



HAL
open science

ATOL: Measure Vectorisation for Automatic Topologically-Oriented Learning

Martin Royer, Frédéric Chazal, Clément Levrard, Yuichi Ike, Yuhei Umeda

► **To cite this version:**

Martin Royer, Frédéric Chazal, Clément Levrard, Yuichi Ike, Yuhei Umeda. ATOL: Measure Vectorisation for Automatic Topologically-Oriented Learning. 2020. hal-02296513v2

HAL Id: hal-02296513

<https://hal.science/hal-02296513v2>

Preprint submitted on 10 Feb 2020 (v2), last revised 13 Oct 2020 (v3)

HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.

ATOL: Measure Vectorisation for Automatic Topologically-Oriented Learning

Martin Royer
Inria Saclay

Frdric Chazal
Inria Saclay

Clment Levrard
Universit Paris Diderot

Yuichi Ike
Fujitsu Laboratories

Yuhei Umeda
Fujitsu Laboratories

Abstract

Robust topological information commonly comes in the form of a set of persistence diagrams, finite measures that are in nature uneasy to affix to generic machine learning frameworks. We introduce a learnt, unsupervised measure vectorisation method and use it for reflecting underlying changes in topological behaviour in machine learning contexts. Relying on optimal measure quantisation results the method is tailored to efficiently discriminate important plane regions where meaningful differences arise. We showcase the strength and robustness of our approach on a number of applications, from emulous and modern graph collections where the method reaches state-of-the-art performance to a geometric synthetic dynamical orbits problem. The proposed methodology comes with only high level tuning parameters such as the total measure encoding budget, and we provide a completely open access software.

1 Introduction

Topological Data Analysis (TDA) is a field dedicated to the capture and description of relevant geometric or topological information from data. The use of TDA with standard machine learning tools has proved particularly advantageous in dealing with all sorts of complex data, meaning objects that are not or partly Euclidean, for instance graphs, time series, etc. The applications

are abundant, from social network analysis, bio and chemoinformatics, to medical imaging and computer vision, to name a few.

Through persistent homology, a multi-scale analysis of the topological properties of the data, robust and stable information can be extracted. The resulting features are commonly computed in the form of a persistence diagram whose structure (an unordered set of points in the plane representing birth and death times for the features) does not easily fit the general machine learning input format. Therefore TDA is generally combined to machine learning by way of an embedding method for persistence diagrams.

Contributions. Our work is set in that trend. First using recent measure quantisation results we introduce a learnt, unsupervised vectorisation method for measures in Euclidean spaces of any dimension (Section 2.1). Then we specialise this method for handling persistence diagrams (Section 2.2), allowing for easy integration of topological features into challenging machine learning problems with theoretical guarantees (Theorem 1). We illustrate our approach on a set of experiments that leads to state-of-the-art results on difficult problems (Section 3). Lastly we provide an open source implementation and notebook.

Our quantisation of the space of diagrams is statistically optimal and the resulting algorithm, a simple variant of the Lloyd’s algorithm, is simple and efficient. It can also be used in a minibatch fashion, making it practical for large scale and high dimensional problems, and competitive with respect to more sophisticated methods involving kernels, deep learning, or computations of Wasserstein distance. To the best of our knowledge, our results provide the first vectorisation method for persistence diagrams that is proven to be able to separate clusters. There is little to no tuning to this method, and no knowledge of TDA is required for this framework.

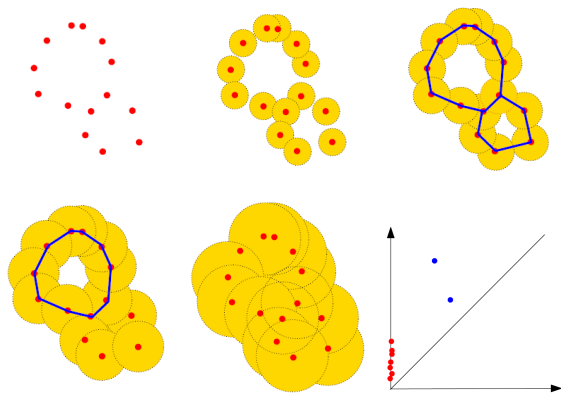


Figure 1: Example of a filtration by union of balls built on top of a 2-dimensional data set (red points) and its corresponding persistence diagram. As the radius of the balls increases, the connected components initially represented by each data point get merged; two cycles appears and disappears along the filtration. The connected components give rise to the red points on the vertical axis of the diagram as their birth time are all 0, and the cycles give rise to the two blue points.

Related work. Finding representations of persistence diagrams that are well-suited to be combined with standard machine learning pipeline is a problem that has attracted a lot of interest these last years. A first family of approaches consists in finding convenient vector representations of persistence diagrams. For instance it involves interpreting diagrams as images in [AEK⁺17], extracting topological signatures with respect to fixed points whose optimal position are supervisedly learnt in [HKNU17], a square-root transform of their approximated pdf in [AVRT16].

A second family of approaches consists in designing specific kernel on the space of persistence diagrams, such as the multi-scale kernel of [RHBK15], the weighted Gaussian kernel of [KHF16] or the sliced Wasserstein kernel of [CCO17]. Those techniques have state-of-the-art behaviour on problems, but for drawback they require another step for an explicit representation, and are known to scale poorly.

A recent other line of work has managed to directly combine the uneasy structure of persistence diagrams to neural networks architectures [ZKR⁺17], [CCI⁺19]. Despite their successful performances, these neural networks are heavy to deploy and hard to understand. They are sometimes paired with a representation method as in [HKNU17], [HKN19].

Persistent homology in TDA

Persistent homology provides a rigorous mathematical framework and efficient algorithms to encode relevant

multi-scale topological features of complex data such as point clouds, time-series, 3D images... More precisely, persistent homology encodes the evolution of the topology of families of nested topological spaces $(F_\alpha)_{\alpha \in A}$, called filtrations, built on top of the data and indexed by a set of real numbers A that can be seen as scale parameters. For example, for a point cloud in a Euclidean space, F_α can be the union of the balls of radius α centered on the data points - see Figure 1. Given a filtration $(F_\alpha)_{\alpha \in A}$, its topology (homology) changes as α increases: new connected components can appear, existing connected components can merge, loops and cavities can appear or be filled, etc. Persistent homology tracks these changes, identifies features and associates, to each of them, an interval or lifetime from α_{birth} to α_{death} . For instance, a connected component is a feature that is born at the smallest α such that the component is present in F_α , and dies when it merges with an older connected component. The set of intervals representing the lifetime of the identified features is called the barcode of the filtration. As an interval can also be represented as a point in the plane with coordinates $(\alpha_{birth}, \alpha_{death})$, the persistence barcode is equivalently represented as an union of such points and called the persistence diagram - see [EH10, BCY18] for a more detailed introduction. The classical main advantage of persistence diagrams is that:

- (i) they are proven to provide robust qualitative and quantitative topological information about the data [CdSGO16];
- (ii) since each point of the diagram represents a specific topological feature with its lifespan, they are easily interpretable as features;
- (iii) from a practical perspective, persistence diagrams can be efficiently computed from a wide family of filtrations [The15].

However, as persistence diagrams come as unordered set of points with non constant cardinality, they cannot be immediately processed as standard vector features in machine learning algorithms. It can be beneficial to interpret persistence diagrams as measures, see for instance [CdSGO16], [CD18], and in this work we will use this paradigm.

Notations

Consider \mathcal{M}_d the set of finite measures on the d -dimensional ball $\mathcal{B}(0, R)$ of the Euclidean space \mathbb{R}^d with total mass smaller than M , for some given $M, R \in \mathbb{R}_+^2$.

We assume that the set of input persistence diagrams comes as an i.i.d. sample from a distribution of uniformly bounded diagrams, that is given $M, R \in \mathbb{R}_+^2$, let \mathcal{D} be the space of persistence diagrams with at most M points contained in the Euclidean disc $\mathcal{B}_2(0, R)$. The space \mathcal{D} is considered as a subspace of the set \mathcal{M}_2 of finite measures on $\mathcal{B}_2(0, R)$ with total mass smaller

than M : for any $D \in \mathcal{D}$, $D := \sum_{p \in D} \delta_p$ where δ_p is the Dirac measure centered at point p .

2 Methodology

In this section we introduce ATOL, a simple unsupervised data-driven method for measure vectorisation. ATOL allows to automatically convert a distribution of persistence diagrams into a distribution of feature vectors that are well-suited for use as topological features in standard machine learning pipelines.

To summarise, given a positive integer b , ATOL proceeds in two steps: it first computes a discrete measure in \mathbb{R}^d supported on b points that approximates the average measure of the distribution from which the input observations have been sampled. Second, it computes a set of well-chosen contrast functions centered on each point of the support of this measure, that are then used to convert each observation into a vector of size b .

2.1 Measure vectorisation through quantisation

We now introduce Algorithm 1 ATOL-featurisation in its generality, that is a featurisation method for elements of \mathcal{M}_d . The first step in our procedure is to use quantisation in space \mathcal{M}_d . Starting from an i.i.d. sample of measures X_1, \dots, X_n drawn from probability distribution \mathcal{L}_X on \mathcal{M}_d and given an integer budget $b \in \mathbb{N}^*$, we leverage recent algorithms and results from [LRC20] and produce a compact representation for the mean measure $\mathbb{E}(X)$. That is, we produce a distribution $P_{\hat{c}_{n,b}}$ supported on a fixed-length codebook $\hat{c}_{n,b} = \{c^{(1)}, \dots, c^{(b)}\}$ in the ambient space that aims to minimize over such distribution the distortion $R(P_c) := W_2^2(P_c, \mathbb{E}(X))$, the squared 2-Wasserstein distance to the mean measure. In practice, one considers the empirical mean measure \bar{X}_n and the k -means problem for this \bar{X}_n measure. Then the respective adaptations of Lloyd's [Llo82] and MacQueen's [Mac67] algorithms to the format of measures are introduced in [LRC20] and can readily be employed.

From this quantisation our aim is to derive spatial information on measures in order to discriminate between them. Much like one would compactly describe a point cloud with respect to its barycenter in a PCA, we describe measures based on a number of reduced difference to our mean measure approximate. To this end, our second step is to tailor b individual contrast functions each based on the estimated codebook that individually describe local regions. In other words we set to find regions of the space where measures seem to aggregate on average, and build a dedicated descriptor for those regions. We define and use the two following

contrast families $\mathbb{R}^2 \rightarrow \mathbb{R}_+$, for $i \in [b]$:

$$\Psi_i(\cdot, \hat{c}_{n,b}) : x \mapsto \exp \left[- \frac{|x - \hat{c}_{n,b}^{(i)}|_2}{\sigma_i(\hat{c}_{n,b})} \right], \quad (1)$$

$$\Phi_i(\cdot, \hat{c}_{n,b}) : x \mapsto \exp \left[- \frac{|x - \hat{c}_{n,b}^{(i)}|_2^2}{\sigma_i^2(\hat{c}_{n,b})} \right], \quad (2)$$

$$\text{where } \sigma_i(\hat{c}_{n,b}) := \min_{j \in [b], j \neq i} |\hat{c}_{n,b}^{(i)} - \hat{c}_{n,b}^{(j)}|_2 / 2. \quad (3)$$

These specific contrast functions are chosen to decrease away from the approximate mean centroid in a Laplacian (Ψ) or Gaussian (Φ) fashion. We choose the scale to roughly correspond to the minimum distance to the closest Voronoi cell in the corresponding codebook $\hat{c}_{n,b}$. The Ψ -exponential decrease will allow to properly separate a measure mixture in Theorem 1 so by default we make our arguments with that family in mind and will denote ATOL_Φ and ATOL_Ψ the Gaussian and Laplacian versions of the algorithm. To our knowledge there is nothing that prevents other, well designed contrast families to be substituted in their place, but this is beyond the scope of this paper. Given a family of contrast function and a mean measure codebook approximate, each element of \mathcal{M}_d can now be compactly described through the integrated contribution to each contrast functions: for $X \in \mathcal{M}_d$ and contrast function χ , let

$$X \cdot \chi(\cdot, \hat{c}_{n,b}) := \int_{x \in \mathbb{R}^d} \chi(x, \hat{c}_{n,b}) X(dx). \quad (4)$$

Our algorithm simply concatenates into a vector each of those contributions.

Algorithm 1: ATOL-featurisation

Data: Collection of measures $X_1, \dots, X_n \in (\mathcal{M}_d)^n$.

parameters: budget $b \in \mathbb{N}^*$,

contrast family $\chi \in \{\Psi, \Phi\}$.

Result: vectorisation map $v_{\text{ATOL}} : \mathcal{M}_d \rightarrow \mathbb{R}^b$.

- 1 Calibration step 1: apply $\{\text{batch or minibatch}\}$ quantization algorithm of the mean measure with fixed-length support from [LRC20], output: $\hat{c}_{n,b}$;
 - 2 Calibration step 2: adjust b measurable "contrast" functions $(\chi_i(\cdot, \hat{c}_{n,b}))_{i \in [b]}$ to the mean measure centroids;
 - 3 Vectorisation step: compute featurisation map:

$$v_{\text{ATOL}} : \mathcal{M}_d \rightarrow \mathbb{R}^b, X \mapsto \left[X \cdot \chi_i(\cdot, \hat{c}_{n,b}) \right]_{i \in [b]}.$$
-

Calibration step 1 is optimal for deriving space quantisation in the following sense: let the excess distortion be the difference between $R(P_{\hat{c}_{n,b}})$ the distortion of the resulting distribution based on codebook $\hat{c}_{n,b}$ and the optimal distortion of a distribution supported on b points. Then for either version of the quantisation algorithm (batch or minibatch), the excess distortion

can be controlled (respectively with high probability or on average) at a $\log n/n$ minimax speed under margin conditions, see Theorems 4 and 5 from [LRC20].

2.2 Topological learning

We now specialise our featurisation method to the context of topological learning when we are set in dimension $d = 2$ with $\mathcal{D} \subset \mathcal{M}_2$. Applying Algorithm 1 to a collection from \mathcal{D} is straightforward and allows to embed the complex, unstructured space \mathcal{D} in Euclidean terms. Set in the context of a standard learning problem, we introduce Algorithm 2 ATOL: Automatic Topologically-Oriented Learning. Let $\Omega := (X, y)$ with given observations X in some space \mathcal{X} corresponding to a known, partially available or hidden label $y \in \mathcal{Y}$. Assume that one has a way to extract topological features from \mathcal{X} , i.e. to derive a collection of diagrams associated to those elements, and let $\kappa : \mathcal{X} \rightarrow \mathcal{D}, X \mapsto D(X)$ be the corresponding map. Then applying Algorithm 1 to the resulting collection of diagrams provides some simplified topological understanding on elements X of this problem.

Algorithm 2: ATOL: Automatic Topologically-Oriented Learning

Data: Learning problem $\Omega := (X, y)$

with $X \in \mathcal{X}$ observations and $y \in \mathcal{Y}$ labels.

parameters: $\kappa : \mathcal{X} \rightarrow \mathcal{D}$ yielding topological descriptors, and budget $b \in \mathbb{N}^*$.

Result: Enhanced learning problem

$\tilde{\Omega} := ((X, v_{\text{ATOL}} \circ \kappa(X)), y)$ with euclidean, topological features $v_{\text{ATOL}} \circ \kappa(X) \in \mathbb{R}^b$.

- 1 Compute intermediate learning problem
 $\Omega_{\text{PH}} := ((X, \kappa(X)), y) \in (\mathcal{X} \times \mathcal{D}) \times \mathcal{Y}$ with persistent homology features, unfit for general machine learning routines;
 - 2 Use Algorithm 1 to transform it into a generic machine learning problem
 $\tilde{\Omega} := ((X, v_{\text{ATOL}} \circ \kappa(X)), y) \in (\mathcal{X} \times \mathbb{R}^b) \times \mathcal{Y}$.
-

Suppose now that persistence diagrams originate from distinct sources: assume that observed diagrams D_1, \dots, D_n are sampled with noise from a mixture model $D = \sum_{l=1}^L \pi_l D^{(l)}$ of distinct measures $D^{(1)}, \dots, D^{(L)}$ — by that we mean that any two measures in this set differ by at least one point. Call Z the latent variable associated to the mixture so that $D|Z = l \sim D^{(l)}$. The following results ensures that v_{ATOL} has separative power, i.e. that the vectorisation clearly separates the different sources:

Theorem 1 (Separation with ATOL).

For a given noise level assuming $\mathbb{E}(D)$ satisfies some (explicit) margin condition and for n and b large enough

there exists a non-empty segment for $\sigma_1, \dots, \sigma_b$ in Equation (1) such that for all $i, j \in [n]^2$, with high probability:

$$Z_i = Z_j \implies \|v_{\text{ATOL}}(D_i) - v_{\text{ATOL}}(D_j)\|_\infty \leq 1/4, \quad (5)$$

$$Z_i \neq Z_j \implies \|v_{\text{ATOL}}(D_i) - v_{\text{ATOL}}(D_j)\|_\infty \geq 1/2. \quad (6)$$

This result follows from Corollary 19 in [LRC20] and the explicit statement of the assumptions and margin conditions are classical but rather technical and are fully described in [LRC20] (see Definition 3 for the margin conditions).

Notice that establishing that a persistence diagram vectorisation method allows for separation, i.e. that diagrams from different clusters will be well-discriminated, has never been achieved to our knowledge. It is not sufficient to find a space quantisation that allows to discriminate a collection of diagrams from \mathcal{D} , the vectorisation based upon this quantisation could still miss a difference of interest depending on the chosen contrast family Ψ . The above theorem shows that ATOL overcomes this issue.

Note that in order to adjust the values for $\sigma_1, \dots, \sigma_b$ in Equation (1), we use a common heuristic instead of constant values and it is our intuition that the chosen, adaptive values of Equation (3) help the vectorisation perform better than what the theory can predict.

We point that as embedding map v_{ATOL} is automatically computed without knowledge of a learning task, its derivation is fully unsupervised. The representation is learned since it is data-dependent, but it is also agnostic to the task and only depends on getting a glimpse at an average persistence diagram. Using the minibatch quantisation step of [LRC20] is single-pass so the vectorisation algorithm has linear computation time in $O(n \times M \times b)$, therefore it is able to handle high-dimensional problems as long as corresponding diagrams are provided.

This featurisation is conceptually close to two other recent works. [HKNU17] computes a persistence diagram vectorisation through a deep learning layer that adjusts Gaussian contrast functions used to produce topological signatures much like our Calibration step 2. So in essence our approach substitutes quantisation to deep learning, with no need of supervision and allowing to provide mathematical guarantees. Next, the bag of word method of [ZLJ⁺19] uses an ad-hoc form of quantisation for the space of diagrams, then count functions as contrast functions to produce histograms as topological signatures. There are in fact sensible differences, that will ultimately translate in terms of effectiveness: Section 3.1 shows the ATOL-featurisation to produce state-of-the-art mean accuracy on two difficult multi-class classification problems (66.9 % on REDDIT5K and

51.6 % on REDDIT12K) that are also tackled by those papers where [HKNU17] report a mean accuracy of respectively 54.5% and 44.5%, and [ZLJ⁺19] report an accuracy of respectively 49.9% and 38.6%.

3 Competitive TDA-Learning

In this section we demonstrate experimentally the advantages of our approach. We show the ATOL framework to be competitive and state-of-the-art, but also versatile and easy to use with high automaticity.

3.1 Graph Classification

Learning problems involving graph data are receiving a strong interest at the moment, consider graph classification: $\Omega := (X, y) \in \mathcal{X} \times \mathcal{Y}$ is a finite family of graphs and available labels and one learns to map $\mathcal{X} \rightarrow \mathcal{Y}$.

Recently [CCI⁺19] have introduced a powerful way of extracting topological information from graph structures. They make use of heat kernel signatures (HKS) for graphs [HRG14], a spectral family of signatures (with diffusion parameter $t > 0$) whose topological structure can be encoded in the extended persistence framework, yielding four types of topological features with exclusively finite persistence. On both those points we refer to Sections 4.2 and 2 from [CCI⁺19]. Therefore for each graph and HKS diffusion time t the resulting topological descriptor are four persistence diagrams with all finite coordinates. For the entire set of problems to come we choose to use the same two HKS diffusion times to be .1 and 10, fueling the extended graph persistence framework and resulting in 8 persistence diagrams per considered graph. For budget in Algorithm 2 we choose $b = 80$ for all experiments, which means Algorithm 1 will rely on approximating the mean measure on ten points per diagram type and filtration. We make no use of (and automatically discard) graph attributes on edges or vertices that some dataset do possess, and no other sort of features are collected, so that our results are solely based on the graph structure of the problems. To sum up, Algorithm 2 here simply consists in reducing the original problem from Ω to $\tilde{\Omega} := (v_{\text{ATOL}} \circ \Phi(X), y)$ with $v_{\text{ATOL}} \circ \Phi(X) \in \mathbb{R}^{80}$. We stress that the embedding map v_{ATOL} from Algorithm 1 is computed each time using all diagrams from the training set, without supervision. To measure the worth of this embedding in this learning context, we evaluate the featurisation for classification purposes using the standard `scikit-learn` [PVG⁺11] random-forest classification tool with 200 trees and all other parameters set as default. On each problem we perform a 10-fold cross-validation procedure and average the resulting accuracies; we report accuracies and standard deviations over ten such experiments.

We use two sets of graph classification problems for benchmarking, one of Social Network origin and one of Chemoinformatics and Bioinformatics origin. They include small and large sets of graphs (MUTAG has 188 graphs, REDDIT12K has 12000), small and large graphs (IMDB-M has 13 nodes on average, REDDIT5K has more than 500), dense and sparse graphs (FRANKENSTEIN has around 12 edges per nodes, COLLAB has more than 2000), binary and multi-class problems (REDDIT12K has 11 classes), all available in the public page [KKM⁺16]. Computations are run on a single laptop (i5-7440HQ 2.80 GHz CPU), in batch version for datasets smaller than a thousand observations and mini-batch version otherwise. Average computing time of Algorithm 1 (the average time to calibrate the vectorisation map on the training set then compute the vectorisation on the entire dataset), are: less than .1 seconds for datasets with less than a thousand observations, less than 10 seconds for datasets that have less than 5 thousand observations, 25 seconds for REDDIT-5K, 50 seconds for REDDIT-12K and 110 seconds for the densest problem COLLAB. The results presented here are openly accessible (requiring open source library Gudhi [The15] and reproducible with the public repository github.com/martinroyer/atol).

We compare performances to the top scoring methods for these problems, to the best of our knowledge. Those methods are mostly graph kernels methods tailored to graph problems: two graph kernel methods based on random walks (RetGK1, RetGK11 from [ZWX⁺18]), one graph embedding method based on spectral distances (FGSD from [VZ17]), two topological graph kernel method (WKPI-kM and WKPI-kC from [ZW19]), one graph kernel combined with a graph neural network (GNTK from [DHS⁺19]) and one topological vectorisation method learnt by a neural network (PersLay from [CCI⁺19]). Competitor accuracy are quoted from their respective publication and we detail how they should be interpreted: for RetGK and WKPI and PersLay the evaluating procedure is done over ten 10-fold, just as ours is so the results directly compare; for FGSD the average accuracy over a single 10-fold is reported, and for GNTK the average accuracy and deviations is reported over a single 10-fold as well. When there are two or more methods under one label, we always report the best outcome.

Our results Table 1 are state-of-the-art or substantially improving the state-of-the-art on the Large Social Network datasets that are rather difficult multi-class problems. The results on the Chemoinformatics and Bioinformatics datasets Table 2 are state-of-the-art. These results are especially positive seeing how Algorithm 1 is generic and has been designed neither for graph experiments nor for persistence diagrams specif-

method problem	RetGK	FGSD	WKPI	GNTK	PersLay	ATOL $_{\Psi}$ (Laplacian)	ATOL $_{\Phi}$ (Gaussian)
REDDIT (5K, 5 classes)	56.1±.5	47.8	59.5±.6	—	55.6±.3	66.9±.3	67±.3
REDDIT (12K, 11 classes)	48.7±.2	—	48.5±.5	—	47.7±.2	51.6±.1	51.6±.2
COLLAB (5K, 3 classes)	81.0±.3	80.0	—	83.6±.1	76.4±.4	87.8±.2	88.1±.1
IMDB-B (1K, 2 classes)	71.9±1.	73.6	75.1±1.1	76.9±3.6	71.2±.7	74.3±.8	74.5±.5
IMDB-M (1.5K, 3 classes)	47.7±.3	52.4	48.4±.5	52.8±4.6	48.8±.6	47.8±.8	48.3±.7

Table 1: Mean accuracy and standard deviations for Large Social Network datasets.

method problem (size)	RetGK	FGSD	WKPI	GNTK	PersLay	ATOL $_{\Psi}$ (Laplacian)	ATOL $_{\Phi}$ (Gaussian)
MUTAG (188)	90.3±1.1	92.1	88.3±2.6	90.0±8.5	89.8±.9	86.7 ± .8	87.5 ± .6
COX2 (467)	81.4±.6	—	—	—	80.9±1.	78.8 ± .5	79.4±1.3
DHFR (756)	81.5±.9	—	—	—	80.3±.8	81.9 ± .8	83.1 ± .8
PROTEINS (1113)	78.0±.3	73.4	78.5±.4	75.6±4.2	74.8±.3	72.7 ± .4	72.4±.5
NCI1 (4110)	84.5±.2	79.8	87.5±.5	84.2±1.5	73.5±.3	78.8 ± .3	79.9 ±.2
NCI109 (4127)	—	78.8	87.4±.3	—	69.5±.3	77.6 ± .2	78.5±.3
FRNKSTN (4337)	76.4±.3	—	—	—	70.7±.4	72.8 ± .2	73.1±.3

Table 2: Mean accuracy and standard deviations for Chemoinformatics and Bioinformatics datasets - all binary classification problems.

ically, and seeing how the classification task has been entrusted to an external and generic learning tool. Contrary to competitors, the method does not require to construct a kernel or a neural network. Overall, the simplicity and absence of tuning hint at robustness and good generalisation power.

3.2 Discrete dynamical systems seen as measures

[AEK⁺17] use a synthetic, discrete dynamical system (used to model flows in DNA microarrays) with the following property: the resulting chaotic trajectories exhibit distinct topological characteristics depending on a parameter $r > 0$. The dynamical system is:

$$\begin{aligned} x_{n+1} &:= x_n + ry_n(1 - y_n) \quad \text{mod } 1, \\ y_{n+1} &:= y_n + rx_{n+1}(1 - x_{n+1}) \quad \text{mod } 1. \end{aligned}$$

With random initialisation and five different parameters $r \in \{2.5, 3.5, 4, 4.1, 4.3\}$, a thousand iterations per trajectory and a thousand orbits per parameter, a datasets of five thousand orbits is constituted and commonly used for evaluating topological methods. Figure 2 shows a few orbits generated with parameters $r \in \{4.0, 4.1\}$. For orbits generated with parameter $r = 4.1$, it happens that the initialisation spawns close to an attractor point that gives it the special shape as in the leftmost orbit. The problem of classifying this datasets in accordance to their underlying param-

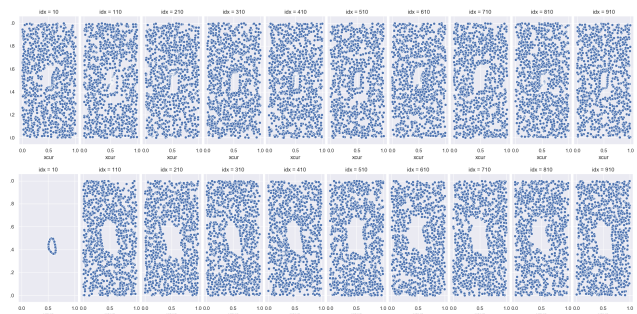


Figure 2: Example of synthesised orbits (x and y coordinates in the flat torus $[0, 1]^2$) with parameter 4.0 (top row) and 4.1 (bottom).

eter is rather uneasy and challenging. Some competitive topological methods have tackled this problem in the following way: after a learning phase with a 70/30 split, accuracy with the standard deviation over a hundred such experiments. The following results have been reported: 72.38±2.4 [RHBK15], 76.63±0.7 [KHF16], 83.6±0.9 [CCO17], 85.9±0.8 [LY18], and the state-of-the-art 87.7±1.0 with persistence diagrams in [CCI⁺19].

Since those discrete orbits can be seen as measures in $[0, 1]^2$, we apply our learning framework directly on the observed point cloud i.e. we use Algorithm 1 on the synthetic orbits and for learning we use the `scikit-learn` [PVG⁺11] random-forest classification tool (with 100 trees and all other parameters set as de-

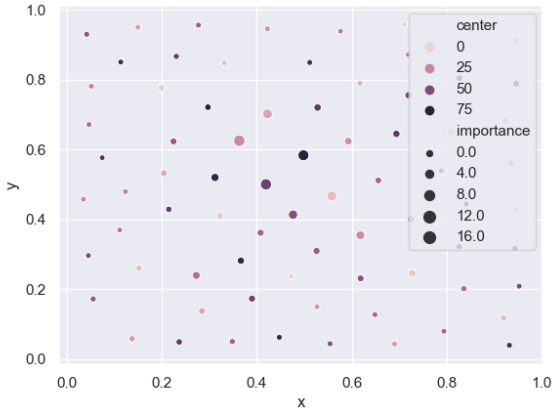


Figure 3: Example location of $b = 80$ quantisation centers and their respective importance in the clustering task.

fault) on the resulting vector. Note that in this context, our framework resembles that of image classification where instead of a fixed grid for measurement we have learnt centers from which to look at the data. After learning the center importance can be represented (see an example with 80 centers Figure 3) and naturally centers that can gather important geometrical information gain importance in this process.

Using the same 70/30 split procedure and repeating a hundred times with the Ψ -Laplacian contrast family and $b = 80$ centers (so in the exact same configuration as the graph experiment), we obtain 88.3 ± 0.8 mean accuracy and deviation. Therefore our results are also competitive for this high dimensional problem. But what is more, increasing the budget on this experiment yields sensible gains: using the Φ -Gaussian contrast family and a $b = 1000$ centers for cloud description allows to reach 95% accuracy or more, so it seems that this problem can be precisely described by a purely spatial approach — and our framework can be labeled as such in this context.

We also use this synthetic dataset to present additional experiments displayed Table 3, designed to understand parameter influence of Algorithm 1. The considered parameters are: (i) a high or low budget $b \in \mathbb{N}^*$ for describing the measure space, (ii) possible effect of the contrast functions Ψ or Φ to use for vectorisation of the quantised space, (iii) the proportion of training observations to use for deriving the quantisation, with 10% indicating that all a random selection of a tenth of the measures from the training set were used to calibrate Algorithm 1.

Naturally it is expected that augmenting the budget for vectorising the measure space will yield a better descrip-

tion of said space, and this intuition is confirmed by Table 1. But we stress that a weakness of Algorithm 1 is that once centers are fixed in the quantisation, space regions that are too far from these centers will necessarily be left out and the information they can carry with them. Therefore this intuition can sometimes be wrong if a lower number of centers happens to lead to a more pertinent quantisation. Next, the influence of the chosen contrast functions clearly show the Gaussian contrast functions to perform better than the other. Understanding the ability of such contrast functions to describe some particular observation space is challenging and left for future work. Lastly, ORBIT5K seems to be a dataset where the percentage of observations used in the calibration part of the algorithm does not weigh much on the final result for a budget $b = 80$ (it does have a significant influence when the budget is lower). This tells us that the calibration can be stable for a given level of information.

3.3 Topological score for time series, an industrial application

Finally we present an industrial application for time series, in a case where the learning problem is hard and no obvious solutions are to be found. This dataset consists in the following experiments: using commercially available simulator of a Japanese city road circuit course, about a hundred subjects are monitored and the intervals between successive heartbeats are recorded (RRI data sampled at 4Hz) for a 80 minutes drive that includes two periods of high-speed driving at the beginning and at the end of the experiment, and a low-speed driving period in the middle designed to induce sleepiness. For each experiment, an expert annotation (labeled NEDO score) produced from visual observation of the driver is made available, indicating sleepiness on a 1 to 5 class scale. We show four such experiments in Figure 4 (the RR-intervals have then been normalised).

This problem of retrieving the sleepiness level based on RRI levels is hard and ill-posed: there are strong individual differences in perceived reaction to a given situation, a single experiment per subject to learn behaviour from, and apparent noise or absence of signal in annotations, see e.g. subject 3 in Figure 4. Nevertheless we propose to use the ATOL framework to produce features meant reflect the sleepiness level in subjects based on RRI variations. The intent is that even though this will poorly reflect the latent sleepiness level, this could be enough to allow to catch jumps in the perceived attention level. The framework can readily be applied to time series in any given dimension and used to produce topological features. For this application we will follow a classical path: (i) use a sliding window decomposition on the RRI time-series, (ii) use

Budget effect					Contrast functions		Calibration effect		
$b = 2$	$b = 8$	$b = 20$	$b = 80$	$b = 200$	Φ -Gaussian	Ψ - Laplacian	10%	50%	100%
42.7 \pm .7	68.5 \pm 2.5	82.1 \pm 2.3	88.3 \pm .8	90.9 \pm .7	93.0 \pm .7	88.3 \pm .8	88.3 \pm .8	88.6 \pm .8	88.6 \pm .8
16.5 s	19.8 s	20.1 s	25.8 s	65.8 s	26 s	25.8 s	25.8 s	25 s	26 s

Table 3: Mean accuracy and deviation and vectorisation time over 10 experiments for ORBIT5K. Boldface blue indicate parameters by default, and only one parameter is varied at a time.

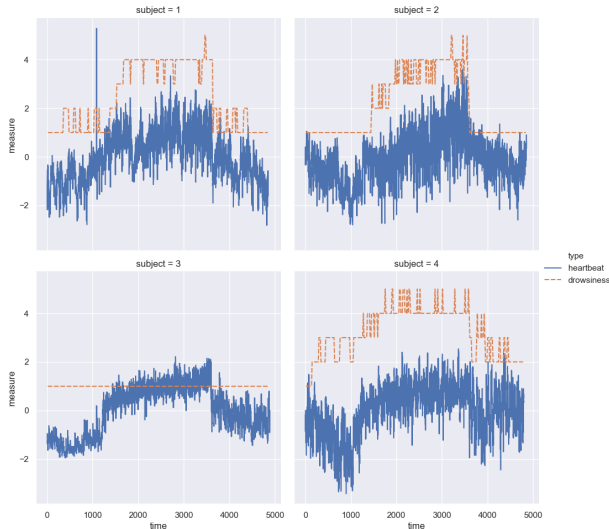


Figure 4: Normalized RRI time-series (blue) and annotated NEDO score (orange) for four subjects with the simulation time (x-axis, in seconds).

a time-delay embedding to transform said window into a point cloud, (iii) apply persistent homology analysis (we use DTM-filtration [ACG⁺18]) to produce persistence diagrams and (iv) vectorise persistence diagrams using Algorithm 1.

We concatenate those features with the mean and standard deviation statistics on the sliding-window. As for learning, we compute a learner based on other individuals’ features regressed to their NEDO scores, and use it to generate a score based on ATOL features (see middle and bottom row in Figure 5). Although this score imperfectly reflects the underlying NEDO score for a given patient, it can still have some uses. We set to detect two jumps on this topologically-augmented score using a Gaussian Kernel. We also compute a regressor based on the standard features without additional topological features, for comparison purposes, and also detect two jumps on this standard score.

Figure 5 shows two example results of our analysis. Each panel (top and bottom) consists in three time-series: the (hidden) NEDO score (top row), the ATOL-score computed from a regressor based on topological features (middle row), and a standard score computed from a regressor based solely on standard features (bot-

tom row). The changes of colour from blue to red and to blue indicates the changes in the experimental design for the driving simulation, i.e. the red portion indicates low-speed driving whereas the blue portions indicate high-speed driving periods. The black dotted lines indicate jumps detected from the ATOL representation, whereas the red dotted lines indicate jumps detected from the standard representation. In the top panel, the two series of jumps are concomitant, and almost an exact match to the underlying changes in the experimental design. In the bottom panel, an improvement over the standard score is caught with the ATOL score that better reflects the changes in latent NEDO score for this subject, two the point that the detected jumps are an exact match for the changes in experimental conditions. Overall, the ATOL score has less spikes and more regularity than the standard score, which is expected as the topological features are extracted posterior to a time-delay embedding procedure.

4 Conclusion

This paper introduces a vectorisation for measures in Euclidean spaces based on optimal quantisation procedures, then shows how this method can be employed in machine learning context and help process topological features. ATOL has a rather simple design, is multifaceted and ties theoretical guarantees to practical efficiency.

Moreover ATOL only depends on few simple parameters, namely the size b of the codebook and the choice of contrast functions. The study of the effect and the design of a method for automatic choice of these parameters deserves further analysis and is left to future work.

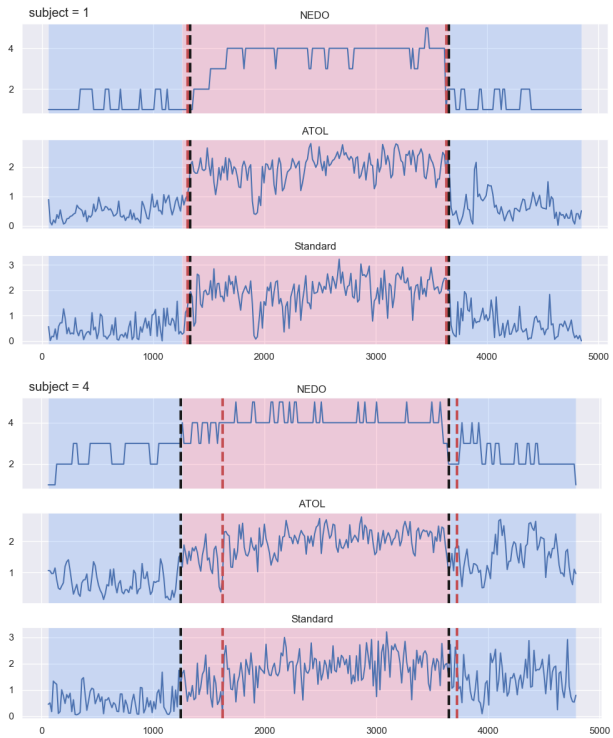


Figure 5: For two subjects (1 and 4), results of NEDO score (top row) regression and of a 2-jumps detection procedure, from ATOL procedure (ATOL-regression in middle row, yields jumps in black dashes) and standard features alone (standard-regression in bottom row, yields jumps in red dashes). Red zones indicates low-speed and blue zones indicates high-speed section in the experiment.

References

- [ACG⁺18] Hirokazu Anai, Frédéric Chazal, Marc Glisse, Yuichi Ike, Hiroya Inakoshi, Raphaël Tinarrage, and Yuhei Umeda. Dtm-based filtrations. In *Symposium on Computational Geometry*, 2018.
- [AEK⁺17] Henry Adams, Tegan Emerson, Michael Kirby, Rachel Neville, Chris Peterson, Patrick Shipman, Sofya Chepushtanova, Eric Hanson, Francis Motta, and Lori Ziegelmeier. Persistence images: a stable vector representation of persistent homology. *Journal of Machine Learning Research*, 18(8), 2017.
- [AVRT16] R. Anirudh, V. Venkataraman, K. N. Ramamurthy, and P. Turaga. A riemannian framework for statistical analysis of topological persistence diagrams. In *2016 IEEE Conference on Computer Vision and Pattern Recognition Workshops (CVPRW)*, pages 1023–1031, June 2016.
- [BCY18] Jean-Daniel Boissonnat, Frédéric Chazal, and Mariette Yvinec. *Geometric and Topological Inference*, volume 57. Cambridge University Press, 2018.
- [CCI⁺19] Mathieu Carrière, Frédéric Chazal, Yuichi Ike, Théo Lacombe, Martin Royer, and Yuhei Umeda. PersLay: A Simple and Versatile Neural Network Layer for Persistence Diagrams. *To appear in AISTATS 2020*, page arXiv:1904.09378, Apr 2019.
- [CCO17] Mathieu Carrière, Marco Cuturi, and Steve Oudot. Sliced Wasserstein kernel for persistence diagrams. In *International Conference on Machine Learning*, volume 70, pages 664–673, jul 2017.
- [CD18] Frédéric Chazal and Vincent Divol. The density of expected persistence diagrams and its kernel based estimation. In *SoCG 2018 - Symposium of Computational Geometry*, Budapest, Hungary, June 2018. Extended version of the SoCG proceedings, submitted to a journal.
- [CdSGO16] Frédéric Chazal, Vin de Silva, Marc Glisse, and Steve Oudot. *The structure and stability of persistence modules*. SpringerBriefs in Mathematics. Springer, 2016.
- [DHS⁺19] Simon S Du, Kangcheng Hou, Russ R Salakhutdinov, Barnabas Poczos, Ruosong Wang, and Keyulu Xu. Graph neural tangent kernel: Fusing graph neural networks with graph kernels. In H. Wallach, H. Larochelle, A. Beygelzimer, F. d’Alché Buc, E. Fox, and R. Garnett, editors, *Advances in Neural Information Processing Systems 32*, pages 5724–5734. Curran Associates, Inc., 2019.
- [EH10] Herbert Edelsbrunner and John Harer. *Computational Topology: An Introduction*. AMS, 2010.
- [HKN19] Christoph Hofer, Roland Kwitt, and Marc Niethammer. Graph filtration learning, 2019.
- [HKNU17] Christoph Hofer, Roland Kwitt, Marc Niethammer, and Andreas Uhl. Deep learning with topological signatures. In *Advances in Neural Information Processing Systems*, pages 1634–1644, 2017.
- [HRG14] Nan Hu, Raif Rustamov, and Leonidas Guibas. Stable and informative spectral signatures for graph matching. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 2305–2312, 2014.
- [KHF16] Genki Kusano, Yasuaki Hiraoka, and Kenji Fukumizu. Persistence weighted Gaussian kernel for topological data analysis. In *International Conference on Machine Learning*, volume 48, pages 2004–2013, jun 2016.
- [KKM⁺16] Kristian Kersting, Nils M. Kriege, Christopher Morris, Petra Mutzel, and Marion Neumann. Benchmark data sets for graph kernels, 2016. <http://graphkernels.cs.tu-dortmund.de>.
- [Llo82] Stuart P. Lloyd. Least squares quantization in PCM. *IEEE Trans. Inform. Theory*, 28(2):129–137, 1982.
- [LRC20] Clément Levrard, Martin Royer, and Frédéric Chazal. Optimal quantization of the mean measure and application to clustering of measures, 2020.
- [LY18] Tam Le and Makoto Yamada. Persistence Fisher kernel: a Riemannian manifold kernel for persistence diagrams. In *Advances in Neural Information Processing Systems*, pages 10027–10038, 2018.
- [Mac67] J. MacQueen. Some methods for classification and analysis of multivariate observations. In *Proc. Fifth Berkeley Sympos.*

-
- Math. Statist. and Probability (Berkeley, Calif., 1965/66)*, pages Vol. I: Statistics, pp. 281–297. Univ. California Press, Berkeley, Calif., 1967.
- [PVG⁺11] F. Pedregosa, G. Varoquaux, A. Gramfort, V. Michel, B. Thirion, O. Grisel, M. Blondel, P. Prettenhofer, R. Weiss, V. Dubourg, J. Vanderplas, A. Passos, D. Cournapeau, M. Brucher, M. Perrot, and E. Duchesnay. Scikit-learn: Machine learning in Python. *Journal of Machine Learning Research*, 12:2825–2830, 2011.
- [RHBK15] Jan Reininghaus, Stefan Huber, Ulrich Bauer, and Roland Kwitt. A stable multi-scale kernel for topological machine learning. In *IEEE Conference on Computer Vision and Pattern Recognition*, 2015.
- [The15] The GUDHI Project. *GUDHI User and Reference Manual*. GUDHI Editorial Board, 2015.
- [VZ17] Saurabh Verma and Zhi-Li Zhang. Hunt for the unique, stable, sparse and fast feature learning on graphs. In *Advances in Neural Information Processing Systems*, pages 88–98, 2017.
- [ZKR⁺17] Manzil Zaheer, Satwik Kottur, Siamak Ravanbakhsh, Barnabas Poczos, Ruslan Salakhutdinov, and Alexander Smola. Deep sets. In *Advances in Neural Information Processing Systems*, pages 3391–3401, 2017.
- [ZLJ⁺19] Bartosz Zieliski, Micha Lipiski, Mateusz Juda, Matthias Zeppelzauer, and Pawe Dotko. Persistence bag-of-words for topological data analysis. In *Proceedings of the Twenty-Eighth International Joint Conference on Artificial Intelligence, IJCAI-19*, pages 4489–4495. International Joint Conferences on Artificial Intelligence Organization, 7 2019.
- [ZW19] Qi Zhao and Yusu Wang. Learning metrics for persistence-based summaries and applications for graph classification. In H. Wallach, H. Larochelle, A. Beygelzimer, F. d’Alché Buc, E. Fox, and R. Garnett, editors, *Advances in Neural Information Processing Systems 32*, pages 9855–9866. Curran Associates, Inc., 2019.
- [ZWX⁺18] Zhen Zhang, Mianzhi Wang, Yijian Xiang, Yan Huang, and Arye Nehorai. RetGK: Graph Kernels based on Return Probabilities of Random Walks. In *Advances in Neural Information Processing Systems*, pages 3968–3978, 2018.