Hypothesis and analysis on the relationship between bivariate data correlation and Mapper graph structure

Phillip Korolev

January 2026

1 Introduction

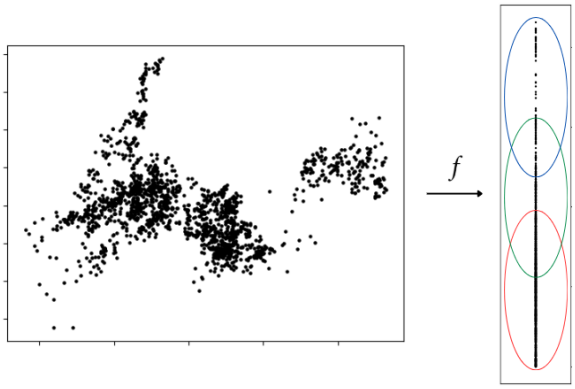
When tasked with analyzing data, we are often looking to draw intuitive, visual patterns. With bivariate time series data, our task is much simpler and typically narrows our available options for data analysis techniques. In this case, the goal is to draw some sort of pattern between our two observed values $X(t)$, $Y(t)$ over a portion or the entirety of our data set's time span T. Although a $2-$dimensional scatter plot is often our first step to visualize some sort of correlation between the values, it can be difficult to tell just how populated clusters of points are, or to draw conclusions given unintuitive plot shapes.

With methods in topological data analysis, we are granted techniques that let us explore the structure and correlation of our data past the initial scatter plot. In this informal analysis, we will introduce the Mapper algorithm, which transforms arbitrary dimensional data into a flat discrete graph with nodes, edges, and coloring, helping us identify things like cluster density and potential cycles in our time series data. In particular, we will analyze the structure of graphs generated from highly correlated and uncorrelated bivariate time series data sets.

2 Mapper Algorithm

Many more detailed versions of the algorithm and motivations can be found in outside resources. The general summary of the algorithm will be presented, as well as a more formal process.

Simply, the Mapper algorithm uses filtration processes (of our choice) to project data $X \subset \mathbb{R}^d$ onto $\mathbb{R}^m$ with $m < d$. The filtration function is often called the lens, and typically we choose a function that projects our data onto $\mathbb{R}^2$ or $\mathbb{R}$. Based on given parameters, we look to construct overlapping covers of the lens space $f(X)$ to split and group the projected image points into their respective regime.



The colored portions of our lens space $f(X)$ above represent a different cover (interval) of the space and help us classify our original points $f^{-1}(X)$. After classifying the original points and applying some clustering algorithm, we are able to construct a graph where each node represents some cluster, and an edge is only present if there are overlapping points between two clusters. Also note that the size of a particular node depends on the respective cluster's density (population).

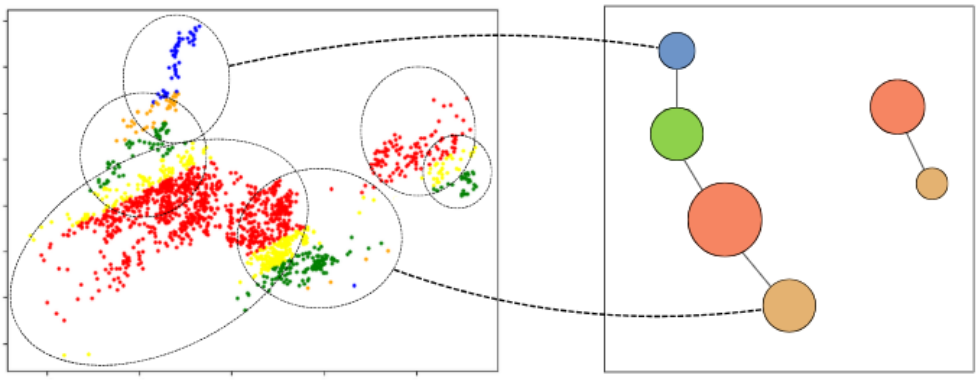


Figure 1: Clustering classified data points and generating graph

Generally, the process aims to accomplish the following:

1. Normalize our data set $X \subset \mathbb{R}^d$ using a normalization function of choice Norm : $X \to \tilde{X}$
2. Choose a filtration (lens) function $f : \mathbb{R}^d \to \mathbb{R}^m$, typically with $m \in \{1, 2\}$, and build the image set $f(\tilde{X}) = \{f(x) : x \in \tilde{X}\}$
3. In the image space, construct a cover $\mathcal{U}_f = \{U_a\}_{a \in A}$, with A an arbitrary index set, such that:
   (i) their union covers all of the image space: $f(\tilde{X}) \subseteq \bigcup U_a$
   (ii) for $a, b \in A$: U_a, U_b adjacent $\iff U_a \cap U_b \neq \emptyset$
4. Classify points $x \in \tilde{X}$ with class c if $f(x) \in U_c$
5. Apply clustering to our data $\tilde{X}$
6. Construct the graph $G = (V, E)$ with vertices v_i representing clusters C_i and edges e_i connecting two vertices v_i, v_j if and only if there exists a point $x \in \tilde{X}$ such that $x \in C_i \cap C_j$

In the case of the experiments of this paper, the algorithm can be slightly more defined. To start, we are interested in bivariate data. That is, our data set X is $2-$dimensional of the form $X = \{(x_i, y_i)\}$, so the normalization we use is the process $\text{Norm}_Z : X \to \tilde{X}$, defined by:

$$\text{Norm}_Z(x, y) = \left(\frac{x - \mu_x}{\sigma_x}, \frac{y - \mu_y}{\sigma_y} \right),$$

with μ_x and σ_x the mean and standard deviation with respect to the x argument in X. Likewise for μ_y, σ_y.

With X as described above, we are restricted to using filtration functions in $\mathcal{F} = \{f \mid f : \mathbb{R}^2 \to \mathbb{R}^2 \cup f : \mathbb{R}^2 \to \mathbb{R}\}$. In the experiments seen in further sections, we use the filtration function $f : \mathbb{R}^2 \to \mathbb{R}$ defined by

$$f(x, y) = |x - y|,$$

to stay on the theme of correlation. Given that $\tilde{X}$ is the normalized data set, the function f above gives us a good idea of how far apart the values $\tilde{x}_i$ and $\tilde{y}_i$ are. For some intuition, a perfectly correlated normalized dataset $\tilde{X}$ would lie on the line $y = x$ in $\mathbb{R}^2$. In our case, the image space $f(\tilde{X})$ lies on the real number line, so our collection of overlapping covers is just a collection of overlapping intervals $I_1, I_2, ..., I_N$ satisfying:

$$\bigcup_{i=1}^{N} I_i = [\min f(\tilde{X}), \max f(\tilde{X})].$$

To achieve some overlap, we set an overlap parameter $\alpha \in (0, 1)$ that tells the interval construction algorithm the amount of overlap that should occur between each consecutive interval. Note that for any two consecutive intervals I_k, I_{k+1} we have that $I_k \cap I_{k+1} = [\min I_{k+1}, \max I_k]$ and with guaranteed overlap, this intersection is never empty.

Example. Interval Construction
Take the interval $\mathcal{I} = [0, 10]$, we build $N = 3$ equally wide intervals with overlap $\alpha = 0.5$ such that I_1, I_2, I_3 cover all of $\mathcal{I}$. We want to do so "efficiently", so our first interval I_1 should begin at $\min \mathcal{I} = 0$, and $I_N = I_3$ should have maximum $\max \mathcal{I} = 10$. This algorithm is described below, but for the sake of the example, our covers would be the following:

$$I_1 = [0, 5]; I_2 = [2.5, 7.5]; I_3 = [5, 10],$$

with intersections $I_1 \cap I_2 = [2.5, 5] = [\min I_2, \max I_1]$ and $I_2 \cap I_3 = [5, 7.5] = [\min I_3, \max I_2]$.

The following is the algorithm used to generate graphs and results seen in this paper. As described above, it is tuned for exploring correlation structures in bivariate data.

Algorithm 1: Mapper Construction with Absolute-Difference Filtration

Input: Data set $X \subset \mathbb{R}^2$, number of intervals N, cover overlap parameter α
Output: Nerve graph G
1. $\tilde{X} \leftarrow \text{Norm}_z(X)$
2. Define function $f(x, y) = |x - y|$
3. $Y \leftarrow f(\tilde{X})$
4. $\mathcal{I} \leftarrow [\min f(\tilde{X}), \max f(\tilde{X})]$
5. $\mathcal{U} \leftarrow \text{getCovers}(\mathcal{I}, N, \alpha)$
6. **foreach** $I_i \in \mathcal{U}$ **do**
   - Classify points $c_i \leftarrow \{f(x) \in Y : f(x) \in I_i\}$
   - Pull back $V_i \leftarrow f^{-1}(I_i)$
   - Compute clusters $\mathcal{C}_i \leftarrow \mathcal{C}(V_i)$
7. Construct graph G:
(i) add node N_i representing cluster $\mathcal{C}_i$
(ii) add edge (N_i, N_j) if and only if $\mathcal{C}_i \cap \mathcal{C}_j \neq \emptyset$ for $i \neq j$
return G

For the cover construction process described previously, and seen in Algorithm 1 (**getCovers**), we use the following. Note that the algorithm constructs intervals as we assume that all filtration functions used throughout the experiments in the paper are of the form $f : \mathbb{R}^2 \to \mathbb{R}$.

Algorithm 2: Interval Construction in $\mathbb{R}$

Input: Interval $\mathcal{I} \subset \mathbb{R}$, number of intervals N, overlap parameter α
Output: List of intervals $I_1, \ldots, I_N$ covering $\mathcal{I}$
1. Initialize list L
2. $w \leftarrow (\max \mathcal{I} - \min \mathcal{I})/(N - (N-1)\alpha)$
3. $s \leftarrow w(1-\alpha)$
4. **for** $i \in \{0, 1, \ldots, N-1\}$ **do**
   - $l \leftarrow \min \mathcal{I} + i \cdot s$
   - $u \leftarrow l + w$
   - $L \leftarrow I_i = [l, u]$

return L

3 Hypothesis

We wish to gain an intuition on Mapper graph structures based on the bivariate data set X which it is built from. Conversely, we also want to identify if our original data set X is highly correlated based on the generated Mapper graph. *For the rest of the section, we will assume that X has already been normalized, and in later experiments, we use* Norm_Z *described previously to normalize synthetic data.*

Given a normalized and well-connected bivariate data set $X = \{x_i, y_i\}_{i=1}^N$ with $x = (x_1, x_2, ..., x_n)$, $y = (y_1, y_2, ..., y_n)$, the lens function $f : \mathbb{R}^2 \to \mathbb{R}$ defined by $f(x, y) = |x - y|$, fixed interval construction parameters (N, α), and some fixed clustering algorithm, we expect the resulting Mapper graph to contain the following properties:

1. As $|\rho(x, y)| \to 1$, the likeliness of resulting graph G being fully connected $\mathbb{P}[\beta_0(G) = 1]$ increases, and the expected number of independent cycles $\mathbb{E}[\beta_1(G)]$ tends to 0.

2. As $|\rho(x, y)| \to 0$, the expected number of cycles $\mathbb{E}[\beta_1(G)]$ increases.

Note that the Betti numbers β_0, β_1 track connectedness and cycles (loops) of a graph respectively. In the case that $\beta_0 = 1$, the graph is fully connected, i.e.: there always exists a path from node v_i to node v_j. On the other hand, $\beta_0 > 0$ signals that the graph has multiple disconnected components. In the same manner, $\beta_1 = 0$ tells us that there are no cycles in our graph, while $\beta_1 > 0$ signals the existence of one or more cycles.

To understand the intuition behind the statement in the first expected result, we assume that the series x, y of data set X are highly correlated. From a regression standpoint, the regression line plotted against the plot of X fits our data almost perfectly ($R^2 \approx 1$). With appropriate normalization, the points in X lie almost exactly on the line $y = x$, so we can write each point $(x, y) \in X$ as $(x, x + \varepsilon)$ for some small value $\varepsilon \in \mathbb{R}$. The lens function given above captures the distance of any point in X from the $1-$dimensional diagonal manifold $y = x$. With each point $(x, y) = (x, x + \varepsilon)$, the image space $f(X)$ is contained in the interval $[0, \varepsilon] \subset \mathbb{R}$. Since points $f(x, y) \in f(X)$ are so closely connected, any interval cover $\mathcal{I}_a$ of $f(X)$ will satisfy $I_k \cap I_{k+1} \neq \emptyset$ since we guarantee some overlap α, which causes a chain of overlaps and consequently, each pullback $f^{-1}(I_k)$ is connected. Thus, for a highly correlated (by series) bivariate data set X, we expect a Mapper graph with one globally connected component:

$$\rho(x, y) \to 1 \implies \mathbb{E}[\beta_0(G)] \to 1$$

Staying on the first expected result, we explore the second half of the statement similarly. To obtain a value $\beta_1 > 0$, we must have some cycle in our cluster chain:

$$C_k \longleftrightarrow C_{k+1} \longleftrightarrow C_{k+2} \longleftrightarrow C_k.$$

Again, we have the image space $f(X) \subset [0, \varepsilon]$ with each pullback $f^{-1}(I_K)$ a diagonal band of thickness proportional to $|I_K|$. With the data tube thickness ε much smaller than the band width, each band intersects the data in a single connected strip. Moreover, that strip only overlaps with strips from adjacent intervals, thus we obtain the cluster chain below, which is a tree and thus $\beta_1 = 0$.

$$C_1 \longleftrightarrow C_2 \longleftrightarrow C_3 \longleftrightarrow \cdots$$

To make sense of the second expected result, we start out by taking a bivariate data set X with weak correlation between the series x and y. Thinking of any arbitrary heavily-uncorrelated scatter on $\mathbb{R}^2$, we usually get irregular shapes mixed with (or completely made up of) random noise. As correlation weakens, the data ceases to concentrate near the diagonal $y = x$ and beings occupying some genuine $2-$dimensional region of $\mathbb{R}^2$.

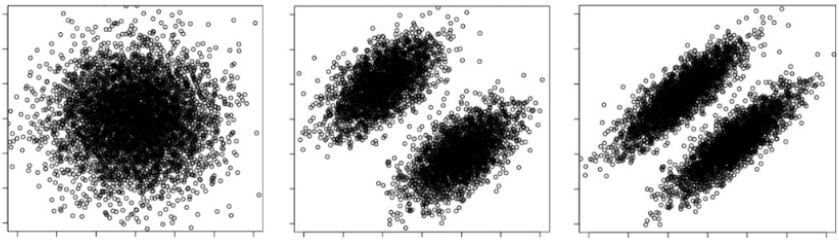


Figure 2: Examples of uncorrelated bivariate data scatters

When $|\rho|$ is small, our lens values $|x - y|$ carry larger variance, so the image $f(X)$ spreads over a greater interval $[0, M] \subset \mathbb{R}$. In the previous case, we deduced that pullbacks $f^{-1}(I_k)$ are bands that intersect our cloud of points in something which is essentially $1-$dimensional (diagonal manifold $y = x$), and overlaps happen mostly between adjacent bands. However, with a low ρ value, each band intersects the point cloud in a wider region, and regions from several neighboring bands can intersect through multiple pathways. In a high-correlation scatter, our thin cloud of points along $y = x$ are such that the intersection between a class of points (band) $U_k = f^{-1}(I_k)$ and the whole cloud is a shape of roughly $1-$dimension:

$$\dim(U_k \cap X) \approx 1.$$

Whereas, with a thick cloud of points, the space $U_k \cap X$ is $2-$dimensional, thus bridges between bands U_i, U_j are no longer limited to (essentially) $1-$dimensional space, but can exist in $2-$dimensions. This increases the chance of 3 or more bands U_k, U_{k+1}, U_{k+2} to be transitively connected, causing a cycle.

Also, for any graph G,

$$\beta_1 = |E| - |V| + \beta_0.$$

Under the assumption that our data X has connected-support and $\beta_0 \approx 1$ stays stable, the driver of the value β_1 is essentially $|E| - |V|$. In the case of a tree (no cycles) with $|V|$ nodes, we have exactly $|V| - 1$ edges, which results in $\beta_1 = 0$, as expected for a tree. On the other hand, any extra edges created by nontrivial overlaps of bands push the number of edges $|E|$ in the graph past $|V| - 1$, leading to a β_1 value greater than 0 (cycle exists). With $|\rho| \to 0$, the thickening of the cloud of points increases the probability of such extra overlap between bands as previously explained, so:

$$\mathbb{E}[|E| - (|V| - 1)] \text{ increases } \implies \mathbb{E}[\beta_1] \text{ increases}$$

4 Experimental Design

We test the hypothesis that, under the absolute-difference lens $f(x, y) = |x - y|$, the Mapper nerve G_ρ simplifies as $|\rho| \to 1$ (with $\beta_0 \to 1$ and $\beta_1 \to 0$ under connected diagonal support), and that for noise-like bivariate clouds $\mathbb{E}[\beta_1]$ increases as $|\rho| \downarrow 0$. To isolate the effect of correlation, we fix all Mapper hyperparameters across runs and vary only the data-generating correlation parameter.

Family A (linear, correlation-controlled; primary). For each target $\rho \in [-1, 1]$, generate n i.i.d. samples

$$(x, y) \sim \mathcal{N}\left(0, \begin{pmatrix} 1 & \rho \\ \rho & 1 \end{pmatrix}\right).$$

This family ensures that decreasing $|\rho|$ corresponds to a thicker, more two-dimensional point cloud.

Control 1 (nonlinear dependence with low Pearson correlation). Generate $x \sim \text{Unif}[-1, 1]$ and $y = x^2 + \sigma\varepsilon$ with $\varepsilon \sim \mathcal{N}(0, 1)$. With symmetric x, Pearson correlation may be near 0 despite strong dependence; this control demonstrates that low $|\rho|$ alone does not imply large β_1.

Simulation. Let $\mathcal{R} = \{0, \Delta\rho, 2\Delta\rho, \ldots, 1\}$ be a grid over $|\rho|$ (e.g. $\Delta\rho = 0.05$). For each $r \in \mathcal{R}$:

1. For $j = 1, \ldots, R$ independent trials, generate a dataset $X_r^{(j)}$ from Family A with target correlation $\rho = \pm r$ (optionally run both signs).

2. Construct the Mapper graph $G_r^{(j)}$ and record the graph invariants

$$\beta_0\left(G_r^{(j)}\right), \quad \beta_1\left(G_r^{(j)}\right), \quad |V_r^{(j)}|, \quad |E_r^{(j)}|.$$

3. Compute summary statistics:

$$\overline{\beta_k}(r) = \frac{1}{R}\sum_{j=1}^{R} \beta_k\left(G_r^{(j)}\right), \quad s_{\beta_k}(r) = \sqrt{\frac{1}{R-1}\sum_{j=1}^{R}\left(\beta_k\left(G_r^{(j)}\right) - \overline{\beta_k}(r)\right)^2},$$

for $k \in \{0, 1\}$ (and similarly for $|V|, |E|$).

Parameter Summary (Example Defaults)

Quantity	Default / Example Choice
Sample size	$n \in \{500, 1000\}$
Correlation grid	$\lvert\rho\rvert \in \{0, 0.05, \ldots, 0.95, 1\}$
Trials per grid point	$R \in \{30, 50, 100\}$
Normalization	coordinate-wise Z-normalization
Lens	$f(x, y) = \lvert x - y\rvert$
Cover intervals	$N \in \{10, 15, 20\}$
Overlap	$\alpha \in \{0.3, 0.5\}$
Clustering	DBSCAN (fixed ε, min_samples)
Outputs	$\beta_0, \beta_1, \lvert V\rvert, \lvert E\rvert$ (and realized $\lvert\hat{\rho}\rvert$)

Table 1: Example experimental settings