Swap Randomization of Bases of Sequences for Mining Satellite Image Times Series

Nicolas Méger[1(✉)], Christophe Rigotti[2], and Catherine Pothier[3]

[1] LISTIC Laboratory, Université Savoie Mont Blanc, Polytech Annecy-Chambéry, B.P. 80439, 74944 Annecy-le-vieux Cedex, France
nicolas.meger@univ-smb.fr

[2] LIRIS Laboratory (UMR 5205), Université de Lyon, CNRS, INRIA, INSA-Lyon, 20 Avenue A. Einstein, 69621 Villeurbanne Cedex, France
christophe.rigotti@insa-lyon.fr

[3] LGCIE Laboratory, Université de Lyon, INSA-Lyon, 20 Av. A. Einstein, 69621 Villeurbanne Cedex, France
catherine.pothier@insa-lyon.fr

Abstract. Swap randomization has been shown to be an effective technique for assessing the significance of data mining results such as Boolean matrices, frequent itemsets, correlations or clusterings. Basically, instead of applying statistical tests on selected attributes, the global structure of the actual dataset is taken into account by checking whether obtained results are likely or not to occur in randomized datasets whose column and row margins are equal to the ones of the actual dataset. In this paper, a swap randomization approach for bases of sequences is proposed with the aim of assessing sequential patterns extracted from Satellite Image Time Series (SITS). This assessment relies on the spatiotemporal locations of the extracted patterns. Using an entropy-based measure, the locations obtained on the actual dataset and a single swap randomized dataset are compared. The potential and generality of the proposed approach is evidenced by experiments on both optical and radar SITS.

1 Introduction

Earth observation satellite technology is continuously being enhanced, providing end users with ever ever-growing data volumes. Improvements relate to the number of acquisition channels, the spatial resolution and the revisit frequency. The revisit capability makes possible to gather acquisitions of a same geographical zone through time and form *Satellite Image Time Series (SITS)*. SITS are large datasets containing complex spatiotemporal information that can be affected both by atmospheric perturbations and sensor problems. In order to fully exploit such SITS, information retrieval and data mining techniques are being developed. Among them, unsupervised data mining techniques demonstrate their potential when it comes to describe and discover spatiotemporal phenomena. They rely either on global models such as clusterings (e.g., [13] or [21]) or on local patterns such as sequential patterns (e.g., [16] or [14]). In particular, a SITS can be considered as a special kind of base of sequences, as first introduced in [1]. In that

A. Appice et al. (Eds.): ECML PKDD 2015, Part II, LNAI 9285, pp. 190–205, 2015.
DOI: 10.1007/978-3-319-23525-7_12

initial context, each sequence gives the transactions of a customer whereas, in the case of a SITS, each sequence contains the descriptions of the values of a pixel through time and is thus located spatially. As proposed in [17], *Grouped Fequent Sequential patterns (GFS-patterns)* can be extracted from such a base of sequences. Besides expressing pixel temporal evolutions, these sequential patterns also take into account the spatial information brought by SITS: each GFS-pattern is required to affect a group of pixels that are sufficiently numerous and connected to each other. Reciprocally, each pixel can be affected by different GFS-patterns. As a consequence, pixel groups corresponding to extracted GFS-patterns can partially or fully overlap each other: they can refine each other. Extracting GFS-patterns thus differs from segmenting or clustering a SITS. Experiments reported in [17] or [22] show that GFS-patterns can be used both on radar and optical data, for various applications ranging from agricultural to crustal deformation monitoring. Despite their ability to address various types of datasets and applications, these patterns can be numerous, even if maximal ones are focused on. How to select the most significant ones without making any assumption? We aim to answer that question by adapting swap randomization to the SITS mining context.

In statistics, the significance of a result (e.g., the number of correlations found in a dataset) can be assessed via randomization testing methods [12]. Basically, they check whether the result observed on the actual dataset is likely to be obtained or not on randomized datasets. These datasets are meant to sufficiently differ from the actual one while sharing some of its structural properties such as the number of 0's and 1's in the case of a Boolean matrix. With this aim in view, randomized datasets are built by shuffling the actual dataset. Considering randomized datasets avoids generating random ones by sampling a distribution law that has to be defined a priori. Swap randomization follows these guidelines and focuses on more fined-grained structural properties such as the column and row margins of a Boolean matrix [5]. In data mining, as evidenced in [9], [10] or [15], swap randomization can be exploited to assess the significance of global models characterizing the whole actual dataset. These models can be clusterings, sets of frequent itemsets, sets of correlations or singular values. Even if they do not describe the entire dataset, local patterns such as frequent itemsets can also be evaluated individually (e.g., [10] or [15]).

To our knowledge, no swap randomization techniques handling bases of sequences or SITS have been proposed so far. In this paper, such a proposal is made with the aim of evaluating GFS-patterns [17] individually. While being dedicated to GFS-patterns, the presented approach could also be used for any kind of sequential patterns or episodes. Assessing GFS-patterns is not a trivial task. Their spatiotemporal nature must be taken into consideration and the following questions must be answered: which fine-grained structure should be maintained when randomizing the base of sequences representing a SITS? Which GFS-pattern-related information should be considered for their individual assessment? How to compare the information observed on the actual dataset with the one obtained for the randomized datasets? How to be efficient when considering a SITS containing millions of pixel values? Our answers are as follows: with regards to the structure

to be maintained while randomizing, the distributions of the values of each image and each pixel sequence are preserved. The assessment of a GFS-pattern is then performed by comparing its spatiotemporal locations on the actual dataset with the ones on the randomized datasets. This comparison relies on the *Normalised Mutual Information (NMI)* [6], an entropy-based measure. Efficiency is achieved by performing the comparison using a single randomized dataset, as opposed to hundreds of randomized datasets when considering the standard swap randomization approach. This paper is organized as follows: Section 2 gives some preliminary definitions regarding SITS and GFS-patterns. The swap randomization approach proposed to shuffle bases of sequences representing a SITS is detailed in Sect. 3. Section 4 explicates GFS-pattern assessment and its use for SITS summarization. Experiments are presented in Sect. 5. They show that the proposed approach is general enough to mine either radar or optical SITS, yields relevant patterns on real datasets and can support different applications such as land cover or crustal deformation monitoring. Section 6 concludes this paper and gives future work directions.

2 Grouped Frequent Sequential Patterns

In this section, the definition of *Grouped Frequent Sequential Patterns (GFS-patterns)*, as first introduced in [17], is recalled. Let us consider a SITS, i.e., a satellite image time series covering the same area at n different dates. Within each image, each pixel is associated with a value, e.g., the reflectance intensity of the geographical zone it represents. These values are discretized to get *event types* (symbols) encoding *events* under the form of a pair (t, e) with e an event type and t its occurrence date (here the date will be the index of the image in the series). Event types can correspond to ranges obtained by image quantization or to pixel clusters. A *symbolic SITS* is a set of *pixel evolution sequences*, each one containing the coordinates (x, y) of a pixel and its corresponding event sequence, i.e., a tuple of events $\langle (t_1, e_1), (t_2, e_2), ..., (t_n, e_n) \rangle$. In pattern mining, a typical base of sequences is a set of sequences of discrete events, in which each sequence has a unique sequence identifier. Each location (x, y) being unique, a symbolic SITS is a base of sequences and the standard notions of sequential patterns, support and frequent sequential patterns introduced in [1] can be easily reused as follows[1]. A *sequential pattern* α is a tuple of m event types $\langle \alpha_1, \alpha_2, \ldots, \alpha_m \rangle$. The *support* of α in a SITS, denoted by $support(\alpha)$, is the number of pixel evolution sequences in which α occurs at least once. Note that the event types do not need to occur contiguously. Sequential pattern α is a *frequent sequential pattern* if $support(\alpha) \geq \sigma$ with σ a support threshold. Reusing the definitions of sequential patterns permits to take advantage of the efficient extraction techniques developed in this domain (e.g., [1], [25] or [20]). The pixels where a pattern α occurs are said to be *covered* by α. For a SITS, the notion of support can be interpreted very naturally as an area. In order to obtain pixels forming regions in space, an average connectivity measure is also used. It is based on the *8-nearest neighbors*

[1] Sequences are simpler here since there is a single event type for each timestamp.

(8-NN) convention [8]. For α, the *connectivity* of a pixel (x, y) is the number of pixels covered by α among the 8 nearest neighbors of (x, y) (i.e., the pixels surrounding (x, y)). The *average connectivity* of α, denoted $AC(\alpha)$, is simply the average of the connectivity over all pixels covered by α. Finally, a *Grouped Frequent Sequential pattern (GFS-pattern)* α is a frequent sequential pattern such that $AC(\alpha) \geq \kappa$ with κ a positive real number termed *average connectivity threshold*. Depending on the parameter settings and the dataset, numerous GFS-patterns can be produced. In order to reduce the redundancy among the patterns, a standard method is to retain only the maximal ones (e.g., [19]). This approach is also used here. The *maximal* GFS-patterns of a collection of GFS-patterns $\mathcal{C}$ are the elements in $\mathcal{C}$ that are not subpattern of any other pattern in $\mathcal{C}$. In other words, the GFS-patterns focusing on the most specific evolutions are retained. Though the number of GFS-patterns can be drastically reduced by adopting such a strategy, it can still be large. How to select the most significant ones without making any additional assumption with respect to covered pixels (e.g., assumptions about the shape or the texture of pixel groups)? We propose to answer that question by adapting the swap randomization of Boolean matrices to the SITS mining context.

3 Swap Randomization of Base of Sequences Representing SITS

Swap randomization is aimed at generating Boolean matrices having the same row and column margins without assuming any underlying distribution law. To this end, the elements of the matrices are swapped. A swap is defined as follows [23]: let B be a $m \times n$ Boolean matrix. Let u and v be two rows. Let i and j be two columns. If $B_{u,i} = B_{v,j} = 0$ and $B_{u,j} = B_{v,i} = 1$ then rows (or columns) are changed so that $B_{u,i} = B_{v,j} = 1$ and $B_{u,j} = B_{v,i} = 0$: values 0 and 1 are swapped. By construction, such a swap does not modify column and row margins. These margins give the number of occurrences of symbol '1' (or symbol '0', its dual symbol) for each column and each row. An example is given in Fig. 1. Boolean matrix B' is obtained from matrix B via a single swap such that $u = 2$, $v = 4$, $i = 1$ and $j = 3$. Swapped 0's and 1's are underlined.

In [23], Ryser shows that it is possible, starting from a given Boolean matrix, to generate all possible Boolean matrices having the same row and column margins by applying a series of swaps, each swap being applied to the latest matrix that had been obtained. In [5], on the basis of this result, the authors show that it is possible to randomly generate equiprobable matrices having the same row and column margins. More precisely, starting from a given Boolean matrix, a series of swap is performed by choosing rows and columns at random. Rows and columns can be chose more than once. As a consequence, swaps can be undone. Each swap can be seen as a random step from a vertex to another one in a graph whose vertices represent all possible matrices and whose edges represent transitions that can be performed by swapping 0' s and 1's. The series of swaps can thus be interpreted as a random walk on a graph that, in turn, can be formalized as a Markov

$$B = \begin{pmatrix} 0\ 1\ 0\ 0 \\ \underline{1}\ 0\ \underline{0}\ 1 \\ 1\ 0\ 1\ 1 \\ \underline{0}\ 0\ \underline{1}\ 0 \end{pmatrix}, B' = \begin{pmatrix} 0\ 1\ 0\ 0 \\ \underline{0}\ 0\ \underline{1}\ 1 \\ 1\ 0\ 1\ 1 \\ \underline{1}\ 0\ \underline{0}\ 0 \end{pmatrix}$$

Fig. 1. Boolean matrix B' is obtained from B by swapping underlined values. Both matrices have the same row and column margins.

chain. In such a chain, the authors explain that the probability of state (i.e., each vertex/matrix when considering the graph) to be reached by a sequence of transitions can differ from one state to another. The proposed solution consists in adding self-loops to have all vertices being reached by the same amount of edges, which guarantees that vertices, and thus Boolean matrices are equiprobable [5]. One important question remains: how many random walk steps are needed to get a Boolean matrix that is sufficiently randomized, i.e., that sufficiently differs from the actual dataset? This is still an open research question. See [3] and [2] for discussions regarding the obtention of p-values using a Markov chain. Nevertheless, empirical results are available (e.g., [10]). Holding in place with self-loops is not efficient when trying to get data sets that are sufficiently randomized. An optimization can be achieved by relying on the Metropolis-Hastings algorithm (e.g., [5] or [10]). Another simpler and efficient optimization is proposed in [10]. It is based on the same approach than [5] but requires less self-loops. It relies on a set P containing all pairs (u, i) such that $B_{u,i} = 1$. This structure is made available throughout the whole algorithm. The swapping procedure differs from the standard one: u, v, i and j are not fully chosen at random. They are chosen by randomly selecting two pairs (u, i) and (v, j) in P. If pairs (u, j) and (v, i) are not in P, then $B_{u,j} = B_{v,i} = 0$ and the swap is made effective. Otherwise, the swap attempt is counted as a self-loop. By avoiding a full random walk, the convergence is accelerated and the overhead induced by the management of P is absorbed. In [10], using this algorithm, it is empirically estimated that the number of random walk steps should be in order of the number of 1's of the matrix to converge to a sufficiently randomized Boolean matrix.

Swap randomization is basically applied to Boolean matrices to assess data mining results using p-values. The bottom line is to define a null hypothesis stating that the result observed for the actual dataset is likely to be observed on randomized datasets having the same structure, i.e., the same column and row margins. If the null hypothesis is rejected then the result is considered to be significant. In order to run such a test, a metric of interest has to be chosen. With regards to correlations, it is proposed in [10] to compute the number of correlations or the maximum and the minimum correlation values. The same kind of strategy is also used to analyze sets of frequent itemsets by considering the number of extracted frequent itemsets, the fraction of frequent itemsets that are preserved and the fraction of frequent itemsets that disappear. For this latter case, the analysis is run by directly comparing these numbers and fractions, without using p-values. Still, if required, it would be possible to compute

them. Finally, clusterings are studied through clustering errors. Besides global models, local patterns such as frequent itemsets can also be evaluated individually through their support measure directly or via p-values (e.g., [10]) or [15]). The ratio between the support observed on the actual dataset and the mean support observed for randomized datasets is also mentionned as an interesting alternative. The experiments reported in [5], [9], [10] or [15] all demonstrate the potential of the swap randomization approach in the case of Boolean matrices.

With regard to a $m \times n$ non-Boolean symbolic matrix S, i.e., a matrix containing elements defined with more than two distinct symbols such as '0' and '1', the standard Boolean swap defined in [23] can be extended as follows : let u and v be two rows, and let i and j be two columns. If $S_{u,i} = S_{v,j} = \alpha$ and $S_{u,j} = S_{v,i} = \beta$ with α and β two distinct symbols, then rows (or columns) are changed so that $S_{u,i} = S_{v,j} = \beta$ and $S_{u,j} = S_{v,i} = \alpha$: symbols α and β are swapped. This *symbolic swap* preserves row and column margins. For each symbol used to define S, these margins give the number of its occurrences for each row and each column. A symbolic swap is illustrated in Fig. 2. Non-boolean symbolic matrix C' is obtained from C via a single swap such that $u = 1$, $v = 3$, $i = 1$ and $j = 2$. Swapped symbols '2' and '3' are underlined. Both matrices share the same row and column margins. Sadly, it is not possible to generate all non-Boolean symbolic matrices having the same row and column margins by swapping data. Fig. 2 gives an example: no swap series can be found to transform D into D' though both matrices have the same row and column margins. Consequently, if swap randomization is performed on such matrices, then swap randomized datasets must be compared with the actual dataset to check whether they sufficiently differ from each other.

Following the principles of swap randomization as defined for Boolean matrices, we aim to assess GFS-patterns by randomizing bases of sequences, and more specifically symbolic SITS. This randomization is thus required to maintain a fine-grained structure of the dataset while breaking event connectivity within each image and event ordering within each pixel evolution sequence. This raises the following question: which structure can be preserved? In order to break event connectivity and ordering only, we propose to maintain event type frequencies within each image and each pixel evolution sequence. This can be achieved by considering spatiotemporal swaps, i.e symbolic swaps. Indeed, as long as more than two even types are considered, a symbolic SITS representing n acquisitions of m pixels can be transformed into a $m \times n$ non-Boolean symbolic matrix (and

$$C = \begin{pmatrix} \underline{3}\ \underline{2} \\ 1\ 1 \\ \underline{2}\ \underline{3} \end{pmatrix}, C' = \begin{pmatrix} \underline{2}\ \underline{3} \\ 1\ 1 \\ \underline{3}\ \underline{2} \end{pmatrix}, D = \begin{pmatrix} 1\ 2 \\ 2\ 3 \\ 3\ 1 \end{pmatrix}, D' = \begin{pmatrix} 2\ 1 \\ 3\ 2 \\ 1\ 3 \end{pmatrix}$$

Fig. 2. Non-boolean symbolic matrix C' is obtained from C by swapping underlined values. C and C' have the same row and column margins. D' can not obtained from D by swapping data though they share the same column and row margins.

vice versa). In such a matrix, an element located at row k and column l gives the event type describing a pixel whose coordinates are mapped bijectively to k in the l^{th} image. Consequently, in order to swap randomize a symbolic SITS, we propose to adapt the algorithm described in [10] by performing a series of *swap attempts* which are defined as follows:

Definition 1. ***(swap attempt)*** Let S be a $m \times n$ non-Boolean symbolic matrix representing a symbolic SITS defined over E, the set of event types. Let $P = \{\{(u,i),(v,j)\} | S_{u,i} = S_{v,j} = \alpha,\ \forall \alpha \in E\}$. A *swap attempt* is selecting $p = \{(u,i),(v,j)\} = \alpha \in P$ randomly. If $\exists\, p' \in P$ such that $p' = \{(u,j),(v,i)\} = \beta \mid \beta \neq \alpha$, then a symbolic swap is performed so that $S_{u,i} = S_{v,j} = \beta$ and $S_{u,j} = S_{v,i} = \alpha$. Otherwise no swap is performed but it is still counted as a self-loop.

By performing such spatiotemporal swaps, a first structure level of SITS is maintained, i.e., event type frequencies. Maintaining event type frequencies in images is equivalent to preserving their histograms which are standard first level image descriptors [11]. With respect to pixel evolution sequences, their first structure level can also be given by event type frequencies. From the application point of view, this makes sense. At the image level, an image affected by clouds should not be converted into an image expressing the presence of vegetation (and vice versa). Similarly, vegetation should not be transformed into a glacier. At the pixel evolution sequence level, since each sequence relates to a specific location, if the presence of water is expressed through a sequence, then there is no reason to change it to a sequence relating to bare soils. The same holds for a pixel whose sequence is giving variations between snow and rocks with little vegetation: swap randomization should not transform it into a sequence of permanent vegetation. Maintaining the spatiotemporal structure of a SITS is a strategy similar to the one adopted in [24] to randomize time series collections. In that case, a time series collection is represented by J real-valued matrices, where J is the number of wavelet coefficients used to describe the series, i.e., the maximum detail level. An element located at position (i,j) of the f^{th} matrix gives the value of the f^{th} wavelet coefficient for series i at time point j. These matrices are independently randomized by approximately preserving the temporal distributions (row distributions) and the series domain distributions (column distributions) of the wavelet coefficients. Hence, this approach could be adapted to SITS randomization. Nevertheless, in addition to performing a discrete wavelet transform of the original time series and randomizing several matrices (one per coefficient), an inverse discrete wavelet transform is required to transform each randomized dataset back to the original representation. Finally, if one were to assess GFS-patterns using this approach, then every randomized dataset should also be quantized. Back to our approach, even if the SITS first structure level is preserved, the connectivity and the order of the event types forming GFS-patterns is affected. This allows to detect the GFS-patterns that are due or not to such a structure. As for the algorithm of [10], convergence is accelerated through the use of set P and self-loops allow to generate equiprobable datasets. However, in

practice, as already stated previously in this section, it is not possible to generate all $m \times n$ non Boolean symbolic matrices (and thus symbolic SITS) having the same row and column margins. Still, as shown empirically in Sect. 5, it is possible to generate and explore randomized datasets that differ from each other and that also differ from the actual SITS sufficiently. As long as it makes sense to preserve row and column margins, this kind of technique can also be applied to other types of bases of sequences.

4 GFS-Pattern Assessment and SITS Summarization

Using the SITS swap randomization approach proposed in Sect. 3, we aim to assess GFS-patterns individually. As explained in Sect. 3, when considering the swap randomization of Booleean matrices, frequent itemsets can be assessed through their support measures directly, support ratios or p-values (e.g., [10]) or [15]). With regard to GFS-patterns, considering their support measure only is not sufficient since their spatiotemporal nature is not taken into account fully. The coordinates and the temporal locations (starting dates, ending dates, timespans, etc.) of the pixels affected by a GFS-pattern must also be considered. Therefore, we propose to focus on pixel coordinates and ending dates by relying on *SpatioTemporal Localization Map (STL-maps)*. An STL-map is an image generated for each GFS-pattern given a symbolic SITS (randomized or not). In such an image, if a pixel is covered by the GFS-pattern for which the image was generated, then its value gives the ending date of the earliest occurrence available for the corresponding coordinates. Otherwise, no ending date is stored (a *black pixel value* is used). By construction, STL-maps also include the information related to the support of GFS-pattern. As shown in Sect. 5, and though other types of temporal locations are also interesting, considering ending dates only allows to perform an efficient and reliable GFS-pattern assessment. Efficiency is also achieved by considering a single swap randomized symbolic SITS only: this avoid generating lots of STL-maps and running numerous comparisons.

How to compare the STL-map M, obtained on the actual SITS for a pattern α, with M', the STL-map obtained for α on a single swap-randomized SITS? How to compare them without having to make any assumption about their relation? At this stage, we are interested by the following two settings:

- M and M' are dissimilar: M is singular as it can not be obtained for a randomized dataset with the same structure in terms of event type frequencies,
- M and M' are similar: the swap-randomization does not destroy the occurrences of α and thus C expresses a prominent phenomena explained by the margins.

The first setting is in line with the standard swap randomization approach while the second one is usually not considered since one-tailed tests are focused on. Still, the second setting is of primary interest. Geographical zones affected by few changes are expressed through event types that are sowewhat always the same. Hence, the corresponding events are hardly randomized. If we were to reject

them, the SITS exploration would be biased towards GFS-patterns expressing changes and interesting areas such as deserts, lakes or cities would disappear from extracted descriptions. How to assess and distinguish the latter two settings using a single measure? Let Ω be the sample space containing all ending dates. Let us consider each ending date x of M as the realization of a discrete random variable X and each ending date y of M' as the realization of a discrete random variable Y. We propose to rely on the *Normalized Mutual Information (NMI)* as presented in details in [6].

$$NMI(X;Y) = \frac{\sum_{x,y \in \Omega^2} P(x,y) \log \frac{P(x,y)}{P(x)P(y)}}{min(H(X), H(Y))} \tag{1}$$

where $H(X) = -\sum_{x \in \Omega} P(x) \log P(x)$ and $P(x,y)$ represents the probability of co-occurrence of the two ending dates x and y at the same pixel position, in M and M'. The NMI quantifies the information content shared by two random variables. In other words, knowing the realizations of two random variables X and Y, it measures the extent to which the realizations of variable X can be deduced from the ones of Y, and vice versa. It can been therefore seen as a measure of the mutual dependence between X and Y. The more X and Y are independent (respectively dependent), the more the NMI tends to 0 (respectively 1) since no bit is shared between the two variables. A particular case must be handled: the black pixels. These pixels show no realizations, no ending dates. Since we extract GFS-patterns that may only cover little fractions of the observed zone, black pixels can be numerous with respect to non-black ones. If these numerous black pixels were to be considered as showing another *special* ending date, a lot of black pixels in M could be associated to other black pixels in M': their joint probability would be high, raising the NMI measure artificially and masking the other, but more important, joint probabilities. Consequently, the joint probability of black pixels is not considered. Nevertheless, because of the swap-randomization, black pixels can differ from M to M': these other cases are taken into account thanks to joint probabilities having one of the two values set to a black pixel value.

Once the NMI is computed for each STL-map/GFS-pattern, then STL-maps/GFS-patterns are ranked accordingly. The NMI-based ranking that is obtained can be easily browsed to build a SITS summary by focusing on both ends of the ranking. Phenomena that can not be obtained on a swap-randomized SITS have low NMI scores and prominent phenomena that are still present in a swap-randomized STIS have high NMI scores. As shown in Sect. 5, if several swap randomized SITS are computed, then rankings are stable for high and low NMI GFS-patterns: a single swap randomized SITS can thus be considered. By relying on the NMI, no assumption about the relation between the ending dates is done. Beside extracting GFS-patterns, this allows us to produce summaries which are as unsupervised as possible.

5 Experiments

The swap randomization approach presented in this paper was assessed by conducting experiments on two different SITS, a radar one and an optical one. Their characteristics are given by Table 1. For each SITS, raw data are transformed into a single synthesized channel dedicated to the application domain. Regarding Etna, phase delays were computed [7] by Marie-Pierre Doin (ISTerre laboratory, CNRS). These floats express vertical and/or lateral displacements w.r.t. a master acquisition. An example is given by Fig. 9 where Mount Etna is revealed in the upper part of the image. For NC, the *Normalized Difference Vegetation Index (NDVI)* [4] was generated by Rémi Andréoli (Bluecham S.A.S. *www.bluecham.net*). It expresses the presence of biomass. An example is shown in Fig. 10: the ocean (resp. land) is mainly located in the lower right part (resp. upper left part) of the image. Radar shadows, atmospheric perturbations, clouds and sensor defaults are still present in these synthesized channels. Preprocessing details are available in Table 1.

The experiments were run on a standard computing platform (a single core on a 2.7 GHz Intel Core i7) using our own prototype *SITS-miner* implemented in C and Python. On the side of parameter settings, average connectivity threshold κ is set to 5 neighbors to extract zones making sense spatially. This is a standard setting [17]. In order to assess reasonable amounts of GFS-patterns, we focus on maximal ones, as explained in Sect. 2. With regard to minimum support threshold σ, it is set such that the *richest/most diverse* description is obtained. This achieved by finding the lowest value of σ such that the number of maximal GFS-patterns is maximum: the widest possible range of surfaces, from σ to the surface of the image itself, is considered. Following this strategy, minimum support threshold σ was found to be 7000 for both SITS (covering about 2.11% of an image in Etna and 2.66% in NC). By consuming no more than 655 MB of RAM and in less than one minute, 508 maximal GFS-patterns are extracted from Etna and 297 maximal GFS-patterns are mined in NC[2].

These patterns were assessed using the swap randomization approach and the NMI ranking procedure described in this paper. Regarding swap randomization, the parameter to be set is N_s, the number of swap attempts to be performed. In [10], it is empirically estimated that N_s should be in order of the number 1's of the matrix to converge to a sufficiently randomized Boolean matrix. In our case, we will consider the number of events multiplied by about 20 to adopt a very conservative setting: $N_s = 100.000.000$. This setting makes sense since it can be empirically shown that the two SITS are sufficiently randomized to get stable NMI values for the patterns we are interested in, i.e., those located at both ends of the NMI rankings (see Sect. 4). Let us the consider the 20 highest and the 20 lowest NMI patterns obtained for 100M swaps. Their respective NMI values were also computed for $N_s = 20\text{M}, 40\text{M}, \ldots, 140\text{M}$, and are reported as randomizations labelled 0 to 6 in the figures 3, 4, 5 and 6. As it can be observed, they rapidly converge to levels that are quite stable, especially around 100M of swaps. With regard to swapped

[2] The reader is referred to [17] and [18] for discussions regarding the impact of σ and of the number of event types on the number of extracted patterns.

Table 1. SITS properties, preprocessing and extraction settings.

SITS name	Etna	NC
provider/credit	ESA	USGS/NASA Landsat
satellite	ENVISAT	LANDSAT 7
SITS type	Synthetic Aperture Radar	Multispectral
time period	16 images 2003-2010	16 images 2000-2011
site	Geohazards Supersite: Mount Etna	UNESCO World Heritage Site: lagoons of New Caledonia
application	crustal deformation monitoring	soil erosion monitoring
data quality	pixel values are not always available (radar shadows), atmopheric perturbations	a lot of clouds, sensor defaults
image size	598×553	513×513
resolution	160 m	30 m
synthesized channel	phase delays	NDVI
discretization	quantization/all images (33^{rd} and 66^{th} centiles)	quantization/each image (33^{rd} and 66^{th} centiles)
event types	'1': motion towards satellite (satellite on the left) '2': stable '3': motion away from satellite	'1': few biomass '2': average biomass '3': lot of biomass
parameters	$\sigma = 7000, \kappa = 5$	$\sigma = 7000, \kappa = 5$

randomized SITS themselves, we generated 1000 swap randomized datasets for each SITS to evaluate them. Though 73.9% of the Etna events and 16.2% of the NC events can not be swapped, in average, 6.5% of the Etna events and 32.9% of the NC events were swapped. The standard deviation of these swapped event rates tends to 0, which shows the stability of our swap randomization process. Finally, if we consider a single randomization and focus on effective swaps (self-loops are not counted), it should be mentioned that 1.070.219 different swap randomized datasets are explored when randomizing Etna. Among them, one dataset is generated 8 times and others are obtained only once. In the case of SITS NC, 8.911.591 different datasets are generated once, one dataset is obtained 4189 times and another one is reach 44 times. Consequently, and though no all SITS having the same column and row margins can be reached (see Sect. 3), the proposed swap randomization approach does explore a lot of different SITS having the same structure. As proposed in Sect. 4, for efficiency reasons, rankings are established using a single swap randomized dataset. This makes sense for both SITS since rankings are stable for high and low NMI GFS-patterns. As shown by Fig. 7 and Fig. 8, the rank standard deviation is less than 1 for both ends of the ranking. It was computed using the rankings obtained for the 1000 swap randomized datasets we generated for both SITS. Similar results are obtained when plotting the rank standard deviation against the rank mode or a reference ranking. For both SITS, memory consumption and execution times do no not exceed 1.66 GB of RAM and 700 seconds to perform the pattern extraction, the STL-map computation for the maximal patterns, a single randomization and the final ranking of STL-maps.

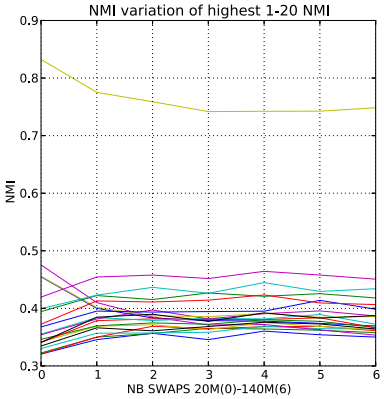

Fig. 3. 20 highest NMI values vs. N_s, Etna.

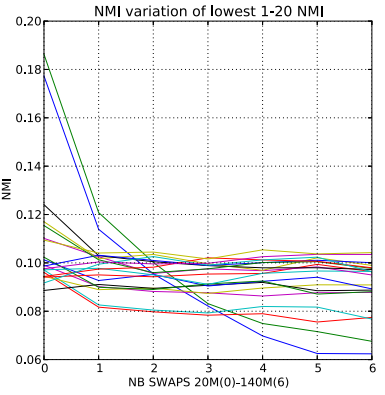

Fig. 4. 20 lowest NMI values vs. N_s, Etna.

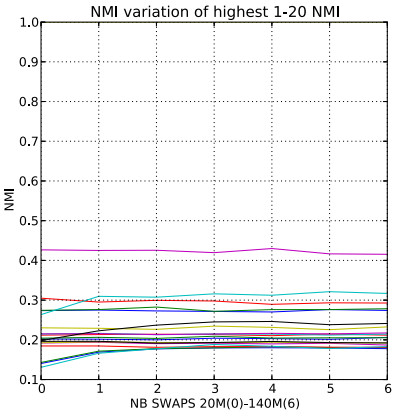

Fig. 5. 20 highest NMI values vs. N_s, NC.

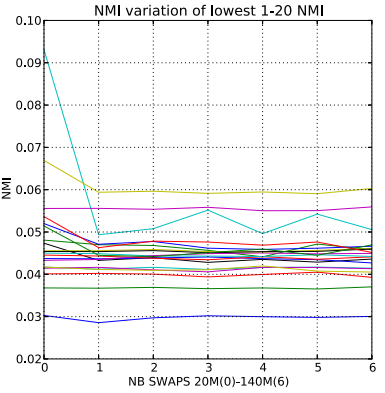

Fig. 6. 20 lowest NMI values vs. N_s, NC.

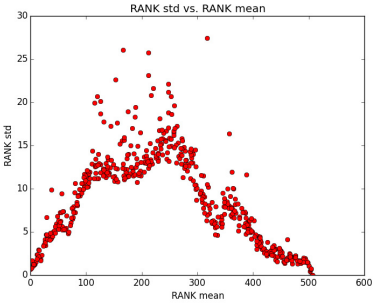

Fig. 7. Rank std. vs. rank mean, Etna.

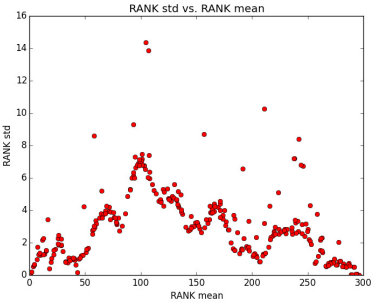

Fig. 8. Rank std. vs. rank mean, NC.

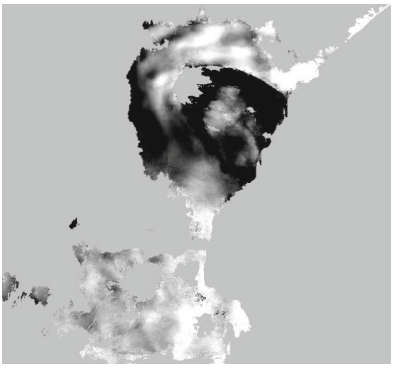

Fig. 9. Total phase delays, from negative values (black) to positive values (white), 2003/01/22, Marie-Pierre Doin, Etna.

Fig. 10. NDVI, from low values (black) to high values (white), 2004/01/13, Bluecham S.A.S., NC.

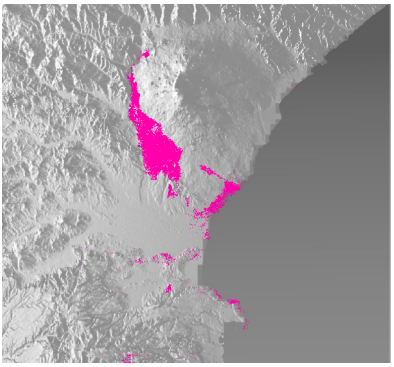

Fig. 11. STL-map: 1^{st} lowest NMI pattern $\langle 1,1,2,1,1,1,1,3 \rangle$, Etna.

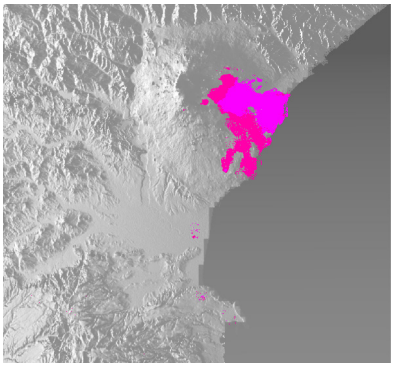

Fig. 12. STL-map: 1^{st} highest NMI pattern $\langle 1,2,3,3,3,3,3,3,3,3,3,3,3,3,3,3 \rangle$, Etna.

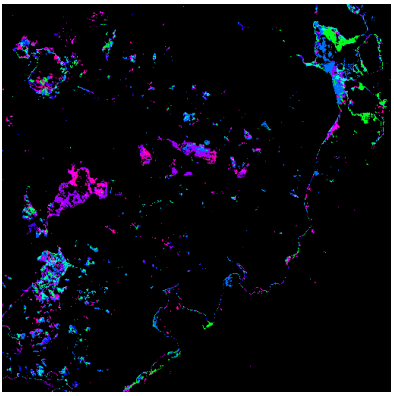

Fig. 13. STL-map: 6^{th} lowest NMI pattern $\langle 2,2,1,1,1,2 \rangle$, NC.

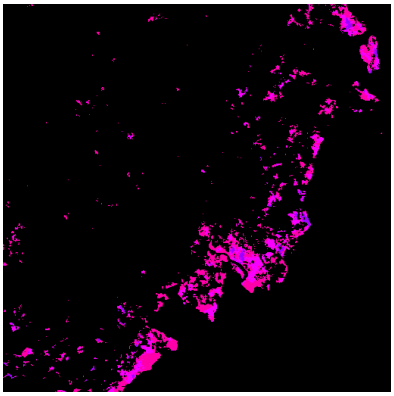

Fig. 14. STL-map: 2^{nd} highest NMI pattern $\langle 3,3,3,3,3,3,3,3,3,3,3,3,3,3,3 \rangle$, NC.

Fig. 15. Color scale: from the SITS starting date in red to the SITS ending date in violet.

Regarding qualitative results, it is possible to extract known and unknown meaningful phenomena, at both ends of the NMI rankings and for both datasets. Different STL-maps, representative of the well ranked ones, are shown in figures 11-12 (for the Etna SITS) and in figures 13-14 (for the NC SITS). Pixels where there is no occurence of the pattern are represented in black for NC and in gray for Etna (depicting a digital elevation model available for the area). The color scale used to represent the occurrence dates is given Fig. 15. In Fig. 11, pattern ⟨1,1,2,1,1,1,1,3⟩ (1^{st} lowest NMI pattern) shows, at the foot of the volcano, a zone moving towards the satellite before going away from the satellite (for this SITS the location of the satellite is on the left side of the image). It matches a sedimentary zone that is affected by movements due to subduction plates. In Fig. 12, pattern ⟨1,2,3,3,3,3,3,3,3,3,3,3,3,3,3⟩ (1^{st} highest NMI pattern) denotes a short motion towards the satellite and then a very long motion away from the satellite. It covers a part of the east flank of the volcano, called the *Valle del Bove*, which is known to be slipping into the sea. In Fig. 13, pattern ⟨2,2,1,1,1,2⟩ (6^{th} lowest NMI pattern) traces losses of vegetation due to anthropic activities (mining area at center and middle-left, mining facilities bottom-left). It also uncovers the impact of drought on a lakeshore (top-left) and exhibits sediment deposition (top-right). Notice that the color scale shows clear differences among the dates of occurrence of the phenomena. In Fig. 14, the simple pattern ⟨3,3,3,3,3,3,3,3,3,3,3,3,3,3,3⟩ (2^{nd} highest NMI pattern) locates dense vegetation along the coastline. The STL-maps obtained on the NC SITS are commercialized through the web-based decision support system operated by Bluecham S.A.S (Qëhnelö plateform www.yate.nc). Finally, the fact that encouraging results are obtained for very different datasets (radar or optical, different spatiotemporal resolutions and different rates of swappable events) shows the general nature of the approach.

6 Conclusion

This paper extends the swap randomization of Boolean matrices to the swap randomization of a base of sequences representing a Satellite Image Time Series (SITS). The proposed approach is aimed at assessing spatiotemporal patterns extracted from SITS. It preserves event frequencies, spatially and temporally, while breaking event connectivity and ordering. Once swap randomized datasets are generated, patterns are ranked using the Normalized Mutual Information (NMI). Low NMI patterns underline singular phenomena that are unlikely in randomized datasets while high NMI patterns express prominent phenomena that cannot be destroyed via swap randomization. Experiments on an optical

and a radar SITS evidence the stability of the swap randomization approach and its ability to explore a lot of different datasets. They also confirm that efficiency can be achieved by considering a single swap randomized dataset. Since the method is made as unsupervised as possible, extracted patterns allow to explore known and unknown phenomena, which gives access to different application domains ranging from agricultural monitoring to crustal deformation monitoring. Results regarding soil erosion monitoring are already commercialized. Future work include handling multispectral SITS, building clustering on top of extracted patterns and pushing NMI constraints within the extraction process.

References

1. Agrawal, R., Srikant, R.: Mining sequential patterns. In: Proceedings of the Eleventh International Conference on Data Engineering, Taipei, Taiwan, pp. 3–14, March 1995
2. Besag, J.: Markov chain monte carlo methods for statistical inference (2004). http://www-users.mat.umk.pl/~wniem/SemMgr/besag_MCMC.pdf
3. Besag, J., Clifford, P.: Generalized monte carlo significance tests. Biometrika **76**(4), 633–642 (1989)
4. Chuvieco, E., Huete, A.: Fundamentals of Satellite Remote Sensing. CRC Press, Boca Raton (2009)
5. Cobb, G.W., Chen, Y.: An application of markov chain monte carlo to community ecology. The American Mathematical Monthly **110**(4), 265–288 (2003)
6. Cover, T.M., Thomas, J.A.: Elements of Information Theory (Wiley Series in Telecommunications and Signal Processing). Wiley-Interscience (2006)
7. Doin, M., Lodge, F., Guillaso, S., Jolivet, R., Lasserre, C., Ducret, G., Grandin, R., Pathier, E., Pinel, V.: Presentation of the small baseline nsbas processing chain on a case example: the etna deformation monitoring from 2003 to 2010 using envisat data. In: Proceedings of the Fringe Symposium, pp. 3434–3437. ESA SP-697, ESA Communications, Frascati, September 2011
8. Fisher, R., Dawson-Howe, K., Fitzgibbon, A., Robertson, C., Trucco, E.: Dictionary of Computer Vision and Image Processing. John Wiley and Sons, New York (2005)
9. Gionis, A., Mannila, H., Mielikäinen, T., Tsaparas, P.: Assessing data mining results via swap randomization. In: Proceedings of the Twelfth ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, Philadelphia, PA, USA, pp. 167–176, August 2006
10. Gionis, A., Mannila, H., Mielikäinen, T., Tsaparas, P.: Assessing data mining results via swap randomization. TKDD **1**(3) (2007)
11. Gonzalez, R.C., Woods, R.E.: Digital Image Processing, 3rd edn. Prentice-Hall Inc., Upper Saddle River (2006)
12. Good, P.: Permutation tests : a practical guide to resampling methods for testing hypotheses. Springer series in statistics. Springer, New York (2000)
13. Gueguen, L., Datcu, M.: Image time-series data mining based on the information-bottleneck principle. IEEE Trans. Geoscience and Remote Sensing **45**(4), 827–838 (2007)
14. Guttler, F., Ienco, D., Teisseire, M., Nin, J., Poncelet, P.: Towards the use of sequential patterns for detection and characterization of natural and agricultural areas. In: Laurent, A., Strauss, O., Bouchon-Meunier, B., Yager, R.R. (eds.) IPMU 2014, Part I. CCIS, vol. 442, pp. 97–106. Springer, Heidelberg (2014)

15. Hanhijärvi, S., Ojala, M., Vuokko, N., Puolamäki, K., Tatti, N., Mannila, H.: Tell me something I don't know: randomization strategies for iterative data mining. In: Proceedings of the 15th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, Paris, France, pp. 379–388, June-July 2009
16. Honda, R., Konishi, O.: Temporal rule discovery for time-series satellite images and integration with RDB. In: Siebes, A., De Raedt, L. (eds.) PKDD 2001. LNCS (LNAI), vol. 2168, p. 204. Springer, Heidelberg (2001)
17. Julea, A., Méger, N., Bolon, P., Rigotti, C., Doin, M., Lasserre, C., Trouvé, E., Lazarescu, V.: Unsupervised spatiotemporal mining of satellite image time series using grouped frequent sequential patterns. IEEE Trans. Geoscience and Remote Sensing **49**(4), 1417–1430 (2011)
18. Julea, A., Méger, N., Rigotti, C., Trouvé, E., Jolivet, R., Bolon, P.: Efficient spatiotemporal mining of satellite image time series for agricultural monitoring. Trans. MLDM **5**(1), 23–44 (2012)
19. Luo, C., Chung, S.M.: Efficient mining of maximal sequential patterns using multiple samples. In: Proceedings of the 2005 SIAM International Conference on Data Mining, SDM 2005, Newport Beach, CA, USA, pp. 415–426. SIAM, April 2005
20. Pei, J., Han, J., Wang, W.: Constraint-based sequential pattern mining: the pattern-growth methods. J. Intell. Inf. Syst. **28**(2), 133–160 (2007)
21. Petitjean, F., Inglada, J., Gançarski, P.: Satellite image time series analysis under time warping. IEEE Trans. Geoscience and Remote Sensing **50**(8), 3081–3095 (2012)
22. Rigotti, C., Lodge, F., Méger, N., Pothier, C., Jolivet, R., Lasserre, C.: Monitoring of tectonic deformation by mining satellite image time series. In: Reconnaissance de Formes et Intelligence Artificielle (RFIA), Rouen, France, June 2014
23. Ryser, H.J.: Combinatorial properties of matrices of zeros and ones. Canadian Journal of Mathematics **9**, 371–377 (1957)
24. Vuokko, N., Kaski, P.: Significance of patterns in time series collections. In: Proceedings of the Eleventh SIAM International Conference on Data Mining, SDM 2011, April 28–30, Mesa, Arizona, USA, pp. 676–686 (2011)
25. Zaki, M.J.: Sequence mining in categorical domains: Incorporating constraints. In: Proceedings of the Ninth International Conference on Information and Knowledge Management, CIKM 2000, pp. 422–429. ACM, New York, November 2000