A Topological Approach to Representational Data Models

Emilie Purvine[1](✉), Sinan Aksoy[2], Cliff Joslyn[1], Kathleen Nowak[2],
Brenda Praggastis[1], and Michael Robinson[3]

[1] Pacific Northwest National Laboratory, Seattle, WA 98109, USA
{emilie.purvine,cliff.joslyn,brenda.praggastis}@pnnl.gov
[2] Pacific Northwest National Laboratory, Richland, WA 99354, USA
{sinan.aksoy,kathleen.nowak}@pnnl.gov
[3] American University, Washington, DC, USA
michaelr@american.edu

Abstract. As data accumulate faster and bigger, building representational models has turned into an art form. Despite sharing common data types, each scientific discipline often takes a different approach. In this work, we propose representational models grounded in the mathematics of algebraic topology to understand foundational data types. We present hypergraphs for multi-relational data, point clouds for vector data, and sheaf models when both data types are present and interrelated. These three models use similar principles from algebraic topology and provide a domain-agnostic framework. We will discuss each method, provide references to their foundational mathematical papers, and give examples of their use.

Keywords: Relational data · Vector data · Hypergraph models
Topological data models · Data-agnostic models

1 Introduction

The potential for drowning in data has become ubiquitous across all scientific domains. Instruments are built to collect massive amounts of data from anything we can get our hands on. When that does not suffice, those instruments are refined to reduce error rates and collect even more data. As data collection methods evolve so must the models used to study the systems. The way data are represented is important to how it is interacted with, visualized, and understood.

In many application domains, researchers have developed highly specialized methods to represent and build models for their specific data (e.g., signature-based malware detection in cyber security or bottom-up mass-spectrometry-based protein identification processes in proteomics). We are not advocating for ignoring or dismissing these targeted models. Rather, we offer three examples of topological methods that are broadly applicable across many domains and may allow researchers to see a new side of their data. Methods built over many years

S. Yamamoto and H. Mori (Eds.): HIMI 2018, LNCS 10904, pp. 90–109, 2018.
https://doi.org/10.1007/978-3-319-92043-6_8

within specific domains are well understood and yield predictable results. We propose these more generic topological methods to reveal hidden structure and strengthen prior hypotheses or build new ones.

2 Hypergraphs for Relational Data

Data that describe relationships are pervasive. The famous Enron email data set describes relationships between people via being on the same email [1]; protein interaction data help to understand what happens when proteins encounter each other in a cell [2]; bibliometrics tracks coauthorship relationships [3].

As one kind of relational mathematical model, graphs are popular in data science, typically used where there are sparse connections among a large collection of entities. However, graphs code pairwise associations between entities. Thus they should be seen as a special case of multi-way associations among an arbitrary number of entities, which are encoded by hypergraphs. In fact, some domain scientists have argued recently that hypergraphs are more appropriate [4–6]. As hypergraph-structured data are more general than graph-structured data, hypergraphs formally generalize graphs as mathematical objects. Therefore, we can conceive of hypergraphs literally as "multidimensional graphs." And, while hypergraph methods and applications are relatively rare, hypergraph-structured data are relatively ubiquitous, for example, whenever information presents naturally as set-valued, tabular, or bipartite data.

Hypergraphs typically have been studied from the network science (i.e., graph) perspective or in connection to each application separately. Our group is pursuing a hybrid graph and topological treatment. This allows the generalization from graphs to hypergraphs in a canonical, principled manner rather than for each application, which accounts for the high-dimensional structure contained in the multi-way relationships while remaining in agrement with network science and graph theory. We note that in network science, authors occasionally use the word *topology* to refer to graph structure or connectivity, e.g., [7]. Rather, we mean the mathematical domains of topology [8] and algebraic topology [9].

2.1 Definitions

Hypergraphs are generalizations of graphs [10]. Formally, a hypergraph $\mathcal{H}$ is a collection of subsets $\mathcal{E}$, called *edges*, of a set of elements V, called *vertices*, and we write $\mathcal{H} = (V, \mathcal{E})$. We sometimes refer to edges in an hypergraph as *hyperedges* to distinguish them from graph edges. Consequently, whereas a graph edge, (v, w), consists of precisely two vertices, a hyperedge, $f \in \mathcal{E}$, can contain any number of vertices, $f \subseteq V$. The number of edges a vertex belongs to is its *vertex degree*. Conversely, the number of vertices contained within an edge is its *edge cardinality*. A *k-uniform* hypergraph is a hypergraph where all edges have cardinality k. In particular, a 2-uniform hypergraph is just a graph.

Research on hypergraphs, particularly in the mathematics literature, mostly is limited to studying k-uniform hypergraphs. For instance, much work on the spectral theory of hypergraphs [11,12], hypergraph coloring [13], extremal problems [14], and hypergraph transversals [15] considers the k-uniform case. However, as real hypergraph data are often non-uniform, we require tools to analyze, interact with, and understand such data.

A fundamental notion underlying a multitude of graph analytics tools is *graph distance*. Oft-studied metrics such as diameter, average shortest path length, and centrality measures are all based on measuring how "far" vertices in an network are from each other, defined as the length of the shortest walk connecting them. Beyond distances and diameters, walks in a graph can be used to make sense of large data through the concept of a random walk. For example, an efficient implementation of PageRank, one of the algorithms Google uses to rank webpages in search results, relies heavily on random walks in graphs [16]. However, extending the concept of a walk to general hypergraphs introduces subtleties that do not arise in the case of graphs.

In a graph, a *walk of length k* is a sequence of vertices $v_0, v_1, \ldots, v_k$, each of which is adjacent to the next via an edge. Because two adjacent vertices belong to exactly one edge in a graph (and two incident edges intersect at exactly one vertex), we can equivalently describe a walk as a sequence of incident edges or adjacent vertices, i.e.,

$$\underbrace{v_0, v_1}_{\text{adjacent}}, \ldots, \underbrace{v_{k-1}, v_k}_{\text{adjacent}} \quad \longleftrightarrow \quad \underbrace{e_1}_{\{v_0,v_1\}}, \ \ldots, \ \underbrace{e_k}_{\{v_{k-1},v_k\}}.$$

A third equivalent way to represent a walk is the alternating vertex-edge form: $v_0, e_1, v_1, e_2, \ldots, e_k, v_k$. When vertices are not allowed to repeat, we call the walk a *path*. These three equivalent definitions of a graph path also happen to coincide with the topological definition of a path, i.e., a continuous map from the interval $[0, 1] \subset \mathbb{R}$ to the graph thought of as a topological space.

In contrast, this three-way equivalence does not hold for hypergraphs: two hypergraph edges can intersect at any number of vertices, and two vertices can belong to any number of shared edges. This simple observation, along with the goal to keep the correspondence with the topological definition of a path, motivates two walk concepts for hypergraphs that are dual but nonetheless different: walks on the vertex level (consisting of successively adjacent vertices) and walks on the edge level (consisting of successively intersecting edges). For ease of presentation, we will focus on edge-level walks, keeping in mind there is a dual concept for vertex-based walks. We define an s-walk on a hypergraph, where s controls the size of the edge intersection, as follows:

Definition 1. *For $s \geq 1$, an s-walk of length k between edges $f, g \in \mathcal{E}$ is a sequence of edges, $f = e_0, e_1, \ldots, e_k = g$, where for $i = 1, \ldots, k$ and $I_i = e_{i-1} \cap e_i$, we have $|I_i| \geq s$ and $e_{i-1} \neq e_i$.*

Observe that a 1-walk on a graph corresponds to the usual graph walk, whereas s-walks for $s \geq 2$ are only possible for hypergraphs. We note that several notions closely related to s-walks have previously appeared in the literature. In [17], Lu and Peng consider walks in k-uniform hypergraphs to build an

s-Laplacian matrix of a hypergraph. Wang and Lee [18] consider edge intersection properties in walks unrelated to the size of the intersections. Perhaps most pertinent to this investigation, Bermond et al. [19] introduce and analyze *s-line graphs* of hypergraphs, which are graphs derived from hypergaphs by representing each hyperedge as a vertex, and link two such vertices if their corresponding hyperedges intersect in at least s vertices. Our proposed s-walks on hypergraphs correspond precisely to graph walks on these s-line graphs. Hence s-line graphs may be used as auxiliary graphs for computing s-walk-based metrics.

A number of basic, yet important, properties of walks in graphs immediately extend to s-walks on hypergraphs. Consequently, just as the length of the shortest walk defines a bona fide distance metric for graphs, s-distance, defined as follows, serves as a distance metric in hypergraphs.

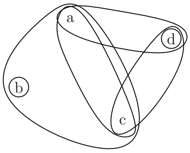


Fig. 1. An illustration of a hypergraph with edges $\{a, b, c\}$, $\{a, c\}$, $\{a, d\}$, $\{c, d\}$, $\{b\}$, and $\{d\}$. This hypergraph has 1-diameter of 3 (which is achieved by the 1-walk b, abc, ad, d), average 1-distance of 1.5, infinite 2-diameter, and 3-diameter of 0.

Definition 2. *Let $\mathcal{H} = (V, \mathcal{E})$ be a hypergraph and $\mathcal{E}_s$ denote the set of hyperedges in $\mathcal{H}$ with at least s vertices. Define the distance function $d_s : \mathcal{E}_s \times \mathcal{E}_s \to \mathbb{Z}_{\geq 0}$*

$$d_s(f, g) = \begin{cases} \text{length of shortest } s\text{-walk} & \text{if } s\text{-walk between } f,\ g \text{ exists} \\ \infty & \text{otherwise} \end{cases}.$$

From this distance metric, a number of graph-theoretic concepts generalize naturally, For example, s-diameter (resp. average s-distance) of a hypergraph is the maximum (resp. average) s-distance between all pairs of hyperedges with at least s edges. A hypergraph is s-connected if its s-diameter is finite. We illustrate these with an example in Fig. 1.

From the perspective of data analytics, s-walk-based metrics may uncover higher-order interactions among the entities that are evident neither from classical hypergraph walks which don't control for intersection size nor the raw relational data itself. For example, a user-group hypergraph exhibiting lower average 1-distance than 4-distance suggests groups are more closely related via sparse overlappings of their members. In general, depending on the hyperdata in question, it may be the case that higher-cardinality intersections signify stronger ties between groups, or only intersections within a certain size range are meaningful due to noise or other artifacts in the data. In such cases, s-walk-based metrics provide a framework that not only recovers classical hypergraph and graph walks (by taking $s = 1$), but also allows the flexibility to filter and control for higher-dimensional interactions as well.

Hypergraph Visualization and Sensemaking. Understanding hypergraph data can be significantly more difficult than understanding graph data. Adequate hypergraph visualizations are lacking when data get large, and more intricate interactions inevitably lead to more complex data analysis and interpretation. However, making the move to use hypergraphs as representational models should allow researchers to learn more about their complex relational data. Sensemaking in graph data has been an active research area in recent years [20–22]. Before that large graph visualization became prominent [23,24]. If researchers in application domains focus more on hypergraphs over graphs, we expect that work on sensemaking and visualization for graphs can extend into hypergraphs. For example, Hoff et al. observe that although their latent space approach to social network analysis and visualization is focused on binary relations (i.e., graphs), the work can extend to more general relational data [25].

2.2 Example: Enron Email Data

Email or other mass communications are a good example of data that naturally fit into a hypergraph structure. Originally released in 2004 by the Federal Energy Regulatory Commission during its investigation, the Enron email data set is a highly cited and studied corpus of emails from the inboxes of roughly 150 Enron employees [1]. While there has been considerable research on classification and clustering based on the email text [1,26,27], our focus is on the hypergraph structure of user interactions implied by the set of emails.

To date, the Enron corpus has been studied overwhelmingly with graphs. Properties such as the distance between two users or the diameter of the entire graph can shed light on the extent to which knowledge is shared or disseminated within the company. A graph can be built from these data by letting the vertices be the set of users, or email addresses, in the data. Then, an edge is created between `user1` and `user2` if there is an email from `user1` to `user2` or vice versa (see graph type 1 in Fig. 2). This graph can help understand how information flows through the group of users and has been well studied in the literature [28–31]. Another potential graph model considers the set of recipients for each email, if there are multiple, and adds all possible edges between the users in that set (see graph type 2 in Fig. 2). Rather than flow of information, this models shared knowledge. Connected users have some common knowledge by virtue of receiving the same email.

This second representation lends itself most naturally to a hypergraph structure. Rather than connecting all pairs of users that receive the same email, we create a single hyperedge for each email. The vertices in that hyperedge are all of the recipients of that email (see hypergraph in Fig. 2). To build this hypergraph, we used a data set hosted by KONECT [32] that provides a set of `sender, recipient, timestamp` rows. We collected all recipients that occur for any given ⟨`sender, timestamp`⟩ pair and put them into a single hyperedge. Admittedly this assumes that only one email was sent from the given sender at the given time. Although that may not always be true, the incidents of email collision are probably rare.

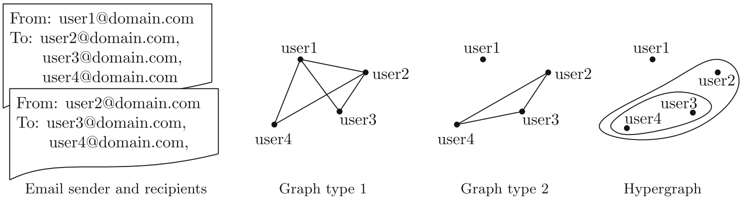


Fig. 2. Example creation of two types of graphs and one hypergraph from an email. Notice that graph type 2 would be the same if only the first email were seen, while the hypergraph representation is able to disambiguate the two emails.

It is not our goal to do a comprehensive study of the Enron email hypergraph. Instead, we aim to show one example, s-diameter, where the hypergraph can be more informative to a user than a traditional graph. Rather than considering the Enron hypergraph as one single system, we break the data into two-week subsets. Figure 3 shows the vertex and hyperedge counts over time. The maximum number of hyperedges (emails) occurs in the hypergraph representing the two-week period October 19-November 2, 2001, which coincides with the beginning of the Securities and Exchange Commission inquiry into Enron finances. We will study the evolution of the s-diameter over time for various values of s and how that evolution correlates with the vertex and hyperedge counts.

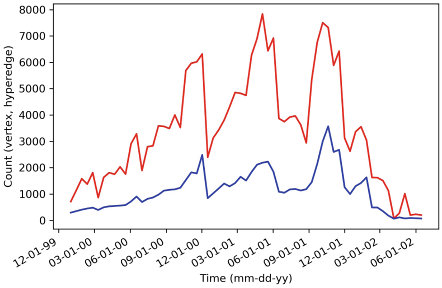


Fig. 3. Vertex (red) and hyperedge (blue) counts in the two-week Enron hypergraphs. (Color figure online)

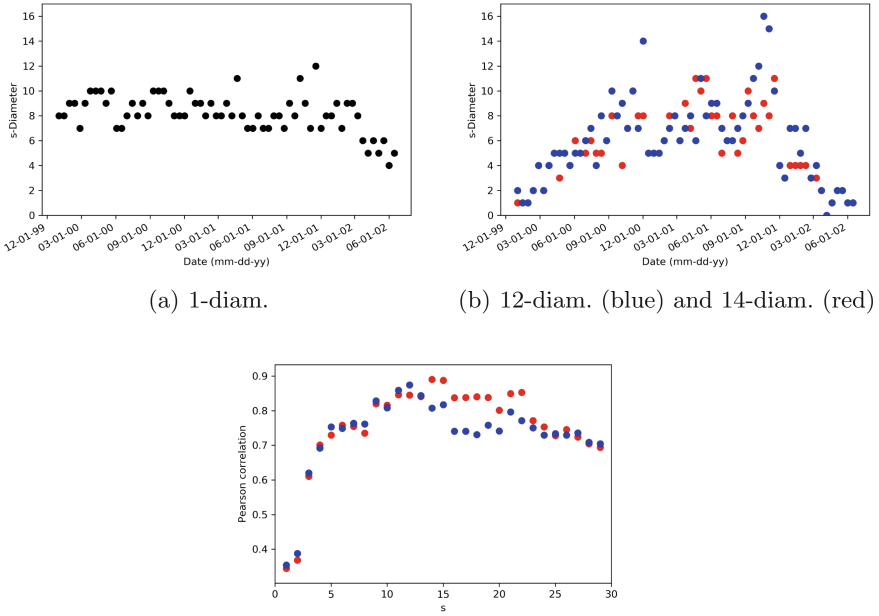
Notably, it is not difficult to show that if the 1-diameter of a hypergraph is d, then the diameter of the associated graph is either $d-1$, d or $d+1$. Specifically, if we consider the graph where each hyperedge is replaced with graph edges for all pairs of vertices within that hyperedge, then its graph diameter would be roughly equal to the 1-diameter of the hypergraph. The question is: is there added value to calculating the s diameter for any value of $s > 1$? Do we learn any new information that the 1-diameter could not tell us? In the case of the Enron email hypergraph, the answer appearss to be: *yes*.

We consider the s-diameter of only the largest s-connected component of each hypergraph (rather than saying that s-diameter is ∞ if the hypergraph is not s-connected). In Fig. 4(a), we show the 1-diameter of the Enron hypergraph over time. Although there are two maxima roughly around the same times as two of the peaks in the vertex and hyperedge counts, there is a slight downward or flat trend in the 1-diameter (slope of a linear fit is -3×10^{-8}). The Pearson correlation between the 1-diameter and the number of vertices (resp. hyperedges) is 0.34 (resp. 0.35). If this were a graph, we would be forced to stop here and perhaps conclude that the diameter, or largest path distance between two users, of the Enron graph is only slightly correlated to the number of users or emails, i.e., connection distance does not seem to grow as the company grows. However, because of the richer structure contained in the hypergraph, we are able to look further at s-diameter for $s > 1$. When we do this, it becomes clear that the larger values of s change our perception of the distances in the system. As s grows so does the correlation between s-diameter and both vertex and hyperedge count. Figure 4(c) shows how the correlation between the s-diameter and the vertex (resp. hyperedge) counts increases as s increases, reaching a maximum at $s = 14$ (resp. $s = 12$) before decreasing slightly. In Fig. 4(b), we show the 12- and 14-diameters over time. The resemblance of these to the corresponding vertex and hyperedge counts in Fig. 3 is clearly much higher than the 1-diameter, as expected by a correlation of nearly 0.9. This indicates that as we force the intersection size (s) to be larger, we must travel farther to form a path between two users. This, by itself, is unsurprising as putting more constraints on the intersections could never require shorter paths. However, what is surprising is that paths exist at all when intersections are required to be large.

To validate that high correlation of the s-diameter with vertex and hyperedge counts for large values of s does not stem from random chance, we created a sequence of random hypergraphs using a Chung-Lu model [33]. This sequence of random hypergraphs has the same vertex degree and edge size distributions (in expectation) as the sequence of two-week Enron hypergraphs. In these random hypergraphs, the 1-diameter has slight negative correlation with the vertex and hyperedge counts (-0.34 and -0.24 respectively), but by $s = 4$, there are no s-paths. In other words, all intersections between hyperedges in the random model are of size 3, at most. This means that the existence of s-paths for large s in the Enron hypergraphs along with the high correlation with vertex and edge size is significant. There is more value in considering the Enron data as a hypergraph than as a graph.

3 Topological Analysis of Vector Data

We pivot now from complex relational data to high-dimensional numeric data. This type of data, which can be thought of as a high-dimensional point cloud, is common and often studied under the lens of dimensionality reduction tools, such as principle component analysis (PCA) or nonnegative matrix factorization (NMF). However, these techniques can introduce loss of information and may



(a) 1-diam.

(b) 12-diam. (blue) and 14-diam. (red)

(c) Correlation of s-diameter with vertex count (red) and hyperedge count (blue).

Fig. 4. (a), (b) s-diameters versus time for the two-week Enron hypergraphs. (c) Correlation of s-diameter versus s. (Color figure online)

make attribution of anomalies difficult. Instead, our group, and others, have worked with data's intrinsic high dimensionality using persistent homology, a technique within topological data analysis (TDA). Persistent homology skirts around the curse of high computational times that often come along with high-dimensional data by considering only distances between points rather than the ambient dimensionality.

3.1 Definitions

Topology is the study of geometric properties that are unaffected by bending or stretching a space (such properties are called *invariants*). TDA approaches identify changes in a system of interest by tracking topology that describes the system. Because topological properties are invariant under scaling, they ignore noise and only identify true fundamental changes to the underlying system.

Most TDA methods operate on (topological spaces derived from) numerical point clouds, so the first step must be to construct vectors $x \in \mathbb{R}^n$ that represent the system state. In some cases, data come in the form of vectors natively. For example, if the system is a biological sample, the measured data could be protein abundance values, i.e., an ordered list of measured values for each protein found in the sample. This list becomes the vector representing the biological sample state. However, there are other systems where a numerical vector is not the measured data. Consider a cyber network where the data describing the system is collected by a *packet analyzer* (also called a network analyzer or packet sniffer). This computer program or piece of hardware can intercept traffic as it passes over a digital network or part of a network. As data streams flow across the network, the sniffer captures each packet and generates a corresponding log. The information recorded in each log can vary with the packet analyzer, but it almost always contains time intercepted, source Internet Protocol (IP) address, destination IP address, source port, destination port, protocol, number of packets, and number of bytes. Thus, given a set of flow logs, some natural quantitative statistics to compute include number of source/destination IPs, number of packets, number of bytes, number of (source IP, destination IP) pairs, and number of protocols.

In this work, we concentrate on time series data, but other organizing principles, such as geographic area or biological family, can be used to group data. To vectorize time series data, there are two main approaches. The first, which can be thought of as a *feature vectorization*, is to construct a vector of statistics for each time window, from t to $t + \epsilon$. For example, for a cyber network given all network flow logs in the time window we could construct a vector $x_t \in \mathbb{R}^4$ as

$$\begin{aligned} x_t = (&\#\text{ of source IPs seen}, \#\text{ of destination IPs seen}, \\ &\text{total }\#\text{ of packets sent, total }\#\text{ of bytes sent}). \end{aligned}$$

Then, a collection of time windows yields a set of feature vectors; a point cloud. An example of this is featured in Sect. 3.2.

Another way to vectorize a time series is to use a *Takens embedding*. Given a time series $X(t)$, fix a lag parameter $\eta > 0$, and a dimension $m \in \mathbb{Z}_{>0}$. The Takens embedding of the time series then is a lift to the map

$$t \to x_t = (X(t), X(t-\eta), X(t-2\eta), \ldots, X(t-(m-1)\eta)).$$

In other words, the time series sequence $X(0), \ldots, X(N)$ becomes a set of m-dimensional points $x_{(m-1)\eta}, \ldots, x_N$, or another point cloud. Takens proved that under the correct choice of m and η, this embedding represents the underlying dynamics of the dynamical system [34][1]. In particular, if the time series is periodic, the Takens embedding will contain a cycle (see example in Sect. 3.3).

Both vectorization methods for time series provide mathematical abstractions, or representations, for the data that are domain agnostic. Specifically, once the transformation to a point cloud has been made, the analysis proceeds only based on those values not on the original domain of the data.

The main tool in TDA for analyzing a point cloud is persistent homology, a topological invariant that distinguishes spatially robust topological features from smaller ones more likely to be noise. Broadly speaking, persistent homology takes a point cloud; constructs a sequence of topological spaces over the

[1] There are ways to intelligently choose m and η. For example, Khasawneh and Munch in [35] use false nearest neighbors to choose m and the first zero of the autocorrelation function to choose η.

point cloud by considering a filtration of spatial resolutions; and computes a set of 2-dimensional points, called a *persistence diagram*, where each point (b, d) delineates the birth and death of an important topological feature in the filtration sequence of topological spaces. Shorter-lived features can be thought of as noise, while longer-lived ones represent the more robust features in the space, representative of the *true* shape of the underlying space.

Given a point cloud, $V \subset \mathbb{R}^n$, the two natural topological filtrations that can be constructed are the Čech filtration and Vietoris-Rips filtration. For both, one starts by placing a ball of some small radius ϵ around each data point. The *Čech complex*, C_ϵ, is constructed by taking all subsets of points whose associated balls have a nonempty intersection. On the other hand, the *Vietoris-Rips complex*, R_ϵ, is constructed by taking all subsets of points whose associated balls have nonempty *pairwise* intersections. The two constructions are demonstrated geometrically in Fig. 5, where two-element sets are shown with a line connecting the two elements, three-element sets are filled-in triangles, and four-element sets are filled-in tetrahedra. Notice that the Vietoris-Rips complex has a filled-in triangle connecting the three points on the right, whereas the Čech complex has an open triangle. This is because the corresponding circles overlap pairwise but not all together. The triangle and tetrahedron on the left are present in both because of the respective 3- and 4-way intersections. A filtration is formed by increasing ϵ and growing the complexes accordingly.

Given a filtration, the persistence diagram can be computed for topological features of different dimensions (see Fig. 6). Features in dimension 0 are connected components; in dimension 1 they are loops (e.g.,the open triangle or longer cycles); in dimension 2, (not pictured), they are voids such as a hollow tetrahedron. Topological features in dimensions larger than two become difficult to conceptualize. For a rigorous discussion of the persistent homology methodology, the interested reader is directed to [36–38].

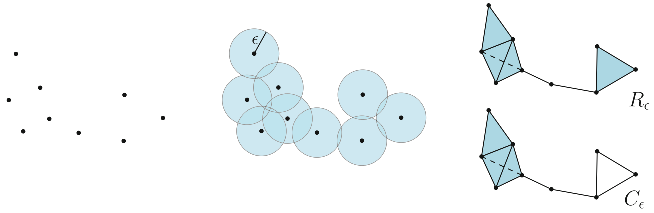
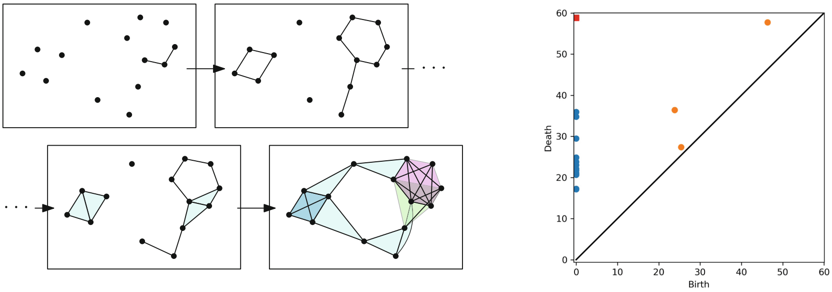


Fig. 5. A point cloud (left) can be completed to a Čech complex $\mathcal{C}_\epsilon$ (lower right) or to a Vietoris-Rips complex $\mathcal{R}_\epsilon$ (upper right) based on a resolution parameter ϵ (center).

To topologically compare two point clouds using persistent homology, their persistence diagrams are compared using a distance metric. The most intuitive and widely used persistence diagram distance metric is the Wasserstein distance. Roughly speaking, this metric details the minimum movement required to move points in one persistence diagram to those in another diagram where points are



(a) Vietoris-Rips filtration

(b) Persistence diagram

Fig. 6. Example Vietoris-Rips filtration and corresponding persistence diagram for a point cloud. Blue points in the persistence diagram (and the red square) correspond to 0-dimensional features, and orange points correspond to 1-dimensional features. Note that there is always one feature in dimension 0 with infinite death time (the red square) corresponding to the one connected component that does not die once fully connected. (Color figure online)

allowed to move to the diagonal (where birth = death) if no better choice exists. The exact formulation of the Wasserstein distance is not required to understand the concept, so we refer the reader to Sect. 3 in [39].

To use persistent homology for anomaly detection, assume we are given a system baseline representing normal behavior (if this is not the case, a baseline can be learned from an initial time period within the data). All data are broken into overlapping widows, and a numerical vector is associated to each window using one of the procedures already described. Collecting the vectors from the baseline windows provides a point cloud, $\mathcal{B}$, which represents normal system behavior. Computing persistent homology on $\mathcal{B}$ provides a reference barcode, C. Then, for each new window, construct its numerical vector x using the same procedure. Compute the persistence barcode, C', for $\mathcal{B} \cup \{x\}$ and the distance between C and C'. This distance affords a measure of how much the system changed by adding this one new window. For a detailed description of this anomaly detection procedure, see [36]. One advantage to this technique is an interpretation via the persistence stability theorem [40,41]. In short, this theorem says that if the distance between the barcodes is large, then there must be a large distance in the point clouds. Because of how we created our two point clouds, this means that a large distance in the barcodes can only come from x being far away from the baseline point cloud.

Interpretability and Interaction. This method of transforming data from its raw form into a high-dimensional point cloud may seem to be such an abstraction that interpretability is lost. However, the opposite seems to be true. The topological summary through the lens of persistent homology simplifies the data by cutting through the noise (by way of the persistence stability theorems) to find the most robust features [42]. Additionally, keeping the data in its natural high-dimensional setting rather than using dimension reduction techniques allows for root cause analysis once anomaly scores have been calculated.

While not discussed herein, Mapper [43] is another tool worth mentioning because of its wide use to visualize and interpret high-dimensional data. Mapper uses local clustering behavior to reconstruct global data shape. It has been used in a variety of domains to provide new interpretations of data sets and discover previously unknown clustering behavior [44]. A "users guide" for TDA is provided in [45] that offers more detail into practical uses, interpretations, and specific implementations of both persistent homology and Mapper.

3.2 Cyber Kill Chain

Here, we provide an example of using our TDA anomaly detection algorithm to identify malicious behavior in a cyber network. It is not our goal to describe the cyber background in detail. Rather, we hope to show the method's flexibility. For more detail on cyber kill chains and attacks, see [46]. The data for our case study were simulated by a cyber security team at Pacific Northwest National Laboratory. The experimental network consisted of 16 virtual machines: 15 Linux workstations and one web server. Each workstation ran a Markov chain script to simulate browsing the external web and the corporate network site. The entire experiment ran in four phases for a total of 52 min:

Phase 1: 15 min of baseline network traffic
Phase 2: Network reconnaissance using nmap port scans
Phase 3: Normal network traffic to simulate an attacker "planning"
Phase 4: Remote buffer overflow, gain privileged shell, change root password.

We ran our persistent homology anomaly detection algorithm with two different feature vectorization strategies. For the first, we constructed a vector of length 9 with the following entries:

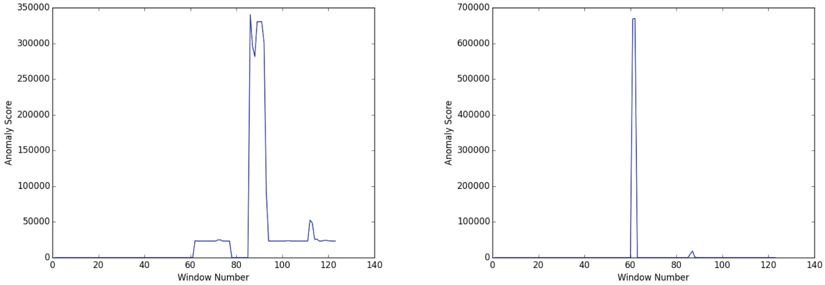
- # distinct source IPs (SIP)
- # distinct (SIP, DIP) pairs
- total # bytes
- # distinct (SIP, Source Port) pairs
- # records
- # distinct destination IPs (DIP)
- # distinct protocols
- total # packets
- # distinct (DIP, Dest Port) pairs

Then, each entry in the vector was normalized by the window length (60 s in our case). The second scheme was similar to the first. Slightly different statistics were targeted to specific protocols and normalized by the number of the records in each window. For each protocol $p \in \{\text{ICMP, TCP, UDP}\}$, let $|p|$ be the number of records in the window with protocol p. Then, the vector contains:

- $|p|/(\text{total number of records})$
- $(\text{number of packets in protocol } p)/|p|$
- $(\text{number of bytes in protocol } p)/|p|$

- (bytes in protocol p)/(packets in protocol p)
- $|p|$/(bytes in p/packets in p)

Figure 7 shows the anomaly scores for each window. We see that both have clear spikes, but they pick up on different phases of the attack. The first vectorization (Fig. 7(a)) measures the average statistical properties for each record. Notice that this flags anomalies during the later phase of the scenario. Specifically, it picks up on the remote buffer overflow exploit when more data are being sent per record than one would expect. Likewise, the second vectorization (Fig. 7(b)) measures the average statistical properties per unit time. This flags the reconnaissance phase because a port scan is generating many more records than usual.



(a) Record-normalized vectorization

(b) Time-normalized vectorization

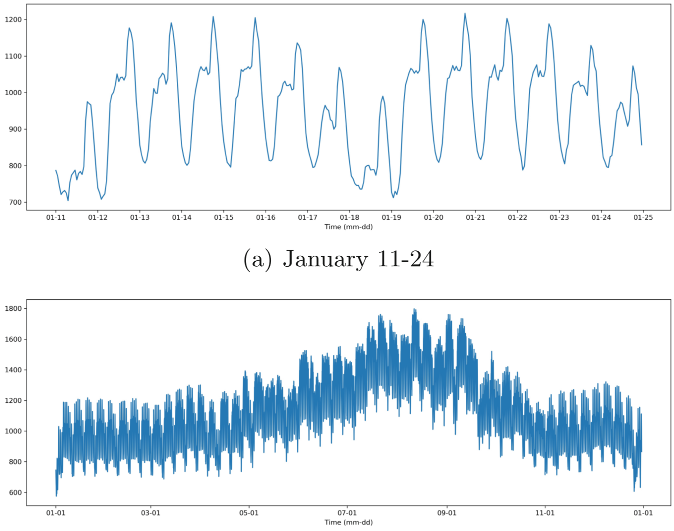
Fig. 7. Anomaly scores (y axis) over time (x axis) for our cyber kill chain data set.

3.3 Power Grid Demand Data

Operating a power grid is a careful balance between generating enough power to satisfy consumer demand without generating too much power that cannot be stored and is therefore wasted. To solve this problem of *unit commitment* [47], operators must constantly forecast the demand and increase or decrease production to match the forecasted data. Understanding the cyclic patterns in demand can be crucial to the demand forecasting problem.

The sample data used in this example was generated from a 240-bus model of the California Independent System Operator (CAISO) electric power market within the Western Interconnection grid. This model was developed in [48] as an extension to earlier 225-bus [49] and 179-bus [50] CAISO models. Using data derived from published CAISO transmission studies and Western Electricity Coordinating Council's Transmission Expansion Planning Policy Committee to seed the model, they formed resource data for system conditions modeled as the year 2004. Others have used this model and data to study transmission planning, e.g., [51]. In this work, we only use the hourly demand profiles for one of the available 21 sub-regions of CAISO to show how the Takens embedding can help discover cyclic patterns.

In Fig. 8(a), we show sample demand for a two-week interval (Sunday–Saturday) in one region of the CAISO. Notice that Sunday has the lowest demand followed by Saturday and then Friday, while Monday–Thursday have similar demand. A two-dimensional ($m = 2$) Takens embedding for just Monday–Thursday, with lag $\eta = 5$ (the first approximate zero of the autocorrelation function), is shown in Fig. 9(a) alongside its persistence diagram in Fig. 9(b)[2]. Points close to the diagonal represent short-lived noisy topological features, whereas the single point farther away from the diagonal signals that there is a one-dimensional topological feature (a loop) in the Takens embedding. The presence of this loop tells us there is periodicity to the data, while the *maximum persistence*, the maximum $(b - d)$, gives us an idea of how much this is real periodicity versus just due to noise.



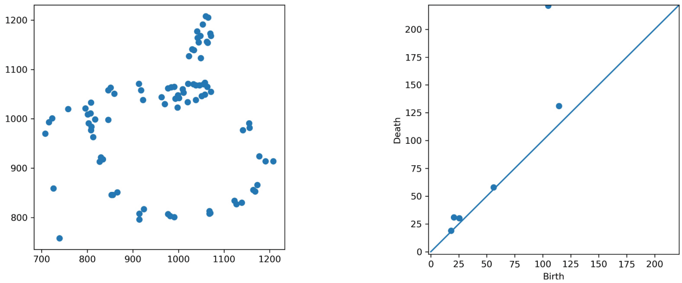
(a) January 11-24

(b) Full year

Fig. 8. Modeled demand in the "20MEXICO" region of the CAISO for two-weeks and the full year of sample generated data. The y axis is in kilowatt hours (kwH).

If we instead take a year-long view, as shown in Fig. 8(b), it becomes clear that there is still some periodicity within the weeks, but the demand rises and falls with the seasons. The Takens embedding of the full year will not yield any interesting topological features because of the seasonal variability in behavior. Still, local Takens embeddings can be used to discover and classify different types of local periodic behavior.

[2] The persistence diagram is a 2D representation of the barcode. A bar with birth $= b$ and death $= d$ is plotted at the point (b, d). Because death occurs after birth, all points will be above the $y = x$ diagonal.



(a) Two-dimensional Takens embedding for January 12-15

(b) Persistence diagram for Takens embedding

Fig. 9. Two-dimensional Takens embedding and corresponding persistence diagram.

4 Sheaf Models

Data may also come in a combination of relational tables and vectors. For example protein expression information within a protein interaction network, or employee rank and salary for people in the Enron email database. Sheaves are a way to combine both hypergraphs and TDA to infer global information from the local data. Whereas typically this type of data is combined in an ad hoc manner, sheaves provide a canonical principled inference method [52,53]. The basics of sheaf theory require too many details for this survey. Thus, we provide only a toy example here and refer the curious reader to [52].

Consider the data shown in Table 1, which depicts what several people reported to police when questioned as witnesses to a crime. Each observation is local to the individual, but we ask a global question: How many perpetrators might there be, and who agrees? This table represents a scenario where each row corresponds to an *observation*, or, in this case, the story from a single witness. Each observation consists of one or more data *fields*, each corresponding to a column in the table. Each column specifies a data *type*, the observed property.

Table 1. Witness descriptions of crime scene details (perpetrator and incident time).

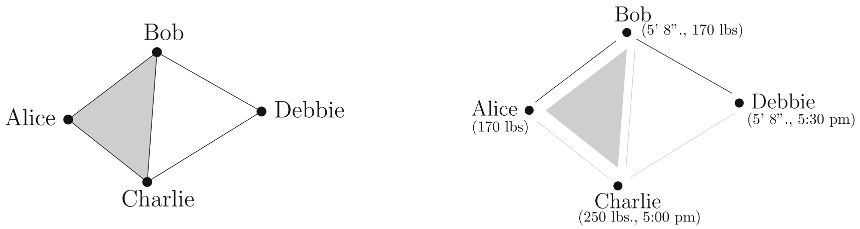
Row	Height	Weight	Time
Alice	(null)	170 lbs	(null)
Bob	5’8”	170 lbs	(null)
Charlie	(null)	250 lbs	5:00 pm
Debbie	5’8”	(null)	5:30 pm

A variety of distinct hypergraphs can be constructed from this kind of tabular data. For this example, it is most convenient to consider the hypergraph $\mathcal{T}$ in which each row corresponds to a vertex, and each edge corresponds to a set of rows with at least one nonempty column in common [54]. For Table 1, this hypergraph is shown in Fig. 10(a). Notice in particular that each vertex is itself a 1-edge. The presence of the 3-edge indicates that three witnesses (Alice, Bob, and Charlie) reported the criminal's weight. Although Debbie did not report the weight, the 2-edges indicate that, like Bob, she reported the height, and, like Charlie, she reported the time of the incident.

Formally, let $T = (t_{i,j})$ be a data table with m rows and n columns. Notationally, we will let $i = 1, \ldots, m$ index rows and $j = 1, \ldots, n$ index columns. Then, $\mathcal{T} = (V, \mathcal{E})$ is a hypergraph whose vertices $V = \{1, \ldots, m\}$, and $e = \{i_1, \ldots, i_k\} \subset V$ is an edge if there is a j such that $t_{i,j}$ is nonempty for all $i \in e$.

Observe that this hypergraph only represents which fields (columns) of the data table are shared among observations (rows), but it does not account for the data types of the entries nor the entries themselves. This is remedied by building a *sheaf* structure on the hypergraph. As a technical matter, sheaves are not built on hypergraphs, rather on topological spaces. However, because this hypergraph has the property that all subsets of a given edge are also edges themselves, there is a natural way to associate a topological space (the Alexandrov topology).

Thus, we can represent the entirety of the data by constructing a sheaf in the following way [55]. Notice that each edge in the hypergraph implicitly specifies a set of columns – ones that are nonempty in the rows corresponding to the edge. For the 1-edges (which are also the rows), these are merely the nonempty columns in each row. Hence, to each edge of the hypergraph, we associate the set of all possible data vectors indexed over the nonempty columns corresponding to that edge. Each of these sets of data vectors are called *stalks* over the edges.



(a) Hypergraph associated to the data in Table 1.

(b) A graphical representation of the consistent data in Table 1.

Fig. 10. Underlying hypergraph and consistent sections for data in Table 1.

Given this construction, the data table entries are elements of the stalks over the 1-edges. However, what about the stalks for the other edges? If two edges are adjacent, they share at least one column in the data table. Thus, their stalks contain a common subspace, which means those corresponding entries are comparable. If those rows agree upon projecting to the common subspace, then this projection is what should be chosen as the element of the stalk. On the other hand, if the data elements are not in agreement on this common subspace, this implies that the rows themselves are in conflict. If each column is additionally associated to a metric, a numerical value can be assigned to the disagreement. The maximum amount of disagreement over the entire hypergraph is called the *consistency radius* [52] and it can provide evidence for determining the root cause of disagreements, e.g., noise, equipment malfunction, or a faulty underlying hypergraph model. Additionally the consistency radius can serve as a basis for imputing corrections to noisy data.

A collection of edges and elements from the stalks over those edges that agree upon projection is called a *section*. Larger sections, consisting of more rows, represent a more self-consistent data set, while smaller sections, those where adding any other row causes a conflict, are less so. In the specific Table 1 example, those edges in which there is agreement are shown in Fig. 10(b), while those in disagreement are grayed out. The largest section consists of the triple Alice, Bob, and Debbie because Alice and Bob agree that the criminal weighed 170 lbs., and Bob and Debbie both said that he or she was 5'8". On the other hand, Charlie (alone) also is a section, but it cannot be extended to a larger section without disagreement. In fact, his disagreements might indicate that Charlie witnessed an entirely different criminal than Alice, Bob, and Debbie.

The Local-to-Global Promise of Sheaf Theory. Since its origin dates back to the 1930s and 40s, sheaf theory is not a new mathematical discipline. However, using sheaves in data analysis is rather new. Sheaf theory is about local-to-global inference involving a collection of overlapping and interacting local data sources to infer a global state. Thus, as data sources have become more universal, the need to synthesize disparate sensors around a common theme is more important.

5 Discussion

In this paper, we have described three related models resting in the mathematics of topology for representing and exploring data: hypergraphs for relational data, point clouds for vector data, and sheaves when both are present and related. Moreover, these mathematical structures each incorporate domain-agnostic analysis tools that are able to discover insights into the data that domain-specific tools may miss through the use of higher-order structures. First, topologically inspired paths in hypergraphs see chains of highly intersecting groups that can shed light on how knowledge is shared through a network. Next, persistent homology applied to point clouds uses topological structure to detect anomalies and differentiate between noise and periodicity. Finally, sheaf theory provides a local-to-global inference mechanism. As the Internet of things grows and sensors become both smaller and more ubiquitous, a principled and general technique, like sheaf theory, will be necessary to make sense of it all. Topology, on its own, may not be able to draw declarative conclusions or confirm phenomenological hypotheses. However, these methods can provide a different kind of feature for more robust classification algorithms.

Acknowledgements. We wish to thank Prof. Francisco Munoz, University of Chile, for providing the sample power grid data studied in Sect. 3.3, and Will Hutton, Pacific Northwest National Laboratory, and his team for providing the data described in Sect. 3.2. This work was supported in part by (a) the Applied Mathematics Program of the Office of Advanced Scientific Computing Research within the Office of Science of the U.S. Department of Energy (DOE) through the Multifaceted Mathematics for Complex Energy Systems (M2ACS) project, (b) the Asymmetric Resilient Cybersecurity Initiative at Pacific Northwest National Laboratory, and (c) the High Performance Data Analytics program at the Pacific Northwest National Laboratory (PNNL). PNNL is operated by Battelle for the United States Department of Energy under Contract DE-AC05-76RL01830.

References

1. Klimt, B., Yang, Y.: The enron corpus: a new dataset for email classification research. In: Boulicaut, J.-F., Esposito, F., Giannotti, F., Pedreschi, D. (eds.) ECML 2004. LNCS (LNAI), vol. 3201, pp. 217–226. Springer, Heidelberg (2004). https://doi.org/10.1007/978-3-540-30115-8_22
2. Pržulj, N.: Protein-protein interactions: making sense of networks via graph-theoretic modeling. BioEssays **33**(2), 115–123 (2011)
3. Newman, M.E.J.: Coauthorship networks and patterns of scientific collaboration. Proc. Nat. Acad. Sci. **101**(Suppl. 1), 5200–5205 (2004)
4. Silva, J., Willett, R.: Hypergraph-based anomaly detection of high-dimensional co-occurrences. IEEE Trans. Pattern Anal. Mach. Intell. **31**, 563–569 (2009)
5. Guzzo, A., Pugliese, A., Rullo, A., Saccá, D., Piccolo, A.: Malevolent activity detection with hypergraph-based models. IEEE Trans. Knowl. Data Eng. **29**, 1115–1128 (2017)
6. Hwang, T., Tian, Z., Kuangy, R., Kocher, J.P.: Learning on weighted hypergraphs to integrate protein interactions and gene expressions for cancer outcome prediction. In: International Conference on Data Mining (2008)
7. Winterbach, W., Mieghem, P.V., Reinders, M., Wang, H., de Ridder, D.: Topology of molecular interaction networks. BMC Syst. Biol. **7**(1), 90 (2013)
8. Munkres, J.R.: Topology. Prentice Hall Incorporated, Upper Saddle River (2000)
9. Hatcher, A.: Algebraic Topology. Cambridge University Press, Cambridge (2002)
10. Berge, C.: Hypergraphs: Combinatorics of Finite Sets. North Holland, Amsterdam (1989)
11. Chung, F.: The laplacian of a hypergraph. Expanding graphs (DIMACS series), pp. 21–36 (1993)
12. Cooper, J., Dutle, A.: Spectra of uniform hypergraphs. Linear Algebra Appl. **436**(9), 3268–3292 (2012)
13. Krivelevich, M., Sudakov, B.: Approximate coloring of uniform hypergraphs. J. Algorithms **49**(1), 2–12 (2003)
14. Rödl, V., Skokan, J.: Regularity lemma for k-uniform hypergraphs. Random Struct. Algorithms **25**(1), 1–42 (2004)
15. Alon, N.: Transversal numbers of uniform hypergraphs. Graphs Comb. **6**(1), 1–4 (1990)

16. Sarma, A.D., Molla, A.R., Pandurangan, G., Upfal, E.: Fast distributed pagerank computation. Theoret. Comput. Sci. **561**, 113–121 (2015)
17. Lu, L., Peng, X.: High-ordered random walks and generalized laplacians on hypergraphs. In: Frieze, A., Horn, P., Prałat, P. (eds.) WAW 2011. LNCS, vol. 6732, pp. 14–25. Springer, Heidelberg (2011). https://doi.org/10.1007/978-3-642-21286-4_2
18. Wang, J., Lee, T.T.: Paths and cycles of hypergraphs. Sci. China, Ser. A Math. **42**(1), 1–12 (1999)
19. Bermond, J.C., Heydemann, M.C., Sotteau, D.: Line graphs of hypergraphs I. Discret. Math. **18**(3), 235–241 (1977)
20. Pienta, R., Abello, J., Kahng, M., Chau, D.H.: Scalable graph exploration and visualization: sensemaking challenges and opportunities. In: 2015 International Conference on Big Data and Smart Computing (BigComp), pp. 271–278. IEEE (2015)
21. Chau, D.H., Kittur, A., Hong, J.I., Faloutsos, C.: Apolo: interactive large graph sensemaking by combining machine learning and visualization. In: Proceedings of the 17th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, pp. 739–742. ACM (2011)
22. Chau, D.H.P.: Data mining meets HCI: making sense of large graphs. Ph.D. thesis, Carnegie Mellon University (2012)
23. Van Ham, F., Perer, A.: "Search, show context, expand on demand": supporting large graph exploration with degree-of-interest. IEEE Trans. Vis. Comput. Graph. **15**(6), 953–960 (2009)
24. Herman, I., Melançon, G., Marshall, M.S.: Graph visualization and navigation in information visualization: a survey. IEEE Trans. Vis. Comput. Graph. **6**(1), 24–43 (2000)
25. Hoff, P.D., Raftery, A.E., Handcock, M.S.: Latent space approaches to social network analysis. J. Am. Stat. Assoc. **97**(460), 1090–1098 (2002)
26. Bader, B.W., Berry, M.W., Browne, M.: Discussion tracking in enron email using PARAFAC. In: Berry, M.W., Castellanos, M. (eds.) Survey of Text Mining II. Springer, London (2008). https://doi.org/10.1007/978-1-84800-046-9_8
27. Decherchi, S., Tacconi, S., Redi, J., Leoncini, A., Sangiacomo, F., Zunino, R.: Text clustering for digital forensics analysis. In: Herrero, Á., Gastaldo, P., Zunino, R., Corchado, E. (eds.) Computational Intelligence in Security for Information Systems, pp. 29–36. Springer, Heidelberg (2009). https://doi.org/10.1007/978-3-642-04091-7_4
28. Diesner, J., Carley, K.M.: Exploration of communication networks from the enron email corpus. In: Proceedings of Workshop on Link Analysis, Counterterrorism and Security, SIAM International Conference on Data Mining 2005, pp. 3–14 (2005)
29. Leskovec, J., Lang, K.J., Dasgupta, A., Mahoney, M.W.: Community structure in large networks: natural cluster sizes and the absence of large well-defined clusters. Internet Math. **6**(1), 20–123 (2010)
30. Chapanond, A., Krishnamoorthy, M.S., Yener, B.: Graph theoretic and spectral analysis of enron email data. Comput. Math. Organ. Theory **11**(3), 265–281 (2005)
31. Priebe, C.E., Conroy, J.M., Marchette, D.J., Park, Y.: Scan statistics on enron graphs. Comput. Math. Organ. Theory **11**(3), 229–247 (2005)
32. KONECT: Enron Network Dataset, April 2017. http://konect.uni-koblenz.de/networks/enron
33. Aksoy, S.G., Kolda, T.G., Pinar, A.: Measuring and modeling bipartite graphs with community structure. J. Complex Netw. **5**, 581–603 (2017)

34. Takens, F.: Detecting strange attractors in turbulence. In: Rand, D., Young, L.S. (eds.) Dynamical Systems and Turbulence. Springer, Heidelberg (1981). https://doi.org/10.1007/BFb0091924
35. Khasawneh, F.A., Munch, E.: Chatter detection in turning using persistent homology. Mech. Syst. Sig. Process. **70–71**, 527–541 (2016)
36. Bruillard, P., Nowak, K., Purvine, E.: Anomaly detection using persistent homology. In: Cybersecurity Symposium 2016. IEEE (2016)
37. Edelsbrunner, H., Harer, J.: Persistent homology-a survey. In: Surveys on Discrete and Computational Geometry: Twenty Years Later. AMS (2007)
38. Ghrist, R.: Barcodes: the persistent topology of data. Bull. AMS **45**(1), 61–75 (2008)
39. Cohen-Steiner, D., Edelsbrunner, H., Harer, J., Mileyko, Y.: Lipschitz functions have L p-stable persistence. Found. Comput. Math. **10**(2), 127–139 (2010)
40. Chazal, F., Cohen-Steiner, D., Glisse, M., Guibas, L.J., Oudot, S.Y.: Proximity of persistence modules and their diagrams. In: Proceedings of the Twenty-fifth Annual Symposium on Computational Geometry, SCG 2009, pp. 237–246. ACM, New York (2009)
41. Chazal, F., de Silva, V., Oudot, S.: Persistence stability for geometric complexes. Geom. Dedicata **173**(1), 193–214 (2014)
42. Chazal, F.: High-Dimensional Topological Data Analysis. CRC Press, Boca Raton (2017)
43. Singh, G., Mémoli, F., Carlsson, G.E.: Topological methods for the analysis of high dimensional data sets and 3D object recognition. In: SPBG, pp. 91–100 (2007)
44. Lum, P.Y., Singh, G., Lehman, A., Ishkanov, T., Vejdemo-Johansson, M., Alagappan, M., Carlsson, J., Carlsson, G.: Extracting insights from the shape of complex data using topology. Sci. Rep. **3**, srep01236 (2013)
45. Munch, E.: A user's guide to topological data analysis. J. Learn. Anal. **4**, 47–61 (2017)
46. Korolov, M., Myers, L.: What is the cyber kill chain? Why it's not always the right approach to cyber attacks. CSO Online, November 2017
47. Padhy, N.P.: Unit commitment-a bibliographical survey. IEEE Trans. Power Syst. **19**(2), 1196–1205 (2004)
48. Price, J.E., Goodin, J.: Reduced network modeling of WECC as a market design prototype. In: Power and Energy Society General Meeting, pp. 1–6. IEEE (2011)
49. Yu, N.P., Liu, C.C., Price, J.: Evaluation of market rules using a multi-agent system method. IEEE Trans. Power Syst. **25**(1), 470–479 (2010)
50. Jung, J., Liu, C.C., Tanimoto, S., Vital, V.: Adaptation in load sheddding under vulnerable operating conditions. IEEE Trans. Power Syst. **17**(4), 1199–1205 (2002)
51. Munoz, F.D., Hobbs, B.F., Ho, J.L., Kasina, S.: An engineering-economic approach to transmission planning under market and regulatory uncertainties: WECC case study. IEEE Trans. Power Syst. **29**(1), 307–317 (2014)
52. Robinson, M.: Sheaves are the canonical datastructure for information integration. Inf. Fusion **36**, 208–224 (2017)
53. Joslyn, C.A., Hogan, E.A., Robinson, M.: Towards a topological framework for integrating semantic information sources. In: Semantic Technology for Intelligence, Defense and Security (2014)
54. Dowker, C.: Homology groups of relations. Ann. Math. **56**, 84–95 (1952)
55. Robinson, M.: Sheaf and duality methods for analyzing multi-model systems. In: Pesenson, I., Gia, Q.L., Mayeli, A., Mhaskar, H., Zhou, D.X. (eds.) Novel Methods in Harmonic Analysis. Birkhäuser (2017, in press)