Classification and Recommendation With Data Streams

Bruno Veloso
https://orcid.org/0000-0001-7980-0972
INESC TEC, Portugal & University Portucalense, Portugal

João Gama
https://orcid.org/0000-0003-3357-1195
INESC TEC, Portugal & FEP, University of Porto, Portugal

Benedita Malheiro
https://orcid.org/0000-0001-9083-4292
Polytechnic Institute of Porto, Portugal & INESC TEC, Portugal

INTRODUCTION

Internet of Things (IoT), wireless sensor networks, social networks, crowdsourcing platforms and personal mobile devices are some examples of data stream sources. They continuously generate large volumes of real time data at variable rates, making traditional off-line processing obsolete. Data streams are sequences of timely ordered interactions between members of dynamic collections (Hopfgartner, Kille, Heintz, & Turrin, 2015). While continuous flows of information, they need real time incremental processing techniques to extract meaningful and timely information. With the exponential growth of data stream sources, data stream processing has become indispensable. Data stream algorithms have three main challenges: (*i*) present lower complexity to keep up with the data rate arrival and good scalability; (*ii*) display a manageable memory footprint; and (*iii*) adopt incremental techniques to build and maintain the model(s) updated. Specifically, the model updating process needs to prioritise information, *e.g.*, using novelty and relevance criteria, as well as to identify relevant data changes. Such challenges and problems require the design and development of new techniques to balance the computational complexity and the prediction performance. Data stream processing techniques have been extensively covered in several data mining books (Muthukrishnan, 2005), (Aggarwal, 2007), (Gama & Gaber, 2007), (Bifet, 2010), (Galić, 2016) (Garofalakis, Gehrke, & Rastogi, 2016). This article addresses data stream knowledge processing, ranging from classification to recommendation and evaluation, describing existing techniques, typical fields of application and identifying open challenges.

BACKGROUND

Data stream processing typically includes classification, recommendation and evaluation. Data stream classifiers work incrementally and attempt to classify events as they occur, *i.e.*, in near real time and not *a posteriori*. Micro-clusters (Aggarwal, Han, Wang, & Yu, 2004), decision trees (Rutkowski, Pietruczuk, Duda, & Jaworski, 2013), ensemble classifiers (Osojnik, Panov, & Dzeroski, 2017) and adaptive model

DOI: 10.4018/978-1-7998-3479-3.ch047

rules (Duarte, Gama, & Bifet, 2016) are well-known classification techniques which have been adapted to work with data streams. However, they suffer from concept drift, data outliers and missing data. Concept drifts can occur with time, *e.g.*, when interests change, and may arise from different situations, *e.g.*, changes in the properties of the data or inappropriate hyper-parameter tuning, are identifiable by alterations in the statistical properties of the data. In such cases, past observations become irrelevant to the current state and the algorithm needs to forget to improve its accuracy. Outliers correspond to data spatially distant from the trend of most of the observations. Outlier observations can be considered as hidden variables, inducing large statistical errors, and may indicate experimental error or unexpected variability in the measurement or noise. Missing values result from sensor malfunctioning or communication problems.

The data stream recommendation engines found in the literature rely on content-based or collaborative filters to generate personalised recommendations and popularity, recency, randomness or trending filters to make general recommendations. Popularity filters order items based on the item selection frequency / average rating; recency filters order the items based on the arrival date; the trend-based filters compute the frequency or rating trend of the items within a temporal sliding window; and random filters select randomly items to create a serendipity effect. To make personalised recommendations, content-based filters compute the similarity between the items previously selected and those not yet chosen by the active user, whereas collaborative filters learn user preferences and chose items appreciated by users with interests identical to the active user. With exception of collaborative filtering, the remaining approaches do not incorporate learning, which is essential to be efficient and reactive to the different user behaviours. Consequently, many of the simpler general filters tend to be used in combination with collaborative filtering. For this reason, we will focus our analysis on real time collaborative filters, which are extended versions of standard off-line model-based collaborative filters. They adopt incremental matrix factorisation (Peterek, 2007), (Takács, Pilászy, Németh, & Tikk, 2009), (Vinagre, Jorge, & Gama, 2014) to decompose the large user-item matrix into a product of smaller matrices and, then, apply other mathematical operations such as gradient based optimisation. By default, this technique has multiple challenges such as scalability, cold start, hyper-parameter optimisation and data relevance.

Finally, the evaluation of data stream algorithms, *i.e.*, the assessment of the quality of the predictions generated by a data stream classifier or recommendation engine, requires dedicated techniques such as sliding windows (Li, Maier, Tufte, Papadimos, & Tucker, 2005) (Ghanem, Hammad, Mokbel, Aref, & Elmagarmid, 2007), the prequential approach presented by (Gama, Sebastião, & Rodrigues, 2009) and the prequential Area Under the Curve (AUC) (Brzezinski, & Stefanowski, 2017).

LIMITATIONS OF EXISTING CLASSIFICATION AND RECOMMENDATION ALGORITHMS

The stream-based recommendation and classification algorithms present limitations in terms of data representation, model maintenance, performance and evaluation. Typically, researchers develop and tune their algorithms for off-line processing, using stored data sets and representing all entities. However, this approach is incompatible with on-line real-world data stream processing applications where events may arrive at various rates and tend to accumulate rapidly, generating large volumes of data. Furthermore, data streams, due to their inherent properties, are highly prone to data variability or concept drift, outliers and missing data. To timely process all events received, not only the incremental data processing has to remain simple, *i.e.*, have lower complexity, but parsimonious data representation techniques must be