On the Discovery of Time Distance Constrained Temporal Association Rules

Heitor Murilo Gomes[1(✉)], Deborah Ribeiro de Carvalho[2], Lourdes Zubieta[3], Jean Paul Barddal[1], and Andreia Malucelli[1]

[1] Programa de Pós-Graduação em Informática, Pontifícia Universidade Católica do Paraná, Curitiba, Brazil
hmgomes@ppgia.pucpr.br
[2] Programa de Pós-Graduação em Tecnologia Aplicada em Saúde, Pontifícia Universidade Católica do Paraná, Curitiba, Brazil
[3] Williams School of Business, Bishop's University, Sherbrooke, QC, Canada

Abstract. The increased use of data mining algorithms reflects the need for automatic extraction of knowledge from large volumes of data. This work presents a temporal data mining algorithm that discovers frequent Association Rules from timestamped data. These rules are named Cause-Effect Rules, each represented by a multiset of unordered events (Cause) followed by a singleton event (Effect). Also, a Cause-Effect Rule is valid within an specific constraint that defines the minimum and maximum time distance between its Cause and Effect. Our algorithm was tested on a data set from two hospital emergency departments in Sherbrooke, QC, Canada.

1 Introduction

The increased use of data mining algorithms reflects the need for automatic extraction of knowledge from large volumes of data. In many contexts, data contains timestamps or other sequential index. To extract knowledge from this type of data, the temporal data mining research area was established. Temporal data mining can be defined as the efficient discovery of frequent sequential patterns. Identifying such patterns provides the ability to predict a future event, with certain confidence, if a set of events is identified. Its applications cover different areas of knowledge ranging from banking and telecommunications to web browsing and adverse drug reactions. The data being processed in each of these areas varies greatly with regard to the number of examples, attributes, noise level, and others. Temporal data mining algorithms are designed to cope with specific requirements from the domain in which they are applied. For example, in healthcare analysis there is a vast amount of data arranged as transactions of events with timestamps. Specialists in healthcare analysis often want to find associations between events in patients' medical history over time in order to make evidence-based decisions. Informative patterns can be extracted to identify, for example, that "10 days before being hospitalized patients frequently

S. Arik et al. (Eds.): ICONIP 2015, Part II, LNCS 9490, pp. 510–519, 2015.
DOI: 10.1007/978-3-319-26535-3_58

took test A and B" rather than "hospitalized patients frequently took test A and B".

Sequential data mining algorithms attempt to find frequent occurrences of itemsets in a specific order. Introduced by [1], the itemset is a non-empty set of items, while a sequence [2] is an ordered non-empty list of itemsets. Previous works [4,7,8] identify sequential patterns that appear more often than a user-specified minimum support while maintaining their item occurrence order. Time intervals are commonly added by defining time distance constraints, limiting the timespan between events. More recently, there has been increasing interest in extracting Association Rules from temporal data [3].

In this paper we present a new temporal data mining algorithm, namely Cause-Effect Rules (CER). CER extends and combines definitions found in previous works to generate insightful, yet simple to understand patterns. CER generates Association Rules based on the assumption that an event (effect or consequent) E might have been caused by a set of other events (cause set or antecedent) C if, and only if, all events in C occurred before E within a timespan δ. In other words, not every event that happened before E could have influenced it. For example, if C is composed of patients' medical history and we are looking for a specific event like being hospitalized, it is likely that events that occurred years before the hospital admission may not have influenced E as much as events that happened a few days or weeks before. Obtaining rules in a similar format, although from sequential data, has been explored in previous works, like the MARBLES algorithm [3]. However, combining it with the well-known technique [6] of time distance constraint has not yet been explored in the literature and we show that it is very useful for analysing timestamped data, such as emergency department visits' data.

The rest of this paper is organized as follows. Section 2 briefly surveys temporal data mining methods. Section 3 formally describes the problem of discovering CERs, while Sect. 4 outlines CER implementation. In Sect. 5, we present the results and discuss around the experiments conducted on an Emergency Department dataset. Finally, Sect. 6 concludes the paper.

2 Related Work

In [2] the problem of mining sequential patterns was introduced along with three algorithms to solve it. Despite the difference between the three approaches, they aim for the same goal: discovering frequent sequences. A variation of rule support [1] was used to quantitatively assess the discovered sequences. This support measure reflects the fraction of transactions in which a sequence occurs. At a later date the same authors of [2] presented the GSP algorithm in [7], which performed better than the previous algorithms and permitted the user to specify constrains to the discovered sequences, such as the minimum and maximum gap between adjacent itemsets.

In [8] the SPADE algorithm for discovering frequent sequences was presented. It diverges from previous work as it does not inherit from Association Rules

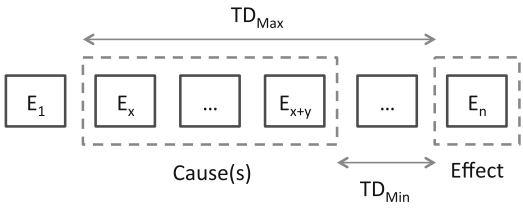

Fig. 1. Cause-Effect Sequence derived from a raw sequence of events.

discovery techniques, i.e., it does not explore the anti-monotonicity property, instead the authors use a Lattice-based approach along with efficient search methods to discover and count the occurrence of sequences.

In [4], authors present Chrono Assoc, a hybrid data mining method, designed to combine knowledge from an Association Rules discovery [1] process along with chronological order evaluation. The algorithm receives as input Association Rules and the dataset used to generate them, incrementing chronological support and confidence for each rule based on events that are consistent with the hypothesis "every antecedent must have happened before consequent".

The MARBLES [3] algorithm is capable of uncovering Association Rules between general episodes. A general episode is formed by a parallel episode that precede a sequential episode, i.e. an unordered set of events that occur before a set of strictly ordered events. Works before MARBLES used to focus on either sequential or parallel episodes [6], yet it is intuitive that rules combining both of them can be useful in many problems domains.

Even though these algorithms can uncover meaningful knowledge they do not fully explore important information included in some transactional databases. We claim that the next step beyond the informative rules obtained by an algorithm such as MARBLES, is the exploitation of time distance constraints according to the problem domain.

3 Cause-Effect Rules

A Cause-Effect Sequence (CES) is defined as a triplet (C, E, δ) where C is a cause multiset[1] of events, E is a singleton effect event, and $\delta = (TD_{Min}, TD_{Max})$ is a Time Distance Constraint (TDC) over which the sequence is valid. It is assumed that events in C happened close to each other in time, thus their order is deemed as irrelevant. δ delimits the minimum (TD_{Min}) and maximum (TD_{Max}) time distance between C and E. Concretely, δ is the building template for CES, following the intuitive notion that causes may have influenced a later effect if, and only if, they have happened neither too close, nor too far apart in time. Figure 1 presents how a CES is built given a raw event sequence.

[1] The multiset is a flexible representation that permits the Cause to contain repeated events.

The complete set of CES C_δ of the input database is a different representation of the original data restricted to a specific TDC δ. Therefore, not all CES represent frequent patterns, they are only a representation of the original data. A Cause-Effect Rule (CER) represents a CES that satisfies the user-given thresholds: Minimum Effect Frequency (E_{min}), Minimum Support S_{min}, and Minimum Confidence C_{min}. E_{min} is meant to filter out any candidate CER which effect is not frequent, while S_{min} and C_{min} are more specific to the rule frequency as a whole, and related to our Support and Confidence definitions (see Eqs. 2 and 3).

Ultimately, one CER is a CES that either occur or are contained in "a sufficiently large number of CES". A CES x is said to be contained within CES y if, and only if, they share the same E and δ, and all events in x cause appear in y cause. To assess the obtained CERs we use three measures. The first one is the Effect Frequency $E_f(e, \delta)$ (Eq. 1), which represents the ratio of occurrence of an effect for a given TDC δ.

$$E_f(e, \delta) = \frac{|\{x : x \in C_\delta, \text{Effect}(x) = e\}|}{|C_\delta|} \tag{1}$$

The other two measures are inspired by the Support and Confidence definition are based on those presented originally on [1]. Although, these measures must be defined differently for CER. The Cause-Effect Support (S_{CE}) outlines the ratio of CES that are consistent with a given CER (Eq. 2). In other words, it is the ratio of occurrence of a given CER according to all C_δ.

$$S_{CE}(r, \delta) = \frac{|\{x : x \in C_\delta, r \subseteq x\}|}{|C_\delta|} \tag{2}$$

Cause-Effect Confidence (C_{CE}) represents the ratio of CES that are consistent with a given CER, limited to those that have the same Effect as that CER (Eq. 3).

$$C_{CE}(r, \delta) = \frac{|\{x : x \in C_\delta, r \subseteq x\}|}{|\{x : x \in C_\delta, \text{Effect}(x) = \text{Effect}(r)\}|} \tag{3}$$

Other works have been developed to uncover frequent sequences; in CER the only sense of order that matters is that all events in C must have happened at a constrained distance from the event in E. Nevertheless, we do treat differently the same event happening multiple times as this can yield different rules. For example, a situation in which a patient has visited the emergence department with the same problem four times in a row is different than a situation with a single visit.

4 CER 1.0 ALGORITHM

The proposed algorithm for Cause-Effect Rules mining, namely CER 1.0, performs multiple passes over a transformed version of the original database. This new database is represented in terms of Cause-Effect Sequences. A key component of the counting subroutine is the data structure, which represents the

CER candidates. We use an n-ary tree, which accommodates all the candidates associated with the same effect for a TDC. The algorithm is divided into three phases: Transformation, Initialization and Discovery, detailed in the following sections.

4.1 Transformation Phase

Some preparation steps are performed in the first pass over the database, including mapping the events description from string to integer representation. The most important and time-consuming step is the transformation of original input database to CESs. The CES form can be used to directly match CER candidates. It is straightforward to count the occurrences of individual events, e.g., causes and effects, during this phase, since the whole database is inspected. By doing that, the first candidates generation can be restricted to those that can actually yield a CER. At the end of this phase, the input dataset is transformed from a raw event data into a collection of CES, according to the time constraint chosen. The effect counter is stored for later use during the Effect Frequency calculation explained next.

4.2 Initialization Phase

In the Initialization phase, the first generation of candidate trees are created, such that for each combination of event e, and time constraint δ the corresponding candidate trees $M_\delta = \mu_1, \mu_2, \cdots, \mu_e$ are created. This phase also includes the calculation of E_f and the removal of CES whose E_f is less than E_{min}, shrinking the scope of the mining process. The counters used to calculate E_f are updated during the Transformation phase (see Sect. 4.1). Therefore, it is only necessary to iterate over every effect and calculate their frequency.

The initialization of the candidate trees requires combining frequent effects and causes that withstand filtering. If none were filtered out, the number of candidates of the first generation grows as much as the number of occurrences of its effect $N_e \times N_\delta$, although this is rare in practice. The number of trees is exactly the same as the number of Frequent Effects, since each tree represents candidates that share the same effect. The assertion behind which candidates can yield a CER is based on the anti-monotonicity (Apriori) property, which states that one set can only be frequent if each of its subsets is also frequent. Another interpretation, focused on the current work, is that a CES can only become a CER if its Effect and its Cause set are frequent. Besides, one Cause set can only be frequent if each of its subsets are also frequent.

4.3 Discovery Phase

The last phase incrementally expands the candidate trees and filters out candidates. There are two levels of filtering. The first filter is the Cause-Effect Support (S_{CE}) followed by the Cause-Effect Confidence (C_{CE}), both set by the user as

Table 1. Number of CERs found for different cause set sizes ($|C|$).

TD_{min}	TD_{max}	$\|C\| = 1$		$\|C\| = 2$		$\|C\| = 3$		$\|C\| = 4$	
		#	%	#	%	#	%	#	%
1 s	1h	4	4.3	4	1.9	0	0	0	0
1h	1 day	17	18.1	29	13.6	5	7.6	1	5
1h	2 days	18	19.1	44	20.7	11	16.7	1	5
1h	3 days	19	20.2	52	24.4	20	30.3	8	40
1h	4 days	19	20.2	55	25.8	25	37.9	9	45

Table 2. CERs with effect = 24 (LWBS).

Frequent CERs	S_{CE} (%)	C_{CE} (%)
24 (1 h, 1 day, 24)	5.00	52.15
24 (1 h, 2 days, 24)	5.11	45.42
24 (1 h, 3 days, 24)	4.74	40.99
24 (1 h, 4 days, 24)	4.58	39.35
17 (1 h, 1 day, 24)	0.92	09.57
17 (1 h, 3 days, 24)	1.23	10.62
5 (1 h, 2 day, 24)	1.10	09.76
5 (1 h, 3 day, 24)	1.23	10.62
5 (1 h, 4 day, 24)	1.43	12.26

the minimum support (S_{min}) and Minimum Confidence (C_{min}), respectively. In order to calculate a candidate's S_{CE} and C_{CE} it is necessary to count how many CES contains it. A given CES is said to contain a candidate CER if their effects match and if every member of the candidate's cause multiset is present in the CES cause multiset. For example, consider candidate CER $x = (\{A, B, C\}, D, \delta_z)$ and CES $y = (\{B, C, D, A\}, D, \delta_z)$, then y contains x. The candidates' cause sets are represented recursively in the candidate trees, as follows:

Nodes. Each node contains a list of event identifiers. If it is a parent node, each of its events may or may not have a child node.

Leaves. Every event on the leaves list is the last cause of a candidate; thus the size of all leaves lists combined corresponds exactly to the number of candidates that exists for a tree.

Height. A tree with height h represents candidates up to generation $h + 1$, e.g., trees with height equals two have candidates on the third candidate (Gen_3) generation.

New generations, except the first one (Gen_1), are created by adding a level on all existing trees. The last cause of a candidate, i.e., the event at the leaf, will receive a child node if, and only if, the candidate has surpassed S_{min} and C_{min} thresholds. This effectively eliminates candidates that cannot become CER. Trees that do not have nodes on the current generation level are removed. For every generation, the set of CES is compared to each tree according to the effect and TDC that the tree represents. The comparison of a CES and a candidate tree starts at the root and advances to lower levels according to the CES events in the cause set. If a leaf node is reached, then a counter is incremented indicating that the CER candidate, represented by the inverse traverse of the tree (leaf to root), support has been increased.

5 Experiments

We tested the CER 1.0 algorithm on a dataset from two hospital emergency departments (ED) in Sherbrooke, QC. Canada. The dataset contained 138,107 low acuity visits to the EDs during a period of 3.5 years, from June 2006 to December 2009. Besides the visit's date and arrival time, each record include a 40-character patient identifier that allows to follow the person over time; other visit details include age, sex, postal code, diagnostic code(s) as written by a doctor, and whether the patient was admitted to the hospital. We were particularly interested in uncompleted visits because the patient left without being seen by a doctor (LWBS). These LWBS visits receive a particular code (24) and represent 13.9 % of all visits in the dataset. The rate of LWBS has been used as an indicator of quality of care, although there is controversy about its use and interpretation. LWBS behaviour has been associated with very long waiting times at the ED, lack of access to medical care, and higher risks of short term adverse events [5]. A sequence was defined as a list of visits by the same patient. The TDCs used to mine the dataset were small time intervals: between 1 s and 1 h, between 1 h and 1 day, 1 h and 2 days, 1 h and 3 days, and 1 h and 4 days, for $|C| \in [1; 5]$.

5.1 Results Before a LWBS Visit

In the first set of experiments we find out which diagnostic codes patients received previously to a LWBS visit; the "causes" are the previous diagnostic codes and the "effect" is an incomplete visit (code 24). We found many CERs having effect = 24 (LWBS), but not many with a high support. For each time interval, code 24 came out as the second or third most frequent one. Other frequent CERs had effect = 17 (Infections and parasitic diseases), 5 (Mental illness) and 16 (Ill-defined conditions). These are also the most frequent codes in the original database [9]. The most frequent rules for effect = 24 are shown in Table 2. For each TDC, the most common event prior to a LWBS visit is another LWBS visit, so patients who left the ED are more likely to come back in a very near future and leave again. The rules with the highest support were $24 \rightarrow 24$, with near 5 % support but with high confidence values (C_{CE}) ranging from 52 % for a repeat behaviour within a day, down to 39.35 % within 4 days. The originality of our results is the fact that patients come back and leave again within a short period of time, indicating their healthcare needs were unfulfilled.

5.2 Results Before a Hospital Admission via ED

In the second set of experiments we find out whether LWBS patients were hospitalized after an incomplete visit to the ED; the "effect" is being hospitalized and the "causes" are any previous diagnostic codes received by the patients. As we do not have data about all hospitalizations that occurred during the time intervals studied (only those admissions from the ED), we must be careful in the interpretation of the results.

Table 3. CERs of size 1 with effect = hospitalization.

TD_{min}	TD_{max}	Cause	Effect	E_F (%)	S_{CE} (%)	C_{CE} (%)
1sec	1h	{24}	Hosp.	5.75	2.30	40.00
1sec	1 day	{17}	Hosp.	7.72	1.72	22.22
1sec	1 day	{16}	Hosp.	7.72	1.39	17.99
1sec	1 day	{24}	Hosp.	7.72	0.86	11.11
1 h	1 day	{17}	Hosp.	7.79	1.78	22.83
1 h	1 day	{16}	Hosp.	7.79	1.44	18.48
1 h	1 day	{24}	Hosp.	7.79	0.80	10.33
1 h	2 days	{17}	Hosp.	6.99	1.56	22.37
1 h	2 days	{16}	Hosp.	6.99	1.34	19.18
1 h	2 days	{9}	Hosp.	6.99	0.70	10.05
1 h	2 days	{24}	Hosp.	6.99	0.67	09.59
1 h	3 days	{17}	Hosp.	6.93	1.41	20.31
1 h	3 days	{16}	Hosp.	6.93	1.30	18.77
1 h	3 days	{8}	Hosp.	6.93	0.82	11.88
1 h	3 days	{Hosp.}	Hosp.	6.93	0.82	11.88
1 h	3 days	{9}	Hosp.	6.93	0.74	10.73
1 h	3 days	{24}	Hosp.	6.93	0.69	09.96
1 h	4 days	{17}	Hosp.	6.51	1.26	19.42
1 h	4 days	{16}	Hosp.	6.51	1.15	17.63
1 h	4 days	{Hosp.}	Hosp.	6.51	0.87	13.31
1 h	4 days	{8}	Hosp.	6.51	0.75	11.51
1 h	4 days	{9}	Hosp.	6.51	0.70	10.79
1 h	4 days	{24}	Hosp.	6.51	0.63	09.71

Table 4. CERs with effect = LWBS visit (24).

TD_{min}	TD_{max}	Cause	Effect	E_f %	S_{CE} %	C_{CE} %
1 s	1h	{24}	24	09.20	4.60	50.00
1 s	1h	{5, 5}	24	09.20	1.15	12.50
1 s	1h	{16, 9, 5}	24	08.86	0.04	00.46
1h	1 day	{24}	24	08.84	4.61	52.15
1h	1 day	{5, 5}	24	08.84	0.17	01.91
1h	1 day	{16, 9, 5}	24	08.84	0.04	00.48
1h	2 days	{24}	24	10.46	4.75	45.43
1h	2 days	{5, 5}	24	10.46	0.38	03.66
1h	2 days	{16, 9, 5}	24	10.46	0.03	00.30
1h	3 days	{24}	24	10.76	4.41	40.99
1h	3 days	{5, 5}	24	10.76	0.45	04.20
1h	3 days	{16, 9, 5}	24	10.76	0.03	00.25
1h	4 days	{24}	24	10.89	4.28	39.35
1h	4 days	{5, 5}	24	10.89	0.56	05.16
1h	4 days	{16, 9, 5}	24	10.89	0.02	00.22

When considering only the most recent previous visit, we found LWBS was the most frequent "cause" only in the very short interval [1 s, 1 h]. For other periods of time and with 2 or more previous visits, LWBS was not the most frequent code. Other diagnostic codes like ill-defined conditions (16), diseases of the digestive system (9), respiratory problems (8) and a previous hospitalization were found more frequently than a LWBS visit, but we noticed the low frequence values E_f of all rules. Table 3 shows the detailed results for $|C| = 1$ only.

5.3 Discussion

Experiments conducted using the CER 1.0 algorithm revealed for the first time according to the authors, that patients who leave the emergency department before seeing a doctor (LWBS) are more likely to come back in a short period of time ranging up to 4 days, and leave again before seeing a doctor. This behaviour was found more frequently in men than in women. We also found that other common diagnostic codes previous to an LWBS visit are infections and parasitic diseases and mental illnesses, indicating that the healthcare needs of these patients were not fulfilled. A third result of our experiments showed no major risk of hospitalization after an incomplete LWBS visit. However, our study is not conclusive because of two reasons: (i) we do not know if these patients came back to the ED but were assigned a more urgent triage code, or (ii) they were admitted to the hospital shortly after an LWBS visit by other pathway than the ED.

6 Conclusion

In this paper we presented a new data mining algorithm for finding temporal rules constrained by user-defined Time Distance Constraints. To illustrate our method, we applied the algorithm to a dataset of low acuity emergency department visits from two hospitals in Sherbrooke, QC, Canada. Our results provide information on specific diagnosis associated with a higher likelihood of bouncing back and contributed to assess the risks of short term adverse effects like hospitalization. The results are noteworthy as the same rule structures held for different TDCs. Future work includes automatic discovery of TDCs, since these require domain knowledge and may be difficult to determine, and generating rules that combine static attributes and temporal events.

References

1. Agrawal, R., Srikant, R.: Fast algorithms for mining association rules. Proc. VLDB **1215**, 487–499 (1994)
2. Agrawal, R., Srikant, R.: Mining sequential patterns. In: Proceedings of the 11th ICDE, pp. 3–14. IEEE (1995)
3. Cule, B., Tatti, N., Goethals, B.: Marbles: mining association rules buried in long event sequences. Stat. Anal. DM: ASA Data Sci. J. **7**(2), 93–110 (2014)
4. Gomes, H.M., Carvalho, D.R.: A hybrid data mining method: exploring sequential indicators over association rules. Iberoamerican J. Appl. Comput. **1**(1), 40–60 (2011)
5. Lucas, J., Batt, R.J., Soremekun, O.A.: Setting wait times to achieve targeted left-without-being-seen rates. Am. J. Emerg. Med. **32**(4), 342–345 (2014)
6. Mannila, H., Toivonen, H., Inkeri Verkamo, A.: Discovery of frequent episodes in event sequences. Data Min. Knowl. Discov. **1**(3), 259–289 (1997)
7. Srikant, R., Agrawal, R.: Mining sequential patterns: generalizations and performance improvements. In: Apers, P.M.G., Bouzeghoub, M., Gardarin, G. (eds.) EDBT 1996. LNCS, vol. 1057, pp. 1–17. Springer, Heidelberg (1996)

8. Zaki, M.J.: Spade: an efficient algorithm for mining frequent sequences. Mach. Learn. **42**(1–2), 31–60 (2001)
9. Zubieta, L., Fernández-Peña, J.R.: A retrospective study on emergency visits in two hospitals in sherbrooke, Canada. J. Hosp. Adm. **3**(4), p61 (2014)