A Topological Reading Lesson - TDA Pipeline for Image Analysis

Adélie Garin and Guillaume Tauzin
Laboratory for Topology and Neurociences, EPFL
adelie.garin@epfl.ch guillaume.tauzin@epfl.ch

Motivation

- Lots of implemetations exist to apply Topological Data Analysis to a wide range of datasets... but it is hard to combine them with existing data analysis and machine learning pipelines.
- Image datasets provide a very intuitive use case as their topological features are easy to interprate... but they are hard to extract.
- Similar ideas and technics can be applied to networks, and the pipeline can be modified to be adapted to different input data types.

Summary

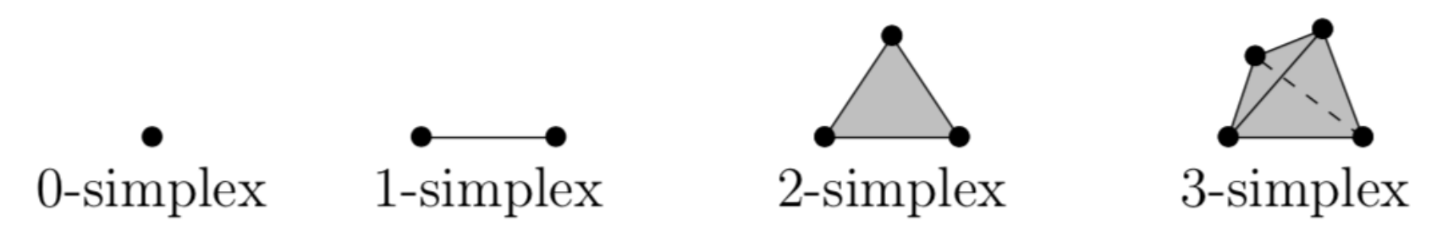
- We use Topological Data Analysis (TDA) for machine learning tasks on grayscale images.
- We build a generic pipeline combining a wide range of different TDA techniques for images based on different filtrations and diagram features.
- We conduct a feature selection and study their correlations while providing an intuitive interpretation of their importance, which is relevant in both machine learning and TDA.
- This interpretation allows us to chararcterise the geometric and topological differences between images of different labels.
- We show that this topological machine learning pipeline can be used as a highly relevant dimensionality reduction by applying it to the MNIST digits dataset.

Mathematical Background

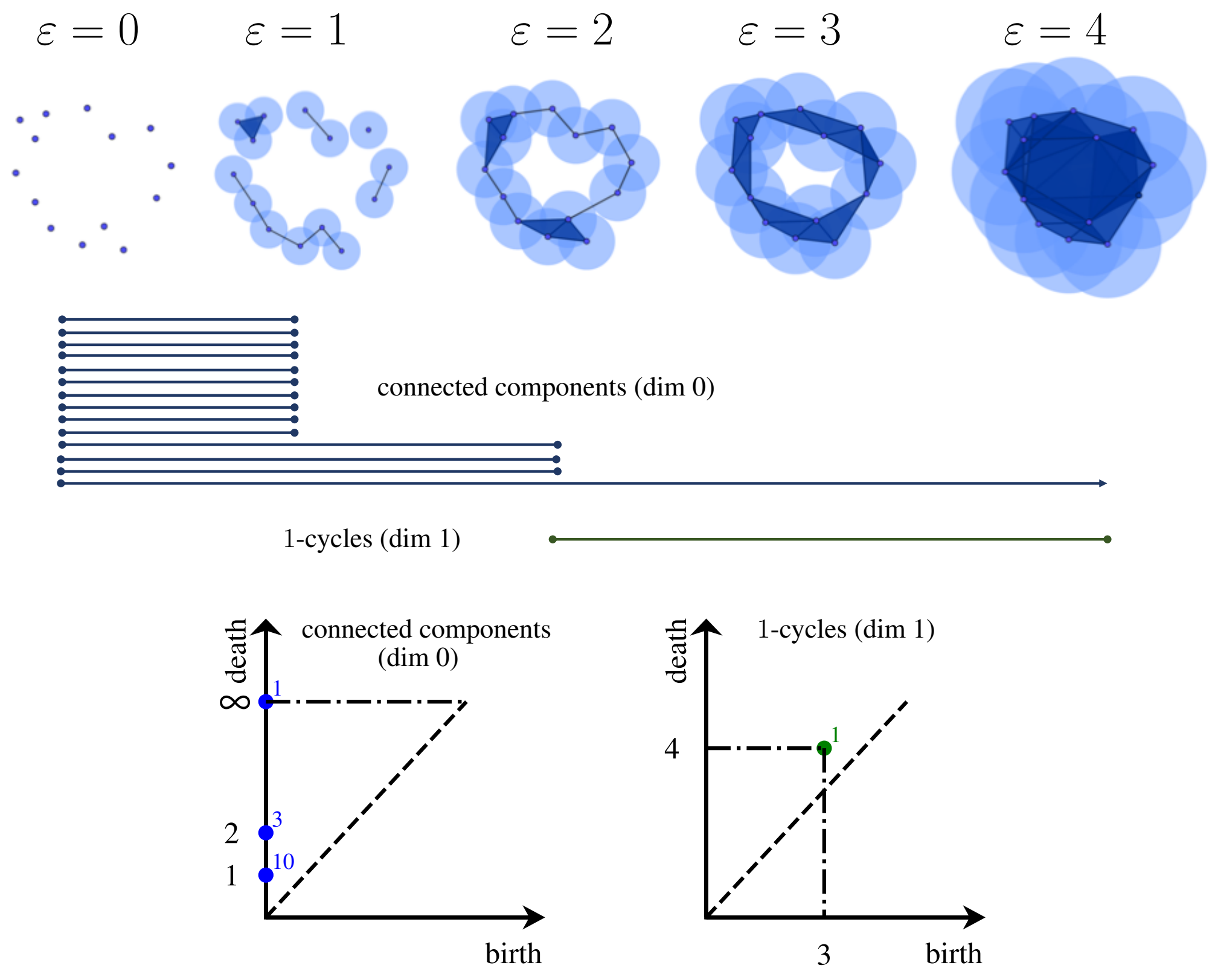
Persistent homology studies the evolution of *topological features* (connected components, loops, cavities,...) throughout an iterative process called a *filtration*, *i.e.* a nested sequence of objects (complexes): $X_1 \subseteq X_2 \subseteq \ldots X_n$.
Output: A *barcode* containing the birth and death of these features, or a *persistence diagram*, that caries the same information.

Persistent Homology for Point Clouds

Persistent homology on point clouds relies on objects referred as *simplices*, which are the building blocks of the higher-dimensional counterparts of graphs called *simplicial complexes*. A k-simplex is the convex hull of $k+1$ affinely independent points:



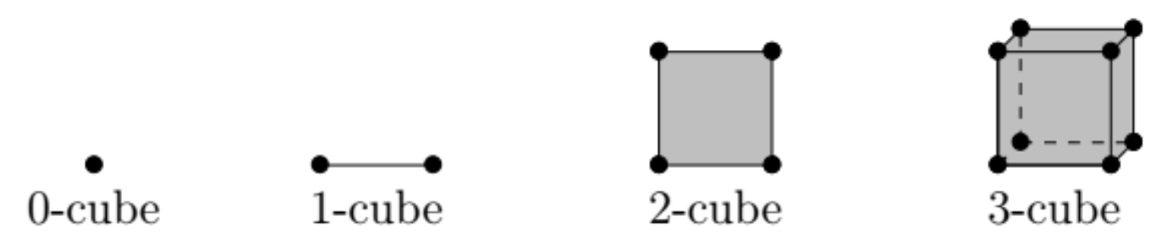
Extracting topological features from a data set X:
The *Vietoris-Rips complex* of parameter ε, $VR(X, \varepsilon)$, is the simplicial complex with vertex set X and where $\{x_0, x_1, ..., x_k\}$ spans a k-simplex if and only if $d(x_i, x_j) \leq \varepsilon$ for all $0 \leq i, j \leq k$.
As ε grows, so does the Vietoris-Rips complex of a point cloud. This defines a filtration of simplicial complexes.



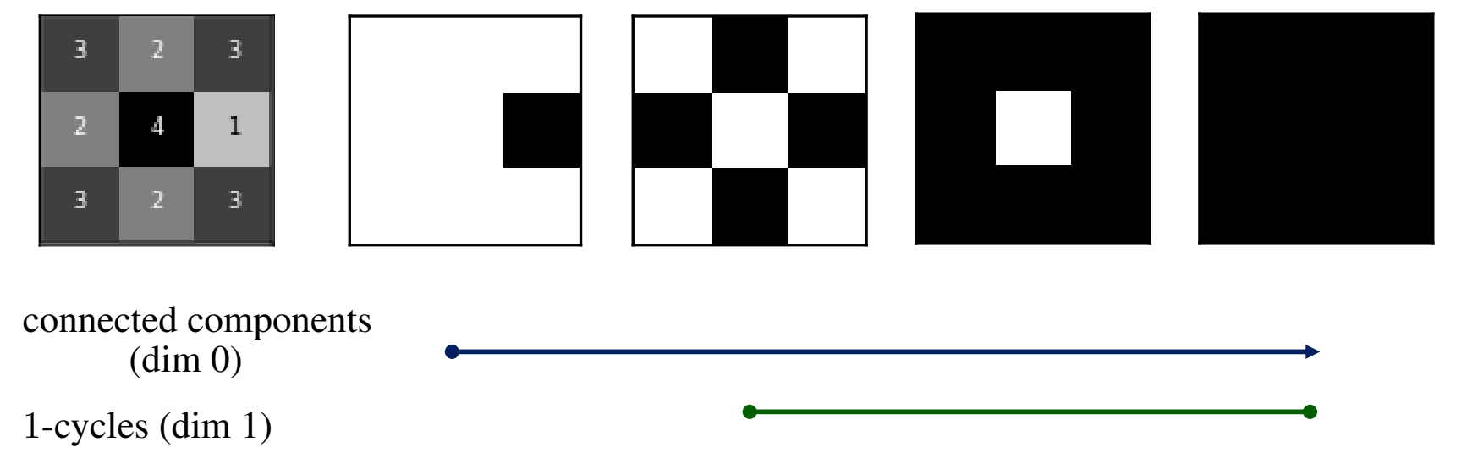
A point cloud and the Vietoris-Rips filtration induced (top figure). We only take five values $\varepsilon = 0, 1, 2, 3$ and $\varepsilon = 4$, where everything is filled in. The birth-death pairs of connected components are indicated in blue and the 1-cycles (loops) pair in green in both the barcodes (middle figure) and the persistence diagrams (bottom figure).

Persistent Homology for Grayscale Images

Images are made of pixels (or voxels in higher dimension). The cubical analog of a simplicial complex is a cubical complex, in which the role of simplices is played by cubes of different dimensions. A finite *cubical complex* is a union of cubes aligned on the grid $\mathbb{Z}^d$.



A grayscale image comes with a natural filtration embedded in the grayscale values of its pixels, where the cubical complex grows as the value of the grayscale increases.



An example of a grayscale image and the induced filtration. The barcode is indicated below, the blue bar stands for the connected components and the green one for the 1-cycle (loop).

Filtrations for Binary Images

Goal: From a binary image, define different grayscale images to extract topological features using their persistence barcodes.

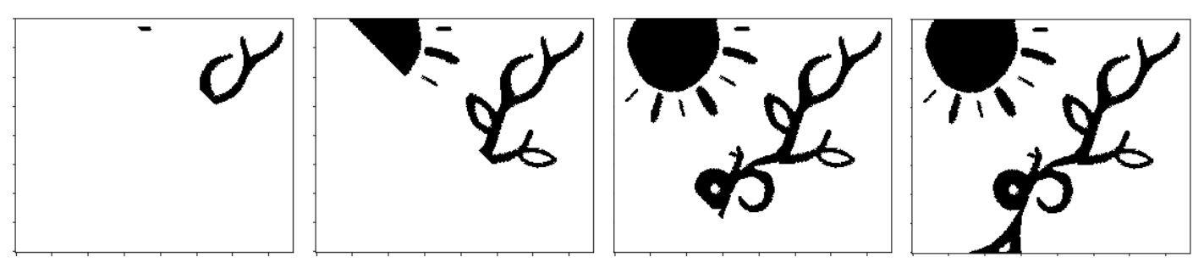
$$\text{Binary image} \rightarrow \text{ Grayscale image} \rightarrow \text{Filtration}.$$

Consider $\mathcal{B} : I \subset \mathbb{Z}^d \longrightarrow \{0, 1\}$ a d-dimensional binary image.

Binary filtration Consider the binary values of the pixels as a two-level filtration.

Height filtration (in direction $v \in S^{d-1}$)

$$\mathcal{H} : I \longrightarrow \mathbb{R}, p \mapsto \begin{cases} \langle p, v\rangle & \text{if } \mathcal{B}(p) = 1 \\ \mathcal{H}_\infty := \max_{p \in I} \langle p, v\rangle & \text{if } \mathcal{B}(p) = 0. \end{cases}$$



An example of some steps of the height filtration in direction $[-1, -1]$.

Radial filtration (of center $c \in I$)

$$\mathcal{R} : I \longrightarrow \mathbb{R}, p \mapsto \begin{cases} \|c - p\|_2 & \text{if } \mathcal{B}(p) = 1 \\ \mathcal{R}_\infty := \max_{p \in I} \|c - p\|_2 & \text{if } \mathcal{B}(p) = 0. \end{cases}$$

Density filtration (of parameter r)

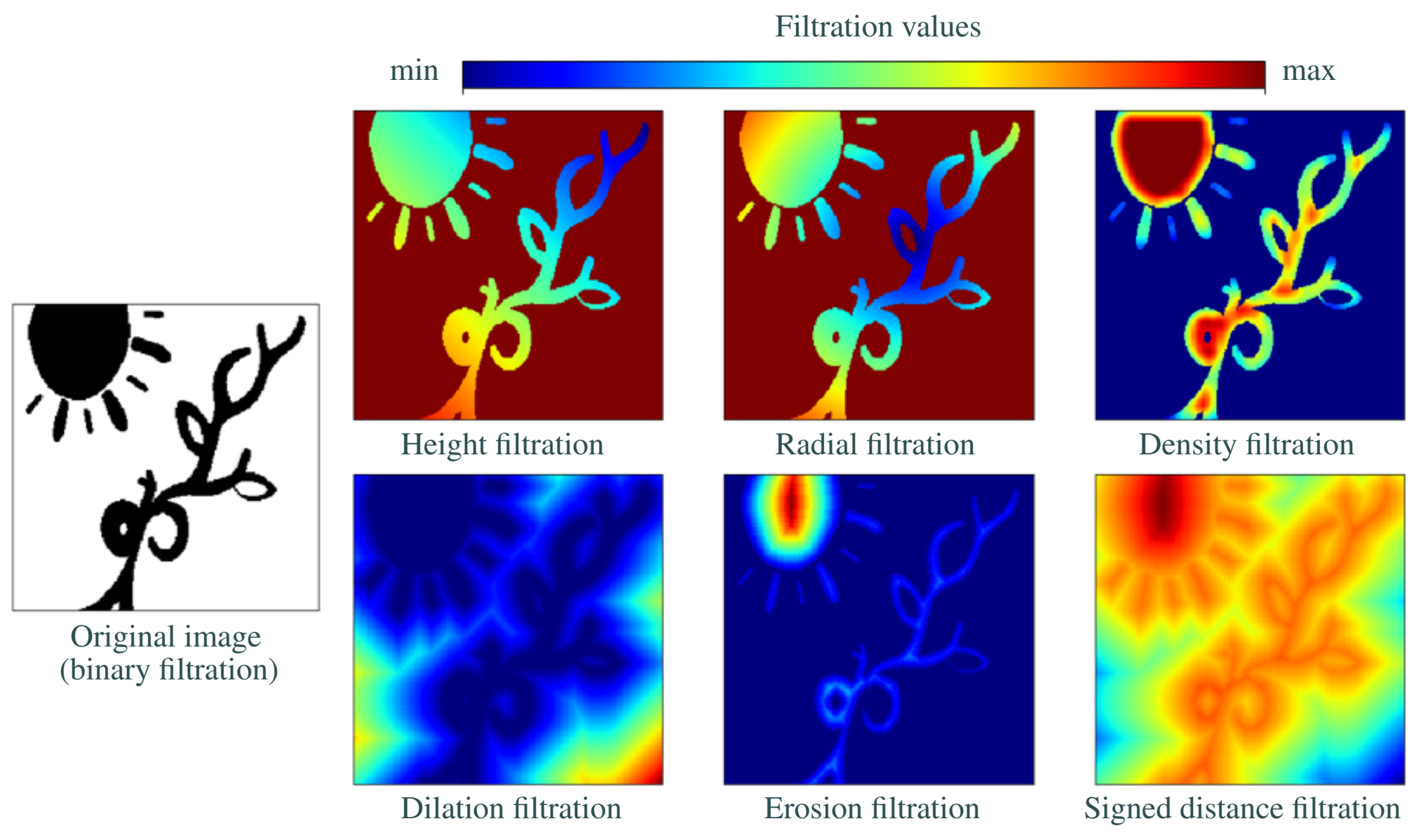
$$\mathcal{D}_e(p) := \#\{v \in I, \mathcal{B}(v) = 1 \text{ and } \|p - v\| \leq r\}.$$

Dilation filtration

$$\mathcal{D}(p) := \min\{\|p - v\|_1, \ \mathcal{B}(v) = 1\}.$$

Erosion filtration
The *erosion filtration* is obtained by applying the dilation filtration to the inverse image (where 0s and 1s are switched).

Signed distance filtration
"Dilation - Erosion = Signed distance"



Original image (binary filtration)
Height filtration
Radial filtration
Density filtration
Dilation filtration
Erosion filtration
Signed distance filtration

An example of each of the filtrations defined above.

TDA Pipeline for Machine Learning

How can we extract features from a persistence diagram?

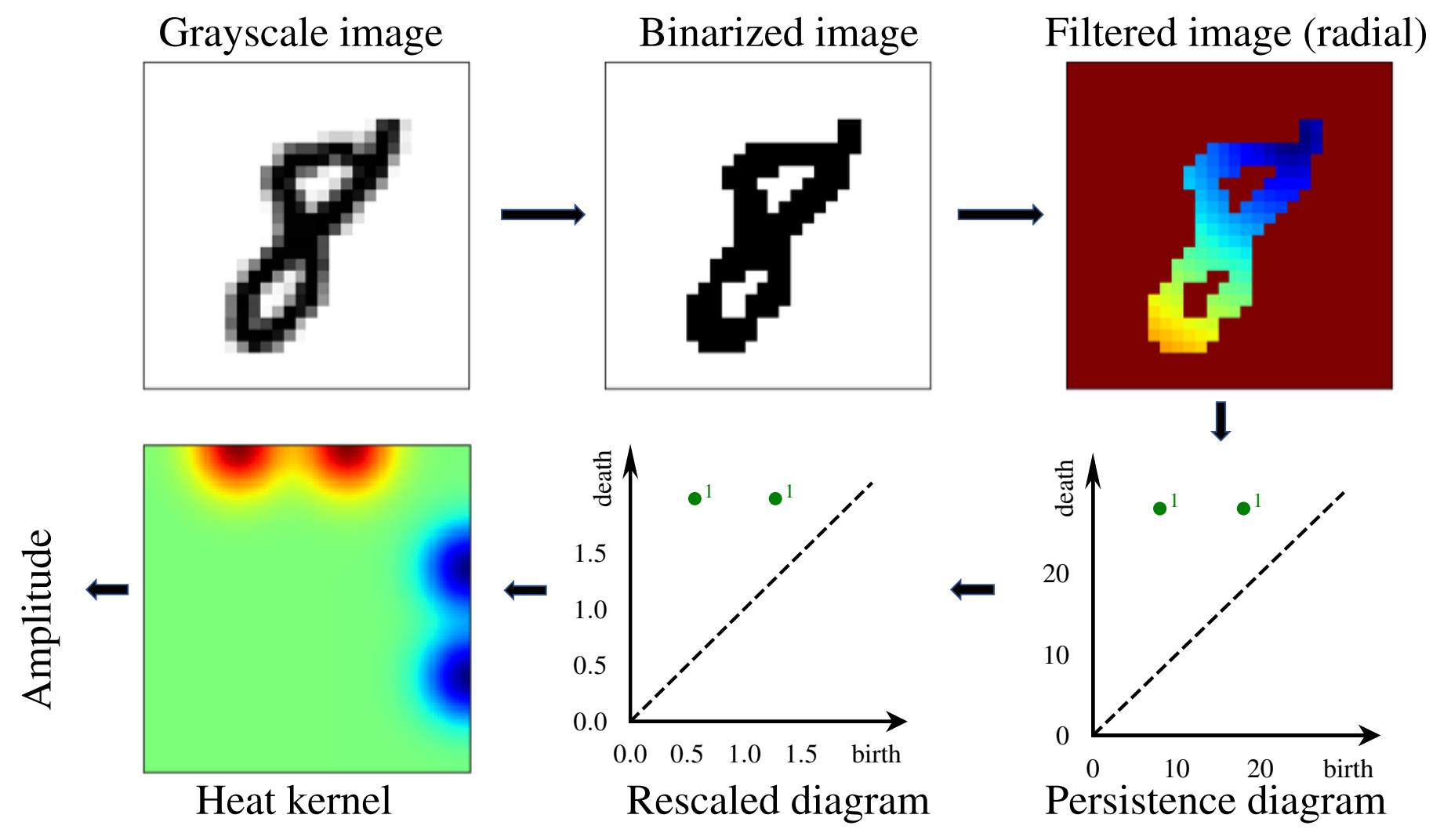
Amplitude: We use *distances* between persistence diagrams (Wasserstein distance, Bottleneck distance, $L1$ and $L2$ norms between their betti curves, their persistence landscapes, and their heat kernel) and the empty diagram.

Persistent Entropy: The *persistent entropy* of a barcode $\{(b_i, d_i)\}_{i=1,...n}$ is a real number extracted by taking the Shannon entropy of the persistence (lifetime) of all cycles:

$$PE(D) = -\sum_{i=1}^{n} \frac{l_i}{L(B)} \log(\frac{l_i}{L(B)}),$$

where $l_i := d_i - b_i$ and $L(B) := l_1+, ... + l_n$.

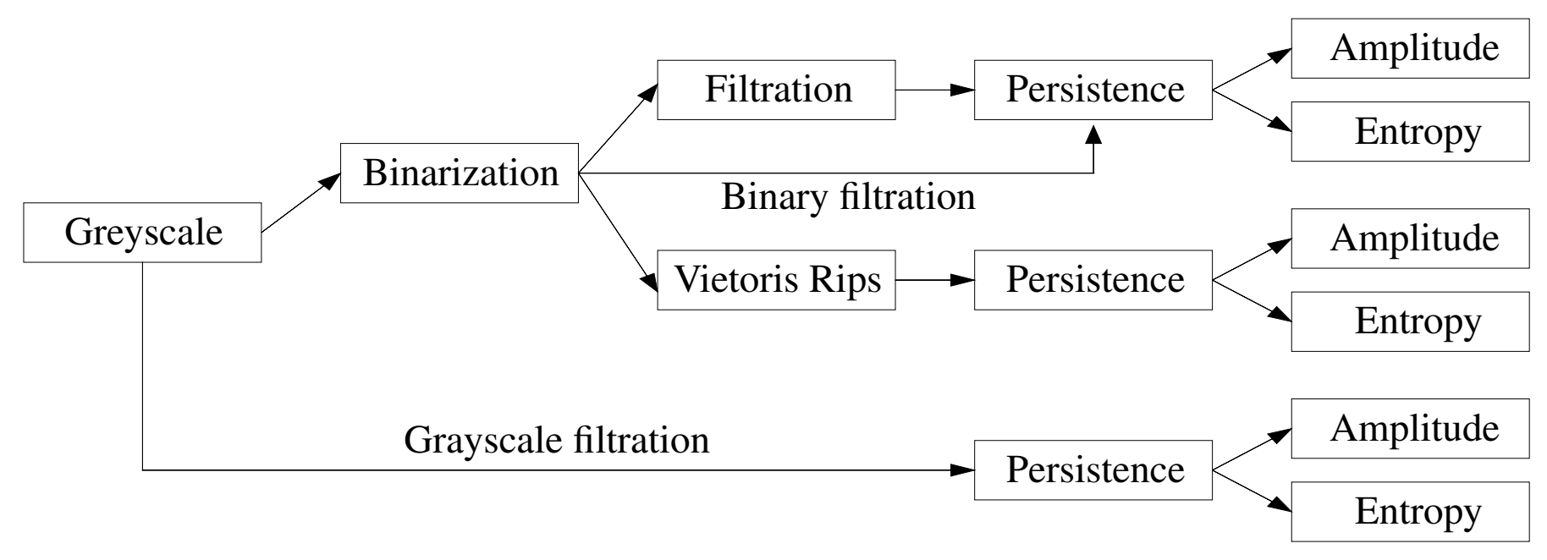
$$\text{Image} \rightsquigarrow \text{Persistence diagram} \rightsquigarrow \text{Features}$$



Heat kernel
Rescaled diagram
Persistence diagram

An example of a feature extracted from an MNIST image (we apply a threshold of 0.4 to obtain a binary image and then proceed with the pipeline).

Generic Generation of Topological Features

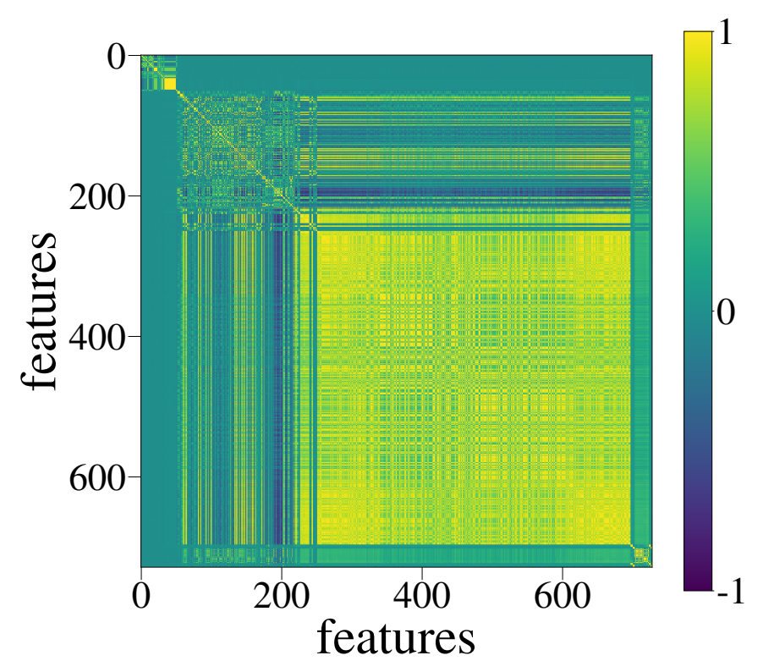


Greyscale
Binarization
Filtration
Persistence
Amplitude
Entropy
Binary filtration
Vietoris Rips
Persistence
Amplitude
Entropy
Grayscale filtration
Persistence
Amplitude
Entropy

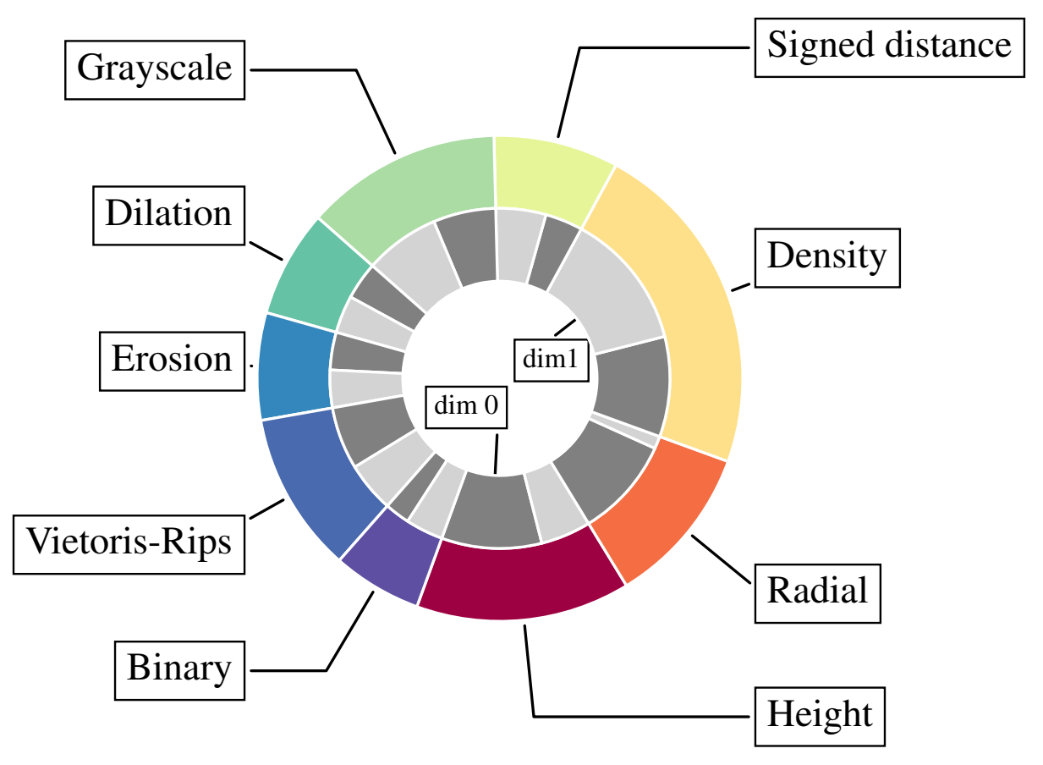
Example: MNIST Analysis

Correlation and feature importances

The features generated on the MNIST dataset are very correlated. To select the most significant decorrelated features, we apply a random forest classifier with $10,000$ trees as a feature selection algorithm and filter features with a threshold of 0.9 on the Pearson's correlation. We obtain 84 decorrelated features.



Pearson's correlation between features



Repartition of the filtrations and dimensions of the 84 most important decorrelated features

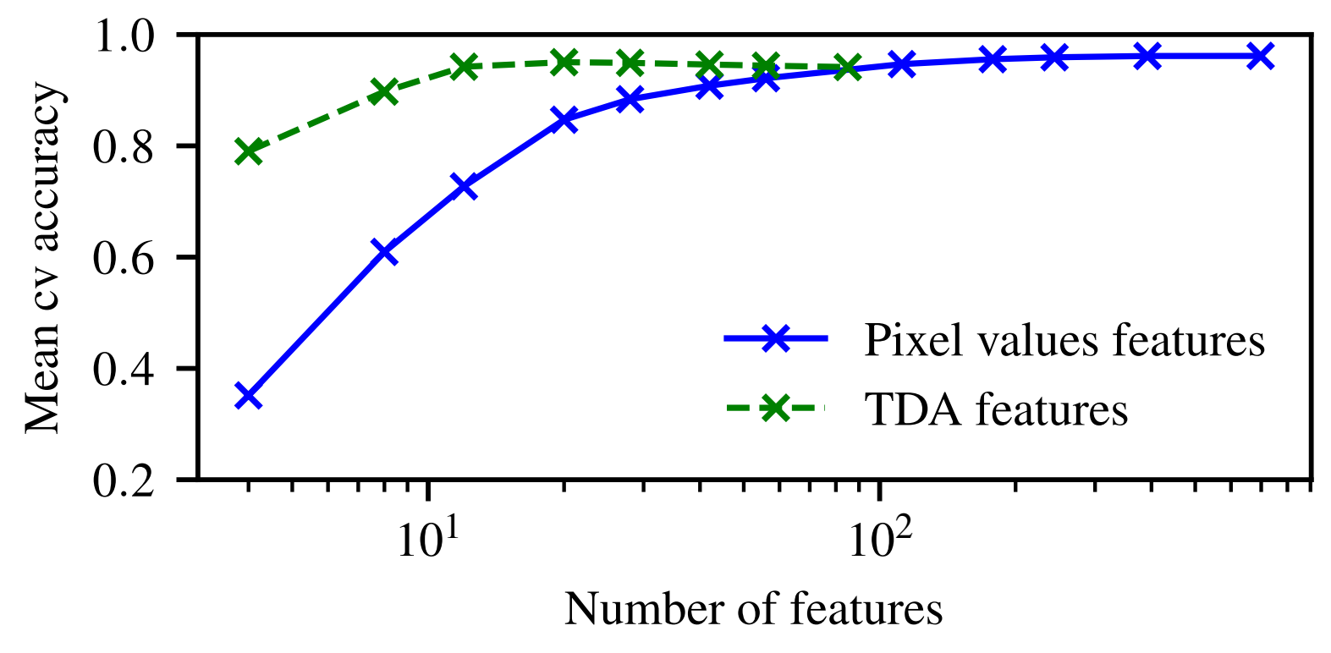
Filtration	Parameters	Dim	Vectorization
Height	Direction: $[1, 0]$	0	Persistent entropy
Radial	Center: $(6, 13)$	0	Persistent entropy
Height	Direction: $[1, -1]$	0	Persistent entropy
Radial	Center: $(0, 6)$	0	Persistent entropy
Density	#Neighbors: 6	1	Persistent entropy
Height	Direction: $[1, 1]$	0	Persistent entropy
Height	Direction: $[-1, 0]$	0	Persistent entropy
Height	Direction: $[0, 1]$	0	Persistent entropy

Description of the top 8 most important features.

The top 8 of topological features shows the clear domination of the height, radial and density filtrations and of the persistent entropy in the classification process. The height and radial filtrations provide global information about the number of connected components as well as their positions in the image. For example, a 6 and a 9 can be easily differentiated with height filtrations of directions $[1, 0]$ and $[-1, 0]$. The density filtration introduced in this work is proven to be especially effective as it detects the persistence of $1-$cycles in terms of the thickness of the lines that surround them. Persistent entropy extracts information on the distribution of bar lengths and proves to be more useful than amplitudes.

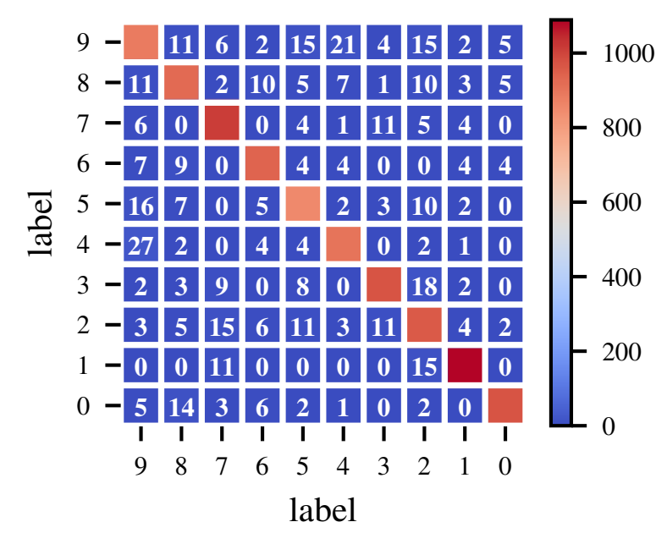
Classification

In order to classify the digits, we apply a random forest classifier with $10,000$ trees. As a reference performance, we use the pixel values of the images as features. To show that essential information about the shape of the image is captured by a much smaller number of topological features, we train both classifiers on a changing number of features ordered by importance. Our pipeline of selected decorrelated topological features achieves similar accuracy using only 28 features.



Cross validation means accuracy of the random forest against the number of features.

Where topology fails: Topological features cannot classify correctly an object when it does not have its expected shape. However, no labels are especially wrongly classified as shown on the confusion matrix.



Confusion matrix.

7 classified as a 3.

9 classified as a 4.

Even though we recognize a 7 because of the sharp angles, the topological pipeline mostly captures the three connected components that appear on height or radial filtrations that start from the left side of the image. As for the 9, it gets misclassified as a 4 because the top loop is not closed.

Conclusion

- We combined machine learning and TDA into a generic pipeline which gathers a wide range of TDA techniques to generate topological and geometric features from images.
- We provide some topological and geometrical intepretation of data: it was shown to be a powerful approach to understand the underlying characteristic shapes on the images, especially using height, radial, and our density filtrations.
- Through the systematic study of feature importance, one can easily validate and support one's choice of topological features for novel datasets.
- The topological features generated by our pipeline can be combined with more "standard" ones to improve the results.
- Similar pipelines can be applied to different data types (networks, manifolds,...) by defining filtrations on those objects.