Metrics for Learning in Topological Persistence

Henri Riihimäki[1] and José Licón-Saláiz[2][0000−0002−8733−2256]

[1] Tampere University, Korkeakoulunkatu 7, 33720 Tampere, Finland
[2] Mathematical Institute, University of Cologne, Weyertal 86-90, 50931 Cologne, Germany
henri.riihimaki@tuni.fi,licon@math.uni-koeln.de

Abstract. Persistent homology analysis provides means to capture the connectivity structure of data sets in various dimensions. On the mathematical level, by defining a metric between the objects that persistence attaches to data sets, we can stabilize invariants characterizing these objects. We outline how so called contour functions induce relevant metrics for stabilizing the rank invariant. On the practical level, the stable ranks are used as fingerprints for data. Different choices of contour lead to different stable ranks and the topological learning is then the question of finding the optimal contour. We outline our analysis pipeline and show how it can enhance classification of physical activities data. As our main application we study how stable ranks and contours provide robust descriptors of spatial patterns of atmospheric cloud fields.

Keywords: Persistent homology · Topological learning · Stable rank · Atmospheric science

1 Persistence pipeline

1.1 Modelling data spaces

Topological data analysis (TDA) and particularly its subfield persistent homology, or persistence, aim at quantifying the global connectivity structure of data sets [1–3]. Given a set of data points it is often possible to endow it with some reasonable notion of relation between points, e.g. distance measure or correlation. Study of the connectivity is facilitated by first combining points into larger entities called simplices. A k-simplex is a declared subset of $k+1$ related points from the data set. Collection of simplices makes up a simplicial complex C, namely it is a collection of certain subsets of the data. Requirements are that if σ is a simplex in C then any subset of σ is also a simplex in C and that the intersection of two simplices is a simplex or the empty set. Above we have described an abstract simplicial complex. Simplices can always be realized geometrically in some $\mathbf{R}^n$ as convex hulls of their vertices: 0-simplices as points, 1-simplices as line segments, 2-simplices as filled triangles, 3-simplices as filled tetrahedra etc.

Simplicial complex is hence a model of the relational structure in the data. Relational structure can be modelled by a graph but graphs only consider pairwise relations between points. In many cases it makes sense to use higher-dimensional

connectivity instead modelled with simplices. As a justification consider the example explained in Fig. 1. More fundamental reason is that the simpicial approach views data as spaces spanned by their points and enables the use of powerful mathematical machinery of algebraic topology for the analysis of these spaces, as will be outlined in the following section.

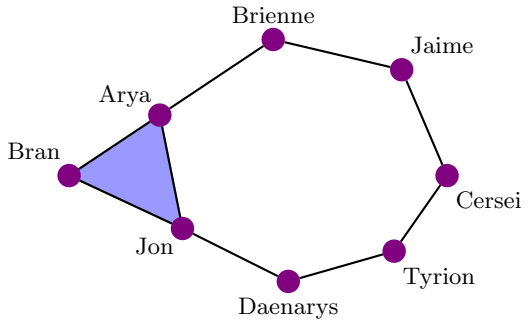

Fig. 1: Simplicial model of social relations. To model relations between $k+1$ points it is reasonable to use k-simplex for the purpose. Here the relations between {Arya, Bran, Jon} is depicted by the 2-simplex represented by the purple triangle. From the point of view of TDA the prominent feature of this data is the single loop structure, whereas a graph would see two loops (the closed path (Arya, Bran, Jon) spanning the other loop in this case).

For persistence analysis we define the relation to be a function R on the data with values in $\mathbf{R} = [0, \infty)$, i.e. $R(x, y) \mapsto t \in \mathbf{R}$ for data points x and y. Concretely we say that $k + 1$ data points x_i create a k-simplex at scale t if the points satisfy pairwise $R(x_i, x_j) \leq t$. This construction is called the Vietoris-Rips simplicial complex at scale t. At fixed scale we can then study the connectivity structure. As a standard example, when data is endowed with a distance measure, clustering at some fixed scale corresponds to the 0-dimensional connectivity by only looking at the connected components of the simplicial complex. Simplicial complexes can also contain 1-dimensional connectivity information in the form of loops and holes (see Fig. 1), 2-dimensional information in the form of voids or cavities, etc. These are collectively called topological features.

Persistence aims to quantify the topological features in a data set and use this information for data analysis. Loop structure might signal about a recurrent dynamics of the phenomenon behind the data. Various dimensional voids can mark lack of information and connectivity or insufficient data collection. Finding such voids in data sets has aroused interest in different areas of data analysis community, see for example [4] and references therein. As noted in [4], voids can also indicate non-allowed combinations of feature values of data vectors.

One immediate difficulty arises in the simplicial modelling above: what is the appropriate scale of R to capture the connectivity in various dimensions of

an arbitrary set of points? Persistence circumvents this by forming simplicial complexes at all scales $t \geq 0$ and capturing the evolution of topological features. If a simplex is generated at scale t it is then present at any subsequent scale and the simplicial complexes are connected by inclusions: $\cdots \subseteq C_a \subseteq C_b \subseteq C_c \subseteq \cdots$ for $\cdots \leq a \leq b \leq c \leq \ldots$ The end result of the modelling step is then a mapping called filtration, $(D, R) \times \mathbf{R} \to (C_t, \subseteq_t)_{t \in \mathbf{R}}$, where (D, R) denotes a data set with real-valued relation and $(C_t, \subseteq_t)_{t \in \mathbf{R}}$ denotes an $\mathbf{R}$-parameterized sequence of simplicial complexes and inclusions.

1.2 Algebraic fingerprinting

Filtration contains all the information about the relations in the data set on various scales. It is therefore very complicated object for infering the global structure of data and simplification is thus necessary. TDA employs tools from mathematical field of algebraic topology, essentially it uses homology of simplicial complexes which transforms the geometric information into algebraic information. We will outline the algorithm for computing homology to illustrate its very implementable nature and to gain intuition on why we are interested in homology in data analysis. For details into homology and its computation see [5–7]. For simplicity we fix the field of coefficients to be $\mathbf{F}_2$, the field with two elements 0 and 1. Let C be a simplicial complex and denote by C_k its set of k-simplices. Concretely, C_0 consists of the points of the original data set.

1) Choose an ordering (starting from zero) on C_0 and use it to order elements in any simplex. If $\{\text{Arya}, \text{Bran}, \text{Jon}\}$ is a 2-simplex in Fig. 1, fix the order in which the points are listed and denote this ordered simplex by $[\text{Arya}, \text{Bran}, \text{Jon}]$.

2) For natural numbers k and $0 \leq i \leq k$ and a simplex σ in C_k, define a function $d_i \colon C_k \to C_{k-1}$ such that $d_i(\sigma)$ is a simplex in C_{k-1} formed by removing from σ its i-th element. The ordering on C_0 was needed to specify the i-th element in a simplex. For example, $d_1([\text{Arya}, \text{Bran}, \text{Jon}]) = [\text{Arya}, \text{Jon}]$.

3) For any natural number k, let $\Delta(C)_k$ be the vector space over $\mathbf{F}_2$ with a base given by all simplices in C_k. An element τ in $\Delta(C)_k$ is then given by a linear combination $\tau = \sum_{\sigma \in C_k} t_\sigma \sigma,\ t_\sigma \in \mathbf{F}_2$. The base for $\Delta(C)_2$ of the simplicial complex in Fig. 1 would be [Arya, Bran, Jon] whereas [Arya, Bran]+[Bran, Jon]+[Arya, Jon] would be linear combination of three basis elements in $\Delta(C)_1$.

4) Define $\partial_k \colon \Delta(C)_k \to \Delta(C)_{k-1}$ to be the linear function assigning to a base element given by a simplex σ in C_k the linear combination $\sum_{i=0}^{k} d_i(\sigma)$ of k-1-simplices. The map ∂_k is called the boundary operator. Then $\partial_k([\text{Arya}, \text{Bran}, \text{Jon}])$ $= [\text{Bran}, \text{Jon}] + [\text{Arya}, \text{Jon}] + [\text{Arya}, \text{Bran}]$. The boundary operator thus formalizes the intuition that $[\text{Bran}, \text{Jon}] + [\text{Arya}, \text{Jon}] + [\text{Arya}, \text{Bran}]$ forms the boundary of $[\text{Arya}, \text{Bran}, \text{Jon}]$. Define $\Delta(C)_{-1} = 0$ and $\Delta(C)_k = 0$ for $k > m$, where 0 denotes the zero vector space.

5) The boundary operators connect the various simplices of a simplicial complex together. Computationally the matrices of boundary operators store the global connectivity information in their elements, with coefficient field $\mathbf{F}_2$ these are just binary matrices. Homology on degree k of a simplicial complex C (over

coefficients $\mathbf{F}_2$) is then defined as a quotient vector space:

$$H_k(C) = \frac{\text{kernel of } \partial_k \colon \Delta(C)_k \to \Delta(C)_{k-1}}{\text{image of } \partial_{k+1} \colon \Delta(C)_{k+1} \to \Delta(C)_k}, \quad \text{for } k \geq 0.$$

As noted in step 4) above, some 1-simplices might form the boundary of a 2-simplex. Some 1-simplices on the other hand might form the boundary of an actual hole in the simplicial complex as in Fig. 1. Similarly some k-1-simplices might form the boundary of a k-simplex and some might form the boundary of a k-dimensional hole. By its definition homology quotients out linear combinations of simplices that are boundaries and we are left with those that actually represent linearly independent k-dimensional holes in the complex. For $k = 0$, H_0 measures the number of linearly independent points that make up boundaries of 1-simplices, effectively the number of connected components.

Homology thus gives us exactly the global connectivity information of the relational structure of data that we seek. The full complexity of a filtration is now simplified by applying homology on degree k. Each simplicial complex is turned into a homology vector space and the inclusion functions are turned into linear maps. The result is an $\mathbf{R}$-parameterized sequence of vector spaces and linear maps: $\cdots \to H_k(C_a) \to H_k(C_b) \to H_k(C_c) \to \cdots$. We will abbreviate $H_k(C_a)$ as $H_{k,a}$. In this parameterized sequence the dimensions of homology vector spaces encode topological information: $H_{0,t}$ effectively measuring the number of connected components, $H_{1,t}$ measuring the number of one-dimensional holes and $H_{k,t}$ those of k-dimensional voids at scale t.

This algebraic step gives a mapping $(C_t, \subseteq_t)_{t\in\mathbf{R}} \to (H_{k,t}, \to_t)_{t\in\mathbf{R}}$. The obtained result is not an arbitrary $\mathbf{R}$-parameterized vector space. The vector spaces $H_{k,t}$ are finite dimensional and there are finitely many numbers $0 < t_0 < \cdots < t_n$ in $\mathbf{R}$ such that the map $H_{k,a} \to H_{k,b}$ may not be an isomorphism only if $a < t_i \leq b$, for i in $\{0, \ldots, n\}$. These considerations follow from the fact that data sets always contain only finite number of points so topological changes in the relational structure can only occur in discrete steps. Such parameterized vector spaces are called tame [8]. An essential result in persistence theory is that any tame $\mathbf{R}$-parameterized vector space decomposes into interval indecomposables called bars and the collection of bars in such a decomposition is unique [9]. Bars are enumerated by pairs of numbers $b < d$ in $\mathbf{R}$. The bar $[b, d)$ at scale t is either a 1-dimensional vector space, if $b \leq t < d$, and the zero vector space otherwise. The maps between any non-zero vector spaces in a bar are isomorphisms. For a bar $[b, d)$, some topological feature is understood to have appeared in the simplicial complex at filtration value b. It is then present in the subsequent simplicial complexes until filtration value d. For example, points in the data might connect to create a 1-dimensional loop. This loop persists until at some larger filtration value the points connect further to higher dimensional simplices and the loop vanishes. The bar decomposition can be visualized in a stem plot on a $(b, d - b)$-coordinate system as shown later in Fig. 3.

2 Topological learning

The actual data analysis step in persistence pipeline is to infer information from the **R**-parameterized sequence of homology vector spaces and linear maps obtained from the map $(D, R) \times \mathbf{R} \to (C_t, \subseteq_t)_{t\in\mathbf{R}} \to (H_{k,t}, \to_t)_{t\in\mathbf{R}}$ constructed above. To simplify notation we let **R-Vec** denote the space of tame **R**-parameterized sequences of vector spaces $V = \cdots \to V_a \to V_b \to V_c \to \cdots$. Our framework of extracting information from objects in this space is through stabilizing a rank invariant attached to them. Aim of the paper is on the practical data analysis aspects and we only outline the theoretical backgound. For more details we refer to [8, 14, 15].

2.1 Rank invariant

The rank, or the dimension, is the fundamental invariant characterizing vector spaces. Similarly we want to assign rank for sequences of vector spaces in **R-Vec**. Let V be in **R-Vec**. Due to tameness there is a sequence $0 < t_0 < \cdots < t_k$ in $\mathbf{R}$ such that $V_a \to V_b$ is not an isomorphism only if $a < t_i \leq b$. Recall that for a linear map $f\colon X \to Y$ its cokernel is the quotient vector space of Y by the image of f: $\mathrm{coker} f = Y/\mathrm{im} f$. We then define

$$\beta_0(V) = V_0 \oplus \mathrm{coker}(V_0 \to V_{t_0}) \oplus \mathrm{coker}(V_{t_0} \to V_{t_1}) \cdots \oplus \mathrm{coker}(V_{t_{k-1}} \to V_{t_k}),$$

where V_0 is the homology vector space in V at filtration value 0. Let us consider what information $\beta_0(V)$ carries. Since the maps $V_{t_i} \to V_{t_{i+1}}$ are not isomorphisms the cokernels may not be zero. The quotient by the image removes from the homology vector space $V_{t_{i+1}}$ the generators, or basis elements, which come from previous non-isomorphic homology vector space. β_0 is thus a vector space of the new homology generators that appear in the sequence of homology vector spaces. In the context of filtrations of input data sets, this is a way of keeping track of how topological features created by the relational structure evolve in the simplicial complexes of the filtration.

For V in **R-Vec**, its rank is now defined to be a discrete invariant given by the number

$$\mathrm{rank}(V) = \dim(\beta_0(V)) =$$
$$\dim(V_0) + \dim(\mathrm{coker}(V_0 \to V_{t_0})) + \cdots + \dim(\mathrm{coker}(V_{t_{k-1}} \to V_{t_k})).$$

2.2 Hierarchical stabilization and contour metrics

The rank defined above is not a stable invariant. Effectively the number $\mathrm{rank}(V)$ measures the smallest number of homology generators of V. A small perturbation of input data can result in a number of non-essential homology generators. We therefore seek to stabilize the rank invariant to deal with inherent noise in data. Our approach is a general framework for stabilizing discrete invariants.

Let T be a set of interesting objects and I the attached invariant. For us T is of course a collection of $\mathbf{R}$-parameterized vector spaces associated to data sets with $\mathbf{R}$-valued relation and I is the rank. The key in converting a discrete invariant into a stable one is to choose a (pseudo)metric d on T. Once a metric is chosen, we can define an ε-radius ball around $X \in T$, $B(X, \varepsilon) = \{Y \mid d(X, Y) \leq \varepsilon\}$, and look at the function $\widehat{I}_d(X)$ taking the minimum value of I on balls around X with increasing radii ε:

$$\widehat{I}_d(X)(\varepsilon) = \min\{I(Y) \mid Y \in B(X, \varepsilon)\}.$$

Since we are minimizing the invariant in larger and larger balls around X, the function $\widehat{I}_d(X)$ is decreasing and piecewise constant, namely a simple function. Due to being a decreasing function with non-negative values, there is some t such that for all $s \geq t$ in $\mathbf{R}$, $\widehat{I}_d(X)(s) = \widehat{I}_d(X)(t)$. The function $\widehat{I}_d(X)$ is thus eventually constant with a limit, $\lim \widehat{I}_d(X)$.

The needed metrics in the stabilization can be shown [15] to arise from so called contours. Contour is function $C : \mathbf{R} \times \mathbf{R} \to \mathbf{R}$ satisfying the following inequalities for all v, w, ε, τ in $\mathbf{R}$:

1. $v \leq C(v, \varepsilon) \leq C(w, \tau)$, for $v \leq w$ and $\varepsilon \leq \tau$,
2. $C(C(v, \varepsilon), \tau) \leq C(v, \varepsilon + \tau)$.

For example, $C(v, \varepsilon) = v + \varepsilon$, $C(v, \varepsilon) = v + \varepsilon^2$ and $C(v, \varepsilon) = r^\varepsilon v$ with a positive number r are all examples of contours. The contour

$$C(v, \varepsilon) = v + \varepsilon \tag{1}$$

is called the standard contour. There is a generic way of producing contours. Let $f\colon \mathbf{R} \to (0, \infty)$ be a function with strictly positive values which we refer to as density. Then it can be shown that the function $C(v, \varepsilon)$ given by

$$C(v, \varepsilon) = v + \int_y^{y+\varepsilon} f(x)dx,$$

where for v in $\mathbf{R}$, we have taken the unique y in $\mathbf{R}$ such that $v = \int_0^y f(x)dx$. For more background on contours we refer to [14].

It is also shown in [14] how the choice of a contour leads to a pseudometric d_C in $\mathbf{R}$-$\mathbf{Vec}$. The stabilization of the rank invariant with respect to the chosen contour is then defined as

$$\widehat{\operatorname{rank}}_C V(\varepsilon) = \min\{\operatorname{rank}(W) \mid W \in \mathbf{R}\text{-}\mathbf{Vec} \text{ and } d_C(V, W) \leq \varepsilon\}. \tag{2}$$

As noted above, the stable rank function $\widehat{\operatorname{rank}}_C V$ is decreasing and piecewise constant and from $\mathbf{R}$ to $\mathbf{R}$.

Our approach does not conceptually rely on the bar decomposition of V in $\mathbf{R}$-$\mathbf{Vec}$. Computation of the decomposition is however standard procedure in persistence analysis with various dedicated implementations [3] and when the

decomposition is given, the stable rank can be computed algorithmically in a very efficient way:

$$\widehat{\mathrm{rank}}_C V(\varepsilon) = |\{[b_i, d_i) \mid C(b_i, \varepsilon) < d_i\}|. \tag{3}$$

The stable rank of V at ε is thus the number of those bars in the decomposition that satisfy the relation between the start and end points given by the contour. In practical computations the limit of $\widehat{\mathrm{rank}}_C V$ is always zero, or can be set to zero.

By fixing some values of ε the contour $C(v, \varepsilon)$ reduces to a single variable function and we can plot it. In Fig. 3 this is illustrated with few values of ε in the stem plot of a bar decomposition. This visualization is helpful in understanding how the contour affects the stable rank in Eq. 3: the value of stable rank $\widehat{\mathrm{rank}}_C V(\varepsilon)$ at ε is the number of bars that reach over the function $C(v, \varepsilon)$. If the function $C(v, \varepsilon)$ has lower values it therefore makes bars relatively longer and vice versa with larger values. The contour can thus be seen as controlling pointwise with respect to b_i the length scale that we use to measure bars.

2.3 Topological learning with stable ranks

The stable rank attached to an input data set is a topological fingerprint of the data. In the actual data analysis task these fingerprints are used in, for example, classifying various data sets. Recall from the construction above that the stable rank is derived by choosing a contour function C which induces a metric d_C needed for the stabilization in Eq. 2. Each choice of a contour gives a different stable rank capturing different aspects of the data. The learning step in our pipeline is then to choose an appropriate contour for the analysis at hand and we explore this in Section 3.

As stable ranks are $\mathbf{R}$-valued functions we have various choices of metrics for comparing them. In particular we have standard L_p-metrics for $p \geq 1$:

$$L_p(f, g) = \left(\int_0^\infty |f(t) - g(t)|^p dt \right)^{1/p}.$$

We can also define interleaving distance between functions f and g. We first define the set of horizontal shifts of the functions satisfying the indicated inequalities:

$$S = \{\varepsilon \in \mathbf{R} \,|\, f(t) \geq g(t + \varepsilon) \text{ and } g(t) \geq f(t + \varepsilon) \text{ for all } t \in \mathbf{R}\}.$$

The interleaving distance $d_{\bowtie}$ is then defined as the minimum of those shifts:

$$d_{\bowtie}(f, g) = \begin{cases} \inf(S) & \text{, if } S \text{ is non-empty,} \\ \infty & \text{, otherwise.} \end{cases}$$

In Section 3 we use these constructions in demonstrating our approach with concrete data analyses. We emphasize that our approach does not rely on any

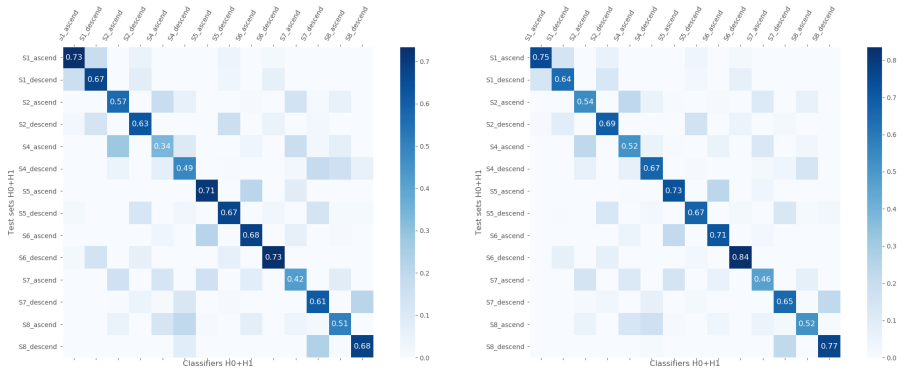

Fig. 2: Confusion matrices for the classification of ascending and descending stairs activities with standard contour (left) and with contour visualized in Fig. 3.

algebraic decomposition of persistence and is thus applicable to multiparameter persistence [16]. The initial theory behind our pipeline was indeed formulated for multiparameter persistence in [8] and later specialized for 1-parameter persistence in [14]. In the case of one parameter we obtain the convenient algorithm, Eq. 3, for computing stable rank.

Traditional view in persistence analysis has been that long bars in the bar decomposition are of importance and smaller bars are noise. This view, however, is challenged by many recent studies showing that smaller features carry important information: study of brain artery trees in [17], functional networks of [18], analysis of protein structure in [19] and the relation of observed diffraction peaks to small loops in atomic configurations of amorphous silica in [20]. With our pipeline we can flexibly choose different contours to learn what are in fact the essential features in the data. To produce the bar decompositions we used Ripser software [22].

3 Applications

3.1 Classifying physical activities

We studied PAMAP2 data obtained from [10] to classify different physical activities. The data consisted of seven persons performing different activities such as walking, cycling or sitting. Test subjects were fitted with three Inertial Measurements Units (IMUs) and a heart rate monitor. Measurements were registered every 0.1 seconds. Each IMU measured 3D acceleration, 3D gyroscopic and 3D magnetometer data. One data set thus consisted 28-dimensional data points indexed by 0.1 second timesteps.

We looked at two activities which from the outset are very similar and expected to be difficult to distinguish: ascending and descending stairs. For the analysis we randomly sampled without replacement 100 points from each data set, repeated

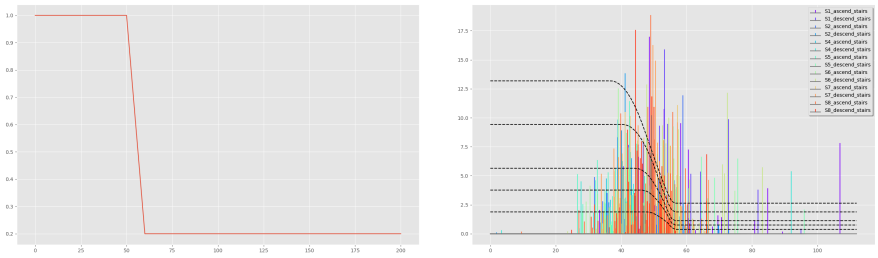

Fig. 3: Density function used for H_1 stable rank in the activities classification (left) and contour lines for few values of ε (right). Persistence bar stems are shown for single data sets from each (subject,activity) class.

100 times. For each subject we thus obtained 100 resamplings from the activity data and computed their stable ranks with respect to a chosen contour. Out of these we computed the point-wise means of 40 stable ranks in H_0 and H_1. These means were used as classifiers, denoted by $\hat{P}_{H_0}$ and $\hat{P}_{H_1}$. Altogether we had 14 classifier pairs $(\hat{P}_{H_0}, \hat{P}_{H_1})$ corresponding to all (subject, activity) combinations. Remaining 60 stable ranks in H_0 and H_1 were used as test data and denoted by T_{H_0} and T_{H_1}. For a test pair (T_{H_0}, T_{H_1}) we found

$$\min(L_1(\hat{P}_{H_0}, T_{H_0}) + L_1(\hat{P}_{H_1}, T_{H_1}))$$

by computing L_1 distances between the test pair and all classifier pairs. The classification is successful if the minimum is obtained with $\hat{P}_\bullet$ and $T_\bullet$ belonging to the same (subject, activity) class in both H_0 and H_1.

For cross validation we randomly sampled which of the stable ranks constitute classifier and which are test data for the class. Result for 20-fold cross validation is shown in the confusion matrix on the left in Fig. 2 for the standard contour (defined in Eq. 1). Each cell of the confusion matrix is the number of classifications in the corresponding classifier (columns) and test data (rows) pair relative to the total number of test stable ranks which was 60. Correct classifications are on the diagonal. Overall accuracy (mean over diagonal of the confusion matrix) with standard contour was 60%.

We then repeated the above cross validation process but using a different contour in computing H_1 stable rank. Contour was obtained from the density function on the left side of Fig. 3. Contour lines and the bars from persistence computation are visualized on the right side of Fig. 3. This contour puts more weight on topological features appearing with larger filtration scales. Cross-validation results are shown on the right in Fig. 2. Overall accuracy increased to 65%. Note particularly increase in the accuracy of subject 4. Also noteworthy is that ascendings mainly get confused with ascendings of different subjects and the same for descendings. These (subject,activity) data thus exhibit different character and changing the contour we could make this difference more pronounced.

3.2 Cloud pattern characterization

We analysed the spatial distribution of shallow cumulus clouds. These clouds form in fair-weather conditions due to the convective transport of heat and moisture in the atmosphere. Convection is a classic example of a pattern-forming system [12, 13]. Cloud formation is known to be influenced by diverse physical processes across spatial scales ranging from molecular sizes to kilometers. Such spatial scales and all their physical variables cannot be explicitly resolved in numerical climate models, which calls for the development of cloud parametrization schemes. Moreover, the spatial distribution of clouds influences their formation processes. It is therefore important to include this distribution in parametrization schemes. This problem has been studied from different perspectives, notably the influence of land surface conditions on cloud formation [23]. Here we describe an approach based on persistence and the use of stable ranks as descriptors of the spatial distribution of clouds. See [11] for further results and references.

The data was produced by the Dutch Atmospheric Large-Eddy Simulation model and covered the time period between 09:00h and 18:00h during one day, saved for analysis at 15 minute intervals, with model setup similar to that in [24]. We simulated 10 days with different initial conditions. The data consists of large amount of physical information from which cloud fields can be extracted. The spatial simulation domain in x, y, z coordinates is $12.8 \times 12.8 \times 5$ in kilometers with horizontal resolution of 50 meters and vertical resolution of 40 meters. The computation domain thus consists of cells. A homogeneous land surface is prescribed and the lateral boundaries are periodic. The 3D cloud field from the simulation domain was then flattened in the z-direction onto a 2D plane by taking the maximum liquid water content, ql, values in the vertical direction. The resulting cloud fields are then as visualized in Fig. 4(b).

An important issue in the study of cloud formation is the quantification of spatial organization, or lack thereof, in a given cloud field. While methods to study spatial distributions exist in the statistical literature for objects which can be idealized as points, it is harder to work with objects that possess a spatial extent (i.e. area or volume), as clouds do. This leads to the necessity of computing a point representation for a cloud before being able to assess the spatial distribution of the cloud field. Here we consider three different representations: assigning to each cloud its geometric centroid, its point with maximum ql value, and a set of its points chosen at random.

A common metric in the assessment of spatial organization is the I_{org} index [21], defined as follows. For a two-dimensional cloud field, such as the one shown in Fig. 4(b), index the connected components (the individual clouds) as c_i, and compute their geometric centroids, $\bar{c}_i$. We are interested in how the spatial distribution of the $\bar{c}_i$ compares to what we would expect under complete spatial randomness (CSR), that is, if the centroids represent a realization of a homogeneous Poisson point process. To that end, we consider the nearest-neighbor distances d_i, which are defined as $d_i = \min\{d(\bar{c}_i, x) \mid x \in \bar{\mathcal{C}} \setminus \{\bar{c}_i\}\}$, where $\bar{\mathcal{C}}$ represents the set of all centroids. The cumulative distribution function (CDF)

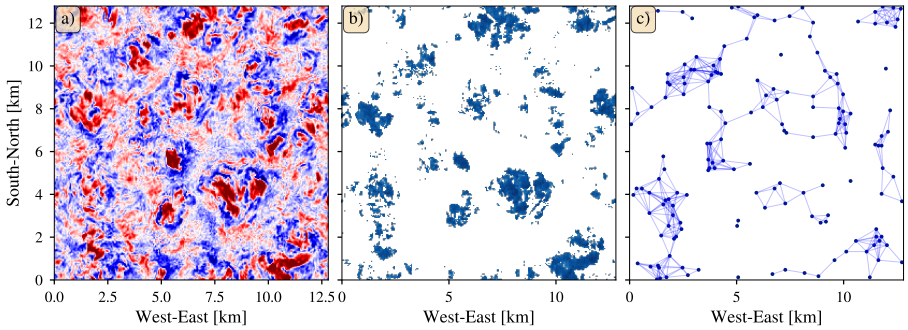

Fig. 4: a) Values of the vertical wind velocity w for a two-dimensional horizontal slice at an altitude of 1.8 km. This corresponds to cloud base height (red – $w > 0$; blue – $w < 0$). b) Column liquid water content ql (i. e. the maximum liquid water value in the vertical direction). c) Point representation of the cloud field by the local maxima of ql (only connected components formed by at least 3 cells are considered), and 1-simplices of the Vietoris-Rips filtration using the distance relation between the points, at a distance scale of 1.5 km.

of the d_i is

$$G_{d_i}(r) = P[d_i \leq r],$$

which in the case of a Poisson point process has the analytic expression

$$G_{CSR}(r) = 1 - \exp\left(-\lambda \pi r^2\right),$$

where λ is the Poisson intensity parameter. The value of I_{org} is then defined to be the area under the graph $(G_{CSR}(r), \hat{G}(r))$, where

$$\hat{G}(r) = \frac{\#\{\bar{c}_i \in \bar{\mathcal{C}} \mid d_i \leq r\}}{\#\{\bar{c}_i \in \bar{\mathcal{C}}\}}$$

is the empirical estimator of $G(r)$. If $\hat{G}$ matches well with G_{CSR}, the value of I_{org} will be close to 0.5. A value larger than this suggests spatial clustering, while a smaller one suggests dispersion or regularity.

Let S_i^* denote the stable rank of H_i with respect to the standard contour (Eq. 3), normalized by its value at 0. If we define the function $G_{PH}^i(r) = 1 - S_i^*(r)$, we note that it increases monotonically towards 1. In fact, since the normalized stable rank at r is an indication of the relative amount of homological features that persist beyond r, the function $G_{PH}^i(r)$ can be understood as the empirical CDF of homological persistence.

For n realizations of a Poisson point process with intensity parameter λ, we find that their normalized stable ranks S_i^*, and therefore also G_{PH}^i, oscillate within a narrow band (see Fig. 5). At this point we do not have an analytic expression for the stable rank functions obtained from a Poisson point process, but we can define persistent homology analogues to the I_{org} index via a Monte Carlo

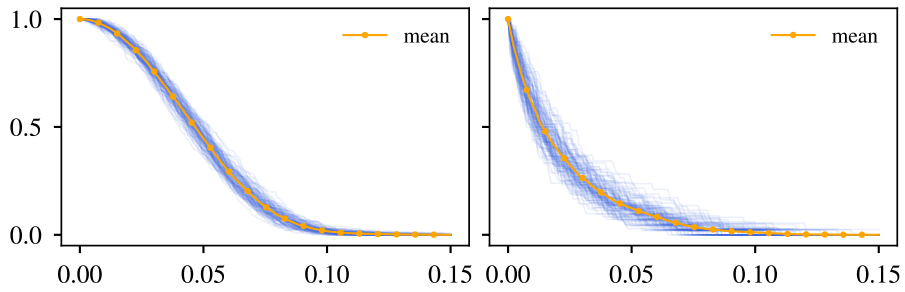

Fig. 5: Stable rank functions obtained from 100 realizations of a homogeneous Poisson point process with $\lambda = 100$. Left: S_0^*. Right: S_1^*.

procedure by taking the area under the curves defined by $(G^i_{PH,CSR}(r), G^i_{PH}(r))$. In the case of a point process in the plane we would then get two values $I_{PH,0}$ and $I_{PH,1}$. We define the index as their arithmetic mean,

$$I_{PH} = \frac{I_{PH,0} + I_{PH,1}}{2}. \tag{4}$$

We tested the performance of the index I_{PH} defined above, and compared it to the corresponding values of I_{org} in the dataset consisting of 360 distinct cloud fields (36 per simulation day). The values of both indices are shown in Fig. 6. Each panel shows the 360 values of each index for all cloud fields, computed using 4 different point representations. Panel A shows the values obtained from assigning to each connected component its point with maximum ql value (local maxima); panel B shows the indices obtained when using the local maxima but only of those components with size at least 3 grid cells (all smaller components are ignored). Panel C shows the results of using the geometric centroid of each connected component. Finally, for panel D the geometric centroids were used after discarding the smaller components. These small components can be attributed to numerical imprecision in the underlying model, and hence are not physically meaningful.

As discussed above, if these indices have a value close to 0.5, it would indicate that the point process that they are evaluated on is close to complete spatial randomness, or a Poisson point process. In the simulations used here, we have cause to expect spatially random behavior: the domain size is too small to allow for deep convection and spatial organization to happen. Moreover, the lack of land surface features or patterns means there are no forcings at different spatial scales. Thus the spatial distribution of physical variables is dominated by the characteristic patterns present in atmospheric turbulence, itself an essentially random process. The values of the persistent homology index I_{PH} strongly support this hypothesis, while I_{org} exhibits values in general larger than 0.5. This can be attributed to the fact that it is based on nearest-neighbor distances only, whereas the stable rank functions reflect the spatial relationships of the points

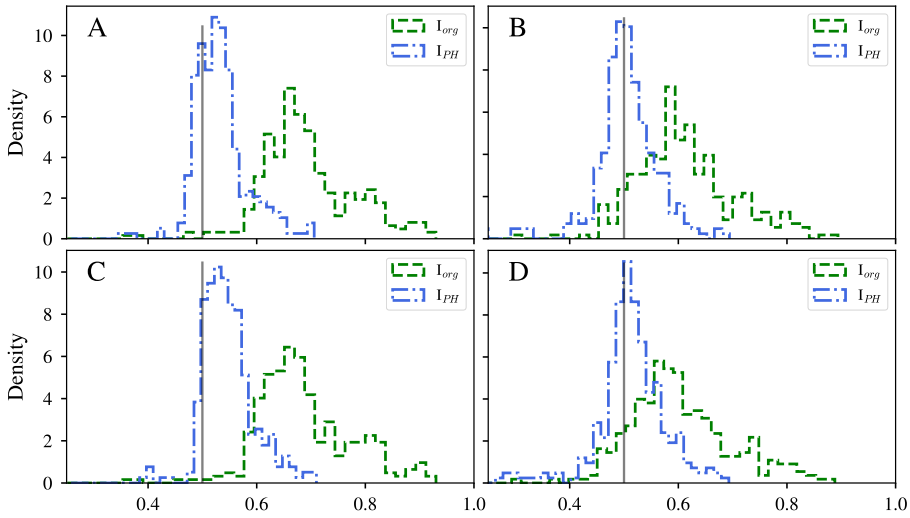

Fig. 6: Density histograms of the I_{org} index and I_{PH} (Eq. 4) for 360 distinct cloud fields. **A**: ql max, **B**: ql max removing cloud structures with size smaller than 3 cells, **C**: Geometric centroids, **D**: Geometric centroids removing cloud structures with size smaller than 3 cells.

throughout all spatial scales. This is confirmed by the fact that removing the smaller structures in the fields (those less than 3 grid cells in size) brings the values of I_{org} closer to 0.5 on average, whereas the average for I_{PH} is barely affected. This highlights the fact that, by virtue of using all the spatial information available, the persistent homology based method is inherently more robust than any nearest-neighbor method.

This result has been arrived at by using the standard contour only, which implies that spatial randomness in these cloud fields is obtained when all spatial scales present in the data are given the same weight. It is possible to obtain different morphological classifications of the same fields by using alternative contours, which emphasize spatial features differently at varying scales, as presented with the classification in Section 3.1. We used standard contour and contours visualized in Fig. 7. These contours are referred to as contour 1, denoted C_1, and contour 2, denoted C_2.

To reduce the effect of sampling, 10 random samples were drawn from each of the 360 cloud fields, with sample rate 5% of cloud size. To each cloud field we assign the mean stable rank of these 10 samples. Stable ranks were computed in H_1 with respect to standard contour, contour 1 and contour 2 and normalized to give S_1^* function as explained above. After removing those cloud fields without H_1 features, we have 254 normalized stable ranks S_1^* for each class of contours. Distance matrices using interleaving, L_1- and L_2-metrics (see Section 2.3) were then computed for the three different classes of stable ranks. Dendrograms from

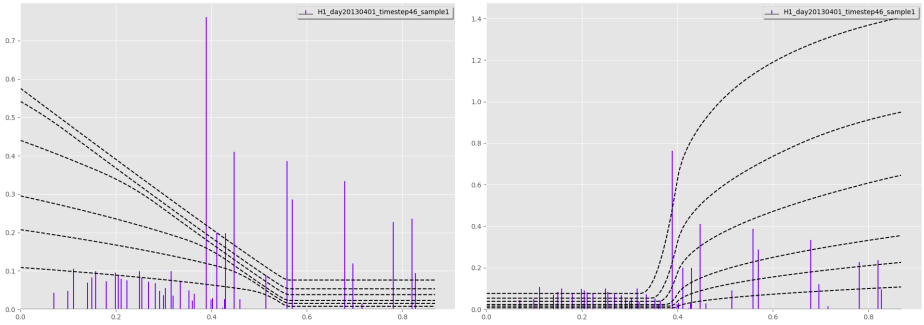

Fig. 7: Contour 1 (left) and contour 2 (right) used in the analysis of cloud fields. Stem plot is from one sampling of a cloud field at one time step.

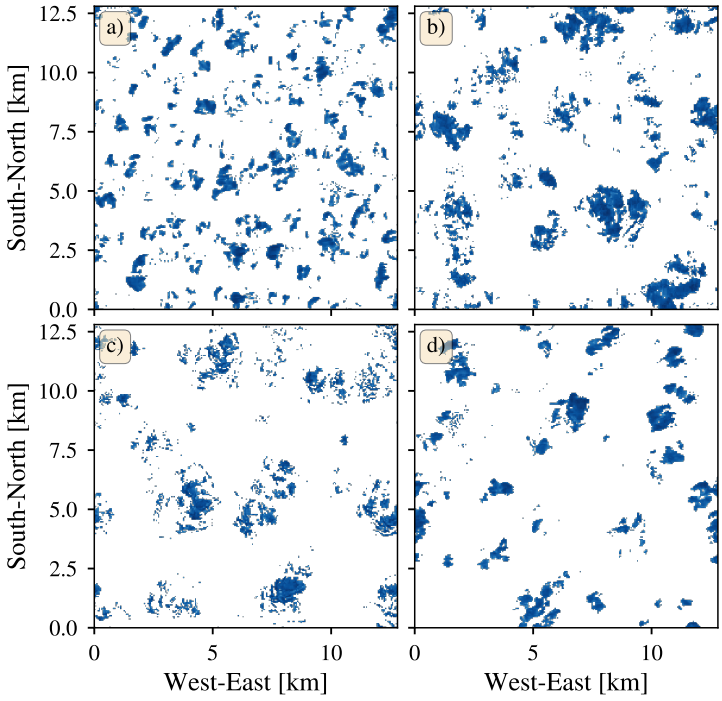

Fig. 8: Cloud fields which are classified into different clusters, according to the methodology described in the text. We use the H_1 stable ranks and the interleaving metric to compute the distances between them. a) and b) are classified using contour C_1, and have I_{org} values of 0.45 and 0.53 respectively. Cloud cover is similar at 14% for both. c) and d) are classified with C_2, and have I_{org} values of 0.65 and 0.63 respectively, and cloud cover for both is 9.2%.

the distance matrices were visually analyzed to decide on a number of clusters of stable ranks. From these computations the interleaving distance gave the clearest clustering results. With respect to contours, C_1 and C_2 gave better clustering than standard contour.

An example of diverging morphological characteristics educed from the $C_{1,2}$ clustering schemes is shown in Fig. 8: (a) and (b) are representatives of two different clusters obtained by using contour C_1, while (c) and (d) stem from clusters in the C_2 classification. As expected from the definition of the contours, the classifications they induce are influenced by different spatial scales. Namely, despite the fact that cloud fields a) and b) have identical cloud cover, and their I_{org} values are very similar, the large-scale distribution of the individual clouds is significantly different for both. In similar fashion, both c) and d) are indistinguishable in terms of cloud cover and I_{org}, yet are distinguished by the spatial pattern of smaller structures, even if the large-scale distribution is similar in both. These kind of geometrical considerations can be used to determine an appropriate contour for a specific task. With higher dimensional data, such as with physical activities above, stem plots can guide in determining what features to emphasize and what contour achieves this.

This study of cloud fields shows that the use of stable rank functions as descriptors for spatial distributions can reveal morphological properties which other methods cannot. Crucially, the possibility of changing the contour enriches the scope for determining such properties. Future investigation in this direction will address questions such as: what the optimal contour is for a given problem, what these methods can reveal about the temporal evolution of cloud formation, and how the homological properties thus discovered can be related to different physical variables in the system. From general data analysis point of view, particularly the automatic optimization of contours is crucial for making our pipeline a full scale machine learning approach.

Acknowledgments

We gratefully acknowledge Roel Neggers for providing the DALES simulation data. JLS acknowledges support by the DFG-funded transregional research collaborative TR32 on Patterns in Soil–Vegetation–Atmosphere Systems.

References

1. Carlsson, G.: Topological pattern recognition for point cloud data. Acta Numerica **23**, pp. 289–368 (2014)
2. Oudot, S.: Persistence Theory: From Quiver Representations to Data Analysis. American Mathematical Society, Providence, RI (2015)
3. Otter, N., Porter, M., Tillmann, U., Grindrod, P., Harrington, H.: A Roadmap for the Computation of Persistent Homology. EPJ Data Science **6**(17), (2017)
4. Lemley, J., Jagodzinski, F., Andonie, R.: Big Holes in Big Data: A Monte Carlo Algorithm for Detecting Large Hyper-rectangles in High Dimensional Data. In: IEEE

40th Annual Computer Software and Applications Conference, pp. 563–571. IEEE Computer Society, Los Alamitos, CA (2016)
5. Rotman, J.: An Introduction to Algebraic Topology. Springer, New York, NY (1998)
6. Edelsbrunner, H., Harer, J.: Computational Topology: An Introduction. American Mathematical Society, Providence, RI (2010)
7. Kaczynski, T., Mischaikow, K., Mrozek, M.: Computational Homology. Springer, New York, NY (2004)
8. Chachólski, W., Lundman, A., Ramanujam, R:, Scolamiero, M., Öberg, S.: Multidimensional Persistence and Noise. Foundations of Computational Mathematics **17**(6), pp. 1367–1406 (2017)
9. Zomorodian, A., Carlsson, G.: Computing Persistent Homology. Discrete & Computational Geometry **33**(2), pp. 249–274 (2005)
10. PAMAP, Physical Activity Monitoring for Aging People homepage, http://www.pamap.org
11. Licón-Saláiz, J., Riihimäki, H., van Laar, T.: Topological characterization of shallow cumulus cloud fields using persistent homology. In: Proceedings of the 8th International Workshop on Climate Informatics, pp. 107–110. National Center for Atmospheric Research, Boulder, CO (2018)
12. Mizushima, J.: Mechanism of the Pattern Formation in Rayleigh–Bénard Convection. Journal of the Physical Society of Japan **63**(1), pp. 101–110 (1994)
13. Cerisier, P., Rahal, S., Rivier, N.: Topological correlations in Bénard–Marangoni convective structure. Physical Review E - Statistical, Nonlinear, and Soft Matter Physics **54**(5), pp. 5086–5094 (1996)
14. Chachólski, W., Riihimäki, H.: Metrics and stabilization in one parameter persistence. arXiv:1904.02905, (2019)
15. Gäfvert, O., Chachólski, W.: Stable Invariants for Multidimensional Persistence. arXiv:1703.03632, (2017)
16. Carlsson, G., Zomorodian, A.: The theory of multidimensional persistence. Discrete & Computational Geometry **42**(1), pp. 71–93 (2009)
17. Bendich, P., Marron, J., Miller, E., Pieloch, A., Skwerer, S.: Persistent homology analysis of brain artery trees. The Annals of Applied Statistics **10**(1), pp. 198–218 (2016)
18. Stolz, B., Harrington, H., Porter, M.: Persistent homology of time-dependent functional networks constructed from coupled time series. Chaos **27**(4), pp. 047410-1–047410-17 (2017)
19. Xia, K., Wei, G.-W.: Persistent homology analysis of protein structure, flexibility, and folding. International Journal for Numerical Methods in Biomedical Engineering **30**(8), pp. 814–844 (2014)
20. Hiraoka, Y., Nakamura, T., Hirata, A., Escolar, E., Matsue, K., Nishiura, Y.: Hierarchical structures of amorphous solids characterized by persistent homology. Proceedings of National Academy of Sciences **113**(26), pp. 7035–7040 (2016)
21. Tompkins, A., Semie, A.: Organization of tropical convection in low vertical wind shears: Role of updraft entrainment. Journal of Advances in Modeling Earth Systems **9**(2), pp. 1046–1068 (2017)
22. Bauer, U., Ripser software. github.com/Ripser/ripser
23. Rieck, M., Hohenegger, C., van Heerwaarden, C. C.: The Influence of Land Surface Heterogeneities on Cloud Size Development. Monthly Weather Review **142**(10), pp. 3830–3846 (2014)
24. Neggers, R. A., Siebesma, A., and Heus, T.: Continuous single-column model evaluation at a permanent meteorological supersite. Bulletin of the American Meteorological Society **93**(9), pp. 1389–1400 (2012)