EAGE
INCLUDING SPE International EUROPEC
EMBRACING CHANGE - CREATIVITY FOR THE FUTURE
2019 EAGE ANNUAL
81ST CONFERENCE + EXHIBITION
LONDON | UNITED KINGDOM

Texture Based Classification Of Seismic Image Patches Using Topological Data Analysis

June 6, 2019

3 - 6 JUNE 2019
WWW.EAGEANNUAL2019.ORG

2019 EAGE ANNUAL
81ST CONFERENCE + EXHIBITION
LONDON | UNITED KINGDOM

Abstract 640

Rahul Sarkar[‡] and Bradley J. Nelson

Institute for Computational and Mathematical Engineering
Stanford University

‡ Speaker

Abbreviations

The following abbreviations will appear in this talk in various places.

TDA: Topological Data Analysis

PH: Persistent Homology

I will explain them in this talk.

ML: Machine Learning

SVM: Support Vector Machines

RF: Random Forest

NN: Neural Network

CNN: Convolutional Neural Network

These are machine learning specific terminologies. I'll assume working knowledge of these methods.

Our contribution

- ➢ This is quite possibly the first application of TDA based methods that use **persistent homology** for a seismic imaging application.

More generally...

- ➢ This is quite possibly one of the first applications of TDA based methods that use **persistent homology** for a problem relevant to the oil and gas industry.

Seismic textures

- ➢ In a seismic image, different lithologies often have very different "visual appearances".
- ➢ For example, salt bodies appear different from sedimentary sections.
- ➢ The trained human eye of seismic interpreters can easily detect these differences.

Seismic interpreter's job (simplistic viewpoint)

Segment seismic images based on a combination of

- Seismic texture
- Historical memory
- Geological knowledge

ML challenges — texture classification

Challenges of texture classification

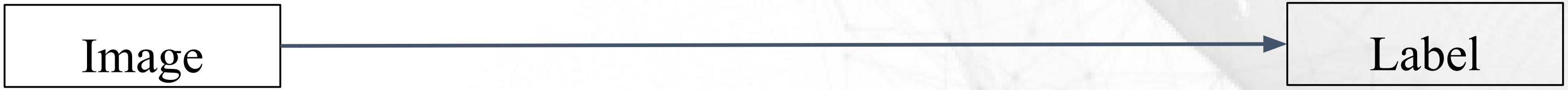
- Areas with similar "look and feel". This can be hard to quantify.
  (Think: I know it when I see it, but can't describe exactly what I'm seeing.)

- Repetitive / recurrent (but not necessarily periodic).

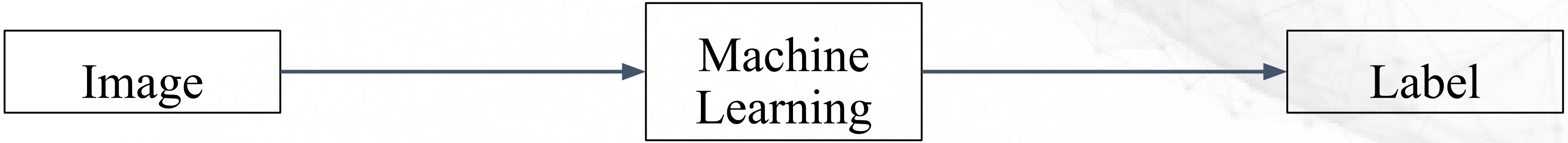
- What kind of features can capture these properties?

Seismic texture classification

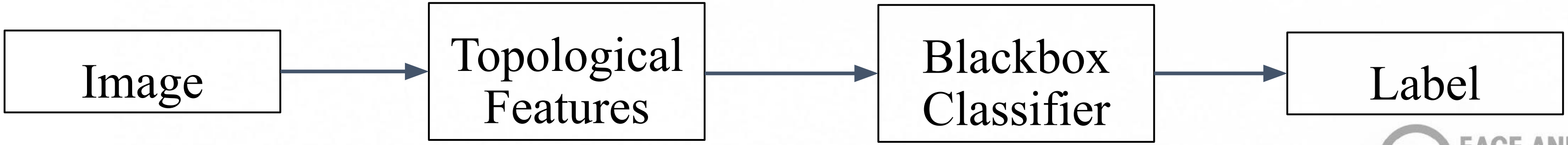
What we want



A popular strategy



Our roadmap



Why topology?

Features of “algebraic topology”

- ➢ Study of topological spaces up to homotopy equivalence (continuous deformation).
- ➢ Identifies quantities that are **scale**, **translation**, **rotation**, and **deformation** invariant.

Topological data analysis

- ➢ Tools to understand topology in data.
- ➢ Turns topological information into features (real numbers), that computers can process.
- ➢ Adapts tools from algebraic topology to study discrete point cloud data.

Continuous deformation of a coffee mug to a doughnut

Simplicial Complex

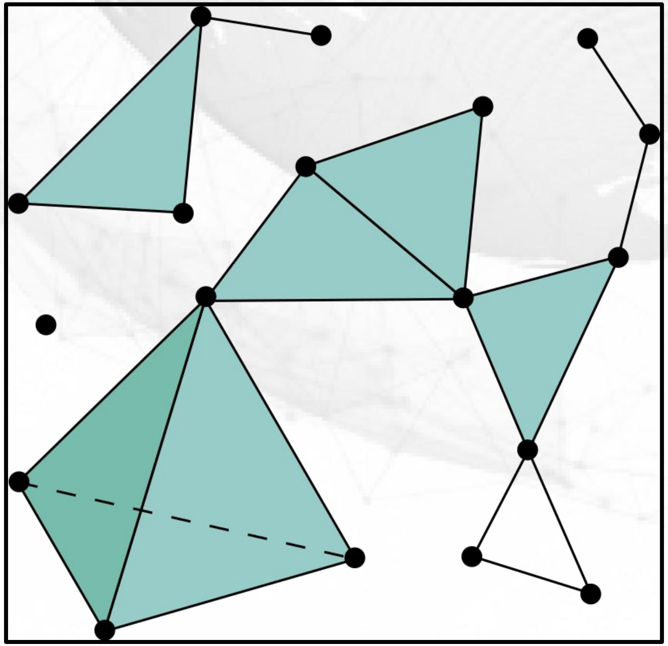
The key topological object (relevant to our work) is a **simplicial complex**. Abstractly this is a triangulation of a topological space.

Definition of a simplicial complex

A set of simplices* (points, lines, triangles, and higher dimensional objects) that satisfy the following two properties:

- ➢ Every face of a simplex is also a simplex.
- ➢ Intersection of any two simplices is a face of each simplex.

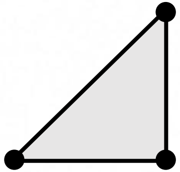
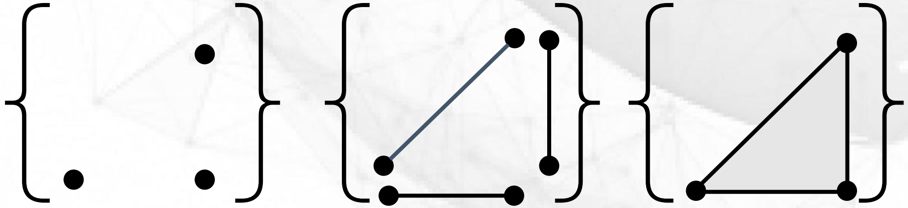
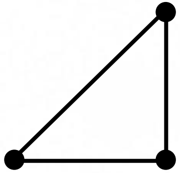
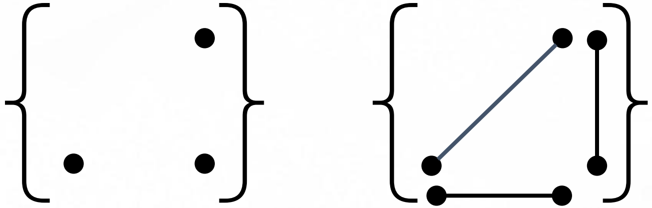
A simplicial complex



Source: Wikipedia

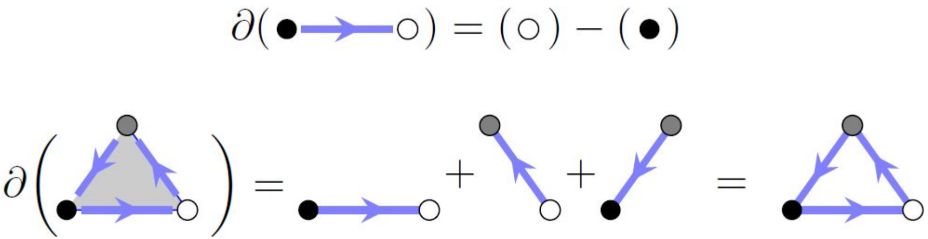
* “Simplices” is the plural of the word “simplex”.

Simplices of a simplicial complex

		Simplicial complex
Topological space		**Simplicial complex**
Filled triangle		**1 - Simplices**  **0 - Simplices** **2 - Simplices**
Triangle with a hole		 **0 - Simplices** **1 - Simplices**

Homology of a simplicial complex

Consider formal linear combinations of vertices / edges / triangles in a simplicial complex X of dimension 2. This produces a set of vector spaces $C_k(X)$ (k = 0 for vertices, k = 1 for edges...). There are linear boundary maps $\partial_k : C_k(X) \rightarrow C_{k-1}(X)$



with the property that $\partial \circ \partial = 0$.

The **k^th^ homology group**, and the **k^th^ Betti number** are defined as

$$H_k(X) = \ker(\partial_k)/\mathrm{img}(\partial_{k+1}), \quad \beta_k = \dim H_k(X).$$

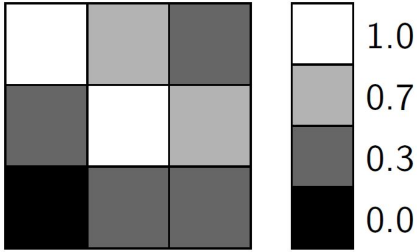
- ➢ β_0 counts clusters that are not connected (called **connected components**).
- ➢ β_1 counts cycles that are not boundaries (called **holes**).

2019 EAGE ANNUAL 81ST CONFERENCE + EXHIBITION LONDON | UNITED KINGDOM

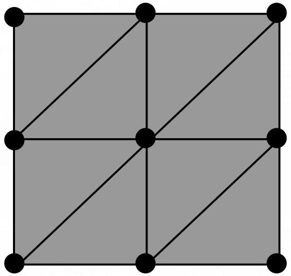
Turning an image into a topological space

One way to do this is to form a simplicial complex as follows:

- ➢ Pixels become points in the space
- ➢ Adjacent pixels are connected by an edge
- ➢ Diagonal edges added by **Freudenthal triangulation**
- ➢ 3 adjacent pixels are spanned by a triangle



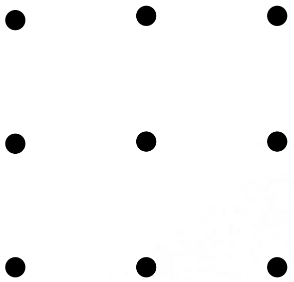
3 x 3 image



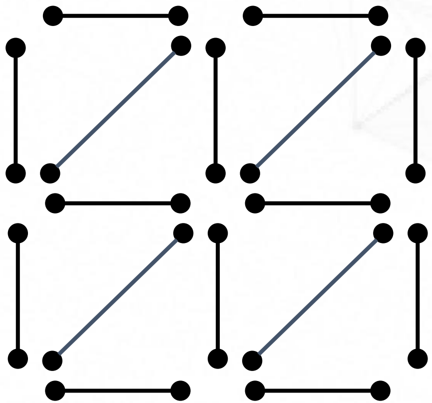
Freudenthal triangulation

Resulting simplicial complex

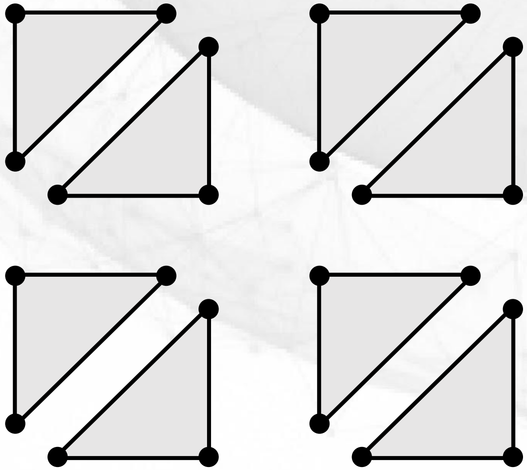
0 - Simplices



1 - Simplices

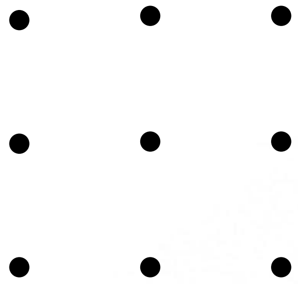


2 - Simplices

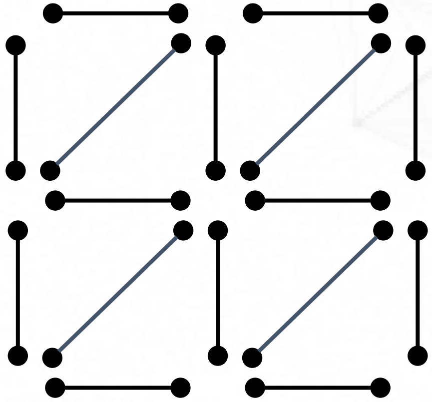


Need for filtered topological spaces

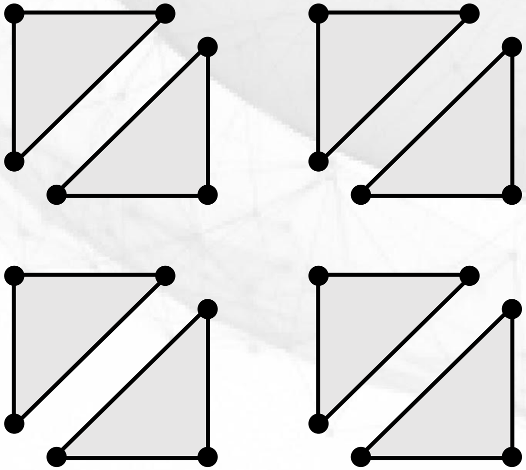
0 - Simplices



1 - Simplices



2 - Simplices

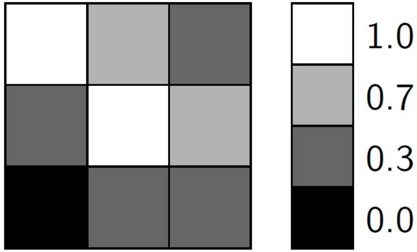


Problem: Topological spaces created from all pixels in the image always generate exactly the same simplicial complex — useless for classification.

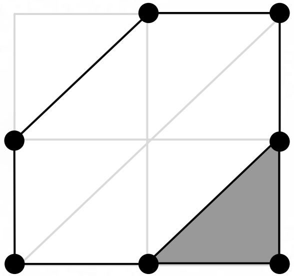
Filtered topological spaces

A more interesting topological space:

- ➢ Choose some pixel value *w*.
- ➢ Only points with pixel values $\leq w$ are used.
- ➢ Only edges with both endpoints are included.
- ➢ Only triangles with boundary edges are included.



3 x 3 image



Topological space at *w* = 0.7

Filtration and persistence

Key ideas

- ➢ Create a sequence of nested topological spaces.
- ➢ Track homology changes across the topological spaces.
- ➢ Turn this information into quantifiable numbers.

Nested topological spaces or **Filtration**

We use a ***sublevel set filtration***.

- ➢ Vary pixel value w from minimum to maximum pixel value.
- ➢ For each w, we construct a filtered topological space X_w.
- ➢ Property: $u \leq w \Rightarrow X_u \subseteq X_w$.

Persistent homology

Persistent homology is the tool that quantifies how homology changes across a filtration.

Input: A filtration $\{X_w\}_w$.

Output: A collection of pairs of real numbers for each homology dimension k, calculated as

$$PH_k(\{X_w\}_w) = \{(b_j, d_j)\}_j.$$

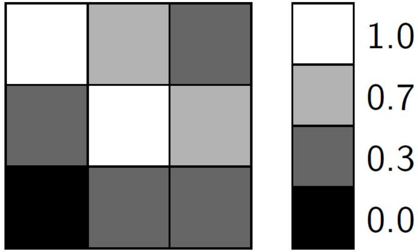
These are called **birth-death pairs**, and track how homology changes over the filtration.

Properties:

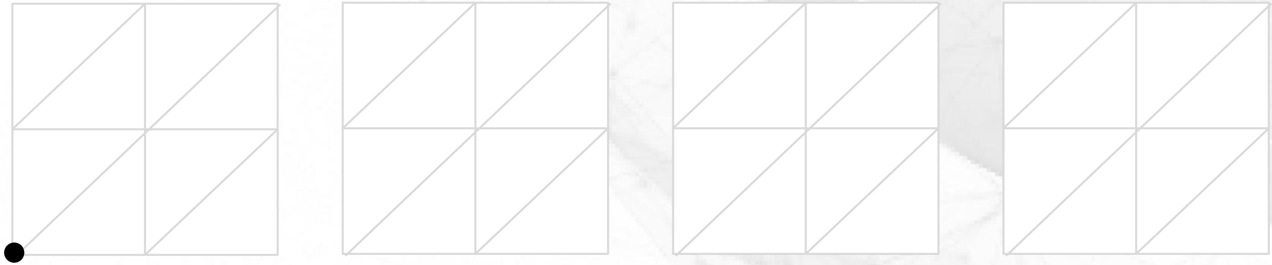
- Homotopy invariant (deformation, rotation, translation).
- Stable to perturbations of pixel values.

Example of how a filtration is built

Example Image



Corresponding Filtration



$w = 0$

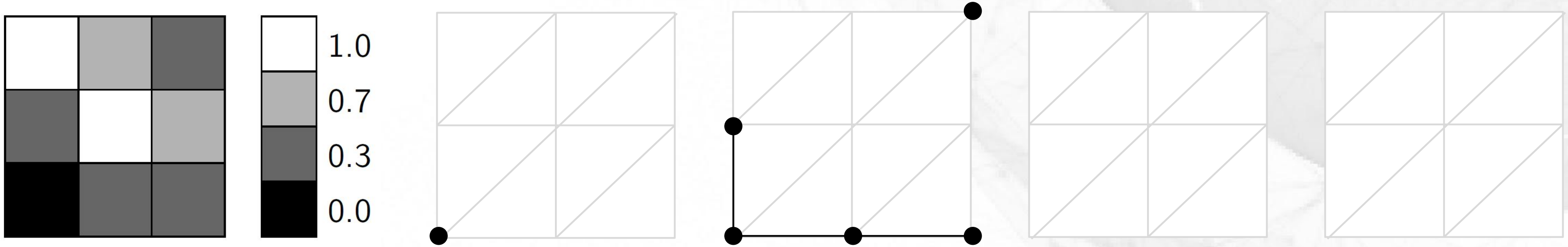
$\beta_0 = 1, \beta_1 = 0$

At $w = 0$, a single point appears, and H_0 homology is born.

Example of how a filtration is built

Example Image

Corresponding Filtration



$w = 0$

$\beta_0 = 1, \beta_1 = 0$

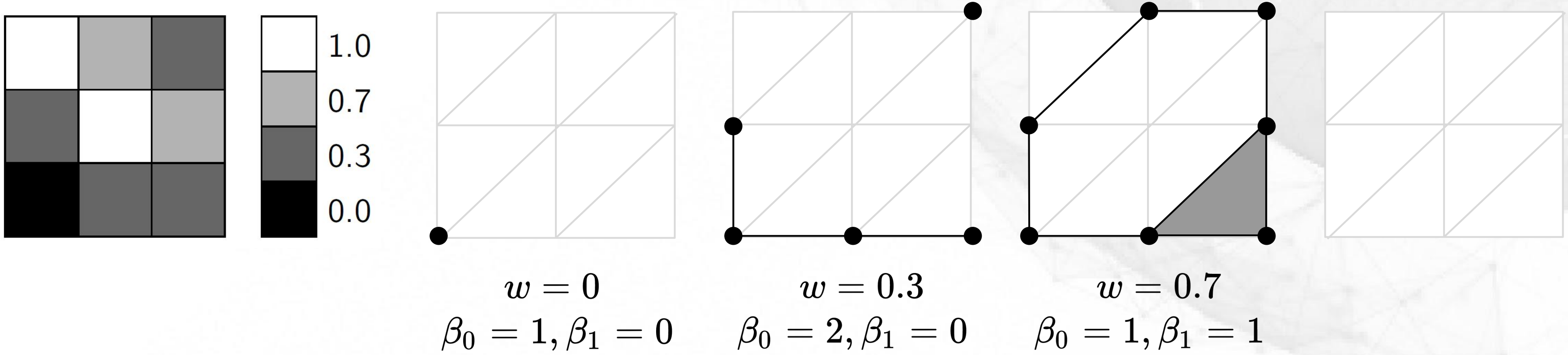
$w = 0.3$

$\beta_0 = 2, \beta_1 = 0$

At $w = 0.3$, several points connect to the first point, and a new component emerges. H_0 homology is born one more time.

Example of how a filtration is built

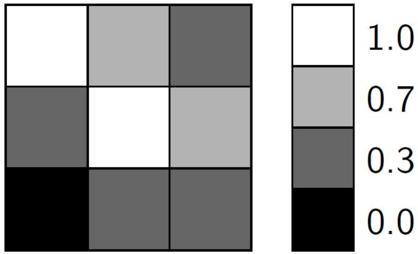
Example Image | **Corresponding Filtration**

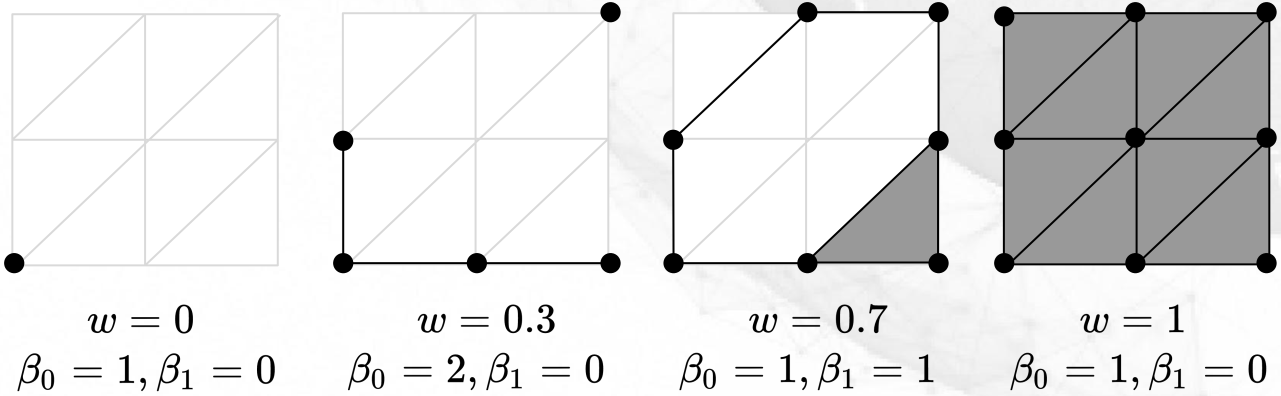


At $w = 0.7$, the two components join, and a hole appears. We also see our first triangle. So H_0 homology has died, while H_1 homology is born.

Example of how a filtration is built

Example Image — **Corresponding Filtration**

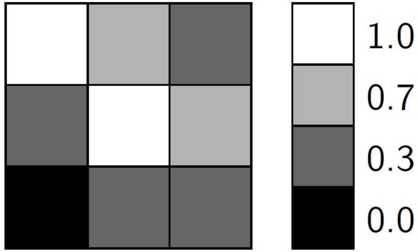




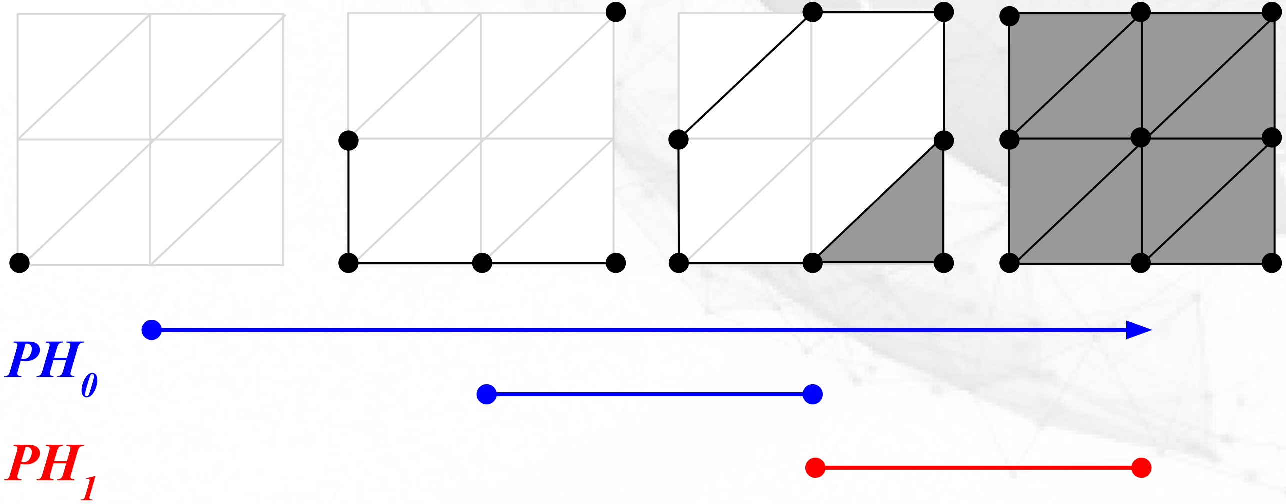
At $w = 1$, all points are now present, and all edges and triangles fill in the space.
The hole has now disappeared, and so H_1 homology has died.

Example of how a filtration is built

Example Image



Corresponding Filtration



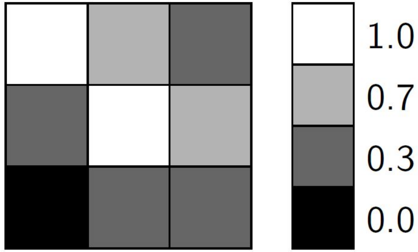
Persistence Barcode:

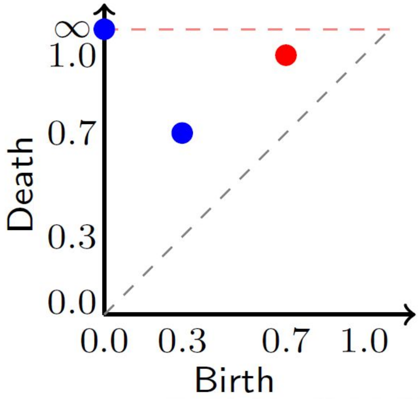
Information about how components appear and merge is encoded in PH_0.

Information about how 1D holes appear and fill is encoded in PH_1.

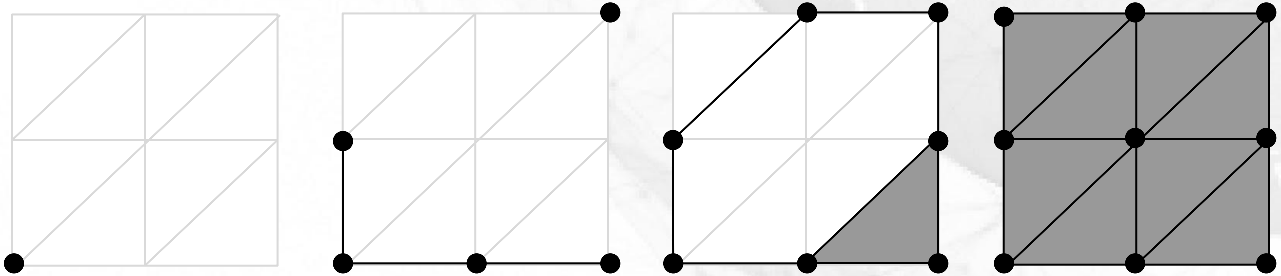
Example of how a filtration is built

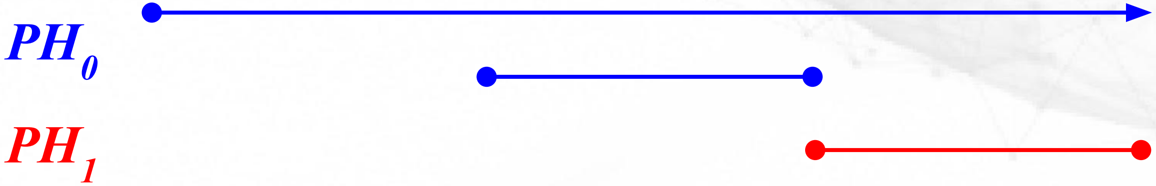
Example Image





Corresponding Filtration





Persistence Diagram:

The start and endpoints of the barcode are plotted in the plane.

Each point is referred to as a **birth-death** pair.

Applications on a real 2D dataset

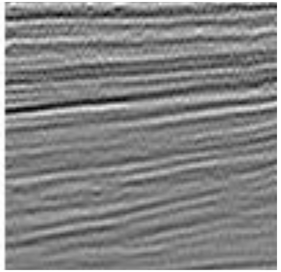
For the rest of this talk we will use the LANDMASS[↟] dataset to demonstrate the workflow and our results. This is a publicly available dataset of **two sets** of labeled 2D seismic image patches, each with **4 classes**.

	LANDMASS-1	LANDMASS-2
Image Size (pixels)	99 x 99	150 x 300
Class Names	**Number of Images**	**Number of Images**
1. Horizons	9385	1000
2. Chaotic Horizons	5140	1000
3. Fault Patches	1251	1000
4. Salt Domes	1891	1000

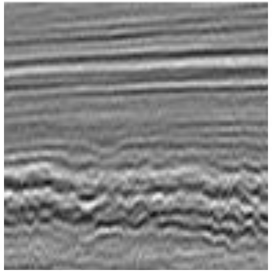
↟Alaudah, Y., Wang, Z., Long, Z. and AlRegib, G. [2015] LANDMASS Seismic Dataset.

Sample images (images not to scale)

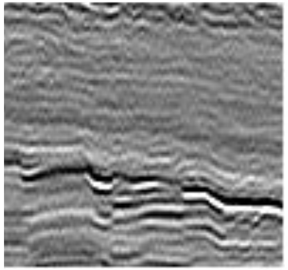
LANDMASS-1



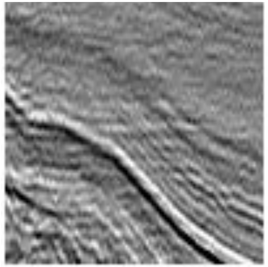
Horizons



Chaotic Horizons

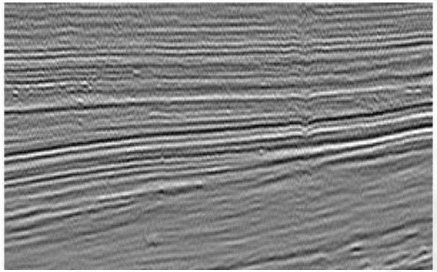


Fault Patches

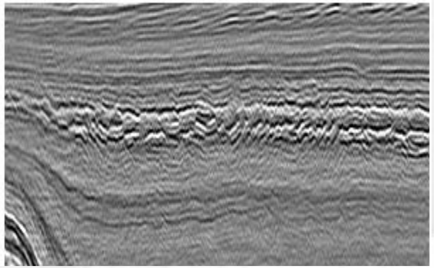


Salt Domes

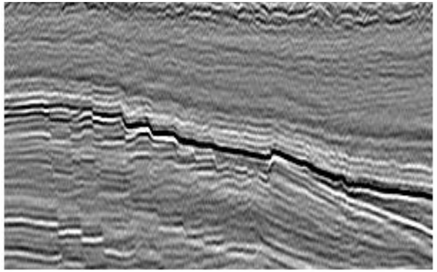
LANDMASS-2



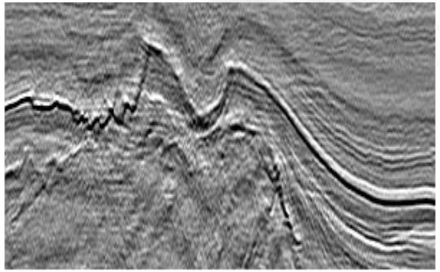
Horizons



Chaotic Horizons



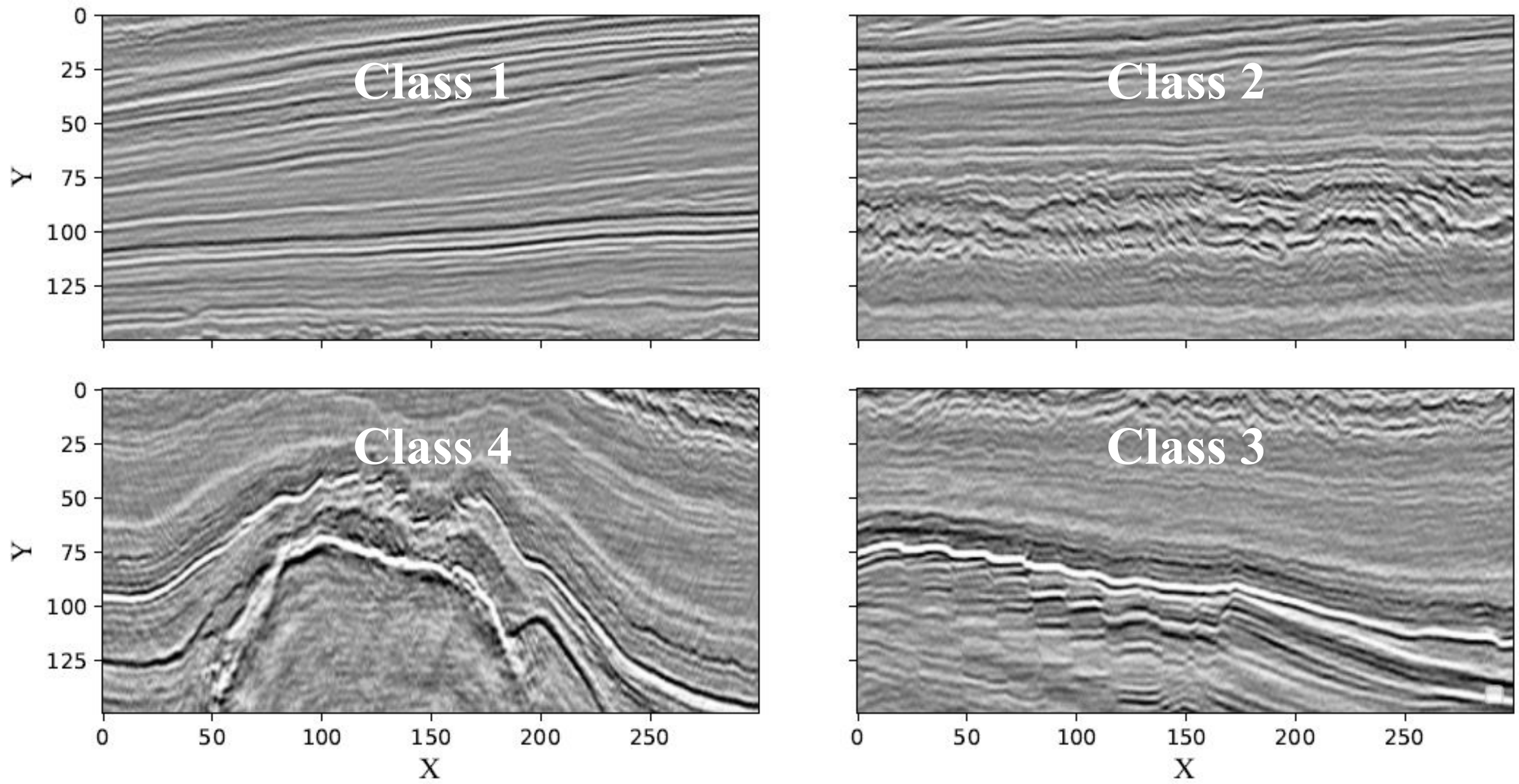
Fault Patches



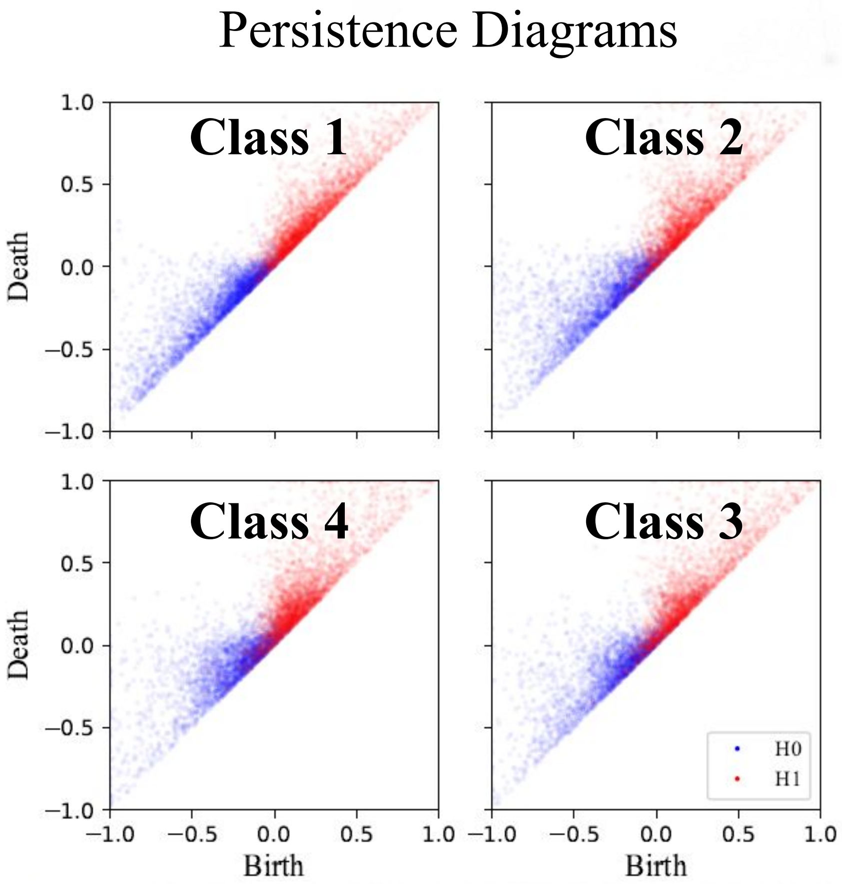
Salt Domes

Persistence diagram results (LANDMASS-2)

Sample Images



Persistence diagram results (LANDMASS-2)



Subtle differences between the persistence diagrams.

To train a classifier we need:

- ➢ Statistically significant **intra-class similarity**.
- ➢ Statistically significant **inter-class dissimilarity**.

Currently working on how to make this more precise, and generate metrics.

Need for featurization of persistence diagrams

We want to use a machine learning (ML) approach for training a classifier based on the persistence diagrams.

So far: 2D Images → Persistence Diagrams

Key points about the persistence diagrams:

- Every image produces a different number of birth-death pairs.
- We want a ***standard number of features*** for a ML workflow.

Polynomial featurization

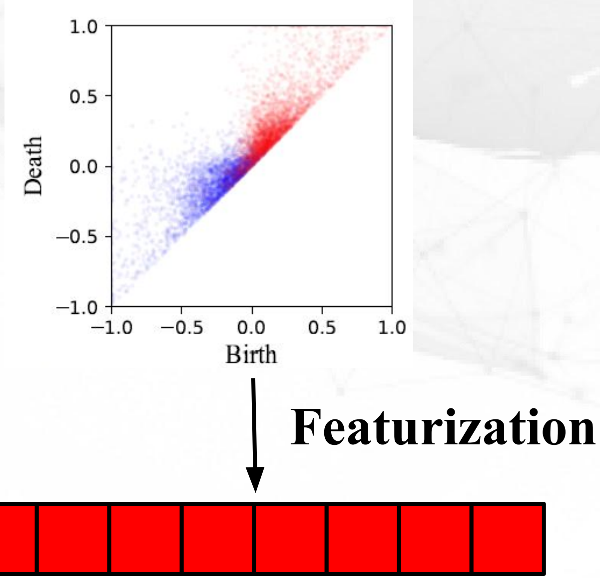
One approach is based on polynomial functions[†], which we adopt in our work:

$$p(\alpha; \{b_i, d_i\}_{i \in J}) = \frac{1}{|J|} \sum_{i \in J} \sum_{j,k} \alpha_{j,k} (d_i - b_i)^j (d_i + b_i)^k.$$

For both homology dimensions 0 and 1 we choose:

$$\alpha_{j,k} = \delta_{j=j_0, k=k_0}$$

where $(j_0, k_0) \in \{0,1,2,3\}^2 - \{(0,0)\}$.

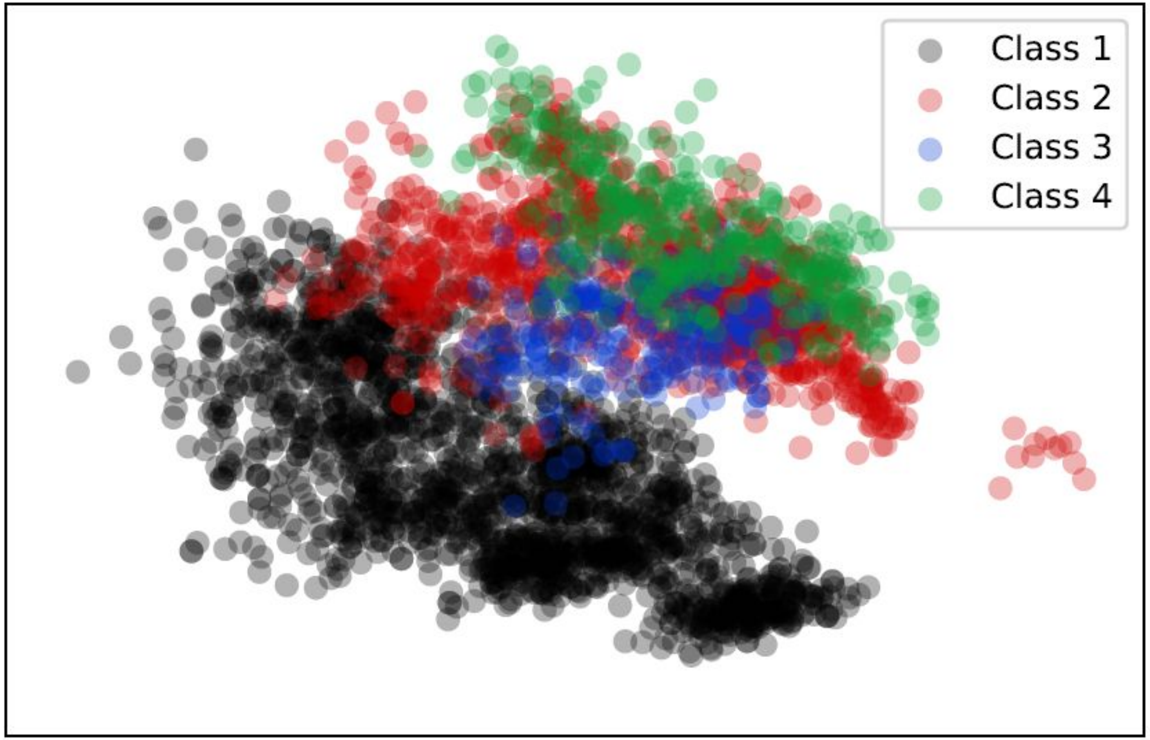


This gives us a total of 15 x 2 = 30 features per persistence diagram.

† A. Adcock, E. Carlsson, G. Carlsson. The ring of algebraic functions on persistence barcodes. Homology, Homotopy and Applications. 18(1) 2016.

LANDMASS-1 features

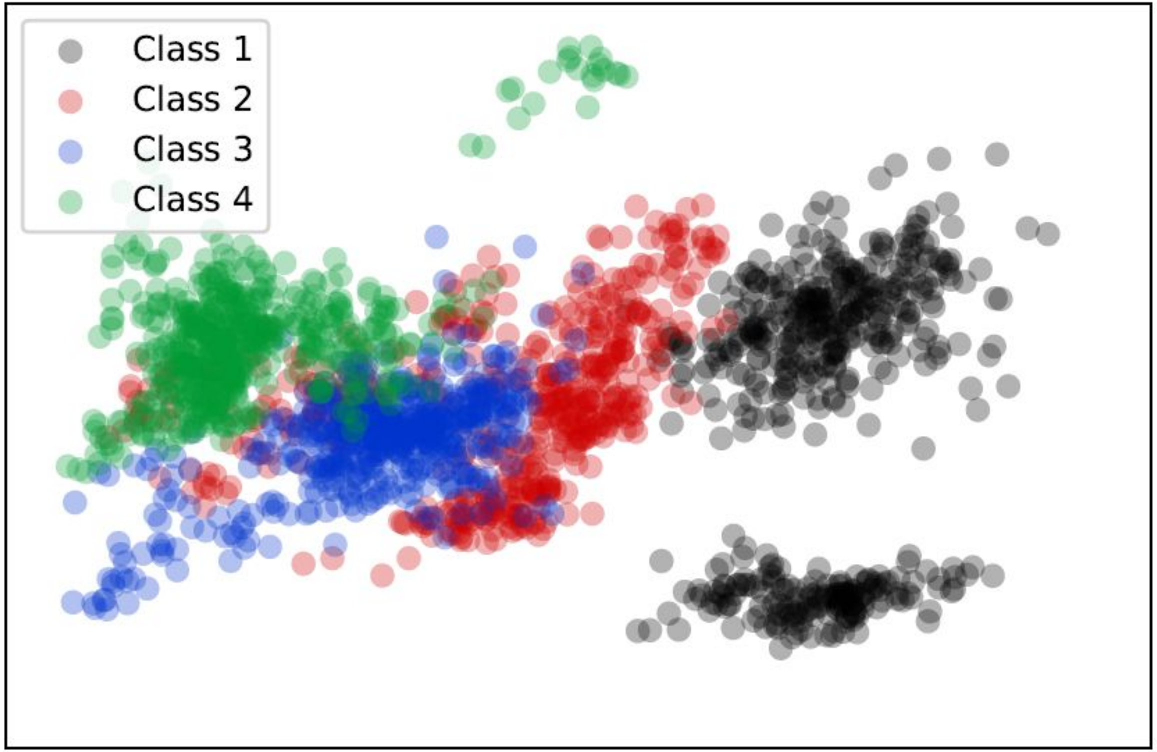
Projection of polynomial features into top two principal components. Each point is an image in the LANDMASS-1 dataset.



- Class 1 separates nicely from the other classes.
- With 2 principal components, classes are not well separated.
- More components are needed.

LANDMASS-2 features

Projection of polynomial features into top two principal components. Each point is an image in the LANDMASS-2 dataset.



- ➢ Classes reasonably well separated with just top 2 principal components.

- ➢ Equal class sizes help classification.

ML workflow

Split data into train (70%) and test (30%) sets, per class, randomly.

Produce persistence diagrams for each image.

Produce polynomial features from each persistence diagram.

Train and test blackbox classifiers on polynomial features.

Three algorithms tested:

- Multiclass SVM
- RF
- NN

Derived attribute image based ML workflow

Split data into train (70%) and test (30%) sets, per class, randomly.

Create derived attribute images from the raw images **(e.g. root mean square amplitude, GLCM* cubes)**

Produce persistence diagrams for each image.

Produce polynomial features from each persistence diagram.

Train and test blackbox classifiers on polynomial features.

Three algorithms tested:

- Multiclass SVM
- RF
- NN

* GLCM: Gray-Level Co-Occurrence Matrix

Classification results: Multiclass SVM classifier

Attribute	SVM Accuracy
Raw Image	99.8 / 75.2 / 0.0 / 0.0 100.0 / 55.0 / 88.3 / 74.3
GLCM Mean	100.0 / 18.6 / 34.1 / 29.3 62.7 / 19.0 / 4.0 / 100.0
RMS Amplitude	100.0 / 1.0 / 0.0 / 0.0 74.7 / 85.7 / 71.3 / 61.7
GLCM Correlation	100.0 / 0.0 / 0.0 / 0.0 64.7 / 32.0 / 89.3 / 32.3
GLCM Variance	96.6 / 94.1 / 92.8 / 67.7 97.3 / 93.3 / 91.7 / 87.0

Class 1 / Class 2 / Class 3 / Class 4

Top Row: **LANDMASS-1**

Bottom Row: **LANDMASS-2**

Classification accuracy of **raw image, and best 4 attributes** with respect to RF classifier.

- ➢ Linear classifiers like SVM perform poorly.
- ➢ Need nonlinear decision boundaries.

Classification results: RF classifier

Attribute	RF Accuracy
Raw Image	99.9 / 98.6 / 95.2 / 93.3 100.0 / 98.0 / 100.0 / 96.3
GLCM Mean	99.9 / 97.9 / 82.1 / 93.3 100.0 / 97.0 / 97.3 / 91.7
RMS Amplitude	99.3 / 96.1 / 88.0 / 82.0 99.7 / 96.0 / 96.0 / 91.7
GLCM Correlation	99.3 / 94.9 / 80.8 / 91.2 99.7 / 93.7 / 92.0 / 97.0
GLCM Variance	98.5 / 95.7 / 96.3 / 74.0 99.0 / 95.3 / 96.7 / 89.7

Class 1 / Class 2 / Class 3 / Class 4

Top Row: **LANDMASS-1**

Bottom Row: **LANDMASS-2**

Classification accuracy of **raw image, and best 4 attributes** with respect to RF classifier.

- ➢ Nonlinear classifiers do much better.

Classification results: NN classifier

Attribute	NN Accuracy
Raw Image	100.0 / 99.6 / 99.7 / 98.4
	100.0 / 100.0 / 99.0 / 95.0
GLCM Mean	100.0 / 97.8 / 92.8 / 97.0
	100.0 / 96.0 / 95.7 / 96.3
RMS Amplitude	99.5 / 99.1 / 96.3 / 91.5
	99.7 / 99.0 / 93.7 / 91.3
GLCM Correlation	99.8 / 93.6 / 87.7 / 96.7
	100.0 / 95.7 / 93.7 / 98.3
GLCM Variance	99.3 / 98.3 / 98.1 / 87.3
	99.7 / 99.0 / 99.3 / 95.0

Class 1 / Class 2 / Class 3 / Class 4

Top Row: **LANDMASS-1**

Bottom Row: **LANDMASS-2**

Classification accuracy of **raw image, and best 4 attributes** with respect to RF classifier.

➢ Nonlinear classifiers do much better.

Conclusions

- TDA derived features perform well for texture classification in seismic images.
- Nonlinear decision boundary classifiers are necessary for good classification accuracy.
- These features could augment existing ML workflows for similar tasks.

Software used in this study

- **GUDHI**[1] in Python — persistent homology calculations.
- **Scikit-learn**[2] in Python — SVM and RF classifiers.
- **Tensorflow**[3] in Python — NN classifier.

[1] C. Maria, "Filtered Complexes, GUDHI User and Reference Manual", http://gudhi.gforge.inria.fr/doc/latest/group simplex tree.html, 2015.

[2] F. Pedregosa et al., "Scikit-learn: Machine Learning in Python", Journal of Machine Learning Research 12, 2011.

[3] M. Abadi et al., "TensorFlow: Large-Scale Machine Learning on Heterogeneous Systems", Whitepaper, https://www.tensorflow.org/, 2015.

Acknowledgments

We would like to thank our advisors **Biondo Biondi**[⇟‡] and **Gunnar Carlsson**[⇟↟] for mentoring, and providing helpful suggestions along the way.

Disclosure of funding:

- Rahul Sarkar was partially funded by the Stanford Exploration Project for the duration of this study.
- Bradley J. Nelson was partially funded by the US DoD NDSEG fellowship program.

⇟ *Institute for Computational and Mathematical Engineering, Stanford University*

‡ *Department of Geophysics, Stanford University*

↟ *Department of Mathematics, Stanford University*

Questions

Thank you for listening!

Questions?

If you need more information contact us by email at:

rsarkar@stanford.edu, bjnelson@stanford.edu