TiP: Analyzing Periodic Time Series Patterns

Thomas Bernecker, Hans-Peter Kriegel, Peer Kröger, and Matthias Renz

Institute for Informatics, Ludwig-Maximilians-Universität München
Oettingenstr. 67, 80538 München, Germany
{bernecker,kriegel,kroeger,renz}@dbs.ifi.lmu.de
http://www.dbs.ifi.lmu.de

Abstract. Time series of sensor databases and scientific time series often consist of periodic patterns. Examples can be found in environmental analysis, where repeated measurements of climatic attributes like temperature, humidity or barometric pressure are taken. Depending on season-specific meteorological influences, curves of consecutive days can be strongly related to each other, whereas days of different seasons show different characteristics. Analyzing such phenomena could be very valuable for many application domains. Convenient similarity models that support similarity queries and mining based on periodic patterns are realized in the framework *TiP*, which provides methods for the comparison of similarity query results based on different threshold-based feature extraction methods. In this demonstration, we present the visual and analytical methods of *TiP* of detecting and evaluating periodic patterns in time series using the example of environmental data.

1 Introduction

In a large range of application domains, e.g. environmental analysis, evolution of stock charts, research on medical behavior of organisms, or analysis and detection of motion activities, we are faced with time series data featuring activities which are composed of regularly repeating sequences of activity events. For that purpose, existing periodic patterns that repeatedly occur in specified periods over time have to be considered. Though consecutive motion patterns show similar characteristics, they are not equal. We can observe changes of significant importance in the shape of consecutive periodic patterns.

TiP utilizes the dual-domain representation of time series and the threshold-based approach [1] to extract periodic patterns from them. Beyond the interest of [2], the temporal location and the evolution of consecutive patterns are focused. For efficient similarity computation, relevant feature information is extracted so that data mining techniques can apply. By visualization, *TiP* provides first information about the existence and the location of periodic patterns in the time domains in the 3D space. Further knowledge about large datasets and the choice of adequate parameter settings for similarity queries and mining can be obtained by diverse analysis methods.

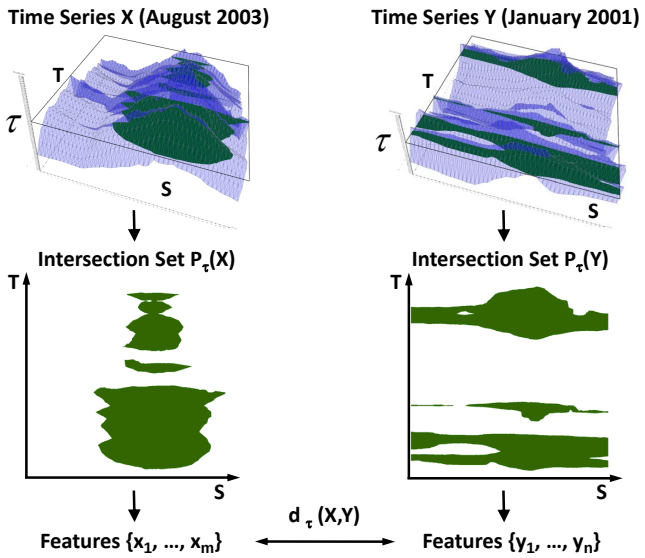


Fig. 1. Feature extraction and similarity computation on dual-domain time series.

Overall, *TiP* serves as a framework to effectively and efficiently manage dual-domain time series. Furthermore, it provides several methods to process similarity queries on this type of time series based on user-defined features as well as an intuitive graphical user interface. Theoretical background of the applied techniques in *TiP* will be given in Section 2. Details of the system and the architecture will be explained in Section 3. Section 4 describes the planned demo tour in more detail.

2 Theoretical Background

Time Series Representation A time series X can be split into a sequence of subsequences of fixed length by adding an additional time domain [1]. This yields a *dual-domain* representation having a 3D surface. A dual-domain time series represents the temporal behavior along two time axes. Consider an example of environmental research. The trend of temperature within one month of a year having one value for each hour of a day is then represented by the evolution of the temperature within one day (first time domain), and, additionally, over consecutive days of the entire time period (second time domain). Formally, a dual-domain time series is defined as

$$X_{dual} = \langle\langle x_{1,1}, \ldots, x_{1,N-1}, x_{1,N}\rangle, \ldots, \langle x_{M,1}, \ldots, x_{M,N-1}, x_{M,N}\rangle\rangle$$

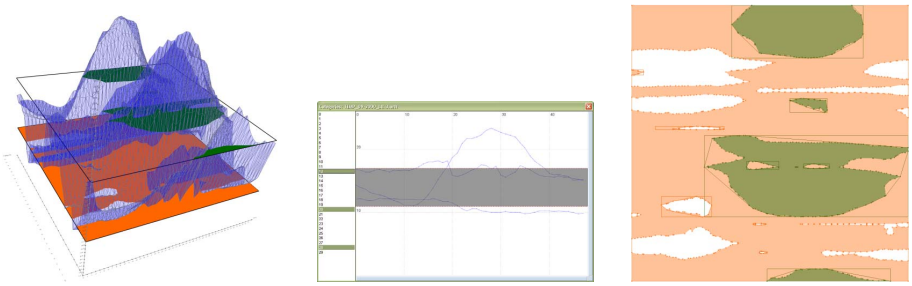
where $x_{i,j}$ denotes the value of the time series at time slot i in the first (discrete) time domain $T = \{t_1, \ldots, t_N\}$ and at time slot j in the second (discrete) time domain $S = \{s_1, \ldots, s_M\}$.

Time Series Similarity *TiP* compares dual-domain time series based on their periodic patterns that we call *intersection sets*. An intersection set $P_\tau(X)$ of a time series X is created by a set of polygons that are derived according to [2], where time series with one time domain are represented as a sequence of intervals w.r.t. to a threshold value τ. In this case, τ corresponds to a 2D plane which intersects the 3D time series X. The polygons of an intersection set represent the evolution of amplitude-level patterns as spatial objects, as they deliver all information about the periods of time during which the amplitudes of the time series exceed τ. In the temperature example, the polygons contain only values beyond a certain temperature level. Thus, the patterns emerge from temperature values that are greater than τ. However, patterns of summer and winter months are different in their occurrence and their dimensions, as depicted in Figure 1: the patterns of August 2003 are more concentrated within the days, but they are more constant over the whole month, whereas January 2001 contains patterns having a lower variance over consecutive days but that hardly change within the days. The degree of periodicity of a time series is reflected by the extent of the polygons, so even non-periodic patterns can be detected and addressed by their spatial location. The distance $d_\tau(X, Y)$ of two time series X and Y reflects the dissimilarity of the intersection sets $P_\tau(X)$ and $P_\tau(Y)$ for a certain τ. This step reduces the comparison of time series to comparing the sets of polygons such that spatial similarity computation methods can apply. To save computational cost while computing the distance as well as to allow for the usage of index structures like the R^**-tree* [3], *TiP* derives local features (e.g. approximations or numerical values) from the polygons and global features (e.g. characteristics) from the intersection sets and computes the distance based on them (cf. Figure 1). The number of polygons and, thus, the number of local features varies for different intersection sets, so *TiP* employs the Sum of Minimal Distances (SMD) measure [4]. The distance based on global features is simply calculated as the L_p-distance between the associated features, as, for each intersection set, *TiP* derives the same amount of global features. For similarity calculation and further analysis, intersection sets for different threshold values can be pre-computed for a dataset and stored in an underlying database. Thus, relevant feature information of intersection sets can simply be reloaded for similarity queries. Furthermore, this threshold-based method is very efficient as, contrary to simple measures like the Euclidean distance, it does not need to consider the originally high-dimensional structure of time series.

3 Architecture

TiP has been implemented using Java and the Java3D API. The framework is able to load datasets of time series following the ARFF format[1] into the application. For each time series, the user can set several threshold planes to compute intersection sets w.r.t. these threshold values. The polygons can be approximated by simple conservative bounds like minimum bounding rectangles

[1] http://weka.wiki.sourceforge.net/ARFF



(a) Dual-domain time series visualization.

(b) 2D view.

(c) *iView* displaying polygons and features.

Fig. 2. Visual exploration of dual-domain time series with *TiP*.

(MBRs) and thus be stored efficiently in the internal database by the support of internal index structures. This also accelerates similarity queries, as multi-step query processing can be applied. For a time series X, each intersection set $P_\tau(X)$ w.r.t. a specific value for the threshold τ does only have to be computed once. Once stored in the internal database, the ID of $P_\tau(X)$ can simply be associated with the ID of the time series X from which was derived when X is opened by *TiP* for another time. The user is able to perform an extensive evaluation using several techniques to mine databases of time series that are based on periodic patterns, such as k-nearest-neighbor (kNN) queries with distance ranking, precision-recall analysis and kNN classification.

TiP provides an intuitive graphical user interface. Exemplary elements are depicted in Figure 2. The core elements include two Java3D applets displaying two time series simultaneously in their dual-domain representation (cf. Fig. 2(a)). As several user-defined threshold planes can be set for each time series, *TiP* offers two additional applets called *iView* to depict the polygons and the presentable features (i.e. approximations) of the corresponding intersection sets. Additionally, the user is able to split a time series along each axis. Then each subsequence of a dual-domain time series can be displayed in an external 2D chart (cf. Fig. 2(b)).

4 Demo Tour

In this section, a short overview of the nature of the demonstration of *TiP* is given, which will be performed applying the example of environmental research where the time series are built of normalized temperature measurements. The recorded and prepared data, labelled corresponding to the four seasons of a year, was provided by the Environmental State Office of Bavaria, Germany[2].

[2] http://www.lfu.bayern.de/

In the demo, we first present how the dual-domain time series representation can discover a higher content of information compared to the original, sequential representation. For example, depending on the season, the temperature curves show different characteristics in their course, so representative patterns can already be derived visually from this 3D surface. When setting threshold planes arbitrarily, the user can materialize those patterns. By setting several threshold planes, the user can examine how to find suitable values for τ that should be used while performing similarity queries. In general, the higher a threshold is, the less is the number of amplitude values of a time series that exceed the threshold and, thus, contribute to a polygon of the intersection set. By defining a value or a range for τ, we show how to define the relevant time series values for the queries. A high τ value, e.g. $\tau = 80°\text{F}$, in the temperature example corresponds to the query: "Given the curve of a query month showing temperature values higher than 80°F on the first days at one certain time of day, please return all months that also show values higher than 80°F on their first days at this time of day.". If the query time series contains measurements of a summer month, it is likely to get summer months returned in the result set, as winter months show a different behavior, even if the amplitudes of the dataset are normalized. As an additional feature, the system supports the automatic detection of the value for τ that yields the best results without specifying any minimum query temperature level. For that purpose, a kNN classification is performed in order to find a τ value to obtain the best accuracy results. Exemplarily for the normalized temperature dataset, choosing $\tau = 0.6$ yields an accuracy value of 0.65, where by applying the Euclidean distance we get an accuracy of 0.56. To get an overview of the quality of the results that can be achieved with different values for τ, a parameter analysis shows the classification accuracies and average precision values for every possible τ within the amplitude range of the time series. Of course the result strongly depends on the selection of the features of the intersection sets. Therefore, we show that features can be combined to improve the expressiveness of the results.

References

1. J. Aßfalg, T. Bernecker, H.-P. Kriegel, P. Kröger, and M. Renz. Periodic pattern analysis in time series databases. In *Database Systems for Advanced Applications, 14th International Conference, DASFAA 2009, Brisbane, Australia, April 21-23, 2009*, pages 354–368, 2009.
2. J. Aßfalg, H.-P. Kriegel, P. Kröger, P. Kunath, A. Pryakhin, and M. Renz. Threshold similarity queries in large time series databases. In *Proceedings of the 22nd International Conference on Data Engineering, ICDE 2006, 3-8 April 2006, Atlanta, GA, USA*, page 149, 2006.
3. N. Beckmann, H.-P. Kriegel, R. Schneider, and B. Seeger. The R*-tree: An efficient and robust access method for points and rectangles. In *Proceedings of the 1990 ACM SIGMOD International Conference on Management of Data, Atlantic City, NJ, May 23-25, 1990*, pages 322–331, 1990.
4. T. Eiter and H. Mannila. Distance measures for point sets and their computation. *Acta Informatica*, 34(2):109–133, 1997.