Chapter XXXII
Similarity Search in Time Series

Maria Kontaki
Aristotle University, Greece

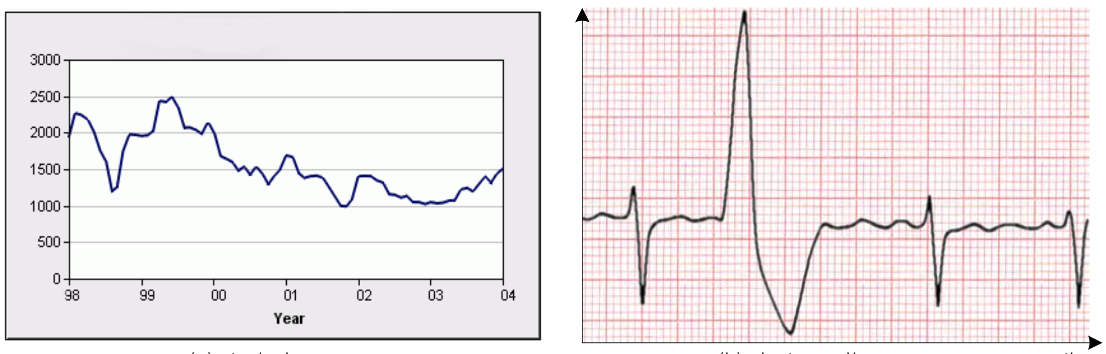
Apostolos N. Papadopoulos
Aristotle University, Greece

Yannis Manolopoulos
Aristotle University, Greece

INTRODUCTION

In many application domains, data are represented as a series of values in different time instances (*time series*). Examples include stocks, seismic signals, audio, and so forth. Similarity search in time series databases is an important research direction. Several methods have been proposed to provide efficient query processing in the case of static time series of fixed length. Research in this field has focused on the development of effective transformation techniques, the application of dimensionality reduction methods, and the design of efficient *indexing schemes*. These tools enable the process of *similarity queries* in time series databases. In the case where time series are continuously updated with new values (*streaming time series*), the similarity problem becomes even more difficult to solve, since we must take into consideration the new values of the series. The dynamic nature of streaming time series precludes the use of methods proposed for the static case. To attack the problem, significant research has been performed towards the development of effective and efficient methods for streaming time series processing. In this article, we introduce the most important issues concerning similarity search in static and streaming time series databases, presenting fundamental concepts and techniques that have been proposed by the research community.

Figure 1. Examples of time series data



(a) stock changes (b) electrocardiogram

BACKGROUND

Time series are used in a broad range of applications, modeling data that change over time. For example, stock changes, audio signals, seismic signals, electrocardiograms, can be represented as time series data. In fact, any measurement that changes over time can be represented as a time series. Two simple time series examples are depicted in Figure 1.

We differentiate between two types of time series, namely: 1) *static time series*, and 2) *streaming time series*. In the first case, we assume that the time series is composed of a finite number of sample values, whereas in the second case the size of the series is increasing since new values are appended. For example, if the data correspond to stock prices for the year 2004, then we can use static time series to capture the stock prices of the time period of interest. On the other hand, if there is a need for continuous stock monitoring as time progresses, then streaming time series are more appropriate.

Streaming time series is a special case of streaming data, which nowadays are considered very important and there is an increasing research interest in the area. Traditional database methods can not be applied directly to data streams. Therefore, new techniques and algorithms are required to guarantee efficient and query processing in terms of the CPU time and the number of disk accesses. The most important difficulty that these techniques must address is the continuous change, which poses serious restrictions.

The purpose of a time series database is to organize the time series in such a way that user queries can be answered efficiently. Although user queries may vary according to the application, there are some fundamental query types that are supported:

- *whole-match queries*, where all time series have the same length, and
- *subsequence-match queries*, where the user's time series is smaller than the time series in the database, and therefore we are interested in time series which contain the user's time series.

In contrast to traditional database systems, time series databases may contain erroneous or noisy data. This means that the probability that two time series have exactly the same values in the same time instances is very small. In such a case, *exact search* is not very useful, and therefore *similarity search* is more appropriate. There are three basic types of similarity queries: