Landscapes of Crashes: A Topological Analysis of Financial Time Series

Giuseppe Nicola Liso

Geometry and Topology for Data Analysis
University of Trento

July 4, 2025

Presentation Outline

The Challenge: Predicting Financial Crises

- Financial markets are complex dynamic systems.
- Key challenges in their analysis:
  - **Non-stationarity**: Statistical properties change over time.
  - **High Dimensionality & Noise**: Many intertwined series, often with significant noise.
  - **Non-linearity**: Relationships are rarely simple or linear.
  - **Fat Tails**: Extreme events are more common than predicted by normal distributions.
  - **Implicit Interdependencies**: Complex, evolving correlations between assets.
- Traditional models often make simplifying assumptions (e.g., linearity, stationarity) that limit their effectiveness during periods of stress.

A New Perspective: Is There a "Shape" of a Crash?

One of the most critical goals in financial analysis is to identify **precursors to market crises or crashes**.

Traditional Approach:

- Model the underlying stochastic process.
- Estimate parameters (volatility, correlation).
- Often fails to capture systemic shifts.

Our Guiding Question:

- Can we analyze the **shape** or **geometry** of market data directly?
- Do periods preceding a crash exhibit a distinct geometric structure compared to "calm" periods?

The Proposed Tool: Topological Data Analysis (TDA)

- TDA is a field that studies the "shape" of data using concepts from topology.
- **Why TDA for Finance?**
  - **Robust to Noise:** Topological features are stable under small perturbations of the data.
  - **Coordinate-Free:** Invariant under transformations like scaling and rotation.
  - **Model-Free:** Does not require an a priori stochastic model of the data generation process.
- **The question**: Can TDA reveal hidden structural changes in financial data that act as early warning signals for crashes?

Core Idea

Extract robust, qualitative shape information from high-dimensional and noisy datasets.

Objectives & Agenda

Our Main Objectives:

1. Transform financial time series into a sequence of geometric objects (point clouds).
2. Quantify the "topological complexity" of these objects over time.
3. Test whether this quantitative measure can act as an early-warning signal for market crashes.

Presentation Agenda:

- The Topological Toolkit
- The Proposed Method
- Validation on Synthetic Data
- Empirical Analysis of Financial Data

The Core Task: Describing the Shape of a Point Cloud

- Our input is a finite set of points, a **point cloud** $X = \{x_1, \ldots, x_n\} \subset \mathbb{R}^d$.
- **Question:** How can we capture its essential features as connected components or loops in a quantitative and stable manner?
- **Answer:** Persistent Homology.

Simplices and Boundaries: A Formal View

Definition: k-Simplex

A **k-simplex**, $[v_0, \ldots, v_k]$, is the **convex hull** of $k+1$ affinely independent points. It is the set of all convex combinations of its vertices:

$$[v_0, \ldots, v_k] = \left\{ \sum_{i=0}^{k} \lambda_i v_i \mid \sum_{i=0}^{k} \lambda_i = 1,\ \lambda_i \geq 0 \right\}$$

Definition: Face and Boundary

A **face** of a simplex is the convex hull of any non-empty subset of its vertices. The **boundary**, $\partial\sigma$, is the union of all its faces except for the simplex itself (its "proper faces").

The Simplicial Complex

A **simplicial complex** K is a collection of simplices such that if a simplex σ is in K, all of its faces must also be in K.

Tool 1: The Vietoris-Rips (VR) Filtration

We build a sequence of simplicial complexes on the point cloud by varying a scale parameter, ε.

- **Intuition:** We "thicken" the points by growing balls of radius $\varepsilon/2$ around them and record the intersections.

Definition: The Vietoris-Rips Complex, $R(X, \varepsilon)$

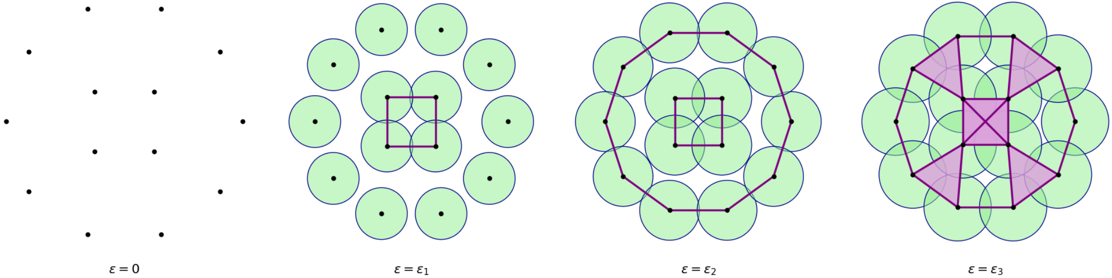
A k-simplex $\{x_{i_0}, \ldots, x_{i_k}\}$ is included in the VR complex $R(X, \varepsilon)$ if all pairwise distances between its vertices are less than ε.

$$d(x_{i_j}, x_{i_l}) < \varepsilon \quad \forall j, l \in \{0, \ldots, k\}$$

The **filtration** is the nested sequence of these complexes generated by continuously increasing the scale parameter ε:

$$R(X, \varepsilon_1) \subseteq R(X, \varepsilon_2) \quad \text{for} \quad \varepsilon_1 < \varepsilon_2$$

Visualizing a Filtration: The Process



We analyze the data at all scales by "growing" balls around each point. This creates a nested sequence of shapes, called a **filtration**, which allows us to see how topological features like loops are born and die.

Visualizing a Filtration: A Step-by-Step Analysis

Let's break down the evolution of the topology shown in the previous slide:

- At $\varepsilon = 0$: The Point Cloud: We begin with 12 distinct points. The homology is simple: 12 connected components ($\beta_0 = 12$) and no loops ($\beta_1 = 0$).
- At $\varepsilon = \varepsilon_1$ and ε_2: Birth of Loops: As ε increases, an inner loop is **born** (at ε_1), followed by an outer loop (at ε_2). At the ε_2 stage, two independent loops coexist ($\beta_1 = 2$).
- At $\varepsilon = \varepsilon_3$: Death of a Loop: The inner loop gets filled in by triangles. It becomes a boundary and therefore **dies**. The outer, more robust loop persists. The loop count drops back to $\beta_1 = 1$.

Looking at the Persistence Diagram

This process generates two points in the H_1 persistence diagram: a point with lower persistence for the inner loop, and a point with higher persistence for the outer loop.

Tool 2: Homology Group

- For each complex $R(X, \varepsilon)$, we compute its homology groups $H_k(R(X, \varepsilon))$.
- **Intuitive Meaning:**
  - H_0: Counts the number of connected components.
  - H_1: Counts the number of independent loops or cycles (2D "holes").
  - H_2: Counts the number of independent voids or cavities (3D "holes").
- By tracking these groups across the filtration, we can identify when features are **born** and when they **die**.

Focus of this Work

We concentrate on H_1, as loops can represent cyclic behaviors or complex correlation structures in financial markets.

To compute "holes," we first translate the geometry of a simplicial complex K into an algebraic structure. We use coefficients in a field, typically $\mathbb{Z}_2 = \{0, 1\}$.

Definition 1: The Group of k-Chains, $C_k(K; \mathbb{Z}_2)$

A **k-chain** is a formal sum of k-simplices. With $\mathbb{Z}_2$ coefficients, it's simply a **subset** of k-simplices. C_k is the vector space of all k-chains.

- C_0: The space of all subsets of vertices.
- C_1: The space of all subsets of edges.
- C_2: The space of all subsets of triangles.

Definition 2: The Boundary Map, ∂_k

The boundary map $\partial_k : C_k \to C_{k-1}$ is a linear map that computes the boundary of a k-chain. On a single k-simplex $\sigma = [v_0, \ldots, v_k]$, it is defined as:

$$\partial_k([v_0, \ldots, v_k]) = \sum_{i=0}^{k} [v_0, \ldots, \hat{v}_i, \ldots, v_k]$$

(In $\mathbb{Z}_2$, all signs are positive.)

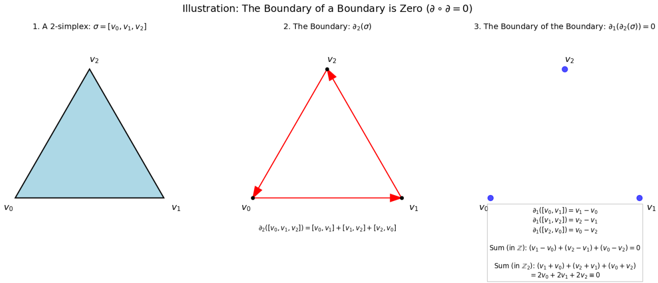
- $\partial_1([v_0, v_1]) = v_0 + v_1$ (The boundary of an edge is its two endpoints).
- $\partial_2([v_0, v_1, v_2]) = [v_0, v_1] + [v_1, v_2] + [v_0, v_2]$ (The boundary of a triangle is its three edges).

A Fundamental Property: The Boundary of a Boundary is Zero

Fundamental Theorem of Homology

For any dimension k, the composition $\partial_{k-1} \circ \partial_k$ is the zero map.

$$\partial_{k-1} \circ \partial_k = 0$$



Intuition: The boundary of a solid shape has no boundary itself. For the triangle above, $\partial_1(\partial_2([v_0, v_1, v_2])) = (v_0 + v_1) + (v_1 + v_2) + (v_0 + v_2) = 2v_0 + 2v_1 + 2v_2 = 0$ in $\mathbb{Z}_2$.

Defining "Holes": Cycles and Boundaries

Using the boundary map, we can now precisely define what a "hole" is.

Definition 3: The Group of k-Cycles, Z_k

The k-cycles are the k-chains **without a boundary**. They form the **kernel** of the boundary map ∂_k:

$$Z_k(K) = \ker(\partial_k) = \{c \in C_k \mid \partial_k(c) = 0\}$$

Definition 4: The Group of k-Boundaries, B_k

The k-boundaries are k-chains that are themselves the **boundary of a (k+1)-chain**. They form the **image** of the map ∂_{k+1}:

$$B_k(K) = \mathrm{im}(\partial_{k+1}) = \{\partial_{k+1}(d) \mid d \in C_{k+1}\}$$

The property $\partial \circ \partial = 0$ implies that every boundary is a cycle ($B_k \subseteq Z_k$).

The Homology Group: Counting the "Real" Holes

Homology measures the cycles that are not just boundaries.

Definition 5: The k-th Homology Group, $H_k(K)$

The k-th homology group is the quotient group of the k-cycles by the k-boundaries:

$$H_k(K) = \frac{Z_k}{B_k} = \frac{\ker(\partial_k)}{\mathrm{im}(\partial_{k+1})}$$

- H_k captures the "true" k-dimensional holes—those cycles that are not filled in by a higher-dimensional simplex.
- Its dimension, the **k-th Betti number** $\beta_k = \dim(H_k)$, counts the number of independent k-dimensional holes.

Persistent Homology tracks how these groups, $H_k(R(X, \varepsilon))$, change as we move through the filtration, identifying when holes are **born** and when they **die**.

Why Focus on $k = 1$ (Loops) in this Study?

The choice to analyze 1-dimensional homology (H_1) is deliberate and driven by three key factors:

- **Financial Interpretation:**
  - H_1 (loops) suggest **quasi-cyclic behavior** or **anomalous correlation structures**. This is a strong sign of a non-random, structured market dynamic, unlike the simpler H_0 or the hard-to-interpret H_2.
- **Signal Sensitivity:**
  - It's possible to show that the "topological signal" of a market crisis is found in H_1. Calm markets are topologically "boring" in this dimension, while stressed markets develop a rich and persistent H_1 topology.
- **Computational Feasibility:**
  - The complexity of homology computation grows exponentially with the dimension k. Analyzing H_1 for thousands of time windows is feasible, while H_2 or higher would be computationally prohibitive.

We don't just compute $H_k(R(X, \varepsilon))$ for a single ε. Instead, we analyze the entire sequence of maps induced by the inclusions in the Vietoris-Rips filtration:

$$H_k(R(X, \varepsilon_1)) \to H_k(R(X, \varepsilon_2)) \to \cdots \to H_k(R(X, \varepsilon_m))$$

Persistent homology provides an algorithm to track individual topological features through this sequence. For each feature α (e.g., a specific loop):

- **Birth (b_α):** The scale ε at which the feature α first appears as a generator of a homology group. It is a cycle that is not yet a boundary.
- **Death (d_α):** The scale ε at which α becomes a boundary, meaning it gets "filled in" by higher-dimensional simplices and merges with another, older feature.

The algorithm pairs the simplex that creates a feature with the one that destroys it.

The Bridge: From Homology Groups to Birth-Death Pairs

The Role of Induced Maps

The connection between homology groups and the pairs (b_α, d_α) is made by analysing the linear maps $f_{i,j*} : H_k(R(X, \varepsilon_i)) \to H_k(R(X, \varepsilon_j))$ for $i < j$.

The Birth of a Feature α

A feature α is **born** at scale $b_\alpha = \varepsilon_i$ if it is a new generator in the homology group $H_k(R(X, \varepsilon_i))$.

Formally, its homology class is in $H_k(R(X, \varepsilon_i))$ but **not** in the image of the map from the previous step, $f_{i-1,i*}$. It is a "brand new" hole.

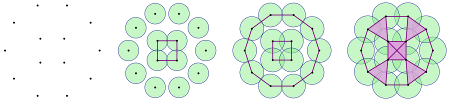
The Death of a Feature α

The feature α (born at ε_i) **dies** at scale $d_\alpha = \varepsilon_j$ if it becomes trivial for the first time.

Formally, its homology class is mapped to zero by the inclusion map: $f_{i,j*}(\alpha) = 0 \in H_k(R(X, \varepsilon_j))$. This means the cycle that represented α has now become a boundary.

An Example: Tracking a Loop's Life

Let's apply this to our previous filtration example, focusing on the two main loops: the inner loop (α) and the outer loop (β).



Life of the Inner Loop (α):

- **Birth (b_α):** At ε_1, a new generator α appears in H_1. It is born.
- **Survival:** At ε_2, α is mapped to a non-zero element in the next homology group. It survives.
- **Death (d_α):** At ε_3, α gets "filled in" and is mapped to zero. It dies.

Pair: $(\varepsilon_1, \varepsilon_3)$

Life of the Outer Loop (β):

- **Birth (b_β):** At ε_2, a second, independent generator β appears in H_1. It is born.
- **Survival:** At ε_3, β is mapped to a non-zero element. It survives.
- **Death (d_β):** At some later scale $\varepsilon_4 > \varepsilon_3$ (not pictured), it would be filled in and die.

Pair: $(\varepsilon_2, \varepsilon_4)$

Output: The Persistence Diagram (P_k)

The result of this tracking process is summarized in a **persistence diagram**, P_k.

- P_k is a multiset of points (b_α, d_α) in the plane, where each point corresponds to a single k-dimensional feature α.
- The diagram also includes the diagonal $y = x$ with infinite multiplicity, representing features with zero lifetime.

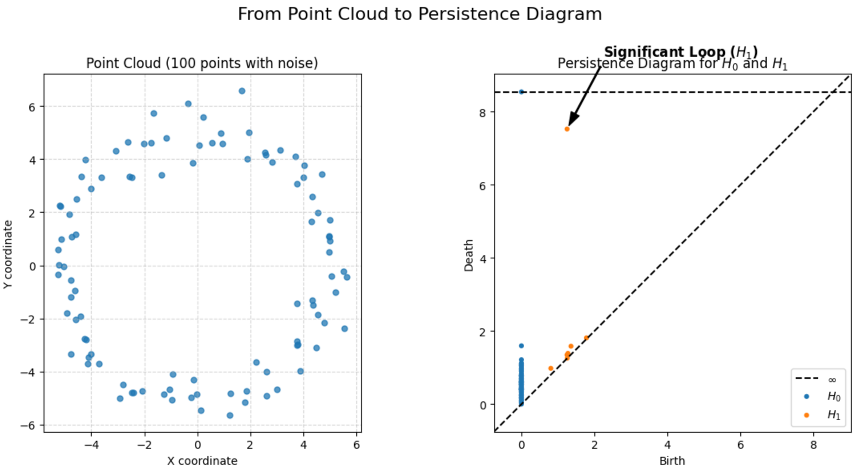
Interpretation:

- **Points far from the diagonal** represent robust, structurally significant features. Their vertical distance to the diagonal, $d_\alpha - b_\alpha$, is their **persistence**.
- **Points near the diagonal** represent "topological noise", so transient features that appear and disappear almost immediately.

Goal

Our analysis will focus on quantifying the "energy" of the points far from the diagonal.

A Concrete Example: From Point Cloud to Persistence Diagram



A Concrete Example: From Point Cloud to Persistence Diagram

We start with a point cloud generated in the shape of a circle with added noise. Visually, we can identify one main structural feature: a large central loop.
This is the resulting persistence diagram. It translates the visual geometry into a quantitative summary.

- **Blue dots (H_0):** Most are near the diagonal, representing topological noise. The one feature at infinity (dashed line) correctly identifies the cloud as a single connected component.
- **Orange dots (H_1):** There is **one point with very high persistence**, far from the diagonal.

Conclusion

The single, highly persistent H_1 point is the **algebraic signature of the significant loop** we saw in the data. TDA has successfully identified the true topological feature and separated it from the noise. This is the exact principle we apply to financial data.

A Problem with Diagrams: The Limits of Statistics

- The space of persistence diagrams is not a vector space.
- We can define distances between diagrams (e.g., Bottleneck, Wasserstein distance), but standard statistical operations are not straightforward.
- **The Challenge:** How do we compute the "average" of several diagrams, or their variance? This is crucial for time series analysis.

Need for a new representation

We need to map diagrams into a space where statistics is well-defined, like a Banach space.

Solution: Persistence Landscapes

- A persistence landscape is a function-based representation of a persistence diagram.
- It maps a diagram into a sequence of functions $\lambda = (\lambda_k)_{k \geq 1}$ in a Banach space, typically $L^p(\mathbb{N} \times \mathbb{R})$.
- This representation is stable and allows for standard statistical analysis.

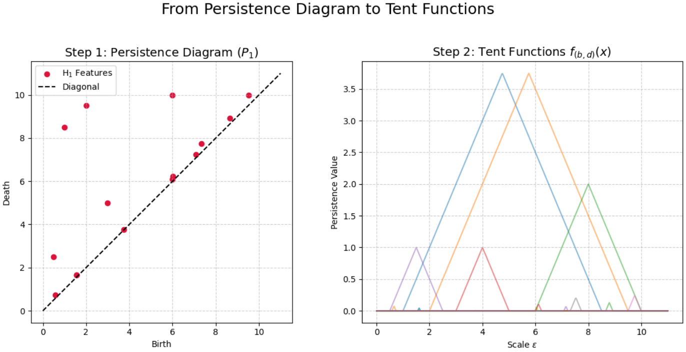
Construction in Two Steps:

1. Convert each point in the diagram to a piecewise function.
2. Combine these pieces to form the landscape functions.

Landscape Construction

- For each point (b_α, d_α) in the persistence diagram, we define a piecewise linear function $f_{(b_\alpha,d_\alpha)} : \mathbb{R} \to \mathbb{R}$:

$$f_{(b_\alpha,d_\alpha)}(x) = \begin{cases} x - b_\alpha & \text{if } x \in (b_\alpha, \frac{b_\alpha+d_\alpha}{2}] \\ -x + d_\alpha & \text{if } x \in (\frac{b_\alpha+d_\alpha}{2}, d_\alpha) \\ 0 & \text{otherwise} \end{cases}$$

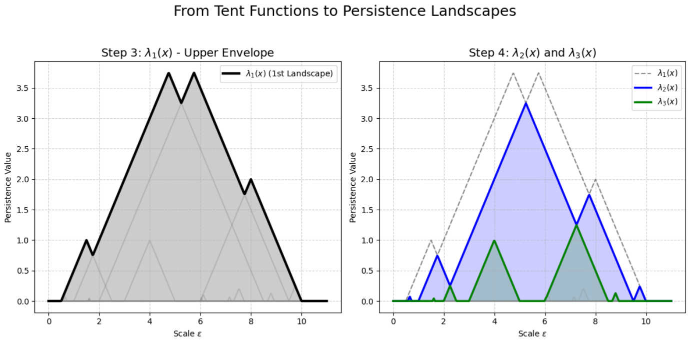


Landscape Construction: Step 2

- The landscape consists of a sequence of functions λ_k.
- The k-th landscape function, $\lambda_k(x)$, is the k-th largest value of all tent functions at point x.

$$\lambda_k(x) = k\text{-max}\{f_{(b_\alpha, d_\alpha)}(x)\}_\alpha$$

- $\lambda_1(x)$ is the upper envelope of all functions. $\lambda_2(x)$ is the second-highest envelope, and so on.



Quantifying the Landscape: The L^p Norm

- The persistence landscape $\lambda = (\lambda_k)_{k\geq 1}$ lives in the Banach space $L^p(\mathbb{N} \times \mathbb{R})$.
- We can summarize its "size" or "total persistence" with a single number: its L^p norm.

$$\|\lambda\|_p = \left(\sum_{k=1}^{\infty} \int_{-\infty}^{\infty} |\lambda_k(x)|^p dx \right)^{1/p} = \left(\sum_{k=1}^{\infty} \|\lambda_k\|_p^p \right)^{1/p}$$

- In this study, the L^1 and L^2 norms are used as key indicators.

The Central Idea

A large $\|\lambda\|_p$ value signifies the presence of many significant, persistent topological features in the original point cloud.

Input Data: Financial Time Series

- **Data:** Daily adjusted closing prices for 4 major US stock indices: S&P 500, DJIA, NASDAQ, Russell 2000.
- **Period:** Dec 23, 1987 – Dec 8, 2016 (7301 trading days).
- **Preprocessing:** We compute daily log-returns for each index j:

$$r_{i,j} = \log(P_{i,j}/P_{i-1,j})$$

  where $P_{i,j}$ is the adjusted closing price of index j on day i.
- This results in $d = 4$ time series $\{x_n^k\}_{k=1,2,3,4}$.

Step 1: The Sliding Window Technique

- To capture the time-varying dynamics of the market, we use a **sliding window** approach.
- A window of a fixed length w (e.g., $w = 50$) slides along the $d = 4$ time series, one day at a time.

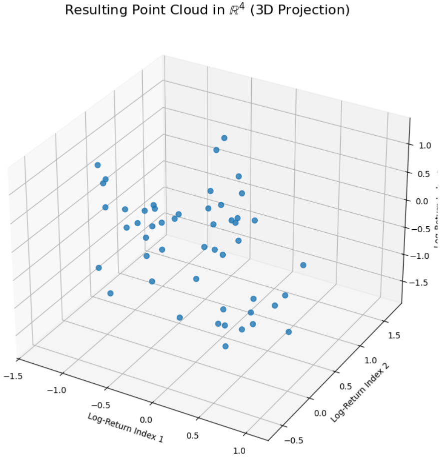
Figure: A window of width w (red shaded area) slides over $d = 4$ time series.

Step 2: Point Cloud Generation

- Each position of the sliding window defines a point cloud.
- A window of length w over d time series generates a cloud of w points in $\mathbb{R}^d$, which can be thought as a $d \times w$ matrix.
- **Example:** For a window starting at day t, the point cloud X_t is the set of vectors:

$$X_t = \{(x_n^1, x_n^2, \ldots, x_n^d) \in \mathbb{R}^d \mid n = t, \ldots, t + w - 1\}$$

- This gives a sequence of point clouds $\{X_t\}$, each representing the geometric state of the market over a w-day period. Then, for every point cloud, we apply the TDA procedure.



The Full Analysis Pipeline

Financial Time Series

↓

Sliding Window

↓

Sequence of Point Clouds $\{X_t\}$

↓

Sequence of Landscapes $\{\lambda_t\}$

↓

Time Series of Norms $\{\|\lambda_t\|_p\}$

Sliding Window vs Time-Delay Embedding

Our approach is intentionally different from the classic time-delay embedding method.

Classic Time-Delay Embedding

- **Input:** A single 1D time series.
- **Goal:** Reconstruct a high-dimensional attractor of a dynamical system.
- **Parameters:** Requires choosing embedding dimension m and time lag τ.
- **Challenge:** Difficult to apply with noise and may be impractical if no attractor exists.

Sliding Window Method

- **Input:** A handful (d) of multivariate time series.
- **Goal:** Represent the time series in its natural low-dimensional space, $\mathbb{R}^d$.
- **Parameters:** Requires only the window size w.
- **Advantage:** More practical for noisy, stochastic, and non-stationary data, as is common in finance.

The Question: Is Our TDA Indicator Reliable?

- Before applying our pipeline to real financial data, we must validate it.
- We test the method on synthetic datasets where we know the ground truth.
- **Goal:** Verify that the L^p norm of the landscape is a sensitive detector of changes in system dynamics and volatility.

Test 1: Transition to Chaos (Hénon Map)

Goal

To test if our TDA indicator can detect a well-known dynamical transition: the onset of chaos in a system analogous to economic cycles.

The Stochastic Model:

- To better mimic financial data, the model is:
  1. The real parameter a_n is made to change slowly over time $(a_n)_n$, while the real parameter b has fixed values.
  2. Gaussian noise (W_n) is added to simulate stochastic effects:

$$\begin{cases} x_{n+1} = 1 - a_n x_n^2 + b y_n + \sigma W_n \sqrt{\Delta t}, \\ y_{n+1} = x_n + \sigma W_n \sqrt{\Delta t}, \\ a_{n+1} = a_n + \Delta t \end{cases}$$

where $\Delta t > 0$ is a small step size and $\sigma > 0$ is the noise intensity.

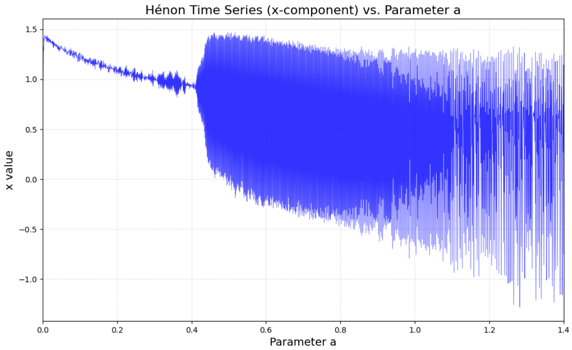
Test 1: Hénon Map - The Experimental Pipeline

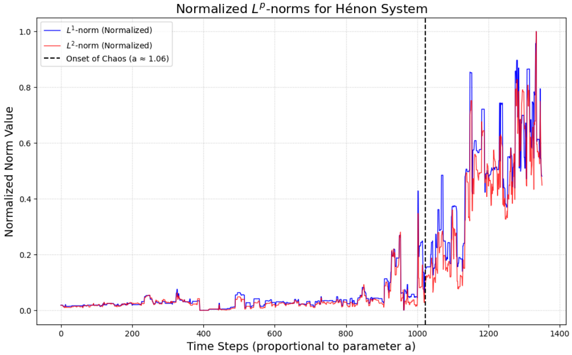
This is the step-by-step procedure applied to the stochastic Hénon model:

1. **Generate Time Series Data:**
   - We generate $d = 4$ different time series. Each series uses a different, fixed value for the parameter b (from 0.27 to 0.3).
   - The control parameter a is slowly increased from 0 to 1.4, driving the system from a stable state towards chaos.
2. **Create a Sequence of Point Clouds:**
   - A sliding window of size $w = 50$ is moved along the 4 time series.
   - Each window produces a point cloud X_n of 50 points in $\mathbb{R}^4$.
3. **Apply the TDA Pipeline to Each Point Cloud:**
   - For each X_n, we compute the Rips filtration.
   - From the filtration, we compute the H_1 persistence diagram, $P_1(X_n)$.
   - From the diagram, we construct the persistence landscape, $\lambda(X_n)$.
   - Finally, we calculate the L^1 and L^2 norms of the landscape.

The final output is a time series of the L^p-norm values.

Test 1: Hénon Map - Results and Conclusion





Comparing the two plots, we see a **sharp and significant increase** in the norms (right plot) precisely at the time step where the system transitions to chaos (left plot, around $a \approx 1.06$).

Conclusion

The L^p-norms of persistence landscapes act as a sensitive detector for the topological changes induced by a shift in the system's underlying dynamics, even in the presence of noise.

Goal

To test if the L^p-norms are sensitive to the dispersion (variance) of the data, which is a proxy for market volatility.

The Setup (Monte Carlo Simulation):

- Instead of a time series, we generate static point clouds and measure their topological "size".
- For a given variance σ^2, we generate a point cloud of 100 points in $\mathbb{R}^4$. Each coordinate is drawn from a Normal distribution $N(0, (\sigma + \delta_i)^2)$, where δ_i is a small random term.
- We repeat this process 100 times (100 realizations) and compute the **average** L^1 and L^2 norms.
- This entire simulation is then repeated for 10 different values of σ, from 1 to 10.

We know that scaling the standard deviation by a factor F transforms a persistence point (b, d) to (Fb, Fd).This implies the L^p-norms should be proportional to the variance, σ^2.

Test 2: White Noise with Growing Variance

The numerical experiment confirms the theory. The average L^1 and L^2 norms increase **linearly** with the noise variance.

Conclusion

The L^p-norms of persistence landscapes provide a direct, quantitative, and predictable measure of the system's variance. They are reliable volatility sensors.

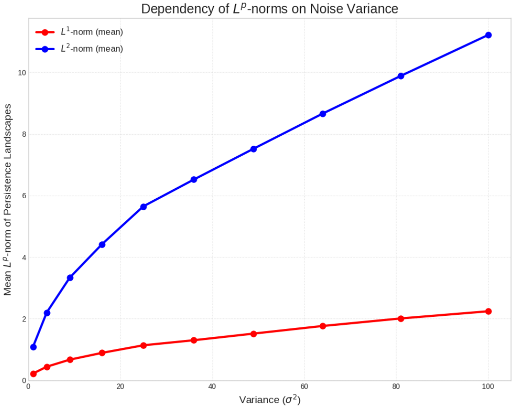


Figure: Mean L^1 (red) and L^2 (blue) norms vs. noise variance σ^2.

Goal

To test the method on a model that mimics two key stylized facts of financial returns: **stochastic volatility** and **"fat tails"**.

The Setup (Superstatistics Model):

- We model the data as a mixture of Gaussian distributions, where the inverse variance $\gamma = 1/\sigma^2$ is itself a random variable.
- We choose a Gamma distribution for γ of parameters α and β (fixed), which results in an overall Student's t-distribution for the returns, so a classic fat-tailed model.
- We simulate a transition from a **"cold" market state** (low variance, high shape parameter $\alpha = 8$) to a **"hot" market state** (high variance, by decreasing α to 1.75).

The L^1-norm of the persistence landscapes is highly sensitive to this transition. It shows a sharp, substantial increase as the underlying distribution develops fatter tails and higher variance.

Conclusion

The TDA indicator effectively detects shifts into high-volatility, fat-tailed regimes, which are widely considered to be characteristic of pre-crash market conditions. This validates its potential for real-world financial data.

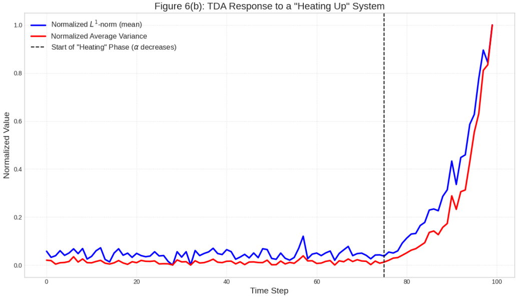


Figure: L^1-norm (blue) vs. α, with average noise variance (red) for reference.

Validation Summary

Key Takeaways from Synthetic Tests

The L^p norm of the persistence landscape is a robust quantitative indicator that reliably detects:

- ✓ Transitions to chaotic dynamics.
- ✓ Increases in data variance/dispersion.
- ✓ Shifts in stochastic volatility regimes.

We are now confident to apply this method to real financial data.

The Experiment Setup

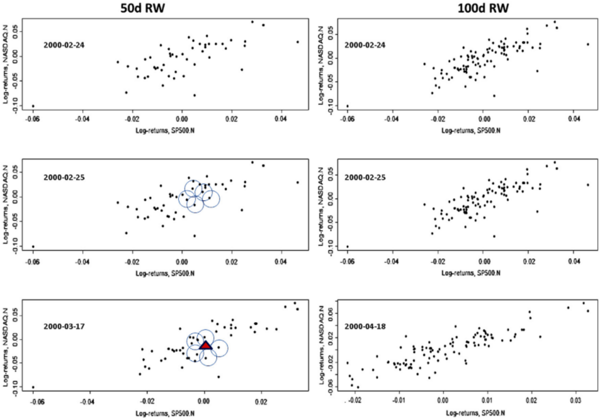
- **Data:** Daily log-returns of 4 major US indices (S&P 500, DJIA, NASDAQ, Russell 2000) from 1987 to 2016.
- **Method:** Apply the TDA pipeline using a sliding window of size $w = 50$ and $w = 100$ days.
- **Goal:** Analyze the resulting time series of L^p-norms to detect and anticipate market crashes.

Question

Can we visually and quantitatively see a change in the data's geometry during a crisis?

Visual Analysis: The Shape of Point Clouds

First of all, we can show the 2D projections of the 50-days and 100-days point clouds (S&P 500 vs NASDAQ) around the 2000 tech crash.

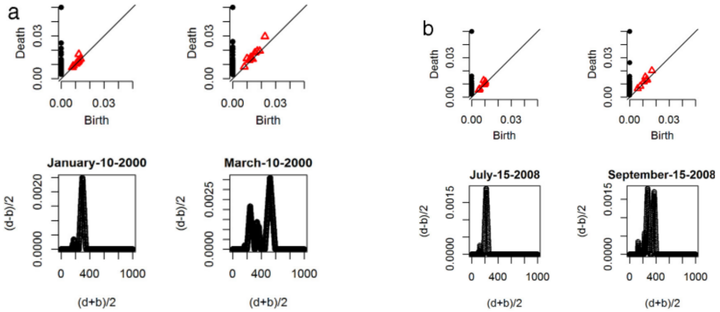


Observation

The shape of the point cloud visibly changes. In times of stress (e.g., 2000-02-25), the data becomes more structured and elongated, making the formation of topological loops more likely.

This provides a visual justification for our TDA approach.

Visual Analysis: TDA output for two major crises



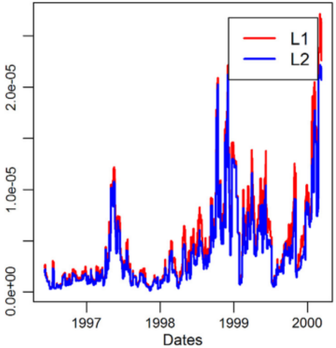
Tech Crash (2000)   Financial Crisis (2008)

Key Observation

In both crises, the persistence diagrams (top row) show a significant increase in the number and persistence of H_1 loops (red triangles). Consequently, the corresponding landscapes (bottom row) become much larger.

Market volatility makes the topology more complex.

Now we quantify this behavior by plotting the normalized L^1 and L^2 norms for the entire period.



The Main Result

The time series of the norms exhibits **massive spikes that align precisely with major financial crises**, including the 1998 Russian default, the 2000 dot-com crash, and the 2008 global financial crisis.

This confirms the L^p-norm is an effective crash detector.

The Question

Is our TDA indicator just a complicated way to measure volatility, or does it provide new information? To find out, we compare it to the standard market "fear gauge".

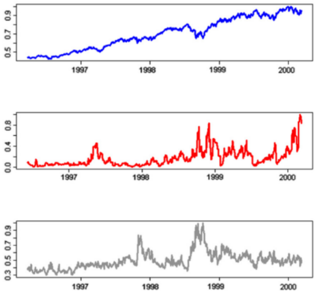
What is the VIX Index?

- It computes the market's expectation of **30-day forward-looking volatility** for the S&P 500 index.

We can compare the normalized L^1-norm and the VIX in the 1000 days leading up to two different crises.

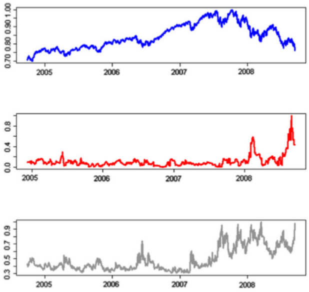
TDA vs. VIX: A Tale of Two Crashes

2008 Financial Crisis



Both the TDA norm and the VIX show clear rising trends, indicating growing risk. **Both indicators worked well.**

2000 Dot-com Crash



The TDA norm shows a strong rising trend. However, the **VIX fails to signal the impending crash**.

Conclusion: Why TDA Captures Different Information

TDA captures a different kind of risk.

- The **VIX** tracks volatility for **one single market** (the S&P 500).
- Our **TDA method** analyzes the relationships **between multiple markets**.
- **The 2000 crash** was a structural shift that began in the tech sector (NASDAQ). Our TDA method captured the breakdown in the market's internal **correlation structure**, a risk that the single-asset volatility measure missed.

This demonstrates that TDA is not just another volatility indicator, but a powerful tool for detecting changes in the systemic structure of the market.

Summary of Key Findings

- We presented a TDA-based method to quantify the topological complexity of multivariate financial time series.
- The L^p norm of persistence landscapes acts as a powerful indicator of this complexity.

✓ **Detection:** The norm series exhibits massive spikes that align perfectly with major market crashes.

✓ **Anticipation:** The **variance** of this norm series serves as a robust **early-warning signal**, showing a significant rise months before crises.

✓ **Superiority:** This TDA-based signal provides information that is complementary and, in key cases (like the 2000 crash), superior to standard indicators like the VIX.

The Take-Home Message

Topology offers a powerful new lens for financial analysis. By studying the **shape of data**, we can uncover hidden structures and detect rising systemic fragility that traditional statistical measures might miss.

References

M. Gidea and Y. Katz.
Topological data analysis of financial time series: Landscapes of crashes.
Physica A: Statistical Mechanics and its Applications, 491:820-834, 2018.

Thank you for the attention.