Continuous Learning of System Security Thru Deep Topological Learning

Peter Chin

Dept. of Computer Science, Boston University

C3E Annual Meeting, October, 2021

Topology

- The study of geometric properties and spatial relations unaffected by the continuous change of shape or size of figures.
- Two spaces are topologically equivalent if one can be formed into the other without tearing edges, puncturing holes or attaching non-attached edges

How can a mug and a torus be equivalent?

Topology: The Structure of Manifolds

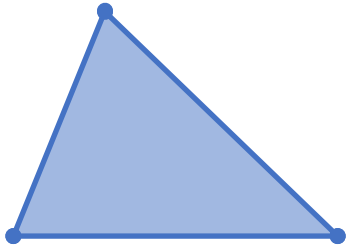
- Topology is used to study the structure of sets
- Why do the mug and donut have same topology?
  - They have differences, different geometries for example
  - What do they share? What is invariant between them?
- They share the same shape (Betti numbers, etc.)

Simplex & simplicial complex

Simplex:

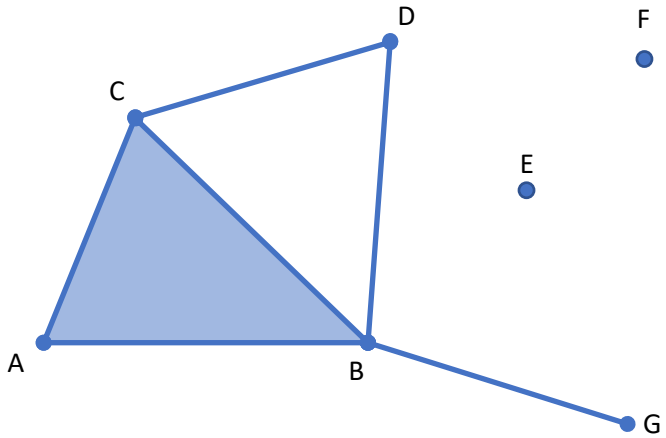
0-simplex
(point, vertex)

1-simplex
(edge)



2-simplex
(triangle mesh, 3-clique)

Simplicial complex:



Simplicial complex: a set of

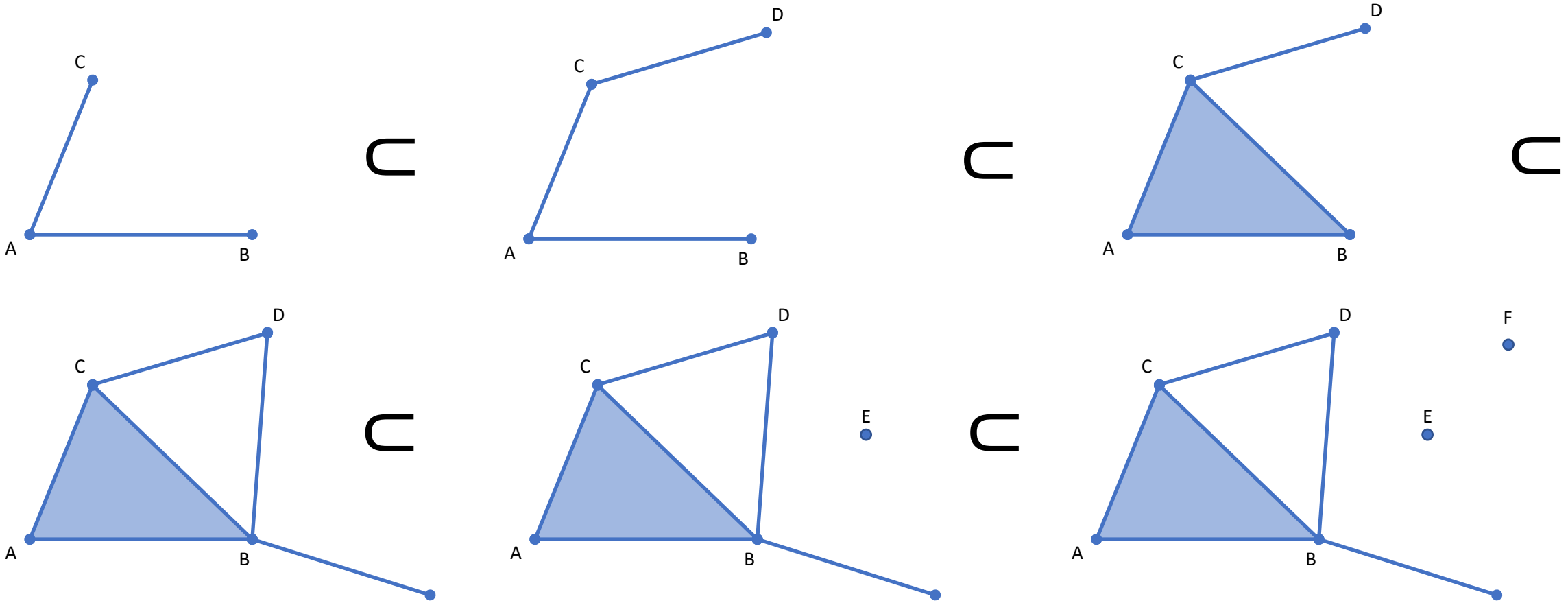
- 0-simplex: A, B, C, D, E, F, G
- 1-simplex: AB, AC, BC, CD, BD, BG
- 2-simplex: ABC

Simplex & simplicial complex

- An i-simplex is the convex hull of $i+1$ affinely independent points, i.e the set of all convex combinations $\lambda_0 v_0 + \lambda_1 v_1 + \cdots + \lambda_i v_i$ where $\lambda_1 + \lambda_2 + \cdots + \lambda_i = 1$ and $\lambda_j \leq 1, \forall j \in [i]$
- A simplicial complex is a finite collection of simplices K such that
  1. Every face of a simplex in K also belongs to K
  2. For any two simplices σ_1 and σ_2, if $\sigma_1 \cap \sigma_2 \neq \emptyset$ then $\sigma_1 \cap \sigma_2$ is a common face of both σ_1 and σ_2

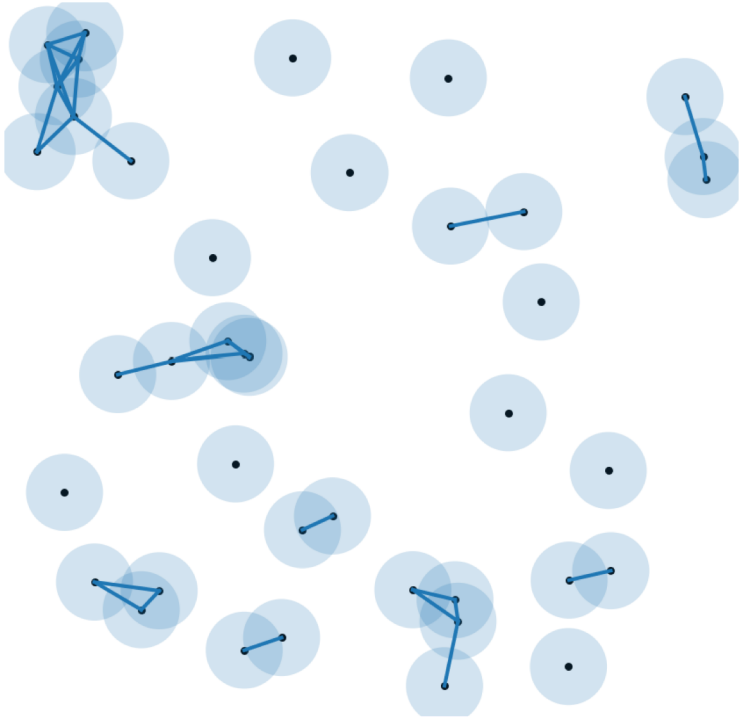
Filtration

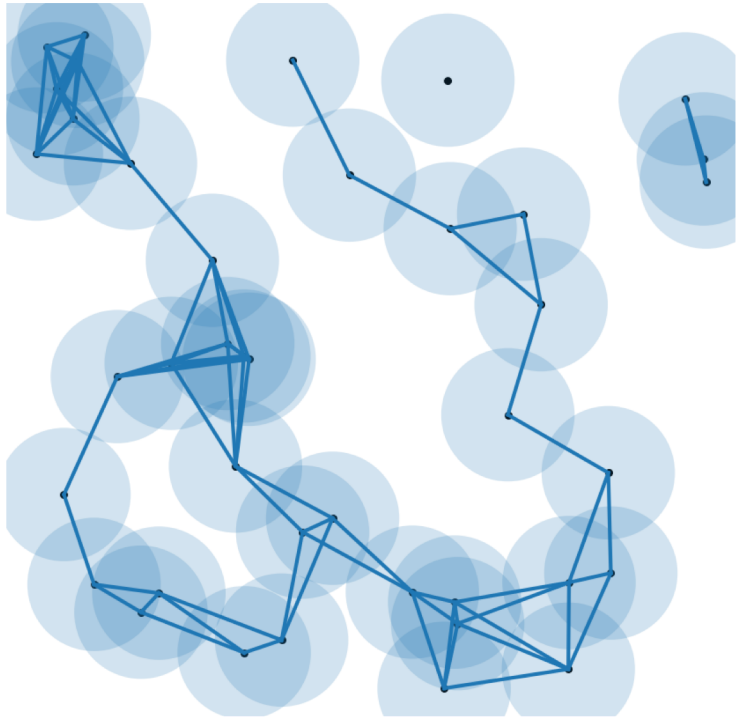
- Nested family of simplicial complexes



Example: Vietoris-Rips Filtration

- Defined from any metric space M and distance σ
- Forming a simplex for every finite set of points that has diameter at most σ

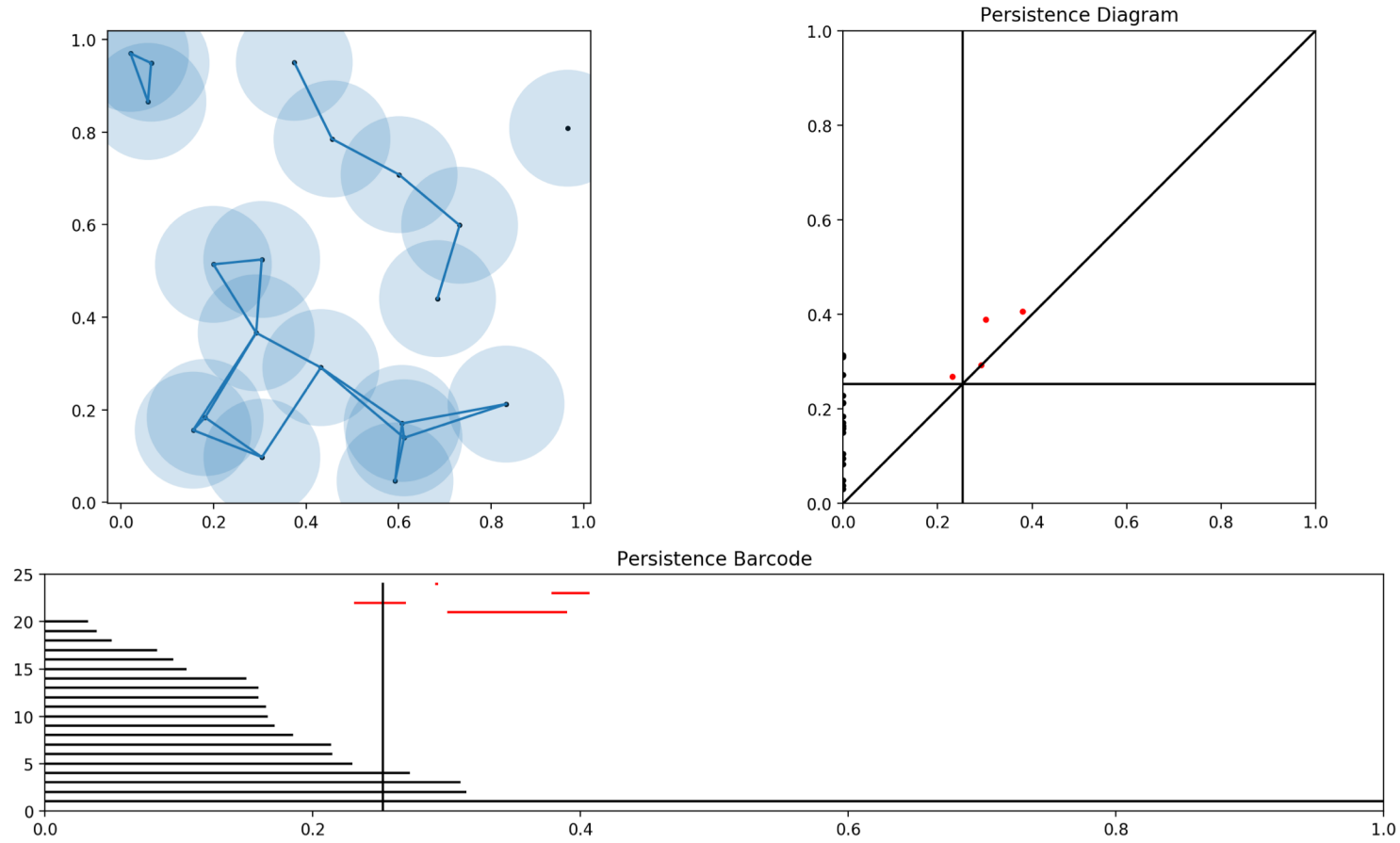




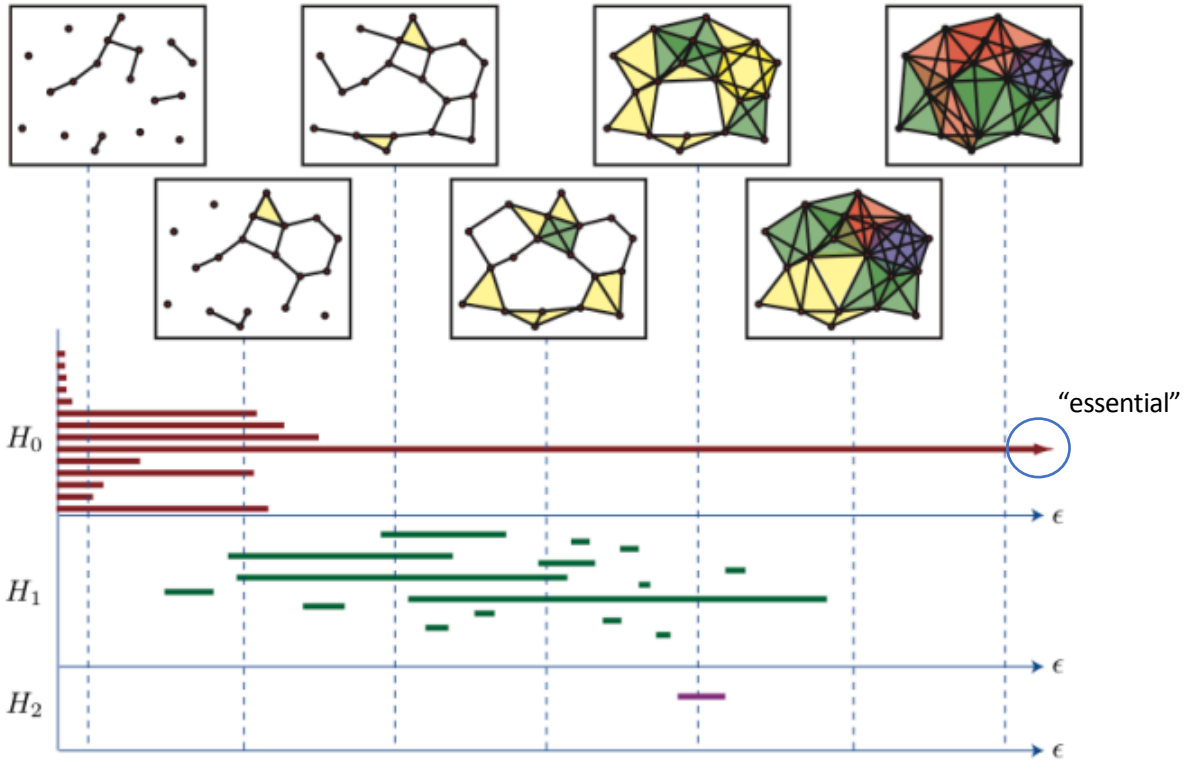
Persistence Homology

- *Topological Features:* Information related to components, holes, voids, etc. in the data
- *Persistence Homology*: A topological summary; a method for computing topological features of a space at different spatial resolutions
- *Persistence Diagrams*: Set of (birth, death) where each is corresponding to a topological feature of the data

Visualization



Persistence Homology



Homology:

- Degree 0: connectedness of the data
- Degree 1: holes and tunnels
- Degree 2: voids

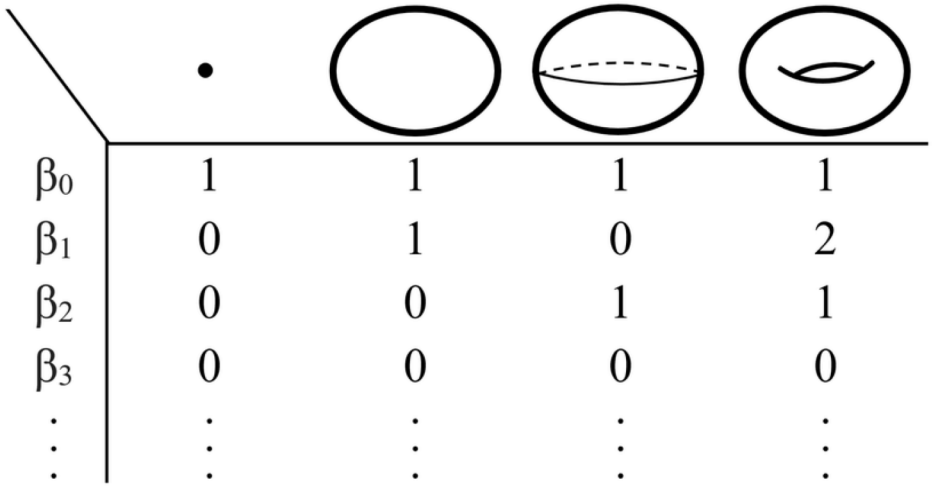
Betti number: (rank of homology groups = number of barcodes)

- β_0: number of components
- β_1: number of holes (or cycles)

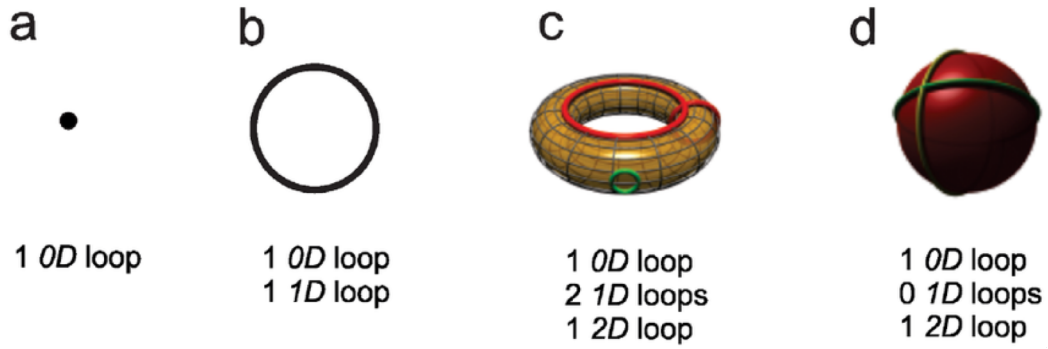
https://datawarrior.wordpress.com/2015/12/20/tda-4-persistence/

Betti Numbers

- Betti numbers count the # and dimensionality of holes in a set
  - B_0 number of connected components
  - B_1 number of 1D or “circular” holes
  - B_2 number of 2D “voids” or “cavities”
- B_k is maximum # of k-dimensional curves which can be removed while object remains connected
- # of k-dimensional holes
- # of “non-contractible” k-dimensional loops

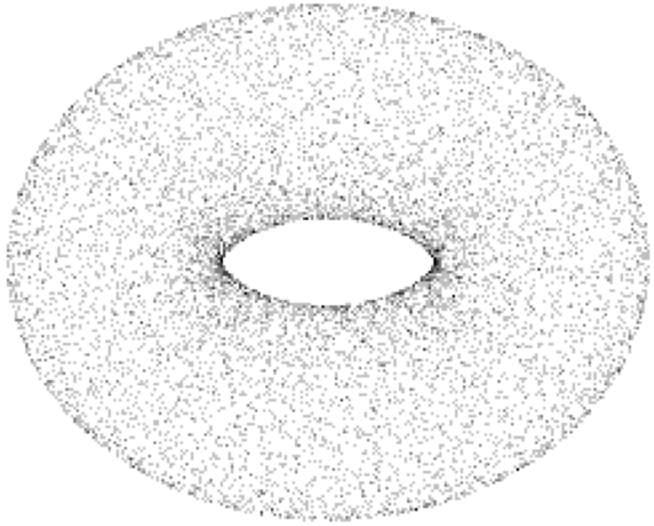


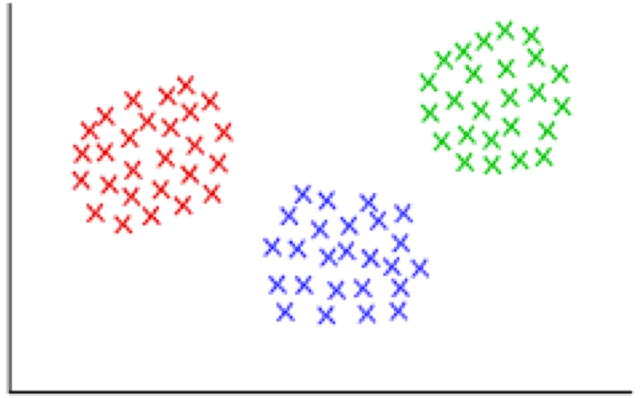
	•			
β_0	1	1	1	1
β_1	0	1	0	2
β_2	0	0	1	1
β_3	0	0	0	0
⋮	⋮	⋮	⋮	⋮



Topology of Datasets

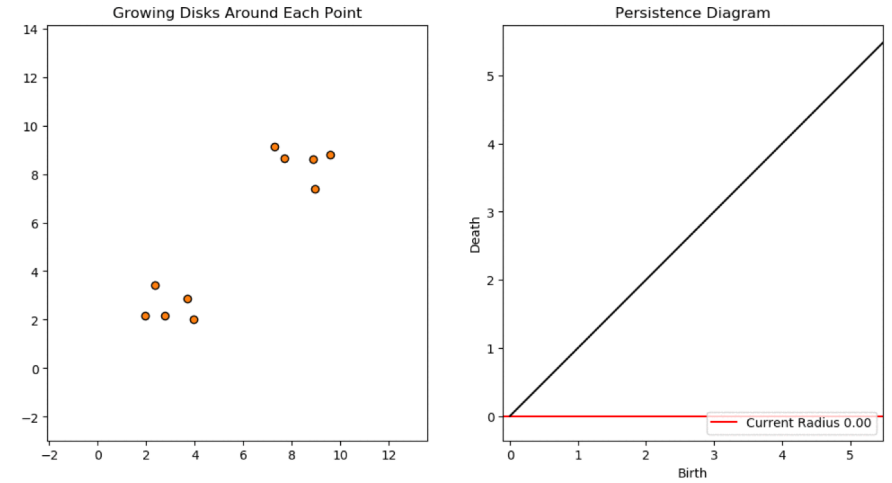
- If we apply topology to dataset alone
  - B_0 = # of datapoints, everything else 0
- What we see and know:
  - Top is torus, $B_0 = 1$, $B_1 = 2$, $B_2 = 1$
  - Bottom 3 clusters
    - 3 connected components
    - $B_0 = 3$

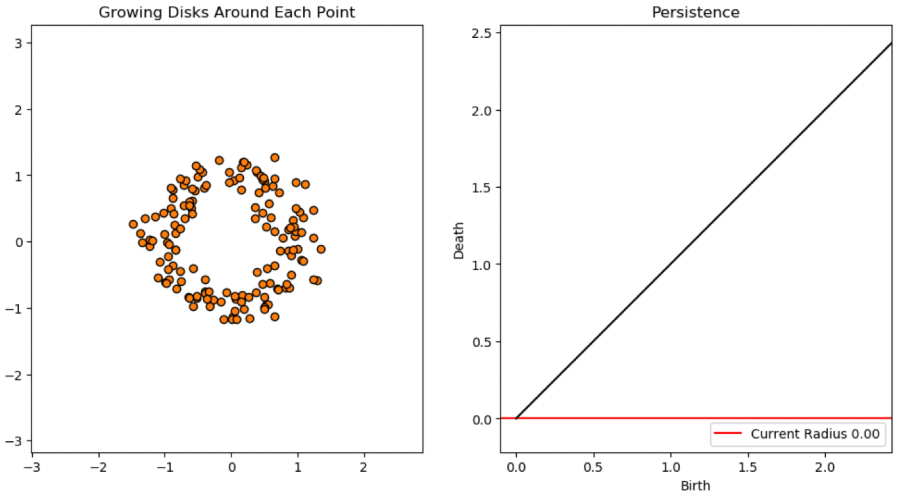




Persistent Homology

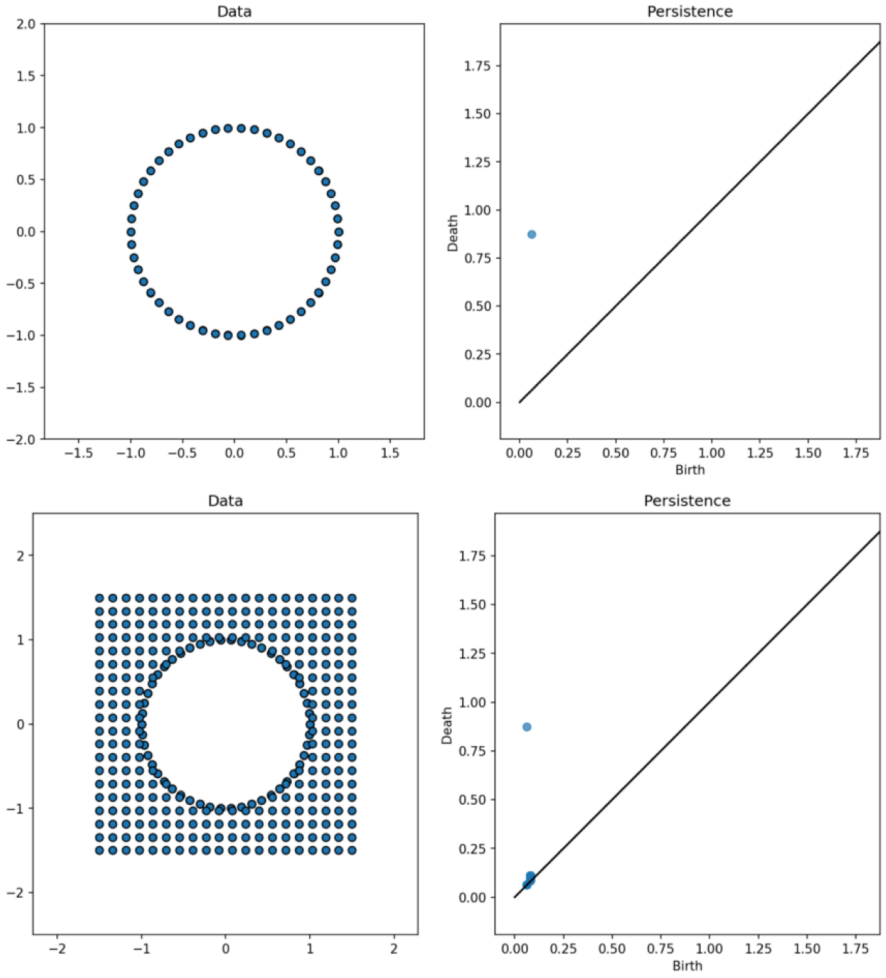
- Persistence Diagrams track the “births” and “deaths” of Betti holes as data points are expanded into disks
- Can identify number of clusters, structure of data, types of periodicity and much more





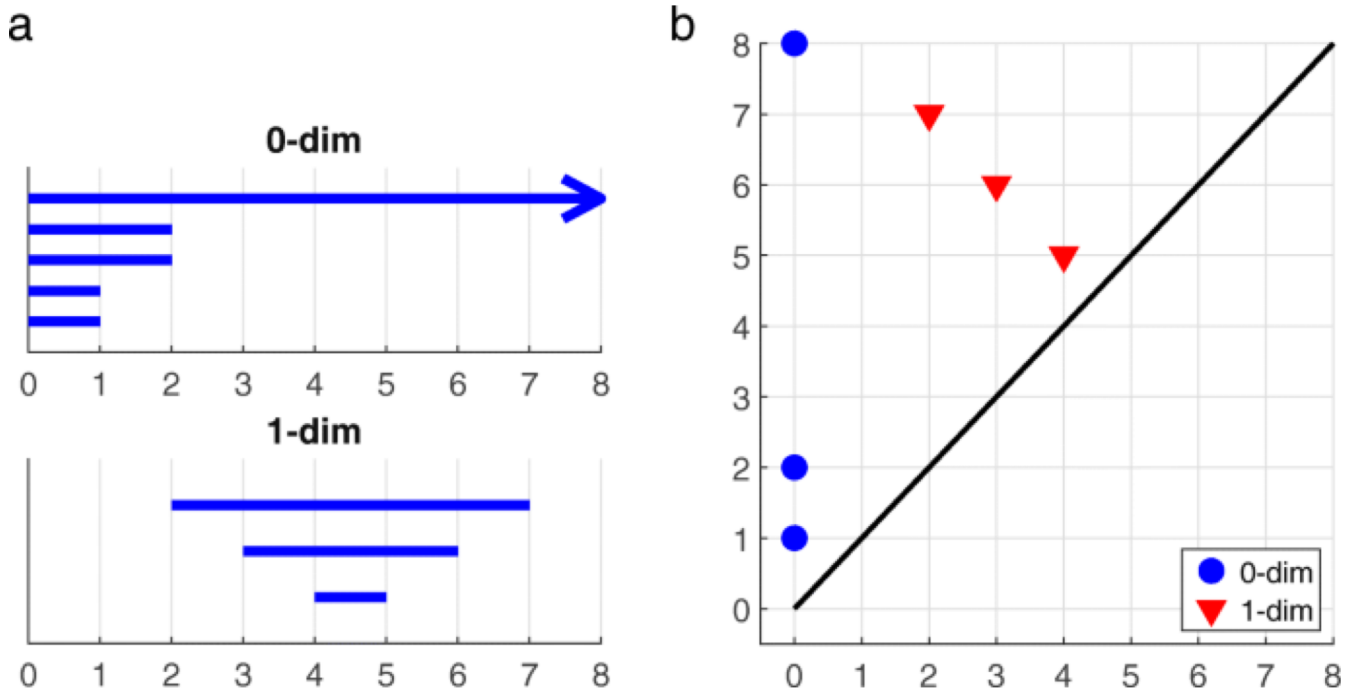
Persistent Homology: Robustness

- Has demonstrated robustness to noise
  - Not robust to anomalous noise
- Demonstrated stability
  - Data subsetting
  - Noise
  - Point addition



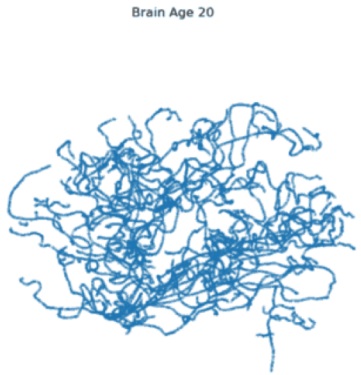
Persistence Diagrams and Bar Codes

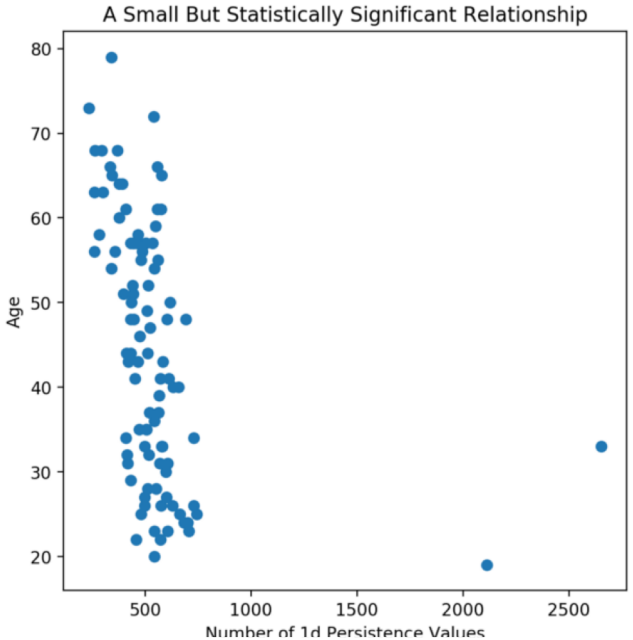
- We have used persistence diagrams in our analysis thus far
  - Persistence bar codes contain the same information in a different format
  - Both used frequently and interchangeably in the field

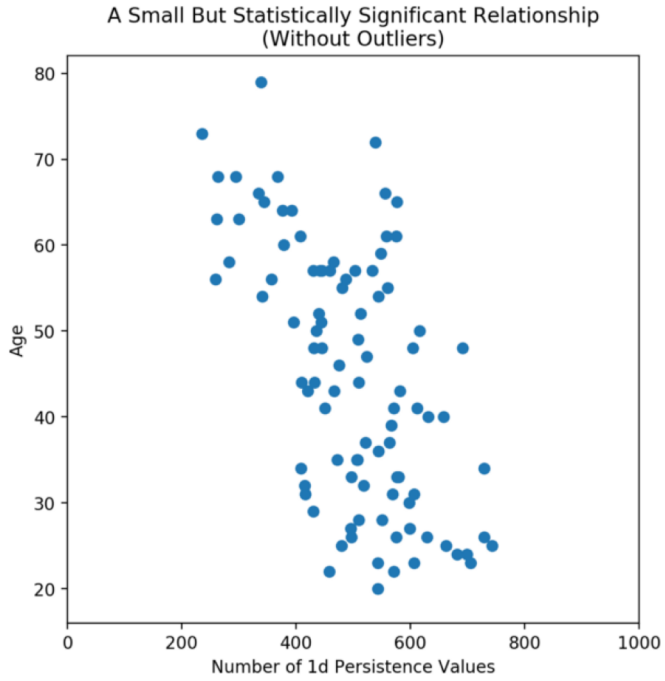


Powerful Features

- Powerful engineered feature
- Can detect anomalies simply in complex data
- Can detect trends simply in complex data







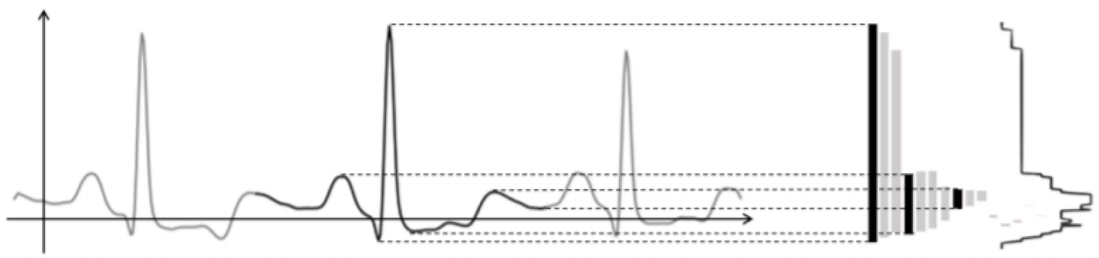
Powerful Features work Separately from traditional Statistical Features

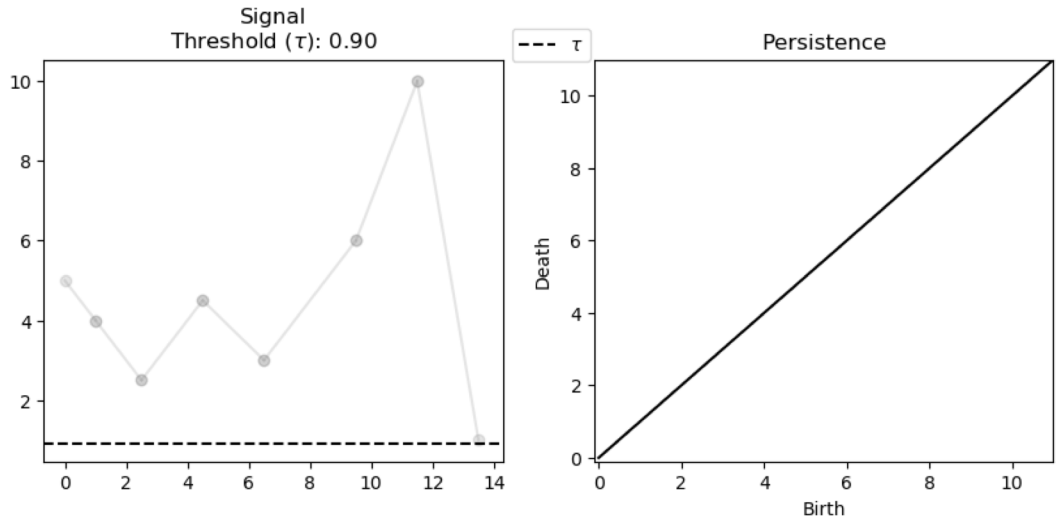
- Traditional statistics has already simplified intuitive properties of data such as “center” and “spread”
  - Mean and Variance
- Persistent Homology can simplify slightly more complex but still highly intuitive properties of data, its structure
- Statistics and Persistent Homology Can be used together, they measure different properties

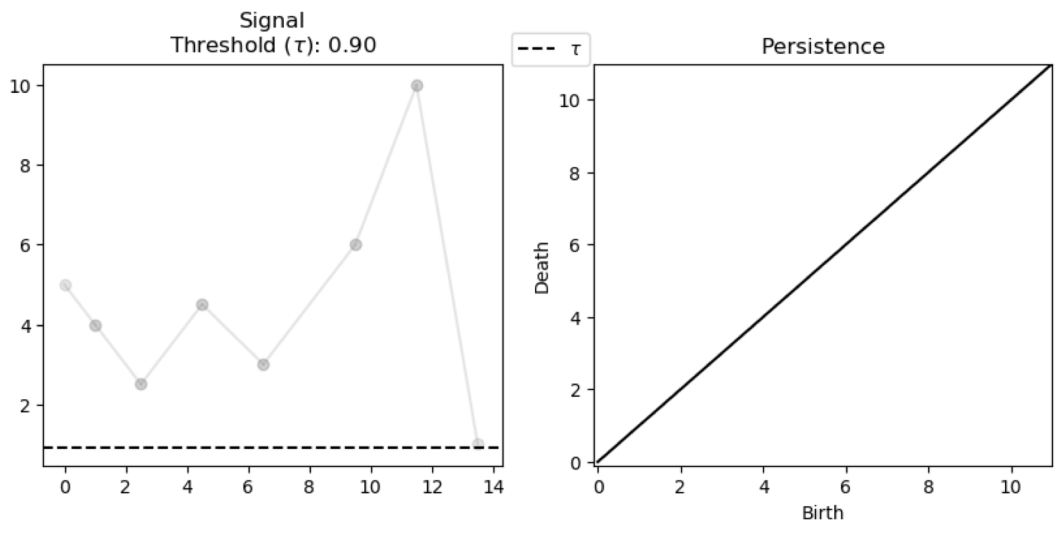
Figure 2
Example of data having the same mean and variance.

Persistent Homology for Signals

- Thresholded signals
- Give persistent signal values
- Simple, but effective

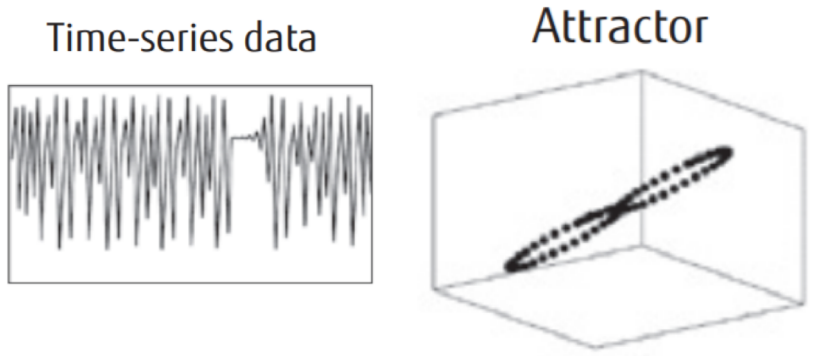


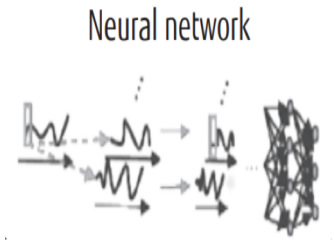




Time Series Applications and Anomaly Detection

- Has been used as powerful preprocessing
- Can and has been used for anomaly detection
- Compare persistence diagram of current time window to history of persistence diagrams
- Periodicity
  - Simple Periodicity => $B_1 = 1$, ie: sin(3x)
  - More complex Periodicities also:
    - Torus Betti Numbers, ie: 2*sin(pi*x)+sin(e*x)





Side-Chanel Analysis & Data Collection

Math Loop

```
void loop() {
  int randNum1, randNum2;
  noInterrupts();
  digitalWrite(LED_BUILTIN, LOW);
  digitalWrite(LED_BUILTIN, HIGH);

  randNum1 = random(65535);
  randNum2 = random(65535);

  interrupts();
}
```

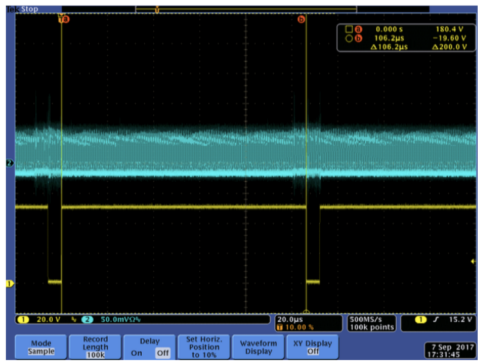
Disable Interrupts

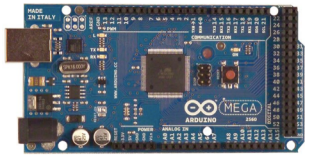
Sync

Calculate two random numbers

Enable Interrupts

Math operation representing a typical operation in the system





+Vcc

Split-Core Current Transformer

Magnetic field loop sensor

Arduino Mega 2560

Chan 1

PicoScope 3405

Chan 2

Trigger signal

Distance measurements:

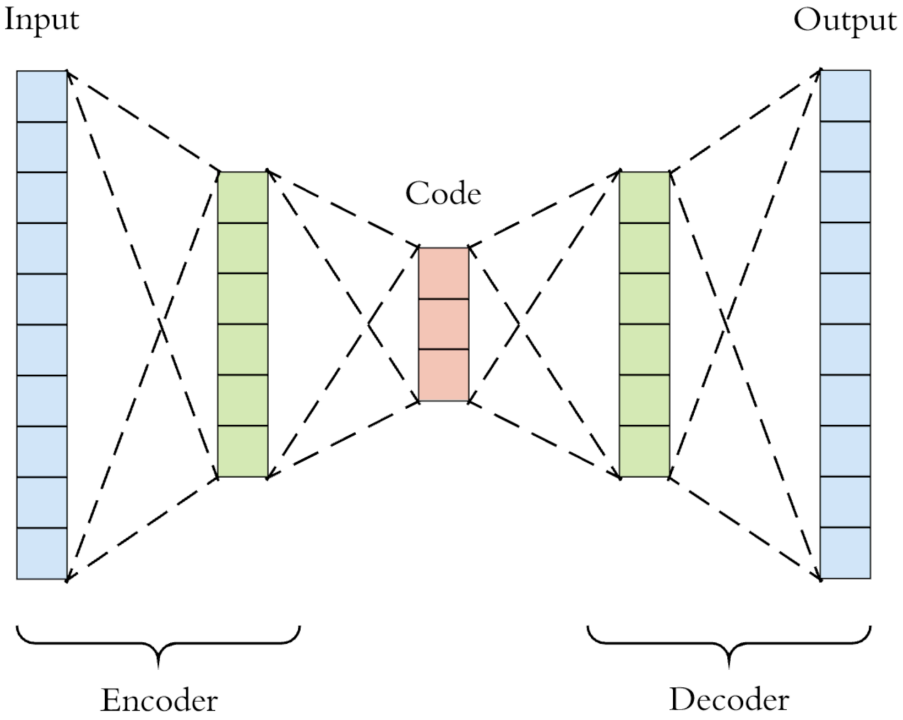
Ideal: 0”, 4”, 8”, 12”

Examples of Classification Experiments

Devices	Classes	Class explanation	Samples	Sampling rate	Trace Length	Sync	Classification Acc.
Arduino	2	Same math operation with 2 set of different operands	800	125Mhz	4096	Yes	100%
Raspberry Pi	2	Same operation with or without botnet attack	600	30518 Hz	305180	No	100%
Siemen's PLC	5	4 different machine operations + botnet attack	1000	30518 Hz	305180	No	91.28%

Anomaly Detection by Autoencoder

- Anomaly detection
  - Use only negative examples for training
  - Unsupervised learning
  - Detect positive examples in inference
- Autoencoder
  - Reconstruction of input
  - From compressed latent variable

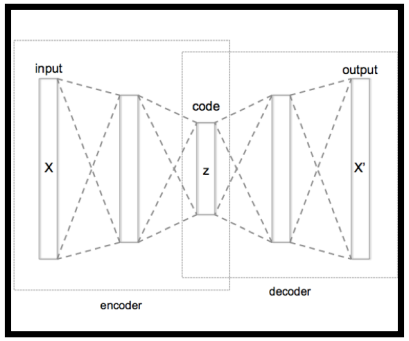


Side-channel Analysis: Anomaly Detection

Training

Normal examples

Autoencoder

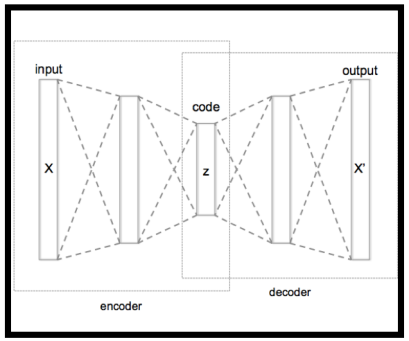


loss

Inference

Normal examples & examples samples

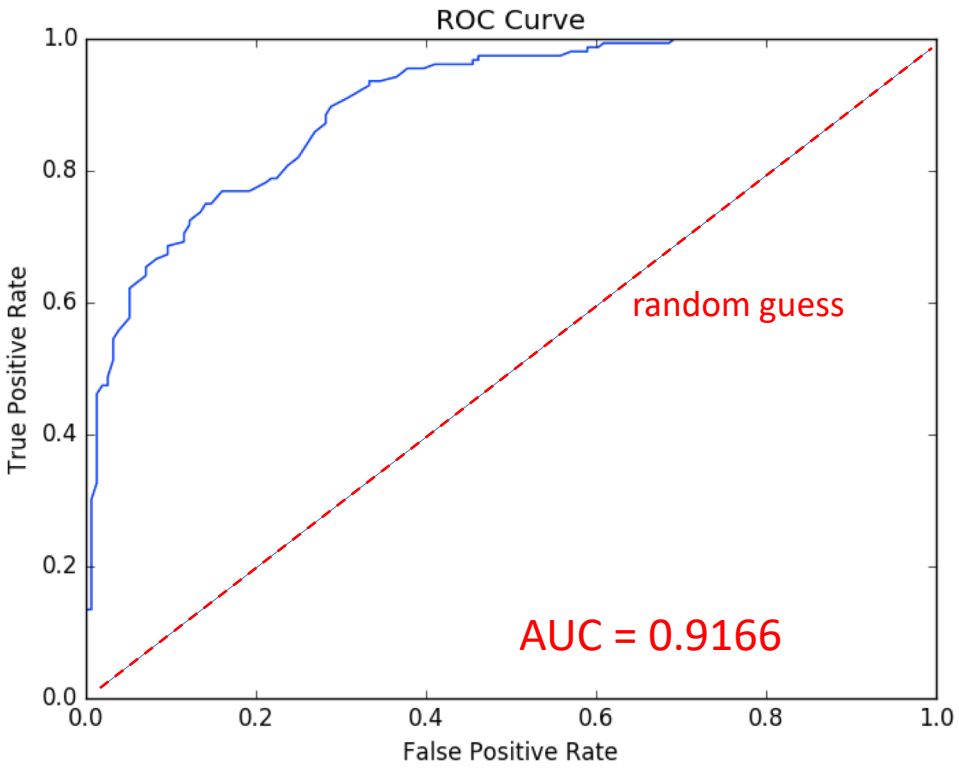
Autoencoder



compare loss → loss threshold

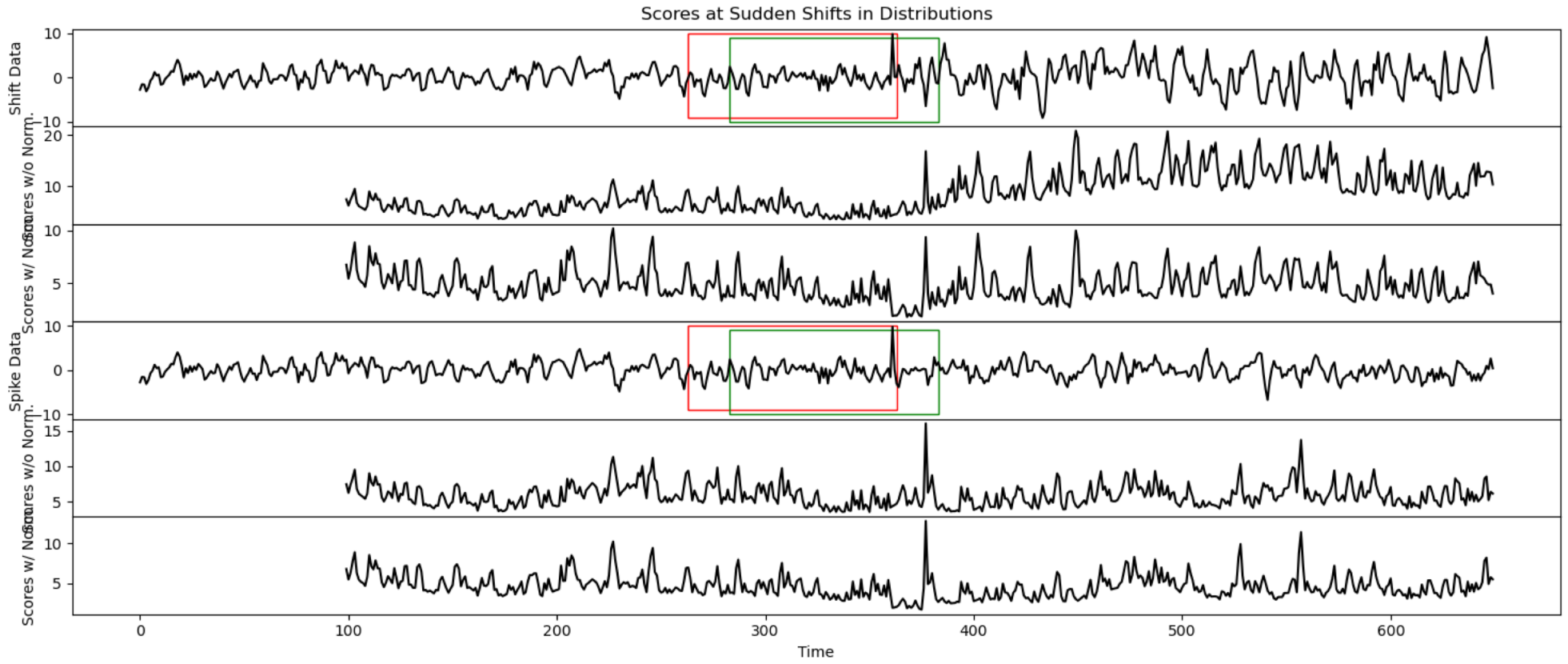
If larger than the threshold, flag it as anomalous

Anomaly Detection: Performance Evaluation



Best F1-score point:
True positive=0.935
False positive=0.33
Threshold=113.59

Windowing the time-series signals



Results on Benchmark datasets

TABLE I
EXPERIMENTAL RESULTS, F1 SCORES

Model	**NASA**		**YAHOO**				**NAB**						
	SMAP	MSL	A1	A2	A3	A4	Art	AdEx	AWS	Traf	Tweets	μ	σ
DenseAE w/ Post	0.623	**0.797**	**0.916**	**0.995**	**0.976**	**0.912**	**0.8**	0.762	0.762	**0.8**	**0.71**	**0.82**3	0.115
DenseAE w/o Post	**0.655**	0.608	0.496	0.283	0.097	0.041	0.667	0.533	**0.764**	0.333	0.742	0.474	0.252
TADGAN [6]	0.623	0.704	0.8	0.867	0.685	0.6	**0.8**	**0.8**	0.644	0.486	0.609	0.693	0.114
LSTM [4]	0.46	0.69	0.744	0.98	0.772	0.645	0.375	0.538	0.474	0.634	0.543	0.623	0.171
ARIMA [3]	0.492	0.42	0.726	0.836	0.815	0.703	0.353	0.583	0.518	0.571	0.567	0.599	0.156
Deep AR [18]	0.583	0.453	0.532	0.929	0.467	0.454	0.545	0.615	0.39	0.6	0.542	0.555	0.142
HTM [10]	0.412	0.557	0.588	0.662	0.325	0.287	0.455	0.519	0.571	0.474	0.526	0.489	0.113
MADGAN [19]	0.111	0.128	0.37	0.439	0.589	0.464	0.324	0.297	0.273	0.412	0.444	0.35	0.144
MS Azure [20]	0.218	0.118	0.352	0.612	0.257	0.204	0.125	0.066	0.173	0.166	0.118	0.219	0.152

Graph Classification

- Input: a set of graph (V, E) with its label
- Optional properties: vertex label / weight, edge label / weight
- Some benchmark data sets for classification task

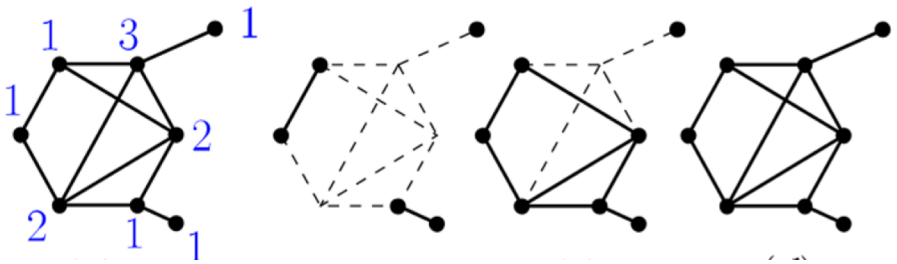
	No. graphs	No. classes	Avg. no. nodes	Avg. no. edges	Note
MUTAG	188	2	18.0	19.8	Chemical compounds
NCI109	4127	2	29.7	32.1	Chemical compounds
COLLAB	5000	3	74.5	2457.8	Scientific collaboration
IMDB-BINARY	1000	2	19.8	96.5	Movie collaboration
REDDIT-MULTI-5K	4999	5	508.5	594.9	Online discussion
REDDIT-MULTI-12K	11929	11	391.4	456.9	Online discussion

Graph filtration: vertex-based

- Let $f: V \to \mathbb{R}$ be a function defined on vertices (vertex weights)
- Sublevel graphs $G_\alpha = (V_\alpha, E_\alpha)$

$$V_\alpha = \{v \in V : f(v) \leq \alpha\},$$
$$E_\alpha = \{(u, v) \in E : u, v \in V_\alpha\}$$

- Possible choices of f:
  - given vertex label
  - vertex degree, curvature
  - heat kernel signature, etc.

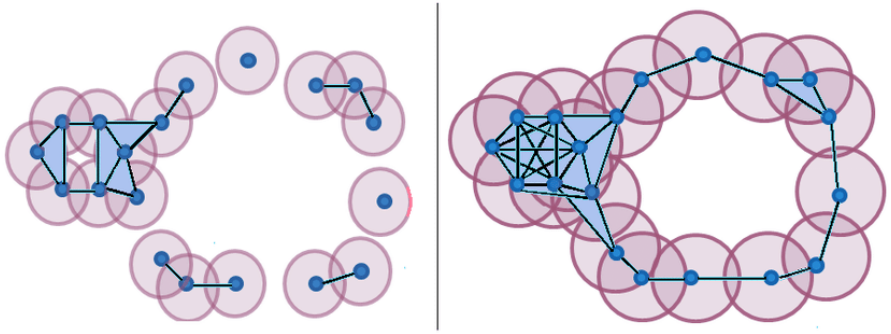


Graph filtration: edge-based

- Let $w: V \times V \to \mathbb{R}$ be a function defined on edges (edge weights)
- Subgraphs $G_\alpha = (V, E_\alpha)$

$$E_\alpha = \{(u, v) \in E : w(u, v) \leq \alpha\}$$

- Possible choices of w: given edge weight, distance by node embedding, etc.



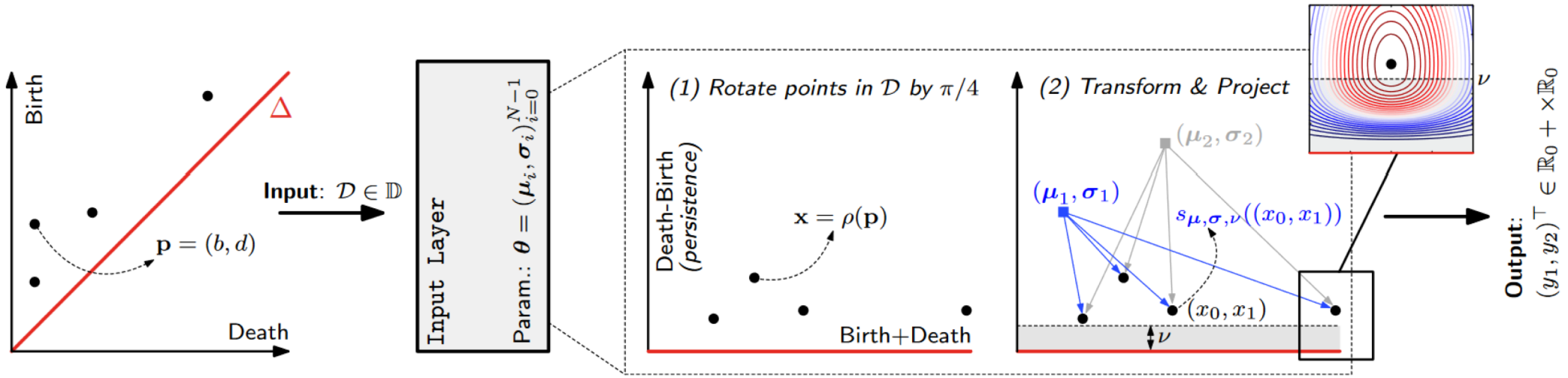
Persistent Homology in Machine Learning

Persistence diagram is a set of points: (birth, death). How to utilize the unusual structure of topological signatures in machine learning?

- Embed PD to a vector space
  - Vectorization: Persistence Image [Adams et. al, 2015], Persistence Landscape [Bubenik et. al, 2015] maps PD to a k-dimensional vector
  - Learning Representation: train a neural network to learn the embedding
- Define a distance or kernel
  - Define a distance between PDs: bottleneck distance, k-Wasserstein distance
  - Positive definite kernels: Slice Wasserstein kernel [Kolouri et. al, 2015], Persistence Fisher kernel [Le et. al., 2018]

Deep Learning with Topological Signatures [Hofer et. al., 2017]

- Drawback of previous approaches:
  - Vectorization: pre-defined, suboptimal, agnostic to any specific task
  - Kernel methods: suffer scalability issues
- How to find a task-optimal representation of topological signatures?



Deep Learning with Topological Signatures [Hofer et. al., 2017]

- Results on graph classification: considerably outperforms state-of-the-art methods (metric: accuracy)

	reddit-5k	reddit-12k
GK [31]	41.0	31.8
DGK [31]	41.3	32.2
PSCN [24]	49.1	41.3
RF [4]	50.9	42.7
Ours (w/o essential)	49.1	38.5
Ours (w/ essential)	**54.5**	**44.5**

WIP: Use multi-filtration input

- Use return probabilities of random walks with different hops to build multiple filtrations, each captures the local structure at different scales
- Design a neural network to embed multi-PD (RP and RP-W below):
  - Embed each PD to a k-dimensional vector
  - Weight-pooling over PD (weights are trained/fixed to 1)

	COLLAB	REDDIT-B	REDDIT-M5K	REDDIT-M12K	IMDB-B	IMDB-M
RETGK	81.0±0.3	92.6±0.3	56.1±0.5	48.7±0.2	71.9±1.0	47.7±0.2
GCAPS-CNN	77.7±2.5	87.6±2.5	50.1±1.7	-	71.7±3.4	48.5±4.1
PERSLAY	76.4	-	55.6	47.7	71.2	48.8
SV	79.6±0.3	87.8±0.3	53.1±0.2	-	74.2±0.9	49.9±0.3
DEEPTOPO	-	-	54.5	44.5		
RP	-	91.7±1.2	56.6±1.7	49.2±1.5	-	-
RP-W	-	92.8±1.1	57.1±1.5	48.8±1.7	-	-
RP-A	-	-	-	-	-	-

References

https://towardsdatascience.com/persistent-homology-with-examples-1974d4b9c3d0

https://www.ncbi.nlm.nih.gov/pmc/articles/PMC5026243/pdf/nihms777844.pdf

https://www.fujitsu.com/global/documents/about/resources/publications/fstj/archives/vol55-2/paper15.pdf

https://arxiv.org/pdf/1906.05795.pdf