Algorithms for Climatic Patterns: survey and comparative study
Marco Miguel Guerreiro Calapez
Instituto Superior Técnico – Technical University of Lisbon
Av. Rovisco Pais 1, 1049-001 Lisboa
mmgc@ist.utl.pt

Abstract

The large amounts of collected climatic data in the Earth offer a unique opportunity to foresee and prevent ecological problems that may occur in the future. As a particular case of complex data, the analysis of climatic data (usually expressed as time series) requires efficient mining tools, either general or dedicated ones to this specific problem. The specificities of this kind of data also demand the existence of effective pre-processing techniques.

In this paper, we present a survey and a comparative study of the main different approaches to analyze climatic data, including pre-processing techniques. The algorithms considered range from generic algorithms for sequential and transactional pattern mining, to specific algorithms for dealing with temporal factors.

1. Introduction

Climatic data collected by terrestrial observation, satellites and produced by ecosystem models, offer a unique opportunity to predict and prevent ecological problems in the future, improving the management of the ecology and planet health. Among faced challenges is the projection of regional climate changes, and the prediction of their effects in those and other regions. This type of data is usually expressed as time series, i.e., numeric sequences with elements collected at regular time intervals. Examples of them include atmospheric, terrestrial and oceanic variables (as the sea surface temperature, precipitation or the NPP – the carbon photosynthetic accumulation in vegetables). Despite time series is the best well-studied form of complex data, the exploration of climatic time series with data mining techniques is applied seldom, and, to our knowledge, there is no survey about the application of those techniques in this context.

One of the mining techniques used to analyze climatic data is pattern mining. Its goal is to identify the set of events that occur together or in a specific order, a significant number of times. In the context of climatic data, pattern mining aims at finding the sequence of climatic values that are collected at regular time intervals (usually daily), in a particular local (or in a set of locals), a significant number of times.

On the last years, some algorithms were used and proposed to mine climatic data, all facing the challenge of dealing with time series. The problem is related to the fact that most of the existing algorithms work with non-numeric data, which requires that pre-processing has to be done to put the data ready for mining.

This paper presents an overview of proposed and used algorithms for the identification of climatic patterns, describing the experimental results obtained by comparing the different methods, when applied to climatic data. Since the success of those methods largely depends on the data pre-processing, the paper also presents a succinct overview and comparison of the pre-processing techniques to transform climatic data, giving a particular emphasis to seasonality removal and discretization ones.

The rest of the paper is organized as follows: in section 2, we present the main problems associated with pattern mining over climatic data, describing the different methods proposed. Next, in section 3, the description of pre-processing techniques and its comparison is done. Experimental results about mining algorithms are presented in section 4, where the datasets, methodology and experiments setup are also explained. In the end of the paper, we draw some conclusions taken from these experiments.

2. Pattern Mining in Climatic Data

Data mining applied to climatic data consists of two main components: modeling ecological data (such as modeling the events and modeling different zones of the planet) and searching for spatiotemporal patterns, which includes the discovery of association rules and forecast models. Those rules and models are relevant because they may contribute to the discovery of relations between extremes in weather with natural disturbances, which may modify the usual functioning
of the ecosystem. In this set of natural disturbances we may include fires, hurricanes, floods, droughts, lava flows, ice storms, and other disturbances caused by Humans such as deforestation, pollution and the introduction of non-native species to an area.

Data about natural occurrences are usually expressed as time series, in which, each series reference a location in the planet. For example, \( \{1.24; 1.15; 0.91; 1.05; 1.46; 1.56; -0.18; 0.17; 0.04; -0.19; -0.54; -0.43\} \) is a time series that presents the value of the standard precipitation index from January to December of 1999 collected in Nebraska. The interest of analyzing this kind of data is to find events that occur relatively close in time and that belong to a class of episodes. With an episode being a collection of events (measures) collected in a particular order at a particular time interval, also called sliding window [13]. These episodes are described as serial if the predicate of each event has a fixed order, or in parallel if the events order are not specified.

Usual forms to approach this kind of data is converting these time series into episode sequences, and then apply pattern mining techniques to discover interesting patterns. From this, two questions arise: ‘how to make the conversion?’ and ‘how to discover the patterns?’.

Next, those questions are answered, with the description of the different proposed methods.

### 2.1. Pre-processing

Pre-processing of time series is mandatory, since they usually suffer from three problems: the existence of outlying points and noise in data, the existence of differences on amplitude (scaling problems) and the existence of time distortions (translation problem) [3]. Climatic time series, by their side, have two additional characteristics: the existence of seasonality and auto-correlation among close events [11]. Since generic problems have been explored extensively, only the techniques for dealing with seasonality and numerical data are described next.

#### 2.1.1 Dealing with seasonality

Due to its nature, climatic patterns are usually dominated by the presence of seasonal variations. Although annual patterns as spring, summer, autumn and winter and the rains/dry seasons are important, they are already well known from all of us. Thus, we are interested, in first place, on patterns that represent deviations from the usual patterns of seasonality [20]. The seasonal adjustment is used in the analysis of temporal series in order to remove the periodic component of well known patterns of the temporal series observed. The reason is the fact that great periodic variations in the temporal series can hide weaker but sufficiently important variations. Since one of the priorities is the detection of deviations from the usual, ones which is difficult due to the strength of the seasonal patterns in the data, it is necessary to remove them so that other patterns, more interesting, can be detected. (That seasonality doesn’t have to be related with annual seasons. Any periodic component with a known period can be removed from the temporal series.)

There are several transformations that can be done to remove the seasonal variation, and experimental results show that the majority of them are effective and allow for the discovery of interesting deviations. It is also to be noted, that this removal has the advantage that it also removes large part of existing autocorrelation [11] , [12].

**Filtering based on Discrete Fourier Transform.** This approach is based on a standard signal processing technique – the Discrete Fourier Transform. By taking the discrete Fourier transform, the original time series is transformed from the time domain into the frequency domain. In climatic patterns, removing this yearly component and then performing the inverse Fourier transform yields a new time series, which would be free of any seasonal component.

**Monthly-Z score.** This transformation takes the set of values for a given month, e.g., all Januarys, calculates the mean and standard deviation for that, and then standardizes each value by calculating its Z score. This is calculated by subtracting off the mean and dividing by the standard deviation. While this approach seems similar to the first approach, it is actually quite different since it uses the monthly mean and standard deviation, instead of the overall mean and overall standard deviation.

Put another way, we express each data value in the time series in terms of its deviation from the mean value for its corresponding month, scaled by the volatility factor for that month. The month-by-month rescaling used in this transformation causes seasonal fluctuations to disappear. Furthermore, scaling by the monthly standard deviation makes the changes more pronounced for those months in which the volatility is lower.

**Moving average.** A 12-month moving average is effective at removing seasonality and it also smoothes the data. For each value of the time series that happened at least one year after the beginning of the time series, it consists on calculating the sum of the values during the time window of one year and divides that value by the number of observations done during
Apart from those transformations, others are possible to deviations to its neighbouring points in time. However, it tends to flat any deviation from the average values by spreading the effects of the deviations to its neighbouring points in time. Apart from those transformations, others are possible to be done, like the Singular Value Decomposition [18].

2.1.2 Dealing with numeric data
Another change needed when working with time series, in particular for pattern mining purposes, is the translation of numeric elements to symbolic ones. This is due, since the majority of existing pattern mining algorithms are not prepared to deal with numeric data. Indeed, it is often beneficial to aggregate data into a small number of points, easing computational requirements and (typically) reducing the amount of noise. However, it can be difficult for researchers to choose the proper level of aggregation, since too much limits the patterns that can be detected, while too little results in noisy data in which only the strongest patterns can be discovered.

One way to improve the performance of the discovery methods is to use segmentation techniques, to split the data in useful groups and with a particular meaning. Particularly, in the identification of regions of the Earth which constitute points with similar characteristics in a short and long term. Through the analysis of the correlations between the climatic variables in those regions, it is possible to rediscover existing patterns, as well as new and previously unknown correlations.

There are four main approaches to discretize time series into symbolic event sequences, taking into consideration their own values: the ones based on the use of standard deviation, the ones based on the use of entropy, the ones based on clustering and the ones made by hand.

Standard deviation based. The event can be defined has an unexpected low/high value in the temporal series if it has a certain difference to the standard deviation of the mean of the series.

Entropy based. It finds the best split in order to the majority of values in the segment that has the same class name. It is known to find the split with the maximum information gain.

Clustering: It tries to partition the dataset into subsets, such the data in each subset is closer to other elements in the same cluster, than to other elements.

Manual segmentation. It allows the user to select the segmentation he wants to apply to each set.

2.2. Pattern mining in climatic data

To discover patterns it is necessary to calculate and to represent the influence that an event has in the occurrence of a different one. Thus, patterns characterize how much the presence of an item set in the data (set of events for climatic data) implies the presence of some another distinct itemset in the same registers [2]. These patterns are usually expressed as association rules, represented as an implication \( X, Y \rightarrow Z \), where \( X \) and \( Y \) are called antecedents and \( Z \) the consequent, all defined by disjoint sets of items. Two parameters are usually defined before its exploration: the support, which indicates the frequency or probability that antecedents occur together in the transactions, and the confidence that join the support with the total number of transactions where the first antecedent occurs.

The analysis of the associations can be used to predict spatial and temporal relations hidden in the Earth scientific data. The objective of this analysis is to extract significant patterns, in rules format or in a collection of events that will predict the occurrence of a certain kind of event, based in the occurrence of other events.

Due to the spatiotemporal nature of the collection of scientific data, in general, and of climatic data, in particular, there are four types of patterns that can be found: non-temporal, temporal, spatial and spatio-temporal patterns [12]. Next, we describe the approaches to non-temporal and temporal analysis of climatic data, non-considering spatial methods.

2.2.1 Finding non-spatiotemporal patterns
In the simplest case, we can look for non-sequential intra-zone associations between the events that occur in the same spatial location, without consider the temporal aspects of the data.

In this context, time series are transformed into transactions, and then standard algorithms for association analysis like Apriori [2] or FP-growth [5] can be applied to extract the collection of rules from the transformed data.

The best well-known algorithm for transactional pattern mining is Apriori. This algorithm uses a breadth-first search and a hash tree structure to count candidate item sets efficiently. It generates candidates of length \( k \) from item sets of length \( k-1 \). Then it prunes de candidates which have an infrequent sub pattern. After that, it scans the database to determine frequent itemsets among the candidates.

FP-growth and TD-FP-growth algorithms follow a different approach. They mine frequent patterns without generating candidates, basing all the process in...
a tree containing all the transactions. This tree describes the existing database, which have the advantage of being complete and compact, because infrequent items are removed and the items are ordered by their frequency. The difference between both is the way they construct the tree. While FP-growth always start from the less frequent item, the TD-FP-growth follow the opposite approach, starting from the most frequent one.

Despite the simplicity of the approach, however, some of the transformations create dense columns in the transaction matrix, causing an exponential growth of computational requirements. Moreover, there is no minimum measure of interest that reflects the quality of the derived patterns. So it has been discussed that this approach to find association rules and sequential patterns is not adapted to find all the interesting patterns due to the spatiotemporal nature of the data, showing us that this exploration is not easy and it is evidenced by the problems that can appear during some stages of the data mining analysis [5].

Once patterns are discovered, it is difficult to distinguish the significant ones from the others. For example, given 40,000 time series recording sea surface temperature at various points on the ocean’s surface and 60,000 time series representing precipitation on land, some of these series may have strong correlations. While a number of statistical approaches estimate significance levels, it is not possible to apply such approaches directly due to spatial and temporal autocorrelation. When genuine patterns are identified, domain-specific knowledge is inevitably still needed to identify patterns of interest to Earth scientists.

In order to surpass this situation, VARGA algorithm follows an apriori-based approach, but defining input measures as vectors, where each field represents an observed variable. Due to the regularity of these vectors, apriori can be modified in order to take same advantage from this characteristic resulting in the VARGA algorithm [7].

The development of this algorithm serves two main goals. The first one is to efficiently deal with the large amount of vectors that satellites remotely collect. The second one is the fact that some of the most interesting association rules can be those with low support, that’s because they can identify interesting but rare phenomena. The algorithm tries to minimize the needed memory to do the mining task and has also the useful characteristic that its memory requirements are not variable according to the selected support. For an n-element vector, the list of n-item collection is just a collection of unique vectors. Each new vector is compared with the existing ones and if it does not exist yet, it is added; otherwise the counter for that vector is incremented. The sum of all counters means the total number of vectors that are being mined, while the counter associated to each vector, split by the total number of original vectors is called support. After all n-collection have been identified and counted, the association rules can be mined.

A second approach is the one proposed by Mannila [13] (as described below), but just considering serial episodes, and using a window width of one.

### 2.2.2 Finding non-spatiotemporal patterns with taxonomies

A way to surpass the difficulties in the analysis of the rules is the use of taxonomies. These structures may reflect a characterization of how an item can be hierarchically classified, assisting the user in the identification of interesting and useful information, among the great amount of generated rules. Each algorithm for transactional pattern mining can be extended for using taxonomies, which results in the existence of several different techniques. In this text, we had only considered Stratify [17] (that follows an apriori-based approach), Top-Down FP-Tax and Bottom-up FP-Tax (a version of Fpgrowth that uses taxonomies) [15]. There are much more algorithms for this purpose, such as: Cumulate [17], Genex [19] or Prutax [8], however they generate all the possible association rules and then use some interestingness measure to remove the less interesting rules.

### 2.2.3 Finding temporal patterns

If the temporal data is not discarded, then it is possible to mine temporal relations between events that occur in the same location using sequence pattern discover algorithms, such GSP [16] or GenPrefixSpan [4] and dedicated ones, like the one proposed by Mannila [13]. In generic sequential pattern mining approaches, each data sequence corresponds to an ordered list of events, with an event being an itemset. In climatic data, a sequence could for example describe the set of different measures (temperature, humidity and wind) collected in some place at regular time intervals. In this context, a pattern is frequent if it is present in a significant number of event sequences, with each sequence counting at most one time for the support of the pattern. That way, the support and confidence measures are only dependent of the number of spatial locations for which the patterns were observed. This kind of exploration strategy may not be the most appropriated one, because it doesn’t take in consideration the number of times the pattern occurs in each location (each sequence). Note that large time
series usually has negative or close to zero correlation values, which requires their translation to event sequences in the case of climatic data. In the Mannila approach [13], the algorithm performs a level-wise (breadth-first) search in the class of episodes following the sub episode relation. The search starts from the most general episodes, i.e., episodes with only one event. On each level, the algorithm first computes a collection of candidate episodes, and then checks their frequencies in the event sequence. This approach is based on the lemma that states if an episode is frequent in an event sequence, then all his sub episodes are frequent.

2.2.4 Finding temporal patterns with constraints
Methods for discovering temporal patterns can be adapted to incorporate some constraints, in order to improve both their efficiency and efficacy. In particular, both algorithms described next look for patterns expressed as rules, with a specific consequent in mind.

The REAR algorithm (Representative Episodial Association Rules) [6] tries to find relationships that occur within a defined time interval. It can be used to find both parallel and serial patterns, but for serial patterns it requires that the consequent do not start earlier than the antecedent. The order of the occurrences of items inside the antecedent and the consequent are irrelevant. The MOWCATL algorithm (Minimum Occurrences With Constraints and Time Lags) [6] tries to find relations when there is a time interval between the antecedent and the consequent. It can also be used to find parallel or serial episodes, being their biggest difference, the fact that we define different window sizes for the antecedent and the consequent and a time lag. This time lag defines the time (it can be a fix time or a maximum time) that exists between the beginning of the antecedent and the beginning of the consequent, with the restriction for serial patterns that consequent cannot start before the end of the antecedent. The algorithm just makes one pass all over the database, using the read windows where each occurrence occurs. This was impossible to be implemented in this study due to memory requirements for more than 18,000 windows.

3. Comparative Study
The decision of which algorithms to include in this study followed the goal of comparing dedicated methods with generic ones. In this manner, we have implemented all the described algorithms in the Weka package [21], sharing the same structures whenever possible. The comparison of pre-processing mechanisms followed the same approach.

Datasets: The study was conducted using the data collected for the region of Maastricht from 1956 to 2005. The datasets include 18050 records composed by eight attributes each: Precipitation, Sunshine time, Pressure, Snow level, Humidity, Maximum Temperature, Minimum Temperature and Cloud Cover. This dataset overcomes the mean size of datasets (10000 time series) used in identical studies [9]. This dataset was pre-processed with the different pre-processing techniques described, by applying the same technique to each attribute. Since all mining algorithms only work with non-numeric data, we combined each pre-processing technique of discretization (all of them getting five disjoint groups) with each seasonality removal technique. From these combinations, we derived twelve different datasets. This number is 6 times larger than the average of 1.85 datasets usually used to test the efficiency of machine learning algorithms [9].

Experiments: The computer used to run all the experiments was a NEC Pentium IV 3.0 GHz with 1Gb of RAM. To make the time measurements more reliable, no other application was running on the machine while the experiments were running. The operating system used was Windows XP and the algorithms were implemented using the Java programming language (Java Virtual Machine version 1.4.2_01). The datasets were maintained in main memory during the algorithms processing, avoiding hard disk accesses.

For non-temporal algorithms with and without taxonomies, we tested each of them in each of the 12 datasets using a value for confidence of 0.0 and a value for the support that varies from 0.1 to 0.001. For temporal algorithms all the 12 datasets were used too, with the same value of confidence and support values ranging from 0.2 to 0.08. For the GSP [16] and GenPrefixSpan [4] algorithms the only measure used was the minimum temperature and for each dataset, its data was split according to its months in order to have several datasets for the same location. The maximum distance between two events was set to zero.

The REAR [6] and MOWCATL [6] algorithms used a window size of two days (the datasets are composed of daily records), constrained by a consequent involving the minimum temperature.
3.1. Comparing pre-processing techniques

Experiments show some discrepancies on the application of discretization methods. The transformations based on standard deviation and clustering give higher relevance to the records closer to the mean. This implies that they recognize more records as normal values and less as extreme ones, which probably should be the usual.

The entropy based discretization almost split the data into classes with the same number of elements. This fact has a huge influence in the number of patterns found. Since each class is split equally along the database it gets more difficult to find patterns using the same support. With a higher concentration of values in the centre classes there is a possibility to find more patterns, most of them as a result of the cross of that “centre” attributes.

Between these methods there are not relevant differences, both allowing for the discovery of more or less the same amount of patterns.

In terms of seasonality removal, experiments suggest that if the high frequency fluctuations in the original time series are factored out, then the 12-month moving average of the original time series should be quite similar to the monthly Z score time series.

3.2. Comparing mining algorithms

3.2.1 Mining non-temporal patterns

When comparing the performance of transactional pattern mining algorithms applied to our datasets, experiments confirm the usual results: algorithms without candidate generation are the fastest on the presence of very low values of support. With that difference specially noted for supports below 0.09, as we can see in Figure 2.

In terms of memory, we see that they have similar memory consumptions. This is explained by the fact that while Apriori need to store candidates, FP-Growth has to keep conditional trees in memory.
Comparing both Apriori based algorithms, Apriori and VARGA, we can see that for certain datasets like DFT with discretization based in Standard Deviation and for supports below 0.01, the VARGA algorithm can be faster than Apriori. Note that VARGA has constant times for each dataset, since it always generates all the possible patterns for each dataset, including non-frequent ones. This becomes a problem when the dataset is sparse, with few frequent patterns. By considering lower supports, the performance of VARGA algorithm becomes more acceptable.

In terms of memory consumption, it is easy to understand why VARGA presents the worst results: it always generate all possible patterns, so it will need more memory than Apriori, which only generates patterns that potentially have the minimum support required.

At last, and as expected, experiments confirm that the Mannila algorithm was not adequate for this task: it generates the candidates, but the way it reads the patterns from the dataset is extremely different compared to the way Apriori based algorithms does it. For windows of size 1, it presents very low performance because it reads each record in the dataset twice: once to add it to the counter and another to release it. This algorithm is good when we want to find patterns that occur during a window larger than one.

3.2.2 Mining non-temporal patterns with taxonomies

In the presence of taxonomies, results are similar. We used a simple taxonomy, just consisting of two
levels (Figure 4).

Tree based algorithms can find patterns using less time than the one that generates candidates, being the difference noted when we pass the 0.1 support barrier. Also, like for algorithms without taxonomies, Stratify algorithm can be more memory efficient than tree based ones. This was expected, since comparing memory usage of Apriori and Stratify, there is not much difference. However, the differences between FP-Growth and FP-Tax are notorious. Using taxonomies the size and the depth of the trees will be higher which have memory costs.

### 3.2.3 Finding temporal patterns

Comparing GSP and GenPrefixSpan we can see that they present similar performances in time and memory for the majority of the datasets, being the major difference when we are running them in datasets where exist long sequences. In that case, GSP algorithm has a worst time performance (8 times more time used than GenPrefixSpan) but it shows a little better performance than GenPrefixSpan, in terms of memory consumptions (see Figure 5).

Note that Mannila method does not return the same patterns as GSP or GenPrefixSpan: while Mannila finds event sequences that occur in serially inside a time window repeatedly over a unique sequence, GSP and GenPrefixSpan find event sequences that occur frequently in different sequences of the dataset. For the used memory, Mannila is an extremely quick algorithm to find rules and can be used when we want to find serial patterns in a limited time.

### 3.2.4 Finding temporal patterns with constraints

As seen above, REAR and MOWCATL algorithms can
be constrained to find the same rules, we just have to set the time lag between antecedent and consequent to zero, and adjust both antecedent and consequent windows. However, experiments show that they have significant differences in performance.

As we can see in Figure 6, REAR is usually much more efficient in time, and mostly better in terms of memory consumption. While REAR starts looking for the patterns using the antecedent and the consequent simultaneously, MOWCATL uses them separately just joining them in the end (in the Episodes generation phase), which creates a lot of candidate rules that are cut by REAR in the beginning. That way the decision about the usage of one of them is extremely dependent of the type of dataset used.

4. Conclusions

The amount of climatic data collected nowadays, offers a unique opportunity to predict and prevent ecological problems in the future, with a better management of ecology and the planet health. To that purpose a lot of techniques exist to help us on finding interesting rules for several types of situations. There are no good and bad techniques. Each technique is useful when properly applied to find the rules they were meant to find.

In this paper we have identified several techniques that can be applied to extract rules when we do not want to incorporate spatial nor temporal information, in other words, when we want to relate events that occur at the same time in the same location, being the tree based algorithms more efficient when we just have time concerns and Apriori based when we also have memory concerns.

Techniques with the same goal but with some domain knowledge incorporated using a hierarchical characterization, present the same results as those that do not use taxonomies.

We also evaluated algorithms where temporal information was added and there, we did not find better and worst algorithms, too. Each of them has their main goals and when we use them for that purpose they are definitely the best. To find patterns inside a window we should use Mannila; when we want event sequences that occur with a maximum distance between them we should use GenPrefixSpan; and when we want to find patterns with a well defined consequent, we should use REAR (when there are no time lag between them) or MOWCATL (when a time lag exists between the antecedent and the consequent).

We have also analyzed several ways to remove seasonality and to discretize climatic data. All the seasonal removal methods have their benefits with DFT and Monthly-Z score presenting very similar results.

For discretization methods, experiments show that on our datasets, the best results were derived from the use of clustering and the discretization based on standard deviation. They both keep together the sets of events that are close to each other, identifying as extreme ones those that are really far from the mean value. The discretization based on entropy allows for finding fewer patterns, but this is mainly due to the fact that with continuous series the algorithm split the data in groups without significant difference between the number of points that are part of each group, which causes a lower number of patterns founded for small supports.

5. References


[5] J. Han, J.Pei, Y.Yin: Mining Frequent Patterns without Candidate Generation, In Proc. 2000 ACM-SIGMOD Int. Conf. on Management of Data (SIGMOD’00), 2000


Knowledge Discovery (PKDD’98, Nantes, France), 74-82, LNAI 1510, Springer, Heidelberg, Germany 1998


[10] Krzysztof Koperski, Jiawei Han; Discovery of Spatial Association Rules in Geographic Information Databases, 1995


[14] Jian Pei, Jiawei Han, Behzad Mortazavi-Asl, Helen Pinto, Qiming Chen, Uneshwar Dayal, Mei-Chun Shu: Prefix Span: Mining Sequential Patterns Efficiently by Prefix-Projected Pattern Growth in Proc. Int. Conf. Data Engineering, Heidelberg, Germany, April 2001


