Expert Systems With Applications 97 (2018) 1–17

Contents lists available at ScienceDirect

Expert Systems With Applications

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

The incremental Fourier classifier: Leveraging the discrete Fourier transform for classifying high speed data streams

Chamari I. Kithulgoda*, Russel Pears, M. Asif Naeem

School of Engineering, Computer and Mathematical Sciences, Auckland University of Technology, Auckland, New Zealand

ARTICLE INFO

Article history:
Received 12 August 2017
Revised 11 December 2017
Accepted 12 December 2017
Available online 12 December 2017

Keywords:
Data stream
Ensemble classifier
Discrete Fourier Transform
Concept drift
Fourier spectrum
Feature selection

ABSTRACT

Two major performance bottlenecks with decision tree based classifiers in a data stream environment are the depth of the tree and the update overhead of maintaining leaf node statistics on an instance-wise basis to ensure that classification is consistent with the current state of the data stream. Previous research has shown that classifiers based on Fourier spectra derived from decision trees produce compact array structures that can be searched and maintained much more efficiently than deep tree based structures. However, the key issue of incrementally adapting the spectrum to changes has not been addressed. In this research we present a strategy for incremental maintenance of the Fourier spectrum to changes in concept that take place in data stream environments. Along with the incremental approach we also propose schemes for feature selection and synopsis generation that enable the coefficient array to be refreshed efficiently on a periodic basis. Our empirical evaluation on a number of widely used stream classifiers reveals that the Fourier classifier outperforms them, both in terms of classification accuracy as well as speed of classification.

© 2017 Elsevier Ltd. All rights reserved.

1. Introduction

The need for scaling up the process of mining high speed data streams is now paramount than ever before. However, greater throughput should not come at the price of prediction accuracy. Incremental learning techniques have been used extensively to address the data stream classification problem and to maintain a good balance between accuracy and efficiency (Mena-Torres & Aguilar-Ruiz, 2014). In this research we adopt an incremental strategy based on the use of the Discrete Fourier Transform (DFT).

We propose a novel approach for leveraging the DFT to scale up throughput while maintaining or improving classification accuracy over current state-of-the-art data stream classifiers.

The Discrete Fourier Transform has long been a key tool in signal processing and has also been applied to data mining (Kargupta & Park, 2004; Kargupta, Park, & Dutta, 2006; Park, 2001). It has several attractive properties for capturing patterns that sets it apart from conventional mechanisms such as decision trees and other types of classifiers. Firstly, it has been shown rigorously that spectra generated from hierarchical classifiers such as decision trees can be represented in compact form thus speeding up the classification process (Sripirakas & Pears, 2014). Secondly, Fourier spectra have the ability to embed several different patterns (concepts) into one entity unlike conventional ensemble classifier systems which maintain multiple models. This is due to the fact that Spectra can be represented in array form and hence spectra generated at several different points in time can be aggregated into one unifying spectrum that embeds the properties of its constituent spectra. Thirdly, classification can be performed in Fourier space and Fourier spectra, once generated from conventional classifiers, can be used independently of them. Fourthly, the distributive nature of the inverse Fourier transform operation offers the possibility of exploiting parallelism in the classification process.

Such properties have been exploited (Kithulgoda & Pears, 2016; Sakthithasan, Pears, Bifet, & Pfahringer, 2015) in mining data streams but their use comes at a price. The application of the DFT on multivariate data to produce a spectrum is a non-trivial operation and has time complexity $O(|X|^2)$, where $|X|$ is the size of the feature space (Kargupta & Park, 2004). The size of the feature space $|X|$ grows exponentially with the dimensionality of the data. In a highly dynamic data stream environment the time spent on repeated application of the DFT at each concept detection point can quickly become prohibitive as our experimentation in Section 6.9 shows.

A much more effective strategy would be to incrementally maintain a spectrum in a fashion analogous to the incremental maintenance of a conventional classifier such as a decision tree. We draw on the Staged Online Learning (SOL) approach proposed

* Corresponding author.
E-mail addresses: ckithulg@aut.ac.nz (C.I. Kithulgoda), rpears@aut.ac.nz (R. Pears), mnaeem@aut.ac.nz (M.A. Naeem).

https://doi.org/10.1016/j.eswa.2017.12.023
0957-4174/© 2017 Elsevier Ltd. All rights reserved.

Table 1
Mapping of Fourier concepts to their intuitive meanings.

Symbol	Meaning
x	A schema consists of a vector of feature values drawn from features that comprise the dataset. A schema is a compact way of defining a set of data instances, all of which share the same set of feature values.
X	The schema set which contains the set of all possible schema for a given dataset.
j	This is a partition of the feature space. Essentially, it is also a vector of feature values, just as with a schema. The only (conceptual) difference is that a schema refers to the data whereas a partition indexes a Fourier spectrum.
J	The partition set that defines the number of coefficients in the spectrum and its size.
$w_{\vec{j}}$	A coefficient in the Fourier spectrum
$\psi_{\vec{j}}^{\vec{\lambda}}(\vec{x})$	This is the Fourier basis function that takes as input a feature vector and a partition vector and produces an integer for a dataset with binary valued features or a complex number for a dataset with non-binary feature values.

by Kithulgoda and Pears (2016) that uses a two staged approach to data stream classification. The SOL approach divides data streams into two types of segments based on the level of volatility in the stream, which is measured by the rate of appearance of new concepts in the stream. Stage 1 represents a high volatility phase while Stage 2 operates in a low volatility phase. We deploy the IFC in the low volatility phase for two reasons. Firstly, refining an already established spectrum by making small incremental changes to it is far more attractive than having to generate a new spectrum from a decision tree. Our experimental results in Section 6.9 supports this premise very strongly. Secondly, the underlying philosophy of the staged approach is that Stage 1 provides a basis for Stage 2 by capturing new patterns as they arrive in a high volatility stage. These patterns, stored as spectra can then be refined in an incremental manner in the low volatility Stage 2 phase without compromising on accuracy.

This research presents an approach that monitors and performs updates at variable-sized windows determined by the rate of change of concepts in the stream. The contributions we make are:

1. an incremental approach for maintaining a spectrum that eliminates the need for regeneration of spectra at update intervals
2. a self indexing hashing scheme that provides fast access to a reservoir that contains a synopsis of changes that occur in a given time window
3. the development of a novel schema pruning scheme which directly targets noisy features in the raw data

The paper is organized as follows. In Section 2 we present background on the properties of Fourier spectra and their derivation from Decision Trees. In Section 3 we give a brief overview of the application of the DFT to multivariate data and give some important insights into the use of the Fourier spectrum as a classifier. In Section 4 we present an incremental approach to maintaining a Fourier spectrum. Section 5 presents an efficient scheme for capturing a synopsis of the data that is critical to the update of the Fourier coefficient array. Section 6 presents our empirical study and in Section 7 we discuss an application of DFT in data engineering context. We conclude the paper in Section 8 with some directions for future work in mining high speed data streams.

2. Background and related work

The use of the DFT in data mining has been of recent origin and has been focused on deriving a Fourier spectrum from Decision trees. We first present a basic overview of the derivation of the multivariate DFT from a decision tree and then go on to describe the setting in which our incremental scheme is applied. Before we present the mathematical foundations of the DFT we map fundamental Fourier concepts to their meanings in Table 1 in order to communicate their roles in an intuitive fashion.

A Fourier spectrum is derived from a Fourier basis set which consists of a set of orthogonal functions that are used to represent a discrete function. Consider the set of all d-dimensional feature vectors where the lth feature can take λ_l different discrete values, $\{0, 1, \ldots, \lambda_l - 1\}$. The Fourier basis set that spans this space consists of $\prod_{l=1}^{d} \lambda_l$ basis functions. Each Fourier basis function is defined as:

$$\psi_{\vec{j}}^{\vec{\lambda}}(\vec{x}) = \frac{1}{\sqrt[d]{\prod_{l=1}^{d} \lambda_l}} \prod_{l=1}^{d} \exp\left(\frac{2\pi i x_l j_l}{\lambda_l}\right) \quad (1)$$

where $\vec{j}$ and $\vec{x}$ are vectors of length d; $x(m)$, $j(m)$ are the mth attribute values in $\vec{x}$ and $\vec{j}$, respectively. The vector $\vec{j}$ is called a partition and its order is the number of nonzero feature values it contains.

A function f: $X^d \to \mathcal{R}$ that maps a d-dimensional discrete domain to a real-valued range can be represented using the Fourier basis functions:

$$f(\vec{x}) = \sum_{j \in X} \overline{\psi_{\vec{j}}}^{\vec{\lambda}}(\vec{x}) w_{\vec{j}}, \quad (2)$$

where $w_{\vec{j}}$ is the Fourier Coefficient (FC) corresponding to the partition $\vec{j}$ and $\overline{\psi_{\vec{j}}}^{\vec{\lambda}}(\vec{x})$ is the complex conjugate of $\psi_{\vec{j}}^{\vec{\lambda}}(\vec{x})$. Henceforth we shall drop the superscript λ from the ψ_j function formulation to simplify the presentation. The Fourier coefficient $w_{\vec{j}}$ can be viewed as the relative contribution of the partition $\vec{j}$ to the function value of $f(x)$ and is computed from:

$$w_{\vec{j}} = \prod_{i=1}^{l} \frac{1}{\lambda_i} \sum_{x \in X} \psi_{\vec{j}}(\vec{x}) f(\vec{x}) \quad (3)$$

In a data mining context, $f(\vec{x})$ represents the classification outcome of a given data instance $\vec{x} \in X$. Each data $\vec{x}$ must conform to a schema and many data instances in the stream may map to the same schema. For example, in Fig. 1, many data instances for schema (0, 0, 1) may occur at different points in the stream. Henceforth in the paper we shall refer to schema instances rather than data instances as our Fourier classifier operates at the schema, rather than at the data instance level. Thus we shall adopt the notation $\vec{x}$ to denote a schema instance, rather than a data instance. The set X is the set of all possible schema, and for the simple example in Fig. 1 it is of size 8.

The absolute value of $w_{\vec{j}}$ can be used as the significance of the corresponding partition $\vec{j}$. If the magnitude of some $w_{\vec{j}}$ is very small compared to other coefficients, we consider the $\vec{j}$th partition to be insignificant and neglect its contribution. The order of a Fourier coefficient is simply the order of its corresponding partition. We will use terms like high order or low order coefficients to refer to a set of Fourier coefficients whose orders are relatively large or small, respectively.

The Fourier spectrum of a Decision Tree can be computed using the class outcomes predicted by its leaf nodes. As an example consider the decision tree in Fig. 1 defined on a binary valued domain consisting of 3 features. Its truth table derived from the predictions

Download English Version:

https://daneshyari.com/en/article/6855175

Download Persian Version:

https://daneshyari.com/article/6855175

Daneshyari.com