STAR: Summarizing Timed Association Rules

CRISTIAN MOLINARO, University of Calabria, Italy

CHIARA PULICE, ANJA SUBASIC, ABIGAIL BARTOLOME, and V. S. SUBRAHMANIAN,
Dartmouth College, USA

Timed association rules (TARs) generalize classical association rules (ARs) so that we can express temporal dependencies of the form “If X is true at time t , then Y will likely be true at time $(t + \tau)$.” As with ARs, solving the TAR mining problem can generate huge numbers of rules. We show that methods to summarize ARs cannot work directly with TARs, and we develop two notions—*strong* and *weak* summaries—to summarize a set of TARs. We show that the problems of finding strong/weak summaries are NP-hard, and we provide polynomial-time approximation algorithms. We show experimentally that the coverage provided by our summarization methods is very high. Both technical measures based on coverage and human experiments on six World Bank datasets using 100 subjects from Mechanical Turk and a separate experiment with terrorism experts on a terrorism dataset show that while both summarization methods perform well, weak summaries are preferred, despite their taking more time to compute than strong summaries.

CCS Concepts: • **Information systems** → **Association rules**; *Summarization*; *Temporal data*;

Additional Key Words and Phrases: Timed association rules, summarization

ACM Reference format:

Cristian Molinaro, Chiara Pulice, Anja Subasic, Abigail Bartolome, and V. S. Subrahmanian. 2020. STAR: Summarizing Timed Association Rules. *ACM/IMS Trans. Data Sci.* 2, 1, Article 6 (December 2020), 36 pages.
<https://doi.org/10.1145/3419107>

1 INTRODUCTION

There are many applications where we want to consider Timed Association Rules (or TARs) that have the form “if X holds at time t , then Y likely holds at time $(t + \tau)$ ” for some $\tau \geq 0$. TARs enhance classical association rules (ARs) [3] with time. Such rules have found applications in predicting movement of stocks in the future based on conditions that hold currently [10], predicting future failures in manufacturing processes based on conditions that hold currently [24], and more. For instance, Reference [28] shows that such rules occur naturally in reasoning about terrorist groups in general. They derive over 15,000 predictive rules of the above type to describe different types of attacks carried out by the Lashkar-e-Taiba (LeT) terror group, which can be used to predict if LeT carries out attacks and when. Clearly, this is an overwhelming number, and it is hard for analysts

We are grateful to ONR grant N00014-16-1-2918 for partial support of this work.

Authors’ addresses: C. Molinaro, University of Calabria, via P. Bucci 42C, 87036, Rende (CS), Italy; email: cmolinaro@dimes.unical.it; C. Pulice, A. Subasic, A. Bartolome, and V. S. Subrahmanian, Sudikoff Lab, Dartmouth College, Hanover, NH 03755, USA; emails: {chiara.pulice, anja.subasic.gr, abigail.bartolome.gr, vs}@dartmouth.edu.

Permission to make digital or hard copies of all or part of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. Copyrights for components of this work owned by others than ACM must be honored. Abstracting with credit is permitted. To copy otherwise, or republish, to post on servers or to redistribute to lists, requires prior specific permission and/or a fee. Request permissions from permissions@acm.org.

© 2020 Association for Computing Machinery.

2577-3224/2020/12-ART6 \$15.00

<https://doi.org/10.1145/3419107>

to understand so many rules, even if they have predictive power. Understanding these rules is key for them, as it is the basis for receiving resources to implement better security methods. We develop two types of summaries in this article that are capable of summarizing such rule sets. For instance, for a set of 142 rules taken from the aforementioned ones, the “strong summary” consists of just 14 rules. The “weak summary” consists of just 4 rules—and in fact, we introduce a notion of coverage that shows that two of these 4 weak summary rules cover 97% of the original 142 rules. These two weak summary TARs informally state that “if LeT did not splinter in a given month m , they carry out one to three attacks in month $m + 2$ ” and “if 2 or less LeT commanders died in a month m , then LeT carried out one to three attacks on civilians in month $m + 2$.” In contrast, the original set of 142 rules had far more complex pre-conditions and were far more numerous to boot.

While ARs can only express patterns occurring within *individual* itemsets, TARs can also express patterns occurring *across multiple itemsets in time*. For instance, the two aforementioned sample TARs express phenomena occurring both in months m and $m + 2$. Thus, TARs $\tau : X \rightarrow Y$ express patterns like: If X occurs in an itemset, then Y is likely to occur in an itemset after τ time points. TARs (strictly) generalize ARs—when $\tau = 0$, they express ARs. It is well known in the context of ARs that a huge number of rules can be generated when reasonable support and confidence constraints are considered [17, 19, 20, 22], and this has led to the development of several excellent papers to summarize *classical* association rules [1, 2, 5, 6, 11, 13, 17, 19, 20, 22, 27, 31, 32] to manage the deluge of rules generated by AR mining algorithms. However, though there has been work on combining time with association rules [4, 7–9, 15, 16, 18, 21, 29, 30], *there has been no work to date on summarizing temporal association rules*.

There are significant differences between ARs and TARs as illustrated in Example 1 below. One main difference is the following: To see if a AR occurs in the data, we look at each itemset individually (each AR is about a single itemset), while to see if a *timed* association rule occurs, we need to look at different itemsets at different time points (each TAR is a statement concerning two itemsets). This affects the relationships between rules in terms of support, confidence, and other metrics, which are used to determine whether a rule represents another, and in turn makes approaches to summarize ARs not applicable to TARs. The following example illustrates this difference.

Example 1. Consider the ARs $r_1 = AC \rightarrow B$, $r_2 = A \rightarrow BC$, and $r_3 = AB \rightarrow C$. They always (i.e., for every database) have the same support, and the confidence of r_2 is always lower than or equal to that of the other two rules.¹ Since both support and confidence of r_2 are always lower than or equal to the ones of r_1 and r_3 , when r_2 meets minimum support and confidence thresholds, so do r_1 and r_3 . Thus, the latter are considered redundant and r_2 is considered a representative association rule—e.g., see Reference [2].

However, this behavior does not hold anymore in the temporal setting. As an example, consider the following temporal database, where each itemset is associated with a time point:

$$\begin{aligned} &(0, AB), \\ &(1, ABC), \\ &(2, C). \end{aligned}$$

For instance, the meaning of $(0, AB)$ is that A and B occur at time point 0.

¹We recall that the *support* of an AR $X \rightarrow Y$ is the fraction of itemsets in the database that contain both X and Y , while the *confidence* is the fraction of itemsets in the database containing X that also contain Y .

Consider now the following timed association rules:

$$\begin{aligned} r'_1 &= 1 : AC \rightarrow B, \\ r'_2 &= 1 : A \rightarrow BC, \\ r'_3 &= 1 : AB \rightarrow C. \end{aligned}$$

The intuitive meaning of r'_1 is as follows: If A and C occur in an itemset, then B is likely to occur in an itemset after 1 time point. The support of r'_1 is the fraction of times r'_1 occurs, which is the case when A and C occur in an itemset and B occurs in the subsequent itemset. The confidence of r'_1 is the fraction of times itemsets containing A and C are followed by an itemset containing B after one time point. We formally define such metrics in Section 2. TARs r'_2 and r'_3 can be similarly read.

The support and confidence of r'_1 , r'_2 , and r'_3 are thus as follows:

$$\begin{aligned} \text{Supp}(r'_1) &= 0, & \text{Conf}(r'_1) &= 0, \\ \text{Supp}(r'_2) &= 1/3, & \text{Conf}(r'_2) &= 1/2, \\ \text{Supp}(r'_3) &= 2/3, & \text{Conf}(r'_3) &= 1. \end{aligned}$$

The TARs above differ significantly from their non-temporal counterparts in many respects: They no longer have the same support, and the confidence of r'_2 is no longer lower than or equal to the one of the other two rules—indeed, both its support and confidence are in between those of the other two timed association rules. Thus, the argument that leads to the claim that r_2 represents r_1 and r_3 cannot be applied in a temporal setting.

We also point out that if TARs are expressed via classical ARs, with the aim of summarizing the latter with existing methods, there is a significant loss of information. As an example, consider the TAR $1 : X \rightarrow Y$. This TAR says that if X holds at an *arbitrary* time t , then Y likely holds at time $t + 1$. Thus, the TAR compactly expresses a statement that applies to several time points—indeed, it applies to every time point t . One might try to express the same statement via a set consisting of the following classical ARs:

$$\begin{aligned} X \text{ at time } = 0 &\rightarrow Y \text{ at time } = 1, \\ X \text{ at time } = 1 &\rightarrow Y \text{ at time } = 2, \\ X \text{ at time } = 2 &\rightarrow Y \text{ at time } = 3, \\ &\vdots \\ X \text{ at time } = \tau_{\max} - 1 &\rightarrow Y \text{ at time } = \tau_{\max}. \end{aligned}$$

However, from a single TAR we have generated a huge number of classical ARs, and by summarizing the latter with existing methods we cannot even recover the original TAR exactly. For instance, several approaches for summarizing standard ARs would not be able to summarize the rules above, because they require that the consequents of the association rules being summarized should be the same (e.g., References [14, 19]).

In addition to the differences discussed so far, techniques for dealing with ARs cannot explicitly handle time, while we treat time as a first-class citizen. For instance, the three TARs $1 : X \rightarrow Y$, $2 : X \rightarrow Y$, and $3 : X \rightarrow Y$, which differ only w.r.t. the time delay, can be summarized by $2 : X \rightarrow Y$, that is, by taking the middle point in the range $[1, 3]$ as a summarization.

Thus, we need to devise new methods to summarize TARs tailored for the temporal setting, which is the main goal of this article.

Contributions. We make the following contributions.

- In the non-temporal setting, it is widely recognized that mining ARs can yield an overwhelming set of rules with high redundancy, which are hard to analyze/interpret. The same happens in the temporal setting. We address this issue by succinctly summarizing a set of

TARs. Specifically, we define two problems. The *strong summary set problem* asks for a representative subset of a given set of TARs. In the *weak summary set problem*, the summary set might also contain TARs that were not in the original set, provided that they are representative enough and “close enough” to the original TARs.

- We show that the strong and weak summary set problems are both NP-hard. We show that NP-hardness holds even when the size of TARs is bounded by a constant.
- In light of the intractability results, we develop two polynomial-time greedy algorithms, STAR-S and STAR-W, to approximately solve the strong and weak summary set problems, respectively. We show that the greedy algorithm to solve the strong summary set problem has approximation guarantees.
- We first experimentally validate our approach over six datasets from the World Bank using 100 Amazon Mechanical Turk workers. However, we also evaluate our approach using 4 terrorism experts on a terrorism dataset from a book on Lashkar-e-Taiba [28]. Experimental results show that real-world timed association rules have high redundancy, and our algorithms are able to summarize them in a very effective manner. In addition, we show that STAR-S is very scalable by demonstrating its runtime on a dataset with just under 1M transactions.

Organization. Section 2 introduces TARs. The strong and weak summary set problems are defined in Section 3, their complexity is analyzed in Section 4, and polynomial-time approximation algorithms to solve such problems are proposed in Section 5. We report on an experimental evaluation in Section 6 including both a technical evaluation and a survey of humans. Section 7 shows samples of the summaries generated during our experimental evaluation. Related work is discussed in Section 8. Conclusions and directions for future work are discussed in Section 9.

2 BACKGROUND ON TIMED ASSOCIATION RULES

We define timed association rules in this section. We assume the existence of a finite set P of *items* and a finite set $Time = \{0, \dots, \tau_{max}\}$ of time points in which we are interested. A user can choose both the granularity of $Time$ and τ_{max} in an application-specific way. In the rest of the article, we will sometimes denote a set of items $\{X_1, X_2, \dots, X_n\}$ simply as $X_1X_2 \dots X_n$.

An *itemset* is a subset of P . A *temporal transaction* is a pair $\mathbf{t} = (\tau, T)$, where $\tau \in Time$ and T is an itemset. The intuitive meaning is that all items in T hold true at time point τ . A *temporal database* is a finite set of temporal transactions. Notice that it might be the case that a temporal database does not include a temporal transaction for one or more time points. The absence of a temporal transaction for a time point τ simply means that no item occurred at τ . Indeed, this can be equivalently modeled by including the temporal transaction $(\tau, \emptyset)$ in the temporal database, as the meaning of $(\tau, \emptyset)$ is that no item holds true at time point τ —however, this is not required by our model, which allows a temporal database not to have temporal transactions for one or more time points. The choice on how to deal with such a scenario (that is, either by explicitly storing $(\tau, \emptyset)$ or by storing no temporal transaction for τ) is up to the algorithm used to mine TARs, which is an issue we do not address in this article, as our goal is to start with a set of (already mined) TARs and summarize them.

Of course, other ways of modeling temporal data are possible. For instance, Reference [26] addressed the problem of mining TARs to discover the toxicity progression of cancer treatments and used a different representation of temporal data. Specifically, given a set C of *codes* (i.e., diagnosis codes), a dataset is a set of patients, where each patient is a set of codes. Furthermore, each code in a patient is associated with a timestamp and the same code in two different patients can be associated with two different timestamps. In this article, we have chosen to use the notion of timed

association rules that have been successfully used for modeling the behavior of terrorist groups such as Lashkar-e-Taiba [28].

A TAR is of the form

$$\tau : X \rightarrow Y,$$

where $\emptyset \subset X, Y \subseteq P$, and $\tau \in \text{Time}$. The intuitive meaning is that *for all* time points t , if all items in X hold true at t , then all items in Y tend to hold true at time point $t + \tau$. We call X and Y the *antecedent* and *consequent* of the timed association rule, respectively. Notice that it might be the case that $X \cap Y \neq \emptyset$. It is important to note that TARs are not manually written by experts. Rather, they will be automatically extracted using algorithms such as those in References [26, 28]. We also point out that X and Y are not required to be “frequent” (they can be arbitrary subsets of P). Users who are interested only in TARs where X and Y are frequent can express such a desideratum during the TAR mining process—in fact, we will allow them to express far more requirements as discussed below.

Example 2. A temporal database might consist of the following temporal transactions:

(0, AB),
(1, ABC),
(2, ABE),
(3, C),
(4, CE),

where A, B, C , and E are items. An example TAR is $1 : AB \rightarrow C$, saying that when A and B occur in a temporal transaction, C is likely to occur after 1 time point.

We first define when a TAR $\tau : X \rightarrow Y$ “occurs” in a temporal database. Intuitively, this happens when there is a temporal transaction (τ', T') where X holds true (i.e., $X \subseteq T'$) followed by a temporal transaction $(\tau' + \tau, T'')$ where Y holds true (i.e., $Y \subseteq T''$). When $\tau = 0$, the two temporal transactions are the same.

Definition 1. Let D be a temporal database and r a TAR $\tau : X \rightarrow Y$. Two (not necessarily distinct) temporal transactions $\mathbf{t}' = (\tau', T')$ and $\mathbf{t}'' = (\tau'', T'')$ of D *support* r if $X \subseteq T'$, $Y \subseteq T''$, and $\tau'' = \tau' + \tau$. The *supporting set* of r is

$$\text{SuppSet}(r) = \{(\mathbf{t}', \mathbf{t}'') \mid \mathbf{t}', \mathbf{t}'' \in D \text{ and } \mathbf{t}', \mathbf{t}'' \text{ support } r\}.$$

Example 3. Consider the temporal database of Example 2 and the TAR $r = 1 : AB \rightarrow C$. The first and second (respectively, third and fourth) temporal transactions support r .

Support, confidence, and lift for TARs are defined in the following natural way.

Definition 2. The support, confidence, and lift of a timed association rule r of the form $\tau : X \rightarrow Y$ w.r.t. a temporal database D are respectively defined as follows:

$$\begin{aligned} \text{Supp}(r) &= \frac{|\text{SuppSet}(r)|}{|\{(\tau', T') \in D \mid \tau' \leq \tau_{\max} - \tau\}|}; \\ \text{Conf}(r) &= \frac{|\text{SuppSet}(r)|}{|\{(\tau', T') \in D \mid \tau' \leq \tau_{\max} - \tau \wedge X \subseteq T'\}|}; \\ \text{Lift}(r) &= \frac{\text{Conf}(r)}{|\{(\tau', T') \in D \mid Y \subseteq T'\}| / |D|}. \end{aligned}$$

Notice that in the definition of support, we do not use $|D|$ in the denominator but rather use $|\{(\tau', T') \in D \mid \tau' \leq \tau_{\max} - \tau\}|$. That is, we do not count the temporal transactions (τ'', T'') in D with $\tau_{\max} - \tau < \tau'' \leq \tau_{\max}$. The reason is that after time point $\tau_{\max} - \tau$, there is no chance for r to occur, and thus the temporal transactions with time points greater than $\tau_{\max} - \tau$ should be

ignored. As an example, suppose $Time = \{0, \dots, 3\}$, and thus $\tau_{max} = 3$. Consider now the following temporal database D :

$$\begin{aligned} &(0, T_0), \\ &(1, T_1), \\ &(2, T_2), \\ &(3, T_3), \end{aligned}$$

where the T_i 's are arbitrary itemsets. Consider also a TAR $r = 2 : X \rightarrow Y$, where X and Y are arbitrary itemsets. Then, there is no chance for r to occur at time points 2 and 3, because there cannot be temporal transactions at time points 4 and 5, so the temporal transactions for time points 2 and 3 should not be considered when computing the support of r . An analogous argument can be applied to other metrics.

Notice that setting a time window $Time$ is not a limitation. In fact, if one wants to count all temporal transactions in a temporal database when computing support (or confidence or other metrics) of a TAR $\tau : X \rightarrow Y$, it suffices to set $\tau_{max} = \tau + \tau'$, where τ' is the highest time point in the temporal database. In such a case, the formula for support becomes $Supp(r) = \frac{|SuppSet(r)|}{|D|}$. For instance, in the example above, τ_{max} can be set to $3 + 2 = 5$ if all temporal transactions have to be counted in the denominator of support.

Example 4. Consider again the scenario of Example 1 and suppose $\tau_{max} = 3$. Notice that the temporal database does not contain a temporal transaction for time point 3. Then, the support and confidence of r'_1 , r'_2 , and r'_3 are the ones reported in Example 1. For instance, $Supp(r'_2) = 1/3$, because r'_2 occurred once (across time points 0 and 1) out of three possible chances it had to occur (namely, in the intervals $[0, 1]$, $[1, 2]$, and $[2, 3]$).

Consider now the scenario of Example 3 and suppose $\tau_{max} = 4$. The support of $r = 1 : AB \rightarrow C$ is $2/4$, its confidence is $2/3$, and its lift is $10/9$. In this case, $Supp(r) = 2/4$, because r occurred twice (namely, in the time intervals $[0, 1]$ and $[2, 3]$) of four possible chances it had to occur (namely, in the time intervals $[0, 1]$, $[1, 2]$, $[2, 3]$, and $[3, 4]$).

Notice that the time points in a temporal database and in TARs refer to the same set $Time$ of time points, that is, they refer to the same granularity of time. This is necessary to correctly identify occurrences of TARs in the data as well as to correctly compute metrics; also, this is a natural scenario in practice. For instance, if time points in the temporal database at hand are months, it is natural for a user to look for TARs $\tau : X \rightarrow Y$ where τ means months. Nonetheless, one might also be interested in a different granularity for TARs, e.g., a user might be interested in patterns occurring across years, that is, in TARs $\tau : X \rightarrow Y$ where τ means years. In this latter case, the temporal database has to be preprocessed and converted into a new one where time points refer to years or, alternatively, the temporal database is not modified and then algorithms to mine TARs have to take care of this issue “on the fly.”

Similarly to mining classical ARs, interesting TARs are determined w.r.t. some metrics of interest. Thus, we assume we are given N arbitrary measures (which might include standard metrics like support, confidence, lift, etc.) and each TAR r is associated with an N -tuple $\bar{v}_r$, called *measure vector*, consisting of one value for each measure of interest (of course, such values are computed w.r.t. a given temporal database). For instance, if we are interested only in support and confidence, then $N = 2$ and each TAR r is associated with a measure vector consisting of two values, which are the support and confidence of r . The measure vector of r in Example 4 above is $(2/4, 2/3)$. As another example, Reference [26] considered the problem of mining TARs of the form considered in this article w.r.t. support and a metric called *direction confidence*. We do not commit to any specific metric(s) but allow users to choose the set of metrics that are relevant for the application at hand, so that the TAR mining process can be customized to fit users' needs.

We also need *relevance constraints* specifying whether a measure vector $\bar{v}_r$ meets a certain criterion. Relevance constraints are defined depending on application needs and are modeled via a function $\varphi : \mathbb{R}^N \rightarrow \{\text{true}, \text{false}\}$ s.t. $\varphi(\bar{v}_r)$ is *true* iff $\bar{v}_r$ meets the desired constraints—in such a case we say that the timed association rule r is *relevant*. For instance, the measures of interest might be support and confidence, and relevant TARs are those having support and confidence greater than or equal to user-specified thresholds. Besides support and confidence, other measures can be defined in our setting similarly to what previously done for support, confidence, and lift. It is important to note that it is not mandatory for system users or manager to set all of these thresholds. Users can choose zero, one or any number of the metrics and the respective thresholds that are of interest for the application at hand. Thus, once again, we give users the flexibility of customizing when a timed association rule is relevant. We also point out that these thresholds only concern the TAR mining task (and they have to be specified also in the classical AR mining process) and do not concern the summarization process, which is the principal goal of this article and that we address in the next several sections.

Finally, we might be interested only in TARs having a time point τ at most a certain time threshold θ_τ . Of course, when $\theta_\tau = \tau_{\max}$, we are interested in all relevant TARs. Thus, the problem of mining TARs is defined as follows.

An instance of the *Relevant Timed Association Rule Mining* problem is a triple $(D, \varphi, \theta_\tau)$, where D is a temporal database, φ are relevance constraints, and θ_τ is a time threshold. The problem is to find all relevant TARs with time point at most θ_τ . Please note that the results of solving the Relevant TAR Mining Problem will constitute an input to the summarization algorithms proposed in this article.

To simplify presentation, from now on, we assume that the measures of interest, the relevance constraints, and the time threshold have been (arbitrarily) fixed. So, an instance of the relevant TAR mining problem simply is a temporal database D .

3 SUMMARIZATION OF TIMED ASSOCIATION RULES

Mining TARs, even when they are subject to constraints like minimum support and confidence, may yield a huge set of such rules with high redundancy, which makes it hard to understand and analyze them—this issue has been already widely recognized in the non-temporal setting [17, 19, 20, 22]. Thus, our goal is to summarize a huge set of TARs via “small” sets of representative TARs.

Our approach is based on the following considerations. More general, simpler rules are preferable and easier to interpret, while very specific rules can be very hard to understand and difficult to use. More specific rules (with many items in the antecedent) tend to overfit the data and have little predictive power; in contrast, more general rules are more broadly applicable because of fewer items in the antecedent. A specific rule should be justified only when it differs significantly from *every* more general rule in terms of some user-defined measures, which means that there is no more general rule that adequately represents the more specific one.

First, we need to define what it means for one TAR to represent another. For this, we compare two TARs (i) in terms of their structure, that is, syntactically, and thus regardless of the data at hand, and (ii) in terms of their measure vectors, thereby taking the data into account. Thus, we assume a distance function Δ measuring the *distance* between two TARs on the basis of their measure vectors (e.g., cosine distance, Manhattan distance, etc.).

The following definition says when a TAR r' represents another TAR r'' .

Definition 3. Let $\theta = (\theta_T, \theta_M)$, where $\theta_T \geq 0$ is called the *time threshold* and $\theta_M \geq 0$ is called the *measure threshold*. Given two TARs $r' = \tau' : X' \rightarrow Y'$ and $r'' = \tau'' : X'' \rightarrow Y''$, we say that r' θ -represents r'' if

- (1) $X' \subseteq X''$, and
- (2) $Y' = Y''$, and
- (3) $|\tau' - \tau''| \leq \theta_T$, and
- (4) $\Delta(r', r'') \leq \theta_M$.

The preceding definition says that r' θ -represents r'' when (1) r' is more general than r'' w.r.t. their antecedent, (2) r' and r'' have the same consequent, (3) the difference between their time points is at most θ_T , and (4) the difference of their measure vectors is at most θ_M . A timed association rule always θ -represents itself. Notice the parameters above can be set to “default” values if they do not have to play any role. For instance, when the time points do not matter, one can set θ_T to a large value (e.g., a value greater than any TAR’s time point) so that Condition (3) in Definition 3 is always satisfied. However, if a user wants to require that r' can θ -represent r'' only if the two TARs have the same time point, then θ_T must be set to 0. Furthermore, if a user wants to ignore measure vectors, it suffices to define Δ so that it always returns 0—in such a case, Condition (4) in Definition 3 is always satisfied. In general, rather than committing to specific criteria, our framework offers a set of parameters that can be used by users to customize the summarization process to meet their needs.

Let $\mathcal{S}$ be a set of TARs and θ a pair of time and measure thresholds.

The *coverage set* of a TAR r w.r.t. $\mathcal{S}$ and θ is the set of all TARs in $\mathcal{S}$ that are θ -represented by r , that is,

$$c\text{-set}(r, \mathcal{S}, \theta) = \{r' \in \mathcal{S} \mid r \text{ } \theta\text{-represents } r'\}.$$

The *coverage set* of a set of timed association rules $\mathcal{S}'$ w.r.t. $\mathcal{S}$ and θ is defined as follows:

$$c\text{-set}(\mathcal{S}', \mathcal{S}, \theta) = \bigcup_{r \in \mathcal{S}'} c\text{-set}(r, \mathcal{S}, \theta).$$

The *coverage score* of r and $\mathcal{S}'$ w.r.t. $\mathcal{S}$ and θ are defined respectively as follows:

$$\begin{aligned} c\text{-score}(r, \mathcal{S}, \theta) &= |c\text{-set}(r, \mathcal{S}, \theta)|, \\ c\text{-score}(\mathcal{S}', \mathcal{S}, \theta) &= |c\text{-set}(\mathcal{S}', \mathcal{S}, \theta)|. \end{aligned}$$

Thus, the coverage score of r (respectively, $\mathcal{S}'$) w.r.t. $\mathcal{S}$ and θ is the number of TARs in $\mathcal{S}$ that are θ -represented by r (respectively, TARs in $\mathcal{S}'$).

We are now ready to define the problem of summarizing a set of TARs.

PROBLEM 1 (STRONG SUMMARY SET PROBLEM). *An instance of the Strong Summary Set (SSS) problem is a tuple $(D, \mathcal{S}, \theta, k)$, where D is a temporal database, $\mathcal{S}$ is the set of all its relevant TARs, $\theta = (\theta_T, \theta_M)$ is a pair consisting of a time threshold θ_T and a measure threshold θ_M , and k is a positive integer with $k \leq |\mathcal{S}|$.*

A strong summary set is a subset $\mathcal{S}'$ of $\mathcal{S}$ with $|\mathcal{S}'| \leq k$.

The SSS problem asks for a strong summary set that maximizes coverage score (w.r.t. $\mathcal{S}$ and θ).

Thus, the SSS problem asks for a subset $\mathcal{S}'$ of $\mathcal{S}$ containing at most k TARs and representing as many TARs in $\mathcal{S}$ as possible, that is, we want to pick from $\mathcal{S}$ the most representative timed association rules.

We now consider a variant of the problem where the summary set $\mathcal{S}'$ does not have to be a subset of $\mathcal{S}$. This means that there might be TARs in $\mathcal{S}'$ that are not relevant; however, they must represent some TARs in $\mathcal{S}$ and thus be “close enough” to them according to a distance metric, as illustrated in the following example.

Example 5. Consider the temporal database consisting of the following temporal transactions:

(0, ABCX),
 (1, ACDX),
 (2, ACDEX),
 (3, ACEX),
 (4, AE).

Suppose a timed association rule is relevant if its support is at least $1/5$ and its confidence is at least $4/5$.

The following timed association rules are relevant (of course, there are also other relevant timed association rules):

1 : $ABC \rightarrow X$
 1 : $ACD \rightarrow X$
 1 : $ACDE \rightarrow X$

The timed association rule 1 : $AC \rightarrow X$ is not relevant, because its confidence is $3/4$. However, it might be considered a reasonable summary of the timed association rules above if we are willing to tolerate its difference in support and confidence from the TARs it summarizes. In this regard, it should be noted that the relevance thresholds are often somewhat arbitrary, and thus, one can argue that there is no reason to stick to relevant TARs that exactly comply with those thresholds.

The previous example shows that it might well be the case that a non-relevant TAR is a reasonable summary of several relevant TARs, but it cannot be included in a strong summary set, because it is not relevant.

These observations lead us to the following relaxation of the SSS problem where a summary set might contain TARs that are not relevant, provided that they represent some timed association rules in $\mathcal{S}$ and are “close enough” to them.

PROBLEM 2 (WEAK SUMMARY SET PROBLEM). *An instance of the Weak Summary Set (WSS) problem is an SSS problem instance $(D, \mathcal{S}, \theta, k)$.*

A weak summary set is a set $\mathcal{S}'$ of timed association rules s.t. $|\mathcal{S}'| \leq k$.

The WSS problem asks for a weak summary set that maximizes coverage score (w.r.t. $\mathcal{S}$ and θ).

As opposed to Problem 1, a weak summary set $\mathcal{S}'$ is not required to be a subset of $\mathcal{S}$, and thus some TARs in $\mathcal{S}'$ might not be relevant. However, in order for a TAR $r \in \mathcal{S}'$ to contribute to the coverage score of $\mathcal{S}'$, it has to θ -represent at least some TARs in $\mathcal{S}$, which is possible only if r is close enough to at least one TAR in $\mathcal{S}$. Moreover, the WSS problem looks at maximizing coverage score, which means that WSS computes summaries that cover as many TARs as possible.

4 COMPLEXITY ANALYSIS

This section studies the computational complexity of the SSS and WSS problems. Specifically, we show that both are NP-hard. We also analyze their complexity under the assumption that the number of items in TARs is bounded by a constant. We show that the problems remain NP-hard even under this restriction.

The following theorems state that the decision versions of the SSS and WSS problems are NP-complete. Membership in NP holds when Δ can be computed in polynomial time.

THEOREM 1. *Let $(D, \mathcal{S}, \theta, k)$ be an SSS problem instance and $T \leq |\mathcal{S}|$. Deciding whether there exists a strong summary set with coverage score at least T is NP-complete.*

PROOF. Membership. We guess a subset $\mathcal{S}'$ of $\mathcal{S}$ and check whether (i) $|\mathcal{S}'| \leq k$ and (ii) the coverage score of $\mathcal{S}'$ is greater than or equal to T (which can be done in polynomial time, provided that Δ can be computed in polynomial time).

Hardness. We reduce the vertex cover problem to our problem. An instance of the vertex cover problem is an undirected graph $G = (V, E)$ and a positive integer C . The question is whether there exists $V' \subseteq V$ (called *vertex cover*) with $|V'| \leq C$ s.t. for each edge $\{V_i, V_j\} \in E$ at least one of V_i and V_j belongs to V' .

Without loss of generality, we assume that G has no isolated vertices, that is, every vertex has at least one incident edge. Notice that this means that the graph has at least two vertices and at least one edge, and the degree of every vertex is at least one.

We derive an instance of our problem as follows.

The set of items is $V \cup \{A, B\}$, where A and B are arbitrary items not occurring in V .

Let confidence be the only measure of interest and $\Delta(r, r')$ be defined as $|\text{Conf}(r) - \text{Conf}(r')|$. A timed association rule is relevant if its confidence is no lower than $1/2$. Let the time threshold θ_τ for mining timed association rules be 1.

Let $V = \{V_1, \dots, V_n\}$ and $E = \{E_1, \dots, E_m\}$.

The temporal database D is defined as follows—for ease of presentation, time points are not reported in the temporal transactions, but it is assumed that the first one has time point 0 and then each subsequent temporal transaction has time point $t + 1$ where t is the time point of the preceding temporal transaction, following the order the temporal transactions are listed below.

- (1) For each $V_i \in V$, we have that D contains the following temporal transactions:

$$\left. \begin{array}{l} (-, AV_i) \\ (-, \emptyset) \\ (-, AV_i) \\ (-, B) \\ (-, \emptyset) \\ (-, V_i) \\ (-, \emptyset) \end{array} \right\} \text{replicated } m + 3 \text{ times}$$

- (2) For each $\{V_i, V_j\} \in E$, we have that D contains the following temporal transactions:

$$\left. \begin{array}{l} (-, AV_i V_j) \\ (-, B) \\ (-, \emptyset) \\ (-, V_i V_j) \\ (-, \emptyset) \end{array} \right\} \text{replicated 3 times}$$

- (3) Finally, D contains also the following temporal transactions:

$$\left. \begin{array}{l} (-, A) \\ (-, \emptyset) \end{array} \right\} \text{replicated } m + 1 \text{ times}$$

$$\begin{array}{l} (-, A) \\ (-, B) \end{array}$$

Notice that $|D| = n(5 + 2(m + 3)) + 9m + (m + 3) = 11n + 10m + 2nm + 3$ and each temporal transaction has at most 3 items, so the size of D is polynomial in the size of G .

Let $\deg_i$ denote the degree of vertex $V_i \in V$. We define the following timed association rules:

- Let r^* be $1 : A \rightarrow B$;
- For each $V_i \in V$, let r_i be $1 : AV_i \rightarrow B$;
- For each $\{V_i, V_j\} \in E$, let r_{ij} be $1 : AV_i V_j \rightarrow B$.

According to the definition of D reported above, the following holds:

- $\text{Conf}(r^*) = 1/2$;
- For each $V_i \in V$, $\text{Conf}(r_i) = \frac{1+\deg_i}{2+\deg_i}$ with $1/2 < \text{Conf}(r_i) < 1$, as $\deg_i > 0$;
- For each $\{V_i, V_j\} \in E$, $\text{Conf}(r_{ij}) = 1$.

The set of all relevant timed association rules of D is

$$\mathcal{S} = \{r^*\} \cup \{r_i \mid V_i \in V\} \cup \{r_{ij} \mid \{V_i, V_j\} \in E\}.$$

Notice that $|\mathcal{S}| = n + m + 1$ and each timed association rule in $\mathcal{S}$ has at most four items. Thus, the size of $\mathcal{S}$ is polynomial in the size of G .

Below we show that $\mathcal{S}$ is indeed the set of all relevant timed association rules of D . It can be easily verified that all other timed association rules with time point 1 have confidence 0, except for the following ones:

- For each $V_i \in V$,

$$1 : V_i \rightarrow B \text{ has confidence } \frac{1 + \deg_i}{m + 5 + 4\deg_i};$$

- For each $\{V_i, V_j\} \in E$,

$$1 : V_i V_j \rightarrow B \text{ has confidence } 1/4.$$

However, all the timed association rules above have confidence strictly lower than $1/2$, so they are not relevant—notice that $0 < \deg_i \leq m$ for every $V_i \in V$.

There are no relevant timed association rules with time point 0, as all of them have confidence 0, except for the following ones, whose confidence is however always strictly lower than $1/2$ —this follows from the fact that $n > 1$ and $0 < \deg_i \leq m$ for every $V_i \in V$:

- For each $V_i \in V$,

$$\begin{aligned} 0 : V_i \rightarrow A & \text{ has confidence } \frac{2+\deg_i}{m+5+4\deg_i}; \\ 0 : A \rightarrow V_i & \text{ has confidence } \frac{2+\deg_i}{2n+2m+2}; \end{aligned}$$

- For each $\{V_i, V_j\} \in E$,

$$\begin{aligned} 0 : V_i \rightarrow V_j & \text{ has confidence } \frac{4}{m+5+4\deg_i}; \\ 0 : V_j \rightarrow V_i & \text{ has confidence } \frac{4}{m+5+4\deg_j}; \\ 0 : AV_i \rightarrow V_j & \text{ has confidence } \frac{1}{2+\deg_i}; \\ 0 : AV_j \rightarrow V_i & \text{ has confidence } \frac{1}{2+\deg_j}; \\ 0 : V_i V_j \rightarrow A & \text{ has confidence } 1/4; \\ 0 : A \rightarrow V_i V_j & \text{ has confidence } \frac{1}{2n+2m+2}; \\ 0 : V_i \rightarrow AV_j & \text{ has confidence } \frac{1}{m+5+4\deg_j}; \\ 0 : V_j \rightarrow AV_i & \text{ has confidence } \frac{1}{m+5+4\deg_i}. \end{aligned}$$

Let θ_M be the maximum value in

$$\left\{ 1 - \frac{1 + \deg_i}{2 + \deg_i} \mid V_i \in V \right\} \cup \left\{ \frac{1 + \deg_i}{2 + \deg_i} - 1/2 \mid V_i \in V \right\}.$$

Notice that $0 < \theta_M < 1/2$, as $1/2 < \frac{1+\deg_i}{2+\deg_i} < 1$ for every $V_i \in V$.

Finally, $\theta = (0, \theta_M)$, $k = C + 1$, and $T = |\mathcal{S}|$.

Notice that the reduction can be done in polynomial time.

In the following we will use the following two properties, which can be easily verified:

- *Property 1:* r^* θ -represents every $r_i \in \mathcal{S}$ but does not θ -represent any of the r_{ij} 's in $\mathcal{S}$ (because $\Delta(r^*, r_{ij}) = 1/2 > \theta_M$);
- *Property 2:* For each $r_{ij} \in \mathcal{S}$, both r_i and r_j θ -represent r_{ij} .

We show that G has a vertex cover V' s.t. $|V'| \leq C$ iff there exists a strong summary set $\mathcal{S}'$ with coverage score at least T .

($\Rightarrow$) Suppose V' is a vertex cover of G s.t. $|V'| \leq C$. Let

$$\mathcal{S}' = \{r^*\} \cup \{r_i \mid V_i \in V'\}.$$

Obviously, $\mathcal{S}' \subseteq \mathcal{S}$. Moreover, notice that $|\mathcal{S}'| \leq k$, as $|\mathcal{S}'| = |V'| + 1$, $k = C + 1$, and $|V'| \leq C$. Thus, $\mathcal{S}'$ is a strong summary set.

Below we show that $\mathcal{S}'$ has a coverage score equal to T . Specifically, we show that for each $r \in \mathcal{S} \setminus \mathcal{S}'$ there exists $r' \in \mathcal{S}'$ that θ -represents r . The following two cases may occur.

- (1) $r = r_i$, for some $V_i \in V$. Then, $r' = r^*$, as r^* belongs to $\mathcal{S}'$ and θ -represents r_i by Property 1.
- (2) $r = r_{ij}$. Since V' is a vertex cover of G , it has to include either V_i or V_j or both. By definition of $\mathcal{S}'$, either r_i or r_j or both belong to $\mathcal{S}'$, and they θ -represent r_{ij} by Property 2.

($\Leftarrow$) Let $\mathcal{S}'$ be a strong summary set with coverage score at least T . Indeed, the coverage score must be equal to T , as $T = |\mathcal{S}|$. Thus, for each $r \in \mathcal{S} \setminus \mathcal{S}'$ there exists $r' \in \mathcal{S}'$ that θ -represents r . Also, by definition of strong summary set, $|\mathcal{S}'| \leq k$. Notice that $r^* \in \mathcal{S}'$ as there is no other $r \in \mathcal{S}$ that θ -represents r^* .

Let $V' \subseteq V$ be defined as follows:

$$V' = \{V_i \mid r_i \in \mathcal{S}'\} \cup \{V_i \mid r_{ij} \in \mathcal{S}'\}.$$

Then, $|V'| < |\mathcal{S}'|$ (the inequality is strict, because $r^* \in \mathcal{S}'$). Since $|\mathcal{S}'| \leq k = C + 1$, $|V'| \leq C$.

Below we show that V' is a vertex cover of G . For each edge $\{V_i, V_j\} \in E$, one of the following two cases occurs:

- (1) $r_{ij} \notin \mathcal{S}'$, and thus either r_i or r_j or both belong to $\mathcal{S}'$ (as they are the only timed association rules that θ -represent r_{ij} , by Properties 1 and 2) and thus V_i or V_j or both belong to V' by definition (see the first set in the definition of V' above);
- (2) $r_{ij} \in \mathcal{S}'$, and thus $V_i \in V'$ by definition of V' (see the second set in its definition above).

Hence, V' is a vertex cover of G . □

THEOREM 2. *Let $(D, \mathcal{S}, \theta, k)$ be a WSS problem instance and $T \leq |\mathcal{S}|$. Deciding whether there exists a weak summary set with coverage score at least T is NP-complete.*

PROOF. *Membership.* We guess a set $\mathcal{S}'$ of timed association rules and check whether (i) $|\mathcal{S}'| \leq k$, (ii) $c\text{-score}(r, \mathcal{S}, \theta) > 0$ for each $r \in \mathcal{S}'$, and (iii) the coverage score of $\mathcal{S}'$ is at least T (the latter two conditions can be checked in polynomial time, provided that Δ can be computed in polynomial time).

Hardness. We reduce the vertex cover problem to our problem. An instance of the vertex cover problem is an undirected graph $G = (V, E)$ and a positive integer C . The question is whether there exists $V' \subseteq V$ (called *vertex cover*) with $|V'| \leq C$ s.t. for each edge $\{V_i, V_j\} \in E$ at least one of V_i and V_j belongs to V' .

Without loss of generality, we assume that G has no isolated vertices, that is, every vertex has at least one incident edge. Notice that this means that the graph has at least two vertices and at least one edge, and the degree of every vertex is at least one.

We derive an instance of our problem as follows.

The set of items is $V \cup \{A\}$, where A is an arbitrary item not occurring in V .

Let confidence be the only measure of interest and $\Delta(r, r')$ be defined as $|\text{Conf}(r) - \text{Conf}(r')|$. A timed association rule is relevant if its confidence is no lower than 1. Let the time threshold θ_τ for mining timed association rules be 1.

Let $V = \{V_1, \dots, V_n\}$ and $E = \{E_1, \dots, E_m\}$.

The temporal database D is defined as follows—for ease of presentation, time points are not reported in the temporal transactions, but it is assumed that the first one has time point 0 and then each temporal transaction has time point $t + 1$ where t is the time point of the preceding temporal transaction, following the order the temporal transactions are listed below.

- (1) For each $V_i \in V$, we have that D contains the following temporal transactions:

$$\begin{aligned} &(-, V_i) \\ &(-, \emptyset) \end{aligned}$$

- (2) For each $\{V_i, V_j\} \in E$, we have that D contains the following temporal transactions:

$$\begin{aligned} &(-, V_i V_j) \\ &(-, A) \\ &(-, \emptyset) \end{aligned}$$

Notice that $|D| = 2n + 3m$ and each temporal transaction in D has at most two items, so the size of D is polynomial in the size of G .

For each $\{V_i, V_j\} \in E$, we define r_{ij} as $1 : V_i V_j \rightarrow A$. It is easy to see that $\text{Conf}(r_{ij}) = 1$. The set of all relevant timed association rules is

$$\mathcal{S} = \{r_{ij} \mid \{V_i, V_j\} \in E\}.$$

As $|\mathcal{S}| = m$ and each timed association rule in $\mathcal{S}$ has three items, the size of $\mathcal{S}$ is polynomial in the size of G .

Let $\deg_i$ denote the degree of vertex $V_i \in V$. It can be easily verified that all other timed association rules with time point 1 have confidence 0, except for the following ones:

- For each $V_i \in V$, we define r_i as follows:

$$1 : V_i \rightarrow A, \text{ which has confidence } \frac{\deg_i}{1 + \deg_i}.$$

However, the timed association rules above have confidence strictly lower than 1, so they are not relevant.

There are no relevant timed association rules with time point 0, as all of them have confidence 0, except for the following ones, whose confidence is, however, always strictly lower than 1, as $0 < \deg_i$ for every $V_i \in V$:

- For each $\{V_i, V_j\} \in E$,

$$\begin{aligned} 0 : V_i \rightarrow V_j &\text{ has confidence } \frac{1}{1 + \deg_i}; \\ 0 : V_j \rightarrow V_i &\text{ has confidence } \frac{1}{1 + \deg_j}. \end{aligned}$$

Let $\theta_M = \max \left\{ 1 - \frac{\deg_i}{1 + \deg_i} \mid V_i \in V \right\}$. Notice that $\theta_M < 1$, as $\deg_i > 0$ for every $V_i \in V$.

Finally, $\theta = (0, \theta_M)$, $k = C$, and $T = |\mathcal{S}|$.

Notice that the reduction can be done in polynomial time.

In the following we will use the following property, which can be easily verified:

Property 1: For each r_{ij} in $\mathcal{S}$, both r_i and r_j θ -represent r_{ij} . In this regard, notice that $\Delta(r_i, r_{ij}) \leq \theta_M$ and $\Delta(r_j, r_{ij}) \leq \theta_M$.

Below we show that G has a vertex cover V' s.t. $|V'| \leq C$ iff there exists a weak summary set with coverage score at least T .

($\Rightarrow$) Suppose V' is a vertex cover of G s.t. $|V'| \leq C$.

Let $\mathcal{S}' = \{r_i \mid V_i \in V'\}$. Obviously, $|\mathcal{S}'| \leq k$. Thus, $\mathcal{S}'$ is a weak summary set.

We now show that $\mathcal{S}'$ has coverage score equal to T . Specifically, we show that for each $r \in \mathcal{S} \setminus \mathcal{S}'$ there exists $r' \in \mathcal{S}'$ that θ -represents r . Clearly, $r = r_{ij}$ for some $\{V_i, V_j\} \in E$. As V' is a vertex cover of G , either V_i or V_j or both belong to V' , and thus either r_i or r_j or both belong to $\mathcal{S}'$ by definition. By Property 1, r_i and r_j θ -represent r_{ij} .

($\Leftarrow$) Let $\mathcal{S}'$ be a weak summary set with coverage score at least T . Indeed, the coverage score of $\mathcal{S}'$ has to be equal to T , as $T = |\mathcal{S}|$. Thus, for each $r \in \mathcal{S} \setminus \mathcal{S}'$, there is $r' \in \mathcal{S}'$ that θ -represents r . Let $V' = \{V_i \mid r_i \in \mathcal{S}'\} \cup \{V_i \mid r_{ij} \in \mathcal{S}'\}$. Then, $|V'| \leq |\mathcal{S}'| \leq k = C$.

We now show that V' is a vertex cover of G . First, notice that for each $r_{ij} \in \mathcal{S}$, the only timed association rules that can θ -represent it are r_{ij} itself, r_i , and r_j (all other timed association rules r have confidence 0 and thus $\Delta(r_{ij}, r) = 1 > \theta_M$).

For each edge $\{V_i, V_j\} \in E$, one of the following cases occurs:

- (1) $r_{ij} \notin \mathcal{S}'$, and thus either r_i or r_j or both belong to $\mathcal{S}'$ (as they are the only timed association rules that can θ -represent r_{ij}) and thus V_i or V_j or both belong to V' by definition;
- (2) $r_{ij} \in \mathcal{S}'$, and thus $V_i \in V'$ by definition of V' .

Hence, V' is a vertex cover of G . □

In light of the intractability results above, a natural question is whether the complexity drops when the number of items in TARs is bounded by a constant. Unfortunately, we show that both problems remain NP-hard even under such a restriction.

Given a positive integer C , we say that a set of TARs is *C-bounded* if every TAR in it contains at most C items.

THEOREM 3. *Let C be a positive integer and $(D, \mathcal{S}, \theta, k)$ an SSS problem instance, where $\mathcal{S}$ can be either the set of all relevant TARs of D or the set of all C -bounded relevant TARs of D . Let $T \leq |\mathcal{S}|$. Deciding whether there exists a C -bounded strong summary set with coverage score at least T is NP-complete.*

PROOF. Notice that the reduction used in the proof of Theorem 1 uses timed association rules with at most 4 items. Hence, the claim follows. □

THEOREM 4. *Let C be a positive integer and $(D, \mathcal{S}, \theta, k)$ a WSS problem instance, where $\mathcal{S}$ can be either the set of all relevant TARs of D or the set of all C -bounded relevant TARs of D . Let $T \leq |\mathcal{S}|$. Deciding whether there exists a C -bounded weak summary set with coverage score at least T is NP-complete.*

PROOF. Notice that the reduction used in the proof of Theorem 2 uses timed association rules with at most 3 items. Hence, the claim follows. □

5 POLYNOMIAL APPROXIMATION ALGORITHMS

As the SSS and WSS problems are intractable, we now develop polynomial-time approximation algorithms, analyze their worst-case time complexity, and show that the algorithm for solving the SSS problem has approximation guarantees.

ALGORITHM 1: STAR-S**Input:** An SSS problem instance $(D, \mathcal{S}, \theta, k)$.**Output:** A set $\mathcal{S}' \subseteq \mathcal{S}$ with $|\mathcal{S}'| \leq k$ and a set $\mathcal{S}_U \subseteq \mathcal{S}$.

```

1:  $\mathcal{S}' = \emptyset$ ;
2:  $\mathcal{S}_U = \mathcal{S}$ ;
3: for  $i = 1$  to  $k$  do
4:   if  $\mathcal{S}_U \neq \emptyset$  then
5:      $r^* = \arg \max_{r \in \mathcal{S} \setminus \mathcal{S}'} c\text{-score}(r, \mathcal{S}_U, \theta)$ ;
6:      $\mathcal{S}' = \mathcal{S}' \cup \{r^*\}$ ;
7:      $\mathcal{S}_C = c\text{-set}(r^*, \mathcal{S}_U, \theta)$ ;
8:      $\mathcal{S}_U = \mathcal{S}_U \setminus \mathcal{S}_C$ ;
9:   else
10:    return  $(\mathcal{S}', \emptyset)$ ;
11: return  $(\mathcal{S}', \mathcal{S}_U)$ ;

```

5.1 A Greedy Algorithm for the SSS Problem

In this section, we present STAR-S, a polynomial-time greedy algorithm to approximately solve the SSS problem. We analyze its worst-case time complexity and show that it is a $(1 - 1/e)$ -approximation algorithm for the SSS problem.

The basic idea of the algorithm is as follows. Let $(D, \mathcal{S}, \theta, k)$ be an SSS problem instance. The algorithm computes a solution by iteratively adding one more TAR to it. Specifically, the algorithm performs (at most) k iterations, and at each iteration it selects a TAR with highest coverage score w.r.t. the set of timed association rules *not yet represented*.

Initially, the coverage score of a TAR r is defined w.r.t. $\mathcal{S}$. As the algorithm proceeds, coverage scores are updated by looking at the set $\mathcal{S}_U$ of currently *uncovered* TARs, i.e. TARs that are not represented by any TAR in the solution being built.

The STAR-S algorithm (Algorithm 1) takes as input an SSS problem instance $(D, \mathcal{S}, \theta, k)$. It gives as output an approximate solution $\mathcal{S}' \subseteq \mathcal{S}$ with $|\mathcal{S}'| \leq k$ along with the set $\mathcal{S}_U \subseteq \mathcal{S}$ of TARs that are not covered by $\mathcal{S}'$. When $\mathcal{S}_U$ is empty, the algorithm has found a strong summary set representing all TARs in $\mathcal{S}$. Since the algorithm also returns $\mathcal{S}_U$, a user can assess the quality of $\mathcal{S}'$ and take actions accordingly. For instance, if the size of $\mathcal{S}_U$ is small enough, a user might want to add $\mathcal{S}_U$ to $\mathcal{S}'$, thereby obtaining a set with slightly more than k TARs but representing all those in $\mathcal{S}$. This is useful when it is hard for users to precisely set a value of k , which is often the case in practice.

We now go into details of the algorithm. It maintains two sets of TARs: $\mathcal{S}'$, which is the summary being built (this will be eventually returned as output), and $\mathcal{S}_U$, which is the set of TARs in $\mathcal{S}$ that are not represented by any TAR in $\mathcal{S}'$ (this will be eventually returned as output along with $\mathcal{S}'$). Both sets are updated as the algorithm proceeds. Initially, $\mathcal{S}'$ is the empty set (line 1) and $\mathcal{S}_U$ is equal to $\mathcal{S}$ (line 2).

The **for** loop on lines 3–10 performs at most k iterations. At each iteration, if $\mathcal{S}_U$ is empty, all TARs in $\mathcal{S}$ are represented by $\mathcal{S}'$, and thus $\mathcal{S}'$ along with the empty set (meaning that are no TARs left to be represented) are returned (line 10).

Otherwise ($\mathcal{S}_U$ is not empty), the algorithm proceeds as follows. A rule r^* in $\mathcal{S} \setminus \mathcal{S}'$ with maximum coverage score (w.r.t. $\mathcal{S}_U$) is selected (line 5), and both $\mathcal{S}'$ and $\mathcal{S}_U$ are updated accordingly (lines 6–8). Specifically, r^* is added to $\mathcal{S}'$ (line 6), and the set $\mathcal{S}_C$ of TARs represented by r^* (computed at line 7) is deleted from $\mathcal{S}_U$ (line 8). Notice that r^* might be a TAR already represented by

ALGORITHM 2: STAR-S-IMPLEMENTATION**Input:** An SSS problem instance $(D, \mathcal{S}, \theta, k)$.**Output:** A set $\mathcal{S}' \subseteq \mathcal{S}$ with $|\mathcal{S}'| \leq k$ and a set $\mathcal{S}_U \subseteq \mathcal{S}$.

```

1:  $\mathcal{S}' = \emptyset$ ;
2:  $n = |\mathcal{S}|$ ;
3:  $M[1..n, 1..n]$ ;
4:  $Score[1..n]$ ;
5:  $Represented[1..n]$ ;
6: for  $1 \leq i \leq n$  do
7:    $M[i, j] = \begin{cases} \text{true} & \text{if } r_i \text{ } \theta\text{-represents } r_j, \\ \text{false} & \text{otherwise.} \end{cases}$ 
8: for  $1 \leq i \leq n$  do
9:    $Score[i] = c\text{-score}(r_i, \mathcal{S}, \theta)$ ;
10: for  $1 \leq i \leq n$  do
11:    $Represented[i] = \text{false}$ ;
12: for  $i = 1$  to  $k$  do
13:   if  $Score[j] \neq 0$  for some  $1 \leq j \leq n$  then
14:      $m = \arg \max_{1 \leq j \leq n} Score[j]$ ;
15:      $\mathcal{S}' = \mathcal{S}' \cup \{r_m\}$ ;
16:     for  $1 \leq j \leq n$  do
17:       if  $M[m, j] = \text{true}$  then
18:          $Represented[j] = \text{true}$ ;
19:         for  $1 \leq \ell \leq n$  do
20:           if  $M[\ell, j] = \text{true}$  then
21:              $M[\ell, j] = \text{false}$ ;
22:              $Score[\ell] = Score[\ell] - 1$ ;
23:   else
24:     return  $(\mathcal{S}', \emptyset)$ ;
25:  $\mathcal{S}_U = \{r_i \mid 1 \leq i \leq n \text{ and } Represented[i] = \text{false}\}$ ;
26: return  $(\mathcal{S}', \mathcal{S}_U)$ ;

```

□

some TAR in $\mathcal{S}'$; nonetheless, if it is the one that offers the best coverage w.r.t. the uncovered timed association rules, then it is chosen.

After at most k iterations, the algorithm returns $\mathcal{S}'$ along with $\mathcal{S}_U$ (line 11).

THEOREM 5. *The worst-case time complexity of algorithm STAR-S is $O(\text{cost}_\Delta \times |\mathcal{S}|^2)$, where cost_Δ is the cost of computing Δ for two timed association rules.*

PROOF. A more low-level implementation of Algorithm 1 is reported in Algorithm 2.

Let $n = |\mathcal{S}|$. Algorithm 2 maintains a Boolean $n \times n$ matrix M , an n -vector $Score$, and a Boolean n -vector $Represented$, whose entries have the following meaning:

- $M[i, j]$ is *true* if r_i θ -represents r_j , and $r_j \in \mathcal{S}_U$; it is *false* otherwise.
- $Score[i]$ stores the coverage score of r_i w.r.t. $\mathcal{S}_U$ and θ . Notice that $Score[i]$ can be computed by counting how many entries $M[i, j]$ are *true*.
- $Represented[i]$ is *true* if there is a timed association rule in $\mathcal{S}'$ that θ -represents r_i ; it is *false* otherwise.

As the algorithm proceeds, M , $Score$, and $Represented$ are updated to reflect that $\mathcal{S}_U$ has changed. Lines 1–5 take constant time.

Lines 6 and 7 takes $O(\text{cost}_\Delta \times n^2)$ time.

Lines 8 and 9 takes $O(n^2)$ times, as each $Score[i]$ can be computed by counting how many entries $M[i, j]$ are *true*.

Lines 10 and 11 take $O(n)$ time.

Below we look at the worst-case time complexity of one iteration of the **for** loop on lines 12–24.

Lines 13 and 14 can be executed with a single pass of $Score$ as follows: find the timed association rule r_m with maximum value in $Score$ along with its value. If the maximum value is 0, then the **if** condition is false, otherwise it is true and r_m has been found. The entire process takes $O(n)$ time.

Line 15 takes constant time.

Lines 16 and 17 take $O(n)$ time.

We use aggregate analysis to determine the overall time complexity of lines 18–22 in the following.

Line 24 takes constant time.

As the **for** loop on lines 12–24 performs at most k iterations, its overall time complexity is $O(k \times n)$, excluding lines 18–22, which we analyze below.

The crucial observation to determine the overall time complexity of lines 18–22 is the following. If the condition $M[m, j] = \text{true}$ on line 17 holds true at a certain iteration, then the condition $M[m', j] = \text{true}$ will never hold true at subsequent iterations. The reason is that if $M[m, j] = \text{true}$ holds true, then lines 18–22 are executed, which set every $M[\ell, j]$ to *false* (see line 21). This means that $M[m', j]$ will be *false*. Hence, lines 18–22 will not be executed for $M[m', j]$. Indeed, this implies that lines 18–22 will never be executed twice for the same $1 \leq j \leq n$. Thus, they will be executed at most n times. Since one execution of lines 18–22 takes $O(n)$ time, the overall time complexity of lines 18–22 is $O(n^2)$.

Line 25 takes $O(n)$ time.

Line 26 takes constant time.

It follows from the analysis above that the worst-case time complexity of algorithm STAR-S is $O(\text{cost}_\Delta \times n^2 + k \times n + n^2)$, which is $O(\text{cost}_\Delta \times n^2)$, as $k \leq n$. $\square$

In the previous theorem, k does not appear in the worst-case time complexity $O(\text{cost}_\Delta \times |\mathcal{S}|^2)$. As shown in the proof of Theorem 5, the worst-case time complexity of STAR-S is $O(k \times |\mathcal{S}| + |\mathcal{S}|^2 + \text{cost}_\Delta \times |\mathcal{S}|^2)$, which can be written simply as $O(\text{cost}_\Delta \times |\mathcal{S}|^2)$, as $k \leq |\mathcal{S}|$. Thus, even if $k \times |\mathcal{S}|$ does not appear in the theoretical worst-case time complexity, in practice k will have an impact on the performance of the algorithm (the lower k , the lower the running time). We also point out that in practice cost_Δ is negligible, as it involves comparing two measure vectors, which is usually a constant time operation (e.g., to compute Euclidean, Manhattan, Cosine, Kullback-Leibler distances).

The following theorem states that STAR-S provides approximation guarantees: the coverage score of the set returned by the algorithm is always at least $(1 - 1/e)$ times the best coverage score.

THEOREM 6. *The STAR-S algorithm is a $(1 - 1/e)$ -approximation algorithm for the SSS problem.*

PROOF. Reference [25] proposed a greedy algorithm to solve the general problem of finding an element of a uniform matroid that maximizes a non-decreasing, submodular function F (defined over the elements of the matroid) s.t. $F(\emptyset) = 0$. The algorithm is very simple, yet guarantees a $(1 - 1/e)$ approximation: It incrementally builds a solution (without backtracking) starting with the empty set; at each iteration it adds an element that most improves the current solution (according to F).

We prove the claim by showing that the SSS problem and the STAR-S algorithm fall in the setting described above.

First, we recall the notions mentioned above.

A *matroid* is a pair (X, I) where X is a finite set and I is a collection of subsets of X (called “independent”), satisfying two axioms:

- (1) $B \in I, A \subset B \Rightarrow A \in I$.
- (2) $A, B \in I, |A| < |B| \Rightarrow \exists x \in B \setminus A \text{ s.t. } A \cup \{x\} \in I$.

A *uniform matroid* is a matroid such that independent sets are all sets of size at most k for some $k \geq 1$.

Given a set X , a function $f : 2^X \rightarrow \mathbb{R}$ is *submodular* iff whenever $X_1 \subseteq X_2 \subseteq X$ and $x \in X \setminus X_2$,

$$f(X_1 \cup \{x\}) - f(X_1) \geq f(X_2 \cup \{x\}) - f(X_2).$$

Furthermore, f is *non-decreasing* if whenever $X_1 \subseteq X$ and $x \in X \setminus X_1$,

$$f(X_1 \cup \{x\}) - f(X_1) \geq 0.$$

Let $(D, \mathcal{S}, \theta, k)$ be an SSS problem instance. The SSS problems asks for a subset $\mathcal{S}' \subseteq \mathcal{S}$ s.t. $|\mathcal{S}'| \leq k$ and $\mathcal{S}'$ maximizes coverage score.

Clearly, all subsets of $\mathcal{S}'$ having at most k elements form a uniform matroid. We need to show that the coverage score function is submodular and non-decreasing.

For the sake of readability, in the following, for a timed association rule or set of timed association rules X we denote $c\text{-score}(X, \mathcal{S}, \theta)$ and $c\text{-set}(X, \mathcal{S}, \theta)$ simply as $c\text{-score}(X)$ and $c\text{-set}(X)$, respectively.

Let $\mathcal{S}_1 \subseteq \mathcal{S}_2 \subseteq \mathcal{S}$ and $r \in \mathcal{S} \setminus \mathcal{S}_2$.

By definition,

$$\begin{aligned} c\text{-score}(\mathcal{S}_1 \cup \{r\}) &= |c\text{-set}(\mathcal{S}_1 \cup \{r\})|, \\ c\text{-score}(\mathcal{S}_1) &= |c\text{-set}(\mathcal{S}_1)|, \\ c\text{-score}(\mathcal{S}_2 \cup \{r\}) &= |c\text{-set}(\mathcal{S}_2 \cup \{r\})|, \\ c\text{-score}(\mathcal{S}_2) &= |c\text{-set}(\mathcal{S}_2)|. \end{aligned}$$

Then,

$$\begin{aligned} c\text{-score}(\mathcal{S}_1 \cup \{r\}) - c\text{-score}(\mathcal{S}_1) &= |c\text{-set}(r) \setminus c\text{-set}(\mathcal{S}_1)|, \\ c\text{-score}(\mathcal{S}_2 \cup \{r\}) - c\text{-score}(\mathcal{S}_2) &= |c\text{-set}(r) \setminus c\text{-set}(\mathcal{S}_2)|. \end{aligned}$$

Since $\mathcal{S}_1 \subseteq \mathcal{S}_2$, it is the case that $c\text{-set}(\mathcal{S}_1) \subseteq c\text{-set}(\mathcal{S}_2)$ and thus

$$c\text{-set}(r) \setminus c\text{-set}(\mathcal{S}_2) \subseteq c\text{-set}(r) \setminus c\text{-set}(\mathcal{S}_1),$$

which implies

$$|c\text{-set}(r) \setminus c\text{-set}(\mathcal{S}_2)| \leq |c\text{-set}(r) \setminus c\text{-set}(\mathcal{S}_1)|.$$

Hence,

$$c\text{-score}(\mathcal{S}_2 \cup \{r\}) - c\text{-score}(\mathcal{S}_2) \leq c\text{-score}(\mathcal{S}_1 \cup \{r\}) - c\text{-score}(\mathcal{S}_1),$$

that is, $c\text{-score}$ is submodular.

Below we show that $c\text{-score}$ is also non-decreasing.

Let $\mathcal{S}_1 \subseteq \mathcal{S}$ and $r \in \mathcal{S} \setminus \mathcal{S}_1$.

By definition,

$$\begin{aligned} c\text{-score}(\mathcal{S}_1 \cup \{r\}) &= |c\text{-set}(\mathcal{S}_1 \cup \{r\})|, \\ c\text{-score}(\mathcal{S}_1) &= |c\text{-set}(\mathcal{S}_1)|. \end{aligned}$$

Then,

$$c\text{-score}(\mathcal{S}_1 \cup \{r\}) - c\text{-score}(\mathcal{S}_1) = |c\text{-set}(r) \setminus c\text{-set}(\mathcal{S}_1)| \geq 0.$$

Hence, $c\text{-score}$ is non-decreasing.

Thus, the SSS problem is an instance of the problem addressed in Reference [25].

Since STAR-S is a specialization of the algorithm in Reference [25] applied to our setting, it follows that STAR-S is an $(1 - 1/e)$ -approximation algorithm. In this regard, notice that STAR-S

ALGORITHM 3: STAR-W**Input:** A WSS problem instance $(D, \mathcal{S}, \theta, k)$.**Output:** A set $\mathcal{S}'$ with $|\mathcal{S}'| \leq k$ and a set $\mathcal{S}_U \subseteq \mathcal{S}$.

```

1:  $\mathcal{S}' = \emptyset$ ;
2:  $\mathcal{S}_U = \mathcal{S}$ ;
3: for  $i = 1$  to  $k$  do
4:   if  $\mathcal{S}_U \neq \emptyset$  then
5:      $(r', r'') = \arg \max_{r', r'' \in \mathcal{S} \cup \mathcal{S}', r' \approx r''} c\text{-score}(\sigma(r', r''), \mathcal{S}_U, \theta)$ ;
6:      $r = \sigma(r', r'')$ ;
7:      $\mathcal{S}' = \mathcal{S}' \cup \{r\}$ ;
8:      $\mathcal{S}_C = c\text{-set}(r, \mathcal{S}_U, \theta)$ ;
9:      $\mathcal{S}_U = \mathcal{S}_U \setminus \mathcal{S}_C$ ;
10:  else
11:    return  $(\mathcal{S}', \emptyset)$ ;
12: return  $(\mathcal{S}', \mathcal{S}_U)$ ;

```

always chooses the timed association rule with the maximum marginal increase in coverage score, as the coverage score is computed w.r.t. $\mathcal{S}_U$, which is the set of uncovered timed association rules (see line 5 of Algorithm 1). $\square$

5.2 A Greedy Algorithm for the WSS Problem

In this section, we present STAR-W, a polynomial-time greedy algorithm to approximately solve the WSS problem.

Algorithm STAR-W is similar to STAR-S in that it builds an approximate solution by iteratively adding one more TAR to it. However, there is an important difference regarding how such a TAR is chosen by STAR-W, which is due to the difference between the WSS and SSS problems: the former does not require a representative rule to belong to $\mathcal{S}$. Thus, as opposed to STAR-S, STAR-W might choose TARs that are not in $\mathcal{S}$.

An important question is how to choose a TAR to add to the current solution. Besides TARs in $\mathcal{S}$ that have not been chosen yet, our heuristic tries to “summarize” two TARs to see if their summarization is representative enough. Such a summary is obtained by taking the intersection of the antecedents of two TARs with the same consequent, and taking a time point that is half-way between the two time points. Recall that the syntactic condition for a TAR $\tau' : X' \rightarrow Y'$ to θ -represent a TAR $\tau'' : X'' \rightarrow Y''$ is that $X' \subseteq X''$, $Y' = Y''$, and $|\tau' - \tau''| \leq \theta_T$ for a threshold θ_T (cf. Definition 3). We formally define this idea below.

Given two (possibly equal) TARs $r' = \tau' : X' \rightarrow Y'$ and $r'' = \tau'' : X'' \rightarrow Y''$, we write $r' \approx r''$ if $X' \cap X'' \neq \emptyset$ and $Y' = Y''$, that is, r' and r'' have a common item in the antecedents and the same consequent.

If $r' \approx r''$, then we define

$$\sigma(r', r'') = \tau : X \rightarrow Y,$$

where $X = X' \cap X''$, $Y = Y' = Y''$, and $\tau = \lceil \tau' + \tau'' \rceil / 2$. Intuitively, $\sigma(r', r'')$ is a syntactic summarization of r' and r'' . Notice that when $r' = r''$, $r' \approx r''$ and $\sigma(r', r'') = r' = r''$.

The STAR-W algorithm (Algorithm 3) takes as input a WSS problem instance $(D, \mathcal{S}, \theta, k)$. It returns as output an approximate solution $\mathcal{S}'$ (not necessarily a subset of $\mathcal{S}$) with $|\mathcal{S}'| \leq k$ along with the set $\mathcal{S}_U$ of timed association rules that are not represented by $\mathcal{S}'$. When $\mathcal{S}_U$ is empty, the algorithm found a weak summary set representing all TARs in $\mathcal{S}$.

The only difference w.r.t. the STAR-S algorithm are lines 5 and 6, which deserve some important remarks. On lines 5 and 6, the algorithm chooses the timed association rule $r = \sigma(r', r'')$ with highest coverage score w.r.t. the set $\mathcal{S}_U$ of uncovered TARs. The choice is made in line 5. Candidate TARs are obtained as the summary of two TARs $r', r'' \in \mathcal{S} \cup \mathcal{S}'$ with $r' \approx r''$. It is worth noting that:

- (1) When $r' = r''$, $\sigma(r', r'') = r' = r''$, and this TAR, say r , belongs to $\mathcal{S} \setminus \mathcal{S}'$. It cannot be the case that r belongs to $\mathcal{S}'$, as its coverage score is 0, because all TARs it represents have been deleted from $\mathcal{S}_U$ when r was added to $\mathcal{S}'$ (see lines 8 and 9). So, in this case, the algorithm is looking at a TAR in $\mathcal{S} \setminus \mathcal{S}'$, that is, a TAR in the input set $\mathcal{S}$ that has not been chosen yet.
- (2) When $r' \neq r''$, the algorithm is summarizing two TARs and looking at the coverage score of their summary. Note that when at least one of r' and r'' is in $\mathcal{S}'$ but not in $\mathcal{S}$, the algorithm is trying to further summarize a TAR that was previously obtained as the summary of two TARs.
- (3) The TAR that is eventually chosen comes from case 1 or 2 depending on which one yields the TAR with the highest coverage score, that is, representing the highest number of currently unrepresented TARs.

The following theorem states the worst-case time complexity of the STAR-W algorithm.

THEOREM 7. *The worst-case time complexity of algorithm STAR-W is $O(\text{cost}_\Delta \times |\mathcal{S}|^3)$, where cost_Δ is the cost of computing Δ for two timed association rules.*

PROOF. A more low-level implementation of Algorithm 3 is reported in Algorithm 4.

Let $n = |\mathcal{S}|$. Algorithm 4 maintains a Boolean $(n + k) \times (n + k) \times n$ (three-dimensional) matrix M , an $(n + k) \times (n + k)$ matrix Score , and a Boolean n -vector Represented , whose entries have the following meaning:

- $M[i, j, \ell]$ is *true* if $r_i \approx r_j$, $\sigma(r_i, r_j)$ θ -represents r_ℓ , and $r_\ell \in \mathcal{S}_U$; it is *false* otherwise.
- $\text{Score}[i, j]$ stores the coverage score of $\sigma(r_i, r_j)$ w.r.t. $\mathcal{S}_U$ and θ , for each r_i and r_j with $r_i \approx r_j$. Notice that $\text{Score}[i, j]$ can be computed by counting how many entries $M[i, j, \ell]$ are *true*. If $r_i \approx r_j$ does not hold true, then $\text{Score}[i, j] = 0$.
- $\text{Represented}[i]$ is *true* if there is a timed association rule in $\mathcal{S}'$ that θ -represents r_i ; it is *false* otherwise.

As the algorithm proceeds, M , Score , and Represented are updated to reflect that $\mathcal{S}_U$ has changed. Below we analyze the worst-case time complexity of Algorithm 4.

Lines 1–5 take constant time.

Lines 6 and 7 can be executed in time $O(\text{cost}_\Delta \times n^3)$.

Lines 8 and 9 can be executed in time $O(n^3)$ as follows: for each $1 \leq i \leq j \leq n$, compute $\text{Score}[i, j]$ by counting how many $M[i, j, \ell]$ are *true*.

Lines 10 and 11 take $O(n)$ time.

Below we analyze the worst-case time complexity of one iteration of the **for** loop on lines 12–29.

Line 13 takes $O(n)$ time.

Line 14 takes $O(n^2)$ time, as $i \leq k \leq n$.

Lines 15 and 16 take constant time.

Lines 17 and 18 take $O(n)$ time.

We use aggregate analysis to determine the overall time complexity of lines 19–23 in the following.

ALGORITHM 4: STAR-W -IMPLEMENTATION**Input:** A WSS problem instance $(D, \mathcal{S}, \theta, k)$.**Output:** A set $\mathcal{S}'$ with $|\mathcal{S}'| \leq k$ and a set $\mathcal{S}_U \subseteq \mathcal{S}$.

```

1:  $\mathcal{S}' = \emptyset$ ;
2:  $n = |\mathcal{S}|$ ;
3:  $M[1..n + k, 1..n + k, 1..n]$ ;
4:  $\text{Score}[1..n + k, 1..n + k]$ ;
5:  $\text{Represented}[1..n]$ ;
6: for  $1 \leq i \leq j \leq n, 1 \leq \ell \leq n$  do
7:    $M[i, j, \ell] = \begin{cases} \text{true} & \text{if } r_i \approx r_j \text{ and } \sigma(r_i, r_j) \theta\text{-represents } r_\ell, \\ \text{false} & \text{otherwise.} \end{cases}$ 
8: for  $1 \leq i \leq j \leq n$  s.t.  $r_i \approx r_j$  do
9:    $\text{Score}[i, j] = c\text{-score}(\sigma(r_i, r_j), \mathcal{S}, \theta)$ ;
10: for  $1 \leq i \leq n$  do
11:    $\text{Represented}[i] = \text{false}$ ;
12: for  $i = 1$  to  $k$  do
13:   if  $\text{Represented}[j] = \text{false}$  for some  $1 \leq j \leq n$  then
14:      $(p', q') = \arg \max_{1 \leq p \leq q \leq n+(i-1)} \text{Score}[p, q]$ ;
15:      $r_{n+i} = \sigma(r_{p'}, r_{q'})$ ;
16:      $\mathcal{S}' = \mathcal{S}' \cup \{r_{n+i}\}$ ;
17:     for  $1 \leq j \leq n$  do
18:       if  $M[p', q', j] = \text{true}$  then
19:          $\text{Represented}[j] = \text{true}$ ;
20:       for  $1 \leq \ell_1 \leq \ell_2 \leq n + i - 1$  do
21:         if  $M[\ell_1, \ell_2, j] = \text{true}$  then
22:            $M[\ell_1, \ell_2, j] = \text{false}$ ;
23:            $\text{Score}[\ell_1, \ell_2] = \text{Score}[\ell_1, \ell_2] - 1$ ;
24:       for  $1 \leq \ell_1 \leq n + i, 1 \leq \ell_2 \leq n$  do
25:          $M[n + i, \ell_1, \ell_2] = \begin{cases} \text{true} & \text{if } r_{n+i} \approx r_{\ell_1} \text{ and } \text{Represented}[\ell_2] = \text{false} \text{ and } \sigma(r_{n+i}, r_{\ell_1}) \theta\text{-represents } r_{\ell_2}, \\ \text{false} & \text{otherwise.} \end{cases}$ 
26:       for  $1 \leq \ell_1 \leq n + i$  s.t.  $r_{n+i} \approx r_{\ell_1}$  do
27:          $\text{Score}[n + i, \ell_1] = c\text{-score}(\sigma(r_{n+i}, r_{\ell_1}), \mathcal{S}_U, \theta)$ ;
28:     else
29:       return  $(\mathcal{S}', \emptyset)$ ;
30:    $\mathcal{S}_U = \{r_i \mid 1 \leq i \leq n \text{ and } \text{Represented}[i] = \text{false}\}$ ;
31: return  $(\mathcal{S}', \mathcal{S}_U)$ ;

```

Lines 24 and 25 take $O(\text{cost}_\Delta \times n^2)$ time, as $i \leq k \leq n$.

Lines 26 and 27 can be executed in $O(n^2)$ time as follows: for each $1 \leq \ell_1 \leq n + i$, compute $\text{Score}[n + i, \ell_1]$ by counting how many $M[n + i, \ell_1, \ell_2]$ are *true*—notice that $i \leq k \leq n$ and M was updated on lines 24 and 25.

Line 29 takes constant time.

As the **for** loop on lines 12–29 performs at most k iterations, its overall time complexity is $O(k \times \text{cost}_\Delta \times n^2)$, excluding lines 19–23, which we analyze below.

The crucial observation to determine the overall time complexity of lines 19–23 is the following. If the condition $M[p', q', j] = \text{true}$ on line 18 holds true at a certain iteration, then the condition $M[p'', q'', j] = \text{true}$ will never hold true at subsequent iterations. The reason is as follows. If $M[p', q', j] = \text{true}$ holds true, then lines 19–23 are executed, which set every $M[\ell_1, \ell_2, j]$ to *false*, for $1 \leq \ell_1 \leq \ell_2 \leq n + i - 1$ (see lines 20–22). Also, for $1 \leq \ell_1 \leq n + i$, $M[n + i, \ell_1, j]$ will be set to *false*

on line 25, because *Represented*[j] has been set to *true* on line 19. This means that $M[p'', q'', j]$ will be *false*. Hence, lines 19–23 will not be executed for $M[p'', q'', j]$. Indeed, this implies that lines 19–23 will never be executed twice for the same $1 \leq j \leq n$. Thus they will be executed at most n times. Since one execution of lines 19–23 takes $O(n^2)$ time, the overall time complexity of lines 19–23 is $O(n^3)$.

Line 30 takes $O(n)$ time.

Line 31 takes constant time.

It follows from the analysis above that the worst-case time complexity of algorithm STAR-W is $O(cost_\Delta \times n^3 + k \times cost_\Delta \times n^2 + n^3)$, which is $O(cost_\Delta \times n^3)$, as $k \leq n$. $\square$

Notice that the worst-case time complexity of STAR-W is higher than that of STAR-S. This is not surprising, as the set of candidate representative TARs (and thus the search space) gets bigger in the WSS problem, because TARs not belonging to $\mathcal{S}$ might be included in a solution. As shown in our experimental evaluation, the results returned by STAR-W are better both in terms of coverage and what humans prefer—suggesting that this tradeoff is well worth it.

6 EXPERIMENTAL EVALUATION

6.1 Setup

We used the World Bank Open Data repository,² the LeT dataset from Reference [28], and the Online Retail dataset³ detailed below.

World Bank Dataset. The World Bank Open Data repository contains 58 years (1960–2018) of approximately 1,600 indicators for over 200 countries. The topics covered by the indicators range from poverty, health, and demographics to GDP, trade, and the environment (e.g., gross domestic product, school enrollment, population, life expectancy at birth, etc.). We looked at data for the entire East African region (Burundi, Congo, Kenya, Rwanda, Tanzania, and Uganda) and identified over 250 indicators whose prediction would be of high interest, such as mortality rates by demographic (e.g., males), age (e.g., infants), as well as fertility rates, life expectancy, educational outcomes (e.g., enrollment in pre-primary, primary, secondary, tertiary institutions, graduating rates) and more. We extracted TARs with $1 \leq \tau \leq 3$ (in this dataset, τ refers to years), that is, TARs of the form $1 : X \rightarrow Y$, $2 : X \rightarrow Y$, $3 : X \rightarrow Y$. To mine TARs, we created three instances of the original dataset, each obtained by shifting the values of the indicators to predict of one, two, and three years, respectively. For instance, to mine TARs that predict the mortality rate one year ahead (i.e., $\tau = 1$), we considered for each transaction the mortality rate of the following year. If the time delay was two years, then we assigned to each transaction the mortality rate of two years later. We then ran the FP-Growth algorithm [12] with minimum support 5%, minimum confidence 65%, and minimum lift 1. This was a domain-driven choice that yielded TARs occurring enough in the data and with good predictive power.

LeT Dataset. The LeT dataset [28] contains 252 months (Jan 1990–Dec 2010) of data involving 770 indicators from which 405 TARs (with $1 \leq \tau \leq 3$) were provided to us. These TARs were evaluated by four terrorism experts who know LeT’s activities well. The TARs only had support and confidence associated with them, not lift.

Retail Dataset. The Retail dataset contains all the transactions occurring for a UK-based and registered, non-store online retail between 01/12/2009 and 09/12/2011, for a total of 981,330 transactions. We extracted TARs with $1 \leq \tau \leq 3$ (in this dataset, τ refers to days) as done for the World Bank dataset with minimum support 5%, minimum confidence 65%, and minimum lift 1. We also

²<https://data.worldbank.org/>.

³<http://archive.ics.uci.edu/ml/datasets/Online+Retail+II>.

tried to extract TARs with lower values of minimum support and minimum confidence, but the total number of TARs did not change.

Table 1 reports the number of TARs per dataset:

Table 1. Number of TARs per Dataset

Dataset	# TARs
Burundi	2,876
Congo	2,254
Kenya	877
Rwanda	3,824
Tanzania	572
Uganda	1,324
LeT	405
Retail	13,071

To determine whether a TAR θ -represents another TAR (cf. Definition 3), (i) we varied the time threshold between 0 and 2 (because the extracted TARs have $1 \leq \tau \leq 3$), and (ii) we compared their confidence and lift and varied the measure threshold from 2% to 10% in steps of 2%—e.g., when the measure threshold is 4% the confidence (respectively, lift) of the two TARs should differ of at most 4%. For the World Bank and Retail datasets we used confidence and lift separately. For the LeT data we used confidence only (as this was the only metric available). We considered confidence and lift as we deal with predictive rules and such two metrics are the ones typically used to measure the predictive ability of association rules (e.g., see Reference [14]).

We varied the maximum size of the summary set from 2% to 10% (of the size of the set of TARs) in steps of 2%. We ran STAR-S and STAR-W over sets of TARs having the same consequent. Recall that a TAR represents another one only if they have the same consequent (cf. Definition 3), so we can partition the dataset into disjoint subsets, one for each consequent, and focus on each subset individually—indeed, several summarization approaches for classical ARs behave in this way (e.g., References [14, 19]). The STAR-S and STAR-W algorithms were implemented in Python. Experiments were run on a Linux cluster of Intel Xeon nodes with RAM ranging from 16 to 64 GB.

6.2 Results

In this section, we analyze our experimental results by looking at (i) the quality of the summaries and (ii) the runtimes of our algorithms. For (i), we both analyzed the coverage scores of the summaries and conducted a survey letting human subjects assess the helpfulness of the summaries (we involved both anonymous subjects recruited via Amazon Mechanical Turk for the World Bank data and domain experts for the LeT dataset).

Comparing Coverage. We ran STAR-S and STAR-W on each configuration described above. For each country in the World Bank dataset, and for the LeT and Retail datasets, Figure 1 shows the average coverage (i.e., the percentage of TARs in the original set that are represented) of the k most representative TARs in a summary returned by STAR-S and STAR-W, where $k \in [1, 10]$ —this indicates to what extent the k most representative TARs in a summary represent the entire original set of TARs. Figure 1 shows that a relatively small number of rules (just 10) achieve high coverage scores, thereby confirming the high redundancy in the original set and the need of summarizing it, as well as the effectiveness of our algorithms at finding few very representative TARs.

More specifically, except for Rwanda and LeT, the coverage score of the most representative timed association rule goes from 60% to over 90% for STAR-W, and from 30% to over 90% for

Fig. 1. Average coverage for best TARs.

STAR-S. In the case of Rwanda, the coverage is slightly worse. For the LeT dataset, STAR-W still performs very well, while STAR-S has low coverage score. The reason is that the original set of TARs contains many specific rules and none of them can be summarized by a TAR in the same set, which is what the STAR-S algorithm looks for. In contrast, the STAR-W algorithm is not constrained to necessarily pick a rule from the original set, and thus it can find a more general TAR (not included in the original set) that summarizes several TARs, whence the high coverage scores yielded by STAR-W. For the Retail dataset, both STAR-W and STAR-S perform very well.

The best k TARs returned by STAR-W are never worse than those returned by STAR-S—for Congo, Rwanda, Uganda, and LeT, the best STAR-W TARs are significantly better.

For both STAR-S and STAR-W, Figure 2 shows the ratio of the coverage score of the k most representative TARs to the coverage score of the entire summary,⁴ where $k \in [1, 10]$. Each bar in the figure is the average value over all the configurations we used. Such ratios indicate the extent to which the k most representative TARs in a summary contribute to the coverage score of the entire summary. For all datasets, on average, the coverage score of the most representative timed association rule is over 60% (WorldBank) and 70% (LeT and Retail) of the coverage score of the entire summary returned by STAR-W. For the World Bank (respectively, Retail) dataset,

⁴Recall that the coverage score of a summary is the number of TARs in the original set that are represented by the summary.

Fig. 2. Average relative coverage for the best TARs.

Fig. 3. IR distribution.

the coverage score of the most representative timed association rule is over 50% (respectively, 60%) of the coverage score of the summary returned by STAR-S. Regarding the LeT dataset, the most representative timed association rule does not have high coverage score and more timed association rules are needed to raise the coverage. As mentioned earlier in the article, the reason is that there are many specific TARs that cannot be summarized by a timed association rule in the same set. Still, a few rules suffice to get very close to the coverage score of the entire summary set—the most representative 10 rules have a coverage score that is almost 90% and 100% of the coverage score of the summary set returned by STAR-S and STAR-W, respectively.

To compare STAR-S with STAR-W, we also computed the improvement ratio (IR), i.e. the ratio of the coverage score of the summary returned by STAR-W to the coverage score of the summary returned by STAR-S. Clearly, when $IR > 1$ (respectively, $IR < 1$) STAR-W provided a summary representing more (respectively, less) TARs than the summary provided by STAR-S; when $IR = 1$, the summaries returned by STAR-W and STAR-S represent the same number of TARs. For the Retail dataset, IR was always close to 1, thus indicating a similar behavior of STAR-W and STAR-S, which was also shown in Figure 1(h) and Figure 2(c). For the World Bank and LeT datasets, Figure 3 shows the distribution of IR for both datasets, where the x -axis reports different intervals for IR, and each bar is the percentage of the cases where IR fell in the respective range—e.g., for the World Bank data, in 0.4% of the cases IR was in $[0, 1)$. In most of the cases, STAR-W provided a summary with higher coverage score. Specifically, for the World Bank data STAR-S provided a better summary in only 0.4% of the cases. STAR-S and STAR-W provided summaries with the same coverage score in 11.3% of the cases, but in 88.3% of the cases STAR-W provided better summaries. For the LeT data, STAR-W always provided better summaries.

Fig. 4. Varying the measure threshold (World Bank).

Overall, the results discussed thus far show that STAR-W achieves higher coverage scores with fewer TARs.

Our experimental results also showed that varying the time threshold (cf. Definition 3) affected the coverage score of the summary set in a very marginal way. Of course, higher time thresholds allow a TAR to represent more TARs, and thus coverage scores are expected to increase. However, this happened only in a few cases for the World Bank data and the increment was negligible. The reason is that there were few TARs with the same consequent and different time points.

Likewise, higher measure thresholds (cf. Definition 3) allow a TAR to represent more TARs, and thus coverage scores tend to increase. Our experimental results showed that varying the measure threshold had more impact than varying the time threshold for the World Bank dataset, even though the increase of coverage score was in general somewhat limited. A representative case is reported in Figure 4. We see that all other parameters being fixed, higher measure thresholds might yield mild improvements in terms of coverage score.

We noticed the opposite trend for the LeT and Retail datasets, where the measure threshold had an even lower impact than the time threshold. This happened because the TARs in the LeT and Retail datasets have similar confidence, and thus varying the measure threshold does not affect the coverage score.

Survey Findings. To assess the helpfulness of the summaries, we conducted a survey for the World Bank and LeT datasets. For the World Bank data, we asked 100 anonymous human subjects recruited via Amazon Mechanical Turk to analyze 15 sets of TARs along with the corresponding summaries. For the LeT dataset, we asked 4 terrorism experts with expertise on LeT to analyze 6 sets of TARs and their summaries. All the aforementioned summaries were computed with the following parameter setting: minimum support 5%, minimum confidence 65%, minimum lift 1, $k = 2\%$ (of the size of the original set of TARs), $\theta_T = 0$, and $\theta_M = 2\%$ (i.e., the metrics of two TARs should differ of at most 2%). As discussed above, both the measure and the time thresholds have a limited impact on the coverage score, and very high coverage scores are typically reached with few TARs. Thus, in the configuration above, we considered small values for the time threshold, the measure threshold, and k , so as to get summaries with few TARs representing many others, with the TARs in the summaries being very close to the represented ones (because the measure and time thresholds are low).

Specifically, on each page, participants could visualize the entire set of TARs along with the two summaries returned by STAR-S and STAR-W. Participants were then asked to express (i) which summary appeared to be helpful (with four possible answers: STAR-S only was helpful, STAR-W

Fig. 5. Survey findings on helpfulness of the summaries.

Fig. 6. Human judgment of summaries.

only was helpful, both were helpful, neither of them was helpful), (ii) if they preferred the STAR-S summary over the STAR-W one, or vice versa, and (iii) how well the summaries captured the rules (with five possible answers: Not well at all, Slightly well, Moderately well, Very well, Extremely well).

The answers to the first question are reported in Figure 5. For the World Bank datasets, 98% of participants found at least one summary helpful. For the LeT dataset, all domain experts found the summaries helpful, with 75% of them finding both STAR-S and STAR-W helpful, and 25% STAR-W only.

Regarding the preference of a summary over the other, for the World Bank data 74% of the participants were in favor of the STAR-W summaries, while for LeT 100% of the terrorism experts were in favor of the STAR-W summaries.

Finally, Figure 6 shows how well the summaries captured the rules according to human evaluators. On average, all summaries had positive scores, STAR-W summaries were judged better than STAR-S summaries, and STAR-W summaries received very high scores from the terrorism experts.

Comparing Lift. A natural question to ask is whether we can compare the lift of the original TARs (as defined in Definition 2) to the lift of the TARs in the summary. Note that all the rules in both weak and strong summaries are TARs and hence the definition of lift in Definition 2 is directly applicable to them. We compared the average of the lift of the set of TARs being summarized and the average of the lift of the set of TARs in both strong and weak summaries generated by the STAR-S and STAR-W algorithms. The results are shown in Table 2. We can immediately see from

Table 2. Comparison of Average Lifts of Original Set of TARs and Average Lifts of TARs in Strong and Weak Summaries

Dataset	Average Lift	Average Lift STAR-S	Average Lift STAR-W
Burundi	3.92	4.15	4.15
Congo	3.62	3.67	2.97
Kenya	3.75	4.08	4.08
Rwanda	3.57	3.76	4.12
Tanzania	3.73	3.21	3.86
Uganda	3.85	3.74	3.85
Retail	1.30	1.29	1.27
Average	3.39	3.42	3.47

As mentioned earlier in this section, the rules we received for the LeT data did not include lift and hence this experiment was not applied to that data.

Fig. 7. Runtime of STAR-S and STAR-W.

Table 2 that on average, the average lift of rules in all three cases (original set of TARs, strong summaries, and weak summaries) are about the same.

Runtimes. Figure 7 compares STAR-S and STAR-W in terms of running time. For the World Bank data, STAR-S took less than 10 s in 95.5% of the configurations we considered (among these, it took less than 1 s in 5% of the cases). STAR-S never took more than 100 s to complete. However,

the running times of STAR-W are between 1 and 5,000 s with most of the cases between 1,000 and 3,000 s (with a few outliers taking around 9,000 s).

Regarding the LeT data, STAR-S took less than 10 s in 95.8% of the cases (less than 1 s in 25% of the cases) and never took more than 50 s. The running times of STAR-W are mostly between 50 and 1,000 s though 12.5% of the cases took between 2,000 and 2,200 s. Recall that STAR-W pays back with better summaries being returned—and this is the cost.

For the Retail dataset, STAR-S took less than 10 s in 12.5% of the cases, and between 10 and 100 s in 37.5% of the cases. In the remaining cases, it took between 500 and 1,400 s (with several cases around 500 s). However, STAR-W took less than 1,000 s in 12.5% of the cases, and between 6,000 and 7,000 s in 12.5% of the cases. In the remaining cases, STAR-W did not terminate within 2 days. We recall the Retail dataset is much bigger than the World Band and LeT datasets, and that STAR-S (which always terminated in a reasonable amount of time) provided summaries with very high coverage, as discussed before.

7 BRIEF SUMMARY SAMPLES

Using our approach, we were able to effectively summarize many timed association rules through small and easier to interpret summaries. For example, in the case of Kenya, the single timed association rule

$$2: \text{HFCU} = 1.17 \times 10^{12} \text{KSH} \rightarrow \text{Neonatal deaths} = [28897.500, 34852.500]$$

summarizes 50 TARs that have numerous different preconditions (not just one pre-condition as in this rule). This TAR says that when the household final consumption expenditure (HFCU) in the country as a whole is at a certain level in Kenyan shillings, then the number of neonatal deaths after two years is at a very large level, between 28,897 and 34,852 deaths.⁵

To discretize continuous attributes (e.g., Neonatal deaths in the TAR above), we used the *Discretize by Binning* operator within the RapidMiner software,⁶ which partitions a range of numerical values into bins of equal size.

Interestingly, we see an immediate related summary TAR for Rwanda for the same dependent variable, neonatal deaths. Here, 21 TARs (involving two or three preconditions) are summarized via a single TAR:

$$2: \text{TCG} = [57695000, 88080000] \text{USD}, \text{HFCU} = 1.18E + 12 \text{ RWF} \rightarrow \text{Neonatal deaths} = [34.550, 42.150].$$

This rule says that when the value of technical cooperation grants is in the range [57695000 – 88080000] and when the household final consumption expenditure in the country as a whole is = $1.18E + 12$ Rwandan Francs, then the number of neonatal deaths after two years is in the 34.55–42.15 range.

We see immediately from both these summary TARs that HFCU is linked to neonatal deaths. While we do not claim that this rule is unknown to experts in African development (e.g., World Bank or WHO), it does suggest that economic factors are linked to number of neonatal deaths in the two countries.

For the LeT dataset, Tables 3–8 report six weak and six strong summaries. Each table reports a weak and a strong summary for the same original set of TARs. Each row reports a TAR along with the percentage of TARs in the original set that are represented. Overall, TARs in weak summaries represent many TARs, while TARs in strong summaries represent less TARs. As already discussed

⁵According to the World Bank, “HFCU is the market value (formerly private consumption) of all goods and services, including durable products (such as cars, washing machines, and home computers), purchased by households.”

⁶https://docs.rapidminer.com/latest/studio/operators/cleansing/binning/discretize_by_bins.html.

Table 3. Strong and Weak Summary Sets for the LeT Dataset—Consequents =
LeT Carries out One to Three Attacks against Civilians x Months Later

Weak Summary Set 1		
Antecedent	Consequent	coverage
No splinter *	1–3 civilian attacks ** 2 months later	61%
Between 0 and 2 LeT commanders died	1–3 civilian attacks ** 2 months later	36%
Between 0 and 2 LeT commanders died	1–3 civilian attacks ** 1 month later	2%
Between 0 and 2 LeT commanders died	1–3 civilian attacks ** 3 months later	1%

Strong Summary Set 1			
Antecedent		Consequent	coverage
It is possible that LeT made territorial claims	No splinter *	1–3 civilian attacks ** 2 months later	2%
Between 0 and 1 LeT commanders died	No splinter *	1–3 civilian attacks ** 2 months later	2%
LeT's basic organization may have been dissolved	No splinter *	1–3 civilian attacks ** 2 months later	2%
0–1 LeT leader was arrested	No splinter *	1–3 civilian attacks ** 2 months later	2%
It is possible that some LeT commanders were arrested	No splinter *	1–3 civilian attacks ** 2 months later	2%
No negotiations between LeT and its state were taking place	No splinter *	1–3 civilian attacks ** 2 months later	2%
LeT was not involved in any intra-organizational conflict regarding its resources	No splinter *	1–3 civilian attacks ** 2 months later	2%
It is possible that some LeT members were arrested	No splinter *	1–3 civilian attacks ** 2 months later	2%
There was a (partial) government ban	No splinter *	1–3 civilian attacks ** 2 months later	2%
It is possible that some arrested LeT members were released	No splinter *	1–3 civilian attacks ** 2 months later	2%
LeT did not dissolve after a split	No splinter *	1–3 civilian attacks ** 2 months later	2%
It is possible that LeT personnel were killed by the government	No splinter *	1–3 civilian attacks ** 2 months later	2%
It is possible that LeT was carrying out technical manipulations	No splinter *	1–3 civilian attacks ** 2 months later	1%

*LeT did not splinter into new groups.

**LeT carries out one to three attacks against civilians.

before, this is not due to the STAR-S algorithm, but rather to the original TARs. In fact, the original set of TARs contains many specific rules and each of them represents only few other TARs of the same set (recall that a strong summary consists of TARs taken from the original set). In contrast, a weak summary might resort to TARs that were not in the original set, and this allow it to come up with new more general TARs representing many others, which is indeed shown by the weak summaries in Tables 3–8. When the weak summary is smaller than the strong one, less than k TARs in the weak summary were enough to represent the entire original set of TARs.

Table 8 also shows some new results generated by the STAR-S and STAR-W algorithms that were not known before. Prior work in Reference [28] on temporal association rules (called TP-rules) showed that Lashkar-e-Taiba does not carry out attacks on symbolic/tourist targets in the two to three months immediately following months in which LeT had locations across the border from India and either (i) a certain number of LeT leaders died or were killed or (ii) a certain number of LeT operatives were released by the Pakistani government. Four such rules, labeled PST2-ST5, were presented in Reference [28]. However, as the reader can see from Table 8, the current results generate new insights into when LeT does not carry out attacks on symbolic/tourist sites. These include situations where (i) LeT did not dissolve three months earlier, (ii) LeT made territorial claims 3 months earlier, and (iii) LeT was not involved in intra-organizational conflict 3 months earlier, all discovered by STAR-W. In fact, STAR-S also suggests that LeT does not attack symbolic/tourist targets three months after months in which the group was not dissolving and some LeT leaders may

Table 4. Strong and Weak Summary Sets for the LeT Dataset—Consequents = LeT Carries out 0 to One Attack against National Security Forces x Months Later

Weak Summary Set 2		
Antecedent	Consequent	coverage
No tech. manipulations *	0–1 NSF attack ** 3 months later	69%
Arrest warrants for 1-16 LeT personnel were issued	0–1 NSF attack ** 1 month later	29%
Arrest warrants for LeT personnel were issued	0–1 NSF attack ** 2 months later	2%

Strong Summary Set 2			
Antecedent		Consequent	coverage
It is possible that LeT personnel were killed by the government	No tech. manipulations *	0–1 NSF attack ** 3 months later	2%
LeT may have endorsed its breakaway organizations	No tech. manipulations *	0–1 NSF attack ** 3 months later	2%
It is possible that splintering within LeT occurred	No tech. manipulations *	0–1 NSF attack ** 3 months later	2%
Disagreement over the leadership did not initiate splintering within LeT	No tech. manipulations *	0–1 NSF attack ** 3 months later	2%
Disagreement over the instruments used to achieve a goal did not initiate splintering within LeT	No tech. manipulations *	0–1 NSF attack ** 3 months later	2%
LeT did not dissolve after a split	No tech. manipulations *	0–1 NSF attack ** 3 months later	2%
Disagreement over its goals may have caused splintering within LeT	No tech. manipulations *	0–1 NSF attack ** 3 months later	2%
Changes in the environment may have caused splintering within LeT	No tech. manipulations *	0–1 NSF attack ** 3 months later	2%
0-16 LeT personnel were killed by the government	No tech. manipulations *	0–1 NSF attack ** 3 months later	2%

*LeT was not carrying out technical manipulations.

**LeT carries out 0-1 attack against national security forces.

Table 5. Strong and Weak Summary Sets for the LeT Dataset—Consequents = LeT Does Not Carry out Attacks against Security Installations x Months Later

Weak Summary Set 3		
Antecedent	Consequent	coverage
LeT's basic organization did not dissolve	no SI attacks * 1 month later	72%
It is possible that LeT made territorial claims	no SI attacks * 2 months later	20%
LeT recruited individuals of specific nationality	no SI attacks * 2 months later	4%

Strong Summary Set 3			
Antecedent		Consequent	coverage
It is possible that LeT made territorial claims	LeT's basic organization did not dissolve	no SI attacks * 2 months later	6%
It is possible that LeT made territorial claims	It is possible that there were raids on LeT	no SI attacks * 2 months later	4%
It is possible that LeT made territorial claims	There were 0-6 raids on LeT	no SI attacks * 2 months later	4%

*LeT does not carry out attacks against security installations.

have resigned. These are new insights that were generated by the STAR-W and STAR-S algorithms that, to the best of our knowledge, were not previously known. Knowing when symbolic/tourist sites are less likely to be targeted is critical for decision making. For instance, most potential targets operate under various threat levels—when attacks on such targets are less likely, decision makers can consider the predictions of our models along with other intelligence they might have

Table 6. Strong and Weak Summary Sets for the LeT Dataset—Consequent = LeT Does Not Carry out Attacks against Symbolic Sites 2 Months Later

Weak Summary Set 4		
Antecedent	Consequent	coverage
0-11 released *	no SS attacks ** 2 months later	73%
LeT was not involved in any intra-organizational conflict regarding its resources	no SS attacks ** 2 months later	27%

Strong Summary Set 4			
Antecedent		Consequent	coverage
0-11 released *	LeT may have splintered into new groups	no SS attacks ** 2 months later	2%
0-11 released *	It is possible that splintering within LeT occurred	no SS attacks ** 2 months later	2%
0-11 released *	Unknown reasons may have caused splintering within LeT	no SS attacks ** 2 months later	2%
0-11 released *	Disagreement over the leadership may have caused splintering within LeT	no SS attacks ** 2 months later	2%
0-11 released *	Disagreement over the instruments used to achieve a goal may have caused splintering within LeT	no SS attacks ** 2 months later	2%
0-11 released *	Disagreement over its principles (ideology, belief) may have caused splintering within LeT	no SS attacks ** 2 months later	2%
0-11 released *	Changes in the environment may have caused splintering within LeT	no SS attacks ** 2 months later	2%
0-11 released *	LeT did not dissolve after a split	no SS attacks ** 2 months later	2%
0-11 released *	There were 0-12 raids on LeT	no SS attacks ** 2 months later	2%
0-11 released *	There were 0-2 raids on LeT	no SS attacks ** 2 months later	2%
0-11 released *	LeT may have endorsed its splinters	no SS attacks ** 2 months later	2%
0-11 released *	LeT personnel were killed by the government	no SS attacks ** 2 months later	2%
0-11 released *	0-1 asset of LeT or its leaders were frozen by the Pakistani government	no SS attacks ** 2 months later	2%
0-11 released *	There was a (partial) government ban	no SS attacks ** 2 months later	2%

*0-11 arrested LeT members were released by the government.

**LeT does not carry out attacks against symbolic sites.

Table 7. Strong and Weak Summary Sets for the LeT Dataset—Consequents = LeT Carries out 1 Attack against Professional Security Forces x Months Later

Weak Summary Set 5		
Antecedent	Consequent	coverage
LeT received military support from the Pakistani government	1 PSF attack * 1 month later	70%
Let was pursuing the elimination of external influence	1 PSF attack * 1 month later	5%
LeT was waging a communications campaign using news/periodicals to get their message out	1 PSF attack * 2 months later	5%

Strong Summary Set 5			
Antecedent		Consequent	coverage
LeT was waging a communications campaign using news or periodicals to get their message out	0-2 LeT members were arrested by the government	1 PSF attack * 1 month later	3%
Let was pursuing the elimination of external influence	LeT maintained three websites	1 PSF attack * 1 month later	3%
The group was reportedly based in more than one country		1 PSF attack * 1 month later	2%

*LeT carries out 1 attack against professional security forces.

Table 8. Strong and Weak Summary Sets for the LeT Dataset—Consequents = LeT Does Not Carry Attacks against Symbolic and Tourist Sites x Months Later

Weak Summary Set 6		
Antecedent	Consequent	coverage
No dissolve *	No S/T attacks ** 3 months later	87%
It is possible that LeT made territorial claims	No S/T attacks ** 3 months later	6%
LeT was not involved in any intra-organizational conflict regarding its resources	No S/T attacks ** 3 months later	4%
0-11 arrested LeT members were released by the government	No S/T attacks ** 1 month later	3%

Strong Summary Set 6			
Antecedent		Consequent	coverage
No dissolve *	It is possible that some LeT commanders resigned	No S/T attacks ** 3 months later	2%
No dissolve *	LeT may have been involved in intra-organizational conflicts regarding its resources	No S/T attacks ** 3 months later	2%
No dissolve *	It is unknown if LeT was involved in intra-organizational conflicts regarding its resources	No S/T attacks ** 3 months later	2%
No dissolve *	It is possible that some LeT commanders were arrested	No S/T attacks ** 3 months later	2%
No dissolve *	0-3 LeT personnel were arrested	No S/T attacks ** 3 months later	2%
No dissolve *	It is possible that some LeT commanders died	No S/T attacks ** 3 months later	2%
No dissolve *	LeT was involved in 0-3 intra-organizational conflicts regarding its leadership	No S/T attacks ** 3 months later	2%
No dissolve *	0-7 LeT commanders died	No S/T attacks ** 3 months later	2%
No dissolve *	0-4 LeT commanders resigned	No S/T attacks ** 3 months later	2%
No dissolve *	Changes in the environment may have caused splintering within LeT	No S/T attacks ** 3 months later	2%

*LeT's basic organization did not dissolve.

**LeT does not carry attacks against symbolic and tourist sites.

to downgrade threat levels appropriately. This can potentially enable security officials to devote protective resources to other threats that might be more likely during the time frame in question.

8 RELATED WORK

Time has been incorporated in association rules in different works in different ways.

References [4, 7, 15, 16, 18, 30] start from the observation that some association rules may hold during some time intervals but not during others, and thus propose *temporal association rules*, which consist of standard association rules along with time intervals where they hold. As an example, a temporal association rule is of the form $(X \rightarrow Y)^{2,4}$, meaning that the *classical* association rule $X \rightarrow Y$ has support and confidence above user-defined thresholds only in the time interval [2, 4]. Support, confidence, and other metrics are computed w.r.t. the so-called *exhibition periods* of items, that is, the time range from when an item first occurs to when it no longer occurs. We point out that such kind of rules cannot express relationships across different itemsets in time, as TARs do. An incremental algorithm for computing temporal association rules in evolving temporal databases has been proposed in Reference [9].

Reference [26] proposed to use TARs like the ones considered in this article to discover the toxicity progression of cancer treatments. [26] developed an efficient method for mining TARs, where a lattice structure is constructed to store all frequent diagnosis codes, and pruning techniques are used to reduce the search space. Their focus is on efficiently mining TARs, while in this article our goal is to *summarize* a set of TARs.

Reference [23] considers the problem of mining so-called *episodes* from a sequence of time-stamped events. An episode specifies a partial order among events, which is the order events are expected to occur. As opposed to TARs, no quantitative information on the time delay can be specified. For instance, an episode can say that an event B is likely to occur after an event A occurs, but it cannot state after how many time points, which can be expressed with a TAR $\tau : A \rightarrow B$, where τ is the desired time delay.

Multidimensional inter-transaction association rules have been proposed in Reference [21]. These rules can represent not only the associations of items within transactions, but also associations of items among different transactions. A model for mining patterns in the presence of evolving numerical attributes has been proposed in Reference [29].

Reference [8] considered rules of the form *if X occurs, then Y occurs within time τ* , which are akin to timed association rules considered in this article. The difference is that a TAR expresses a rule of the form: If X occurs, then Y occurs *after exactly* time points τ . Thus, the rules of Reference [8] cannot model timed association rules, and vice versa. To the best of our knowledge, no approach has been proposed to summarize the kind of rules introduced in Reference [8]. However, we point out that our approach can be easily adapted to deal with their setting. There is only one notion that needs to be redefined: when a rule represents another one—this has to be done in a different way, as the rules in Reference [8] have a different meaning from ours. As an example, a natural definition is to state that a rule $\tau : X \rightarrow Y$ (recall that, in the approach of Reference [8], this rule means that if X occurs, then Y occurs *within* time τ) is more general than any rule of the form $\tau' : X \rightarrow Y$ with $\tau' < \tau$, and thus the former represents the latter. With a proper definition of when a rule represents another one, the strong and weak summary set problems can be defined in the same way as done in this article. Furthermore, even our STAR-S and STAR-W algorithms can be directly applied by plugging into them the different notion of a rule representing another one.

The problem of summarizing (non-temporal) association rules has been addressed in several works [1, 2, 17, 19, 20, 22, 31].

While there has been work on summarizing standard association rules, and work on combining the time dimension with association rules, to the best of our knowledge, this article is the first effort to summarize timed association rules.

9 CONCLUSION AND FUTURE WORK

We have addressed the problem of summarizing TARs for the first time. Specifically, we have considered two variants of the problem: the strong version, where the summary has to be a subset of the original set of TARs, and the weak version, where representative TARs might not belong to the original set of TARs. We have characterized the complexity of both problems and proposed polynomial-time approximation algorithms to solve them. We have experimentally validated our algorithms over data from the World Bank and on terrorist group activities, showing that they are able to summarize real-world TARs in a very effective manner. Both summarization methods have advantages: while both coverage scores and human preference are in favor of weak summaries, strong summaries can be computed in a very short period of time. Hence, we suggest using strong summaries when computation time is very important, but suggest that weak summaries should be used when timed association rules are recomputed periodically (e.g., once a month or less frequently).

As a direction for future work, time intervals might be used as a way of succinctly representing multiple “similar” TARs with time points falling in an interval. Furthermore, TARs might be extended to express that the consequent will likely occur with a time delay falling in a *time range*, rather than specifying a single time point. Another interesting extension is to consider taxonomies on items so that, for instance, two TARs $1 : X \rightarrow Z$ and $1 : Y \rightarrow Z$ might be summarized

by $1 : W \rightarrow Z$ if W is a generalization of X and Y according to a given taxonomy. Pruning redundant TARs during the mining process is another challenging issue we plan to explore.

ACKNOWLEDGMENTS

We thank the anonymous referees for many helpful comments.

REFERENCES

- [1] Foto N. Afrati, Aristides Gionis, and Heikki Mannila. 2004. Approximating a collection of frequent sets. In *KDD*. 12–19.
- [2] Charu C. Aggarwal and Philip S. Yu. 1998. Online generation of association rules. In *ICDE*. 402–411.
- [3] Rakesh Agrawal, Tomasz Imielinski, and Arun N. Swami. 1993. Mining association rules between sets of items in large databases. In *SIGMOD*. 207–216.
- [4] Juan M. Ale and Gustavo Rossi. 2000. An approach to discovering temporal association rules. In *SAC*. 294–300.
- [5] Mohamed-Bachir Belaid, Christian Bessiere, and Nadjib Lazaar. 2019. Constraint programming for association rules. In *SDM*. 127–135.
- [6] Abdelhamid Boudane, Saïd Jabbour, Lakhdar Sais, and Yakoub Salhi. 2017. Enumerating non-redundant association rules using satisfiability. In *PAKDD*. 824–836.
- [7] Cheng-Yue Chang, Ming-Syan Chen, and Chang-Hung Lee. 2002. Mining general temporal association rules for items with different exhibition periods. In *ICDM*. 59–66.
- [8] Gautam Das, King-Ip Lin, Heikki Mannila, Gopal Renganathan, and Padhraic Smyth. 1998. Rule discovery from time series. In *KDD*. 16–22.
- [9] Tarek F. Gharib, Hamed Nassar, Mohamed Taha, and Ajith Abraham. 2010. An efficient algorithm for incremental mining of temporal association rules. *Data Knowl. Eng.* 69, 8 (2010), 800–815.
- [10] Robert Golan and Donald Edwards. 1994. Temporal rules discovery using datalogic/R+ with stock market data. In *Rough Sets, Fuzzy Sets and Knowledge Discovery*. Springer, 74–81.
- [11] Wilhelmiina Hämmäläinen and Geoffrey I. Webb. 2017. Specious rules: An efficient and effective unifying method for removing misleading and uninformative patterns in association rule mining. In *SDM*. 309–317.
- [12] Jiawei Han, Jian Pei, and Yiwen Yin. 2000. Mining frequent patterns without candidate generation. *SIGMOD Rec.* 29, 2 (2000), 1–12.
- [13] Wei Han, Julio De Melo Borges, Peter Neumayer, Yong Ding, Till Riedel, and Michael Beigl. 2017. Interestingness classification of association rules for master data. In *ICDM*. 237–245.
- [14] Roberto J. Bayardo Jr., Rakesh Agrawal, and Dimitrios Gunopulos. 1999. Constraint-based rule mining in large, dense databases. In *ICDE*. 188–197.
- [15] Chang-Hung Lee, Ming-Syan Chen, and Cheng-Ru Lin. 2003. Progressive partition miner: An efficient algorithm for mining general temporal association rules. *IEEE Trans. Knowl. Data Eng.* 15, 4 (2003), 1004–1017.
- [16] Chang-Hung Lee, Cheng-Ru Lin, and Ming-Syan Chen. 2001. On mining general temporal association rules in a publication database. In *ICDM*. 337–344.
- [17] Brian Lent, Arun N. Swami, and Jennifer Widom. 1997. Clustering association rules. In *ICDE*. 220–231.
- [18] Yingjiu Li, Peng Ning, Xiaoyang Sean Wang, and Sushil Jajodia. 2003. Discovering calendar-based temporal association rules. *Data Knowl. Eng.* 44, 2 (2003), 193–218.
- [19] Bing Liu, Wynne Hsu, and Yiming Ma. 1999. Pruning and summarizing the discovered associations. In *KDD*. 125–134.
- [20] Guimei Liu, Haojun Zhang, and Limsoon Wong. 2014. A flexible approach to finding representative pattern sets. *IEEE Trans. Knowl. Data Eng.* 26, 7 (2014), 1562–1574.
- [21] Hongjun Lu, Ling Feng, and Jiawei Han. 2000. Beyond intratransaction association analysis: Mining multidimensional intertransaction association rules. *ACM Trans. Inf. Syst.* 18, 4 (2000), 423–454.
- [22] Michael Mampaey, Nikolaj Tatti, and Jilles Vreeken. 2011. Tell me what i need to know: Succinctly summarizing data with itemsets. In *KDD*. 573–581.
- [23] Heikki Mannila, Hannu Toivonen, and A. Inkeri Verkamo. 1997. Discovery of frequent episodes in event sequences. *Data Min. Knowl. Discov.* 1, 3 (1997), 259–289.
- [24] Francisco Javier Martínez-de Pisón, Andrés Sanz, Eduardo Martínez-de Pisón, Emilio Jiménez, and Dante Conti. 2012. Mining association rules from time series to explain failures in a hot-dip galvanizing steel line. *Comput. Industr. Eng.* 63, 1 (2012), 22–36.
- [25] George L. Nemhauser, Laurence A. Wolsey, and Marshall L. Fisher. 1978. An analysis of approximations for maximizing submodular set functions—I. *Math. Program.* 14, 1 (1978), 265–294.
- [26] Dang Nguyen, Wei Luo, Dinh Q. Phung, and Svetha Venkatesh. 2018. LTARM: A novel temporal association rule mining method to understand toxicities in a routine cancer treatment. *Knowl. Based Syst.* 161 (2018), 313–328.

- [27] Iqbal H. Sarker and Flora D. Salim. 2018. Mining user behavioral rules from smartphone data through association analysis. In *PAKDD*. 450–461.
- [28] V. S. Subrahmanian, Aaron Mannes, Amy Sliva, Jana Shakarian, and John P. Dickerson. 2012. *Computational Analysis of Terrorist Groups: Lashkar-e-Taiba*. Springer.
- [29] Wei Wang, Jiong Yang, and Richard R. Muntz. 2001. TAR: Temporal association rules on evolving numerical attributes. In *ICDE*. 283–292.
- [30] Edi Winarko and John F. Roddick. 2007. ARMADA—An algorithm for discovering richer relative temporal association rules from interval-based data. *Data Knowl. Eng.* 63, 1 (2007), 76–90.
- [31] Dong Xin, Jiawei Han, Xifeng Yan, and Hong Cheng. 2005. Mining compressed frequent-pattern sets. In *VLDB*. 709–720.
- [32] An-Shu Zhang, Wenzhong Shi, and Geoffrey I. Webb. 2016. Mining significant association rules from uncertain data. *Data Min. Knowl. Discov.* 30, 4 (2016), 928–963.

Received November 2019; revised July 2020; accepted August 2020