

# Bongard Problems: A Topological Data Analysis Approach

HAJAR BOUAZZAOU I

Hassan I University  
 Department of Mathematics  
 and Computer Science  
 FST de Sett at, Km 3, B.P. : 577  
 Route de Casablanca, MOROCCO  
 h.bouazzaoui@u hp.ac.ma

MY ISMAIL MAMOUNI

CRMEF Rabat  
 Department of Mathematics  
 Av. Allal Al Fassi, Madinat Al Irfane  
 Rabat, MOROCCO  
 mamouni.myismail@gmail.com

MOHAMED ABDOU ELOMARY

Hassan I University  
 Department of Mathematics  
 and Computer Science  
 FST de Sett at, Km 3, B.P. : 577  
 Route de Casablanca, MOROCCO  
 elomaryabdou@gmail.com

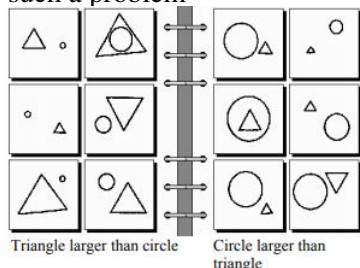
*Abstract:* Bongard problems are a set of 100 visual puzzles posed by M. M. Bongard, where each puzzle consists of twelve images separated into two groups of six images. The task is to find the unique rule separating the two classes in each given problem. The problems were first posed as a challenge for the AI community to test machines ability to imitate complex, context-dependent thinking processes using only minimal information. Although some work was done to solve these problems, none of the previous approaches could automatically solve all of them. The present paper is a contribution to attack these problems with a different approach, combining the tools of persistent homology alongside with machine learning methods. In this work, we present an algorithm and show that it is able to solve problems involving differences in connectivity and size as examples, we also show that it can solve problems involving a much larger set of differences provided the right G-equivariant operators.

*Key- Words:* Bongard Problems, Persistent homology, Topological data analysis, Machine learning, G-equivariant non-expansive operators

Received: September 27, 2019. Revised: April 2, 2020. Accepted: April 18, 2020. Published: May 5, 2020.

## 1 Introduction

Bongard problems [9] consist of twelve boxes, six of which follow a certain rule while the other six break that rule. The task of the problem solver is to identify the underlying pattern. The following figure is an example of such a problem



**Fig. 1.** Example of a Bongard problem

Bongard Problems present a great challenge specifically because of the need for an interplay of high and low-level vision with high-level cognition. They are good indicators of some fundamental aspects of complex thinking processes [24]. Such aspects include pattern formation and abstraction, pattern matching and recognition, clustering and categorization, as well as memory and learning [18].

Bongard problems are interesting as well for the possibility of being extended beyond the original posed problems.

Bongard Problems are not only a challenge for the AI community, they can also be helpful for engineers as well as mathematicians to understand and model brain functions such as learning, finding similarity, creating abstract ideas and acting by intuition [9]. A machine that is able to solve Bongard Problems is indicative of the presence of high-level cognitive functions that can be further used to solve problems that go beyond the original posed problems such as retrieving similar images, finding a network of people with similar interests, counting objects in images and videos or annotating images.

Few efforts have been made to automatically solve Bongard Problems. Some major attempts are "RF4" by Saito and Nakano[30] and "Phaeaco" by Harry Foundalis [18]. RF4 is an inductive logic programming system where images are hand-coded into logical formulas with computer vision being totally avoided. It could solve in this way 41 of the 100 Bongard problems [25]. Phaeaco, on the other hand, uses images as inputs. It proceeds by building a "mental representation" of each of the images. From that representation, it deduces the possible concepts that may be common to the six images on the left, and those

that may be common to the six images on the right, and seeks a concept that is shared only by images on the left, and not by those on the right. Of the original 100 Bongard problems it can only solve around 10. An other recent approach is the one using Bayesian inference and a symbolic visual vocabulary to represent visual concepts. This approach could solve the largest number of Bongard Problems yet [12].

In this paper, we approach Bongard problems from a different angle and consider ones that have not been solved by any of the systems mentioned earlier. The topological tool, as far as we know, has never been used to attack these problems.

Topological Data Analysis (TDA) is a relatively recent field of study that proved successful in retrieving insight from data in many application domains. Examples include image analysis [6], signal processing [27], robot planning [29], sensor networks [26], biology [33] and cosmology [32] to cite only a few.

Our contribution consists in using topological data analysis and specifically persistent homology as a means of representation using topological signatures and a means of comparison and distinction through the use of a metric on the space of these signatures. Finding the unique rule separating the two classes in each problem requires a good representation of the images, and by that we mean a representation that allows us to have 'multiple views' of the data hence the use of G-equivariant non-expansive operators (GENEOs).

In this work, we are combining methods both from TDA and machine learning. As a first step, and depending on the pattern of interest, we either compute persistence diagrams to capture topological features by building lower star filtrations on the grayscale images, or apply GENEOs on filtering functions in the case where we are interested in subgroups of the self-homeomorphism group of the space in question. As a second step, the features computed with each method are transformed into a finite-dimensional vector representation or a distance matrix. They are then fed to a clustering algorithm in order to decide which images are more similar to which. If the grouping is similar to the one in the original problem, the feature responsible for this clustering is taken to be the separating rule. In the next section, we give a brief overview of clustering as the main machine learning method in this paper, we next go through the mathematics of topological data analysis and G-equivariant non-expansive operators. In the section afterward, we expose our algorithm along with sample problems before ending with a conclusion.

## 2 Clustering

Clustering is one of the most important tools used for unsupervised machine learning tasks. It is used in a variety of fields ranging from biology [28], image segmentation [7], natural language processing [14], to crime analysis [2] and climatology [23].

Clustering can both be used as a stand-alone tool to make sense of unlabeled data, or as a preprocessing step for exploratory purposes in order to visualize connexions that might not be otherwise detected. Clustering aims at grouping data points into classes by minimizing intra-cluster similarities and maximizing inter-cluster similarities. Evaluating data similarity depends on the choice of a similarity metric, these can be the Euclidean distance, the Manhattan distance or the cosine distance among many others. There is no single algorithm for clustering. In the literature, different clustering models are used across different problem domains, and for each model, a variety of algorithms are proposed. Such models include among others: Connectivity models (e.g hierarchical clustering), centroid models (e.g k-means algorithm), distribution models (e.g expectation-maximization algorithm) or density models (e.g DBSCAN).

For more details on the core methods for data clustering, the application domains and the insights obtained from the clustering process, we refer the interested reader to [3]

## 3 Mathematical background

### 3.1 Persistent Homology

The central idea in this work is to find the pattern setting apart the two classes of shapes in each Bongard problem. Homology groups formalize the description of the topology of geometric objects, specifically, persistent homology gives us a way to make that distinction by means of comparing topological signatures. The following is a formal overview of how this comparison is made.

#### 3.1.1 Homology

Homology is the mathematical theory concerned with distinguishing shapes and spaces in general on the basis of their holes. With respect to this theory, a sphere and a cube are equivalent as they are both single connected components with no holes. A sphere and a torus, however, are not equivalent since the former contains no holes whereas the latter contains among others a two-dimensional hole. We formalize the intuition behind homology in the rest of this section both

for the continuous and the discrete settings.

Let  $\mathbb{X}$  be a topological space. The construction begins with a chain complex  $C(\mathbb{X})$ . A *chain complex* defined on a topological space  $\mathbb{X}$  is a sequence of abelian groups or modules  $(C_k)_{k \geq 0}$  connected by homomorphisms called boundary operators

$$\dots \xrightarrow{\partial_{n+2}} C_{n+1} \xrightarrow{\partial_{n+1}} C_p \xrightarrow{\partial_n} C_{n-1} \xrightarrow{\partial_{n-1}} \dots$$

such that

$$\text{Im} \partial_{n+1} \subseteq \text{Ker} \partial_n,$$

The boundary operators satisfy the fundamental property:  $\partial_n \circ \partial_{n+1} = 0$  for any  $n \geq 1$

**Definition 1.** Let  $\mathbb{X}$  be a topological space. The  $n$ -th homology group of  $\mathbb{X}$  is the vector space

$$H_n(\mathbb{X}) := \text{Ker}(\partial_n) / \text{Im}(\partial_{n+1})$$

The  $n$ -th Betti number of  $\mathbb{X}$  is the dimension  $\beta_n(\mathbb{X}) = \dim H_n(\mathbb{X})$  of the vector space  $H_n(\mathbb{X})$

The  $n$ -th Betty number actually corresponds to the number of  $n$ - dimensional holes present in the data, thus,  $\beta_0$  corresponds to the number of connected components,  $\beta_1$  is the number of one-dimensional loops,  $\beta_2$  is the number of two dimensional holes and so forth.

### 3.1.2 Persistent homology

Since features can be of different scales or be nested, it makes sense to track their appearance and disappearance across increasing scales. Thus, instead of studying the homology of the space as a whole, we build a filtration of that space. We then compute the persistent homology groups across the resulting subspaces.

*Sublevel sets Filtration:* A filtration of a topological space  $\mathbb{X}$  is a nested sequence of topological subspaces

$$\mathcal{F}_p : \emptyset = \mathbb{X}_0 \subseteq \mathbb{X}_1 \subseteq \mathbb{X}_2 \subseteq \dots \mathbb{X}_n = \mathbb{X},$$

Let  $f : \mathbb{X} \rightarrow \mathbb{R}$  a real valued function. The sublevel sets form a filtration  $\mathbb{X}_r \subseteq \mathbb{X}_s$  for all  $r \leq s$  of the topological space  $\mathbb{X}$ , such that  $\mathbb{X}_r = f^{-1}[-\infty, r]$

*Persistent homology groups and persistent Betti numbers*

For every  $r \leq s$  we have an inclusion map from the underlying space of  $\mathbb{X}_r$  to that of  $\mathbb{X}_s$  and therefore an induced homomorphism,

$$f_p^{r,s} : H_p(\mathbb{X}_r) \rightarrow H_p(\mathbb{X}_s)$$

for each dimension  $p$ . The filtration thus corresponds to a sequence of homology groups connected by homomorphisms,

$$0 = H_p(\mathbb{X}_0) \rightarrow H_p(\mathbb{X}_1) \rightarrow \dots \rightarrow H_p(\mathbb{X}_n) = H_p(\mathbb{X}),$$

one for each dimension  $p$ . The  $p$ -th *persistent homology groups* are the images of the homomorphisms induced by inclusion,

$$H_p^{r,s} = \text{Im} f_p^{r,s},$$

for  $0 \leq r \leq s \leq n$ . The corresponding  $p$ -th persistent Betti numbers are the ranks of these groups,

$$\beta_p^{r,s} = \text{rank} H_p^{r,s}.$$

*Persistence diagrams and persistence images:* The homology groups of the sublevel sets are encoded in *persistence diagrams*. As  $r$  increases, new homology classes are born whereas others die as some components merge with others. The level at which a new component is created is called its birth time and the level at which it merges with another component is called its death time. The persistence of a homology class is the difference between its birth and death time. The  $x$ -coordinates of a persistence diagram indicate birth time and the  $y$ -coordinates death time, formally a *persistence diagram* is a countable multiset of points in  $\mathbb{R}^2$  along with the diagonal  $\Delta = \{(x, y) \in \mathbb{R}^2 | x = y\}$ , where each point on the diagonal has infinite multiplicity.

However, persistence diagrams don't lend themselves well to machine learning tasks. Persistence images, on the other hand, are more suitable for that purpose. Persistence images are finite vector representations of persistence diagrams. For a formal presentation, we refer the reader to [1]

*Bottleneck distance:* To measure similarities in the space of persistence diagrams we endow it with a metric: the bottleneck distance. It is defined as follows:

**Definition 2.** For two persistence diagrams  $D$  and  $E$ , we define their Bottleneck ( $w_\infty$ ) distance by:

$$w_\infty(D, E) := \inf_{\eta} \sup_{x \in D} \|x - \eta(x)\|_\infty$$

where  $\eta$  ranges over bijections between  $D$  and  $E$

For further details, the interested reader is referred to [11], [16], [17].

Images are formed of pixels. In order to solve the present problem, computations will naturally be carried out in the discrete setting. It is thus necessary to

use a model that is adjusted for such a setting. Simplicial homology, which is one of the most common homology theories, presents itself as a valid approach as its is more suitable for computations.

In practice, given point cloud data, we construct a simplicial complex on top of that data.

**Definition 3.** [21]

Let  $S$  be a discrete set. An abstract simplicial complex is a collection  $K$  of finite subsets of  $S$ , closed under restriction: for each  $\sigma \in K$ , all subsets of  $\sigma$  are also in  $K$ . Each element  $\sigma \in K$  is called a  $k$ -simplex, where  $|\sigma| = k + 1$ . Given a  $k$ -simplex  $\sigma$ , its faces are the simplices corresponding to all subsets  $\sigma' \subset \sigma$

We can compute homology for simplicial complexes

**Definition 4.** [15]

Let  $K$  be a simplicial complex. The  $\mathbb{Z}/2\mathbb{Z}$  vector space generated by the  $p$ -dimensional simplices of  $K$  is denoted  $C_p(K)$ . It consists of all  $p$ -chains, which are formal sums

$$c = \sum_j \lambda_j \sigma_j$$

where the  $\lambda_j$  are 0 or 1 and the  $\sigma_j$  are  $p$ -simplices in  $K$ .

The boundary,  $\partial(\sigma_j)$ , is the formal sum of the  $(p - 1)$ -dimensional faces of  $\sigma_j$  and the boundary of the chain is obtained by extending  $\partial$  linearly,

$$\partial(c) = \sum_j \lambda_j \partial(\sigma_j)$$

where addition is modulo 2. It is not difficult to check that  $\partial \circ \partial = \partial^2 = 0$

As in the continuous setting, the  $p$ -th homology group of  $K$  is the vector space

$$H_p(K) := \text{Ker}(\partial_p) / \text{Im}(\partial_{p+1})$$

We can also define persistence for simplicial complexes

**Definition 5.** [15] A subcomplex is a subset of simplices that is closed under the face relation. A filtration of a simplicial complex  $K$  is a nested sequence of subcomplexes that starts with the empty complex and ends with the complete complex,  $\emptyset = K_0 \subseteq K_1 \subseteq \dots \subseteq K_m = K$

The subcomplexes are the analog of the sublevel sets in the continuous setting and their persistence is computed in the same way we described earlier.

### 3.2 G-equivariant non-expansive operators

As stated earlier, one way of getting topological summaries of data is by building sublevel set filtrations on top of that data and computing persistence diagrams. However, persistent homology in the form we described cannot distinguish between summaries produced by a filtering function  $f$  and a filtering function  $f \circ g$  when  $g$  is a self homeomorphism [13]. That being the case, only a few Bongard problems would benefit from the use of classical persistent homology when the invariance group is the group of all self-homeomorphisms, which we'll denote  $\text{Homeo}(X)$ . Other problems would better make use of invariance with respect to proper subgroups of  $\text{Homeo}(X)$ . One example of these problems is the one at the method section.

A major source of inspiration in that regard was the following work [8], [20], [19]. Our main effort is to explore the possibility of using their results in the case of Bongard Problems.

For every BP, we might be interested in getting invariance with respect to priorly chosen subgroups  $G$  of  $\text{Homeo}(X)$ . Applying  $G$ -equivariant non-expansive operators on filtrations will allow us to get multiple measurements associated with each filtration. These measurements can be thought of as different 'lenses' through which we see our data. We then approximate the natural distance between measuring functions and construct a distance matrix which is then fed to the clustering algorithm.

The subgroup  $G$  of  $\text{Homeo}(X)$  transforms the set  $\Phi$  of filtering functions by a right group action

$$f: \Phi \times G \rightarrow \Phi$$

$$(\varphi, g) \mapsto \varphi \circ g$$

We first give the definition of  $G$ -equivariant non-expansive operators

**Definition 6.** Let  $\Phi$  be a topological subspace of  $C^0(X, \mathbb{R})$ , the set of admissible filtering functions on  $X$ , and  $G$  a subgroup of  $\text{Homeo}(X)$ .

$F$  is a  $G$ -equivariant non-expansive operators if it verifies the following properties:

1.  $F$  is a function from  $\Phi$  to  $\Phi$
2.  $F(\varphi \circ g) = F(\varphi) \circ g$  for every  $\varphi \in \Phi$  and every  $g \in G$
3.  $\|F(\varphi_1) - F(\varphi_2)\|_\infty \leq \|\varphi_1 - \varphi_2\|_\infty$  for every  $\varphi_1, \varphi_2 \in \Phi$

As previously said, computing the bottleneck distance between two persistence diagrams is not well

suiting for some problems in shape comparison, hence the need to introduce the natural pseudo-distance as a more powerful approach in comparing two filtrations.

**Definition 7.** *The natural pseudo distance with respect to a group  $G$  is defined by*

$$d_G(\varphi_1, \varphi_2) = \inf_{g \in G} \max_{x \in X} |\varphi_1(x) - \varphi_2(g(x))|$$

The natural pseudo distance, however, presents a challenge; it is difficult to compute. To address this issue,  $\mathcal{D}_{match}^{\mathcal{F}}$  is introduced in [20] as a tool to approximate  $d_G$ , which is also a  $G$ -invariant pseudo-metric on  $\Phi$

**Definition 8.** *Let  $\mathcal{F}(\Phi, G)$  be the set of all  $G$ -equivariant non expansive operators and  $\mathcal{F}$  be a non empty subset of  $\mathcal{F}(\Phi, G)$*

$$\mathcal{D}_{match}^{\mathcal{F}}(\varphi_1, \varphi_2) = \sup_{F \in \mathcal{F}} d_{match}(r_k(F(\varphi_1)), r_k(F(\varphi_2)))$$

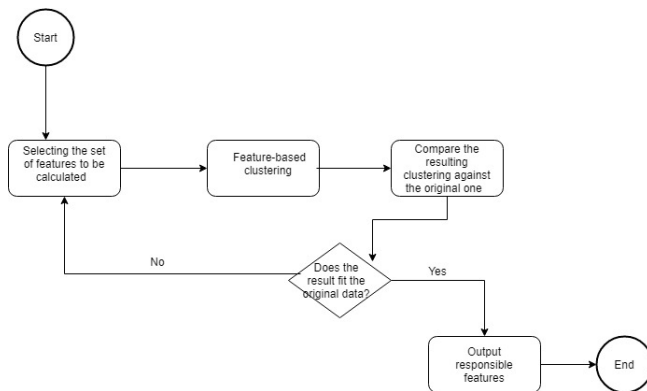
for every  $\varphi_1, \varphi_2 \in \Phi$ .  $d_{match}$  corresponds to the bottleneck distance and  $r_k$  denotes the  $k$ -th persistent Betti number function with respect to the function  $\varphi$ .

## 4 Method and experiments

In this section, we present the general algorithm for solving Bongard problems, as well as the algorithm for two sample feature functions where we both use classical persistent homology and apply  $G$ -equivariant non expansive operators.

In our approach to this problem, we assume that images can be studied and compared through their topological signatures, we also assume that given the right feature function along with the right clustering parameters, it is possible to find a grouping that matches perfectly the one in the original problem.

We first give a summary of the entire method in the figure below.



**Fig. 2.** Summarized Method

## 4.1 The abstract general algorithm

### Algorithm 1: Abstract General Algorithm

---

**Input** : 12 Bongard images with corresponding class

**Output:** The separating feature/rule

```

1 while not all feature functions are called do
2   feature ← random feature call ;
3   cluster images based on feature ;
4   match ← compare original and resulting clusters ;
5   if match is perfect then
6     return feature ;
7 return "solution not found";
  
```

---

### Algorithm 2: Sample feature functions

---

```

1 Function Connex (image) :
2   Convert image to grayscale image;
3   Construct lower star image;
4   Compute  $H_0$ ;
5   Compute persistence diagram ;
6   Transform persistence diagram to persistence image;
7   return persistence diagram , persistence image;
8 ;
9 Function Equivariance (images) :
10  operators ← list of operators ;
11  for img in images do
12    lower_star_image ← compute lower star filtration of img ;
13    apply list of operators on lower_star_image ;
14  end
15  for all pairs of lower_star_image with the same applied operator do
16    Approximate the natural pseudo distance ;
17  end
18  Compute distance matrix ;
19  return Distance Matrix ;
20 ;
  
```

---

### 4.1.1 General Description

The algorithm takes as input the 12 Bongard images along with their corresponding classes. It makes use of a list of randomly callable functions, each of which is responsible for the computation of a certain feature.

The called function returns a feature vector or a distance matrix on the basis of which the clustering is performed. The algorithm then compares the resulting cluster labels with the classes given as input. If the labels match, the algorithm states the function/feature called as the separating rule and the algorithm is suspended. Otherwise, the function is removed from the list and an other one is called. As long as the feature is not found, the algorithm keeps calling functions until the list is exhausted.

#### 4.1.2 The features set

The problems at hand are of geometric and topological nature. In this context, capturing geometric features and topological shape invariants makes sense. These invariants serve as candidates for the separating rule we are looking for. For each problem, we might be interested in different features such as connectivity, the convex hull, the area or the skeleton among others.

We must note that our starting list of features is by no means exhaustive. Yet, it would serve as a good starting point for solving some Bongard problems. We can extend it further by mixing and matching a greater number of patterns to see which ones fit the given input images.

#### 4.1.3 Feature-Based clustering

Once we compute the selected features, we transform them into a suitable format. We then feed them as input to a clustering algorithm. Research on data clustering methods has been extensive. Different clustering algorithms are used for different tasks. In this work, our algorithm of choice is DBSCAN (Density-Based Spatial Clustering of Applications with Noise) [31]. As its name suggests, it is a density based method. It attempts to cluster data points on the basis of a density function. DBSCAN separates the given data into regions of high density and others of low density.

When working with DBSCAN, it is important to specify two important parameters; A minimum number of samples  $n$  and a distance  $\epsilon$ . Using these two parameters, the algorithm is able to determine what we call core samples. These are samples such that there are  $n$  samples within a distance  $\epsilon$ .

Core samples are a central component to DBSCAN. These are samples in areas of high density which constitute the building block of a cluster.

A cluster is then a set of core samples, but not only. The cluster is built by recursively taking each core sample and finding all of its neighbours that are

core samples. The set of non-core samples will be samples on the cluster borderlines.

#### 4.1.4 Clustering evaluation and Separating rule

Since we have a known clustering solution, we have a ground truth clustering against which we can evaluate our results. If the resulting and original clustering match, the features responsible for that outcome are traced back and given as output.

In this paper, we are using the Adjusted Rand Index [22] as a measure of similarity between the original and resulting clusterings. When two clusterings are similar, we get a positive ARI score with 1 being a perfect match.

The raw index is given by

$$RI = \frac{a + b}{C_2^{n_{samples}}},$$

where  $a$  denotes the number of pairs of elements that are both in the same set in the original and resulting clusterings and  $b$  the number of pairs of elements that are in different sets in the two clusterings.

In order to correct for chance, we define the adjusted Rand index as follows:

$$ARI = \frac{RI - E[RI]}{max(RI) - E[RI]}.$$

In the next section, we demonstrate the use of both persistent homology in its classical form and in combination with G-equivariance to solve two different Bongard problems.

#### 4.2 Problem 1: Classical persistent homology

The following figure presents Bongard Problem # 23. In order to solve this problem, we first compute persistence diagrams of the images, transform them into persistence images and then feed them to the clustering algorithm before deciding if the number of components is actually the separating rule.

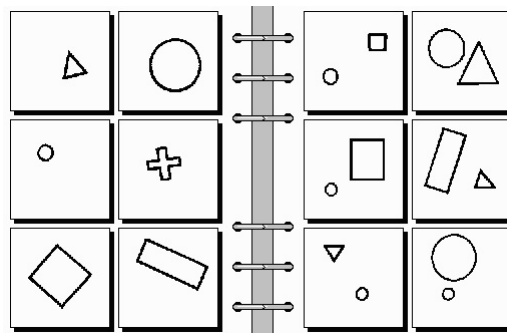
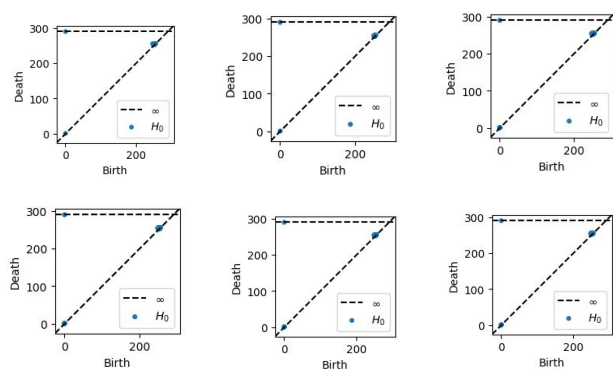


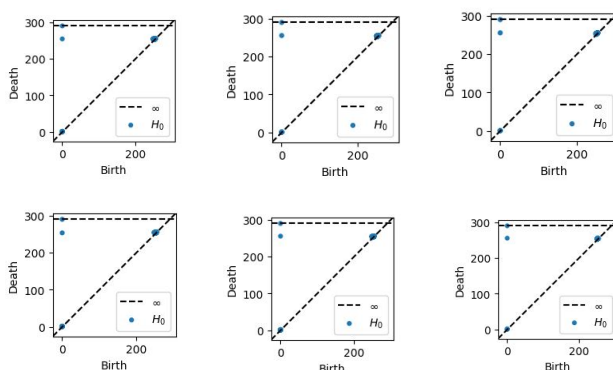
Fig. 3. Problem # 23

### Persistence Diagrams

Using Ripser library [5], we compute the 0-dimensional lower star filtration on each of the twelve images. Internally, the function first constructs a sparse matrix whose elements are the image pixels, these are taken to be vertices and each one of them is connected to its spatial neighbours. Every two connected vertices share an edge whose weight is the maximum of the two pixel values. We get the following results:



(a) Persistence Diagrams of left images

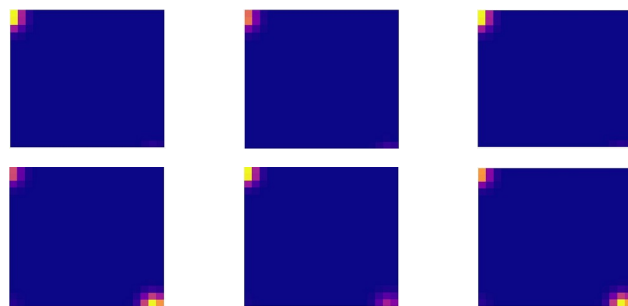


(b) Persistence Diagrams of right images

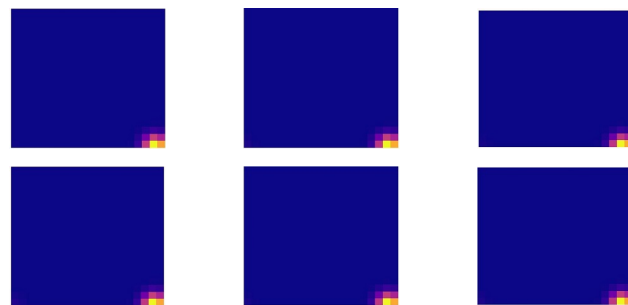
**Fig. 4.** Computed Persistence Diagrams

**Persistence Images** Persistence diagrams serve as a representation of homological information. However, it is hard to directly use them for machine learning tasks. Persistence images, on the other hand, are well suited for that purpose. These are actually finite-dimensional vector representations of persistence diagrams. They too verify stability with respect to small perturbations [1].

We get the following results using the Persim library [34].



(a) Persistence Images of left images



(b) Persistence Images of right images

**Fig. 5.** Computed Persistence Images

In our first problem, we cluster images using the DBSCAN algorithm on the basis of the *connex* function output. The most important parameters to set are the minimum number of samples in each cluster and the  $\epsilon$  parameter that determines the maximum distance between two samples to be considered as neighbors. In this case, the  $\epsilon$  parameter is set to 2. Since the twelve images should be separated into two classes of six images each, we set the minimum number of samples parameter to 6.

By computing the adjusted random index, we verify the match between the resulting and the original groupings. A perfect matching score of 1 indicates that the difference between the two sets of images resides in the number of connected components. It is the separating rule we are looking for.

### 4.3 Problem 2: G-equivariance

The following problem presents homeomorphic objects that can't be separated using persistence diagrams in the classical form. In this setting, the use of G-equivariant non expansive operators presents itself as a better alternative to compare shapes and detect the rule of interest.

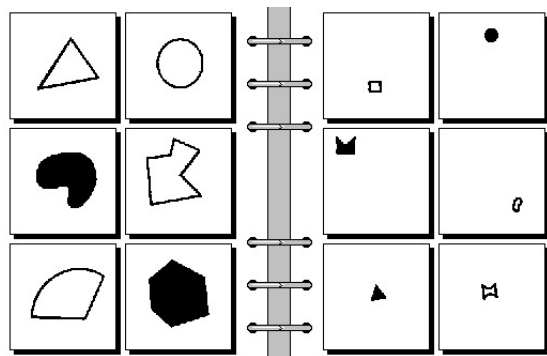


Fig. 6. Problem 2

Let  $G$  be the group of all equiareal transformations of the form  $g : x \mapsto Ax + B$ , where  $A$  is a unimodular matrix and  $B$  a random matrix of the same shape as the given image.

One way of constructing  $G$ -equivariant non-expansive operators is by means of permutants as described in [10]. We present here some of the operators we used to perform our experiments:

$$F_1(\varphi) = \frac{1}{2}(\varphi(Ax + B) + \varphi(Ax - B))$$

$$F_2(\varphi) = \frac{1}{3}(\varphi(Ax + B) + \varphi(Ax) + \varphi(Ax - B))$$

$$F_3(\varphi) = \frac{1}{5}(\varphi(Ax + B) + \varphi(Ax + C) + \varphi(Ax - B) + \varphi(Ax - C))$$

$\varphi$  being the filtering function,  $A$  a unimodular matrix and  $B$  and  $C$  two random matrices.

These operators can serve as a basis to create new ones. The interested reader can refer to [4] for results on generating new operators from pre-built ones along with proofs.

**Remark:** Since the set  $\mathcal{F}$  of all  $G$ -equivariant non expansive operators can be approximated by a finite subset of  $\mathcal{F}$  [20], we only need a small subset of operators to perform a clustering that matches the given one.

The problems we discussed above have been solved by both major approaches discussed in this paper, namely, by Phaeaco and the visual language and pragmatic reasoning approach. The Bongard problems we're presenting next have not been solved by neither approach. In almost all these problems instances, shapes in each group present similarities that can't be captured using literal concepts. For example, a solver that relies on such approach would not make the difference between a triangle represented with straight lines and one represented with curved lines as in the case of BP#10.

For such cases, persistent homology and  $G$ -equivariant operators present powerful tools mainly for two reasons: firstly, because persistent homology can separate relevant features from noise and is more concerned with the general shape of data, and secondly, because GENEOS present the algorithm with

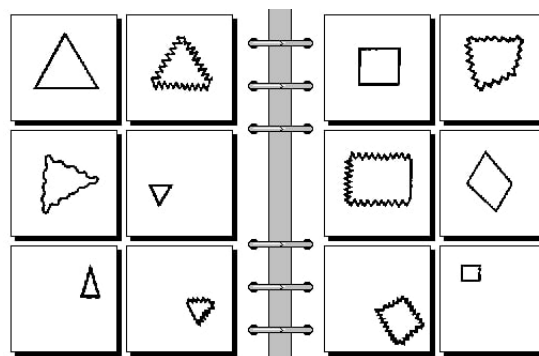


Fig. 7. BP # 10

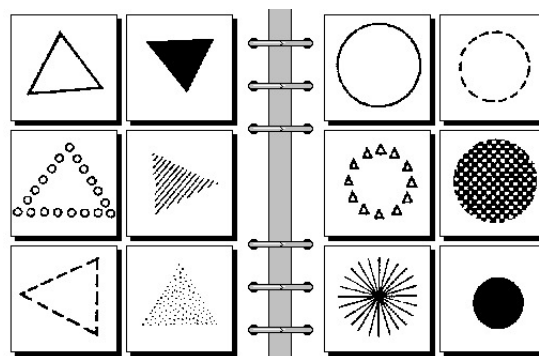


Fig. 8. BP # 97

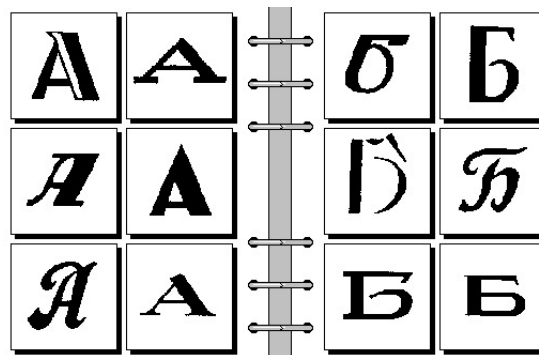


Fig. 9. BP # 100



multiple functions through which to see the data, thus giving it more ability to recognize the possible pattern governing the sets of images.

Letting  $G$  be the group of all similarity transformations, we can construct  $G$ -equivariant non expansive operators which serve as tools to identify images with similar patterns and arrange Bongard images into clusters that match the original ones.

## 5 Conclusion

Given a Bongard problem, one would first observe the given images, make a hypothesis about which property is responsible for the separation of the two sets of images, verify that this property or rule holds for all images on the left but holds for none of the images on the right. One would pursue this process in a loop, multiplying the points of view, taking some properties into account and ignoring others until the separating rule is found.

The algorithm we proposed in this paper works in the same fashion. The algorithm constructs different representations of the same images, through the use of persistent homology summaries and  $G$ -equivariant non-expansive operators, allowing it to view the images using different 'lenses', the choice of the property that might explain the separation of the two sets is made at random. Performing clustering based on the chosen property and matching it with the original grouping allows the algorithm to decide if the separating rule was found. That being said, this line of research, i.e: approaching BP's by means of persistent homology, is still at its beginning. Further experiments are still conducted on discussed BPs and many questions remain unanswered. Some of them are related to the construction of TDA tools like the construction of GENEOS and others are more specific to BPs.

We list here some of these questions:

- How can one better construct new GENEOS from already built ones?
- How can one choose which properties to consider first?
- How can the algorithm benefit from the use of memory to solve new problems?
- Is there a complete set of GENEOS that would allow us to solve all BPs?

We will further explore some of these questions in future work.

## References:

- [1] Henry Adams, Sofya Chepushtanova, Tegan Emerson, Eric Hanson, Michael Kirby, Francis Motta, Rachel Neville, Chris Peterson, Patrick Shipman, and Lori Ziegelmeier, *Persistence Images: A Stable Vector Representation of Persistent Homology* (2015), available at 1507.06217.
- [2] Jyoti Agarwal, *Crime Analysis using K-Means Clustering*, 2013.
- [3] Charu C Aggarwal and Chandan K Reddy, *DATA CLUSTERING Algorithms and Applications Edited by*, 2014.
- [4] Patrizio Frosini B and Nicola Quercioli, *Machine Learning and Knowledge Extraction* **11015** (2018), 14–24.
- [5] Christopher Tralie Bar-On, Nathaniel Saul, and Rann, *{Ripser.py}: A Lean Persistent Homology Library for Python*, The Open Journal, 2018.
- [6] M. Bendich, P.; Edelsbrunner, H.; Kerber, *Computing Robustness and Persistence for Images*, IEEE Transactions on Visualization and Computer Graphics **16** (2010), no. 6, 1251–1260.
- [7] Alex Bewley and Ben Upcroft, *Advantages of Exploiting Projection Structure for Segmenting Dense 3D Point Clouds*.
- [8] S Biasotti, L De Floriani, B Falcidieno, P Frosini, D Giorgi, and M Spagnuolo, *Describing shapes by geometrical-topological properties of real functions*, Technical Report N, 2008.
- [9] M. M. Bongard, *Pattern Recognition*, Hayden Book Co., Spartan Books, Rochelle Park, N.J., 1970.
- [10] Francesco Camporesi, Patrizio Frosini, and Nicola Quercioli, *On a new method to build group equivariant operators by means of permutants*.
- [11] Frédéric Chazal and Bertrand Michel, *An introduction to Topological Data Analysis: fundamental and practical aspects for data scientists* (2017oct), 1–38 pp., available at 1710.04019.
- [12] Stefan Depeweg, Constantin A. Rothkopf, and Frank Jäkel, *Solving Bongard Problems with a Visual Language and Pragmatic Reasoning* (2018), available at 1804.04452.
- [13] B. Di Fabio and P. Frosini, *Filtrations induced by continuous functions*, Topology and its Applications **160** (2013), no. 12, 1413–1422.
- [14] Antonio Di Marco and Roberto Navigli, *Clustering and diversifying web search results with graph-based word sense induction*, Computational Linguistics **39** (2013sep), no. 3, 709–754.
- [15] Herbert Edelsbrunner and John Harer, *Persistent Homology-a Survey*.
- [16] \_\_\_\_\_, *Computational Topology An Introduction*, 2009.
- [17] Herbert Edelsbrunner and Dmitriy Morozov, *Persistent Homology: Theory and Practice*, European Congress of Mathematics Kraków, 2 7 July, 2012 (2013), 31–50.
- [18] Harry E Foundalis, *Phaeaco: A Cognitive Architecture Inspired by Bongard's Problems*, Ph.D. Thesis, 2006.
- [19] Patrizio Frosini, *G -invariant persistent homology* **0** (2018), available at arXiv:1212.0655v5.

- [20] Patrizio Frosini and Grzegorz Jablonski, *Combining persistent homology and invariance groups for shape comparison* (2013), available at 1312.7219.
- [21] R. Ghrist, *Elementary Applied Topology*, 10th ed., Createspace, 2014.
- [22] Lawrence Hubert and Phipps Arabie, *Comparing partitions*, *Journal of Classification* **2** (1985dec), no. 1, 193–218.
- [23] Radan Huth, Christoph Beck, Andreas Philipp, Matthias Demuzere, Zbigniew Ustrnul, Monika Cahynová, Jan Kyselý, and Ole Einar Tveito, *Classifications of atmospheric circulation patterns: recent advances and applications.*, *Annals of the New York Academy of Sciences* **1146** (2008dec), 105–52.
- [24] Alexandre Linhares, *A glimpse at the metaphysics of Bongard problems*.
- [25] Jisha Maniamma and Hiroaki Wagatsuma, *A Semantic Web Technique as Logical Inference Puzzle-Solver for Bongard Problems*.
- [26] P. Dlotko, M. Juda, M. Mrozek and R. Ghrist, *Distributed computation of coverage in sensor networks by homological methods*, *Applicable Algebra in Engineering Communication and Computing* **23** (2012), no. 1-2, 29–58.
- [27] J. Perea, J.A., Harer, *Sliding Windows and Persistence: An Application of Topological Methods to Signal Analysis*, *Found Comput Math* **15** (2015), 799–838.
- [28] Mado Remm, Christian E.V. Storm, and Erik L.L. Sonnhammer, *Automatic clustering of orthologs and in-paralogs from pairwise species comparisons*, *Journal of Molecular Biology* **314** (2001dec), no. 5, 1041–1052.
- [29] S. Bhattachayra, R. Ghrist and V. Kumar, *Persistent homology in Z2 coefficients for robot path planning in uncertain environments*, *IEEE Trans. on Robotics* **31** (2015), no. 3, 578–590.
- [30] K Saito, R Nakano *Trans. of IPSJ*, and undefined 1995, *Adaptive concept learning algorithm: Rf4*.
- [31] Erich Schubert, Jörg Sander, Martin Ester, Hans Peter Kriegel, and Xiaowei Xu, *DBSCAN revisited, revisited: Why and how you should (still) use DBSCAN*, *ACM Transactions on Database Systems* **42** (2017jul), no. 3.
- [32] H. Kawahara T. Sousbie, C. Pichon, *The persistent cosmic web and its filamentary structure II. Illustrations*, *Monthly Notices of the Royal Astronomical Society* **414** (2011), no. 1, 384–403.
- [33] Halverson T Topaz CM, Ziegelmeier L, *Topological Data Analysis of Biological Aggregation Models*, *PLOS ONE* **10** (2015), no. 5.
- [34] Nathaniel Saul Trali and Chris, *Scikit-TDA: Topological Data Analysis for Python*, 2019.