SIRIUS: A MUTUAL INFORMATION NETWORK TOOL FOR EXPLORATORY VISUALIZATION OF MIXED DATA

A PREPRINT

Jane Lydia Adams[*]
Jane.Adams@uvm.edu

Todd F. DeLuca[*]
Todd.DeLuca@uvm.edu

Christopher M. Danforth[*]
Chris.Danforth@uvm.edu

Peter Sheridan Dodds[*]
Peter.Dodds@uvm.edu

Yuhang Zheng[†]
YZheng83@MassMutual.com

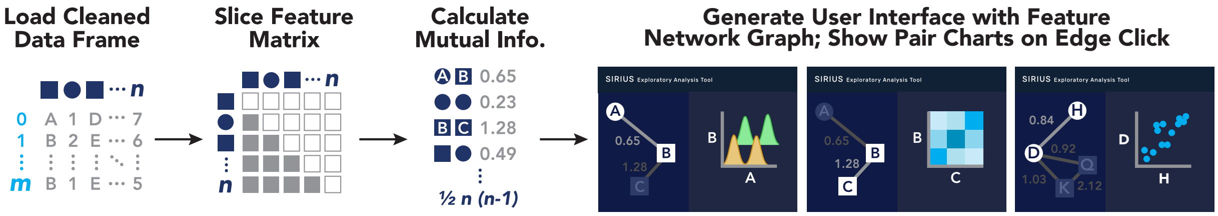
Konstantinos Anastasakis[†]
KAnastasakis26@MassMutual.com

Boyoon Choi[†]
BChoi19@MassMutual.com

Allison Min[†]
AMin13@massmutual.com

Michael M. Bessey[†]
MBessey@MassMutual.com

ABSTRACT



Data scientists across disciplines are increasingly in need of exploratory analysis tools for data sets with a high volume of features [Menon and Hegde, 2015a]. We expand upon graph mining approaches for exploratory analysis of high-dimensional data to introduce Sirius, a visualization package for researchers to explore feature relationships among mixed data types using mutual information and network backbone sparsification. Visualizations of feature relationships aid data scientists in finding meaningful dependence among features, which can engender further analysis for feature selection, feature extraction, projection, identification of proxy variables, or insight into temporal variation at the macro scale. Graph mining approaches for feature analysis exist, such as association networks of binary features, or correlation networks of quantitative features, but mixed data types present a unique challenge for developing comprehensive feature networks for exploratory analysis [Agrawal et al., 1993, Raeder and Chawla, 2011]. Using an information theoretic approach, Sirius supports heterogeneous data sets consisting of binary, continuous quantitative, and discrete categorical data types, and provides a user interface exploring feature pairs with high mutual information scores [Kraskov et al., 2004, Ross, 2014]. We leverage a backbone sparsification approach from network theory as a dimensionality reduction technique, which probabilistically trims edges according to the local network context [Serrano et al., 2009]. Sirius is an open source Python package and Django web application for exploratory visualization, which can be deployed in data analysis pipelines. The Sirius codebase and exemplary data sets can be found at: *https://github.com/compstorylab/sirius*.

[*] University of Vermont Complex Systems Center, College of Engineering and Mathematical Sciences, Burlington, Vermont.
[†] Massachusetts Mutual Life Insurance Company, Springfield, Massachusetts.

Keywords mutual information, exploratory analysis, feature selection, feature extraction, graph mining, network graphs, data visualization

1 Introduction

We introduce the mutual information feature network as a technique compiling established information theory and network science algorithms into a tool for exploratory analysis. We provide Sirius[3], a Python package and web application to demonstrate this technique [Adams et al.]. The result is an interactive network graph of feature dependence weighted by mutual information, with charts of 2-dimensional feature relationships shown upon interaction. The mutual information feature network aids exploratory analysis of feature level relationships by narrowing the scope of visual comparison among features through statistical methods, while preserving access to record level data among feature pairs of potential interest. Sirius is an open source repository for implementation of the mutual information feature network, supported with several example data sets of the sort commonly used in real-world data analysis.

The purpose of this technique is to inform subsequent hypothesis generation, clustering, feature selection, imputation and predictive modeling in a data science pipeline. In this paper, we: review existing high dimensional exploratory visualization approaches; describe the technique and mechanisms; discuss examples of domain-specific applications; and note areas for potential future research in the visualization space of high dimensional feature exploration through mutual information network graphs.

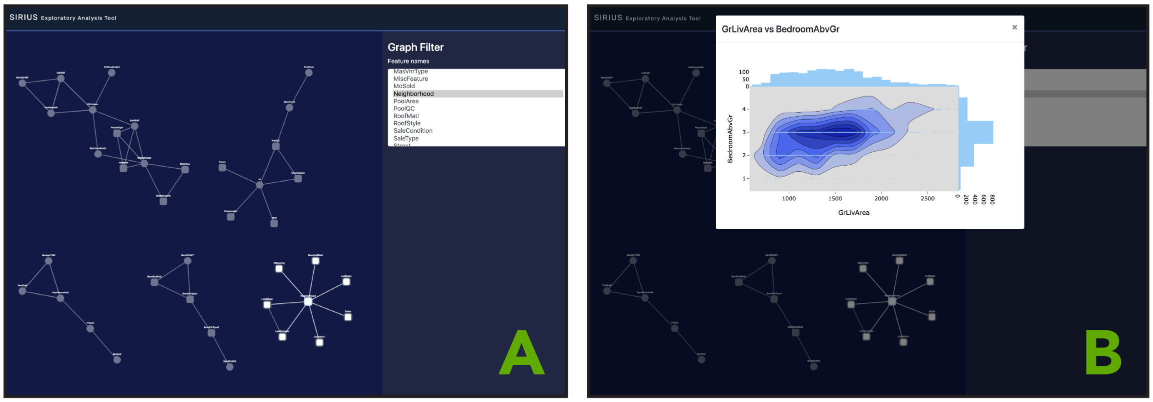


Figure 1: Screenshots of the Sirius web application: (A) The mutual information feature network generated from a provided example data set, along with a graph filter side panel which allows a user to select a feature of interest. (B) A scatter plot showing record-level data for two continuous features of interest, selected by clicking on an edge between feature nodes. Both the network graph and the pairwise charts are interactive, allowing users to pan and zoom for in-depth analysis.

2 Background

Statistical analysis provides robust mathematical insight for a wide range of domain applications, including finance [Griebel and Holtz, 2010], marketing [Russell and Petersen, 2000, Raeder and Chawla, 2011], genomics [Langfelder and Horvath, 2008, Nam et al., 2007, Pyne et al., 2009], chemistry [Wold et al., 1984, Su et al., 2017, McCarthy et al., 2004], environmental science [Warton, 2011], and social interaction [Gao et al., 2015, Gotz and Stavropoulos, 2014, Shuman et al., 2013]. Advances in computational power and storage, coupled with a rapid uptick in data collection and exchange, have caused massive increases in the volume of data collected: by some estimates, 90% of all current data was created in the past few years [Menon and Hegde, 2015b, Al-Jarrah et al., 2015]. This has engendered the development of powerful machine learning tools which support imputation, community detection, clustering, and prediction [Fahad et al., 2014, Ward and Barker, 2013, Chen et al., 2014].

However, there are growing ethical concerns among researchers and the general public about 'black box' models, which can obscure inadvertent statistical biases in the machine learning pipeline and lead to disparate real-world impacts [Zafar et al., 2017, Feldman et al., 2015]. Simply excluding protected class data, such as race or income, from model

[3] The Sirius codebase and exemplary data sets can be found at: *https://github.com/compstorylab/sirius*

inputs is not sufficient to offset disparate impacts when other features act as proxy variables [Wan, 2018, Skeem and Lowenkamp, 2016]. This problem is in part due to the obfuscation of raw data (e.g. through summary statistics, feature projection, or complex neural network architectures) from data scientists during the exploration and development phases of the statistical modeling pipeline. Exploratory Data Analysis (EDA) preserves data scientists' access to record level data, while addressing the growing need to holistically explore a large number of complexly related features [Behrens, 1997].

Traditional visualization methods are generally **vertically scalable**: that is, as the number of records grows, the challenge for visual representation is primarily in computational power, e.g. browsers' memory struggling to support rendering of thousands of circle glyphs in a scatter plot on a web page. However, **horizontal scalability** (increasing the number of features) when visualizing data for exploratory analysis presents a challenge for traditional aesthetic encoding methods: while a data set with 4 features f might visually encode f_1: x-position, f_2: y-position, f_3: point size and f_4: color for each record in a scatterplot, increasing the number of features to 100 is not simply a question of increasing visual complexity to add z-position, shape, texture, time, transparency, sonification, etc. encodings up to f_{100}. Instead of continuing this line of ever-growing encoding complexity, it becomes helpful to consider applications of visual or statistical summary methods for analysis of feature dependence in a visual context.

3 Motivations

Our aims for this project are to support researchers in the following tasks:

- Support EDA of high numbers of features of mixed continuous, discrete, and binary data types, as these large-scale heterogenous data sets are common in a wide range of real-world data analysis tasks.
- Identify groupings of related features to inform feature selection. Including all features in a prediction task can be computationally expensive, and the addition of unrelated features can sometimes inhibit predictive accuracy due to overfitting and the 'curse of dimensionality' [Bertini et al., 2011]. For example, random forest regression models can generate feature importance rankings; subsequent modeling with the exclusion of low-importance features can improve model accuracy, but this iterative approach can be time- and resource-prohibitive.
- Suggest related features which could be grouped together in a lower-dimensional visualization, as through Principle Component Analysis (PCA). For example, in a healthcare setting, there may be many features related to body systems, such as blood sugar, cardiovascular status, or liver health. Quickly identifying all such groupings can allow data scientists to extract a single feature to represent all features in a given system and more easily generate meta-analyses of interacting body systems.
- Highlight potential 'proxy variables' or features with implications for privacy and equity. Unexpected feature relationships can sometimes indicate disparate impacts for certain populations, or deanonymization vulnerabilities in sensitive data. For example, low appraisal values of homes in certain zip codes may warrant investigation of potential redlining effects. An innocuous "ID number" feature intended to anonymize protected patient health data, when shown to be associated with a hospital admittance source, can drastically narrow a search space for nefarious actors seeking to deanonymize data. Bringing these unforeseen associations to the attention of decision-makers can provide vital insight for policy-making and have significant consequences for stakeholders.
- In all of these contexts, provide a dual view of data, allowing rapid task switching between macro-scale, 'big picture' feature relationships; and micro-scale, record-level exploratory analysis, wherein specific subpopulation and individual data can be accessed visually.

4 Related Work

We present prior contributions to scientific literature on the topics of visualization for high-dimensional data, using network graphs to visualize feature relationships, and mutual information.

4.1 Visualizing High Dimensional Data

4.1.1 Small Multiples

One approach for EDA of a limited number of features is small multiples [MacEachren et al., 2003]. By tiling features across rows and columns, all permutations of paired feature charts can be viewed simultaneously. This method is advantageous in that it supports visualization of heterogenous data types, by selecting appropriate charts for each feature

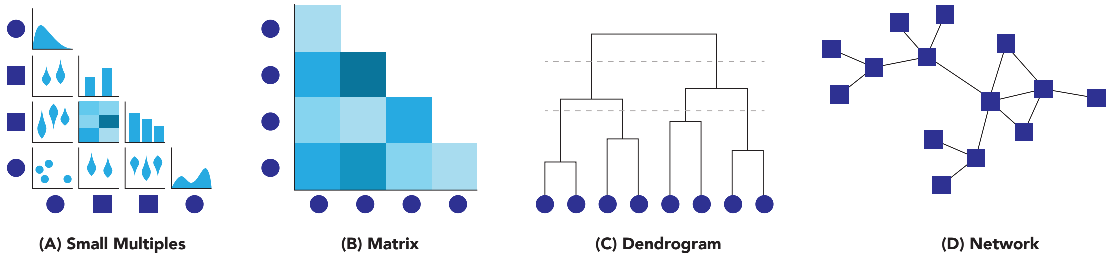


Figure 2: Some different approaches for graphically representing feature relationships. From left to right: Small multiples (A), which are advantageous for small numbers of features of homogenous or heterogenous data types; (B) Matrices, i.e. correlation or conditional probability matrices, and (C) dendrograms, both advantageous for comparing small or medium numbers of features of a homogenous data type using summary statistics or similarity/distance measures; and (D) Networks which support a high number of features, usually of a homogenous data type, but supporting heterogenenous data types through the Sirius mutual information implementation.

pairing (e.g. scatter plots for continuous-continuous pairings, heatmaps or conditional bar charts for discrete-discrete pairings, and violin plots or ridgeline plots for discrete-continuous pairings) as illustrated in Fig. 2. This is the approach of *INFUSE: Interactive Feature Selection for Predictive Modeling of High Dimensional Data*, which uses small multiples of pie charts showing feature importance scores across quantitative features [Bertini et al., 2011].

4.1.2 Dendrograms and Matrices

Dendrograms are hierarchical tree visualizations which leverage a similarity measure to cluster features according to their relationship across records. One such notable example of this is the *Hierarchical Clustering Explorer (HCE)*, which uses hierarchical agglomerative clustering (HAC) to group features by similarity [Seo and Shneiderman, 2005]. The HCE dashboard then generates similarity matrices of features. Matrices are often visually represented as heatmaps which are colored according to the summary statistics of feature comparisons. These can accommodate higher numbers of feature comparisons in the visual space due to their summational behavior.

4.1.3 Network Graphs for Feature Exploration

Two prominent feature selection methods employing network graph visualizations include *Association Networks* and *Correlation Networks*.

Association networks are commonly employed in market basket analysis, in which edges are drawn between items if purchasing one item increases the likelihood of purchasing a different item, a weight referred to as the ‘lift’, derived from the conditional probabilities of purchasing two items in the same transaction [Agrawal et al., 1993]. Thus, these association networks are directed weighted networks, and are classically defined for boolean data types only. The resultant network visualizations show feature groupings among products which are frequently purchased together, and might provide actionable insight for marketers looking to target potential consumers of a certain product. Association networks are discussed in more detail in Section 6.2 in reference to the ‘groceries’ case study.

Correlation networks, popularized by *Weighted Gene Correlation Network Analysis (WGCNA)*, are commonly applied in genomics research [Langfelder and Horvath, 2008]. Large network graphs are generated in which nodes represent genes, and edges are drawn according to co-expression correlation across cell lines. Data types for this kind of visualization are classically all continuous values, such as gene expression probabilities [Niemira et al., 2019].

These kinds of feature-driven networks, in high-dimensional settings, can often become extremely dense and difficult to navigate during exploratory analysis. Both association networks and correlation networks also do not support comparisons across heterogeneous data types. The key value propositions of the Sirius mutual information feature network method described in Section 5.5 are 1) backbone sparsification, which acts as a form of dimensionality reduction to simplify the exploratory space, and 2) comparison among and between continuous and discrete data types, as described in section 5.4.

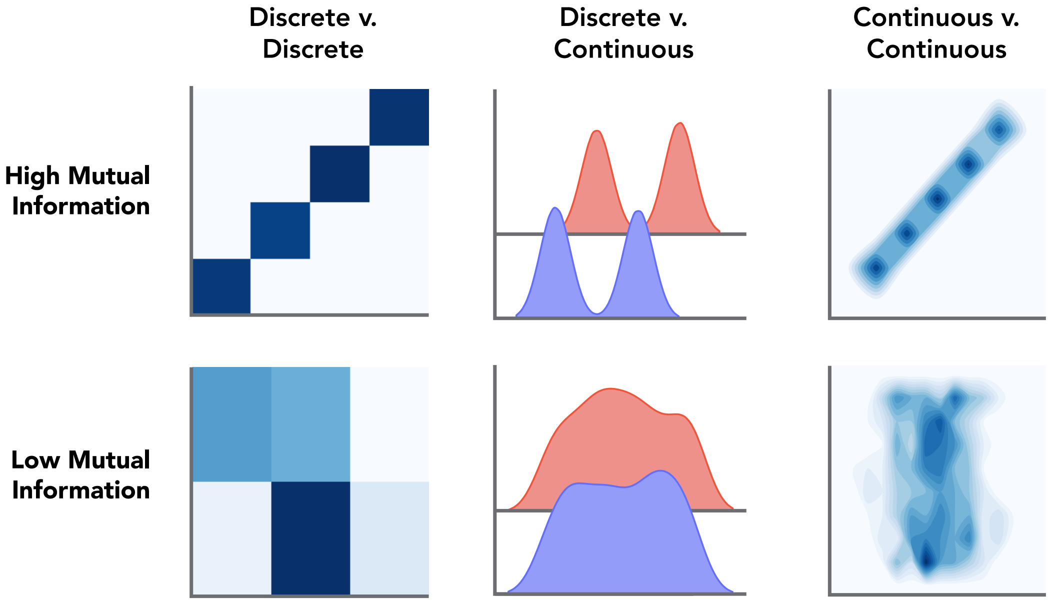


Figure 3: Some charts exemplifying, in abstract representation, the range of mutual information for the three permutations of possible data type pairings.

4.2 Mutual Information

Mutual information is a way of evaluating dependence of two features by measuring entropy reduction. If knowing the response to one feature for a given record increases one's accuracy of predicting that record's value for another feature, it is likely that these two features share high mutual information. In the mutual information feature network technique we present here, features with high mutual information would be connected in the resulting network graph. Conversely, if knowing the response to one feature does not significantly improve prediction of a second feature, those features are likely relatively independent and will have a low mutual information. In the Sirius application, an edge between these two unrelated features' nodes will likely not be displayed in the graph. An abstracted matrix of data type pairings and mutual information ranges to aid understanding is presented in Fig. 3.

5 Methods

5.1 Terminology

We discern between scales in that 'record level' refers to pairwise graphs showing data for individual records in a data set (e.g. "540 East Street", "Patient 001754", or "Grocery Order #AQB54") whereas 'feature level' observations refer to relationships between features generally (e.g. "Number of Bedrooms" and "Lot Size", "Blood Oxygen" and "Asthma", or "Whole Milk" and "Cookies"). It may help to think of 'features' as the columns in a spreadsheet, and 'records' as rows.

5.2 Technique

In this technique, a network graph is generated in which nodes represent features, and edges are weighted between features by mutual information score. The network graph is a relational encoding, well suited to visualizing feature dependence. Nodes are positioned relative to one another, and there are no horizontal and vertical axes in the traditional sense. Therefore, generating a network graph multiple times may render it mirrored or rotated relative to its first render, as these are all valid isomorphisms of the same graph[Tomihisa, 1989]. Edge display is thresholded dynamically, informed by a 'backbone' graph thinning method commonly employed in network science to extract the critical structures of a complex graph[Bagrow et al., 2015, Serrano et al., 2009]. Clicking on an edge between two feature nodes renders a chart of all records conditioned on the pair of features connected by the graph edge. The chart type rendered is determined by the data types of the two features being compared. These charts include ridge line plots, heat maps, and 2D kernel density plots with scatter plot overlays. This information theoretic approach enables researchers to quickly view individual record data through statistically identified feature relationships of potential interest, with many tuning parameters for iterative customization in exploration.

The mutual information feature network approach is outlined as this method:

- Classify features as discrete or continuous;
- Compute mutual information between each pair of features based on detected data type;
- Thin edges according to a 'backbone' method which selects the most statistically unexpected edge weights for each node;
- Generate charts for each feature pair based on data type;
- Generate a network graph of features and weighted edges, and charts for each pair of features to be displayed on edge selection.

The open source implementation of this method, Sirius, performs data processing in Python using Pandas data structures, and renders a network visualization in-browser using a Django web application[McKinney, 2010].

5.3 Setting Parameter Arguments and Loading Data

After installing the Sirius package according to the instructions in the package README, all data processing can be run from the command line or through the provided walkthrough notebook. There are a number of customizable parameters in the Sirius script, which can be changed using flags when running the script from the command line, or directly through the declared variables in the notebook file.

Sirius is designed to take place after the extract, transform, load (ETL) step in a data processing pipeline. It is recommended that data be cleaned to remove outliers, and that temporal and geospatial columns be removed or encoded (e.g. map string birth dates to numeric ages, or map numeric postal codes to string county names). This step is omitted from the scope of this paper due to the unique edge cases that often arise in real-world data science applications. However, the intended generalizability of this technique means that it is designed not to break in these edge cases, but rather force-classify features as continuous or discrete data types.

Each feature is classified as either discrete or continuous for the purposes of mutual information method selection and visualization chart type. These do not necessarily map to numeric or string data types. The unique response count threshold between discrete and continuous features is a parameter that can be adjusted in the parameter file, or in the command line argument flags. If the number of unique responses for a feature is less than the specified threshold, the feature is treated as discrete. Thus, a numeric feature with unique numeric responses {1,2,3} would be treated as a discrete (categorical) value by default, and its corresponding charts would either be heatmaps or ridgeline plots, depending on whether it was paired with a discrete or continuous feature, respectively. Features are also classified as discrete when the number of unique responses exceeds the threshold in the event that the unique feature responses cannot be converted to numeric values (e.g. {'AU','EU','UK'...} when all country codes are present in the data set). This may produce heatmaps with many cells, or ridgeline plots with many rows, in charts comparing this feature to another discrete or continuous feature, respectively.

An **edge list** is obtained by taking the upper or lower triangle of an undirected (symmetric) n x n feature matrix, excluding the diagonal matrix identity. The resulting edge list has a length $\frac{n^2-n}{2}$, such that all feature nodes are connected once, prior to sparsification.

5.4 Calculating mutual information

Mutual information (I) is computed between each pair of features (U, V), and listed as the weight of the corresponding edge. The default behavior of the application is to calculate mutual information only across records for which there is a non-null value for both features. Thus, if there are no records for which U and V are non-null, $I_{U;V} = 0$. Users might opt to consider $r = \texttt{null}$ as an additional discrete 'category' (for example, if the lack of a response might be considered informative for the domain application), or fill missing continuous values with the range minimum or median.

5.4.1 Discrete/Discrete Data Type Pairs

For a pair of discrete data type features, mutual information is calculated using probability mass functions across all non-null $i \in U$ and $j \in V$. The mutual information formula for discrete-discrete pairs is given as:

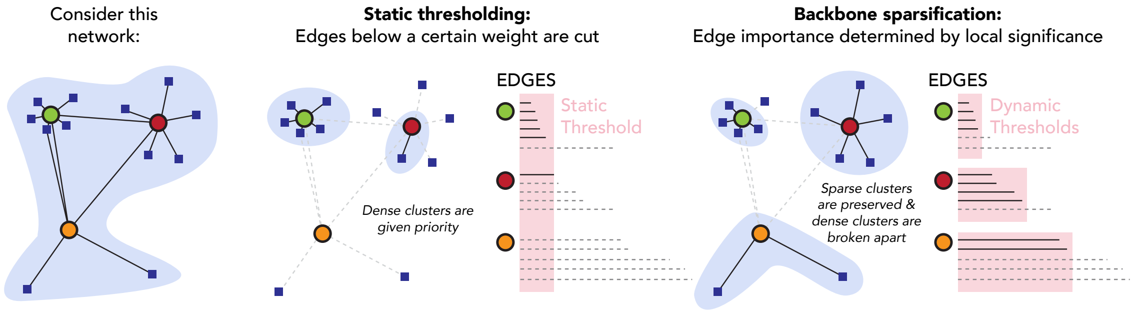

Consider this network:
Static thresholding:
Edges below a certain weight are cut
EDGES
Static Threshold
Dense clusters are given priority
Backbone sparsification:
Edge importance determined by local significance
EDGES
Dynamic Thresholds
Sparse clusters are preserved & dense clusters are broken apart

Figure 4: Explanatory graphic demonstrating the value of the backbone method as a dynamic thresholding sparsification approach, compared to static thresholding.

$$I_{U;V} = \sum_{i \in U} \sum_{j \in V} = P(i,j) + log \frac{P(i,j)}{P(i) * P(j)} \tag{1}$$

5.4.2 Continuous/Continuous Data Type Pairs

Converting this method from probability mass functions to probability density functions for a pair of continuous data type features yields the formula:

$$I_{U;V} = \int_U \int_V P_{(U,V)}(i,j) + log \frac{P_{(U,V)}(i,j)}{P_U(i) * P_V(j)} \, di \, dj \tag{2}$$

where $P_{(U,V)}$ is the joint probability density function of U and V, and P_U,P_V are the marginal probability density functions of U and V, respectively[Kraskov et al., 2004].

5.4.3 Discrete/Continuous Data Type Pairs

A heterogenous pairing of data types employs a nearest neighbor mutual information regression method[Ross, 2014, Pedregosa et al., 2011]. Discrete features with r responses are converted from a 1D array of length m to a sparse matrix of shape (r, m), with columns for each unique response and values of 0 or 1 for all records k. Nearest neighbor approximations of mutual information in a sample population are less susceptible to variance due to parameterization than traditional binning methods; e.g. adjusting the neighbor N count for each point x does not influence the variance in mutual information as significantly as adjusting the bandwidth of a Gaussian kernel[Ross, 2014].

$$I_{(U,V)} = \langle I_i \rangle = \psi(N_x) - \langle \psi(N_x) \rangle + \psi(k) - \psi(m) \tag{3}$$

Following the behavior prescribed by the libraries used as detailed in Section 5.6, $[u \in U, v \in V] : I_{(u;v)} \geq 0$, as mutual information cannot be negative, since it is a measure of dependence and zero corresponds to complete independence[Pedregosa et al., 2011, Kraskov et al., 2004].

SciKit Learn's `feature_selection` library is used to compute mutual information for all data type pairs using the `mutual_info_regression` function. This is described in Section 5.6[Pedregosa et al., 2011]. Further research could warrant additional parameterization or comparative evaluation of various other mutual information estimator methods, but this is generally accepted as a benchmark standard method of computing mutual information.

5.5 Network Graphing

Sparsification of the resulting feature matrix allows for a narrowing of the exploratory visualization space in a structurally aware manner, as shown in Figure 4. In this technique, edge thinning is performed through a 'backbone' method, which preserves links with high statistical significance and removes all others [Bagrow et al., 2015, Serrano et al., 2009]. The equation is outlined as follows:

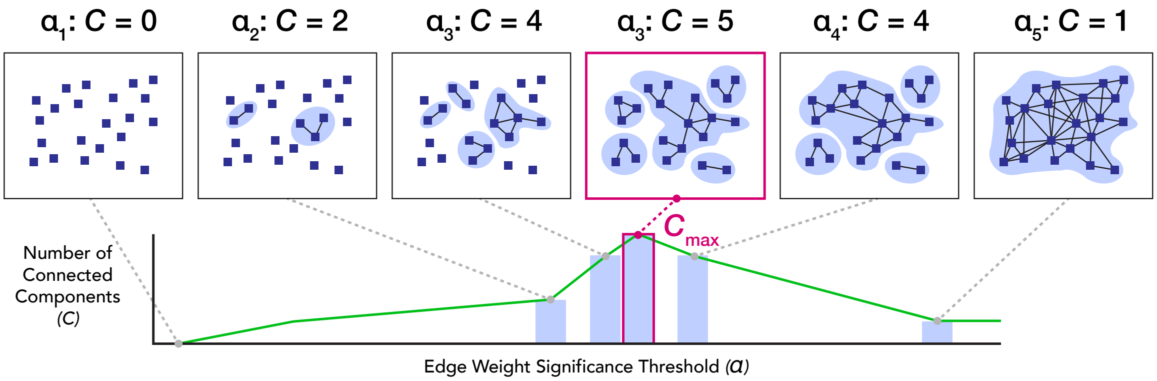

α1: C = 0
α2: C = 2
α3: C = 4
α3: C = 5
α4: C = 4
α5: C = 1
Cmax
Number of Connected Components (C)
Edge Weight Significance Threshold (α)

Figure 5: When the alpha significance is at 0, no nodes are connected, and thus the number of components C is 0 (α_1). As the alpha threshold for the network is adjusted upwards, the number of edges increases, creating more connected components. At a certain alpha threshold (here, α_3), the number of connected components in the graph reaches a maximum (here, n_C = 5 at α_3), before decreasing as discrete components become connected to one another, finally resulting in a single connected component encompassing the entire graph (α_5).

$$\alpha_{ij} = 1 - (k-1)\int_0^{P_{ij}} (1-x)^{k-2}dx < \alpha \tag{4}$$

where α_{ij} is the statistical significance of a given edge, k is the node degree (the number of nodes that this node is connected to), P_{ij} is the probability of an edge with this weight x, and α is the significance threshold.

In this way, there is not a specific mutual information threshold set for the entire network, but rather each node is evaluated individually, and its edges are preserved or thinned according to how relatively important an edge is to that node specifically. An alpha threshold is chosen for which the number of graph components C is maximal, which provides for unique 'constellations' in the visualization, as shown in Fig 5. This is where we have derived the name Sirius for our exploratory analysis package: Sirius, the brightest star in the night sky, provided navigational support for sailors, just as we hope to provide directional insight for wayward data scientists [Gregersen].

Network statistics such as adjacency and component count are calculated using NetworkX. Graph positioning is also calculated using NetworkX[developer team, 2014]. Layout parameters can be adjusted within the graph layout function. Default behavior is to employ the Fruchterman Reingold Layout, a commonly used force-directed positioning algorithm in which, in this mutual information network use case, highly dependent features are more closely positioned than features which are more independent from one another[Fruchterman and Reingold, 1991].

5.6 User Implementation

The Sirius Python package includes a README with directions on how to run the data processing pipeline and web application. After running the Sirius data processing pipeline, either in-notebook or via the command line, users can specify the graph and chart json files generated by the Python package to be used by the application, and interact with the visualizations in a web browser through the provided Django application.

6 Case Studies

We provide three domain-specific example data sets with Sirius, and discuss two such data sets as case studies here. We encourage readers to test this program on other high-dimensional data sets, and to share results, obstacles, or extensible functionality requests.

6.0.1 Data

Example publicly available data sets used with this tool include (in order of record volume):

- A **healthcare** data set related to intensive care unit (ICU) mortality, comprised of 186 features: a mixture of demographic information, lab results, and risk assessments (e.g "Blood Glucose", "Weight", and "Apache 2

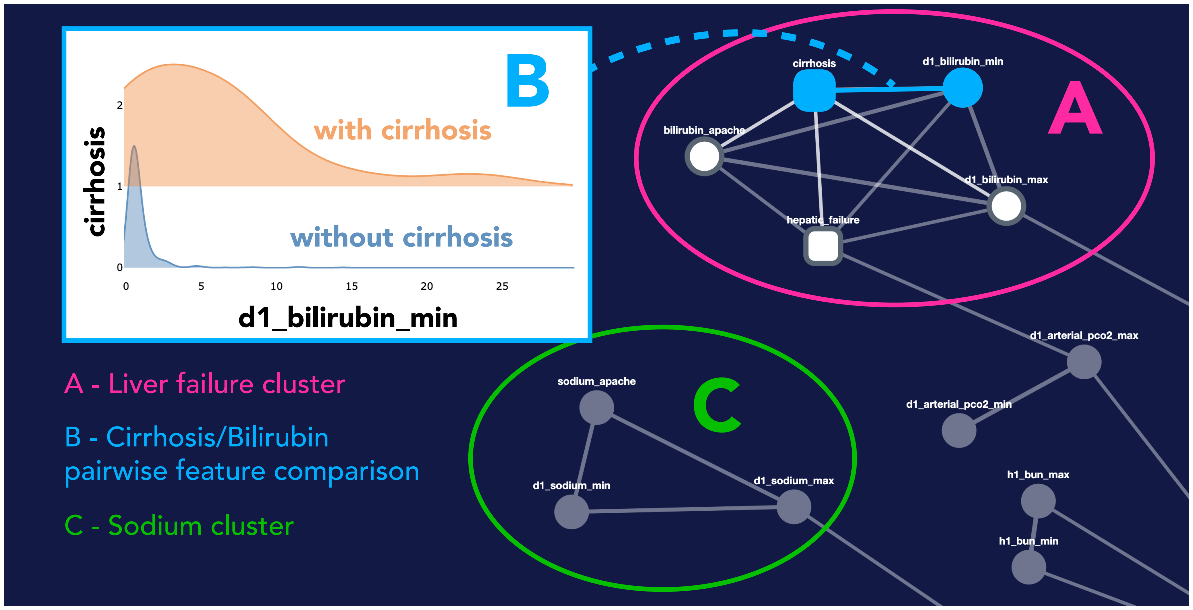


Figure 6: We show an annotated screenshot from Sirius of a portion of the healthcare data set, with groups of feature names related to body systems and measurements. In particular, feature group (A) contains features related to liver function, including quantitative bilirubin readings and boolean flags for cirrhosis and liver failure. This comports with clinical studies which suggest that elevated bilirubin levels may indicate poor liver function [Chen and Lin, 2017]. Paired feature chart (B) shows a ridgeline plot of patient record-level data comparing one such bilirubin value with a diagnostic flag for cirrhosis. Feature group (C) includes features related to sodium levels in patients.

Diagnosis") from 91,713 records of admitted ICU patients [Lee et al., 2020]. *Note that due to the large size of this data set, the raw data s not included in the package itself, but can be pulled from the Kaggle API using the Python script included in the corresponding ICU data folder.*

- A **groceries** data set comprised of 169 features: unique items in a grocery store (e.g. “Vegetables", “Cleaning Supplies", and “Specialty Meats"), with boolean values (corresponding to ‘purchased’ or ‘not purchased’) from 9,835 records of market basket transactions [Nasrullah, 2019].
- The Ames **housing** data set is comprised of 81 features of mixed data types for 1,460 records of homes in Ames, Iowa [Cock, 2019]. It includes physical characteristics of homes such as dimensions and materials, as well as qualitative assessments and sale information.

The resultant network graphs for these data sets group features with statistically significant dependence, with potential applications for: clinicians using data similar to the ICU mortality data set; marketers with data resembling the groceries data set; or real estate agents or tax assessors examining data paralleling the housing data set. This indicates promise for other real-world applications which require comparisons across a large number of features with differing data types.

A synthetic data set is also included in the package, with 34 features of mixed data types for 2,400 artificial records. Feature names correspond to statistical properties of synthesized data. This data is not discussed here, as it is primarily for interface testing and demonstration purposes.

As Sirius is an exploratory analysis tool, many insights and conclusions can be drawn from the data sets provided. We have chosen to describe here a small handful of these insights: feature grouping patterns seen in the **healthcare** data set; and a comparison of the Sirius mutual information feature network to an association network of the **groceries** data set.

6.1 Healthcare

The **healthcare** data set contains 91,713 records of admitted ICU patients, comprised of 186 features. Features include demographic information, lab results, and risk assessments [Lee et al., 2020]. There are a number of features directly related to mortality prediction. Expectedly, many of the risk assessment features scored high in mutual information.

In the healthcare data set, the mutual information feature network gives valuable insight into feature relationships within body systems. As shown in Figure 6, features are often connected that have strong clinical importance to one another. This insight could prove useful for feature extraction, or for clinicians to triage patients by classifying their risk according to prior diagnostic trends.

Number of Patients	Solid Tumor with Metastasis	Mean/Median Arterial pCO$_2$ (min)	Mean/Median Arterial pCO$_2$ (max)
89,820	False	43.465/41.0	44.637/42.1
1,893	True	43.278/41.0	44.361/42.0

The feature networks generated by Sirius from the ICU mortality data also highlight the importance of graphical representation. Consider the feature group highlighted in Figure 7. Referring to the table, summary statistics about the subgroups of patients with and without metastatic solid tumors are extremely close and difficult to differentiate. However, when the mutual information feature network is computed, the slight differences in pCO$_2$ readings among tumor-positive and tumor-negative patients are highlighted in the exploratory analysis space, and researchers are given readily available access to interact with these specific feature comparison charts.

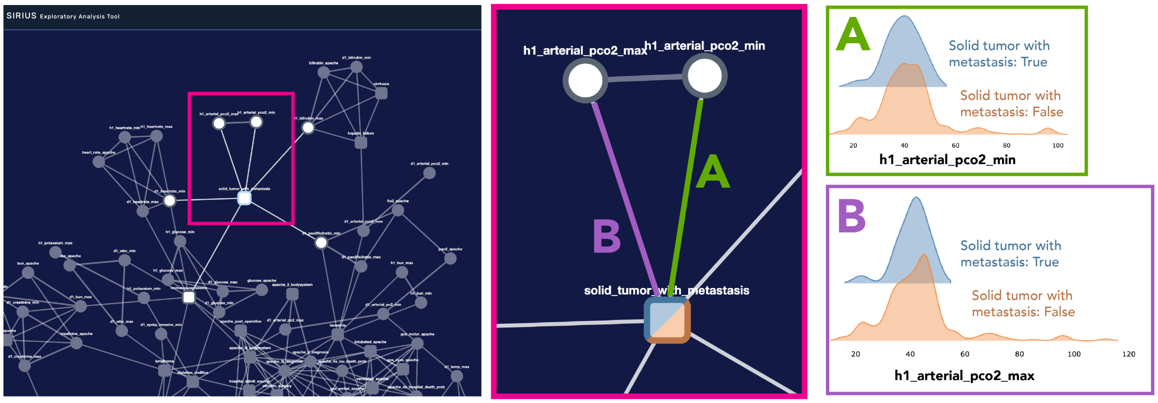

SIRIUS Exploratory Analysis Tool
h1_arterial_pco2_max
h1_arterial_pco2_min
A
B
solid_tumor_with_metastasis
A
Solid tumor with metastasis: True
Solid tumor with metastasis: False
20
40
60
80
100
h1_arterial_pco2_min
B
Solid tumor with metastasis: True
Solid tumor with metastasis: False
20
40
60
80
100
120
h1_arterial_pco2_max

Figure 7: Screenshots from Sirius showing an inset of the healthcare feature network, highlighting variables `solid_tumor_with_metastasis`, `h1_arterial_pco2_min`, and `h1_arterial_pco2_max`. Charts (A) and (B) show the charts which appear on edge click for two of the three edges in the selected subgraph. Each chart shows conditional distributions of `h1_arterial_pco2` values (min/max) for the subpopulations (True/False) for the feature `solid_tumor_with_metastasis`.

6.2 Groceries

The **groceries** data set contains 169 unique items in a grocery store (e.g. “Ham", “Flower Soil / Fertilizer", “Grapes", “Butter"), which we consider our ‘features’, with a “1" or “0" for each record corresponding to whether that item was purchased or not in each of 9,835 transactions [Nasrullah, 2019]. This data set is unique in our examples, in that it only contains a binary response: all features only have 1 of 2 possible answers, and this data set is included primarily to demonstrate that the mutual information feature network technique can also be employed for data sets of homogenous data types, e.g. all discrete or all continuous features, and the resultant networks can be compared to other feature networks such as the association network, as shown in Figure 9.

As described in Section 4.1.3, association networks are commonly used for this type of market basket analysis. In an association network, the marginal and conditional probabilities of each pair of items or features are used to calculate “Lift": the probability of purchasing a specific second item, given that it is known that the first has been purchased [Agrawal et al., 1993]. Some shortfalls of the association network for high-dimensional exploratory analysis are:

- Edges are bidirectional, requiring arrows or other encodings to be appended to the network graph to indicate direction along two edges for each node pair, which can clutter the visual exploration space and lead to confusion or be misinterpreted as indicating causality or chronology.
- “Lift" can only be calculated for one of four possible permutations of a pair of binary features, e.g. *“X and Y were both purchased"*. In other binary contexts, there may be many comparisons of interest e.g. “did *or* didn’t take Drug A *and/or* did *or* didn’t take Drug B".
- Association networks are only for comparing features of a binary data type. There is no opportunity to integrate into the “Lift" algorithm an item for which there are multiple responses, e.g. “Blueberry" *or* “Raspberry" *or*

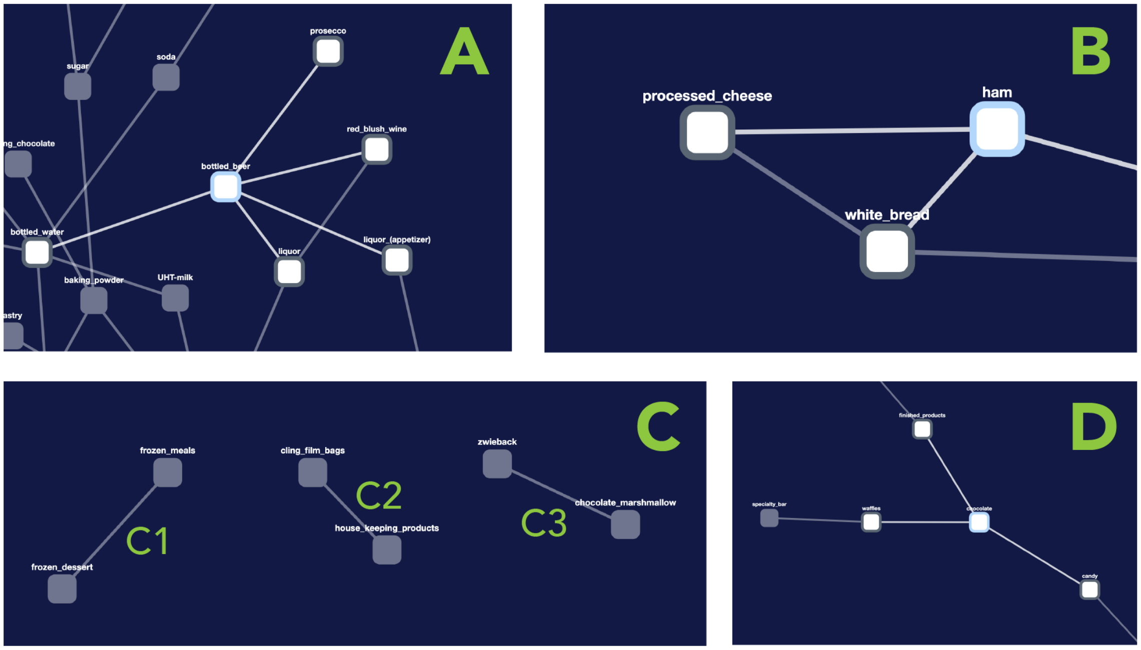


Figure 8: Annotated screenshots from Sirius of portions of the groceries data set show feature groupings related to product types. In (A) we see 'bottled beer' connected to other alcohol-related products: prosecco, red blush wine, liquor (appetizer) and liquor. In (B) we see the makings of a ham sandwich: processed cheese, ham, and white bread. Feature groups in (C) include (C1): Frozen meals and frozen dessert; (C2): cling film bags and housekeeping products; and (C3) zwieback (a kind of cracker that originated in Eastern Europe) connected to 'chocolate marshmallow' which may correspond to Krembo–two items that might be found in the Kosher foods section of a grocery store. Screenshot (D) shows 'chocolate' connected to finished products (i.e. snacks), candy, and waffles. These kinds of feature groupings could inform an analyst about product placement or marketing campaigns.

> “Did not purchase", nor is there a prescribed way to include quantitative information, (e.g. “Milk is bought commonly in transactions exceeding $30 or visits in excess of 15 minutes".) If this data was available, the continuous features “transaction cost" or “visit duration" could be included as nodes in the mutual information network graph, but would have to be converted to a binary encoding in an association network.

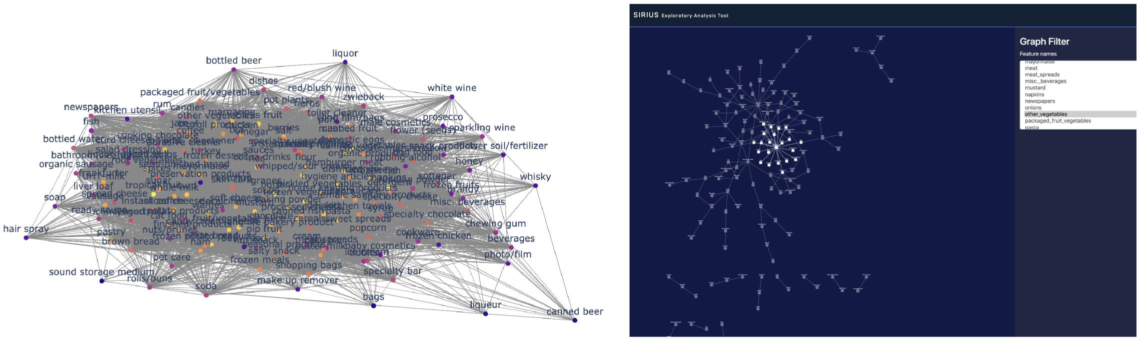
A comparison of the Association Network and the mutual information feature network of the **groceries** data set are shown in Figure 9.

7 Discussion

There are a number of ways in which we see this novel approach as extensible through the addition of modules, and more complex analysis of the mutual information feature network itself:

7.1 Charts

We have selected three chart types (heatmap, ridgeline density, and 2D kernel density) for each of the three permutations of data type pairing supported by Sirius. However, other visual encodings could be investigated. For example, grouped bar charts may be a more intuitive alternative to heatmaps for some users when comparing two discrete features; boxplots, violin plots, or bean plots are valid alternative visual encodings for pairings of discrete and continuous data types. These alternatives could be provided as parameter settings or toggles in future iterations of user interface design.



Association Network
Lift thresheld at static value,
such that 50% of edges remain

Sirius Mutual Information Network
Mutual information thresheld dynamically
using alpha value from backbone sparsification

Figure 9: A side-by-side comparison of (left) an association network of the groceries data set, thresheld at a static edge weight such that 50% of edges remain; and (right) the Sirius mutual information network of the same data set, with dynamic thresholding using the alpha values from the network backbone sparsification step of the data processing. The right panel menu allows rapid user search to find features of interest, with selected node(s) connections highlighted in the graph.

7.2 Advanced Network Analysis Measures

We are careful here not to compare feature networks to real-world networks (i.e. social networks, ecological systems, power grids). The network visualization is used simply as a relational graph. It remains to be seen whether the mutual information feature network (or association networks or gene correlation networks) are subject to the same properties as real-world networks, e.g. the ‘friendship paradox’ (in which the average node is connected to nodes with a higher average number of connections), or eligible for traditional network study such as percolation theory or community detection [Feld, 1991]. Meta-analysis of the properties of various mutual information feature networks could yield interesting insight into organic feature relationship behavior, as compared to null models of synthetically generated data. For example, the connectivity or graph spectra of real mutual information networks could be fundamentally different from feature networks of synthesized high-dimensional data, which may help data scientists to more easily identify cases of data fraud or imputation through examination of anomalous feature relationships [Pennisi, 2020].

7.3 Confounding Variables

The mutual information feature network, as with other feature networks used for feature selection and data mining, is concerned only with edges between pairs of features, not multidimensional interactions between more than two features. Feature networks fall short of addressing statistical concerns such as Simpson’s Paradox, in which unidentified tertiary conditions render summary conclusions invalid [Berman et al., 2012]. Future research could involve topological analysis of feature networks to identify confounding variables and generate multivariate visualizations upon interaction with network graph faces (contained by 3 or more vertices) through the use of simplicial complexes [Jonsson, 2008].

7.4 Temporal Evolution

Functionality could also be implemented for temporal animation of mutual information feature networks, illustrating how feature dependence relationships strengthen or weaken over time. For example, cholesterol values may have been more informative for predictive wellness modeling prior to the widespread usage of cholesterol-lowering drugs; square footage of homes may have been more closely associated with sale price before the introduction of luxury ‘tiny homes’ in a neighborhood. Temporal animation of mutual information feature networks might bring the dynamics of changing feature relationships to the forefront in an exploratory analysis.

8 Conclusions

Initial observations of the mutual information feature network show promise for the method as a new technique for exploratory analysis. Parameterization preserves customizability for specific data structures and domain applications. Further statistical measures of resultant network structures offer potential avenues for insight into more complex feature relationships in high-dimensional data analysis.

9 Acknowledgments

This work was supported in part by funding from Massachusetts Mutual Life Insurance Company, through the establishment of the MassMutual Center of Excellence in Complex Systems and Data Science at the University of Vermont. Our deepest thanks to Mark Maier, Adam Fox, and Sears Merritt at MassMutual Data Science; and Juniper Lovato, Melissa Rubinchuk, and Alexa Woodward at the University of Vermont; for coordinating and supporting this institutional research partnership. Thanks also to David Rushing Dewhurst for aiding in our understanding of mutual information and kernel density estimation.

The authors are grateful to the anonymous reviewers of EuroVis 2020 short papers; TVCG / IEEE VIS 2020; and ACM IUI 2020; for their thorough and thoughtful feedback, which has been incorporated into this pre-print where possible under project constraints.

References

Sindhu P Menon and Nagaratna P Hegde. A survey of tools and applications in big data. In *2015 IEEE 9th International Conference on Intelligent Systems and Control (ISCO)*, pages 1–7, January 2015a. doi:10.1109/ISCO.2015.7282364.

Rakesh Agrawal, Tomasz Imieliundefinedski, and Arun Swami. Mining association rules between sets of items in large databases. In *Proceedings of the 1993 ACM SIGMOD International Conference on Management of Data*, pages 207–216, New York, NY, USA, 1993. Association for Computing Machinery. ISBN 0897915925. doi:10.1145/170035.170072. URL `https://doi.org/10.1145/170035.170072`.

Troy Raeder and Nitesh V. Chawla. Market basket analysis with networks. *Social Network Analysis and Mining*, 1 (2):97–113, April 2011. ISSN 1869-5469. doi:10.1007/s13278-010-0003-7. URL `https://doi.org/10.1007/s13278-010-0003-7`.

Alexander Kraskov, Harald Stögbauer, and Peter Grassberger. Estimating mutual information. *Physical Review E*, 69 (6), Jun 2004. ISSN 1550-2376. doi:10.1103/physreve.69.066138.

Brian C Ross. Mutual information between discrete and continuous data sets. *PLoS One*, 9(2):e87357, 2014. ISSN 1932-6203 (Electronic); 1932-6203 (Linking). doi:10.1371/journal.pone.0087357.

M. Angeles Serrano, Marian Boguna, and Alessandro Vespignani. Extracting the multiscale backbone of complex weighted networks. *Proceedings of the National Academy of Sciences*, 106(16):6483–6488, 2009. ISSN 0027-8424. doi:10.1073/pnas.0808904106. URL `https://www.pnas.org/content/106/16/6483`.

Jane L. Adams, Todd F. Deluca, Yuhan Zheng, Konstantinos Anastasakis, Boyoon Choi, and Allison Min. Sirius: Exploratory analysis tool. URL `github.com/compstorylab/sirius`.

Michael Griebel and Markus Holtz. Dimension-wise integration of high-dimensional functions with applications to finance. *Journal of Complexity*, 26(5):455 – 489, 2010. ISSN 0885-064X. doi:https://doi.org/10.1016/j.jco.2010.06.001. URL `http://www.sciencedirect.com/science/article/pii/S0885064X10000452`.

Gary J Russell and Ann Petersen. Analysis of cross category dependence in market basket selection. *Journal of Retailing*, 76(3):367–392, July 2000. ISSN 0022-4359. doi:10.1016/S0022-4359(00)00030-0. URL `http://www.sciencedirect.com/science/article/pii/S0022435900000300`.

Peter Langfelder and Steve Horvath. Wgcna: an r package for weighted correlation network analysis. *BMC Bioinformatics*, 9(1):559, Dec 2008. ISSN 1471-2105. doi:10.1186/1471-2105-9-559.

Eun Ju Nam, Yiping Han, Klaus Mueller, Alla Zelenyuk, and Dan Imre. ClusterSculptor: A Visual Analytics Tool for High-Dimensional Data. In *2007 IEEE Symposium on Visual Analytics Science and Technology*, pages 75–82, October 2007. doi:10.1109/VAST.2007.4388999.

Saumyadipta Pyne, Xinli Hu, Kui Wang, Elizabeth Rossin, Tsung-I. Lin, Lisa M. Maier, Clare Baecher-Allan, Geoffrey J. McLachlan, Pablo Tamayo, David A. Hafler, Philip L. De Jager, and Jill P. Mesirov. Automated high-dimensional flow cytometric data analysis. *Proceedings of the National Academy of Sciences*, 106(21):8519–8524, May 2009. ISSN 0027-8424, 1091-6490. doi:10.1073/pnas.0903028106. URL `https://www.pnas.org/content/106/21/8519`.

Svante Wold, C. Albano, W. J. Dunn, U. Edlund, K. Esbensen, P. Geladi, S. Hellberg, E. Johansson, W. Lindberg, and M. Sjöström. Multivariate Data Analysis in Chemistry. In Bruce R. Kowalski, editor, *Chemometrics: Mathematics and Statistics in Chemistry*, NATO ASI Series, pages 17–95. Springer Netherlands, Dordrecht, 1984. ISBN 978-94-017-1026-8. doi:10.1007/978-94-017-1026-8_2. URL `https://doi.org/10.1007/978-94-017-1026-8_2`.

Yapeng Su, Qihui Shi, and Wei Wei. Single cell proteomics in biomedicine: High-dimensional data acquisition, visualization, and analysis. *PROTEOMICS*, 17(3-4):1600267, 2017. ISSN 1615-9861. doi:10.1002/pmic.201600267. URL `https://onlinelibrary.wiley.com/doi/abs/10.1002/pmic.201600267`.

John F. McCarthy, Kenneth A. Marx, Patrick E. Hoffman, Alexander G. Gee, Philip O'Neil, M. L. Ujwal, and John Hotchkiss. Applications of Machine Learning and High-Dimensional Visualization in Cancer Detection, Diagnosis, and Management. *Annals of the New York Academy of Sciences*, 1020(1):239–262, 2004. ISSN 1749-6632. doi:10.1196/annals.1310.020. URL `https://nyaspubs.onlinelibrary.wiley.com/doi/abs/10.1196/annals.1310.020`.

David I. Warton. Regularized Sandwich Estimators for Analysis of High-Dimensional Data Using Generalized Estimating Equations. *Biometrics*, 67(1):116–123, 2011. ISSN 1541-0420. doi:10.1111/j.1541-0420.2010.01438.x. URL `https://onlinelibrary.wiley.com/doi/abs/10.1111/j.1541-0420.2010.01438.x`.

Xiaoming Gao, Emilio Ferrara, and Judy Qiu. Parallel Clustering of High-Dimensional Social Media Data Streams. In *2015 15th IEEE/ACM International Symposium on Cluster, Cloud and Grid Computing*, pages 323–332, May 2015. doi:10.1109/CCGrid.2015.19.

David Gotz and Harry Stavropoulos. DecisionFlow: Visual Analytics for High-Dimensional Temporal Event Sequence Data. *IEEE Transactions on Visualization and Computer Graphics*, 20(12):1783–1792, December 2014. ISSN 1941-0506. doi:10.1109/TVCG.2014.2346682.

David I Shuman, Sunil K. Narang, Pascal Frossard, Antonio Ortega, and Pierre Vandergheynst. The emerging field of signal processing on graphs: Extending high-dimensional data analysis to networks and other irregular domains. *IEEE Signal Processing Magazine*, 30(3):83–98, May 2013. ISSN 1558-0792. doi:10.1109/MSP.2012.2235192.

Sindhu P Menon and Nagaratna P Hegde. A survey of tools and applications in big data. In *2015 IEEE 9th International Conference on Intelligent Systems and Control (ISCO)*, pages 1–7, January 2015b. doi:10.1109/ISCO.2015.7282364.

Omar Y. Al-Jarrah, Paul D. Yoo, Sami Muhaidat, George K. Karagiannidis, and Kamal Taha. Efficient Machine Learning for Big Data: A Review. *Big Data Research*, 2(3):87–93, September 2015. ISSN 2214-5796. doi:10.1016/j.bdr.2015.04.001. URL `http://www.sciencedirect.com/science/article/pii/S2214579615000271`.

Adil Fahad, Najlaa Alshatri, Zahir Tari, Abdullah Alamri, Ibrahim Khalil, Albert Y. Zomaya, Sebti Foufou, and Abdelaziz Bouras. A Survey of Clustering Algorithms for Big Data: Taxonomy and Empirical Analysis. *IEEE Transactions on Emerging Topics in Computing*, 2(3):267–279, September 2014. ISSN 2168-6750. doi:10.1109/TETC.2014.2330519.

Jonathan Stuart Ward and Adam Barker. Undefined By Data: A Survey of Big Data Definitions. *arXiv:1309.5821 [cs]*, September 2013. URL `http://arxiv.org/abs/1309.5821`.

Min Chen, Shiwen Mao, and Yunhao Liu. Big Data: A Survey. *Mobile Networks and Applications*, 19(2): 171–209, April 2014. ISSN 1572-8153. doi:10.1007/s11036-013-0489-0. URL `https://doi.org/10.1007/s11036-013-0489-0`.

Muhammad Bilal Zafar, Isabel Valera, Manuel Gomez Rodriguez, and Krishna P. Gummadi. Fairness Beyond Disparate Treatment & Disparate Impact: Learning Classification without Disparate Mistreatment. In *Proceedings of the 26th International Conference on World Wide Web*, WWW '17, pages 1171–1180, Perth, Australia, April 2017. International World Wide Web Conferences Steering Committee. ISBN 978-1-4503-4913-0. doi:10.1145/3038912.3052660. URL `https://doi.org/10.1145/3038912.3052660`.

Michael Feldman, Sorelle A. Friedler, John Moeller, Carlos Scheidegger, and Suresh Venkatasubramanian. Certifying and Removing Disparate Impact. In *Proceedings of the 21th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, KDD '15, pages 259–268, Sydney, NSW, Australia, August 2015. Association for Computing Machinery. ISBN 978-1-4503-3664-2. doi:10.1145/2783258.2783311. URL `https://doi.org/10.1145/2783258.2783311`.

On the Direction of Discrimination: An Information-Theoretic Analysis of Disparate Impact in Machine Learning. pages 126–130, June 2018. doi:10.1109/ISIT.2018.8437661.

Jennifer L. Skeem and Christopher T. Lowenkamp. Risk, Race, and Recidivism: Predictive Bias and Disparate Impact*. *Criminology*, 54(4):680–712, 2016. ISSN 1745-9125. doi:10.1111/1745-9125.12123. URL `https://onlinelibrary.wiley.com/doi/abs/10.1111/1745-9125.12123`.

John T. Behrens. Principles and procedures of exploratory data analysis. *Psychological Methods*, 2(2):131–160, 1997. ISSN 1939-1463(Electronic),1082-989X(Print). doi:10.1037/1082-989X.2.2.131.

E. Bertini, A. Tatu, and D. Keim. Quality metrics in high-dimensional data visualization: An overview and systematization. *IEEE Transactions on Visualization and Computer Graphics*, 17(12):2203–2212, Dec 2011. ISSN 2160-9306. doi:10.1109/TVCG.2011.229.

A. MacEachren, D. Xiping, F. Hardisty, Diansheng Guo, and G. Lengerich. Exploring high-D spaces with multiform matrices and small multiples. In *IEEE Symposium on Information Visualization 2003 (IEEE Cat. No.03TH8714)*, pages 31–38, October 2003. doi:10.1109/INFVIS.2003.1249006.

Jinwook Seo and Ben Shneiderman. A Rank-by-Feature Framework for Interactive Exploration of Multidimensional Data. *Information Visualization*, 4(2):96–113, June 2005. ISSN 1473-8716. doi:10.1057/palgrave.ivs.9500091. URL `https://doi.org/10.1057/palgrave.ivs.9500091`. Publisher: SAGE Publications.

Magdalena Niemira, FranÃ§ois Collin, Szalkowska, Bielska, Karolina Chwialkowska, Reszec, Niklinski, Kwasniewski, and Kretowski. Molecular Signature of Subtypes of Non-Small-Cell Lung Cancer by Large-Scale Transcriptional Profiling: Identification of Key Modules and Genes by Weighted Gene Co-Expression Network Analysis (WGCNA). *Cancers*, 12:37, December 2019. doi:10.3390/cancers12010037.

Kamada Tomihisa. *Visualizing Abstract Objects And Relations*. World Scientific, 1989. ISBN 978-981-4518-86-4.

James P. Bagrow, Sune Lehmann, and Yong-Yeol Ahn. Robustness and modular structure in networks. *Network Science*, 3(4):509–525, Jul 2015. ISSN 2050-1250. doi:10.1017/nws.2015.21.

Wes McKinney. Data structures for statistical computing in python. In Stéfan van der Walt and Jarrod Millman, editors, *Proceedings of the 9th Python in Science Conference*, pages 51 – 56, 2010.

F. Pedregosa, G. Varoquaux, A. Gramfort, V. Michel, B. Thirion, O. Grisel, M. Blondel, P. Prettenhofer, R. Weiss, V. Dubourg, J. Vanderplas, A. Passos, D. Cournapeau, M. Brucher, M. Perrot, and E. Duchesnay. Scikit-learn: Machine learning in Python. *Journal of Machine Learning Research*, 12:2825–2830, 2011.

Erik Gregersen. Sirius | Facts & Location. URL `https://www.britannica.com/place/Sirius-star`.

NetworkX developer team. Networkx, 2014. URL `https://networkx.github.io/`.

Thomas M. J. Fruchterman and Edward M. Reingold. Graph drawing by force-directed placement. *Software: Practice and Experience*, 21(11):1129–1164, 1991. ISSN 1097-024X. doi:10.1002/spe.4380211102. URL `https://onlinelibrary.wiley.com/doi/abs/10.1002/spe.4380211102`.

Bo Chen and Sha Lin. Albumin-bilirubin (ALBI) score at admission predicts possible outcomes in patients with acute-on-chronic liver failure. *Medicine*, 96(24), June 2017. ISSN 0025-7974. doi:10.1097/MD.0000000000007142. URL `https://www.ncbi.nlm.nih.gov/pmc/articles/PMC5478326/`.

Meredith Lee, Jesse Raffa, Marzyeh Ghassemi, Tom Pollard, Sharada Kalanidhi, Omar Badawi, Karen Matthys, and Leo Anthony Celi Celi. Wids (women in data science) datathon 2020: Icu mortality prediction. PhysioBank, PhysioToolkit, and PhysioNet: Components of a New Research Resource for Complex Physiologic Signals, 2020. doi:10.13026/vc0e-th79.

Irfan Nasrullah. Groceries Market Basket Dataset, 2019. URL `https://kaggle.com/irfanasrullah/groceries`.

Dean De Cock. House Prices: Advanced Regression Techniques, 2019. URL `https://kaggle.com/c/house-prices-advanced-regression-techniques`.

Scott L. Feld. Why your friends have more friends than you do. *American Journal of Sociology*, 96(6):1464–1477, 1991. doi:10.1086/229693.

Elizabeth Pennisi. Spider biologist denies suspicions of widespread data fraud in his animal personality research, 2020. URL `https://www.sciencemag.org/news/2020/01/spider-biologist-denies-suspicions-widespread-data-fraud-his-animal-personality`.

Steve Berman, Leandro DalleMule, Michael Greene, and John Lucker. Simpson's paradox: a cautionary tale in advanced analytics. *Significance*, 2012.

Jakob Jonsson. *Simplicial complexes of graphs*. Springer, Berlin New York, 2008. ISBN 978-3-540-75858-7.