Journal of Big Data

METHODOLOGY Open Access

Adaptive network diagram constructions for representing big data event streams on monitoring dashboards

Alexander V. Mantzaris[1*], Thomas G. Walker[2], Cameron E. Taylor[1] and Dustin Ehling[1]

*Correspondence:
alexander.mantzaris@ucf.edu
[1] University of Central Florida, 4000 Central Florida Blvd, Orlando, FL 32816, USA
Full list of author information is available at the end of the article

Abstract

Critical systems that produce big data streams can require human operators to monitor these event streams for changes of interest. Automated systems which oversee many tasks can still have a need for the 'human-in-the-loop' operator to evaluate whether an intervention is required due to a lack of suitable training data initially offered to the system which would allow a correct course of actions to be taken. In order for an operator to be capable of reacting to real-time events, the visual depiction of the event data must be in a form which captures essential associations and is readily understood by visual inspection. A similar requirement can be found during inspections on activity protocols in a large organization where a code of correct conduct is prescribed and there is a need to oversee whether the activity traces match the expectations, with minimal delay. The methodology presented here addresses these concerns by providing an adaptive window sizing measurement for subsetting the data, and subsequently produces a set of network diagrams based upon event label co-occurrence networks. With an intuitive method of network construction the amount of time required for operators to learn how to monitor complex event streams of big datasets can be reduced.

Keywords: Event streams, Big data, Networks, Graph visualization, Co-occurrence networks, Visual analytics, Dash boards

Introduction

With the growing amount of information being gathered and stored the associations between the fields becomes more ambiguous, and challenging to track, as the context of queries from the data can utilize different subsets depending upon the application. Dimensionality reduction techniques upon big datasets [1, 2] can provide techniques to reduce redundancy and therefore the perceived complexity of the data, but this does not take into account some very important aspects of the use cases of these systems. Many of the queries (questions) users have of the datasets are not based upon information theoretic principles and are context sensitive. Removing features of the dataset (obscuring from view) can pose potential hazards in situations where outlier variables bring to the attention information which is not represented in the data but confined within the understanding of the user. There needs to be a visualization method in which a user can monitor features of the dataset association with minimal strain to reduce errors induced

Springer Open

from fatigue (*cognitive strain* discussed in the context of medical overview of patients [3]).

An important application which addresses the current increase (almost monotonic) in urbanization and growth of 'city' sizes [4], is big data for smart urbanism. Described in [5] is the necessity to incorporate 'real-time' constraints not only for the load processing of gathered data but also for the temporal relevance of the inferences to respond to arising situations such as life threatening emergencies [6]. Emergencies may arise from disruptions in services and from a wide range of failures in the system that can be generally termed as 'Critical Events' [7], which takes the view of the system as an entity that requires support across a broad set of facilities with a need for real time responses. A key factor emphasized is the 'Contextual Filtering' which is key to the 'Decision Support' where the data is a stream, and the work presented here refers to such data as 'event logs'. The main difference in the terminology is that it is assumed that the storage of the event logs has correct temporally aligned observation tuples of the information. This challenge has been highlighted for more than 20 years as being part of the general task of multi-sensor data fusion [8] (notes applications for defense as well). The size of these streams is a primary concern not only due to the IO latencies that may be a concern but also that the user input requirements that may be a requirement to provide a relevant query. This is explored in an architecture in [9], which looks at ontology mapping and explicitly puts 'usability' as a primary design functionality. With a large search space it is not enough to provide the feasibility for the correct data and insight to be derived, it must account for the understanding that there can be too many features to explore under constrained time limits. With this in mind, the big data streams are handled in the proposed methodology in such a way as to allow users to monitor event label data streams within a *network diagram* production. An overview of this challenge of reasoning from streaming data is provided in [10], and in [11] the challenges for the huge volumes are described primarily in the effort to understand city dynamics which demands real time processing.

The scope of this proposed methodology is to allow big streams of data to be presented concisely into a dashboard where a human operator/viewer can monitor the patterns and underlying changes in the data [12]. Producing co-occurrence networks based upon the observation event label counts (co-occurrence counts [13]) within a similar time-span or same event sequence, the information store produces a summary representation which is displayed as a network diagram. A network based visual guide to similar datasets of concern is covered in [14], and refers to the data as tabular data with meta-data. The meta-data is necessary to annotate the columns of the observations which are considered to be *variables* (as will be discussed in more detail). There is an important ambiguity to note when referring to different formats of tabular data, that can be used for network constructions and that is when the data can also be interpreted as adjacency matrices which have more than a single dimension of dependence for looking up data points. Event stores such CSV (comma separated values) are used where a single row is representative of an 'event' of unilateral observations. Although CSV data is referred to here as a specific data format, the methodological design can apply for a wider range of columnar data formats which can store sequential events.

As described in [14], the data in these scenarios, can be viewed as *multivariate datasets* where there is a set of possible *attribute relationships* (between columns). The meta-data is considered to be the header of the CSV which could be any associated text label that does not necessarily need to be included in the same file. The tool developed in [14] permits a user experience interacting with the data with a view of the cellular structure, to not deviate too far from the interaction experienced from mainstream spreadsheet applications as columns and rows can subsetted for a network visualization. The emphasis of creating a representation with intuitive foundations, such as a spreadsheet, has a motivation supported by the concept of the 'human-in-the-loop' (HITL) [15, 16] where the data science pipeline integrates human input in the cycle of operations generating decision making processes. A problem with methods such as this for big data is that the large scale changes and associations will be challenging for a human user to construct from the cellular information. An example of where the HITL can be considered essential is in the effort towards extracting ontological concepts from text, [17], and a methodology can be designed in order to efficiently pool from the human understanding where the information relating to the textual concepts is not sufficiently contained in the data. The human factor can be used to look at the tuning and error corrections upon evolving data which may not align with the initial training data. Another application which aims to provide a similar interface into the data via network diagrams is [18] which has a broad set of tools provided to the user based upon well understood network science principles. From such an application the user can focus upon different clusters of associations and an intuitive spatial navigation requires minimal training while providing consistent measurement interpretations. A key challenge to an effective HITL strategy is to avoid relying upon the human actor to look at individual data points which would induce fatigue and introduce a bottleneck which is especially important for big data and applications for critical systems.

The use of metadata in network science is becoming more common as the computing power has grown sufficiently to analyze, visualize and construct large (to huge) sized networks with more associated information. One of the key computational barriers in using metadata in network science is that some (possibly many) variable observations may be the common/shared between many rows (events) while other features can vary even uniquely so that a large number of nodes are produced with a dense edge set which increases the run time for many analytical methods. The work of [19] examines *annotated networks* as connections between individuals in a social network where the metadata (associated data) is used to clarify the community separations. From the paper, figures. 3, 5 and 6 show how different communities which share connections can have a more distinct separation by utilizing more variables in the inference procedure. The use cases of the work proposed here though look beyond that of social networks where there is a clear variable reference point; that of the human identifier in the event trace. In the application to a wider analysis of event datasets, this singular variable of social connectivity being the only focus is not assumed.

There are many different terms used to reference the nature of the data and networks produced from the tabular data which contains associated metadata descriptors or attributes. As mentioned in [19] there is the term *annotated networks* used, and as well in [20] the term *attributed graphs* is applied to analogous constructions. Outside of social systems where the main edge may not be considered to be some type of human to human exchange, there are *heterogeneous networks* [21] which represents the existence of multiple types of nodes and edges (links). In the case where recommendations are being made based upon social connectivity of common interactions for a predicted interaction, [22] uses *heterogeneous information network* and [23] uses the term *multivariate networks*. The work in [23] follows the essential goal of this work which is to target 'non-expert users' by providing a 'novel solution for multivariate network exploration and analysis'. Its approach to handling a wide variety of heterogeneous nodes and edges is by modelling a set of tuples as '(relation, link, edge)' which, although not referenced in that work is akin to the RDF modelling paradigm [24]. Within that framework aggregates upon these relationships can be produced in order to reduce the number of vertices to move from 'details' to 'overview' in a series of stages. For the examples provided in that publication this is possible but for an application where the attributes can vary without constraint, issues can arise from this method and similar ones. The underlying issue not addressed is that with unconstrained datasets, where data fusion should be unrestricted, the relationships between edges themselves can become complex entities which may not have direct correspondances to permit aggregations or even compatible paths. Integrating or combining heterogeneous data for specific applications in terms of storage and efficient extraction is covered in [25] (graphical database application), and association relevances in [26] which permits the assessment of a 'path' between nodes. The relevance computation is particularly important and often overlooked in many applications, but what is a more essential question that arises immediately; is whether connectivity through overlapping attributes (variable observations) provides 'reachability' [27].

The approach provided here outlines a methodology to provide a set of network diagrams (topologies) inferred from count co-occurrence networks based upon the information in event streams. These datasets are archetypically considered to be CSV files, but the difference is that it is assumed that these files are dynamic in the situations where there is a monitoring dashboard [28]. There are various aspects of the requirements upon the data assumed in previous studies and use cases requirement are relaxed in this development in order to broaden the scope of the applications. One such aspect is that the data is not required to pertain to a specific application such as that presented in [28] where it is directed specifically towards an urban landscape. Another issue commonly encountered in data exploration is the technical domain knowledge barrier of entry and incompatibility with programs specified for particular use cases. Handling the relevance of the temporal windowing of a dataset subsetting is also a challenge which users are prompted to chooses the number of events to be displayed. This can have a large effect upon the interpretation and the amount of information that should be encapsulated, so the concept of a necessary minimum is adopted here.. The network diagram representation of different metrics in topological space can intuitively convey information to a user with minimum effort to understand associations therefore by-passing the need of training for a toolkit. The approach is designed to evade the necessity of the data sets to be associated with spatial coordinates [29] to present associations.

The methodology described in "Methodology" section outlines the paradigm for:

1. choosing the window size for the number of most recent events that has the lowest rank on measures of the cluster entropy variations across the recorded event sequence;
2. generates a co-occurrence network based upon variable observations which displayed common attribute labels (other common associated variable labels) in a weighted network diagram;
3. generates a co-occurrence network for variables between themselves and with different variable outcomes (inter and intra variable label co-occurrences).

Visualizations of this information allow a user to assess the changes in the cluster formations of the event variables. The network diagrams will allow a user to conduct a quick visual scan for variable label associations between different labels in the same column and between different columns. Updates can be redrawn for monitoring purposes and the window selection can allow the trace to be computed without requiring the full dataset to be visualized which is not feasible in general for big data. The use of network diagrams effectively produce data summaries for human actor inspection that then need be able to decide whether actions are necessary. This approach to the analysis, when the data is beyond human ability to inspect, is discussed generally in [30] and outlines how this is a recent technique. Although the application to social media may have interesting applications due to a wider spread single use case model than what is taken here the necessity to monitor event logs is in need of further development. One of the assumptions made here is a relaxation of previous assumptions, that the data is both dynamic and potentially heterogeneous in that the underlying edges may not converge to a single distribution with more observations. As demonstrated in [31] the navigation of variable relationships in a network diagram can become a complex process for the user as the complexity of the interconnectivity of the variables also increases. It is a requirement to avoid allowing too much information to be presented as tha can result in producing a visualization which the user cannot extract meaningful insight. The methodology put forward aims to allow network diagrams to be produced from the data streams which do not induce a large cognitive load on the human actor and convey an overview of the event occurence data being observed.

Methodology

The event list can generally be considered to be an event stack, which is the ordered list of a event entries recorded in order. The event stack is chosen as a reference since it emphasizes the importance of the unrestricted growth of the dataset based upon a specific manner in which the dataset is produced. The size of the dataset may be changed as costs of storage may affect the decision for maintaining early event log entries. The methodology first proceeds by discretizing the tabular event sequence data, which is an approach taken in [32] providing a full set of analytic tools to monitor general tabular data by *Binning* continuous variable values. Observations which are continuous and subsequently put into 'bins' are labelled as the string label for their domain membership. This does incur a loss of information but for the task of low delay observations of a monitoring service it can suffice or convey the information needed to direct a subsequent specific examination such as a database query to inspect the exact values for various entries. From the columns which are considered as variables here, each label in the rows delivers a string label which is similar in concept to the terminology used with 'factors' and 'level' values with the main difference in that this work does not attempt to model the distribution of the levels and therefore uses the term 'variable labels' to indicate the set of string representations for the observations in the rows of a particular columns. From these variable labels observed in the event logs a co-occurrence network can then be produced between these discrete observations that become edges in a network diagram.

"Adaptive window size determination" section presents the method in which the window size is determined, and "Variable label association networks" section the approach to which the network topology is constructed (graph vertices and edges). All of the code was written with Julia Lang (v1.01) [33], and the graph visualization production (creating images for the network diagrams) is performed with GraphViz [34].

Adaptive window size determination

A window size is chosen to subset the dataset so that the last event rows are used to produce a network diagram. The objective is to use a window size in which if used consecutively from the start of the event list a clustering of the events will require the lowest average (uniform mean) entropy, the lowest variation in the entropy values of the cluster assignments over the dataset, and the smallest sampled average variation of information of the row/event cluster assignments between different window placements in the dataset.

Having discretized the dataset, the Hamming distance is then used as a metric for the distance between each event entry. The dataset is denoted by D where the dimensions are (N, M), and the Hamming distance between each pair of events is (for a row pair (i_1, i_2)):

$$d_H(i_1, i_2) = Hamming(D(i_1, k), D(i_2, k) : \forall m \in [1, M]). \tag{1}$$

Here m will be used to denote the variable reference which corresponds to a column number in the original dataset. A list of optional window lengths, W, is provided by the user to hold event stacks of the last $w \in \mathrm{W}$ events and possibly subset the column variables as well;

$$D_w \leftarrow D(i, m : \forall i \in [N - w, N], \forall m \in [1, M]). \tag{2}$$

In the applications here the choices of the window sizes are $\mathrm{W} = [30, 45, 60, 100]$. Depending upon the variation in the event variable label distributions between temporally local events, the choice of the maximum window size provides a constraint upon the complexity of the network diagrams drawn. Since the networks are weighted the number of events may not correspond directly with the anticipated increase in unique node or edge numbers and therefore this option is provided to the user. Such a scale for the upper limit on the number of events is expected to be readily understood by the user.

Besides the computational requirements of producing a complex network diagram, the even size of the data structures can inflate quickly depending upon the network diagram drawing package used. Produced without checking for the size considerations and an excessively large window size could create bottlenecks in the memory allocation. Another consideration taken here using GraphViz, is that the .png file format is chosen and the size of the file can grow to the point where the delay in the view is noticeable. The output quality is not considered as an option to reduce since users should be able to zoom in on large dense networks without loss of resolution.

The matrix of pairwise distances between events for the window is constructed using the Hamming distances for each of the pairs as a cartesian product between all of the event vectors of the window. This produces a $w \times w$ matrix:

$$A_{H_w}(i,j) := d_H(i_1, i_2 : i_1 = N - w + i, i_2 = N - w + j), \tag{3}$$

of distances with there being a designated zero along the diagonal. A clustering of the events is produced based upon this distance matrix (A_{H_w}) and is done using Affinity Propagation [35] (with the function call denoted as AP). Affinity Propagation minimizes a cost/utility function and a convention of altering the pairwise distances is followed. The distances are negated so that a maximization is the compatible with the Hamming distances, $A_{AP}(i,j) = (-1) \times A_{H_w}(i,j)$ and that the diagonal set to the median distance $A_{AP}(i,i) = median(A_{AP}) \forall i \in [1, N_w]$. This is done so that a call to a function implementing AP can take A_{AP} as the input. The cluster result, with the end of the time window indicated, is denoted as CR_w and this contains the event row cluster membership.

Over the full set of events the time window is shifted, at uniform increments, in order to compare the changes in the cluster results for different window sizes. This interval shift is denoted as δ, so the set of cluster results for a particular window size $w \in \mathrm{W}$ is; $\mathrm{CR}_w = [CR_w, CR_{w+r\delta}, \ldots, CR_{w+r_{max}\delta}]$ where $\mathrm{r} = [0, \ldots, r_{max}]$ and $r_{max} \times \delta = N$ (assuming δ is chosen in order to divide N without a remainder). Two measures are computed based upon the set of CP_w; the entropy of the cluster assignment distribution for each event subset and the Variation of information [36] (also known as shared information distance). The Variation of Information (VoI) method compares the cluster assignments, and from using mutual information provides a measure of distance between them. For each window size the mean of the entropy values, the standard deviation of the entropy values and the mean VoI between a cluster at a window center and another randomly chosen window clustering are computed. The rank for each measure across the different window sizes is found and the window size with the smallest aggregate rank is chosen. This average lowest rank chosen window size is referred to as $\hat{w}$:

$$\begin{aligned} \hat{w} := min_w(\{&rank(\bar{H}(\mathrm{CR}_{w+r\delta} : \forall r \in \mathrm{r}) : \forall w \in \mathrm{W}), \\ &rank(std(H(\mathrm{CR}_{w+r\delta} : \forall r \in \mathrm{r}) : \forall w \in \mathrm{W})), \\ &rank(VoI(\mathrm{CR}_{w+r\delta} : \forall r \in \mathrm{r}) : \forall w \in \mathrm{W})\}). \end{aligned} \tag{4}$$

The time windows can be chosen not based upon the trace of entropy of the cluster formations, but based upon the time points that separate the different events so that the temporal relevance is context sensitive. Situations where the window size can be chosen explicitly are is not considered here where the scope is to ‘mine’ (explore) datasets in which the user has limited domain knowledge of the data. It is not a necessary assumption that there is a meaningful ordering of the rows, but the clustering segmentation assumes a loose correspondence between the row indices and the window center position. This minimization is essential in order to reduce the number of unique event labels used to produce network diagrams as outlined in the following “Variable label association networks” section. As the event streams progress the big datasets can produce an increasing number of variable labels that will become challenging to interpret by an observer without such a window size applied to subset the data.

Variable label association networks

In this methodology the values of the column variables which are discrete strings, are referred to here as variable labels. They can be considered as a paired tuple to uniquely identify the observations in each event and define the event space as the Cartesian product of all variable labels. Since this methodology aims to present a subsetted analysis of the data to the user in need of monitoring an event stack; the possible edges are limited to the variable labels produced in $D_{\hat{w}}$. This set of tuples forms the basis of the nodes (vertices) which can produce interconnectivity through co-occurrences, per event, and permits a network/graph construction $G = (V, E)$:

$$V := \left\{\left\{D_{\hat{w},m_1}\right\}, \ldots, \left\{D_{\hat{w},m_{max}}\right\}\right\}. \tag{5}$$

This permits the construction of a weighted adjacency matrix A_V, with $\|V\|^2$ elements where each element is the co-occurrence count between variable labels:

$$A_V(i,j) := \left\|\left\{D_{\hat{w}}\left(*, m : V_i = D_{\hat{w}}(*, m)\right) = D_{\hat{w}}\left(*, m' : V_j = D_{\hat{w}}(*, m')\right)\right\}\right\| \tag{6}$$

where $m \neq m'$. It is important to note is that the block diagonals will be of value zero, square and have lengths $\|\{D_{\hat{w},m}\}\| \forall m \in M$ as each variable label cannot co-occur with that of another column. It can be thought of as each label is a concatenated string pair of the column name (constrained unique header list) and the label entry. The edges of the co-occurrence network are the number of non-zero values of the variable label co-occurrence:

$$E := \left\{\left(V_i, V_j\right) : \forall_{i,j} A_V(i,j) \neq 0 \wedge i \neq j\right\}. \tag{7}$$

Given the vertices and edge sets, this can define the connectivity at the base data representation and an implementation could rely upon the use of dictionaries or linked lists to store this information. By incorporating the variable labels into the construction of the adjacency matrix A_V where the sides of this matrix are not equal to the number of vertices but the number of unique variable labels, the size of the matrix is increased but avoids the necessity of utilizing external datastructures or files for attributes (outlined in [37]). Although the effective operations can be fulfilled with previous approaches of a separate store for the attributes of node definitions (associated attributes), the need for external file references introduces a burden of maintaining the file links in the management process. A matrix store can be more inefficient for big networks which cannot fit into memory and need to be paged (in certain implementations), but considering that a windowed truncation of the event data is applied the size is being regulated. Networks too big to fit into memory might not be easily understood by an operators for those visualizations to be of actual practical use to the viewer monitoring changes. Although the whole event store is used to estimate a suitable window size, this then results in a subset of a reduced number on the upper bound of vertices $\|V\|$ and edge number $\|V\|^2 - \|V\|$. Considering each implementation as equal, in the application direction chosen in this work there is a feature to which a matrix store produces both a practical solution to file management, and increase in the data interpretation for constructing walks based upon the edge set.

Using $A_V(i,j)$ the associations between variable label co-occurrences can be succinctly represented using the theory underlying *Katz Centrality* [38, 39], which uses matrix exponentiation to derive the number of *walks* between nodes (which are variable labels in this case). The first network construction produced is based upon the walks of length 2 (demonstrated in Figs. 1b, 2b, 3b):

$$A_2 = A_V^2. \tag{8}$$

This produces a connectivity between variable labels of the same column who share an event co-occurrence with a common variable label other than itself. A weighted

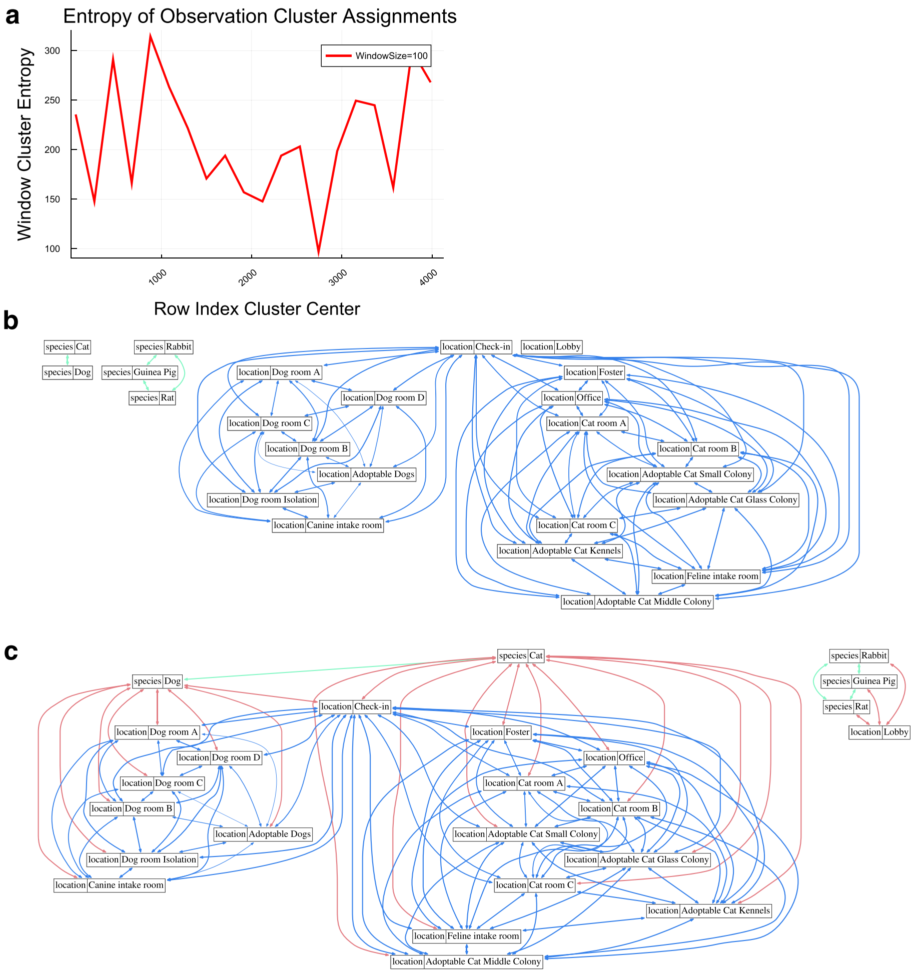

Fig. 1 The results of the methodology applied to data of a pet shelter; 2017 Bloomington Animal Shelter (from *data.gov*). Two variables (columns) are chosen from the data; the 'species' and 'location'. **a** Shows the cluster entropy value changes along the event data for subsetting upon a window length shown in the legend. **b** Shows the number of walks of length 2 between label values within a column (the triangulation edges). This produces a separate network diagram for each variable where edges show that two labels have been in an event where another variable(s) was recorded with the same label, and the width is proportional to those counts. **c** Shows the network with edges of walk length 2 (triangulation) and walk 1 which is the even variable label co-occurrence network

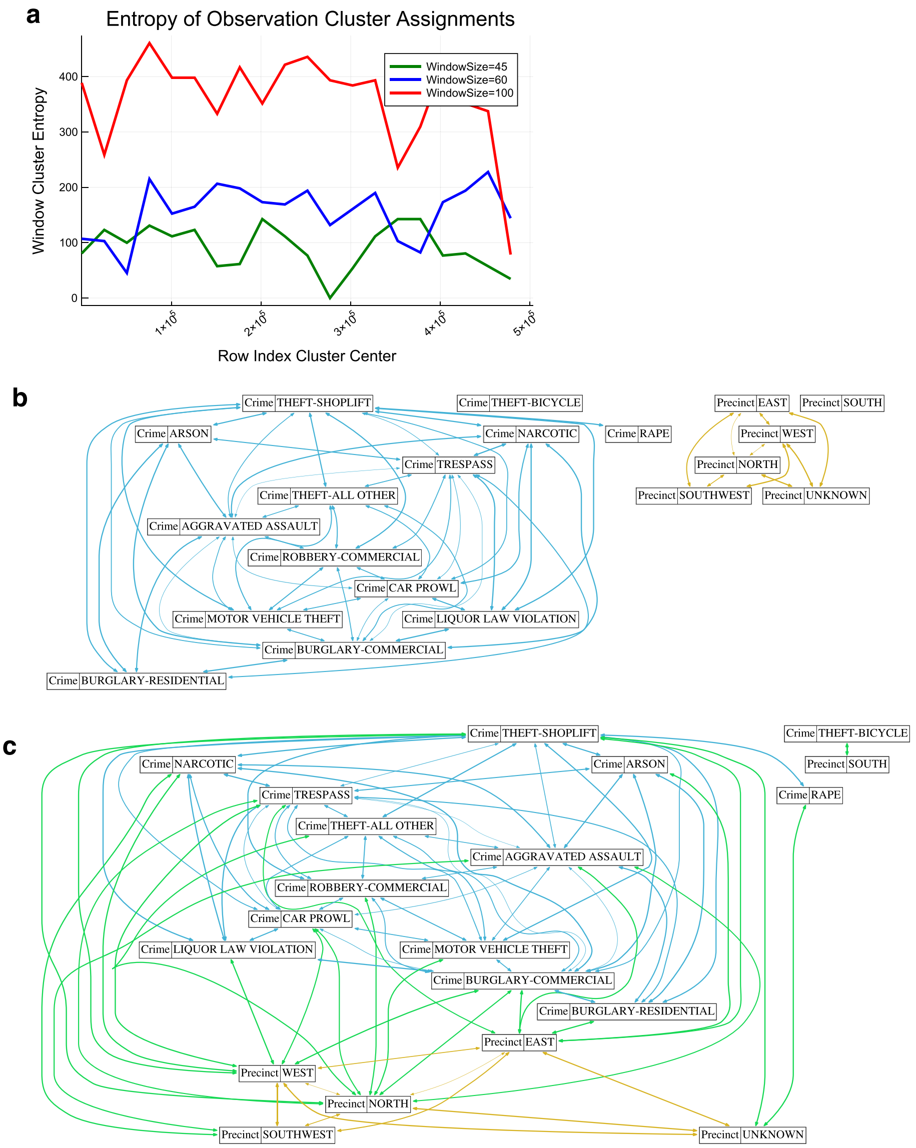

Fig. 2 The methodology used with data from the Seattle Police Department crime event data (taken from data.gov) where the type of 'Crime' and the 'Precinct' are used. **a** Shows the cluster entropy trace for the window size set. **b** Shows the walks of length 2 between labels (triangulation) of the variable which is effectively the sharing of label values in events. For this example this corresponds to crime types having been reported in the same precinct (left/blue edges) and for different precincts to have an edge they have at least 1 common crime reported. **c** Has 3 different edge types for the walks of length 2 between variable labels and the length 1 of the number of label co-occurrence counts between different variables

adjacency matrix is produced where $A_V(i,j) \circ A_2(i,j) = 0\forall_{i,j}$. The only non-zero entries in A_2 are those between labels of the same variable (column) in the dataset since edges between labels of the same variable do not exist in the events which are walks of length 1, the walks of only length 2 will therefore connect variable labels which share an event in which they are both present (co-occur). The other network diagram edge construction measure used is (demonstrated in Figs. 1c, 2c, 3c:

$$A_{2,1} = A_V^2 + A_V^1. \tag{9}$$

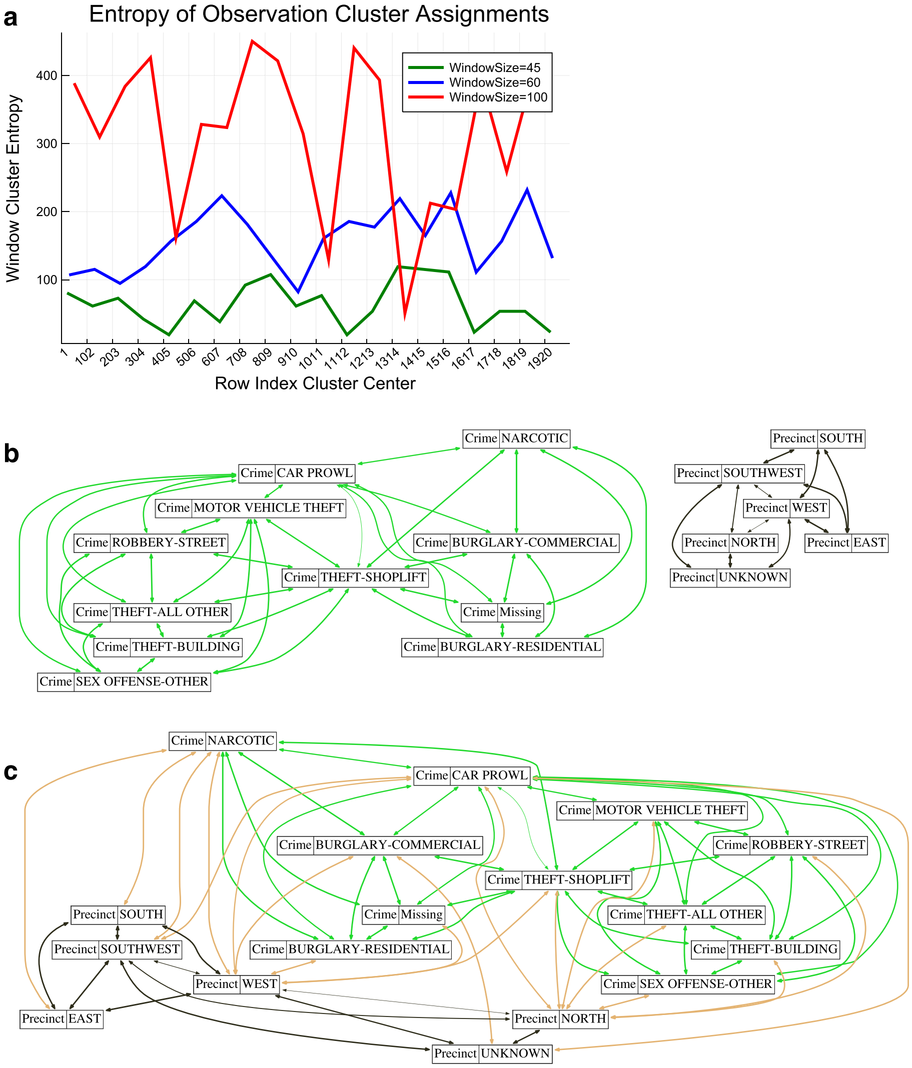

Fig. 3 The first 2018 rows of the Seattle Police Department data are used (taken from *data.gov*), and the results can be compared with those of Fig. 2 where the full set of events are used. Subfigure **a** shows the cluster entropy trace for the window size set. There is greater variability in the trace of the cluster entropy variance estimation for the larger window size of 100 which shows that more data is required to ensure similarities in the subsetting positions and that smaller window sizes have more cross comparisons to reduce the variability. In subfigures **b** and **c** it can be seen that crimes 'ARSON', 'THEFT-BICYCLE' and 'RAPE' are not present in the subset data of these events. It can also be seen how the crime 'NARCOTIC' is recorded in 'Precinct WEST' where in the previous examination it was not

This produces the networks based upon the edge set for length 2 walks as well as the edges present in the original event variable label co-occurrences. In these network diagrams produced, the differentiation of different walk lengths edges are brought to the attention of the viewer by a unique color attribute. These matrices can be viewed directly as a heatmap of the grid variable label associations or as a network diagram. As will be shown in "Results and discussion" section, the heatmap value is primarily useful for coarse large scale assessments of the variable label block structure, and that the network diagrams assist the viewer in monitoring events of practical contextual value by tracing the edges that connect variable labels based upon walks of length 1 or 2. It should be noted that $A_{2,1}$ can be computed in $O(n^{2.373})$ as noted in [40, 41] where this can be regulated according the constraints of the system and included as a maximum window length based on the unique element size for the segment of the event sequences, $\|V\|$. The minimization the window length on the number of events considered for each network construction, Eq. 4, as a requirement to reduce the value of $\|V\|$ is the number of unique labels used in $A_{2,1}$. Without this effort to enforce a minimization the vast number of events in big data streams would produce networks with too much information to be interpretable by humans in the monitoring process.

Data choice

This methodology is demonstrated in "Results and discussion" section, where the application is made with two different datasets of events in complex organizations susceptible to sporadic events that are logged. These organizations are the 'Bloomington Animal Shelter' and 'Seattle Police Department' which can be found readily from http://www.data.gov. The reason that these datasets are chosen is that as an organization susceptible to random events from external factors there is a need to monitor the overall system variable associations over time with low latency. This requires the visualization to be readily interpretable on a monitoring dashboard so that interventions can be made as fast as possible. The event logs can continue to grow to become vastly large and require a choice of temporal relevance as proposed in Eq. 4, and the framework for the Eqs. 5, 6, 7, 9 provide the computations for the network construction.

Results and discussion

In each of the 3 figures presented from 2 different real datasets taken from the accessible to the public data source, http://www.data.gov, 3 subfigures are presented. The first is a display of the entropy variance of the clusters identified from the raw event data for that segment of the data defined by the window length. This aims to show the stability of the cluster sizes for that segmented view. The second subfigure will show the intra variable connectivity between variable labels (labels of the same column and not between different columns). Therefore if there are 2 variables/columns 2 disjoint networks will be present in the same view. The edge widths shown are the values of the number of walks of length 2 based upon the aggregate of the co-occurrence network aggregate from the event rows described in "Methodology" section. This is effectively the 'triangulation' edge ([39] produced by 2 labels of a column occurring with a common 'attribute'). The interpretation upon examining each present edge in these intra connectivity diagrams is; the width of the edge is proportional to the number of events where these two variable labels were present with 1 or more other common variable labels. There will be 2 different colored edges, 1 color for the connectivity between labels of one variable (columns) and another set of colors for each of the other variables between themselves. Subfigure c displays the network diagram of walks of length 1 and 2; the aggregate event count co-occurrences in each row and the number of triangulations between labels within the same variable/column. Therefore, variables which have edges between its labels will also contain an edge which leads to 1 or more of a different variable label which supported the triangulation. A loss of information is created in that multiple triangulations will not allow the viewer to determine which of the variables two labels of the same column were co-dependent upon to connect.

The network diagram nodes contain the variable name and the label string. These 2 strings are separated with a vertical bar and the variable name can be found in the meta data for a dataset or typically as the first label for a particular column. Random colors are put to ensure that different network reproductions (even for the same data) highlight different connectivities between the labels.

Bloomington Animal Shelter

Figure 1 presents the results from applying the proposed methodology to the publically available dataset of https://catalog.data.gov/dataset/animal-care-and-control-adopted-animals-7ed9e. This is available from *data.gov* (in a csv) and is titled "Animal Care and Control Adopted Animals", Updated May 2018. This data is recorded events from an animal shelter in 2017, at the 'Bloomington Animal Shelter'. The dataset submission comments from the shelter reflect a scenario typically encountered in one form or another when attempting to analyze a corpus of data when changes to staff and management is brought about: "In early 2017, the Bloomington Animal Shelter migrated management software from AnimalShelterNet to Shelter Manager. We attempted to preserve as much information as possible from the old system." The challenges with data representations and the understanding of the associations, structure, and relationships can easily be lost in such a scenario. Changes such as these commonly bring about incongruities which are difficult to mend afterwards and with this proposed approach a single execution statement produces the results displayed in the 3 subfigures where the columns chosen are 'species' and 'location'.

Subfigure a shows that there are changes in the distribution of the clusters of the events over the time window of 100 rows (events). Smaller values for the window produced results which conformed to a valid cluster set that could be displayed and are ignored as the window size is grown till a suitable window size can be found. Subfigure b shows the intra variable label triangulation network built from the walks of length 2, which is the co-occurrence matrix raised to the power of 2. In this context, taking the example of the connection between 'species Cat' and 'species Dog' that the edge between them means that their events were recorded with a common 'location' variable label. Seeing how the 'species Dog' and 'species Cat' do not connect with the labels for 'species' of 'Rabbit', 'Guinea Pig' or 'Rat' means that the 'locations' of these groups never overlap. The event log, for the subset window, does not contain an instance where the species of dog or cat is the same for any other species listed. For the 'location' variable it can be seen how the 'location Check-in' variable label connects to most of the other nodes directly, except 'location Lobby' which has no edges. From the lobby it becomes apparent that the dogs and cats are treated separately. Subfigure c shows that the group of species 'Rabbit', 'Guinea Pig' and 'Rat' are located only in the 'Lobby'. The previous conclusion that species 'Cat' and 'Dog' overlap in 'location' only for the 'Check-in' is confirmed, and from each cluster a co-occurrence event is seen with each location.

Appendix shows an example of this network presented in a matrix of co-occurrence counts for the variables of the event stream with a heatmap. This is provided to show the challenges faced with the matrix view (grid view), namely the compression of the data values to accomodate for the explicit display of zero values. For sparse matrices this can cause the valuable information for the viewer to look at small cells and narrowly spaced gridlines. This compressed view for the dataset allows the data to grow continuously as the process generates event observations without reducing the clarity of the monitoring procedure. The consistency of this clarity can be seen as well in the following examples.

Seattle Police Department event data

The dataset used in here is collected from crime event data stored in a csv file obtained from the https://www.data.gov data explorer tool or directly at https://catalog.data.gov/dataset/crime-data-76bd0. The events arise from crime reported by the Seattle Police Department (SPD). Each row contains the record of a unique event where at least one criminal offense was reported by a member of the community or detected by an officer. Figure 2 shows the results of the analysis with the methodology. Subfigure a is the cluster entropy trace for the various row subsetting windows explored. The window size of 45 rows is chosen as it has the lowest average rank across the trace features measured.

The use of the methodology with data from the Seattle Police Department crime event data (taken from data.gov) where the type of 'Crime' and the 'Precinct' are used. Subfigure a shows the cluster entropy trace for the window size set. Larger cluster numbers and membership differences arise when 100 rows are taken into account. Subfigure b shows the intra variable connectivity between variable labels of walk length 2 from the co-occurrence network. Edges between crime variable labels mean that these crimes have occurred in the same precinct within the event rows considered, and precincts with an edge between then have overlapping crimes in the events; eg. 'Crime THEFT-BICYCLE' occurs in a precinct with no other type of crime. It can also be seen that 'Crime ARSON' does not occur in the same precinct as 'Crime CAR PROWL'. From the areas, 'precinct SOUTH' does not share the crime types of other areas. Subfigure c shows 3 types of edges, the walks of length 2 between labels of the same variable, and the edges of walk length 1 which are the number of counts for between variable label co-occurrences (each edge is drawn in a different color). It can be traced that 'BICYLE-THEFT' only occurs in the 'SOUTH' precinct, and the 'Crime' variable labels of 'THEFT-SHOPLIFT' and 'RAPE' are the only ones where the 'Precinct' is labelled as 'UNKOWN'. Another possibly valuable insight is that 'Crime NARCOTIC' does not co-occur with entries for 'Precinct EAST'.

Subsetted Seattle Police Department data

The analysis done here uses a truncated version of the data utilized in "Seattle Police Department event data" section where the first 2018 rows are taken. This is to compare the results with that of the previous investigation to examine a situation where the methodology was used in a previous time point to compare the interpretation of the results and underlying data's consistency (the subfigures a–c) correspond respectfully to that of Figure 2). In subfigure a it can be seen that with fewer subset comparisons that can be made the differences in the cluster formations are greater than when the full dataset is provided. This implies that the variance values decreases due to there being underlying stochastic changes in the event distributions which repeat in some periodic fashion.

Although more metrics could be provided to look at this feature in greater depth, the values can be wrongly interpreted by viewers and necessitate training which not only can delay uses, but incur costs for the time spent. This is especially attenuated with the larger window size of 100, to show that smaller window sizes have the similarities re-occurring with less data. Even from a layman's perspective the reduction in the variability is a positive aspect and can make decisions based upon the results which are in line with the use case.

From subfigures b and c it can be seen how the precinct 'SOUTH' has crime events which do overlap with other precincts, and that 'Crime RAPE' is not recorded in the first 2018 ordered events, 'Crime THEFT-BICYCLE' or is 'Crime ARSON'. A consistency is that the 'Crime NARCOTIC' is not recorded in the precinct 'NORTH'. The entry 'Crime CAR PROWL' is recorded in the 'Precinct NORTH' in both cases. The introduction of new variable labels for a variable (column) can be observed more carefully than with the matrix view or an ordered list, such as 'Crime MISSING' in the 'Precinct WEST' which is a new introduction which may be noteworthy for monitoring services. There are other interesting features to be noticed and the monitoring services can visually scan for the necessary features of concern if there are specific ones to compare.

Conclusion

This article has presented a new methodological approach to analyzing event data. These are considered to be stored as a CSV or in a form where the events are sequentially aggregated in a 'stack' like manner with named columns that are variables of observations collected from some process monitored over time. The goal is to produce a meaningful representation of the data variable labels based upon event label co-occurrences. With multiple sources of data and many variables, producing dashboards to monitor changes in a way which allows a *human-in-the-loop* to examine changes in the associations is an important task. This work addresses these issues by providing a new approach for examining a corpus of data with minimal user interface time expenditure by relying upon network diagram visualizations which loses less information than some other plotting approaches and are easily digested by human users.

Two real datasets of sequential events are used from unrelated processes; that of an animal pet shelter, and crime reports from a police department. The outputs show that there is an intuitive interpretation of the network diagrams presented which reflect the prior expectations of the nature of the data and the variable names (column headers). Interesting features which may come as a surprise can easily be spotted or features which represent an organization of the variable labels. What is of importance to the viewer, of the expected dashboard use case, is to identify the collection of variable label co-occurrences that which have common labels in their events with respect to other variables. This can be more simply described as, 'for each column, the labels seen in the events recorded; which labels had some overlap with another variable outcome'? The methodological exploration shows that this can be represented concisely as the walks of length 2 from the co-occurrence networks that are equivalent to the edges produced from 'triangulation' and is examined in multiple contexts. Examining the co-occurrence network (walks of length 1) and the triangulation (walks of length 2 via a triangulation from the squaring of the weighted network adjacency matrix) provides a concise summary of the events in a single diagram. From the figures produced it can be seen how this methodology could be applied in practice to examine feature associations and the changes with less latency than grid based approaches or those which which display complex feature measurements.

A set of extensions would be the unsupervised detection of variables which represent time stamps and to assess the ordering and temporal locality of the event positions to produce a more accurate placement the subsetting of the data (non-uniform event recordings over time). Another is to look at the different sets of variables which can be produced and to highlight those associations which have more structural network changes. There are situations in which a 'variable label' can have actual natural language stored as text, and this is not taken into account in this approach as it is considered to be closer to a document store than an event log which is the prime purpose of this methodological approach. There is also opportunity to make improvements by adapting the variable selection according to the network structure represented in the matrices produced. This would allow the methodology to handle circumstances where the inputs streams can go offline or for new streams to be included.

Abbreviations

VoI: variation of information; HITL: human in the loop.

Authors' contributions

AVM supervised the effort to design the methodology and implemented/coded the approach in software. AVM and TGW designed the use cases and application direction. AVM and CET worked on the testing and fixing of errors. AVM, TGW, CET, and DE worked together on refining the approach from testing cycles with the results on real and synthetic datasets. AVM wrote the first draft of the manuscript and all the authors contributed to the editing process. All authors read and approved the final manuscript.

Author details

[1] University of Central Florida, 4000 Central Florida Blvd, Orlando, FL 32816, USA. [2] Glasgow, Scotland, UK.

Acknowledgements

Not applicable.

Competing interests

The authors declare that they have no competing interests.

Availability of data and materials

Two real datasets are used in this work which are publicly available at https://www.data.gov, and can be found using the site's 'data explorer' tool. The 'Bloomington Animal Shelter' data can be found at: https://catalog.data.gov/dataset/animal-care-and-control-adopted-animals-7ed9e and the Seattle Police Department data at: https://catalog.data.gov/dataset/crime-data-76bd0.

Funding

Not applicable.

Appendix

Challenges with a matrix view of the event label co-occurrences

The strengths of the different choices on how to view the event data; the benefit of the use of network diagrams is evident when comparing them to the examples using the matrix view with a heatmap for displaying the values in the cells of the data structure. In Fig. 4 is presented the matrix view of co-occurrence event data which is an alternative approach to the network diagrams presented (examples from the animal shelter data shown "Bloomington Animal Shelter" section). Subfigure a shows the original data as an aggregate co-occurrence value for each cell from the events. The diagonal will be zero as will be the co-occurrence between different labels of the same variable (column entry). If there are large values, the small values will typically be seen as absent (erroneously) from

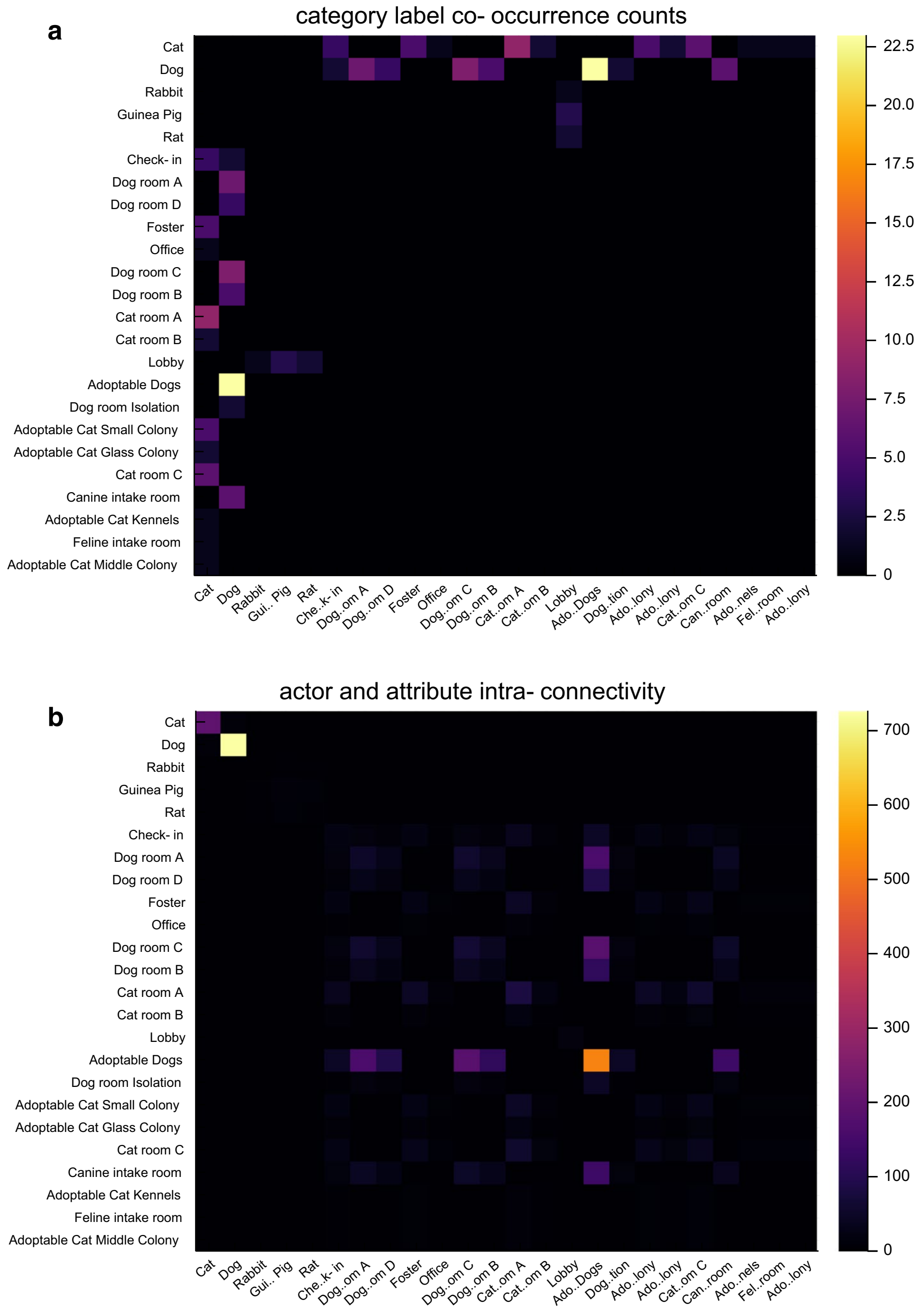

Fig. 4 This is the matrix view of the associations based upon a count of event co-occurrences (data from Bloomington Animal Shelter *data.gov*). **a** Shows the discrete variable value event co-occurrences in the rows of the data (csv in this case) as an aggregate count between labels. **b** Shows the connectivity with labels belonging to the same variable (column) as a result from counting walks of length 2 (taking the square of the co-occurrence matrix). Both subfigures show the challenge of interpreting the data from such a visualization since sparsity, many label numbers, and large value discrepancies create challenges to the viewer (x-axis label names are shortened)

the viewer and the structure of the associations missed. Subfigure b shows the matrix view of the number of walks of length 2 between labels (excluding the walk of length 1). This allows the viewer to see the equivalent data for the within variable association as intra connectivity. All of the entries between difference variable (column) label values will be zero. Looking at the corresponding network diagram in Fig. 1b it can be seen how this is more clear and scalable to trace associations with less effort. In addition, the variable name lengths can require rescaling the already compact view of the associations.

In situations where there are skewed distributions of the values, the stark contrasts of the color scheme in the colormap of the heatmaps may result in incorrect assumptions that no co-occurrence exists. As well an expanded view in which the viewer can trace the grid for the value between label names requires visual concentration which is a very practical problem in comparison to the network diagrams which have parameters for the amount of distance drawn between nodes and edges. The value of this approach can be in situations where there is a need to look at a very coarse high level view of the block structure of the variable label co-occurrences.

Publisher's Note

Springer Nature remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.

Received: 20 December 2018 Accepted: 25 February 2019
Published online: 07 March 2019

References

1. Jia Z, Zhan J, Wang L, Han R, McKee SA, Yang Q, Luo C, Li J. Characterizing and subsetting big data workloads, In: 2014 IEEE international symposium on workload characterization (IISWC). IEEE: New York; 2014. p. 191–201.
2. Azar AT, Hassanien AE. Dimensionality reduction of medical big data using neural-fuzzy classifier. Soft Comput. 2015;19(4):1115.
3. Faiola A, Srinivas P, Hillier S. Improving patient safety: integrating data visualization and communication into ICU workflow to reduce cognitive load, In: Proceedings of the international symposium on human factors and ergonomics in health care, vol. 4. SAGE Publications Sage India: New Delhi; 2015. p. 55–61.
4. U.U. Nations. 2014 revision of the world urbanization prospects. New York: United Nations; 2014.
5. Kitchin R. The real-time city? Big data and smart urbanism. GeoJournal. 2014;79(1):1.
6. Djahel S, Smith N, Wang S, Murphy J. Reducing emergency services response time in smart cities: an advanced adaptive and fuzzy approach, In: 2015 IEEE First International smart cities conference (ISC2). IEEE: New York; 2015. p. 1–8.
7. Puiu D, Barnaghi P, Tönjes R, Kümper D, Ali MI, Mileo A, Parreira JX, Fischer M, Kolozali S, Farajidavar N, et al. Citypulse: large scale data analytics framework for smart cities. IEEE Access. 2016;4:1086.
8. Hall DL, Llinas J. An introduction to multisensor data fusion. Proc IEEE. 1997;85(1):6.
9. Giese M, Soylu A, Vega-Gorgojo G, Waaler A, Haase P, Jiménez-Ruiz E, Lanti D, Rezk M, Xiao G, Özçep Ö, et al. Optique: zooming in on big data. Computer. 2015;48(3):60–7.
10. DellAglio D, Della Valle E, van Harmelen F, Bernstein A. Stream reasoning: a survey and outlook. Data Sci. 2017;1:59–83.
11. Lachhab F, Bakhouya M, Ouladsine R, Essaaidi M. Performance evaluation of linked stream data processing engines for situational awareness applications. Concurr Comput Pract Exp. 2018;30(12):e4380.
12. Simpao A, Ahumada L, Rehman M. Big data and visual analytics in anaesthesia and health care. Br J Anaesth. 2015;115(3):350.
13. Cohen AM, Hersh WR, Dubay C, Spackman K. Using co-occurrence network structure to extract synonymous gene and protein names from MEDLINE abstracts. BMC Bioinform. 2005;6(1):103.
14. Liu Z, Navathe SB, Stasko JT. Network-based visual analysis of tabular data. In: 2011 IEEE conference on visual analytics science and technology (VAST). IEEE: New York; 2011. p. 41–50.
15. Endert A, Hossain MS, Ramakrishnan N, North C, Fiaux P, Andrews C. The human is the loop: new directions for visual analytics. J Intell Inf Syst. 2014;43(3):411.
16. Holzinger A. Interactive machine learning for health informatics: when do we need the human-in-the-loop? Brain Inform. 2016;3(2):119.
17. Alba A, Coden A, Gentile AL, Gruhl D, Ristoski P, Welch S. Multi-lingual concept extraction with linked data and human-in-the-loop. In: Proceedings of the knowledge capture conference. ACM: New York. 2017. p. 24.
18. Basole RC, Srinivasan A, Park H, Patel S. ecoxight: discovery, exploration, and analysis of business ecosystems using interactive visualization. ACM Trans Manag Inf Syst (TMIS). 2018;9(2):6.
19. Newman ME, Clauset A. Structure and inference in annotated networks. Nat Commun. 2016;7:11863.
20. Bothorel C, Cruz JD, Magnani M, Micenkova B. Clustering attributed graphs: models, measures and methods. Netw Sci. 2015;3(3):408.

21. Dong Y, Chawla NV, Swami A. metapath2vec: scalable representation learning for heterogeneous networks. In: Proceedings of the 23rd ACM SIGKDD international conference on knowledge discovery and data mining. ACM: New York; 2017. p. 135–44.
22. Yu X, Ren X, Sun Y, Gu Q, Sturt B, Khandelwal U, Norick B, Han J. Personalized entity recommendation: a heterogeneous information network approach. In: Proceedings of the 7th ACM international conference on Web search and data mining. ACM: New York; 2014. p. 283–92.
23. Van den Elzen S, Van Wijk JJ. Multivariate network exploration and presentation: from detail to overview via selections and aggregations. IEEE Trans Vis Comput Graph. 2014;20(12):2310.
24. Modoni GE, Sacco M, Terkaj W. A survey of RDF store solutions. In: 2014 international ICE conference on engineering, technology and innovation (ICE). IEEE: New York; 2014. p. 1–7.
25. Yoon BH, Kim SK, Kim SY. Use of graph database for the integration of heterogeneous biological data. Genom Inform. 2017;15(1):19.
26. Huang Z, Zheng Y, Cheng R, Sun Y, Mamoulis N, Li X. Meta structure: computing relevance in large heterogeneous information networks. In: Proceedings of the 22nd ACM SIGKDD international conference on knowledge discovery and data mining. ACM: New York; 2016. p. 1595–604.
27. Yung D, Chang SK. Answering how-to-reach query in big attributed graph databases. In: 2017 IEEE third international conference on Big Data computing service and applications (BigDataService). IEEE: New York; 2017. p. 141–8.
28. Kitchin R, Coletta C, McArdle G. Urban informatics, governmentality and the logics of urban control: the programmable city working paper 25; 2017.
29. Huang X, Zhao Y, Ma C, Yang J, Ye X, Zhang C. TrajGraph: a graph-based visual analytics approach to studying urban network centralities using taxi trajectory data. IEEE Trans Vis Comput Graph. 2016;22(1):160.
30. Liu Y, Safavi T, Dighe A, Koutra D. Graph summarization methods and applications: a survey. ACM Comput Surv. 2018;51(3):62.
31. Weaver C. Multidimensional data dissection using attribute relationship graphs. In: 2010 IEEE symposium on visual analytics science and technology (VAST). IEEE: New York; 2010. p. 75–82.
32. Liu Z, Navathe SB, Stasko JT. Ploceus: modeling, visualizing, and analyzing tabular data as networks. Inf Vis. 2014;13(1):59.
33. Bezanson J, Edelman A, Karpinski S, Shah VB. Julia: a fresh approach to numerical computing. SIAM Rev. 2017;59(1):65.
34. Ellson J, Gansner E, Koutsofios L, North SC, Woodhull G. Graphviz open source graph drawing tools. In: International symposium on graph drawing. Springer: New York; 2001. p. 483–4.
35. Frey BJ, Dueck D. Clustering by passing messages between data points. Science. 2007;315(5814):972.
36. Meilă M. Comparing clusterings by the variation of information. In: Learning theory and kernel machines. Springer: Berlin; 2003. p. 173–87.
37. van Rossum G. Python patterns-implementing graphs, Python essays. Python Software Foundation. 1998–2003.
38. Katz L. A new status index derived from sociometric analysis. Psychometrika. 1953;18(1):39.
39. Mantzaris AV, Higham DJ. Infering and calibrating triadic closure in a dynamic network. In: Jini J, editor. Temporal networks. Springer: Berlin; 2013. p. 265–82.
40. Davie AM, Stothers AJ. Improved bound for complexity of matrix multiplication. Proc R Soc Edinb Sect A Math. 2013;143(2):351.
41. Le Gall F. Powers of tensors and fast matrix multiplication. In: Proceedings of the 39th international symposium on symbolic and algebraic computation. ACM: New York; 2014. p. 296–303.