Chapter 13
Querying of Time Series for Big Data Analytics

Vasileios Zois
University of Southern California, USA

Charalampos Chelmis
University of Southern California, USA

Viktor K. Prasanna
University of Southern California, USA

ABSTRACT

Time series data emerge naturally in many fields of applied sciences and engineering including but not limited to statistics, signal processing, mathematical finance, weather and power consumption forecasting. Although time series data have been well studied in the past, they still present a challenge to the scientific community. Advanced operations such as classification, segmentation, prediction, anomaly detection and motif discovery are very useful especially for machine learning as well as other scientific fields. The advent of Big Data in almost every scientific domain motivates us to provide an in-depth study of the state of the art approaches associated with techniques for efficient querying of time series. This chapters aims at providing a comprehensive review of the existing solutions related to time series representation, processing, indexing and querying operations.

INTRODUCTION

Time series data refer to a collection of data points which represent the evolution or behavior of a specific entity in time. Examples include, but are not limited to consumption information from distinct customers on a power grid, stock price closing values (Bao, 2008), patient vital signs as monitored by special equipment and more recently tweets, blog posts. A time series is defined as a sequence of pair $\left[\left(s_1, t_1\right), (s_2, t_2) \ldots (s_k, t_k)\right]$ where s_i is a data point (value) and t_i is the timestamp at which s_i is recorded. Timestamps can be omitted for simplicity, in which case a time series object S is described by the vector $\left[s_1, s_2 \ldots s_k\right]$ where s_j is the observed value of the j-th time interval. Values are assumed to

DOI: 10.4018/978-1-4666-8767-7.ch013

be presented in the same order as they are observed, so for $i < j$, s_i appears before s_j in the corresponding vector. The length of the time interval between consecutive values can be fixed or variable. This definition refers to univariate time series (Chatfield, 2013). Multivariate time series (Box, Jenkins, & Reinsel, 2013), (Wang, Zhu, Li, Wan, & Zhang, 2014) refers to a sequence of observations with multiple value at every given point in time. Time series graphs refer to snapshots of evolving/temporal/dynamic graphs (Park, Priebe, & Youssef, 2013), (Wang, Tang, Park, & Priebe, 2014), (Yan & Eidenbenz, 2014). A collection of these snapshots are what constitute the time series object. This chapter focuses on univariate or multivariate time series and queries related to them. Graph time series are mentioned for completeness and will be mentioned only on a high level as part of the latter developments in the field. Difference in sampling rates can make it difficult for distinct time series objects to be compared. Interpolation is a standard preprocessing operation used to fill gaps between intervals induced either by incompatible sampling rates or missing values. Interpolation techniques are not discussed in this chapter but are mentioned for completeness as part of time series workflows. Other common preprocessing steps include time series normalization. Normalization is performed by eliminating the amplitude value through subtracting the mean and dividing with the standard deviation (Loh, Kim, & Whang, 2000).

Operations on time series data fall into two major categories: analysis and forecasting. Common operations include classification (Kamath, Lin, & De Jong, 2014), clustering (Euán, Ortega, & Alvarez-Esteban, 2014), motif discovery (Mueen, 2014), query by content (Esling & Agon, 2013). Efficient implementation of such operations presents a great challenge to researchers who need to ensure high throughput and consistent availability of data (Loboz, Smyl, & Nath, 2010). As time series data are inherently large to be processed entirely in memory there is a need for solutions that consider and minimize the effect of secondary memory access. Sliding window is commonly used to group fixed-size windows so that it can be processed in memory incrementally. However, this technique can increase the number of random I/Os if data are appropriately (Anderson, Arlitt, Morrey III, & Veitch, 2009), which affects the overall execution time which has been studied extensively over the years (Ding, Trajcevski, Scheuermann, Wang, & Keogh, 2008). Time series indexing can be effectively used for shape based matching operations which are common for many of the complex analytics operations that are mentioned above. Although indexing improves access to the underlying data it does not provide an easy to use management scheme. It is necessary for scientists to have an easy access to services that support real time querying and gathering of information from different sources. This can be achieved by a definition of some kind of a management system along with a query language suitable for the needs of a common user.

TIME SERIES AS BIG DATA

The advent of Big Data and the emergence of the Internet of Things, combined with the emerging applications on cyber-physical systems (Baheti & Gill, 2011) have resulted in an unprecedented volume of time series data, which is updated at staggering speeds. As a consequence, a dramatically increasing amount of interest in querying mining and analysis of time series data has resulted in a rich literature for indexing, search classification, clustering, predictive modeling, visualization, summarization and approximation. Time series data display all the characteristics that describe Big Data and are related to volume, velocity, variety, veracity and volatility of data (Russom, 2011). The large number of individual