An Inductive Database for Mining Temporal Patterns in Event Sequences

Alexandre Vautier, Marie-Odile Cordier and René Quiniou
Irisa - DREAM Project Campus de Beaulieu 35042 RENNES Cedex, France
{Alexandre.Vautier,Marie-Odile.Cordier,Rene.Quiniou}@irisa.fr

1 Introduction

Data mining aims at discovering previously unknown and potentially useful information from large collections of data. The discovered knowledge, in the form of *patterns*, is extracted without any extra information about data. A pattern is a representation of a concept (rules describing properties of data, clusters in databases, etc.) that generalizes some data in a database.

Imielinski and Mannila [Imielinski and Mannila, 1996] designed a data mining formalization framework named inductive database (IDB). The main goal of IDBs is to manage knowledge discovery applications just as database management systems successfully manage business applications. IDBs contain patterns in addition to data. The data mining process is modeled as an interactive process in which users can query data as well as patterns. IDBs represent also a good way to take advantage of the efficiency of database algorithms by using jointly data mining and database algorithms.

Some IDBs implementations [Lee and De Raedt, 2003] use sequences of symbols as data and complex sequences of symbols as patterns (sequences with gaps). In many problems one often needs to discover patterns which are more sophisticated than a simple sequence of symbols. To this end, we propose to extend IDB patterns by adding numerical constraints. In this proposal, the data are event sequences and the patterns are chronicles. A chronicle is composed of events and numerical temporal constraints, in the form of numerical intervals, on delays between their occurrences. Figure 1 illustrates a chronicle $\mathcal{C}_1$ and an event sequence l_1.

A chronicle is recognized in an event sequence as follows: a set o_i of events of the sequence l_1 instantiates the chronicle $\mathcal{C}_1$ if every event of $\mathcal{C}_1$ is instantiated by an event of o_i respecting type and temporal constraints. A set o_i is called a *chronicle instance* of $\mathcal{C}_1$. For example, the event sequence l_1 has six instances of the chronicle $\mathcal{C}_1$: $|\mathcal{I}_{\mathcal{C}_1}(l_1)| = 6$.

2 Time in data mining

Several solutions have been proposed to tackle the problem of time integration in data mining. Some pattern extraction techniques use symbol ordering as a model of time. For instance, Winepi and Minepi [Mannila *et al.*, 1997] extract patterns represented by serial or parallel sequences of symbols. SeqLog [Lee and De Raedt, 2003] extracts ordered sequences

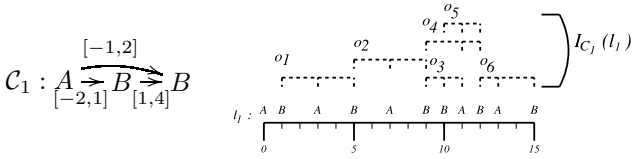
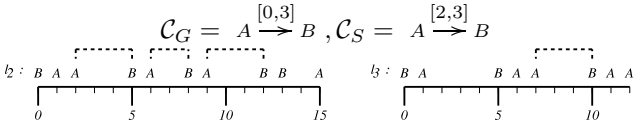


Figure 1: A chronicle $\mathcal{C}_1$ of size 3: intervals labeling edges represent bounds on the distance between event times as $-2 \leq t_B - t_A \leq 1$. Right, 6 instances of $\mathcal{C}_1$ in a sequence l_1.

of symbols with gaps. Parallel and serial notions can also be expressed in chronicles. In addition, they can state numerical constraints on delay between events.

Time can be also represented by a special numerical attribute. For instance, Yoshida [Yoshida *et al.*, 2000] extracts frequent itemsets containing a temporal feature. Their algorithm extracts "delta-patterns" that are ordered lists of itemsets with time intervals between two successive itemsets (as for instance: $\{b\} \xrightarrow{[17,19]} \{a,c\} \xrightarrow{[5,12]} \{b,c\}$) which specify the numerical temporal constraints between the itemsets. However, interval bounds of delta-patterns are always positive whereas interval bounds of chronicles can be negative.

Our goal is to discover chronicles that are frequent in some event sequences and infrequent in other ones. For instance, let l_2 and l_3 be the event sequences of figure 2. A complex query such that $freq(\mathcal{C}, l_2) \geq 3 \wedge freq(\mathcal{C}, l_3) \leq 1$ asks for the set of chronicles that are frequent in l_2 and infrequent in l_3 according respectively to thresholds 3 and 1. Chronicles $\mathcal{C}_S$ and $\mathcal{C}_G$ (fig. 2) satisfy this query because both cover three instances in l_2 and only one in l_3.



Instances of $\mathcal{C}_S$ or $\mathcal{C}_G$ in l_2:
$\{((A,2),(B,5)),((A,6),(B,8)),((A,9),(B,12))\}$
Instance of $\mathcal{C}_S$ or $\mathcal{C}_G$ in l_3: $\{((A,7),(B,10))\}$

Figure 2: Chronicles search under frequency constraints

Furthermore, we introduce a generality relation between chronicles: a chronicle $\mathcal{C}$ is *more general* than a chronicle $\mathcal{C}'$ (noted $\mathcal{C} \sqsubseteq \mathcal{C}'$) if there is a match between $\mathcal{C}$ and $\mathcal{C}'$ such that every event of $\mathcal{C}$ is matched to an event of $\mathcal{C}'$ and the temporal constraints of $\mathcal{C}$ contains the corresponding constraint in $\mathcal{C}'$. Thus, we can note that $\mathcal{C}_G \sqsubseteq \mathcal{C}_S$ (fig. 2) and that there is no solution chronicle more specific than $\mathcal{C}_S$ or more general than $\mathcal{C}_G$.

3 Frequency of chronicles in a sequence

The most used interest measure for patterns mining is based on frequency. The use of a generality relation makes chronicle search based on frequency easier, but only on the condition that constraints on frequency satisfy monotonicity or anti-monotonicity properties. Let $\mathcal{C}$ be more general than $\mathcal{C}'$. The threshold constraints on frequency satisfy one of these properties if the frequency of $\mathcal{C}$ is equal or greater than the frequency of $\mathcal{C}'$ in an event sequence. To ensure these properties for some frequency measure, we introduce a recognition criterion Q that specifies how instances of some chronicle are selected in a sequence and consequently how the frequency of this chronicle is computed.

Some recognition criteria have been defined in the literature. The minimal occurrences criterion [Mannila *et al.*, 1997] selects all the shortest instances of an episode (a partially ordered collection of events occurring together). The earliest distinct instances criterion, $Q_{e\&d}$ [Dousson and Duong, 1999], selects all the instances such that two instances of a same chronicle have no common events and occur as early as possible in the event sequence according to a total order on instances. Let $freq^{Q_{e\&d}}(\mathcal{C}_1, l_1)$ be the frequency of C_1 computed on the event sequence l_1 (fig. 1). The recognition criterion $Q_{e\&d}$ recognizes the set of instances $E = \{o_1, o_3, o_6\}$ and the value of $freq^{Q_{e\&d}}(\mathcal{C}_1, l_1) = |E| = 3$.

4 Queries on event sequences

Let P and N be two sets of event sequences and T be a set of frequency thresholds. Every event sequence l has a threshold T_l. If l is an element of P (resp. N) then T_l is a minimum (resp. maximum) frequency threshold. Our goal is to extract some phenomenon in the form of chronicles. These are frequent in at least one event sequence of P and infrequent in every event sequence of N. A query has the general form:

$$Qu(P,N,T) = \big(\exists l \in P, freq^Q(\mathcal{C}, l) \geq T_l\big) \wedge \big(\forall l \in N, freq^Q(\mathcal{C}, l) < T_l\big) \quad (1)$$

The computation of frequent and maximally specific chronicles, $Fmc^Q(l, T_l)$, in an event sequence l according to a threshold T_l is performed by the levelwise algorithm of FACE [Dousson and Duong, 1999] that we have adapted to the task. The method consists in computing all the *Fmcs* from the different event sequences separately (with FACE), memorizing them, and merging them to get the solutions of the query: this method is efficient if we consider a set of queries that use the same sets T and $(P \cup N)$. We use Mitchell's algorithm to merge these *Fmcs*. This algorithm is extensively used to compute the version spaces of symbol sequences [De Raedt and Kramer, 2001]. It computes the bounds of the version space from a conjunction of constraints. Query (1) is rewritten in such a way as to use Mitchell's algorithm:

$$Qu(P,N,T) = \bigvee_{\mathcal{B} \in Fmc^Q(P,T)} \Big(\mathcal{C} \sqsubseteq \mathcal{B} \bigwedge_{\overline{\mathcal{B}} \in Fmc^Q(N,T)} \mathcal{C} \not\sqsubseteq \overline{\mathcal{B}}\Big) \quad (2)$$

where $Fmc^Q(E,T) = \bigcup_{l \in E} Fmc^Q(l, T_l)$.

For each chronicle from $Fmc^Q(P,T)$, the version space of a conjunction of constraints is computed by Mitchell's algorithm. The union of these version spaces gives the solutions set to query (2). One of the main interests of this method is that it needs to compute only once the *Fmcs* in each sequence.

5 Conclusion

We have presented an original method that extracts temporal information in the form of patterns named chronicles. These patterns use numerical temporal constraints on events. Furthermore, they provide a way to express sequentiality and parallelism between events and they generalize temporal patterns used by Mannila and De Raedt [Mannila *et al.*, 1997; Lee and De Raedt, 2003], among others. Chronicles are extracted by querying an IDB that contains event sequences and chronicles. The user sets the minimum or maximum frequency thresholds of searched chronicles in event sequences.

Our contribution introduces the notion of recognition criterion that generalizes the specification of pattern frequency computation on temporal data. Furthermore, we use a generality relation on chronicles that enables us to adapt version space algorithms to manage numerical temporal constraints. These algorithms require the prior computation of frequent and maximal chronicles for each event sequence used in the query. This computation is performed by a data mining tool that we have adapted to the task. Our approach can compute the complete and correct set of solutions.

References

[De Raedt and Kramer, 2001] L. De Raedt and S. Kramer. The levelwise version space algorithm and its application to molecular fragment finding. In *Proc. of IJCAI*, 2001.

[Dousson and Duong, 1999] C. Dousson and T. Vu Duong. Discovering chronicles with numerical time constraints from alarm logs for monitoring dynamic systems. In *Proc. of IJCAI 1999*, pages 620–626, 1999.

[Imielinski and Mannila, 1996] T. Imielinski and H. Mannila. A database perspective on knowledge discovery. *Communications of the ACM*, 39:58–64, 1996.

[Lee and De Raedt, 2003] S. D. Lee and L. De Raedt. *Database Support for Data Mining Applications*, volume 2682 of *LNCS*, chapter Constraint Based Mining of First Order Sequences in SeqLog. Springer-Verlag, 2003.

[Mannila *et al.*, 1997] H. Mannila, H. Toivonen, and A. Inkeri Verkamo. Discovery of frequent episodes in event sequences. *Data Mining and KD*, 1(3), 1997.

[Yoshida *et al.*, 2000] M. Yoshida, T. Iizuka, H. Shiohara, and M. Ishiguro. Mining sequential patterns including time intervals. In *Data Mining and KD*, 2000.