Information Sciences 301 (2015) 99–123

Contents lists available at ScienceDirect

Information Sciences

journal homepage: www.elsevier.com/locate/ins

Time-series event-based prediction: An unsupervised learning framework based on genetic programming

Ahmed Kattan [a,*], Shaheen Fatima [b], Muhammad Arif [a]

[a] AI Real-world Application Lab, Um Al Qura University, Saudi Arabia
[b] Department of Computer Science, Loughborough University, Loughborough, UK

ARTICLE INFO

Article history:
Received 24 January 2013
Received in revised form 17 August 2014
Accepted 31 December 2014
Available online 9 January 2015

Keywords:
Unsupervised learning
Genetic programming
Time-series
K-Means
Prediction
Event detection

ABSTRACT

In this paper, we propose an unsupervised learning framework based on Genetic Programming (GP) to predict the position of any particular target event (defined by the user) in a time-series. GP is used to automatically build a library of candidate temporal features. The proposed framework receives a training set $S = \{(\mathbf{V_a})|\mathbf{a} = \mathbf{0}\cdots\mathbf{n}\}$, where each V_a is a time-series vector such that $\forall V_a \in S, V_a = \{(\mathbf{x_t})|\mathbf{t} = \mathbf{0}\cdots\mathbf{t}_{\max}\}$ where t_{max} is the size of the time-series. All $V_a \in S$ are assumed to be generated from the same environment. The proposed framework uses a divide-and-conquer strategy for the training phase. The training process of the proposed framework works as follow. The user specifies the target event that needs to be predicted (e.g., *Highest value, Second Highest value*, ..., etc.). Then, the framework classifies the training samples into different *Bins*, where $Bins = \{(\mathbf{b_i})|\mathbf{i} = \mathbf{0}\cdots\mathbf{t}_{\max}\}$, based on the time-slot t of the target event in each V_a training sample. Each $b_i \in Bins$ will contain a subset of S. For each b_i, the proposed framework further classifies its samples into statistically independent clusters. To achieve this, each b_i is treated as an independent problem where GP is used to evolve programs to extract statistical features from each b_i's members and classify them into different clusters using the *K-Means* algorithm. At the end of the training process, GP is used to build an 'event detector' that receives an unseen time-series and predicts the time-slot where the target event is expected to occur. Empirical evidence on artificially generated data and real-world data shows that the proposed framework significantly outperforms standard Radial Basis Function Networks, standard GP system, Gaussian Process regression, Linear regression, and Polynomial Regression.

1. Introduction

A time-series is a sequence of data points, measured typically at successive time instants spaced at equidistant time intervals. Usually, time-series data have a natural temporal ordering, which makes time-series analysis distinct from other common data analysis problems, in which there is no natural ordering of the observations. In many real-world applications, a vector V of observations $\{x_0, x_1, \ldots, x_{t_{max}}\}$ collected from equidistant time periods maintains some form of salient characteristics that can be exploited to predict the near future. Although time-series analysis algorithms may use solid mathematical formulas or complex statistical models, their predictions are entirely limited to the available historical data. Thus, no matter how accurate these algorithms tend to be on training data, they cannot guarantee a 100% correct prediction of the future. For

* Corresponding author.
E-mail addresses: Ajkattan@uqu.edu.sa (A. Kattan), S.S.Fatima@lboro.ac.uk (S. Fatima), Mahamid@uqu.edu.sa (M. Arif).

http://dx.doi.org/10.1016/j.ins.2014.12.054
0020-0255/

this reason, time-series prediction can be seen as conditional statements of the form that "if such-and-such behaviour continues in the future, then so and so may happen..." [7].

Generally, time-series analysis is divided into two categories; (A) Forecasting algorithms in which the aim is to predict the value x at time $t+1$ given that sufficient historical data points are available, and (B) Discovering events in a time-series. Discovering an event means to detect unusual variations in the time-series pattern and label them as rare events. An event in a time-series is defined as *"the occurrence of a variation in values over a time span that is of particular interest to a user"* [37]. The focus of this paper is on time-series events detection. Generally, work on time-series event-based detection is divided into two main categories. The first category is based on extract rule sets from the time-series and correlate them with particular events using machine learning algorithms (e.g., see [36]). The disadvantage of these techniques is that they are suitable only when rules for determining the occurrence of an event are clear and well understood. The second category is to detect changes in the flow of the time-series values and label these changes as events (e.g., see [37]). The underlying assumption of these models is that it is possible to mathematically model a time-series to detect unusual variations. The advantage of this approach is that it requires no previous knowledge of the problem domain. However, its main disadvantage is that it looks at the time-series from only one dimension, assuming events are correlated by the past behaviour in the time-series itself and ignoring the fact that other variables may cause an event. Another disadvantage is that it defines events based on time-series variations and prevents the user from defining a particular event of interest.

For the purpose of this work, we consider an event to be the occurrence of an occasion defined by the user. For example, given a time-series vector $V = \{(\mathbf{x_t})|\mathbf{t} = \mathbf{0}\cdots\mathbf{t}_{\max}\}$, a user may sometimes be interested in knowing when the highest point (i.e., max x_j for $0 \leqslant j \leqslant t_{max}$) is likely to occur ($t_{max}$ is the size of the time-series). In general, the user may be interested in knowing when the nth point will occur (e.g., highest point, second highest, or the lowest point in an unseen time-series), depending on the problem domain.

The contributions of this paper are twofold:

1. We propose an unsupervised learning framework based on GP to predict the position of any particular target event (defined by the user) in an unseen time-series.
2. Unlike other time-series-event based detectors, the proposed framework learns the behaviour of the environment that generates the time-series itself and uses this knowledge to predict when a target event is likely to occur in an unseen time-series.

The proposed framework receives training examples of historical time-series vectors generated from the same environment and uses GP to automatically build a library of candidate temporal features. In this paper we will use the term "*behaviour*" to refer to statistical features. Thus, for example, as illustrated in Fig. 1, two time-series V_1 and V_2 generated from the same environment may not be identical but have similar behaviour in their trends of going up and down. In real-world applications, the environment can be anything including, but not limited to, stock markets, buyer–seller negotiations, or prices of oil, gas, or electricity in international markets.

The proposed framework works as follows. The examples of the training set are first put in different bins based on the exact time (in $[0, t_{max}]$) at which the event of interest happens. All bins are considered independent learning problems,

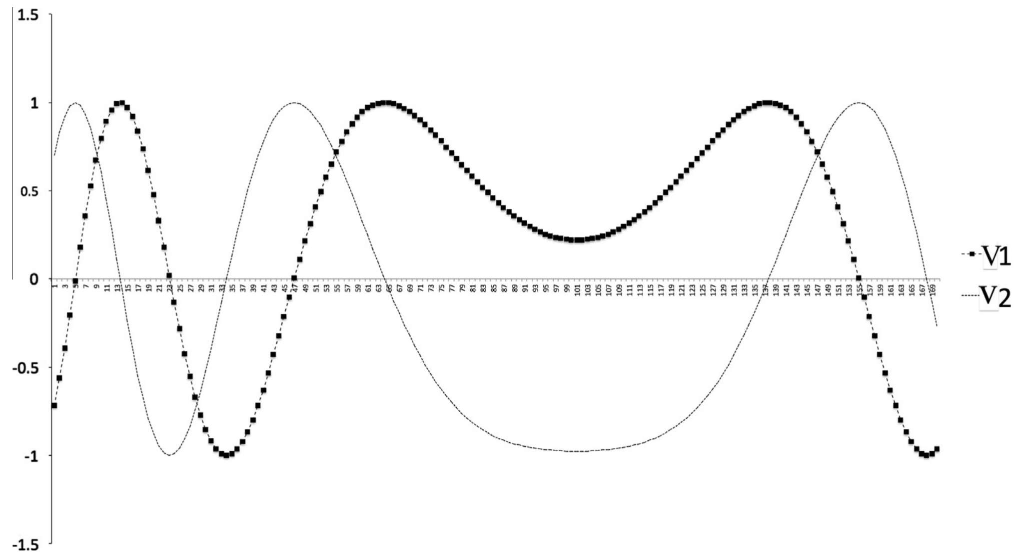

Fig. 1. V_1 and V_2 not identical but have similar behaviour.