Spatial Sequential Pattern Mining for Seismic Data

Riccardo Campisano[1], Fabio Porto[2], Esther Pacitti[3],
Florent Masseglia[3], Eduardo Ogasawara[1]

[1]CEFET/RJ

[2]LNCC - DEXL Lab

[3]Inria and LIRMM - Montpellier, France

riccardo.campisano@linea.gov.br, fporto@lncc.br, Esther.Pacitti@lirmm.fr,

Florent.Masseglia@inria.fr, eogasawara@ieee.org

Abstract. *A myriad of applications from different domains collects time series data for further analysis. In many of them, such as seismic datasets, the observed data is also associated to a space dimension, which corresponds, in fact, to spatial-time series. The analysis of these datasets is difficult due to both the continuous nature of the observed data and the relationship between spatial and time dimensions. Meanwhile, sequential patterns mining techniques have been successfully used in large volume of transactional databases to obtain insights from data. In this work, we start exploring the discovery of frequent sequential patterns in seismic datasets. For that, we discretize continuous values into symbols and adapt well known sequential algorithm to mine spatial-time dataset. To better understand the quality of the identified patterns, we visualize them over the original seismic traces images. Our preliminary results indicate that the study of sequence mining in seismic datasets is promising.*

1. Introduction

Frequent pattern mining is widely used to obtain new exploratory knowledge from data. Earlier frequent patterns techniques focused on finding association rules on transactional databases, where each transaction is assumed to be independent from the others. Later, this concept evolved to sequential pattern mining taking advantage of the fact that some successive transactions belong to the same customer. Such concept was further generalized to handle databases composed of generic types of data-sequences [Mabroukeh and Ezeife, 2010].

Meanwhile, many important event records are characterized as time series, in which data are chronological ordered observations. When such data is continuous, it is not efficient to conduct an exact match comparison between values [Fu, 2011]. Moreover, some important phenomena are indeed spatial-time series, leading to an additional challenge caused by the introduction of spatial dimension that increases the complexity due to the spatial-time relationship associated to the collected observations [Han et al., 2007].

In such scenario, considering that a time series corresponds to a sequence of observations, and a spatial-time series associates a position to such time series, we can state our problem definition as follows: *given a set of spatial-time series, find the sub sequences of these observations that are frequent according to the user-defined constraints.*

We initially tackle the problem by proposing a solution that interprets a set of spatial-time series as a collection of spatial data-sequences, where each data-sequence is a sequence of observations. In order to do that, we initially discretize continuous values, in a way that is possible to find frequent patterns looking for exact matching of observations. Then, we apply an adapted sequential pattern detection algorithm. Finally, results are visually represented to display the positions, according to the original dataset, of each detected sequence.

In order to assess whether such approach can be promising for discovering spatial-time series patterns, we apply our technique in a real world seismic use case to detect zones of geologic boundaries, such as seismic horizons and faults. In seismic surveys, the material reflection values are associated to a particular time (depth) and position. From the visualization of identified patterns, we observed that our approach is promising as it is able to detect frequent sequences that spatially and timely correspond to some geologic boundaries of the subsoil.

Besides this introduction, this work is organized in four more sections. Section 2 describes our methodology, including the main algorithm. Section 3 presents preliminary results. Section 4 provides the main related work for identifying patterns in spatial-time series while Section 5 presents final considerations and indicates future steps for this research.

2. Sequence Mining in Spatial-Time Series

We start by formalizing the problem of sequence mining in spatial-time series.

2.1. Problem Formulation

A spatial-time dataset D is composed of a list of spatial data-sequences such that $D = \{d_1, d_2, ..., d_n\}$. Each data-sequence d_i is composed of a list of observations such that $d_i = < v_1, v_2, ..., v_m >$. Also we define a sequence s_k as a list of observations such that $s_k = < v_1, v_2, ..., v_z > \mid \forall v_i, v_i \in D$. We wish to find all frequent sequences $S_f = < s_1, s_2, ..., s_w > \mid \forall s_i \in S_f, s_i \subset d_j \; and \; \text{support}(s_i, D) > \text{min-support}$. Thus, for each sequence $s_i \in S_f$, the number of data-sequences that contains this sequence divided by the total number of data-sequences in D is greater or equal than a minimum relative support *min-support* chosen by the user. In this way, the *min-support* constraint guarantees that the sequence s_i is a frequent sequence in the spatial-time dataset D.

2.2. Mining Spatial-Time Series

Given the problem formulated in 2.1, we describe the process adopted to mine spatial-time series. Initially, a discretization step transforms continuous value ranges into symbolic values. Symbolic Aggregate Approximation (SAX) algorithm [Lin et al., 2003] was chosen by virtue of its ability in creating an equiprobable symbolic representation for time series observations that considers data distribution. Each symbol is associated to a range with the approximate same number of observations [Lin et al., 2003].

Next, we apply a sequential pattern detection algorithm, adapted from the GSP algorithm [Mabroukeh and Ezeife, 2010]. The algorithm firstly detects frequent single observations. Then, composed candidate frequent sequences are built from the combination of previously detected frequent sequences. Finally, these candidates sequences are evaluated by removing the ones that are not really frequent.

2.3. Sequential Pattern Mining Algorithm

Algorithm 1 depicts the mechanics for identifying spatial-time patterns. Initially, frequent single items are found through function $getFrequentItems$, *i.e.*, items that reach the user defined *min-support*. Each of the discovered frequent items is transformed in 1-sequence structure (sequence of size one) by function $convertToSequences$. Then, in function $joinSequences$, the previously generated (k-1)-sequences are permuted to compose frequent k-sequence candidates. Next, the support of each frequent k-sequence candidate is computed and the ones that do not reach the *min-support* are discarded through function $pruneCandidates$. This preserves only the verified frequent k-sequences. The algorithm goes back to function $joinSequences$ and repeats until no new candidate is generated in $joinSequences$ or until none of the new candidates satisfies the *min-support* constraint defined in function $pruneCandidates$.

Algorithm 1 Sequential pattern mining on Spatial-Time Series

```
function FINDFREQUENTSEQUENCES($d\_seq, min\_sup, max\_stretch$)
    $freq\_items \leftarrow getFrequentItems(d\_seq, min\_sup)$
    $freq\_k\_seq \leftarrow convertToSequences(freq\_items)$
    while $count(freq\_k\_seq) > 0$ do
        $all\_freq\_seq \leftarrow all\_freq\_seq \cup freq\_k\_seq$
        $cand\_k\_seq \leftarrow joinSequences(freq\_k\_seq)$
        $freq\_k\_seq \leftarrow pruneCandidates($
            $cand\_k\_seq,\ d\_seq,\ min\_sup,\ max\_stretch)$
    end while
    return $all\_freq\_seq$
end function
```

The $joinSequences$ function shown in Algorithm 1 produces a list of k-sequences $S_k =< s_1, s_2, ..., s_n >$ joining all the permutations of each pair of (k-1)-sequences (s_1, s_2) in a list of (k-1)-sequences S_{k-1}. As defined in the GSP algorithm, if the input list (k-1)-sequences is a 1-sequences list (*i.e.* list of sequences containing just one item), the result of the $joinSequences$ function is a 2-sequences list generated by the permutation of each pair (s_1, s_2) of the 1-sequences input list. Otherwise, if the input list (k-1)-sequences is composed of sequences containing more than one item, each permutation of pair (s_1, s_2) is joined only if the sequence s_1 without its first item is equal to the sequence s_2 without its last item, and the join result is a k-sequence containing all the s_1 items extended with the last item of the s_2.

Algorithm 1 clearly differs from GSP algorithm in the sense that, due to the nature of the spatial-time series, the former does not include *itemsets* and *sliding time window* constraints that are transaction related. Moreover, our adaptation to support spatial-time series does not needs to include the definition of *taxonomies* (generalizations). We use the *min-support* constraint to define how much the discovered sequences must be frequent over the space, as well as we define the *max-stretch* constraint in replacement of original GSP *max-gap* constraint to relax the support count of candidate sequences. With the *max-stretch* definition, a data-sequence supports a candidate sequence if the items of the candidate are contained in the data-sequence in the same order and if the first and last items the candidate are found in the data-sequence with a maximum distance of the candidate size plus the *max-stretch* value. In such way, both *max-gap* and *max-stretch* supports noise in the sequence comparison. However, *max-stretch* produces more restrictive frequent sequences according to time.

3. Preliminary Results

Seismic datasets are collected from a set of receivers capturing the waves generated from artificially produced shots on the terrain surface and reflected by the different materials of the subsoil. The wave propagates under the ground in a certain velocity, reflect in each material layer and go back to the surface in a determinate time, so that the early received reflection signal corresponds to more superficially depths of the ground, and vice-versa. Particularly important things to recognize are faults and horizons, *i.e.*, zones of major unconformities that corresponds to specific geologic boundary [Zhou, 2014].

The spatial-time dataset chosen as input for this work is part of the *Netherlands Offshore F3 Block* seismic survey. The dataset is produced collecting the reflection values of the subsoil materials, resulting in a 3D cube composed of a range of 100 to 750 *inlines* and 300 to 1250 *crosslines* sections orthogonal to the surface plan. Each *inline* or *crossline* is composed of several sets of observations made at different aligned positions on the surface. For each position at the surface, a set of observations is collected, composing a spatial-time series of seismic reflection values at different reflection time, where each observation time corresponds to a depth of the subsoil.

For the scope of this experiment, only the *inline* 100 was considered. It is a spatial-time dataset composed of 951 spatial-time series, each of them represents the reflection values of different locations of the ground. Each spatial-time series contains 462 observations representing the different depths of the subsoil. The time series can be figured out as vertical columns of pixels in the *inline* 100, shown in Figure 1.a, where the lower reflection values are represented with red pixels and the higher reflection values are represented with dark pixels.

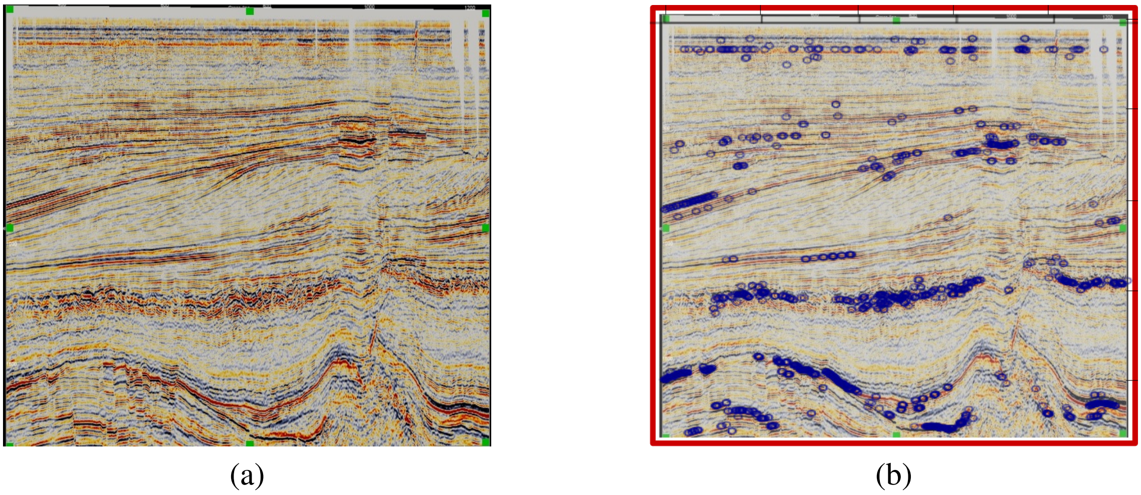

Figure 1. Seismic trace image of *inline* 100 of the *Netherlands Offshore F3 Block* dataset (a). Resulting sequence $< a, a, y, y >$ positions, evidencing where red and dark lines are nearby each other (b).

The discretization was performed using SAX algorithm [Lin et al., 2003] with a *word size* value equal to the SAX input dataset size, since we are not interested in applying dimensionality reduction feature of the SAX algorithm. The choice of this experiment was to empirically test several *alphabet sizes*, more specifically we explored 5, 10, 15, 20, and 25 *alphabet sizes*.

Our algorithm allows for defining the *min-support* as well as the value of *max-stretch* constraint. In our experiment we empirically tested the combinations of ranges of these parameters, more specifically 100%, 95%, 90%, 85%, 80% *min-support* values in combination with 0, 1, 2, 3 *max-stretch* values. Our results indicate that a too low value for *min-support* as well as a too high value of *max-stretch* can lead to detect a large number of not useful sequences. Conversely, a too high value for *min-support* or a too low value of *max-stretch* constraint can lead to detect a little number of frequent sequences or sequences too small to be interesting.

Figure 1.b shows the $< a, a, y, y >$ frequent sequence discovered using the *inline* 100 dataset discretized using an *alphabet size* of 25 and running the algorithm with a *min-support* of 70% and a *max-stretch* of 2. The image is generated drawing several blue circles, each of them indicates the start position of one frequent sequence detection in the original dataset. It corresponds to the area in the original Figure 1.a where red and dark lines were found nearby each other, evidencing several horizon segments. In fact, the sequence $< a, a, y, y >$ represents two red pixels followed by two dark ones.

4. Related Work

Sequential Pattern mining aims in finding statistically relevant patterns from values that are organized in sequences. Over the last decade, many algorithms have been created [Mabroukeh and Ezeife, 2010]. Yet, when it comes to sequence mining on spatial-time series, we observe room for research.

Spatial-time series can be composed of observations from either moving or persistent objects [Frank, 2003]. When one of the varying properties of the objects is the position, we can target for patterns like routes frequently followed by objects or paths with some similar attributes. Such analysis of trajectory data, clearly differs from the goal of this work.

On the other side, spatial-time series can be obtained from observation of multiple locations or static objects that have properties changing over the time. Both Alatrista-Salas et al. [2015] and Leong and Chan [2012] try to discovery patterns in spatial-time series. Alatrista-Salas et al. [2015] uses a discretization technique that leads to four or five bins in each variable. After this data preprocessing, they apply PrefixSpan sequential mining algorithm. Alatrista-Salas et al. [2015] define interesting relations between data-sequences using the space information, but its application is restricted to their studied domain. Leong and Chan [2012] uses GSP algorithm over a synthetic discrete dataset of daily events, organizing the data as spatial data-sequences and computing support count for each candidate pattern. Leong and Chan [2012] use transactions to aggregate multiple observations in the same time, which loses time granularity. In our work, we applied SAX indexation that allows for distributing data equally with respect to the chosen *alphabet size*. Also, we do not lose time granularity by preserving the temporal order of observations, since we are interested in finding frequent sequences that are constrained both space and time. Finally, our algorithm differs, since we introduced the concept of *max-stretch* to allow noise data, in the way as *max-gap*, but it produces more restrictive frequent sequences according to time.

5. Conclusions

The visualization of the detected frequent sequences indicates that the results are promising. The algorithm was able to detect sequential patterns in the spatial-time series dataset. The position of the detected frequent sequences follows some of the geologic boundaries of the subsoil as can be shown in Figure 1.b. However, there are several aspects that can be improved in future works. Specially, when it comes to identify seismic faults, which are patterns that are constrained in a specific region.

The choice of having high values for *min-support* constraint forces the identified frequent patterns to be found nearly all data sequences, *i.e.*, nearly all spatial-time series (vertical columns of the image of Figure 1.a). It does not take in account if frequent sequences are continuous patterns or if they are segmented in the dataset.

To reduce the large number of frequent sequences that can be obtained from the algorithm, we can only include the maximal sequences in the result, as proposed by Agrawal and Srikant [1995]. This principle takes advantage of the fact that each subsequence of a frequent sequence is already frequent. Moreover, ranking the resulting sequences can be another way to handle the large number of frequent sequences obtained. Tight sequences could gain a better rank than sparse ones, as well as sequence with greater information gain, like sequence $< a, d, c, f >$, could gain a better rank than others containing lower information gain, such as $< a, a, a, a >$.

References

Agrawal, R. and Srikant, R. (1995). Mining sequential patterns. In *Proceedings of the Eleventh International Conference on Data Engineering*, ICDE '95, pages 3–14, Washington, DC, USA. IEEE Computer Society.

Alatrista-Salas, H., Azé, J., Bringay, S., Cernesson, F., Selmaoui-Folcher, N., and Teisseire, M. (2015). A knowledge discovery process for spatiotemporal data: Application to river water quality monitoring. *Ecological Informatics*, 26:127–139.

Frank, A. U. (2003). Ontology for spatio-temporal databases. In *Spatio-Temporal Databases*, pages 9–77. Springer.

Fu, T.-c. (2011). A review on time series data mining. *Eng. Appl. Artif. Intell.*, 24(1):164–181.

Han, J., Cheng, H., Xin, D., and Yan, X. (2007). Frequent pattern mining: Current status and future directions. *Data Min. Knowl. Discov.*, 15(1):55–86.

Leong, K. and Chan, S. (2012). Stem: A novel approach for spatiotemporal sequence mining. *Asian Journal of Information Technology*, 11(3):94–99.

Lin, J., Keogh, E., Lonardi, S., and Chiu, B. (2003). A symbolic representation of time series, with implications for streaming algorithms. In *Proceedings of the 8th ACM SIGMOD Workshop on Research Issues in Data Mining and Knowledge Discovery*, DMKD '03, pages 2–11, New York, NY, USA. ACM.

Mabroukeh, N. R. and Ezeife, C. I. (2010). A taxonomy of sequential pattern mining algorithms. *ACM Comput. Surv.*, 43(1):3:1–3:41.

Zhou, H.-W. (2014). *Practical seismic data analysis*. Cambridge University Press.