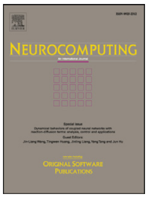
Neurocomputing 239 (2017) 39–57

Contents lists available at ScienceDirect

Neurocomputing

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



A survey on data preprocessing for data stream mining: Current status and future directions

Sergio Ramírez-Gallego[a,*], Bartosz Krawczyk[b], Salvador García[a], Michał Woźniak[c], Francisco Herrera[a,d]

[a] *Department of Computer Science and Artificial Intelligence, CITIC-UGR, University of Granada, Granada 18071, Spain*
[b] *Department of Computer Science, Virginia Commonwealth University, Richmond, VA 23284, USA*
[c] *Department of Systems and Computer Networks, Wrocław University of Science and Technology, Wyb. Wyspiańskiego 27, Wrocław 50-370, Poland*
[d] *Faculty of Computing and Information Technology, King Abdulaziz University, Jeddah, Saudi Arabia*

ARTICLE INFO

Article history:
Received 28 December 2016
Revised 31 January 2017
Accepted 31 January 2017
Available online 14 February 2017

Communicated by Zidong Wang

Keywords:
Data mining
Data stream
Concept drift
Data preprocessing
Data reduction
Feature selection
Instance selection
Data discretization
Online learning

ABSTRACT

Data preprocessing and reduction have become essential techniques in current knowledge discovery scenarios, dominated by increasingly large datasets. These methods aim at reducing the complexity inherent to real-world datasets, so that they can be easily processed by current data mining solutions. Advantages of such approaches include, among others, a faster and more precise learning process, and more understandable structure of raw data. However, in the context of data preprocessing techniques for data streams have a long road ahead of them, despite online learning is growing in importance thanks to the development of Internet and technologies for massive data collection. Throughout this survey, we summarize, categorize and analyze those contributions on data preprocessing that cope with streaming data. This work also takes into account the existing relationships between the different families of methods (feature and instance selection, and discretization). To enrich our study, we conduct thorough experiments using the most relevant contributions and present an analysis of their predictive performance, reduction rates, computational time, and memory usage. Finally, we offer general advices about existing data stream preprocessing algorithms, as well as discuss emerging future challenges to be faced in the domain of data stream preprocessing.

1. Introduction

Data preprocessing [1,2] is one of the major phases within the knowledge discovery process. Despite being less known than other steps like data mining, data preprocessing actually very often involves more effort and time within the entire data analysis process (> 50% of total effort) [3]. Raw data usually comes with many imperfections such as inconsistencies, missing values, noise and/or redundancies. Performance of subsequent learning algorithms will thus be undermined if they are presented with low-quality data. Thus by conducting proper preprocessing steps we are able to significantly influence the quality and reliability of subsequent automatic discoveries and decisions.

Data preparation, as part of preprocessing [1], is aimed at transforming raw input into high-quality one that properly fits the mining process to follow. Preparation is considered as a mandatory step and it includes techniques such as integration, normalization, cleaning and transformation.

Presently, the amount generated data is growing exponentially following the emergence of Big Data phenomenon [4,5]. Contemporary datasets grow in three dimensions –features, examples and cardinality– making complexity reduction a mandatory step if standard algorithms are to be used. Data reduction techniques perform this simplification by selecting and deleting redundant and noisy features and/or instances, or by discretizing complex continuous feature spaces. This allows to maintain the original structure and meaning of the input, but at the same time obtaining a much more manageable size. Faster training and improved generalization capabilities of learning algorithms, as well as better understandability and interpretability of results, are among the many benefits of data reduction.

With the advent of Big Data comes not only an increase in the volume of data, but also the notion of its velocity. In many emerging real-world problems we cannot assume that we will deal with a static set of instances. Instead, they may arrive continuously,

[*] Corresponding author.
E-mail addresses: sramirez@decsai.ugr.es (S. Ramírez-Gallego), bkrawczyk@vcu.edu (B. Krawczyk), salvagl@decsai.ugr.es (S. García), michal.wozniak@pwr.edu.pl (M. Woźniak), herrera@decsai.ugr.es (F. Herrera).

http://dx.doi.org/10.1016/j.neucom.2017.01.078
0925-2312/© 2017 Elsevier B.V. All rights reserved.

leading to a potentially unbounded and ever-growing dataset. It will expand itself over time and new instances will arrive continuously in batches or one by one. Such problems are known as data streams [6] and pose many new challenges to data mining methods. One must be able to constantly update the learning algorithm with new data, to work within time-constraints connected with the speed of arrival of instances, and to deal with memory limitations. Additionally, data streams may be non-stationary, leading to occurrences of the phenomenon called *concept drift*, where the statistical characteristics of the incoming data may change over the time. Thus, learning algorithms should take this into consideration and have adaptation skills that allow for online learning from new instances, but also for quick changes of underlying decision mechanisms [7].

Despite the importance of data reduction, not many proposals in this domain may be found in the literature for online learning from data streams [8]. Most of methods are just incremental algorithms, originally designed to manage finite datasets. Direct adaptation of static reduction techniques is not straightforward since most of techniques assume the whole training set is available from the beginning and properties of data do not change over time:

- Most of static instance selectors require multiple passes over data, at the same time being mainly based on time-consuming neighbor searches that makes them useless for handling high-speed data streams [1].
- On the contrary, feature selection techniques are easily adaptable to online scenarios. Yet, they suffer from other problems such as concept evolution or dynamic [9] and drifting [10] feature space.
- Online supervised discretization methods also remain fairly unexplored. Most of standard solutions require several iterations of sharp adjustments before getting a fully operating solution [11].

Therefore, further development of data pre-processing techniques for data stream environments is thus a major concern for practitioners and scientists in data mining areas.

This survey aims at a thorough enumeration, classification, and analysis of existing contributions for data stream preprocessing. Although there exist previous studies that have performed a coarse-grained analysis on some tasks individually (e.g., feature selection or instance selection) [12,13], this work is a first deep overview of advances in this filed, additionally outlining vital future challenges that need to be addressed to ensure meaningful progress and development of novel methods.

In addition to discussing the literature in preprocessing methods for mining data streams, we propose a thorough experimental study to further enrich this survey. We have analyzed predictive, reduction, time and memory performance of selected most relevant algorithms in this field. Additionally, nonparametric statistical tests are used to give support to the final conclusions. The discussed experimental framework involves a total of 20 datasets and 10 reduction methods: three feature selectors, three discretizers, and four instance selectors.

The structure of this work is as follows. First, we present related concepts such as: data streaming and concept drift (Section 2), and data reduction (Section 3). Then online reduction contributions are grouped by task, and described in Section 4. To assess performance and usefulness of methods, a thorough experimental framework is proposed in Section 5, also grouped by task. Section 6 summarizes the lessons learned from this survey and experimental study, and discusses open challenges in data preprocessing for data stream mining, while Section 7 concludes this work.

2. Data streams and concept drift

Data stream is a potentially unbounded and ordered sequence of instances that arrive over time [14]. Therefore, it imposes specific constraints on the learning system that cannot be fulfilled by canonical algorithms from this domain. Let us list the main differences between static and streaming scenarios:

- instances are not given beforehand, but become available sequentially (one by one) or in the form of data chunks (block by block) as the stream progresses;
- instances may arrive rapidly and with various time intervals between each other;
- streams are of potentially infinite size, thus it is impossible to store all of incoming data in the memory;
- each instance may be only accessed a limited number of times (in specific cases only once) and then discarded to limit the memory and storage space usage;
- instances must be processed within a limited amount of time to offer real-time responsiveness and avoid data queuing;
- access to true class labels is limited due to high cost of label query for each incoming instance;
- access to the true labels may be delayed as well, in many cases they are available after a long period, i.e., for credit approval could be 2–3 years;
- statistical characteristics of instances arriving from the stream may be subject to changes over time.

Let us assume that our stream consists of a set of states $\mathbf{S} = \{S_1, S_2, \ldots, S_n\}$, where S_i is generated by a distribution D_i. By a stationary data stream we will consider a sequence of instances characterized by a transition $S_j \rightarrow S_{j+1}$, where $D_j = D_{j+1}$. However, in most modern real-life problems the nature of data may evolve over time due to various conditions. This phenomenon is known as concept drift [7,15] and may be defined as changes in distributions and definitions of learned concepts over time. Presence of drift can affect the underlying properties of classes that the learning system aims to discover, thus reducing the relevance of used classifier as the change progresses. At some point the deterioration of the quality of used model may be too significant to further consider it as a meaningful component. Therefore, methods for handling drifts in data streams are of crucial importance to this area of research.

Let us now present shortly a taxonomy of concept drift. There are two main aspects that must be taken under consideration when analyzing the nature of changes taking place in the current state of any data stream:

- **Influence on the learned classification boundaries** - here we distinguish two types of concept drift. A **real** concept drift affects the decision boundaries (posterior probabilities) and may impact unconditional probability density function, thus poses a threat to the learning system. A **virtual** concept drift does not impact the decision boundaries (posterior probabilities), but affect the conditional probability density functions, thus not influencing the currently used learning models. However, it should still be detected. Visualization of these drift types is presented in Fig. 1.
- **Types of change** - here we may distinguish three main types of concept drift taking into consideration its rapidness. **Sudden** concept drift is characterized by S_j being rapidly replaced by S_{j+1}, where $D_j \neq D_{j+1}$. **Gradual** concept drift can be considered as a transition phase where examples in S_{j+1} are generated by a mixture of D_j and D_{j+1} with their varying proportions. **Incremental** concept drift has a much slower ratio of changes, where the difference between D_j and D_{j+1} is not so significant, usually not statistically significant.
- We may also face with so-called **Recurring** concept drift, what means that a concept from *k*th previous iteration may