

Comparing Persistence Diagrams Through Complex Vectors

Barbara Di Fabio¹(✉) and Massimo Ferri^{1,2}

¹ ARCES, University of Bologna, Bologna, Italy
{barbara.difabio,massimo.ferri}@unibo.it

² Department of Mathematics, University of Bologna, Bologna, Italy

Abstract. The natural pseudo-distance of spaces endowed with filtering functions is precious for shape classification and retrieval; its optimal estimate coming from persistence diagrams is the bottleneck distance, which unfortunately suffers from combinatorial explosion. A possible algebraic representation of persistence diagrams is offered by complex polynomials; since far polynomials represent far persistence diagrams, a fast comparison of the coefficient vectors can reduce the size of the database to be classified by the bottleneck distance. This article explores experimentally three transformations from diagrams to polynomials and three distances between the complex vectors of coefficients.

Keywords: Persistence diagram · Shape analysis · Viète formulas · Precision · Recall

1 Introduction

Persistent homology has already proven to be an effective tool for shape representation in various applications, in particular when the objects to be classified, compared or retrieved have a natural origin. The interplay of geometry and topology in persistence makes it possible to capture qualitative aspects in a formal and computable way, yet it doesn't suffer of the excessive freedom of mere topological equivalence. The privileged tool for shape comparison is the natural pseudo-distance [11], which is scarcely computable. Luckily, persistence diagrams condense the essence of the shape concept of the observer in finite sets of points in the plane [13, 16]; moreover, the bottleneck distance (a.k.a. matching distance) between persistence diagrams yields an optimal lower bound to the natural pseudo-distance [9, 10]. There is a problem: the bottleneck distance suffers from combinatorial explosion [8], so it becomes hard to scan a large database when it comes to retrieval. Approximations, smart organization of the database according to the metric, progressive application of different classifiers come to help, but the problem is lightened, not solved.

A paradigm shift came from an idea of Claudia Landi [15]: represent the persistence diagram as the set of complex roots of a polynomial; then comparison can be performed on coefficients. Two problems arise: one — which comes

from the nature itself of persistence diagrams — is that in real situations there are a lot of points near the “diagonal” $\{(u, v) \in \mathbb{R}^2 : u = v\}$, due to noise so less meaningful in shape representation, but with a heavy impact on polynomial coefficients; another problem — coming from polynomial theory — is that little distance of polynomial roots implies little distance of coefficients, but the converse is false.

A completely different polynomial representation of barcodes (equivalently: of persistence diagrams) is the one through tropical algebra [17], closely adapting to the bottleneck distance.

The Contribution of the Paper. We face the first problem — the existence of points near the diagonal — by performing a plane warping which takes all the line $u = v$ to 0, so points near the diagonal actually become close together. Making noise points close and around zero diminishes their contribution to polynomial coefficients, above all to the first (and most relevant) ones: sum of roots, sum of pairwise products of roots, etc. As for the second problem — the fact that close coefficients may not mean close roots — we explore the use of polynomial comparison as a preprocessing phase in shape retrieval, i.e. as a very fast way of getting rid of definitely far objects, so that the bottleneck distance can be computed only for a small set of candidates, in the same line as [5]. The results are satisfactory: in some of our experiments the bottleneck distance even turns out to be unnecessary.

2 Preliminaries

In persistence, the shape of an object is usually studied by choosing a topological space X to represent it, and a function $f : X \rightarrow \mathbb{R}$, called a *filtering* (or *measuring*) *function*, to define a family of subspaces $X_u = f^{-1}((-\infty, u])$, $u \in \mathbb{R}$, nested by inclusion, i.e. a filtration of X . The name “persistence” is bound to the idea of ranking topological features by importance, according to the length of their “life” through the filtration. The basic assumption is that the longer a feature survives, the more meaningful or coarse the feature is for shape description. In particular, structural properties of the space X are identified by features that once born never die; vice-versa, noise and shape details are characterized by a short life. To study how topological features vary in passing from a set of the filtration into a larger one we use homology. A nice feature of this approach is modularity: The choice of different filtering functions may account for different viewpoints on the same problem (different shape concepts) or for different tasks. For further details we refer to [1, 12].

Persistent homology groups of the pair (X, f) — i.e. of the filtration $\{X_u\}_{u \in \mathbb{R}}$ — are defined as follows. Given $u \leq v \in \mathbb{R}$, we consider the inclusion of X_u into X_v . This inclusion induces a homomorphism of homology groups $H_k(X_u) \rightarrow H_k(X_v)$ for every $k \in \mathbb{Z}$. Its image consists of the k -homology classes that are born before or at the level u and are still alive at the level v and is called the *kth persistent homology group of (X, f) at (u, v)* . When this group is finitely generated, we denote by $\beta_k^{u,v}(X, f)$ its rank.

The usual, compact description of persistent homology groups of (X, f) is provided by the so-called *persistence diagrams*, i.e. multisets of points whose abscissa and ordinate are, respectively, the level at which k -homology classes are created and the level at which they are annihilated through the filtration. If a homology class does not die along the filtration, the ordinate of the corresponding point is set to $+\infty$.

At the moment, our approach to convert persistence diagrams into complex vectors can be applied only when neglecting these points at infinity. Hence, we focus on the subsets of proper points of the classical persistence diagrams, known in literature as *ordinary persistence diagrams* [6]. For simplicity we still call them “persistence diagrams”. We underline that this choice is not so restrictive since the number of points at infinity depends only on the homology of the space X , and persistent homology provides a finite distance between two pairs if and only if the considered spaces are homeomorphic.

We use the following notation: $\Delta^+ = \{(u, v) \in \mathbb{R}^2 : u < v\}$, $\Delta = \{(u, v) \in \mathbb{R}^2 : u = v\}$, and $\overline{\Delta^+} = \Delta^+ \cup \Delta$.

Definition 1. Let $k \in \mathbb{Z}$ and $(u, v) \in \Delta^+$. The multiplicity $\mu_k(u, v)$ of (u, v) is the finite non-negative number defined by

$$\lim_{\varepsilon \rightarrow 0^+} (\beta_k^{u+\varepsilon, v-\varepsilon}(X, f) - \beta_k^{u-\varepsilon, v-\varepsilon}(X, f) - \beta_k^{u+\varepsilon, v+\varepsilon}(X, f) + \beta_k^{u-\varepsilon, v+\varepsilon}(X, f)).$$

Definition 2. The k th-persistence diagram $D_k(X, f)$ is the set of all points $(u, v) \in \Delta^+$ such that $\mu_k(u, v) > 0$, counted with their multiplicity, union the points of Δ , counted with infinite multiplicity. We call proper points the points of a persistence diagram lying on Δ^+ .

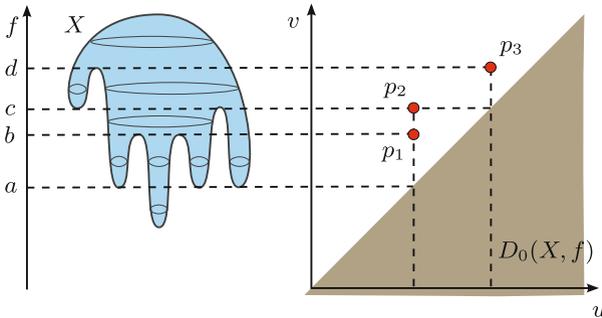


Fig. 1. Left: The height function f on the space X . Right: The associated 0 th-persistence diagram $D_0(X, f)$.

Figure 1 displays an example of persistence diagram for $k = 0$. The surface $X \subset \mathbb{R}^3$ is filtered by the height function f . $D_0(X, f)$ has three proper points p_1, p_2, p_3 since the abscissa of these points corresponds to the level at which new

connected components are born along the filtration, while the ordinate identifies the level at which these connected components merge with existing ones. In terms of multiplicity, this means that $\mu_0(p_i) > 0, i = 1, 2, 3$, and $\mu_0(p) = 0$ for every other point $p \in \Delta^+$. To see, for example, that $\mu_0(p_1) = 2$, where $p_1 = (a, b)$, it is sufficient to observe that, for every $\varepsilon > 0$ sufficiently small, it holds that $\beta_0^{a+\varepsilon, b-\varepsilon}(X, f) = 4, \beta_0^{a+\varepsilon, b+\varepsilon}(X, f) = 2, \beta_0^{a-\varepsilon, b-\varepsilon}(X, f) = \beta_0^{a-\varepsilon, b+\varepsilon}(X, f) = 1$, and apply Definition 1. In an analogous way, it can be observed that $\mu_0(p_2) = \mu_0(p_3) = 1$.

Persistence diagrams comparison is usually carried out through the so called *bottleneck distance* because of the robustness of these descriptors with respect to it. Roughly, small changing in a given filtering function (w.r.t. the max-norm) produces just a small changing in the associated persistence diagram w.r.t. the bottleneck distance [4, 7]. The bottleneck distance between two persistence diagrams measures the cost of finding a correspondence between their points. In doing this, the cost of taking a point p to a point p' is measured as the minimum between the cost of moving one point onto the other and the cost of moving both points onto the diagonal. In particular, the matching of a proper point p with a point of Δ can be interpreted as the destruction of the point p . Formally:

Definition 3. *Let D, D' be two persistence diagrams. The bottleneck distance $d_B(D, D')$ is defined as*

$$d_B(D, D') = \min_{\sigma} \max_{p \in D} d(p, \sigma(p)),$$

where σ varies among all the bijections between D and D' and

$$d((u, v), (u', v')) = \min \left\{ \max \{|u - u'|, |v - v'|\}, \max \left\{ \frac{v - u}{2}, \frac{v' - u'}{2} \right\} \right\}$$

for every $(u, v), (u', v') \in \overline{\Delta^+}$.

3 Persistence Diagrams vs Complex Vectors

Driven by the awareness that, in the experimental framework, evaluating the bottleneck distance can be computationally expensive, making its usage not practicable on large datasets, in this work we propose a new procedure based on the preliminary idea introduced in [15]. We translate the problem of comparing directly two persistence diagrams through the bottleneck distance into the problem of comparing complex vectors associated with each persistence diagram through appropriate metrics between vectors. The components of these complex vectors are complex polynomials' coefficients obtained as follows. Firstly, we define a certain transformation taking points of persistence diagrams to the set of complex numbers. Secondly, we construct a complex polynomial having the obtained complex numbers as roots.

In this paper, we consider the three transformations below:

- $R : \overline{\Delta^+} \rightarrow \mathbb{C}$, with $R(u, v) = u + iv$,

$$\begin{aligned}
 - S : \overline{\Delta^+} &\rightarrow \mathbb{C}, \text{ with } S(u, v) = \begin{cases} \frac{v-u}{\alpha\sqrt{2}} \cdot (u+iv), & \text{if } (u, v) \neq (0, 0) \\ (0, 0), & \text{otherwise} \end{cases}, \\
 - T : \overline{\Delta^+} &\rightarrow \mathbb{C}, \text{ with } T(u, v) = \frac{v-u}{2} \cdot (\cos \alpha - \sin \alpha + i(\cos \alpha + \sin \alpha)),
 \end{aligned}$$

where $\alpha = \sqrt{u^2 + v^2}$.

R, S, T are continuous maps; R and S are also injective on $\overline{\Delta^+}$ and Δ^+ , respectively. We define the multiplicity of a complex number in the range of R, S, T to be the sum of the multiplicities of the points belonging to its preimage (this is necessary because of the non-injectivity of T on Δ^+ , although a preimage containing more than one proper point of the diagram has zero probability to occur). The main differences among these deformations are the following: the deformation R acts as the identity, just passing from \mathbb{R}^2 to \mathbb{C} ; the deformation S warps the diagonal Δ to the origin, and takes points of Δ^+ to points of $\{z \in \mathbb{C} : \text{Re}(z) < \text{Im}(z)\}$; the deformation T warps the diagonal Δ to the origin, and takes points of Δ^+ to points of \mathbb{C} . An example showing how S and T transform a persistence diagram is represented in Figure 2. Both S and T seemed to be preferable to R because points near Δ — due to noise — have to be considered close to each other in the bottleneck distance, although they may be very far apart in Euclidean distance. Taking them to be all near the origin would then also reduce their impact in the sums and sums of products which will build the polynomial coefficients we are going to compare. In particular, T was designed to distribute the image of those noise points around zero, whereas S makes them near zero, but all on one side: in the half-plane of \mathbb{C} corresponding to Δ^+ . T has two drawbacks: it is not injective on Δ^+ and does not behave well with respect to simple transformations.

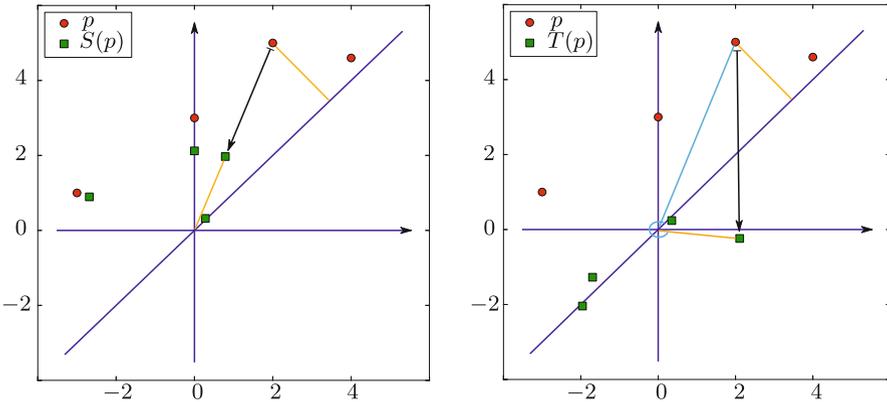


Fig. 2. A persistence diagram with its transformations S (left) and T (right). Same colors identify same lengths.

Let D be a persistence diagram, and $p_1 = (u_1, v_1), \dots, p_s = (u_s, v_s)$ its proper points with multiplicity r_1, \dots, r_s , respectively. Let now the complex numbers z_1, \dots, z_s be obtained from p_1, \dots, p_s by one of the transformations R, S or T . We associate to D the complex polynomial $f_D(t) = \prod_{j=1}^s (t - z_j)^{r_j}$. We are actually interested in the coefficient sequence of $f_D(t)$, which we can compute by Viète's formulas (see Algorithm 2).

Once we have the polynomials $f_D(t) = t^n - a_1 t^{n-1} + \dots + (-1)^i a_i t^{n-i} + \dots + (-1)^n a_n$ and $f_{D'}(t) = t^m - a'_1 t^{m-1} + \dots + (-1)^j a'_j t^{m-j} + \dots + (-1)^m a'_m$ corresponding to persistence diagrams D, D' , we face a first problem, given by the possibly different degrees n and m ($m < n$ say). Because of their expression in terms of roots, we prefer to compare coefficients with the same index, rather than coefficients relative to the same degree of t . We manage this problem by adding $n - m$ null coefficients to $f_{D'}(t)$, i.e. multiplying $f_{D'}(t)$ by t^{n-m} , which amounts to adding the complex number zero with multiplicity $n - m$. In so doing, we can build two vectors of complex numbers $(a_1, \dots, a_n), (a'_1, \dots, a'_n)$ of the same length and are ready to compute a distance between them. By continuity of Viète's formulas, close roots imply close coefficients. Hence, two persistence diagrams that are close in terms of bottleneck distance have close associated polynomials. Unfortunately, the converse is not true.

Preliminary tests suggested that the first coefficients were more meaningful; therefore we experimented with different distances on two complex vectors $(a_1, \dots, a_k), (a'_1, \dots, a'_k), k \in \{1, \dots, n\}$, one treating all coefficients equally, two which give decreasing value to coefficients of increasing indices. The chosen metrics are the following:

$$d_1 = \sum_{j=1}^k |a_j - a'_j|, \quad d_2 = \sum_{j=1}^k \frac{|a_j - a'_j|}{j}, \quad d_3 = \sum_{j=1}^k |a_j - a'_j|^{1/j}.$$

Algorithms and Computational Analysis. The algorithms below resume the principal steps of our scheme. F in Algorithm 1 (line 2) and d in Algorithm 3 (line 4) correspond, respectively, to one of the transformations R, S, T and one of the metrics d_1, d_2, d_3 previously defined.

Algorithm 1: ComplexLists

Input: List A of proper points of a persistence diagram D ,

$M = \max_{|A|} \{A : A \in \text{database } Db\}$

Output: List B of complex numbers associated with D

1: for each $(u, v) \in A$	4: if $ B < M$
2: replace (u, v) by $F(u, v)$	5: append $M - B $ zeros to B
3: end for	6: end if

Algorithm 2: ComplexVectors

Input: $M, B = \text{list}(z_1, \dots, z_M)$ associated with $D, k \in [0, M]$

Output: Complex vector V_k associated with D

```

1: set  $V_k = \text{list}()$ 
2: for  $j \in \{1, \dots, k\}$ 
3:   compute  $c_j(z_1, \dots, z_M) = \sum_{1 \leq i_1 < i_2 < \dots < i_j \leq M} z_{i_1} \cdot z_{i_2} \cdot \dots \cdot z_{i_j}$ 
4:   append  $c_j$  to  $V_k$ 
5: end for

```

Algorithm 3: VectorsComparison

Input: $L = \{V_k : V_k \text{ complex vector associated with } D \text{ for each } D \in Db\}$

Output: Matrix of distances $d(V_k, V'_k)$

```

1: set  $M = (0_{ij}), i, j = 1, \dots, |L|$ 
2: for each  $i \in \{1, \dots, |L|\}$ 
3:   for each  $j \in \{i, \dots, |L|\}$ 
4:     replace  $0_{ij}, 0_{ji}$  by  $d(i, j)$ 
5:   end for
6: end for

```

Let $N = |L| = |Db|$. It is easily seen that the computational complexities of Algorithms 1 and 3 are $C_1 = O(M \cdot N)$ and $C_3 = O(k \cdot N^2)$, respectively. The cost of Algorithm 2 depends on how we have implemented the computation of Viète formulas. Using the induction on the index j , we have $C_2 = O((2k^2 + k \cdot M) \cdot N)$.

We want to show that our approach to database classification, in general, results to be cheaper than using the bottleneck distance by a suitable choice of the number k of computed coefficients. Our comparison is realized in terms of storage locations and not in terms of running time performances since the algorithm proposed here and the one based on the bottleneck distance run on different platforms. We recall that the cost of computing the bottleneck distance $d_B(D, D')$ is $C_B = O((r + r')^{3/2} \log(r + r'))$ if A, A' are the subsets of proper points of two persistence diagrams D, D' with $|A| = r, |A'| = r'$ (see [14]). Instead, using our scheme, with $N = 2$ and $M = \max(r, r')$, we get $O((\max(r, r') + 2k^2 + k \cdot \max(r, r') + 2k))$, so $C = O(k \cdot (\max(r, r') + k))$. Since $k \leq \max(r, r')$, in the worst case, we have $C = O((\max(r, r'))^2)$ which is higher than C_B , but for pre-processing we may choose a favorable k (e.g. $k = \lfloor \sqrt{\max(r, r')} \rfloor$). Also consider that, for a retrieval task, the heavy part of the computation (Viète’s formulas) for the database is performed offline; in other words: if we store the coefficient vectors instead of the proper points lists, then the search can be performed by a distance computation of complexity $O(\max(r, r'))!$

4 Experimental Results

This section is devoted to validate the theoretical framework introduced in Section 3. In particular, through some experiments on persistence diagrams for 0th homology degree associated with 3D-models represented by triangle meshes,

we will prove that our approach allows to perform the persistence diagrams comparison without greatly affecting (and in some cases improving) the goodness of results in terms of database classification.

To test the proposed framework we considered a database of 228 3D-surface mesh models introduced in [2]. The database is divided into 12 classes, each containing 19 elements obtained as follows: A null model taken from the Non Rigid World Benchmark [3] is considered together with six non-rigid transformations applied to it at three different strength levels. An example of the transformations and their greatest strength levels is given in Figure 3.

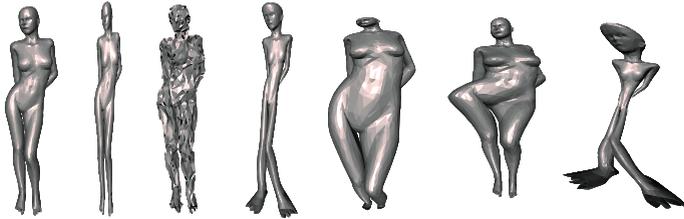


Fig. 3. The null model “Victoria0” and the 3rd strength level for each deformation.

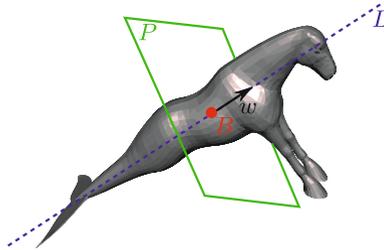


Fig. 4. The null model “seahorse0” depicted with its center of mass B and the associated vector w , which define the filtering functions f_L and f_P .

Two filtering functions f_L, f_P have been defined on the models of the database as follows: For each triangle mesh M of vertices $\{v_1, \dots, v_n\}$, the center of mass B is computed, and the model is normalized to be contained in a unit sphere. Further, a vector w is defined as

$$w = \frac{\sum_{i=1}^n (v_i - B) \|v_i - B\|}{\sum_{i=1}^n \|v_i - B\|^2}.$$

The function f_L is the distance from the line L parallel to w and passing through B , while the function f_P is the distance from the plane P orthogonal to w and passing through B (see, as an example, Figure 4). The values of f_L and f_P are then normalized so that they range in the interval $[0, 1]$. These filtering functions

are translation and rotation invariant, as well as scale invariant because of a priori normalization of the models. Moreover, the considered models are sufficiently generic (no point-symmetries occur etc...) to ensure that the vector w is well-defined over the whole database, as well as its orientation stability.

Our experimental results are synthesized in Tables 1 and 2 in terms of precision/recall (PR) graphs when the filtering functions f_L, f_P , respectively, are considered. Before going into details, we want to emphasize that our intent is not to validate the usage of persistence for shape comparison, retrieval or classification. In fact, as a reader coming from the retrieval domain will probably note, the PR graphs reported in this paper are below the state of the art. This depends on the fact that, in general, good retrieval performances can be achieved only taking into account different filtering functions that give rise to a battery of descriptors associated with each model in the database.

Table 1. PR graphs related to the filtering function f_L , when $0th$ -persistence diagrams are compared directly through the bottleneck distance and in terms of the first k components of the complex vectors obtained from the transformations R (first row), S (second row) and T (third row) through the distances d_1 (first column), d_2 (second column) and d_3 (third column).

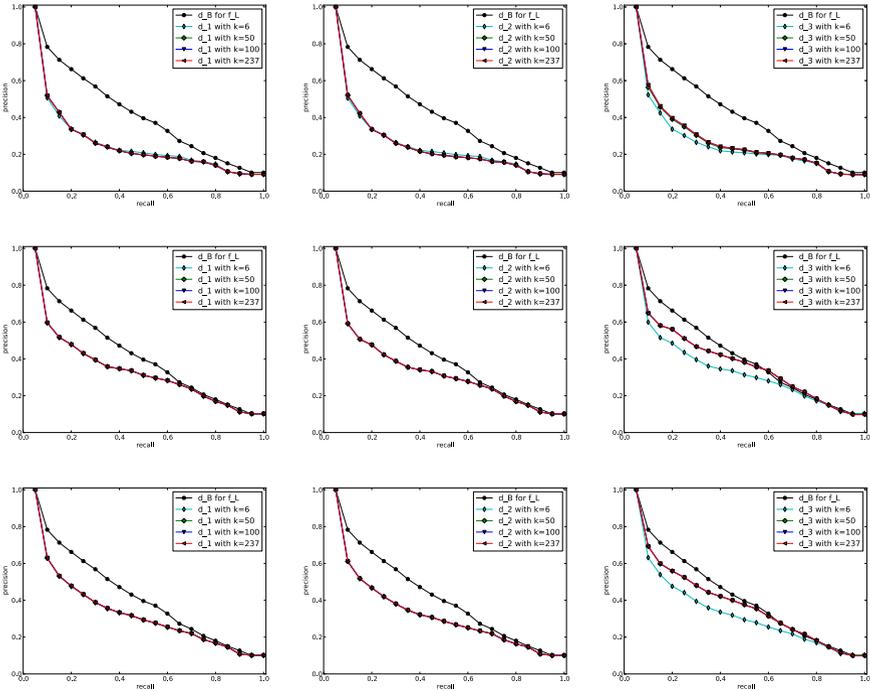
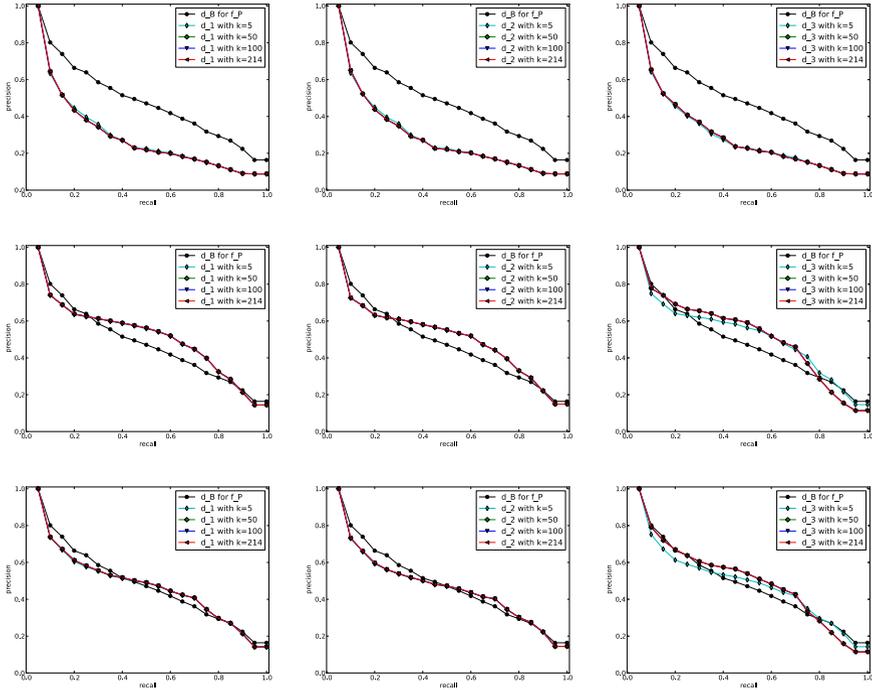


Table 2. PR graphs related to the filtering function f_P , when $0th$ -persistence diagrams are compared directly through the bottleneck distance and in terms of the first k components of the complex vectors obtained from the transformations R (first row), S (second row) and T (third row) through the distances d_1 (first column), d_2 (second column) and d_3 (third column).



What these plots aim to show is the comparison of the performances when the database classification is carried out through the computation of the bottleneck distance d_B between persistence diagrams or the computation of the distances d_1 , d_2 and d_3 between the first k components of the complex vectors obtained through the transformations R , S and T , for different values of k (see Section 3 for the definitions of R , S , T , d_1 , d_2 and d_3). As it can be easily observed, increasing the value of k from the smallest to the biggest number of proper points in the persistence diagrams of our database, the PR graphs do not change so sensibly. This means that the most important information of the persistence diagram is contained in the first few vector components, the ones corresponding to the coefficients of monomials with highest degree. Moreover, we point out also that PR graphs related to vectors which are induced by transformations warping the diagonal Δ to a point (second and third rows in Tables 1 and 2) provide better results than by acting as the identity (first row). This fact depends on the properties of polynomial coefficients: Indeed roots corresponding

to points of persistence diagrams farther from the diagonal weigh more than those closer to it. Hence, applying transformations S and T corresponds, in some sense, to providing points of a persistence diagram with a weight that follows the paradigm of persistence: The longer the lifespan of a homological class, the higher the weight associated with the point having as coordinates the birth and death dates of this class. This outcome is moreover strengthened by the usage of the distance d_3 (third column in Tables 1 and 2) since it greatly enhances the contribution of the first vector components to their dissimilarity measure at the expense of the last.

Finally, note that the precision values at high recall — i.e. by retrieving a large number of objects — are always fairly comparable with the values relative to the bottleneck distance. This assures us that complex vector comparison can act as a fast and reliable preprocessing scheme for reducing the set of objects to be fed to the generally more precise bottleneck distance.

Acknowledgments. We wish to thank Claudia Landi and Paweł Dłotko for very useful discussions.

The first author dedicates this work to a great fighter (Forza Vale!!!).

References

1. Biasotti, S., De Floriani, L., Falcidieno, B., Frosini, P., Giorgi, D., Landi, C., Papaleo, L., Spagnuolo, M.: Describing shapes by geometrical-topological properties of real functions. *ACM Comput. Surv.* **40**(4), 1–87 (2008)
2. Biasotti, S., Cerri, A., Frosini, P., Giorgi, D.: A new algorithm for computing the 2-dimensional matching distance between size functions. *Pattern Recognition Letters* **32**(14), 1735–1746 (2011)
3. Bronstein, A., Bronstein, M., Kimmel, R.: *Numerical Geometry of Non-Rigid Shapes*, 1st edn. Springer Publishing Company, Incorporated (2008)
4. Cerri, A., Di Fabio, B., Ferri, M., Frosini, P., Landi, C.: Betti numbers in multidimensional persistent homology are stable functions. *Mathematical Methods in the Applied Sciences* **36**(12), 1543–1557 (2013)
5. Cerri, A., Di Fabio, B., Jabłoński, G., Medri, F.: Comparing shapes through multi-scale approximations of the matching distance. *Computer Vision and Image Understanding* **121**, 43–56 (2014)
6. Cohen-Steiner, D., Edelsbrunner, H., Harer, J.: Extending persistence using Poincaré and Lefschetz duality. *Foundations of Computational Mathematics* **9**(1), 79–103 (2009)
7. Cohen-Steiner, D., Edelsbrunner, H., Harer, J.: Stability of persistence diagrams. *Discr. Comput. Geom.* **37**(1), 103–120 (2007)
8. d’Amico, M., Frosini, P., Landi, C.: Using matching distance in size theory: A survey. *Int. J. Imag. Syst. Tech.* **16**(5), 154–161 (2006)
9. d’Amico, M., Frosini, P., Landi, C.: Natural pseudo-distance and optimal matching between reduced size functions. *Acta. Appl. Math.* **109**, 527–554 (2010)
10. Donatini, P., Frosini, P.: Lower bounds for natural pseudodistances via size functions. *Archives of Inequalities and Applications* **1**(2), 1–12 (2004)
11. Donatini, P., Frosini, P.: Natural pseudodistances between closed manifolds. *Forum Mathematicum* **16**(5), 695–715 (2004)

12. Edelsbrunner, H., Harer, J.: *Computational Topology: An Introduction*. American Mathematical Society (2009)
13. Edelsbrunner, H., Harer, J.: Persistent homology—a survey. In: *Surveys on discrete and computational geometry*, *Contemp. Math.*, vol. 453, pp. 257–282. Amer. Math. Soc., Providence (2008)
14. Efrat, A., Itai, A., Katz, M.J.: Geometry helps in bottleneck matching and related problems. *Algorithmica* **31**, 1–28 (2001)
15. Ferri, M., Landi, C.: Representing size functions by complex polynomials. *Proc. Math. Met. in Pattern Recognition* **9**, 16–19 (1999)
16. Frosini, P., Landi, C.: Size functions and formal series. *Appl. Algebra Engrg. Comm. Comput.* **12**(4), 327–349 (2001)
17. Kalisnik Verovsek, S., Carlsson, G.: Symmetric and r-symmetric tropical polynomials and rational functions. arXiv preprint [arXiv:1405.2268](https://arxiv.org/abs/1405.2268) (2014)