Similarity Search in Multidimensional Time Series using the Coulomb's Law

Claudinei Garcia de Andrade, Marcela Xavier Ribeiro

Universidade Federal de São Carlos, Brasil
{claudinei.andrade, marcela}@dc.ufscar.br

Abstract. Due to technological innovation and lower production costs of data collecting instruments there has been a sharp increase in the amount of information available for analysis. Additionally, collected data withhold intrinsic relations within itself that cannot be realized without careful analysis, requiring the use of specific techniques to manipulate it. In this context, we propose a time series descriptor based on the principle of Coulomb's Law to perform similarity search over multidimensional time series. The proposed method is composed of a new time series extractor and a flexible module to perform similarity search in multidimensional time series. Moreover, we introduce the Coulomb method, which describes how to employ the proposed descriptor to perform similarity search over multidimensional time series. According to experiments performed over meteorological and medical databases, the proposed method promotes a stark reduction in feature vector sizes and achieves higher accuracy values when compared with other time series descriptors based on Fourier Transform and Brute-force solution.

1. INTRODUCTION

The amount of data available for analysis has increased to a great extent, making the manipulation of such data a great challenge. In addition, an isolated analysis alone may not bring on significant results. Therefore, one of the most important challenges of time series analysis is how to analyze multiple time series simultaneously – a multidimensional time series analysis. Another important challenge in this field is how to represent the series in a compact way while keeping its precision.

This article aims at proposing a descriptor that allows the representation of a series behavior while reducing data size without great loss of information. We also propose a similarity search method that, combined with the descriptor we propose, is able to perform similarity search in both one- and multidimensional time series from several fields of science. A number of descriptors can be found in the literature [Agrawal et al. 1993; Korn et al. 1997; Chan and Fu 1999; Keogh et al. 2001; Chakrabarti et al. 2002; Morinaka et al. 2001]. However, they have some limitations: in the process of dimensional reduction, fundamental features of the series behavior may be lost. Also, they offer no support to running similarity search in multidimensional time series. The Coulomb descriptor deals with these limitations. By employing a centroid measure as reference of the series behavior, the proposed method reduces the feature loss caused by the dimensionality reduction process. Our method employs a graph-based similarity search approach, called FM (flexible module), to allow the usage of multidimensional time series, dealing with the drawback of the previous methods that do not support multidimensional similarity search.

We proposed a new method called Tractable Similarity Searching (TSS) to perform similarity search in multidimensional time series. The new method is composed of: (i) a new descriptor, called Coulomb descriptor; and (ii) a flexible module (FM) to perform similarity search in multidimensional time series. In order to validate the proposed method (TSS), we made use of meteorological, medical and agricultural data as case studies and compared it with the baseline time series descriptors based on Fourier Transform [Agrawal et al. 1993] and Brute-force [Keogh 1997]. The experiment results demonstrate that the proposed method is well-suited to perform similarity search in multidimensional time series.

This article is organized as follows. Section 2 briefly shows the related concepts about similarity search in time series and the main related work. Section 3 explains the proposed method and Section 4 discusses experimental results. Finally, Section 5 presents conclusions.

2. RELATED WORK AND BACKGROUND

Representing series in a manner that simplifies knowledge extraction, that makes its computational manipulation easier and, also, that preserves most of the information from the original is one of the most important issues in the time series analysis field. In this section, we present important concepts related to time series analysis and related work.

A time series can be defined as an ordered series of observations [Wei 2006]. Formally, a time series is a set of observations $\{Y(t), t \, \epsilon \, T\}$ in which Y is the variable of interest and T is an index set. A multidimensional time series T_m of length n is a sequence of m set of real-value variables [Tanaka et al. 2005]. It can be represented as $T_m = (x_{11}, ..., x_{m1}), ..., (x_{1n}, ..., x_{mn})$.

Obtaining the characteristics of a series contributes to discovering and visualizing patterns in it, identifying similar series or intervals that shows similar features, generating groupings, associating rules as well as other actions in which these characteristics may serve as identification guides. A time series can be considered a data sequence in which to each point is attributed an index value (or length) v and that reducing it to a k, with $k \, \prec\prec \, v$, implies in reducing the computational complexity of time series analysis from $O(v)$ to $O(k)$.

Time series are considered complex data so there are no means to establish an order relation between series and their intervals. Moreover, due to the great variability existing in data, it is almost impossible to find equal series or intervals. In this context, the concept of similarity has greater applicability than the concept of equality. The measure of similarity between intervals in a time series can be understood as a semantic distance between them. Regarding an application domain, an interval can be defined as the set of time and measure values contained between two cut points given by a domain specialist. To run similarity search it is necessary to have a way to measure the amount of similarity and dissimilarity that exists between two objects belonging to the domain, thus the objects are represented in a metric space.

An metric space M can be defined by the $\{\mathcal{S} \, , \, d\}$ pair, in which $\mathcal{S}$ defines the data domain and d is a distance function. The distance function provides the measure that expresses how similar or dissimilar an object is from another [Bozkaya and Ozsoyoglu 1999]. The main distance functions used for similarity search in time series are the Minkowski distance functions (Lp family). The L1 distance is calculated by the sum of the differences between the corresponding elements feature vectors. The L2 distance, also known as Euclidian distance, is the distance between two feature vectors calculated using the quadratic difference between them. It is important to highlight that there are other distance functions in time series analysis. Examples are the Levenshtein function used by Lin et al. [2003] or the function defined by Morinaka et al. [2001] to calculate the distance based on time series height and length.

Nevertheless, in order to apply distance functions in complex data it is necessary to generate a

feature vector. This vector is used by the distance functions to calculate similarity and, consequently, for data search and comparison operations, having as the search result a set of similar objects ranked by similarity in relation to the reference object. This approach is called content retrieval. There are two basic types of similarity search: (i) *range query* that aims at finding objects that are at a maximum r distance from the query object, and (ii) *k-nearest neighbor query* or *k-NN query* that aims to recover the most similar k-objects to a query object.

In the literature, current methods for similarity search in multidimensional series (also known as multiple series) are based on the use of search descriptors for each series individually and, after that, data mining techniques are applied, like association rules [Pradhan and Prabhakaran 2009; Zhuo et al. 2008] to obtain similarity intervals. This type of technique can present satisfactory results for a determined domain, but may not be good enough for others [Zhong and Gang 2011], since it has been adapted for a specific domain.

2.1   Series descriptors

In the literature, there is no consensus about the concept of a descriptor for complex data. Some authors define a descriptor as being formed by a tuple (ϵ_D, δ_D) in which ϵ_D is a component responsible for characterizing the object through a feature vector extraction that serves for analyzing the data and δ_D is the dissimilarity (or similarity) function responsible for comparing the feature vectors [Torres and Falcao 2006]. Nonetheless, it is possible to find in the literature the concept of the descriptor referring only to the function that generates the feature vector. In this article, we consider the tuple (ϵ_D, δ_D) as a descriptor. Most descriptors are efficient in a given data domain but they present loss of representativeness in other domains. The main time series descriptors found in the literature are as follows.

—***Sequential Scan*** (SM) also known as Brute-Force Solution, Sequential Matching or Sequential Scanning, is cited in several work [Faloutsos et al. 1994; Keogh 1997] and is considered a trivial method for similarity search in series. It consists of dislocating a query sequence in all the series, calculating the distance, usually using the $L2$ distance function between each point, and sequentially searching all possible subsequences belonging to the sequence that is possibly the most similar to the inserted query. This method presents good accuracy for similarity search. Yet, one of its main problems lies in the computational complexity of its execution. This method complexity is $O((m - n + 1) * n)$ [Keogh 1997] in which m is the number of points of the queried series and n is the number of points existing in the query. Therefore, its application becomes not viable for a series that presents a large amount of points.

—**Discret *Fourier* Transform** (DFT) is a technique based on signal processing proposed by Agrawal et al. [1993] in which a series can be expressed as a linear combination of harmonic solutions through a small number of coefficients. Being a transformation that expresses a time series in terms of a linear combination of sinusoidal basis, it is very efficient to determine the spectrum of a signal frequency, i.e., for determining inflection points in the series. In these cases, the descriptor has satisfactory results. Nonetheless, to analyze stationary time series, in which value variation is small, the result obtained from the series representation by DFT will also present a small variation and this can make the series analysis difficult.

—**Singular Value Decomposition** (SVD) proposed by Korn et al. [1997] is the representation of the series through a linear combination of formats, i.e., the series is represented by an **A**-matrix of m x n size with $A_{mxn} = U_{mxn}S_{nxn}V_{nxn}^T$, in which **S** represents a vector with **A** auto values. The matrices **U** and **V** are the decompositions of an orthonormal basis for the columns and lines of **A**, respectively. The calculation of eigenvectors and eigenvalues has a large computational cost. For the representation of large intervals, the dimensionality reduction presents losses.

—**Discrete *Wavelet* Transform** (DWT), proposed by Chan and Fu [1999], transforms the series into a linear combination of functions based on the mathematician Alfred Haara's definition of

wavelet. This descriptor has become inefficient for representing data that have large amplitudes or a large variability in the data, because there is a deletion of important characteristics by scaling function.

—**Piecewise aggregate approximation** (PAA), proposed by Keogh et al. [2001], represents the series through an equally sized segment series, employing the series average value in the interval. The distance function usually applied is *L1*. This descriptor is inefficient for data representation with great variability, because there is a deletion of important features.

—*Adaptive Piecewise Constant Approximation* (APCA) [Chakrabarti et al. 2002] is an improvement over the PAA descriptor in which the segments present adaptive sizes and it presents several segments in the series periods that show great variability and few segments in intervals of low variability. The distance function usually applied is *L1*. Characteristics of the behavior of the series are deleted by this descriptor.

—*Symbolic Aggregate Approximation* (SAX), proposed by Lin et al. [2003] and improved by Camerra et al. [2010], converts the series into a sequence of characters according to the variability of the data and it uses a text-based distance function for similarity calculation. This descriptor is inefficient for series that exhibit great variability in the data.

—*Dynamic time warping* (DTW), proposed by Berndt and Clifford [1994], is a descriptor that uses a distance function based on non-linear alignments among series or intervals of the series for the calculation of dissimilarity. The calculation of the distance between the segments has a high computational cost.

3.   TSS PROPOSED METHOD

In this section, we put forward a descriptor to reduce series dimensionality and to perform similarity search in time series. Afterwards, we offer a method for similarity search in multiple time series that uses the proposed descriptor.

3.1   Coulomb Descriptor

Coulomb's Law establishes the mathematical relation between the charges of two or more bodies and the resulting electric force, calculating interaction forces (attraction and repulsion) between these charges. The principles of Coulomb's Law can be expressed by: (i) the electric force intensity is directly proportional to the electric force product; and (ii) the electric force intensity is inversely proportional to the square distance between the bodies. The law's formula is expressed in Equation 1.

$$\vec{F} = K\frac{q_1 q_2}{r^2}\hat{r} \tag{1}$$

In this equation, $\vec{F}$ is the force in Newtons, r is the distance between two of the point charges, q_1 and q_2 are the charges intensity, $\hat{r}$ is the unit vector, and K is Coulomb's constant.

Given the above, the proposal for similarity search in series considers the observations of the time series as point charges with constant q charge values located in the coordinate plane formed by the series index and by the observation value. Since we need to compute the distance between the charges to obtain the interaction between them, we consider a Cartesian plane formed by the time series index (X-axis) and the value of the observations (Y-axis) and, then, it is possible to calculate the distance between the charges for the calculation of forces.

A fictional q point charge is inserted at the centroid composed of the observation sets that form the search intervals. Figure 1 shows the interaction between charges. The q point charge aims to provide a representation of the interval, for it is not only located in the geometric center of the interval, but it is also used to compute the interaction between it and the other charges generating the resulting force that represents the interval. Since the resulting force is a vector measure, the charge direction
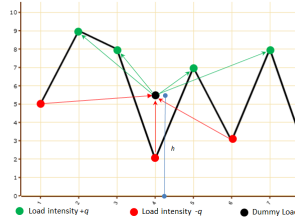
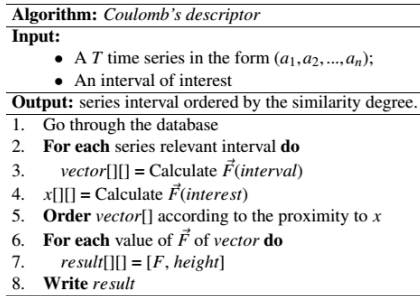Fig. 1: Coulomb's descriptor - Interaction between charges

Algorithm: *Coulomb's descriptor*
Input:
• A T time series in the form $(a_1, a_2, ..., a_n)$;
• An interval of interest
Output: series interval ordered by the similarity degree.
1. Go through the database
2. **For each** series relevant interval **do**
3. $vector[][]$ = Calculate $\vec{F}(interval)$
4. $x[][]$ = Calculate $\vec{F}(interest)$
5. **Order** $vector[]$ according to the proximity to x
6. **For each** value of $\vec{F}$ of $vector$ **do**
7. $result[][] = [F, height]$
8. **Write** $result$

Algorithm: $\vec{F}$ *calculus*
Input:
• T series interval in the form of $(a_1, a_2, ..., a_n)$;
Output: Resultant force $[F, h]$.
1. $c = centroid(F)$
2. **for each** p series interval point **do**
3. $force = Coulomb(c, p)$
4. $F = \sum_{a_1}^{a_n} force$
5. **return** $[F, height(c)]$

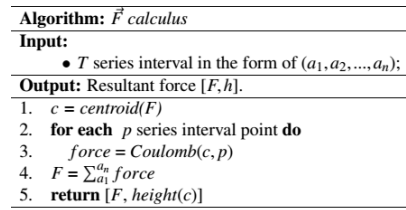Fig. 2: Coulomb's descriptor algorithm

Fig. 3: Auxiliary algorithm for calculating resultant force

and orientation influence the calculation. Because of that, it was established that the charges that are under the charge at the centroid have opposing directions to those that are above it and, consequently, they present negative force intensity. Therefore, it is possible to represent the time series through an electrically-charged particle interaction system and to calculate the resulting force F obtained through the vector sum of all the forces that integrate the system and, then, to achieve the reduction of the series size to assist similarity search without significant loss of information.

In our proposed approach, the feature vector $(V = [\vec{F}, h])$ is formed by the resulting force calculated in the interval of interest and by the centroid height. The need for the centroid height is justified because the resulting force can map the interaction between the interval points that compose it. Nevertheless, no information related to the height existing between the original data is stored and this information is important for similarity calculation.

In order to define the degree of similarity between the instances, using the formerly described feature vectors, the Euclidian distance is applied. The use of this distance is justified because it applies to the data and to Coulomb's Law formula better, maintaining the electric force intensity inversely proportional to the square distance between the bodies.

Figure 2 shows the main algorithm of the Coulomb descriptor, while Figure 3 presents an auxiliary function for force calculation. The Coulomb descriptor algorithm has as input a time series and an interval of interest defined by the user and it returns the resulting intervals ordered by similarity degree making a *knn*-search (without repetition). The algorithm starts by scanning the data, searching for relevant intervals (*line 1*), that is, probable candidates similar to the interval of interest through the analysis of ascending or descending points. In *lines 2, 3 and 4*, the algorithm calculates the resulting force from the interaction between charges through the auxiliary algorithm presented in Figure 3. After that, there is the ordering of the vector that contains values of the resulting force according to the similarity between these intervals and the interval of interest (*lines 5, 6 and 7*). Finally, the result is shown to the user (*line 8*).

The auxiliary algorithm for force calculation receives as input an interval from the series and returns a vector with the resulting force and the centroid height. In *line 1*, the algorithm calculates the interval

centroid, and for each point belonging to the interval, the interaction between this point and the centroid is calculated by Coulomb's Law (*lines 2 and 3*). The resulting force is obtained by summing up the forces calculated at each point and the result is returned to the main algorithm (*lines 4, 5*).

In the TSS method, all the distances used for similarity search are related to a centroid. Because of this, there is no need of a previous data normalization process to employ TSS.

3.2   Similarity Search in Multidimensional Time Series (Multiple Related Time Series)

Similarity search in multiple time series is a considerable challenge given that finding similar subsequences in each of the series from a search interval demands enormous computational processing power in order to produce results with satisfactory accuracy. With a view to minimizing this problem, the Flexible Module (FM) was devised which, allied to the Coulomb descriptor, performs similarity search in multiple series decreasing computational overhead and delivering satisfactory results. The FM is based on the principle of minimum paths where each series similar interval, calculated through a descriptor, is considered the vertex of a graph and the link between vertexes of a series and the vertexes of other series form the edges. The weights of the edges are made up by the degree of dissimilarity (obtained by the descriptor) in the series subsequences until an arbitrarily defined point.

Formally, a graph $G = (V, E)$ is made, in which the weight of the path $p = < v_0, v_1, ..., v_k >$ is the sum of the weights of its constituent corners: $w(p) = \sum_{k}^{i=1} w(v_i - 1, v_i)$. Thus, in order to define the most similar intervals it is necessary to find the lowest weight in an interval of similarity u until a pre-defined point v, given by Equation 2, if there is a path from u until v, or $infinite$ otherwise. Thus, by using Equation 2 we are able to find the intervals that contain the shortest minimum paths and to present the intervals with greater similarity in relation to the given search interval.

$$\delta(u, v) = min\{w(p) : u \rightarrow v\} \tag{2}$$

4.   EXPERIMENTAL RESULTS

With the aim of validating the previously proposed method, our team performed experiments to validate the Coulomb descriptor through the analysis of similarity queries in a one-dimensional series (a single series) and in multidimensional series (multiple related series). In order to compare the proposed method to others, we utilized the most used descriptors found in the literature.

4.1   The Coulomb Descriptor

We used meteorological data, obtained from KMNI Climate Explorer (*http://climexp.knmi.nl*), in which there are measurements of daily temperatures in New York City taken in Central Park between the years of 1835 and 2006. Medical data were also used, obtained from UCI Machine Learning Repository (*http://archive.ics.uci.edu/ml/datasets/Diabetes*), in which there is data about glucose levels in patients during daily activities. Besides, we made use of randomly generated databases.

There is no consolidated validation method for the generation of reliable metrics that might be used for comparing the models and verifying the efficacy of each one. Therefore, we evaluated the methods taking the following aspects into consideration.

—**Computational complexity** that refers to the requirements of indispensable resources for an algorithm to solve a problem, i.e., they refer to the amount of work and/or the time spent to accomplish it [Wilf 2002]. It is a factor of great relevancy for a descriptor's validation, a method of great complexity that demands vast amounts of resources and takes enormous time may not be adequate for many purposes.
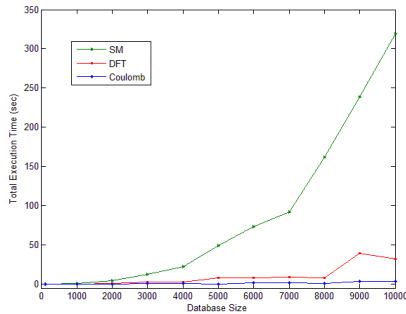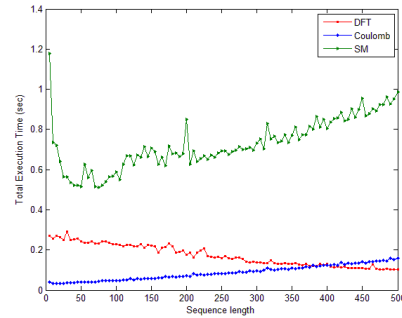
Fig. 4: Time spent by query varying the database size   Fig. 5: Time spent by query varying the query size

—**Accuracy** is a measure used in several areas of science. It is employed to measure the number of instances that were correctly predicted from the input query. In the case of time series, this measure is used by providing a search interval and then verifying the relevance of the output given by the system.

—**Precision vs Recall** (P&R): this technique proposed by Kent et al. [1955] and applied by Meadow et al. [2000] is used for evaluating a method's quality for similarity search. Precision measures the fraction of relevant objects returned in a given query in relation to the total of returned objects. On the other hand, recall measures the fraction of relevant objects returned in a certain search in relation to the total of relevant objects existing in the series. Moreover, the precision vs recall indicates the variation of the precision values along different recall values. When analyzing a P&R curve, the higher the curve the more efficient the descriptor is.

4.1.1 *Computational Complexity.* In order to verify the method complexity, we performed experiments using a randomly generated series. The Coulomb method was compared to the *Sequential Matching* (SM) descriptor, and to *Discrete Fourier Transform* (DFT). SM is known for presenting high accuracy and DFT is known for presenting a good performance in long time series.

The first experiments executed for the verification of the algorithm complexity consisted on executing the same *knn*-search using the three descriptors, varying the size of the basis and taking note of the time spent for the query execution. As the search size increases, the amount of calculations executed also increases. The graph in Figure 4 shows the search execution time for different series size. As evidenced, the Coulomb descriptor presents an execution time lower than the other method. This happens because the Coulomb method is independent of the series size. In addition, the Coulomb descriptor searches only for series intervals that are possible candidates to be similar to the search interval (the intervals that have the same signal as the search positive or negative). On the other hand, the DFT and SM descriptors process the entire series.

Another experiment performed to verify the efficiency of the proposed method was a *knn*-search, varying the search size. By analyzing the graph in Figure 5, we notice that the Coulomb method presents lower execution time than that of the SM descriptor. When compared to the DTF method, it presents good results for search intervals inferior to 400 observations. This happens due to the number of computations executed for each search interval. In fact, when using real time series like the meteorological one, the search windows are smaller than a year, that is, 365 observations. For queries above 400 observations the behavior of Coulomb method is near the DFT method. Thus, the Coulomb method presents satisfactory results regarding complexity.

4.1.2 *Accuracy.* As an initial test for verifying the accuracy of the proposed method, we searched the meteorological time series to locate the intervals of higher similarity to the search interval, according to a certain season of the year. We performed a 100-NN query, given as search window the period concerning the North-American summer (from June 21 to September 23) and winter (from

Table I: Accucacy comparison among descriptors in analysis

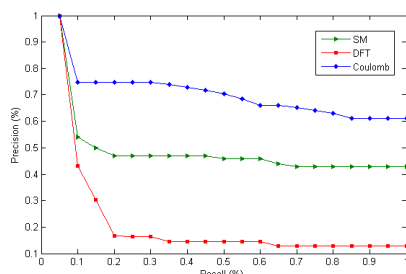|	DFT	SM	Coulomb
Accucacy	20.48%	46.63%	68.95%

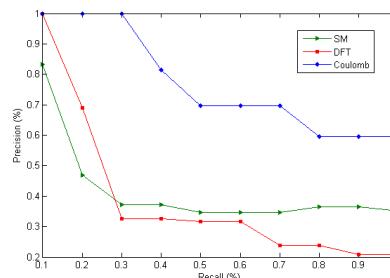Fig. 6: Precision vs recall for the meteorological database

Fig. 7: Precision vs recall for the medical database

December 21 to March 20 of the following year) in 1900. Table I shows the accuracy result. From the experiments results, we argue that the Coulomb method is well-suited to perform similarity search in time series.

4.1.3 *Precision vs Recall.* To perform the precision vs recall experiments, we made use of data from the meteorological and medical time series. To make the precision and recall graphs, we used the recommendations described by Meadow et al. [2000]. Data concerning monthly average temperatures of New York City was used in the meteorological time series. We ran the previously mentioned three descriptors searching for similar seasons. In particular, we were looking for periods in which there are uncommon falls or rises in temperature and periods with some cyclic variability. Figure 6 presents the obtained comparative P&R graph. Analyzing Figure 6, we notice that the Coulomb method presents a satisfactory result in relation to other methods. The precision is high for most recall levels, while other methods have low precision.

Another experiment performed employed a medical time series. Considering that a patient glucose level decreases after insulin application, the experiment was based in searching for periods of high or low glucose level in patients blood before or after insulin administration and, also, in specific periods of the day, such as before or after meals, or in mornings and evenings. By analyzing the graph in Figure 7, we notice that the Coulomb method presents a very satisfactory result related to other methods. Precision is high for a recall lower than 50%, while the other methods presented low precision even for indexes of low recall. We notice, therefore, that the precision of the Coulomb method presents higher values when compared to other methods.

4.2 Flexible Module in Multidimensional (Multiple Related) Time Series

To validate the method of similarity search in multiple time series (FM), we implemented it using the same descriptors of previous experiments. For the three descriptors (SM, FDT and Coulomb), several experiments were conducted and precision and recall graphs were made using the recommendations by Meadow et al. [2000]. For the tests, the following time series were used.

—Time series of monthly average temperatures in seven airports located in different states in the USA during the period from 1939 to 2011, obtained in KMNI Climate Explorer (*http://climexp.knmi.nl*). The search covered seasons of the year and peak temperature periods.

—Time series containing the monthly average temperature in the state of Florida/USA and a time series of the monthly orange production, in tons, from 1983 to 2006, obtained in KMNI Climate Explorer and Climate Prediction Center (*http://climexp.knmi.nl*) and Climate Prediction Center
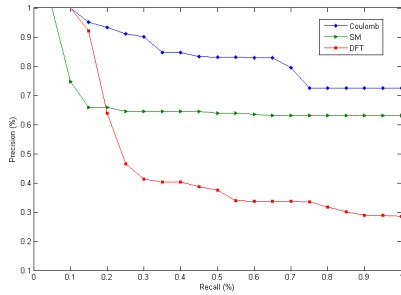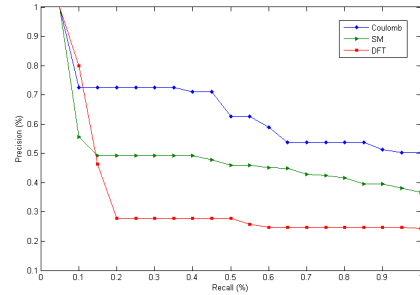
Fig. 8: Precision vs recall for airports
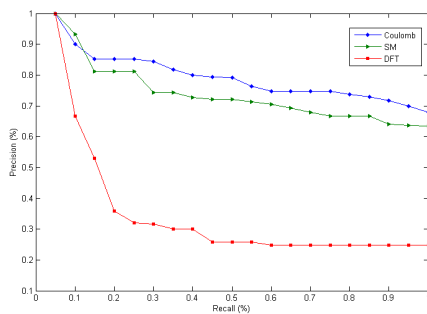

Fig. 9: Precision vs recall for orange production
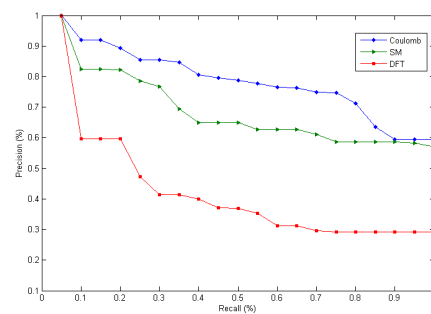

Fig. 10: Precision vs recall for Avaré

Fig. 11: Precision vs recall for Presidente Prudente

(*http://www.cpc.ncep.noaa.gov*). The search covered seasons of the year and periods of high, low and average agricultural production.

—Time series corresponding to minimum and maximum temperatures, and monthly precipitation levels in the cities of Avaré and Presidente Prudente in the State of São Paulo, Brazil, obtained from Agrodatamine Project (*http://www.gbdi.icmc.usp.br/agrodatamine*) from 1961 to 2010. The search covered periods with high and low precipitation levels as well as periods of high and low temperatures not related to precipitation levels.

The P&R graph shown in Figure 8 referring to the average temperature in North-American airports shows the data obtained through the Flexible Module (FM) using the three aforementioned descriptors. Considering that the search was performed in seven quite heterogeneous series, those presented good precision levels for low recall levels and the Coulomb descriptor presented the best results, surpassing even the Sequential Matching descriptor.

Figure 9 shows the precision vs recall graph for the time series of monthly average temperature vs orange production. By analyzing it, it is possible to notice that the FM method presented satisfactory results and that the Coulomb descriptor has surpassed other compared descriptors.

Figures 10 and 11 show the precision vs recall graphs for minimum and maximum monthly average temperatures and precipitation levels of two Brazilian cities. Through its analysis, it is clear that the behavior of the FM of similarity search in multiple series presents satisfactory results. In this case, the search was performed in three series. Besides, the Coulomb descriptor has an advantage over the Sequential Matching descriptor for presenting both lower complexity and a higher accuracy. In fact, the larger the interval size, the higher the possibility of data loss by the TSS method. However, for real applications the queried interval given by user usually tends to be small. This occurs because the user are often searching for an specific aspect of the time series behavior, since searching for general behavior does not have practical utilities.

5. CONCLUSION

In this article, we propose a new descriptor for time series analysis. This descriptor works with one-dimensional and multidimensional time series. By comparing the Coulomb method to traditional ones, it is possible to notice two expressive advantages: smaller execution time and higher precision values. In conclusion, the performed experiments show that the proposed method is well suited to perform similarity search over time series. As future work we intend to include it in a system for visual mining of time series. We intend to improve the developed method adding to it the semantic information given by the user. This will remove the system requirement of defining an initial query interval and also approximate the system answer to the user needs, increasing the applicability of the method.

REFERENCES

AGRAWAL, R., FALOUTSOS, C., AND SWAMI, A. N. Efficient Similarity Search in Sequence Databases. In *Proceedings of the International Conference on Foundations of Data Organization and Algorithms*. pp. 69–84, 1993.

BERNDT, D. J. AND CLIFFORD, J. Using Dynamic Time Warping to Find Patterns in Time Series. In *Workshop on Knowledge Discovery in Databases*. pp. 359–370, 1994.

BOZKAYA, T. AND OZSOYOGLU, M. Indexing Large Metric Spaces for Similarity Search Queries. *ACM Transactions on Database Systems* 24 (3): 361–404, Sept., 1999.

CAMERRA, A., PALPANAS, T., SHIEH, J., AND KEOGH, E. iSAX 2.0: indexing and mining one billion time series. In *Proceedings of the IEEE International Conference on Data Mining*. Washington, DC, USA, pp. 58–67, 2010.

CHAKRABARTI, K., KEOGH, E., MEHROTRA, S., AND PAZZANI, M. Locally Adaptive Dimensionality Reduction for Indexing Large Time Series Databases. *ACM Transactions on Database Systems* 27 (2): 188–228, jun, 2002.

CHAN, K.-P. AND FU, A.-C. Efficient Time Series Matching by Wavelets. In *Proceedings of the International Conference on Data Engineering*. pp. 126–133, 1999.

FALOUTSOS, C., RANGANATHAN, M., AND MANOLOPOULOS, Y. Fast subsequence matching in time-series databases. In *Proceedings of the ACM SIGMOD International Conference on Management of Data*. Minneapolis, Minnesota, USA, pp. 419–429, 1994.

KENT, A., BERRY, M. M., LUEHRS, AND PERRY, J. W. Machine Literature Searching VIII, Operational Criteria for Designing Information Retrieval Systems. *American Documentation* 6 (2): 93–101, 1955.

KEOGH, E. A Fast and Robust Method for Pattern Matching in Time Series Databases. In *Proceedings of Western Users of SAS Software*. Universal City, California, USA, pp. 145–150, 1997.

KEOGH, E., CHAKRABARTI, K., PAZZANI, M., AND MEHROTRA, S. Dimensionality Reduction for Fast Similarity Search in Large Time Series Databases. *Knowledge and Information Systems* 3 (3): 263–286, 2001.

KORN, F., JAGADISH, H. V., AND FALOUTSOS, C. Efficiently Supporting ad hoc Queries in Large Datasets of Time Sequences. *SIGMOD Record* 26 (2): 289–300, June, 1997.

LIN, J., KEOGH, E., LONARDI, S., AND CHIU, B. A Symbolic Representation of Time Series, with Implications for Streaming Algorithms. In *Proceedings of the Workshop on Research Issues in Data Mining and Knowledge Discovery*. San Diego, California, pp. 2–11, 2003.

MEADOW, C., BOYCE, B., AND KRAFT, D. *Text Information Retrieval Systems*. Library and Information Science Series. Academic Press, 2000.

MORINAKA, Y., YOSHIKAWA, M., AMAGASA, T., AND UEMURA, S. The L-index: an Indexing Structure for Efficient Subsequence Matching in Time Sequence Databases. In *International Workshop on Mining Spatial and Temporal Data*. pp. 51–60, 2001.

PRADHAN, G. AND PRABHAKARAN, B. Association Rule Mining in Multiple, Multidimensional Time Series Medical Data. In *IEEE International Conference on Multimedia and Expo*. pp. 1720–1723, 2009.

TANAKA, Y., IWAMOTO, K., AND UEHARA, K. Discovery of Time-Series Motif from Multi-Dimensional Data Based on MDL Principle. *Machine Learning* 58 (2-3): 269–300, Feb., 2005.

TORRES, R. D. S. AND FALCAO, A. X. Content-Based Image Retrieval: theory and applications. *Revista de Informática Teórica e Aplicada* vol. 13, pp. 161–185, 2006.

WEI, W. *Time Series Analysis: univariate and multivariate methods*. Pearson Addison Wesley, 2006.

WILF, H. *Algorithms and Complexity*. A K Peters Limited, 2002.

ZHONG, S. AND GANG, W. Study on Algorithm of Dependent Pattern Discovery of Multiple Time Series Data Stream. In *International Conference on Computer Science and Service System*. pp. 767–769, 2011.

ZHUO, C., BING-RU, Y., FA-GUO, Z., LIN-NA, L., AND YUN-FENG, Z. A New Model for Multiple Time Series Based on Data Mining. In *International Symposium on Knowledge Acquisition and Modeling*. pp. 39–43, 2008.