execution time

Table 1 presents in details a run-time comparison of integrating the influencer events discovery algorithm in the traditional $Minepi$[1] algorithm and integrating it in the algorithm proposed by Fahed[5], for $\theta_{R_{dist}} = |0.2|$ and $\theta_{R_{conf}} = |0.7|$.

As expected, integrating the influencer events discovery algorithm in a dedicated algorithm is more efficient, as it is more than 8 times faster than when adopting a traditional algorithm (Table 1 last column). When relying on $Minepi$, the time required for the first post-processing step (identifying rules with an antecedent as small as possible (see section 3.3)) is relatively high, contrary to the algorithm proposed by Fahed, where these rules are basically mined. Moreover, for $Minepi$, the second post-processing step (identifying the couples $R, R'$) requires a relatively small running time, compared to the dedicated algorithm, where we proposed to mine the rules $R'$ (including non frequent rules) by performing one step further. As mentioned before, once the couples $R, R'$ are identified, the algorithm for mining influencer events requires a small computational time which is the same whatever the algorithm of mining rules is (Table 1 column 4). We would like to mention also that discovering disappearance events by relying on $Minepi$ is extremely costly due to the low $minsupp$ required, so it is not presented in Table 1.

Table 1: Run-time comparaison (in seconds) of integrating the influencer events discovery in two episode rules mining algorithms

| | Identifying $R$ (*first post-processing*) | Identifying couples $R, R'$ (*second post-processing*) | Discovering distance & confidence influencer events | Discovering disappearance events | Total additional time |
|---|---|---|---|---|---|
| $Minepi$[1] | 5320 s | 160 s | 30 s | - - | 5510 s |
| | Identifying $R$ (*basically mined*) | Mining $R'$ (*mined by 1 step further*) | Discovering distance & confidence influencer events | Discovering disappearance events | Total additional time |
| $Fahed$[5] | 0 s | 575 s | 30 s | 50 s | 655 s |

[1] http://www.temis.com

### 4.2. Characteristics of the influencer events

We study here the characteristics of the discovered influencer events obtained when relying on the algorithm of Fahed[5]. 76$k$ rules are formed by the algorithm of Fahed[5], among which about 44$k$ rules have an antecedent of length 1, and 32$k$ rules have an antecedent of length 2. These rules are considered as rules of type $R$ (see section 3.3) and constitute the starting point of the discovery of influencer events. In order to discover the influencer events, the dedicated algorithm runs one step further (see section 3.3) to form rules of type $R'$. Two cases are possible: the resulting rule $R'$ is always frequent which is used to study the distance and the confidence influence of the added event, or the rule $R'$ is no more frequent which will be used to study the disappearance influence of the added event, as explained later on. Appending one event to the antecedent of the 76$k$ rules $R$ results in about 58$k$ frequent rules $R'$, most of which have an antecedent of length 2. The small number of rules is explained by the fact that appending an event often yields to a non frequent rule (studied in section 4.3).

#### 4.2.1. Distance influencer events

In this section, we focus on distance influencer events, according to the distance risk threshold $min_{R_{dist}}$ (see definition 1). We study five elements (presented in Figures 1 and 2): (i) the influencing cases $R'$: the cases where a rule $R$ is influenced by one event, (ii) the influenced rules $R$: a rule can be influenced by several events, which represents several influencing cases, (iv) the influencer events: an event can influence several rules, and (v) influenced consequents: a consequent can be influenced in several rules.

Figure 1 shows the distribution of the number of influencing cases ($\#R'$) according to the distance risk ($Risk_{dist}$) carried by the appended event. The bar associated with $Risk_{dist} \in [-0.4, -0.2[$ represents the number of rules $R'$ where an event makes the consequent get closer of 20% to 40%, which represents a significant influence. We can see that the risk varies from −0.4 to 2.8 and that most of the rules $R'$ are associated with $Risk_{dist} \in [0.2, 0.8[$. The number of rules $R'$ with $Risk_{dist} < 0$ is quite low, which was expected. Indeed, a consequent that gets closer means that the number of occurrences of the rule $R$ with a far consequent and that do not contain the appended event is lower than the number of occurrences of the rule $R'$ with a close consequent and that do contain the appended even, which is unlikely.

To study the cases where an influencer event impacts *significantly* the distance of the consequent, a distance risk threshold has to be fixed. Here, we choose $\theta_{R_{dist}} = -0.2$ and $\theta_{R_{dist}} = 2$, presented in Figure 2. The 132 rules $R'$ when $\theta_{R_{dist}} = -0.2$ (the blue curve in Figure 2) correspond to 118 distinct influenced rules $R$, 40 distinct influencer events and 61 distinct influenced consequents. Notice that this threshold should be fixed by experts of the application field, as for each threshold value a different applicative analysis is carried out. Figure 2(A) shows that most of the 118 rules $R$ are influenced only once, so by only one event. Figure 2(B), represents how many rules the events influence. It shows that 14 events (25%) influence only one rule, but an event can influence till 12 rules. Notice that the events that influence many rules have to be used with caution, as they may have a high influence on the occurrence of several consequents. Figure 2(C), that represents the number of influenced rules a consequent belongs to, shows that more than 50% of the consequents belong to several rules, which means that they may be influenced by several events. These consequents are sensitive, weak and move easily close to the prefix of the root rule.

Figure 2 (the red curve), presents also the 358 cases of rules $R'$ when $\theta_{R_{dist}} = 2$. They represent 158 distinct rules $R$, 82 distinct influencer events and 50 distinct influenced consequents. Observations made from Figures 2(A) and 2(B) where $\theta_{R_{dist}} = -0.2$ are in accordance with those made here with $\theta_{R_{dist}} = 2$. Figure 2(C)) shows that a significant number of consequents are consequents of more than 7 rules, thus may be influenced in more than 7 contexts. However, at the opposite of $\theta_{R_{dist}} = -0.2$, where most of the consequents belong to less than 7 rules.

From the set of distance influencer events we discovered, some of them significantly influence the distance of the consequent of several rules than others. We consider these events as more important than others, due to their large impact. Similarly, some consequents and rules are easily influenced by many influencer events. They have to be considered carefully as they may be often impacted. However, in the current dataset, several consequents are never influenced, such as the *(economic crisis)* event, which is evident in this context. Given the following rule *R: (subscription)* → *(buying a product)*, $Risk_{dist}(R, (customer\ waiting)) = 2.5$. This means that the influencer event *(customer waiting)* moves the consequent 2.5 times away. A deep analysis shows that the absolute difference of the distance is equal to 44, which means that when the event *(customer waiting)* occurs, the consequent moves 44 timestamps farther from the prefix, which is very high. In a real application, this would mean that if the company makes its customer wait, he/she will buy a product significantly later.

*4.2.2. Confidence influencer events*

Figure 3 shows the number of confidence influencing cases ($\#R'$), according to the confidence risk. The bar associated with $Risk_{conf} \in [0.2, 0.4[$ represents the number of rules $R'$ where the confidence of a rule is increased by 20% to 40%. The distribution of the influencing cases ($\#R'$) is different from that of the previous section: the number of rules $R'$ when $\theta_{R_{conf}} >= 0$ is nearly equal that when $\theta_{R_{conf}} < 0$ and most of the rules $R'$ have $Risk_{conf} \in [-0.2, 0.2[$. We focus again on influencer events that impact *significantly* the confidence of the rule. Two values of confidence risk threshold are fixed (Figure 4): $\theta_{R_{conf}} = -0.7$ and $\theta_{R_{conf}} = 0.7$.

Figure 4 (the blue curve) shows in details the 168 confidence influencing cases ($\#R'$) for $\theta_{R_{conf}} = -0.7$, which correspond to 81 distinct influenced rules $R$, 18 distinct influencer events and 24 distinct influenced consequents. In Figure 4(A), among the 81 influenced rules $R$, 45(55%) are influenced only once, others may be influenced till 8 times. Figure 4(B)) shows that the 18 influencer events influence equally from 1 to more than 30 rules. Similarly, Figure 4(C) shows that influenced consequents belong to 1 up to 25 rules, being almost equally distributed.

Figure 4 (the red curve) presents in details the 132 confidence influencing cases ($\#R'$) for $\theta_{R_{conf}} = 0.7$ . They correspond to 69 distinct influenced rules $R$, 88 distinct influencer events and 42 distinct influenced consequents. Conclusions drawn from Figure 4(A), are in accordance with what was observed with $\theta_{R_{conf}} = -0.7$. Figure 4(B) shows that among the 88 influencer events, 58 events (65%) influence only one rule and none of them influence more than 4 different rules. This is contrary to what was observed with $\theta_{R_{conf}} = -0.7$. Notice also that the number of influencer events is 4 times larger with $\theta_{R_{conf}} = 0.7$, than with $\theta_{R_{conf}} = -0.7$, whereas the number of rules was comparable. This may mean that events that decrease the confidence are more important as not only there are less such events, but they also influence more rules. In Figure 4(C)), we remark that the consequents belong to maximum 13 rules and most of them are consequents of only one rule. This distribution is different from the one when $\theta_{R_{conf}} = -0.7$. Moreover, the number of influenced consequents is here almost twice larger than when $\theta_{R_{conf}} = -0.7$, whereas the number of rules is comparable. Once again, this may mean that the consequents that belong to rules with a significant decrease of the confidence are more important as they are not only fewer, but also they belong to more rules.

We can conclude that, in this dataset, many consequents of rules are not stable, they can easily be changed by events appended to the antecedent of the rule. An example of a confidence influencer event is presented here. Let $R$ be the rule: *(savings problem)→ (contact concurrent)*. For the influencer event *(no loan at rate zero)*, $Risk_{conf}$*(R, (no loan at rate zero))*$= 0.75$. This event significantly increases the probability that the customer contacts a concurrent company. A detailed study shows that $Risk_{dist}$*(R, (no loan at rate zero))*$= -0.18$. This means that this event has a double influence: it not also increases the probability of the occurrence of the consequent but also it brings it closer.

*4.3. Disappearance events*

We focus now on the cases of disappearance of events (see definition 3) as shown in Figure 5. $1,181$ rules $R'$ are discovered, in which the event appended to the antecedent is the cause of the low support of the rule. They correspond to 282 distinct disappeared rules $R$, 65 distinct influencer disappearance events and 33 distinct influenced consequents. Figure 5 (A) show that 66% of the disappeared rules are influenced by at least two events, and about 16% of them are influenced more than 10 times. Most of the disappearance events influence several rules, some of them even influence more than 50 rules (Figure 5(B)). These events have a great influence on the support of the rules, making them disappear. Moreover, among the 33 disappeared consequents, some of them are influenced more than 100 times, meaning that they disappear in many rules (Figure 5(C)). A detailed study shows that this disappearance is due to few events and concerns only few consequents.

An example of a disappeared rule is shown here: Let $R$ be the rule: *(bad price) → (buying a product)*. $supp(R) = 236$. When appending the event *(customer waiting)*, the rule $R'$*: (bad price), (customer waiting) → (buying a product)* has a support equal to 1, whereas the support of its antecedent is equal to 97. So, the event *(customer waiting)* is a support influencer event as it causes the disappearance of the rule. In a real application, companies aim at increasing the number of their customer purchases. If a customer is not satisfied by a price and if the *(customer waiting)* event occurs, this will cause the disappearance of the buying event. So, we recommend the company not putting the customer in a waiting situation, otherwise he/she will not buy the article.

As mentioned before, we are interested in discovering consequents that are influenced several times and events that influence several rules. In Table 2, we present a list of the most interesting influenced consequents and influencer events related to the three cases of influence: distance, confidence and disappearance.

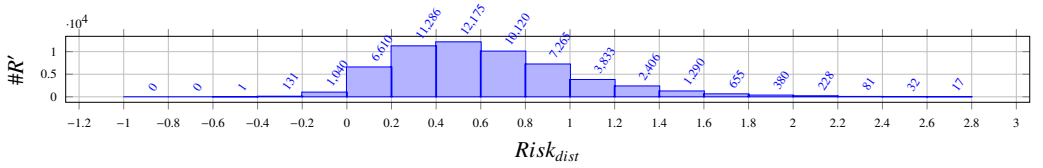


Fig. 1: Distance influencing cases (#$R'$) according to distance risk intervals.

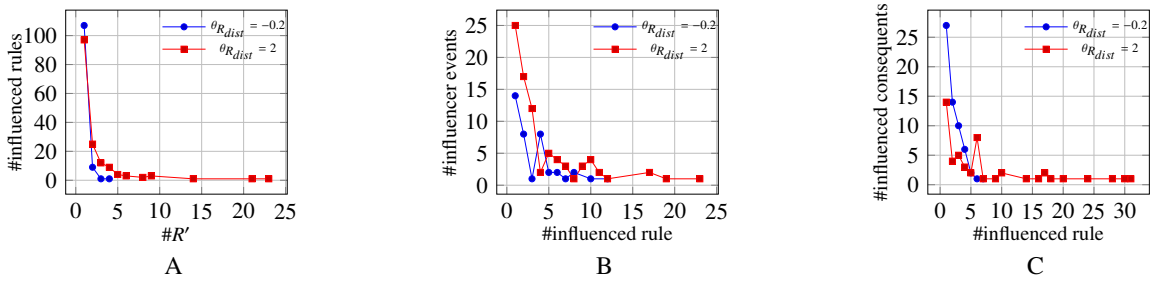


Fig. 2: Distance influenced rules, influencer and influenced events.

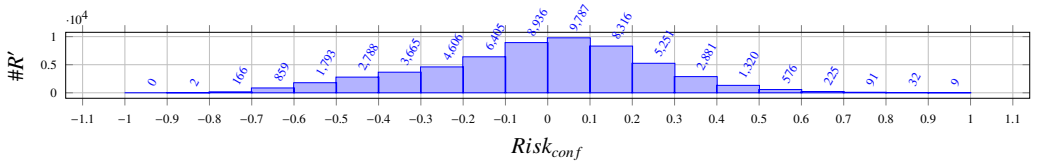


Fig. 3: Confidence influencing cases (#$R'$) according to confidence risk intervals.

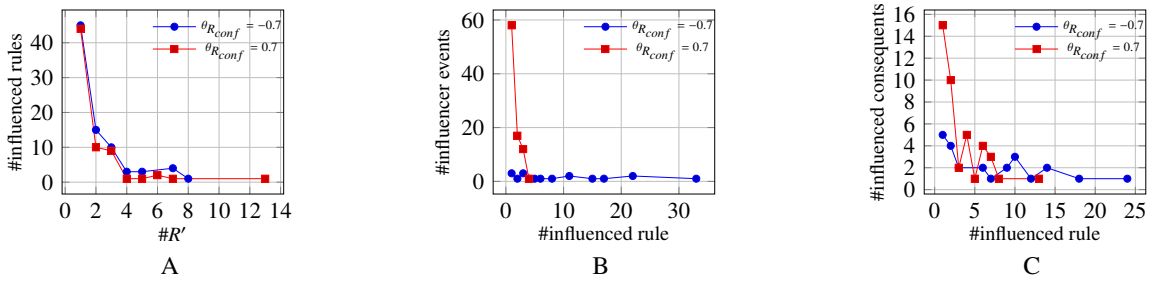


Fig. 4: Confidence influenced rule, influencer and influenced events.

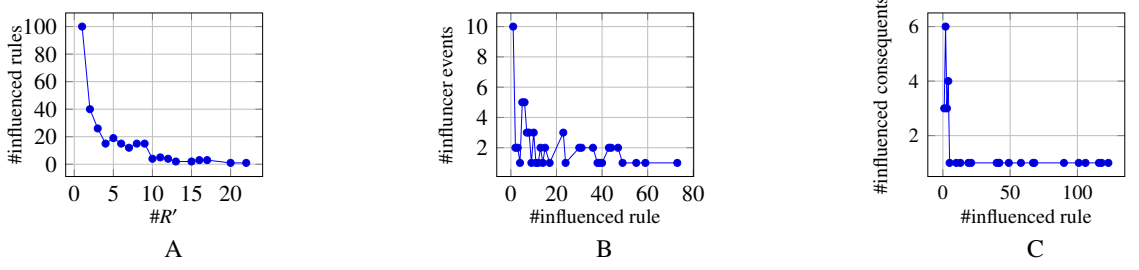


Fig. 5: Disappeared rules, disappearance events and influenced consequents.

Table 2: A list of interesting influenced consequents and influencer events

| | Distance influence: $\theta_{R_{dist}} = 0.2$ | Confidence influence: $\theta_{R_{conf}} = 0.7$ | Disappearance influence |
|---|---|---|---|
| Influenced consequent | open a saving account<br>buying a product<br>ownership saving scheme | cheque problem<br>contacting concurrent<br>buying a product | agency problem<br>no selling<br>subscription |
| Influencer event | not satisfactory cost<br>customer waiting<br>not satisfactory cost | high cost<br>customer waiting<br>ownership saving scheme | bad proposition<br>high price<br>bad contract conditions |

## 5. Conclusion and perspectives

In this paper, we have proposed an algorithm that discovers influencer events in episode rules. The mining of such events is original, and is useful in many applications. To perform this mining, we rely on a set of episode rules. Three types of influencer events are proposed: distance, confidence and disappearance influencer events, through three new risk measures. The algorithm, which was evaluated on an event sequence of social networks messages, discovered interesting and meaningful influencer events. These events can impact the distance of the consequence, the confidence of the rule, or even discard the rule.

In this work, influencer events of size only one are studied. However, in some cases, an event may not be an influencer by itself, but when it occurs with another event, the resulting pair may have a significant influence. In a future work, we aim at discovering pairs or higher order of influencer events. Our long term goal is to discover another type of influencer events, which will be referred to as *weak signals*. They represent events that cause the mutation of episode rules over time, and not only change their characteristics.

## References


1. Mannila, H., Toivonen, H., Verkamo, A.I.. Discovery of frequent episodes in event sequences. *Data Mining and Knowledge Discovery* 1997;**1**(3):259–289.
2. Cho, C.W., Zheng, Y., Chen, A.L.. Continuously matching episode rules for predicting future events over event streams. In: *Advances in Data and Web Management*. Springer; 2007, p. 884–891.
3. He, G., Duan, Y., Qian, T., Chen, X.. Early prediction on imbalanced multivariate time series. In: *Proceedings of the 22nd ACM international conference on Conference on information & knowledge management*. ACM; 2013, p. 1889–1892.
4. Rahal, I., Ren, D., Wu, W., Perrizo, W.. Mining confident minimal rules with fixed-consequents. In: *16th IEEE ICTAI 2004*. 2004, p. 6–13.
5. Fahed, L., Brun, A., Boyer, A.. Episode rules mining algorithm for distant event prediction. In: *KDIR International Conference on Knowledge Discovery and Information Retrieval*. 2014, p. 5–13.
6. Agrawal, R., Imieliński, T., Swami, A.. Mining association rules between sets of items in large databases. In: *ACM SIGMOD*. 1993, p. 207–216.
7. Laxman, S., Sastry, P., Unnikrishnan, K.. A fast algorithm for finding frequent episodes in event streams. In: *13th ACM SIGKDD*. 2007, p. 410–419.
8. Huang, K.Y., Chang, C.H.. Efficient mining of frequent episodes from complex sequences. *Information Systems* 2008;**33**(1):96–114.
9. Li, J., Fu, A.W.c., Fahey, P.. Efficient discovery of risk patterns in medical data. *Artificial intelligence in Medicine* 2009;**45**(1):77–89.
10. Li, J., Fu, A.W.c., He, H., Chen, J., Jin, H., McAullay, D., et al. Mining risk patterns in medical data. In: *11th ACM SIGKDD*. 2005, p. 770–775.
11. Tseng, V., Wu, C., Shie, B., Yu, P.. Up-growth: an efficient algorithm for high utility itemset mining. In: *16th ACM SIGKDD*. 2010, .
12. Liu, M., Qu, J.. Mining high utility itemsets without candidate generation. In: *21st ACM international conference on Information and knowledge management*. ACM; 2012, p. 55–64.
13. Ahmed, C.F., Tanbeer, S.K., Jeong, B.S.. Mining high utility web access sequences in dynamic web log data. In: *SNPD*. IEEE; 2010, p. 76–81.
14. Wu, C., Lin, Y.F., Yu, P.S., Tseng, V.S.. Mining high utility episodes in complex event sequences. In: *19th ACM SIGKDD*. 2013, p. 536–544.
15. Iwanuma, K., Ishihara, R., Takano, Y., Nabeshima, H.. Extracting frequent subsequences from a single long data sequence a novel anti-monotonic measure and a simple on-line algorithm. In: *5th IEEE International Conference on Data Mining*. 2005, p. 8–pp.
16. Zhou, W., Liu, H., Cheng, H.. Mining closed episodes from event sequences efficiently. In: *Advances in Knowledge Discovery and Data Mining*. Springer; 2010, p. 310–318.
17. Tatti, N., Cule, B.. Mining closed strict episodes. *Data Mining and Knowledge Discovery* 2012;**25**(1):34–66.
18. Srikant, R., Vu, Q., Agrawal, R.. Mining association rules with item constraints. In: *KDD*; vol. 97. 1997, p. 67–73.