Dynamic Topological Data Analyzes

Arash Vaezi
Sharif University of Technology
avaezi@sharif.edu

S.M. Hussien Kazemi
Sharif University of Technology
hussein.kazemi75@sharif.edu

ABSTRACT

This extended abstract deals with bringing a new way of thinking to improve topological data analysis. We intend to create comparing measurements between two topological data analyses. The measurements help us to improve an analysis gradually. Regarding streaming data items, if we have an initial data analysis, we can improve the result of the analysis gradually. Namely, frequency moments bring nice approximations on several aspects of the given data, and we can use them to compare a previous result to a new one and check the improvement in the expectations. That is, suppose we have already predicted a topological feature for the data, and the data still comes in a streaming manner; we can see if the expected feature comes into reality with time.

KEYWORDS

Topological Data Analysis, Streaming, Compare

1 INTRODUCTION

Topological Data Analysis (*TDA*) [6, 7, 14] helps analyze high-dimensional and noisy data using topology. Persistent homology, a branch of the *TDA*, measures topological features inferred from a simplicial complex at different spatial resolutions. Persistent homology is generally based on the persistence of k-dimensional holes. The persistence of such topological features is quantified using a barcode [14]. Some studies apply *TDA* to capture the dynamic patterns of time series, including the application to financial data [8] and gene expression data [10]. Songdechakraiwut and Chung [12] leverage the application of the *TDA* to the dynamically changing resting-state fMRI data and investigate whether this could lead to a new topological characterization of the brain. In this work, we attempt to provide ideas using streaming techniques to compute the topological features of a dynamically changing dataset.

2 THE IDEA

Consider a given dataset D. The dataset could be a set of samples taken from a specific data space. Suppose that D receives updates periodically. The aim is to measure the changes that may occur when the topological data analysis (*TDA*) of D differs from its original one. To define such measurements, we supposed the data space is a streaming source, and we have access to the initial D with its corresponding TDA. The updates to the dataset shall be received in a streaming manner; hence we can compute a *new* dataset $D^{'}$ periodically. Note that $D^{'}$ is larger than the previous one. That is, D gradually updates, and $D^{'}$ indicates the new dataset. So, we can compute if some features taken from the barcodes or persistence homologies remain the same. Figure 1 illustrate an example. One can define and/or use sufficient measurements compatible with the streaming setting.

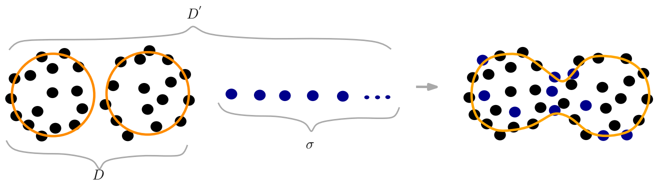

Figure 1: The result obtained from previous data set, D, could be different after recieving a stream of data, σ. $D^{'}$ illustrates the new data set after the current period.

In streaming models, normally, we do not have access to plenty of memory; also, the number of passes on the data could be bounded. So, it is proper to consider analyzing the topology of the data based on some other measurements, such as the frequency moments of a subset of particular persistence homologies. In the following subsections, we suggest a few measurement approaches to obtain a comparison factor between two analyzes of two datasets. Clearly, the factor could be modified and tuned based on the application that needs the analyses.

2.1 Frequency Moments

Consider an initial set of data obtained from a streaming data source at time t_0. Denote this set by $D(t_0)$. Let σ be the data elements coming in a streaming mode. We define a period denoted by p that could be set to be large enough based on the application we deal with. This period indicates the time that enough data items are received to run another iteration of topological data analysis. Denote the subset of data items that are received at time t_i by $\sigma(t_i - t_{i-1})$. Define $D(t_i) = D(t_{i-1}) \cup \sigma(t_i - t_{i-1})$.

Algorithm 1 Compare TDA Dynamically

```
1: procedure FREQUENCY MOMENTS COMPARISON
2:     Initialize : $D(t_0)$ ;
3:     Compute a $TDA$ on $D(t_0)$ ;
4:     Pick $I_0$ obtained from $TDA$ ;
5:     Set $p$ to be a given value ;
6:     IF $p$ new items are recieved
7:         $i++$ ;
8:         $D(t_i) = D(t_{i-1}) \cup \sigma(t_i - t_{i-1})$;
9:     Compute $TDA$ on $D_i$
10:    Output : $I[i]$ ; // or the result of a comparison method.
```

Let I_i be the set of features (persistence homologies or barcodes, etc.) obtained from the topological data analysis run on D_i. A subset of I_i may be necessary for a specific purpose in an application. This subset could be some expected or predicted topological features in the dataset. Denote the frequency of an element of I_i by f_i. The importance of f_i could be revealed in a matching learning application or any other application. However, computing these frequencies or their characteristics could be beneficial. Considering a bounded memory, we can use data structures [1, 2, 5] that estimate frequency moments. Define the kth frequency moment by: $F_k = \Sigma_i f_i^k$

F_0 is defined to be the sum of distinct values in I_i. Through all F_k values, we can *compare* the recent topological data analysis with the previous one and see if the changes are significant. We can obtain more precise results by setting p as a suitable periodic parameter. Algorithm 1 illustrates a simple pseudo code for this approach.

2.2 Vector Differences by F_2 Esimators

Consider datasets $D(t_0), D(t_1), \cdots$. For some applications, one may think of the records of the datasets as distinct points in d-dimensional space. So, the datasets can be seen as vectors, hence their differences can be estimated. As an example, there can be cases in which the aim is to measure the changes of a fixed number of users in a system. If each point, i.e. a user, receives updates at time t_i from $\sigma(t_i - t_{i-1})$, then the point must be placed in a different position in the d-dimensional space. Observe that such changes may imply different topological features of $D(t_i)$ compared to $D(t_{i-1})$.

For more information on estimating the difference between two vectors, see the work of Cormode et. al. [4].

2.3 Point Estimation and Range Queries

Point Estimation. Suppose that there is a stream of data σ s.t. each item $x \in \sigma$ is a d-dimensional point. Depending on the application, one might think of the records of $D(t_0)$ as distinct points in d-dimensional space. So, each item in σ specifies an update on a record $y \in D(t_0)$. In such applications, σ can be seen as a *vector*. Therefore, one can estimate the value of each item of σ, and apply it to its corresponding point in $D(t_0)$. This helps to find out if there is a change in the topology of $D(t_0)$. In [4], the authors define *Point Queries* as follows: *Given vector V as a data stream, accept queries that specify index i, and return an estimate of the corresponding value of index i on V, namely $V[i]$*. The authors then provide an approximation for such queries. This can be used to detect changes in the topological features of $D(t_0)$.

Range Queries. Modifying the approach for point estimation, it is possible to answer range queries, helping to detect changes more conveniently on the topological features of $D(t_0)$. More specifically, rather than estimating each item in σ, items falling within a particular range can be estimated. Consequently, one may observe a change in the topology of $D(t_0)$. See [4, 9] for more information on estimating range queries.

3 APPLICATIONS

Many applications produce continuous high-dimensional data. Data stream algorithms are more useful in such scenarios, as they make the more realistic assumption that data evolves. That is, the distributions governing data generation may change [11]. In addition, many important tasks consist of high-dimensional data. So, the traditional distance measures become irrelevant, as the sparsity of data tends to make all points equidistant.

Big Data as a Stream. Due to the large size of enterprise networks and the number of users, the data is of significant volume and emerging at high velocity. Big data, e.g. network traffic, can be processed as streams [13]. The approach discussed in Section 2 can assist in detecting topological features in a streaming setting. Such features may ease the processing of big data.

The Deadlock Problem. In 2020, Cai et. al. [3] presented a low-overhead (compared to the previous works) deadlock predictor, called AirLock. Their main novelty is that, AirLock maintains the lock reachability graph involving locks only and detects cycles on it efficiently. For each detected cycle, it then constructs a predictive deadlock.

One can use the measurements discussed in this paper for detecting deadlocks in a streaming model using the changes in the topological features. For this, we need to find the topological structure of the deadlocks and search for them in the data set. In time if we find such a topology, then we detect a deadlock.

REFERENCES

[1] Alon, N., Matias, Y., Szegedy, M.: The space complexity of approximating the frequency moments. In: Proceedings of the twenty-eighth annual ACM symposium on Theory of computing. pp. 20–29 (1996)

[2] Bar-Yossef, Z., Jayram, T., Kumar, R., Sivakumar, D., Trevisan, L.: Counting distinct elements in a data stream. In: Randomization and Approximation Techniques in Computer Science: 6th International Workshop, RANDOM 2002 Cambridge, MA, USA, September 13–15, 2002 Proceedings 5. pp. 1–10. Springer (2002)

[3] Cai, Y., Meng, R., Palsberg, J.: Low-overhead deadlock prediction. In: Proceedings of the ACM/IEEE 42nd International Conference on Software Engineering. pp. 1298–1309 (2020)

[4] Cormode, G., Garofalakis, M.: Join sizes, frequency moments, and applications. In: Data Stream Management: Processing High-Speed Data Streams, pp. 87–102. Springer (2016)

[5] Cormode, G., Muthukrishnan, S.: An improved data stream summary: the count-min sketch and its applications. Journal of Algorithms **55**(1), 58–75 (2005)

[6] Edelsbrunner, Letscher, Zomorodian: Topological persistence and simplification. Discrete & computational geometry **28**, 511–533 (2002)

[7] Ghrist, R.: Barcodes: the persistent topology of data. Bulletin of the American Mathematical Society **45**(1), 61–75 (2008)

[8] Gidea, M., Katz, Y.: Topological data analysis of financial time series: Landscapes of crashes. Physica A: Statistical Mechanics and its Applications **491**, 820–834 (2018)

[9] Gilbert, A.C., Kotidis, Y., Muthukrishnan, S., Strauss, M.J.: How to summarize the universe: Dynamic maintenance of quantiles. In: VLDB'02: Proceedings of the 28th International Conference on Very Large Databases. pp. 454–465. Elsevier (2002)

[10] Perea, J.A., Deckard, A., Haase, S.B., Harer, J.: Sw1pers: Sliding windows and 1-persistence scoring; discovering periodicity in gene expression time series data. BMC bioinformatics **16**(1), 1–12 (2015)

[11] Pereira, C.M., de Mello, R.F.: Pts: Projected topological stream clustering algorithm. Neurocomputing **180**, 16–26 (2016). https://doi.org/https://doi.org/10.1016/j.neucom.2015.08.094, https://www.sciencedirect.com/science/article/pii/S0925231215016069, progress in Intelligent Systems Design

[12] Songdechakraiwut, T., Chung, M.K.: Dynamic topological data analysis for functional brain signals. In: 2020 IEEE 17th International Symposium on Biomedical Imaging Workshops (ISBI Workshops). pp. 1–4. IEEE (2020)

[13] Stephen, J.J., Gmach, D., Block, R., Madan, A., AuYoung, A.: Distributed real-time event analysis. In: 2015 IEEE International Conference on Autonomic Computing. pp. 11–20. IEEE (2015)

[14] Zomorodian, A., Carlsson, G., Collins, A., Guibas, L.: Persistence barcodes for shapes. International Journal of Shape Modeling (2oo5) (2004)