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 efficient, 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[1]. 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.

[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/

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, $S = \{\boldsymbol{x}_1, \boldsymbol{x}_2, \ldots, \boldsymbol{x}_n\}$, where $\boldsymbol{x}_j \in R^d$. KDE estimates the density at a point $\boldsymbol{x}$ as: $\hat{f}(\boldsymbol{x}) = \frac{1}{n}\sum_{j=1}^{n} K_h(\boldsymbol{x}, \boldsymbol{x}_j)$, where $K_h(\boldsymbol{x}, \boldsymbol{x}_j)$ is a kernel function. Eq. (**??**) 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 $\boldsymbol{a}$ in 2-dim for example, the PDF at $\boldsymbol{a}$ can be efficiently estimated by bilinear interpolation of the resampling points, as shown in Figure 1 (left),

$$\tilde{f}(\boldsymbol{a}) = \frac{D(\boldsymbol{a}, \boldsymbol{r}_{s2})\, \tilde{f}(\boldsymbol{r}_{s1}) + D(\boldsymbol{r}_{s1}, \boldsymbol{a})\, \tilde{f}(\boldsymbol{r}_{s2})}{D(\boldsymbol{r}_{s1}, \boldsymbol{r}_{s2})}, \qquad (1)$$

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

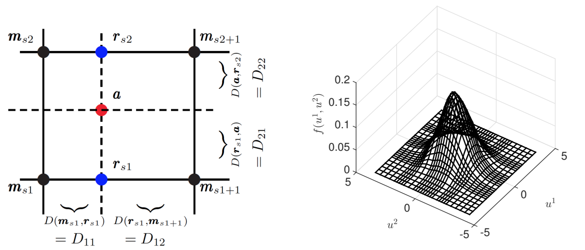

Figure 1: (left) Computing the density at $\boldsymbol{a}$ by interpolation given $\hat{f}$ at $\boldsymbol{m}_{s1}$, $\boldsymbol{m}_{s1+1}$, $\boldsymbol{m}_{s2}$ and $\boldsymbol{m}_{s2+1}$. (right) Example of adaptive resampling: more resampling points are used in regions with high curvature of the function.

$\hat{f}(\boldsymbol{m})$ at resampling points which are much less than the streaming points.

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.

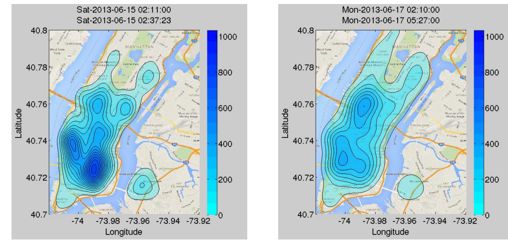

Figure 2: The density estimated from the New York Taxi trips data at weekend (left) and working day (right).

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 $\tilde{f}(x)$ is smaller than 5% of $\bar{f}$, x is reported as a suspicious

[2] Available at: http://www.andresmh.com/nyctaxitrips/

[3] Sample videos of visualization are available at https://youtu.be/YvJZ2aeyLq4 (Global view for 2-D density in the Manhattan Island); https://youtu.be/jq37IRdBUI0 (Detailed view of 2-D density around the Central Park); https://youtu.be/d4n09DYz-o8 (Estimated density using 3-D NY Taxi data)

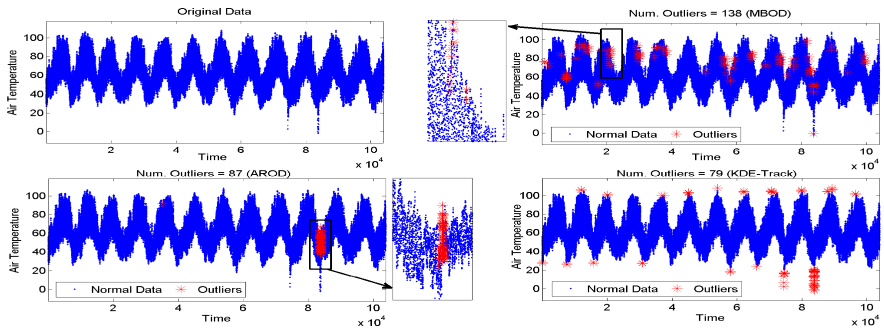

Figure 3: Outliers detected by MBOD, AROD and KDE-Track in the *Air Temperature* from CIMIS dataset

outlier, where $\bar{f}$ 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 useable 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

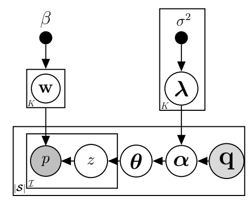

Figure 4: Graphical representation of HuMoR. Nodes represent random variables. Shaded nodes are observed ones (location p and common side feature $\mathbf{q}$, e.g., timestamp). Plates indicate repetition. $\mathcal{I}$ is the length of a sequence, $|\boldsymbol{\mathcal{S}}|$ is the number of sequences. The generative process is: for each sequence, a probability distribution over patterns is drawn, $\boldsymbol{\theta_s}$, from Dirichlet($\boldsymbol{\alpha_s}$), where $\boldsymbol{\alpha_s}$ is a function of the sequence feature $\mathbf{q_s}$, and $\boldsymbol{\Lambda}$ (packing $\boldsymbol{\lambda}_k$ column by column). Then, for every element in the sequence, a mixture assignment is drawn, i.e., a pattern z, from the multinomial pattern distribution given the probabilities in $\boldsymbol{\theta_s}$. Finally, a location is drawn from the multinomial location, parameterized by $\mathbf{w}_{k=z}$ which is randomly drawn from Dirichlet(β).

Datasets	Baseline Measures		Representations			
			FE	PCA	LDA	HuMoR
MDC	CN	0.645	0.859	0.831	0.858	**0.891**
	AA	0.645	0.861	0.831	0.864	**0.888**
	Jacc	0.645	0.845	0.831	0.866	**0.893**
	MF	0.820	0.873	0.893	0.881	**0.943**
GW	CN	0.892	0.727	0.689	0.881	**0.969**
	AA	0.901	0.934	0.938	0.936	**0.972**
	Jacc	0.832	0.721	0.774	0.914	**0.954**
	MF	0.906	0.922	0.942	0.932	**0.950**

Table 1: Mean AUC for link prediction for MDC and GW.

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-

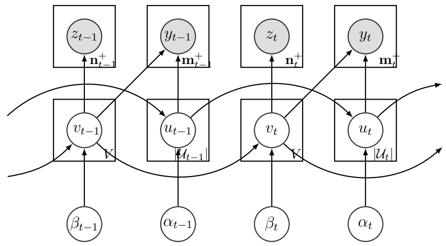

Figure 5: Graphical model of DUWE. u_t and v_t are the dynamic representation of user u and Twitter word v at time t, respectively. z_t is the observed co-occurrence of words, and y_t is the observed user-word pairs from Twitter streams.

pervised matrix factorization (MF) problem. Table 1 shows the AUC when using four different representation produced by Feature Engineering (FE), PCA, Latent Dirichlet Allocation (LDA), and HuMoR to three different topology measures (Common Neighbors (CN), Adamic Adar (AA), and Jaccard Coefficient (Jacc)), on two data sets: Mobile Data Challenge (MDC) [Laurila *et al.*, 2012] and *GoWalla (GW)* [Leskovec and Krevl, 2014]. Values under *Representations* indicate the mean AUC obtained when the new representations used in conjunction with pairs of baseline measures.

Table 1 shows that the overall best result in each dataset is achieved by HuMoR. In both datasets when comparing to the column of *Baseline Measures*, the new representation learned by HuMoR always improves the results significantly and outperforms all other representation learning methods. This is not the case for other baseline methods, especially in the GW dataset, as they even worsen the results because of the inappropriately added representations.

3.3 Representation of Users from Where they and their Friends Went

User location data collected from LBSNs have **incomplete** user movements, as locations are manually entered and not automatically logged. Given that 30% of users have check-ins ≤ 5 in GW, learning of their representation is a challenging task. Existing work completed trajectories with estimates [Farrahi and Gatica-Perez, 2011], or simply eliminated short trajectories like in HuMoR. In [Alharbi and Zhang, 2016], we proposed a model that learns from *incomplete trajectories* with *anonymized locations* by leveraging online **social links**. Comparing to existing work, the proposed model considers extrinsic factors (social ties) while learning mobility patterns, as opposed to existing work which only focuses on individuals' intrinsically motivated mobility patterns (i.e., utilizing metadata directly associated with check-ins, rather than with users). In this regard, this work introduces an important dimension to the human mobility framework.

4 Dynamic User Representation from their Twitter Streams

Twitter users post their current status, recent activities and opinions in short pieces of texts. Learning user representation (a.k.a. user profiling) from their Twitter streams is important to a variety of downstream applications, such as user clustering, and news recommendations. Our recent work proposed a model, called DUWE (dynamic user and word embedding), which builds dynamic user and their Twitter word representation in a common semantic space [Liang *et al.*, 2018]. The graphical model is shown in Figure 5. The core idea is the utilization of Kalman filter for modeling the diffusion process of the vector representations of both users and words over time.

Given user representation at time $t-1$, the evolving user representation at t has a probability

$$p(\mathbf{U}_t \mid \mathbf{U}_{t-1}) \propto \mathcal{N}(\mathbf{U}_{t-1}, \boldsymbol{\alpha}^2_{t-1}\mathbf{I}) \cdot \mathcal{N}(\mathbf{0}, \overline{\boldsymbol{\alpha}}^2_0\,\mathbf{I}), \tag{2}$$

where $\boldsymbol{\alpha}^2_{t-1}$ is the variances of the transition kernels of user representations modeled by Kalman filter. Each element $\alpha^2_{u,t-1} = \varepsilon \cdot g(\mathcal{D}_{u,t}, \mathcal{D}_{u,t-1})(\tau_t - \tau_{t-1})$, where ε is a local diffusion constant $\mathcal{D}_{u,t}$ is a set of documents generated by user u at time t, $g(\mathcal{D}_{u,t}, \mathcal{D}_{u,t-1})$ is a local diffusion value measuring the word distribution changes from $t-1$ to t for user u, which can be modeled by Kullback-Leibler (KL) divergence, and $(\tau_t - \tau_{t-1})$ is the time interval between subsequent observations in the stream. Gaussian prior with mean $\mathbf{0}$ and variance $\overline{\boldsymbol{\alpha}}^2_0$ is added for regularization purpose. Similarly, the probability of word embedding at t, $\mathbf{V}_t$, given the word embedding results at $t-1$, $\mathbf{V}_{t-1}$, is

$$p(\mathbf{V}_t \mid \mathbf{V}_{t-1}) \propto \mathcal{N}(\mathbf{V}_{t-1}, \boldsymbol{\beta}^2_{t-1}\mathbf{I}) \cdot \mathcal{N}(\mathbf{0}, \overline{\boldsymbol{\beta}}^2_0\,\mathbf{I}). \tag{3}$$

Having inferred the new representaitons of words and users, we can generate top-K relevant and diversified keywords to profile users interests over time in streams of tweets. More details can be found in [Liang *et al.*, 2018].

5 Online PLR of Time Series

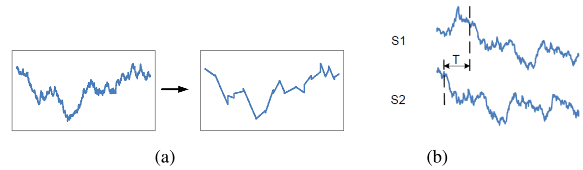

Figure 6: (a) Piecewise linear representation. (b) Asynchronized local correlation detection between S1 and S2.

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.*, 2014] proposed novel linear-time algorithms for performing online PLR, which guarantees optimal representation and lower costs in time and disk space than baseline methods. A direct usage of the new representation is to facilitate the correlation detection between streams (illustrated in Figure 6 (b)). PLR can reduce the cost of correlation detection between two streams, especially when the correlation occurs in burst, lasts for certain duration, and then disappears. Our work in [Xie *et al.*, 2013] solves the asynchronized local correlation detection problem in an online fashion with high efficiency.

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 [Akujuobi and Zhang, 2017].

References

[Akujuobi and Zhang, 2017] U. Akujuobi and X. Zhang. Delve: A dataset-driven scholarly search and analysis system. *SIGKDD Explor. Newsl.*, 19(2):36–46, 2017.

[Alharbi and Zhang, 2016] B. Alharbi and X. Zhang. Learning from your network of friends: a trajectory representation learning model based on online social ties. In *ICDM*, pages 781–786, 2016.

[Alharbi *et al.*, 2016] B. Alharbi, A. Qahtan, and X. Zhang. Minimizing user involvement for learning human mobility patterns from location traces. In *AAAI*, pages 865–871, 2016.

[Basu and Meckesheimer, 2007] S. Basu and M. Meckesheimer. Automatic outlier detection for time series: an application to sensor data. *KAIS*, 11:137–154, 2007.

[CIMIS,] CIMIS. California irrigation management information system, http://www.cimis.water.ca.gov/cimis.

[Curiac *et al.*, 2007] DI. Curiac, O. Banias, and O. Dranga. Malicious node detection in wireless sensor networks using an autoregression technique. In *ICNS*, pages 83–88, 2007.

[Dasu *et al.*, 2006] T. Dasu, S. Krishnan, S. Venkatasubramanian, and K. Yi. An information-theoretic approach to detecting changes in multi-dimensional data streams. In *Interface*, pages 1–24, 2006.

[Farrahi and Gatica-Perez, 2011] K. Farrahi and D. Gatica-Perez. Discovering routines from large-scale human locations using probabilistic topic models. *TIST*, pages 1–27, 2011.

[Kifer *et al.*, 2004] D. Kifer, S. Ben-David, and J. Gehrke. Detecting change in data streams. In *VLDB*, pages 180–191, 2004.

[Laurila *et al.*, 2012] J. Laurila, D. Gatica-Perez, et al. The mobile data challenge: Big data for mobile computing research. In *PerCom*, pages 432–437, 2012.

[Leskovec and Krevl, 2014] J. Leskovec and A. Krevl. SNAP Datasets: Stanford large network dataset collection, 2014.

[Liang *et al.*, 2018] S. Liang, X. Zhang, Z. Ren, and E. Kanoulas. Dynamic embeddings for user profiling in twitter. In *To appear in KDD*, 2018.

[Qahtan *et al.*, 2012] A. Qahtan, X. Zhang, and S. Wang. Efficient estimation of dynamic density functions with an application to outlier detection. In *CIKM*, pages 2159–2163, 2012.

[Qahtan *et al.*, 2015] A. Qahtan, B. Alharbi, S. Wang, and X. Zhang. A PCA-based change detection framework for multidimensional data streams. In *KDD*, pages 935–944, 2015.

[Qahtan *et al.*, 2017] A. Qahtan, S. Wang, and X. Zhang. KDE-Track: An efficient dynamic density estimator for data streams. *TKDE*, 29(3):642–655, 2017.

[Scott, 1992] D. Scott. *Multivariate Density Estimation: Theory, Practice, and Visualization*. 1992.

[Wang *et al.*, 2014] W. Wang, T. Guyet, R. Quiniou, M. Cordier, F. Masseglia, and X. Zhang. Autonomic intrusion detection: Adaptively detecting anomalies over unlabeled audit data streams in computer networks. *Knowl.-Based Syst.*, 70:103–117, 2014.

[Wang *et al.*, 2017] P. Wang, Y. Qi, Y. Sun, X. Zhang, J. Tao, and X. Guan. Approximately counting triangles in large graph streams including edge duplicates with a fixed memory usage. *PVLDB*, pages 162–175, 2017.

[Williams *et al.*, 2014] D. Williams, S. Zheng, X. Zhang, and H. Jamjoom. Tidewatch: Fingerprinting the cyclicality of big data workloads. In *INFOCOM*, pages 2031–2039, 2014.

[Xie *et al.*, 2013] Q. Xie, S. Shang, B. Yuan, X. Pang, and X. Zhang. Local correlation detection with linearity enhancement in streaming data. In *CIKM*, pages 309–318, 2013.

[Xie *et al.*, 2014] Q. Xie, C. Pang, X. Zhou, X. Zhang, and K. Deng. Maximum error-bounded piecewise linear representation for online stream approximation. *VLDB J.*, pages 915–937, 2014.

[Xie *et al.*, 2016] Q. Xie, X. Zhang, Z. Li, and X. Zhou. Optimizing cost of continuous overlapping queries over data streams by filter adaption. *TKDE*, 28(5):1258–1271, 2016.

[Zhang *et al.*, 2012] X. Zhang, Z. Shae, S. Zheng, and H. Jamjoom. Virtual machine migration in an over-committed cloud. In *NOMS*, pages 196–203, 2012.

[Zhang *et al.*, 2014] X. Zhang, C. Furtlehner, C. Germain, and M. Sebag. Data stream clustering with affinity propagation. *TKDE*, 26(7):1644–1656, 2014.

[Zheng *et al.*, 2011] S. Zheng, Z. Shae, X. Zhang, H. Jamjoom, and L. Fong. Analysis and modeling of social influence in high performance computing workloads. In *Euro-Par*, pages 193–204, 2011.