Mining Streaming and Temporal Data: from Representation to Knowledge

Xiangliang Zhang

King Abdullah University of Science and Technology (KAUST), KSA
Xiangliang.Zhang@kaust.edu.sa

Abstract

In this big-data era, vast amount of continuously arriving data can be found in various fields, such as sensor networks, web and financial applications. To process such data, algorithms are challenged by its complex structure and high volume. Representation learning facilitates the data operation by providing a condensed description of patterns underlying the data. Knowledge discovery based on the new representations will then be computationally effective, and be more effective due to the removal of noise and irrelevant information in the step of representation learning. In this paper, we will briefly review state-of-the-art techniques for extracting representation and discovering knowledge from streaming and temporal data, and demonstrate their performance at addressing several real application problems.

1 Introduction

We are entering an information-dominated age. Mining, and analyzing complex and massive data have become a fundamental challenge because the data sources in nature, in industry, in science, and even in everyday life are becoming increasingly large, diverse, dynamic and geographically distributed. To process continuously arriving data (called data streams), the traditional data mining and machine learning algorithms are usually challenged by the volume of data flushed into memory and disk. Representation of streams facilitates the data operation by providing a condensed description of patterns underlying the streams. Moreover, the new representation reveals intrinsic properties or semantics that are hidden in data, and thus makes the downstream knowledge discovery be effective and computationally efficient.

In this paper, representation learning models will be discussed for characterizing the dynamic density of the data stream by an online density estimator, for profiling users from their movement trajectories and their tweets, and for approximating and compacting data stream by a number of consecutive line segments. Knowledge discovery from new representations will be introduced and applications to different problems will be demonstrated.

2 Data Stream Density Estimator and its Applications

2.1 Online Density Estimator

The unbounded, rapid and continuous arrival of data streams have a unique feature, which is the dynamic underlying distribution. Estimating the Probability Density Function (PDF) for data streams enables the visualization and monitoring of the changing distribution of data streams, and the detection of outliers/anomalies/variations in data streams. Most of the existing approaches are based on the Kernel Density Estimation (KDE) method due to its advantages for estimating the true density [Scott, 1992]. Given a set of samples, \( S = \{x_1, x_2, \ldots, x_n\} \), where \( x_j \in \mathbb{R}^d \). KDE estimates the density at a point \( x \) as:

\[
\hat{f}(x) = \frac{1}{n} \sum_{j=1}^{n} K_h(x - x_j),
\]

where \( K_h(x) \) is a kernel function. Eq. (?1) shows that KDE uses all the data samples to estimate the PDF of any given point. In the problem of online density estimation of data stream, KDE has quadratic time complexity w.r.t. the stream size. Also, the space requirement for KDE significantly increases.

We introduced a method called KDE-Track in [Qahtan et al., 2012] for univariate data streams and extended for multi-variate in [Qahtan et al., 2017]. KDE-Track solves the quadratic complexity problem of KDE by introducing a linear interpolation and adaptive resampling strategy. Taking a sample \( a \) in 2-dim for example, the PDF at \( a \) can be efficiently estimated by bilinear interpolation of the resampling points, as shown in Figure 1 (left),

\[
\hat{f}(a) = \frac{D(a, r_{s1}) \hat{f}(r_{s1}) + D(r_{s1}, a) \hat{f}(r_{s2})}{D(r_{s1}, r_{s2})},
\]

where \( \hat{f}(r_{s1}) = \frac{D(r_{s1}, m_{s1+1}) \hat{f}(m_{s1+1}) + D(m_{s1}, r_{s1}) \hat{f}(m_{s1+1})}{D(m_{s1}, m_{s1+1})} \), and \( \hat{f}(r_{s2}) = \frac{D(r_{s2}, m_{s2+1}) \hat{f}(m_{s2+1}) + D(m_{s2}, r_{s2}) \hat{f}(m_{s2+1})}{D(m_{s2}, m_{s2+1})} \). Here, \( D(b, c) \) is the Euclidean distance between \( b \) and \( c \), \( m_{s1}, m_{s1+1}, m_{s2}, \) and \( m_{s2+1} \) are resampling points surrounding \( a \). The interpolation is efficient as it stores only

---

1 According to “what happens in an internet minute (2018)”, every 60 seconds, there are 4.3M video views at YouTube, 973K Facebook updates, 187M emails sent, 3.7M Google search queries, 481K new Tweets, etc. http://www.visualcapitalist.com/internet-minute-2018/
To guarantee the estimation accuracy and to lighten the load on the model (reduce the number of resampling points), an adaptive resampling strategy is employed, i.e., more points are resampled in the areas where the PDF has a larger curvature, while less number of points are resampled in the areas where the function is approximately linear, as shown in Figure 1 (right). The resampling points and their PDF values are updated after receiving a new data sample, which requires computing time linear to the total number of the resampling points.

The KDE-Track has unique properties as follows: (1) it generates density functions that are available to visualize the dynamic density of data streams at any time; (2) it has linear time and space complexities w.r.t. the model size and 8 – 85 times faster than traditional KDE; (3) the estimation accuracy is achieved by adaptive resampling and optimized bandwidth (h), which also address the spatial non-uniformity issue of data streams.

2.2 Discovery of Anomalies and Variations

In this section, we apply the estimated dynamic density to three different application problems: Taxi traffic real-time visualization, unsupervised online change detection and online outlier detection.

Visualizing the Taxi Traffic Data

One of the main advantages of KDE-Track is the availability of the density function at any time point, which can be used for visualization in real time without any further processing. Figure 2 shows our application on visualizing the dynamic traffic distribution in the New York Taxi trips dataset\(^2\) with window size of 10K. We can find more pickup events during weekends (left) than during regular working days (right) in the Greenwich and the East villages where there are many restaurants and nightclubs. Note that these snapshots are provided as examples only\(^3\). Similar patterns of the density function are repeated over time with minor changes. Such patterns not only are useful in planning better services but also provide critical information to reduce social and environmental costs in the transportation systems.

Online Change Detection

Change detection in data streams refers to the problem of finding time points, where for each point, there exists a significant change in the current data distribution. Accurate detection of changes (concept drifts) is important for data stream mining problems such as online classification, online clustering [Zhang et al., 2014], online optimizing cost of continuous queries [Xie et al., 2016]. A typical window-based solution is to extract a fixed S\(_1\) (reference window) from streaming samples and to update an S\(_2\) (test window) with newly arriving samples [Dasu et al., 2006; Kifer et al., 2004]. Changes are then detected by measuring the difference between the distributions in S\(_1\) and S\(_2\).

Modeling the data distribution and selecting a comparison criterion are essential for change detection in data streams. However, density estimation of multidimensional data is difficult. It becomes less accurate and more computationally expensive with increasing dimensionality. In our previous work [Qahtan et al., 2015], we introduce a framework which applies Principal Component Analysis (PCA) to project the multidimensional data from the stream on the principal components to obtain multiple 1D data streams. Density estimation by KDE-Track, distribution comparison, and change-score calculations can then be conducted in parallel on those 1D data streams. Compared with projecting the data on the original coordinates (i.e., using the original variables), projecting on PCs has the following advantages: 1) it allows the detection of changes in data correlations, which cannot be detected in the original individual variables; 2) it guarantees that any changes in the original variables are reflected in PC projections; and 3) it reduces the computation cost by discarding trivial PCs. Theoretical proofs and evaluation results can be found in [Qahtan et al., 2015].

Online Outlier Detection

Given the density estimated by KDE-Track, outliers can be directly detected based on the intuition that data samples having small PDF values are more likely to be outliers, e.g., if \(f(x)\) is smaller than 5% of \(f\), \(x\) is reported as a suspicious outlier.

\(^2\)Available at: http://www.andresmh.com/nyctaxitrips/

\(^3\)Sample videos of visualization are available at https://youtu.be/YvY22aqyLq4 (Global view for 2-D density in the Manhattan Island); https://youtu.be/g37IrBUU0 (Detailed view of 2-D density around the Central Park); https://youtu.be/d4m09DYz-o8 (Estimated density using 3-D NY Taxi data)
outlier, where \( \hat{f} \) is the average density value of the resampling points. A suspicious outlier is released if its density value increases up to be larger than the threshold. Otherwise, it is confirmed as an outlier.

Figure 3 shows the outliers detected by three methods when applied on the air temperature dataset from [CIMIS, ] from Jan 2000 to Apr 2012, which contains more than 100K data points. Besides KDE-Track, one baseline AROD is an Auto-regression based outlier detection method [Curiac et al., 2007]. The other baseline is MBOD (Median-Based Outlier Detection) [Basu and Meckesheimer, 2007]. Figure 3 shows that MBOD and AROD failed to detect outliers correctly, while KDE-Track reports only those points that have either very small or very large values.

3 Representations of Temporal Traces

Information on our location is now recorded almost everywhere we go, either intentionally over social media like Facebook, or unintentionally via our mobile phones and their associated cellular and WiFi networks. The location trace of one person can be represented as a temporal sequence of places he/she visited, or called trajectories. Mining trajectories allow for social analysis, marketing and urban analysis. However, trajectories are not directly usable as people go to different places at different times.

3.1 Representation of Users from Where they Went and When

In [Alharbi et al., 2016], we proposed a novel representation learning model that infers latent patterns from user trajectories with minimum human intervention. The learned latent patterns characterize the habitual movement patterns of individuals. The graphical representation of the proposed Human Mobility Representation (HuMoR) model is shown in Figure 4. The key advantage of HuMoR is the utilization of timestamp feature of trajectory sequences, which can add important contexts to the raw anonymized user location data, where no semantic categories or geographical location is available, to ensure the privacy of users.

HuMoR is a mixed-membership model built on the basis that there exists a set of latent patterns, i.e., global mixture components, underlying the data. Those global mixture components, i.e., patterns, uncover shared recurring patterns from sequences of locations co-visited by users with similar side features. Additionally, the model infers mixture proportions local to each sequence, at which global components occur. In HuMoR, global mixture components are distributions of patterns over locations, and thus provide means for computing the new location representations. The local mixture proportions, on the other hand, are distributions of sequences over patterns, which provide means for computing the new user representations. The details of model inference can be found in [Alharbi et al., 2016].

3.2 Discovery of Future Social Links

The new user representation can be used for link prediction, which is formulated as a feature-based classification and su-
Figure 5: Graphical model of DUWE. \( u_t \) and \( v_t \) are the dynamic representation of user \( u \) and Twitter word \( v \) at time \( t \), respectively. \( z_t \) is the observed co-occurrence of words, and \( y_t \) is the observed user-word pairs from Twitter streams.

The problem of Piecewise Linear Representation (PLR) is illustrated in Figure 6 (a) (the left is a stream and the right is a new representation of it). Formally, the optimal error-bounded PLR is to construct a minimal number of line segments to represent the stream, and the approximation error does not exceed the predefined error bound. What are the benefits of using the new representation? First, it takes less time and space for processing (a set of stream points can be represented by the slope and offset of a line segment). Second, it helps in excluding outliers that may have negative impact on processing. Given a short and fixed stream, existing solutions can find an optimal set of line segments for PLR. However, it is unrealistic to run PLR every time when there is a new point in streaming data. Our work in [Xie et al., 2018] provides a solution to automatically log the new point.

Figure 6: (a) Piecewise linear representation. (b) Asynchronized local correlation detection between S1 and S2.
6 Conclusion and Future Work

We create new representation of data to facilitate the latent pattern discovery and decision-making in efficient and automated ways. Besides the above-mentioned work, we also proposed an efficient model that can online extract the best representative instances from streams, which can be used for clustering [Zhang et al., 2014] and anomalies detection [Wang et al., 2014]. In graph streams, we develop efficient methods that can approximately count triangles and graphlets in large graph streams with a fixed memory usage, which can be used for network anomaly analysis [Wang et al., 2017]. We also studied time series in cloud systems, such as CPU/memory/disk utilization rate, and network traffic, with a purpose to automate the management of computing resources in Cloud system with more elasticity and better utilization rate [Williams et al., 2014; Zhang et al., 2012; Zheng et al., 2011].

Our current research focuses on several streaming data problems. First, learning user representation comprehensively from their social activities by integrating multi-source information like user trajectories, tweets and social links. Second, we develop semi-supervised representation learning methods, as introducing labels in representation learning process greatly helps the downstream predictive applications. Our ongoing study helps on a system called Delve (https://delve.kaust.edu.sa), which we develop for dataset retrieval and document analysis [Akjuobi and Zhang, 2017].

References


