FP7-ICT-2011.8
Contract no.: 318493

www.toposys.org

Deliverable D4.3

Progress and Activity Report on WP4

Deliverable Nature:	Report (R)
Dissemination Level: (Confidentiality)	Public (PU)
Contractual Delivery Date:	M36
Actual Delivery Date:	M36
Version:	2.0

Contents

1 Summary

In this deliverable, we outline work which was done for Work Package 4 in the final year of the project. We made progress in a number of different applications as well as improving our understanding of teh computational complexity of computing persistence. Our first application is in the study of epidiomological data. We show how topological data analysis can be used to compare different flu seasons by combining Takens' delay embeddings with persistent homology, in a similar fashion to understanding periodic/recurrent motion (See also Deliverables 1.1 and 1.2). It contains complementary information to standard techniques, capturing different qualatative feature.

The second application describes a new approach to textual data based on Bregman divergences, although it is more general, as it extends to any type of divergence. Despite asymmetry, there is a natural way to build up a simplicial complex based on the balls as a cover. Surprisingly, it turns out that the Nerve Lemma can be applied making persistence computations readily accesible.

Finally, we describe a new runtime analysis of persistence using nested dissection - a technique which was used to study general matrix reductions. In addition, to showing that we can compute persistence faster for complexes with a certain separability property, This suggests that in order to speed up computation, we should try to create geometric complexes with these properties.

This report is based on the following papers and reports:

- J. Pita Costa, P. Skraba, Topological Epidemiological Data Analysis, ACM Digital Health Conference 2015
- H. Edelsbrunner, H. Wagner, Z. Virk, Topological Analysis with Bregman Divergences, in preparation
- M. Kerber, D. Sheehy, P. Skraba, Persistent Homology and Nested Dissection, accepted to SODA 2016

We report that an extended version of the paper "Multiscale Topological Trajectory Classification with Persistent Homology" which was reported on in Deliverable 4.2 has been published in the International Journal of Robotics Research (IJRR). Also not directly reported on is an overview of computing persistence "CAPD:: RedHom v2-Homology Software Based on Reduction Algorithms," was published after being presented at ICMS (International Conference on Mathematical Software). Also noteworthy, is that a chapter on Topology and Big Data was accepted to the book "Big Data Optimization" which will be published by Springer.

Section 6 describes the experiments performed and bottlenecks encountered in analysing text-based data. Ultimately, this work was impeded by two key factors:

- Scalability of complex construction and/or the persistence algorithm (e.g. for the persistence of eigenspaces)
- Uniformative metrics - this led directly to the work on topological data analysis using divergences

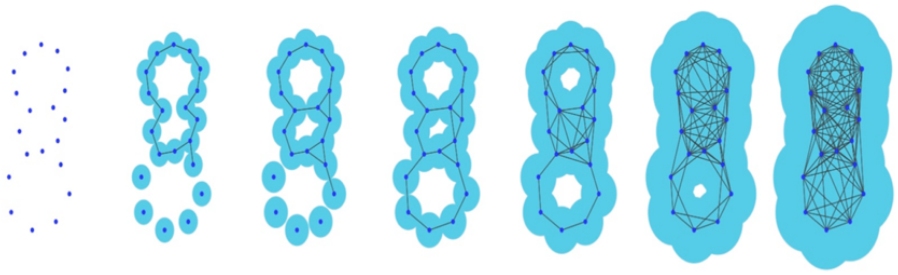

Figure 1: Topological data analysis: the filtration of a simplicial complex of a given point cloud according to the growing radius of balls centered in the input data points.

2 Epidemiology

The first application we report on here is the use of Topological data analysis [TDA] on epidemiological data. The basic technique encodes topological features of a given point cloud by diagrams representing the lifetime of those topological features. Recently, these topological methods on data have seen a relevant application to the study of the influenza virus as described in [34].

The system Influenzanet monitors online the activity of influenza-like-illness [ILI] with the aid of volunteers via the internet. It has been operational for more than 10 years, and at the EU level since 2008. Influenzanet obtains its data directly from the population, contrasting with the traditional system of sentinel networks of mainly primary care physicians. Influenzanet is a fast and flexible monitoring system whose uniformity allows for direct comparison of ILI rates between countries [37].

Our goal with this project is to analyze the Influenzanet data using persistence, identifying topological features relevant to the epidemiological study. To do so, we identify data noise, distinguish higher dimension features and look at the overall structure of the disease as well as its evolution during the flu season in Portugal and Italy. In particular, this provides a way to test agreement at a global scale arising from standard local models.

2.1 Epidemiological Data

The Mahalanobis distance is a measure of the distance between a point P and a distribution D, widely used in cluster analysis and classification techniques. When considering this metric on the space while using TDA, we get a perspective of that space under different scales, where small features will eventually disappear. We have used in [40] several techniques to preprocess the input data, including subsampling and colliding data points that are closer than a given parameter. In particular, we embed the data in higher dimensions, compute persistence, and look for outliers.

The analyzed data lists the number of active participants and the number of ILI onsets, for three different ILI case definitions of the Influenzanet in Italy for every week in years of the Influenza seasons from 2010/11 to 2012/13. Based on this data we have used several algorithms to preprocess it, prior the construction of the Vietoris-Rips complex that corresponds to the given data. This method permits us to encode the qualitative features of that data into a persistence diagram.

The images in Figure 3 show the cloud of input data points, the corresponding simplicial com-

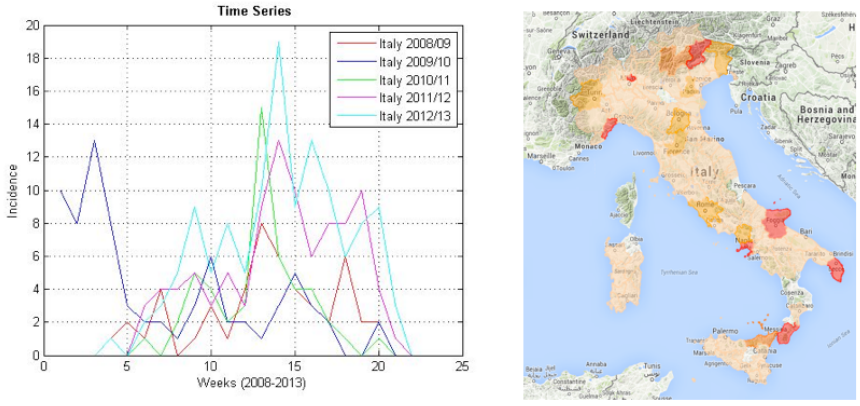

Figure 2: Influenzanet: the time-series for the incidence of influenza in Italy during the flu seasons of 2008-2013 (on the left); a screenshot of the influenzanet system in Italy, taken in May 2015 (on the right).

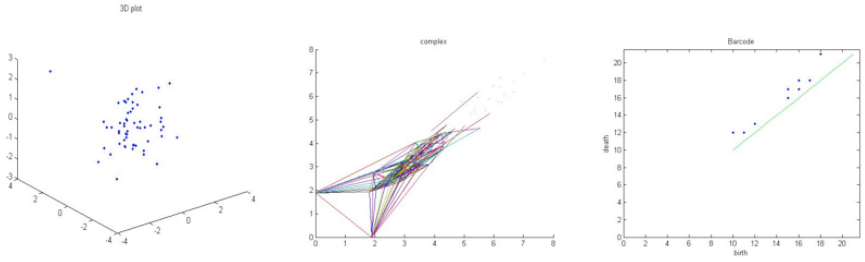

Figure 3: The pipeline for the computation of topological data analysis for the time series of Italy 2009/10: the given pointcloud of the input data (on the left); the Viatoris-Rips complex approximating the space of the pointcloud (in the center); the correspondent persistence diagram encoding the lifetime of the persistent topological features (on the right).

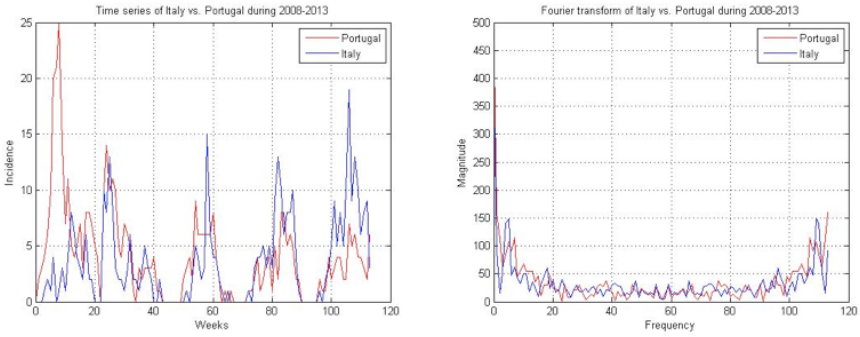

Figure 4: The filtration of the simplicial complex at several levels varying according a parameter r for the input time series of Italy in the flu season of 2009/2010: r = 2 (on the left); r = 3 (in the center); r = 5 (on the right).

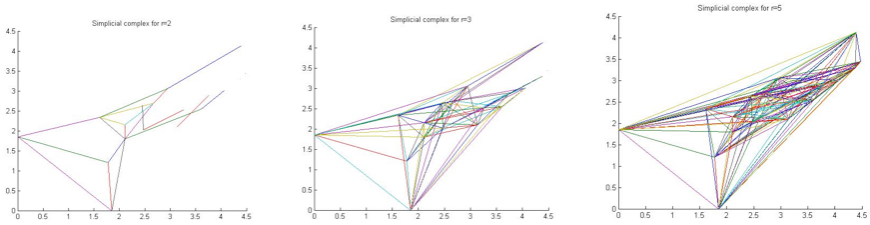

Figure 5: Comparing the flu seasons of Portugal and Italy during 2008-2013: the time series (on the left); the Fourier transform (on the right).

plex, and persistence diagram for dimension 1. These topological tools complement the information obtained by classical data analysis. The computation of the persistence diagrams is done via Vietoris-Rips complexes using Perseus, the open source persistent homology software [Na14]. The input structure is given as a symmetric distance matrix where the entries come from pairwise distances between points in a given point cloud. In the figures below we can see three steps of the construction of the Vietoris-Rips complex that will provide us with the persistence diagram encoding the topological information of the Influenzanet data.

2.2 Quantative and Qualitative Analysis of Influenza

Fourier analysis is widely used to identify patterns in a time series. We used the time series of the incidence of influenza in Portugal and Italy for the flu seasons of 2008-2013. In Figure 4 we can see the plot of the two time series and their correspondent Fourier transform.

We computed in [41] the Fourier transform for each pair of time series (country, year) to compare the flu seasons of Portugal and Italy. In that work we compared the quantitative methods of Fourier

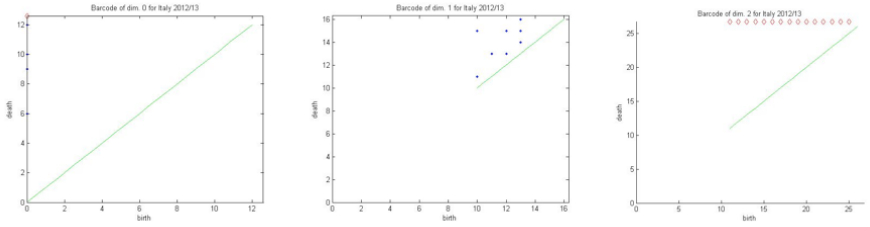

Figure 6: The persistence diagrams for the input time series of Italy in the flu season of 2009/2010: dimension 0 (on the left); dimension 1 (in the center); dimension 2 (on the right). The red circles mean that the lifetime of the considered features does not end.

analysis with the qualitative methods of TDA. In Figure 7 the reader can see an extension of the results of this comparison with highlighted biggest and smallest values.

When comparing two time series that may vary in time or speed it is usual to apply the algorithm dynamic time warping [DTW] measuring the similarity between those temporal sequences. In this study we compared each pair of time series (country, year) obtaining the respective measure that can be seen in the table of Figure 7.

The usage of TDA for the analysis of time series was explored in [38] towards the quantification of periodicity and identification of periodic signals in gene expression in [39]. We also use TDA to analyze the input time series data, following an approach developed specifically for Influenza. Barcodes and correspondent persistence diagrams seen as multi-scale signatures encode the lifetime of topological features within pairs of numbers representing birth and death times. We have computed a persistence diagram for each time series (country, year) embedded in higher dimensions. As shown by the persistence diagrams below, the distinguishable features are seen in dimension 1.

Persistence landscapes are techniques of TDA that permit us to measure the pairwise distance between persistence diagrams at several different levels. The distance value between these two persistence diagrams in the tables of Figure 7 was calculated using the persistence landscapes toolbox [Dl14] to compute the distance between diagrams considering different norms. The following tables represent the comparison between the Fourier analysis, dynamical time warping and topological analysis of the incidence of influenza in Italy and Portugal for the flu seasons of 2008-2013.

When comparing the distances obtained by Fourier analysis, DTW and TDA we can see that these three methods look at different features of the data.

The plots in Figure 8 represent time-series for selected flu seasons from 2008 to 2013. They serve us to compare the different data analysis methods used in this study.

When comparing the distances between Italy 2011/12 and Portugal 2009/08, the Fourier provides us with a high value of 92, while DTW analysis has a low value of 0,86667. On the other hand, for the flu seasons of Italy 2008/09 and Portugal 2010/11, the DTW has a low value of 20 while the Fourier analysis has a relatively high value of 2,0147. In the first case the monotony of the curves match, although the periodicity not being close. The second case shows two high peaks for Italy 2008/09 against one for Portugal 2010/11 explaining the low level of DTW.

To compare the quantitative Fourier analysis with the qualitative analysis of TDA we look at the flu seasons of Italy 2010/11 and Portugal 2008/09 where TDA achieved the low value of 0.288675

Fourier		Portugal				
		2008	2009	2010	2011	2012
Italy	2008	**2,2518**	1,523	2,0147	1,1977	0,95957
	2009	1,523	1,0536	1,3576	0,86667	0,74338
	2010	2,0147	1,3576	1,1635	0,70203	0,67165
	2011	1,1977	0,86667	0,70203	0,67071	0,6352
	2012	0,95957	0,74338	0,67165	0,6352	**0,61559**

DTW		Portugal				
		2008	2009	2010	2011	2012
Italy	2008	89	80	20	15	15
	2009	106	32	58	60	58
	2010	75	88	**13**	23	23
	2011	60	92	16	28	35
	2012	44	**111**	35	48	55

TDA		Portugal				
		2008	2009	2010	2011	2012
Italy	2008	2,91548	2,85774	2,25462	2,81366	2,51661
	2009	2,32737	2,27303	1,73205	1,63299	1,75594
	2010	**0,288675**	0,408248	1,22474	1,32288	0,957427
	2011	0,957427	1	1,35401	1,52753	1,32288
	2012	4,09268	4,04145	3,37886	3,7305	**3,58236**

Figure 7: Comparing the flu seasons of Portugal and Italy during 2008-2013: the distance tables for the Fourier analysis (on the top), the dynamic time warping (on the center), and the topological data analysis (on the bottom).

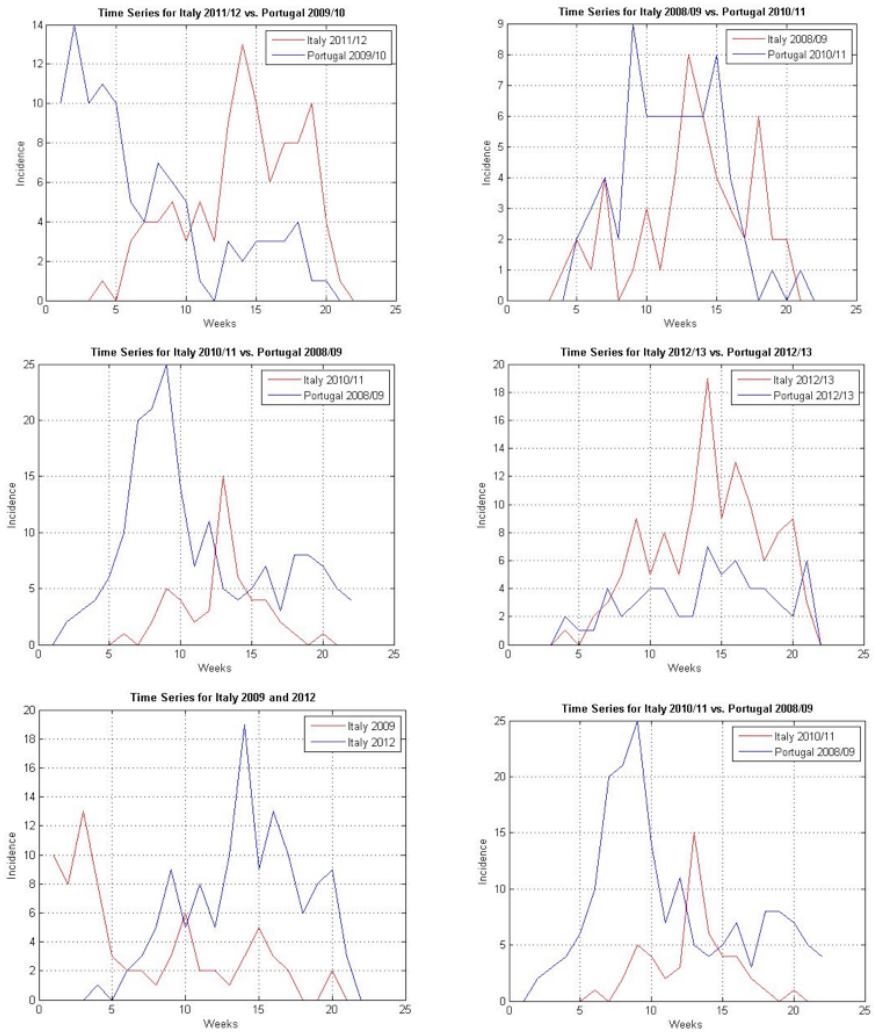

Figure 8: Comparing the flu seasons of Portugal and Italy during 2008-2013: selected plots of time-series to compare the results in the Fourier analysis, the topological data analysis and the dynamical time warping.

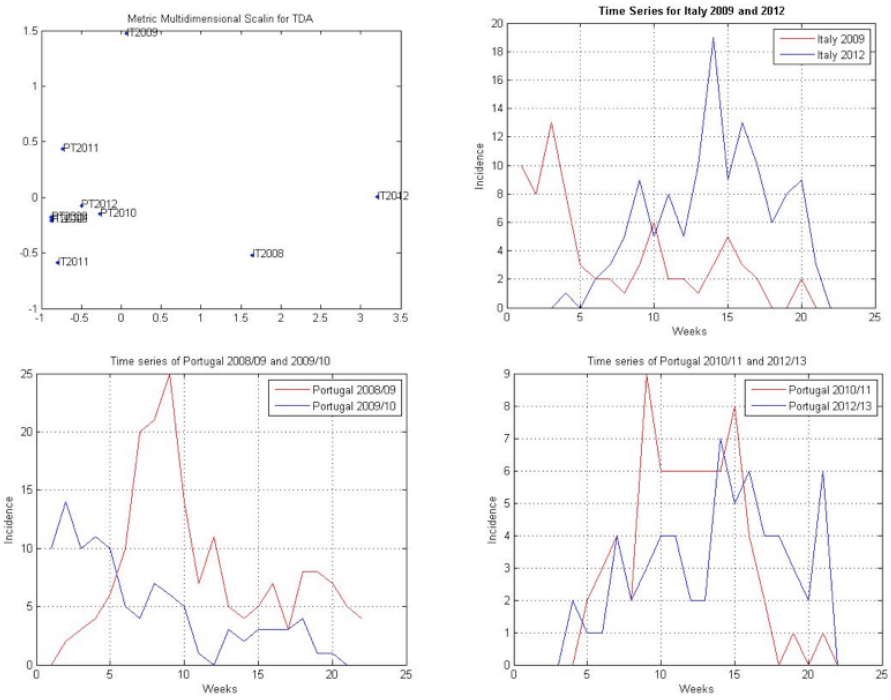

Figure 9: Comparing the flu seasons of Italy and Portugal during 2008-2013 using metric multidimensional scalling (on the upper left) to identify: the outlier flu seasons of Italy 2009/10 and 2012/13, with time series ploted for analysis and interpretation (on the upper right); the close flu seasons of Portugal 2008/09 and 2009/10 (on the lower left); and the flu seasons of Portugal 2010/11 and 2012/13, close to the diagonal (on the lower right).

and the Fourier analysis reached the high value of 2,0147. On the other hand, the flu seasons of Italy and Portugal in 2012/13 reach a high TDA value of 3,58236 and a low Fourier value of 0,61559. The first case shows a big difference of peaks which does not happen in the second case where the periodicity is lower, implying the lower level for the Fourier analysis.

Finally, the comparison between TDA and DTW points us to the flu seasons Italy 2008/09 and Portugal 2012/13, where TDA reached a high value of 2,51661 (due to the higher similarity of peaks) and DTW reached a low value of 15 (describing the different behavior of the curves); and the flu seasons of Italy 2010/11 and Portugal 2008/09, where TDA achieved a low level of 0.288675 (with great difference of peaks as pointed out earlier) and DTW achieved the high value 75 (pointing out the similar behavior of the curves).

We used multidimensional scaling as in Figure 9 to identify outliers for each of the three methods within the flu seasons analyzed in this study. TDA provides a qualitative analysis of the time series of the incidence of influenza, looking in particular at the peaks and dramatic changes. In that perspective, the time series of Italy 2009/10 and 2012/13 plotted in Figure 9 describe very different flu seasons with very different peaks. On the other hand, the flu seasons of Portugal 2008/09 and

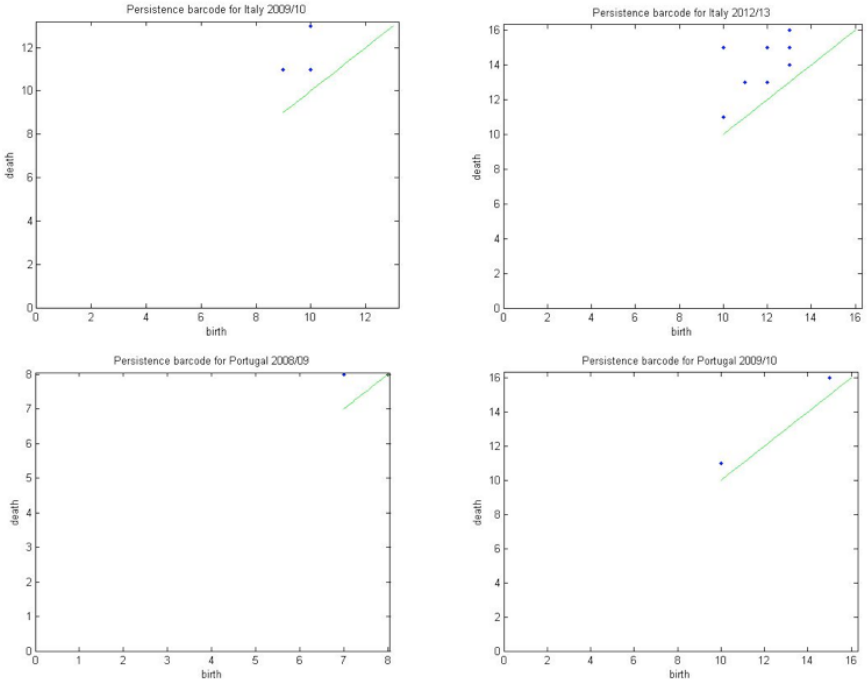

Figure 10: Comparing the flu seasons using persistence diagrams for dimension 1 for: Italy 2009/10 (on the upper left), Italy 2012/13 (on the upper right), Portugal 2008/09 (on the lower left), and Portugal 2009/10 (on the lower right), identified as particular cases in Figure 9.

2009/10 are identified being very close with very similar peaks, although the behavior of the curve being different. The knowledge on secondary attack rates in the influenza season is of importance to access the severity of the seasonal epidemics of the virus, estimated recently with information extracted from social media in [42]. Here lies a strong point of TDA where it can provide relevant contribution complementing other methods.

The persistence diagrams of Figure 10, correspondent to the identified flu seasons of Italy 2009/10 and Italy 2012/13, and Portugal 2008/09 and 2009/10. They encode the lifetimes of the topological features of the curves of the time series of those seasons. Persistence diagrams are a clear and practical tool that allows us the detection of outliers and to capture the qualitative features of the dynamics of the system. These ideas provide a new approach to the analysis of the seasons in the epidemiology of Influenza.

2.3 Discussion

The study of epidemiology is a rich source of problems relating to nonlinear systems, large scale data and development of more accurate models, where TDA can con- tribute, providing high dimension techniques for medical data analysis. In this study we showed how they can be used to analyze the

incidence for different ILI case definitions, contributing to a better understanding of the features distinguished by those definitions. The information provided by quantitative methods such as DTW or the Fourier analysis of time series can be complemented by the topological analysis of that data. The examples considered in Figure 8 show that these methods do not express the same information about the development of the epidemics during the flu season. The knowledge provided by each of these methods complements the knowledge coming from the other methods and can be put together in a global information map. Further research considers the analysis of the impact of the qualitative aspects of TDA for modeling and prediction of the current Influenza season. We will also use state of the art artificial intelligence methods to learn metrics more appropriate to the input time series data aiming, to grasp a better understanding of the severity of the epidemics both in past seasons and during the ongoing season.

3 Topological Analysis of Text-based Data

Originally, we had planned to peform topological data analysis on social media. There are two possible forms of such data: graph-based and text-based. Of the available graph-based data sets, e.g. Twitter, the graphs were

1. highly sparse – this makes it unlikely to find any higher topological structures beyond components, which has already been extensively studied (e.g. the components exhibit a power law type of behaviour)

2. directed – the graphs are inherently directed, e.g. user a follows user b. Though directed topology is a a field of study, there is no currently accepted notion of directed persistence which does not in some way symmetrize the graph (though it is an active area of research)

From the perspective of TOPOSYS, the sparsity of the components proved the larger problem, as it was not possible to find any interesting topological structure.

The other data set considered was ArXiV, which is now available in text form (after being cleaned), on the project website. Here the problem encountered was similar to the Twitter examples. The components were small and/or highly dense (i.e. they formed large cliques). This presented the problem of again, no interesting topological structures. Basing connections based on the abstract text, which is also available presented simliar problems which are described below.

Text-based social media was the other natural candidate. As a proxy for news articles and blogs, we investigated Wikipedia in multiple languages. Wikipedia is quite large, containing many documents. The data was first cleaned, with stubs and stop words removed and a *vector-space representation* was used. Initially, we conducted an investigation of the metric space formed by using the standard Euclidean distance as well as cosine distance between two documents. These are generally the standard metrics used for text analysis. In Figure 11 (left), we see the distance from a document to all the other documents.

This is a fairly consistent picture regardless of the document and language chosen. First, we note that the distances are in a relatively narrow interval, with most document being nearly orthogonal. This is to be expected since the stop words are removed, many documents do not share any common words. However, the number of documents rises very quickly with distance. This, in combination with the dimensionality of extrinsic space (the dimension is in the tens of thousands), means we must limit ourselves to a small distances, where we do not see any structure (similar as in the graph based examples). In Figure 11(right), we see than most documents are nearly equidistant between two landmarks, making it difficult to use subsampling techiques such as Witness complexes – these would also form a clique almost immmediately.

Furthermore, the dimension means we are limited to using the Vietoris-Rips complex which not only add a multiplicative error of 2 (which effectively is larger than any signal we hope to find), but also presents a computational challenge, in that the complex construction (rather than the actual persistence computation) – a small number of points can generate a very large complex. While approximations exist, this would further degrade the signal. In addition, the high dimensionality of the data set implies that constants in the approximations would be very large. Indeed, current implementations of the approximation schemes work well for dimensions less than 50 (since they are based on kd-trees).

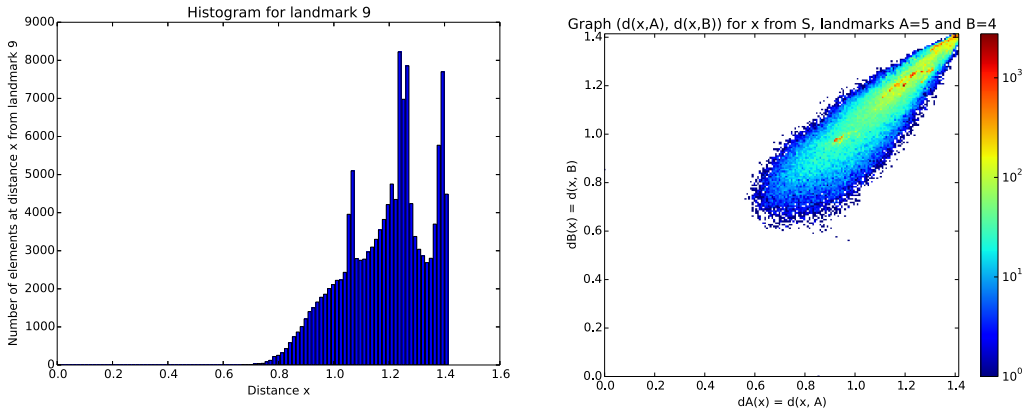

Figure 11: On the left, we have the histogram distances from a randomly chosen document. The number of documents grows quickly with distance limiting how much of the persistence diagram we can compute. On the right, we have a histogram of distances to two landmarks and we see that most documents are nearly equidistant from both. This implies that constructions such as the Witness complex would form cliques almost immediately.

We also chose to test local persistence, to try and understand the metric structure around these points. Considering neighborhoods of a few hundred documents, we found only 0-dimensional persistence diagrams were non-trivial. The higher dimensional persistence diagrams were ultimately empty. This was a consequence of using the Rips complex - all higher cycles where formed by cliques. The 0-dimensional diagrams capture only distances in the minimum spanning tree of the neighborhood.

This ultimately points torwards the conclusion that the standard metrics are not particularly informative. This directly led to the work described in Section 4. This approach has the advantage of being more general as well as gives a more natural interpretation of higher dimensional simplices than either Euclidean distance or cosine distance.

One interesting intermediate result of this work was the comparison of average neighborhoods from different languages of Wikipedia. In Figure12, we show the average persistence landscapes for three different languages, which are clearly different. One ongoing direction of inquiry is can this be used to improve cross-lingual mappings between documents (i.e. finding the closest article from a corpus in a different language).

In addition to only using the metric, we also considered using the link information when available, such as in the case of Wikipedia. One idea to exploit this additional information which was pursued, was to develop a notion of a topological PageRank type of algorithm - based on the persistence of a map. In principle, we can view the underlying space of articles (for example in ArXiV or Wikipedia) as a metric space and the links as a map from the metric space to itself. That is, say that document a links to documents b, c and d. This gives the set map

$$f : \{a\} \mapsto \{b, c, d\}$$

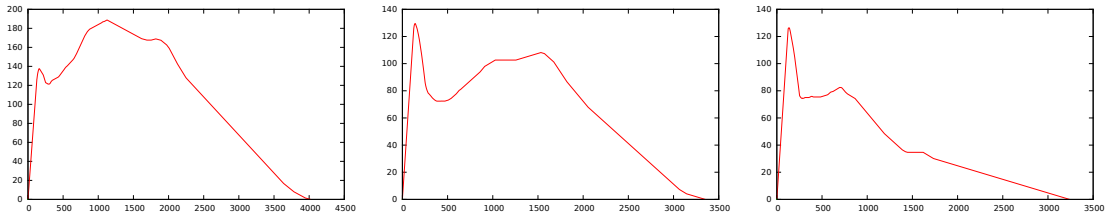

Figure 12: The average 0-dimensional persistence landscape of normalized neighborhoodds of size 200 for spanish wikipedia (left), hindi wikipedia (middle) and portuguese wikipedia (right). This shows that the metric structure varies and is different in different languages.

The persistence of the eigenspace can then be computed. The key problem with this approach was scalability. As described in the report on WP1, the original algorithm does not scale beyond a few hundred points. Therefore, effort was directed at building better representation of maps as well as a more efficient algorithm. Progress was made in both areas, but the work was not concluded and is ongoing beyond the project.

4 Bregman Divergence for Text Data

In our experiments using textual data, we found that Euclidean data is not as informative as we would like, particularly when considering higher order simplices. Here we present an outline of preliminary work, which was done to find a more infomative metric between documents using ideas from information geometry.

Following the lead of term-vectors in turning a corpus into a populated geometric space, we extend the cosine similarity measure from pairs to tuplets. Being context-sensitive, this extension lends itself to representing the corpus as a filtration of simplicial complexes and analyzing it with advanced topological tools, including persistent homology.

4.1 Geometry

Duality transform. First we introduce some required concepts.

Duality transforms are a key technique in computational geometry. We later use it in the context of Bregman distances.

The construction is based on the duality between points in $\mathbb{R}^{n-1} \times \mathbb{R}$ and affine functions $\mathbb{R}^{n-1} \to \mathbb{R}$. The particular transform we use maps $A = (a, \alpha)$ to $A^*(x) = \langle a, x \rangle - \alpha$, and it maps A^* back to $(A^*)^* = A$. Given a second point $Q = (q, \psi)$, the transform preserves the difference between the values:

$$A^*(q) - \psi = Q^*(a) - \alpha. \tag{1}$$

Indeed, both sides of the equation evaluate to $\langle a, q \rangle - \alpha - \psi$.

Voronoi diagram. Imagine we simultaneously grow balls centered at the points in $X \subseteq \mathbb{R}^{n-1}$, stopping the growth at the places where and when the balls meet. The eventual extent of the ball growing from x is called the *Voronoi domain* of x,

$$\mathbb{V}(x) = \{a \in \mathbb{R}^{n-1} \mid D(x, a) \leqslant D(y, a), \forall y \in X\}. \tag{2}$$

The collection of Voronoi domains is the *Voronoi diagram* of X, denoted as $\mathrm{Vor}(X)$.

Delaunay triangulation. Similar to $\Delta(X)$, the *Delaunay triangulation* is a simplicial complex over X, namely the collection of all simplices whose Voronoi domains have a non-empty common intersection:

$$\mathrm{Del}(X) = \{\xi \subseteq X \mid \bigcap_{x \in \xi} \mathbb{V}(x) \neq \varnothing\}. \tag{3}$$

4.2 Combinatorial Topology

The development of the mathematics is guided by the wish to apply tools from algebraic topology to analyze point cloud data. In this section, we introduce the relevant background.

Simplicial complexes. With a few exceptions, the geometric aspects of simplicial complexes are not important in this paper. We therefore stress the combinatorial view, in which we talk about a set system over a finite set U. A *vertex* is an (abstract) element of U, a *simplex* is a non-empty collection of vertices, and a *simplicial complex* is a system of simplices that is closed under taking

subcollections. Following common choices for notation, we let K be a simplicial complex and σ a simplex in K. The *dimension* of σ is $\dim \sigma = \operatorname{card} \sigma - 1$. A *face* of σ is a subcollection of σ, and the above condition requires that with σ, all its faces must belong to K. A *subcomplex* is a system $K' \subseteq K$ that itself is a simplicial complex. An example is $\Delta = 2^U - \{\emptyset\}$, the system of all non-empty simplices over U. Letting m be the number of vertices, Δ contains $2^m - 1$ simplices. In practically important situations, m is large and 2^m is too large. Indeed, some of the developments in this paper are motivated by this challenge.

Nerve. Another common construction of a simplicial complex starts with U being a finite collection of subsets of $\mathbb{R}^n$. The *nerve* of U is then the system of simplices with non-empty common intersection:

$$\operatorname{Nerve}(U) = \{\emptyset \neq \sigma \subseteq U \mid \bigcap \sigma \neq \emptyset\}. \tag{4}$$

It is clear that the nerve is a simplicial complex, no matter what the sets are.

The following lemma plays a crucial role.

Result 4.1 (Nerve lemma)**.** *If the intersection of any sub-collection of U is* contractible*,* $\operatorname{Nerve}(U)$ *has the homotopy type of the union of U.*

Note that Euclidean balls and their intersections are convex, hence contractible. So the nerve captures the topology of Euclidean balls.

Our interest is in the union of balls based of Bregman distances. These are not Euclidean balls, and might not be convex. Still, the lemma states that we capture the topology of the union of these balls, if they intersect contractibly.

The connection to geometry is furnished by mapping every vertex to a point in $\mathbb{R}^n$ and every (abstract) simplex to the corresponding *geometric simplex*, which is the convex hull of the points that are the images of the vertices. Assuming the intersection of any two geometric simplices is either empty of a common face, we call the image of the map a *geometric realization* of the complex. Geometrical realizations always exist provided $n \geqslant 2k + 1$, in which k is the maximum dimension of any simplex in the complex. The *underlying space* of a geometric realization is the union of the geometric simplices, together with the Euclidean topology inherited from $\mathbb{R}^n$.

Čech complex. Let X be a finite set of points in $\mathbb{R}^{n-1}$, and recall that $\mathbb{B}_r(x)$ is the set of points a with distance at most r from x.

We write $\Delta(X)$ for the collection of non-empty subsets of X, referring to the sets as simplices. The *Čech complex* for radius r is the subcollection of simplices whose balls of radius r have a non-empty common intersection:

$$\operatorname{Cech}_r(X) = \{\xi \subseteq X \mid \bigcap_{x \in \xi} \mathbb{B}_r(x) \neq \emptyset\}. \tag{5}$$

For example, $\operatorname{Cech}_r(X) = \emptyset$ for all $r < 0$, and $\operatorname{Cech}_r(X) = \Delta(X)$ for all sufficiently large r. Clearly, $\operatorname{Cech}_r(X) \subseteq \operatorname{Cech}_{r'}(X)$ whenever $r \leqslant r'$.

A major drawback of Čech complexes is the exponential number of simplices they contain for large radii. To cope, we adapt the concepts of Voronoi diagram and Delaunay triangulation.

Rips complex. The *Rips complex* is a simplification of the Čech construction. Instead of checking all the intersections of balls, we only check pairwise intersections. Whenever two balls intersect, we put an edge; when all the edges of a simplex are present, all its faces are also inserted.

[[Any concise definition of Rips without relying on the diameter, which in our case is not equivalent?]]

Clearly, $\mathrm{Rips}_r(X) \subseteq \mathrm{Cech}_r(X)$ for every r.

A major drawback of Čech and Rips complexes is the exponential number of simplices they contain for large radii. To cope, we adapt the concepts of Voronoi diagram and Delaunay triangulation.

Delaunay complex. Clipping the balls within their respective Voronoi domains, we get nerves that are contained in the Čech complexes for the same radii.

As before, we let X be a finite set of points on $\mathbb{R}^{n-1}$. The *Delaunay complex* for radius r consists of all simplices in the Delaunay triangulation for which the corresponding clipped balls have a non-empty common intersection:

$$\mathrm{Del}_r(X) = \{\xi \subseteq X \mid \bigcap_{x\in\xi} [\mathbb{B}_r(x) \cap \mathbb{V}(x)] \neq \emptyset\}. \tag{6}$$

It is clear that $\mathrm{Del}_r(X) \subseteq \mathrm{Cech}_r(X)$ as well as $\mathrm{Del}_r(X) \subseteq \mathrm{Del}(X)$. A common mistake is to think that $\mathrm{Del}_r(X)$ is the intersection of these two complexes, but this is not necessarily the case.

Filtrations. The above constructions are combinatorial representations of the union of balls of radius r, centered around the input points. These simplicial complexes aim at capturing the topology, or precisely homotopy type, of the union of balls. Varying r from 0 to ∞, gives rise to a filtration, namely a nested sequence of complexes:

$$K_0 \subseteq \ K_1 \subseteq \cdots \subseteq K_n$$

Instead of looking at topology of a simplex complex, we can study the evolution of topology along the filtration. This leads to the concept of *persistence*.

Discrete Morse functions. For our purposes, a slightly more generalized version of the original framework by Forman [50] is appropriate; see also [43].

Let K be a simplicial complex. The face relation defines a canonical partial order on K, which we denote by writing $\sigma \leqslant \tau$ whenever $\sigma \subseteq \tau$. The *Hasse diagram* is the transitive reduction of this partial order; it is the directed graph whose nodes are the simplices and whose arcs are the pairs (σ, τ) in which $\sigma \leqslant \tau$ and $\dim \sigma = \dim \tau - 1$. Given simplices $\sigma \leqslant \upsilon$ in K, we call $[\sigma, \upsilon] = \{\tau \mid \sigma \leqslant \tau \leqslant \upsilon\}$ an *interval*, with *lower bound* σ and *upper bound* υ. A *generalized discrete vector field* is a partition V of K into intervals. The special case in which every interval is either a singleton or a pair is what Forman calls a *discrete vector field*. Suppose now that there is a function $f\colon K \to \mathbb{R}$ that satisfies $f(\sigma) \leqslant f(\tau)$ whenever $\sigma \leqslant \tau$, with equality iff σ and τ belong to a common interval in V. Then f is called a *generalized discrete Morse function* and V is its *generalized discrete gradient*. The terminology reminds us that the existence of f is proof for the acyclicity of the directed graph obtained from the Hasse diagram by contracting the intervals to single nodes. If an interval contains only one simplex, then we call the interval *singular* and the simplex as well as its value *critical*. Notwithstanding the fact that Forman refers to discrete Morse

functions and discrete gradients in the aforementioned special case, we will feel free to use these terms to refer to the slightly generalized notions.

The intervals in a (gen.) discrete gradient correspond to collapses as described in [47, Chapter III], except that the lower bounds do not have to be free. A collapse can be realized continuously by a deformation retraction. The transformation of a simplicial complex, K, to another, K', using a sequence of collapses, thus implies that the two complexes have the same homotopy type. The notation for the existence of the collapses is $K \searrow K'$, and the implied relation is slightly stronger, which is usually expressed by saying that K and K' are *simple-homotopy equivalent* [46]. A (gen.) discrete gradient can encode sequences of collapses [50]:

Result 4.2 ((Gen.) Collapsing Proposition)**.** *Let K be a simplicial complex with a (gen.) discrete gradient, and let $K' \subseteq K$ be a subcomplex. If $K \backslash K'$ is a union of non-singular intervals, then $K \searrow K'$.*

4.3 Bregman Divergence

Strictly convex functions. Let $\mathbb{X} \subseteq \mathbb{R}^{n-1}$ be convex, and let $F \colon \mathbb{X} \to \mathbb{R}$ be differentiable and strictly convex. For two points $x, y \in \mathbb{X}$, the *Bregman distance* from x to y associated with F is the difference between the value of F at x and the value of the first-order Taylor expansion of F around y evaluated at x [45]:

$$D_F(x, y) = F(x) - \left[F(y) + \langle \nabla F(y), x - y \rangle\right]; \tag{7}$$

see Figure 13 for an illustration. Accordingly, the *Bregman ball* with center x and radius $r \geqslant 0$ consists of all points at Bregman distance at most r:

$$\mathbb{B}_{F,r}(x) = \{y \in \mathbb{X} \mid D_F(x, y) \leqslant r\}. \tag{8}$$

By assumption of strict convexity, the Bregman distance satisfies the first property required from a metric, namely $D_F(x, y) \geqslant 0$ for all x and y, with equality iff $x = y$. However, it is not necessarily symmetric – which we emphasize by calling $D_F(p, q)$ the distance *from x to y* – and it does not necessarily enjoy the triangle inequality. To compensate, every Bregman distance satisfies the

Result 4.3 (Convexity Property)**.** *$D_F(x, y)$ is strictly convex in the first argument, but not necessarily convex in the second argument.*

Proof. Fixing y, we set $f(x) = D_F(x, y)$ and note that f is the difference between F and an affine function; see (7). The strict convexity of F implies the strict convexity of f. The argument does not apply if we fix x and set $g(y) = D_F(x, y)$. Indeed, it is easy to find an example in which g is not convex. □

Crab lemma. As suggested in Figure 14, we will use the Crab Lemma to prove that the duality transform maps a convex function to a convex function.

We say two ordered pairs of points, A, B and P, Q, form a *crab configuration* if $B^*(a) - \alpha = Q^*(p) - \varphi = 0$ and $B^*(p) - \varphi < 0$ as well as $Q^*(a) - \alpha < 0$; see Figure 14. Applying (1) to the four relations, we see that $A^*(b) - \beta = P^*(q) - \psi = 0$ and $A^*(q) - \psi < 0$ as well as $P^*(b) - \beta < 0$. We state this result for later reference.

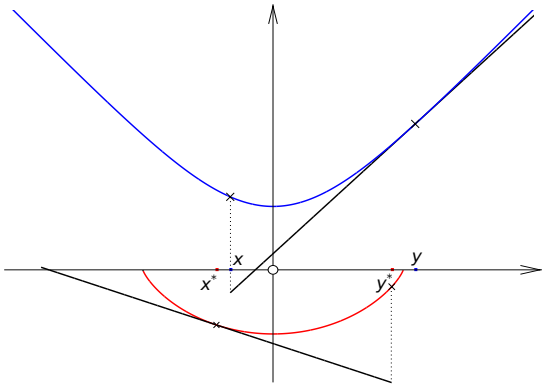

Figure 13: *Top:* the graph of F with highlighted Bregman distance from x to y associated with F. *Bottom:* the graph of F^* with highlighted Bregman distance from y^* and x^* associated with F^*. The two distances are the same.

Figure 14: The duality transform maps the crab configuration on the *left* to the crab configuration on the *right*.

Result 4.4 (Crab Lemma)**.** *The ordered pairs A, B and P, Q form a crab configuration in $\mathbb{R}^{n-1} \times \mathbb{R}$ iff B, A and Q, P form a crab configuration in $\mathbb{R}^{n-1} \times \mathbb{R}$.*

Let A, B and P, Q be two point-plane supports of F. By strict convexity, the two pairs form a crab configuration, and by the Crab Lemma so do B, A and Q, P. This proves that F^* is also strictly convex. Similarly, we can prove that F^* is differentiable and that $\mathbb{X}^* \subseteq \mathbb{R}^{n-1}$ is convex. It follows that there is a Bregman distance associated with F^*. It is common to refer to it as the *conjugate Bregman distance* of F: $D_F^*(x, y) = D_{F^*}(x^*, y^*)$. Importantly, the conjugate Bregman distance is the symmetric version of the Bregman distance.

Result 4.5 (Duality Property)**.** $D_F(x, y) = D_F^*(y, x)$.

Proof. Let $A = (a, \alpha)$, $B = (b, \beta)$ and $P = (p, \varphi)$, $Q = (q, \psi)$ be point-plane supports of F with $a = x$ and $p = y$. Then $D_F(x, y) = \alpha - Q^*(a)$, namely the vertical distance between A and the graph of Q^*. Similarly, $D_{F^*}(y^*, x^*) = \psi - A^*(q)$. By (1), the two distances are the same. □

The *conjugate Bregman ball* with center x and radius $r \geqslant 0$ is the set of points $y \in \mathbb{X}$ such that $D_F^*(x, y) = D_F(y, x) \leqslant r$. Since the Bregman distance is convex in the first argument, the Duality Property implies that the conjugate Bregman ball is strictly convex.

Role of conjugate balls Observe that whenever a collection of primal Bregman balls of radius r intersect, there exists a conjugate Bregman ball of radius r containing all the centers of these balls. Therefore, to compute the first radius r where the balls intersect, it suffices to answer a minimum enclosing Bregman ball query.

For the reader familiar with Voronoi diagrams, we note that they can be cast in the language of Bregman distances. To explain this, let $F\colon \mathbb{R}^{n-1} \to \mathbb{R}$ be defined by mapping x to $\|x\|^2$. Then $D_F(x,q) = \|x-y\|^2$, the squared Euclidean distance between the points. For this particular choice of function, we have $F = F^*$ and $D_F(x,y) = D_F^*(x,y)$. This additional symmetry is the reason that there is no notion of conjugate Voronoi diagram for the Euclidean distance function.

Examples. Bregman distances have numerous practical applications. One prominent example the *Kullback-Leibler divergence*, used in the context of computer vision and text analysis. Generally it can be used to compare histograms, or more generally probability distributions. It has strong theoretical justification, being closely related to the notion of entropy.

Another important measure is the Itakura-Saito divergence, successfully used in voice recognition.

Conjugate. Let $\mathbb{X} \subseteq \mathbb{R}^{n-1}$ be convex, and let $F\colon \mathbb{X} \to \mathbb{R}$ be a differentiable and strictly convex function. A *point-plane support* of F is half a crab, that is: an ordered pair of points $A = (a,\alpha)$, $B = (b,\beta)$ in $\mathbb{R}^{n-1} \times \mathbb{R}$ with $\alpha = F(a)$, $b = \nabla F(a)$, and $\langle b, a\rangle - \alpha - \beta = 0$. The *conjugate* of F is the unique function $F^*\colon \mathbb{X}^* \to \mathbb{R}$ such that B, A is a point-plane support of F^* whenever A, B is a point-plane support of F. Note that F and a determine α, b, β. Indeed, we write $b = a^*$ – calling it the *conjugate point* of a – and we observe that $\mathbb{X}^*$ is the set of points a^* with $a \in \mathbb{X}$. In the literature, the map from F to F^* is referred to as the *Legendre transform*.

Slices. Let $\mathbb{Y} \subseteq \mathbb{R}^{n-1}$ and let $E\colon \mathbb{Y} \to \mathbb{R}$ be differentiable. By a *slice* of E we mean the restriction to a linear subspace $\mathbb{L}$ of $\mathbb{R}^{n-1}$. Writing $\mathbb{X} = \mathbb{Y} \cap \mathbb{L}$ and $F\colon \mathbb{X} \to \mathbb{R}$ for this restriction, we note that F is differentiable, and we require that $\mathbb{X}$ be convex and F be strictly convex. For this to be true, it is sufficient but not necessary that $\mathbb{Y}$ be convex and E be strictly convex. We are interested in the relation between the partial derivatives of E and of F.

Write s^* for the vector of partial derivatives of E at $s \in \mathbb{Y}$, and let $\mathrm{proj}_{\mathbb{L}}\colon \mathbb{R}^{n-1} \to \mathbb{L}$ be the orthogonal projection onto $\mathbb{L}$. For points $s \neq t$ in $\mathbb{X}$, the strict convexity of F implies $s^* \neq t^*$ as well as $\mathrm{proj}_{\mathbb{L}}(s^*) \neq \mathrm{proj}_{\mathbb{L}}(t^*)$. Indeed, $x^* = \mathrm{proj}_{\mathbb{L}}(s^*)$ and $y^* = \mathrm{proj}_{\mathbb{L}}(t^*)$ are the conjugate points of $x, y \in \mathbb{X}$ with respect to F. Furthermore, the function values are inherited from E. More precisely, we have the following characterization of the conjugate of F.

Result 4.6 (Slice Lemma)**.** *Let $E\colon \mathbb{Y} \to \mathbb{R}$ be differentiable, let $F\colon \mathbb{X} \to \mathbb{R}$ be a slice of E, let x be a point in $\mathbb{X} \subseteq \mathbb{Y}$, and write $s^* = \nabla E(x)$ and $x^* = \nabla F(x)$. Then $x^* = \mathrm{proj}_{\mathbb{L}}(s^*)$ and $F^*(x^*) = E(x) - \langle s^*, x\rangle$.*

A special case of the lemma characterizes the relationship between the conjugate of a strictly convex function E and the conjugate of a slice F of E. We will use this lemma in the more general context in which E is not strictly convex.

4.4 Bregman filtrations

In this section we give geometric interpretations of complexes and their filtrations based on a Bregman distance.

4.4.1 Validity of constructions

All our constructions rely on intersecting subsets of a topological space. The key theoretical idea behind this is the Nerve Lemma. To use the lemma, we show that the appropriate subsets intersect in contractible sets.

Result 4.7 (Conjugate mapping is a homeomorphism)**.** *The conjugate mapping defines a homeomorphism.*

Proof. Sketch: we only depend on the gradient of a function that is continuously differentiable, so the conjugacy mapping is continuous. Also, we know there is an inverse, which is continuous for the same reason, so overall it is a homeomorphism. □

Result 4.8 (Bregman balls intersect contractibly)**.** *The intersection of a collection of Bregman balls is contractible.*

Proof. With the above conjugate mapping, we have a homeomorphism between primal and conjugate balls, as well as between intersections of primal balls and intersections of conjugate balls. Since conjugate balls intersect contractibly (due to their convexity), primal balls intersect contractibly as well.. □

Here, we state the following results without proof:

- Bregman Voronoi domains intersect contractibly
- Bregman Voronoi domains intersect balls contractibly
- Bregman Delaunay domains intersect contractibly

4.4.2 Čech Construction

As illustrated in Figure 15, we get the ball centered at x by projecting the patch of $\mathcal{E}$ visible from the point obtained by pulling $s = (x, F_1(x))$ vertically down by r.

Čech radius function. For each simplex $\xi \in \Delta(X)$, there is a smallest radius for which ξ belongs to the Čech complex:

$$r_{\mathrm{C}}(\xi) = \min\{r \mid \xi \in \mathrm{Cech}_r(X)\}. \tag{9}$$

We call $r_{\mathrm{C}} \colon \Delta(X) \to \mathbb{R}$ the *Čech radius function* of X. Referring to Figure 15, we give two geometric interpretations:

[C_1] Consider the convex hull of the points $s = (x, F_1(x))$ with $x \in \xi$. Then $r_{\mathrm{C}}(\xi)$ is the maximum distance between a point in this convex hull and the point of $\mathcal{E}$ vertically below it.

[C_2] Among all points that lie on or below all $(n-1)$-planes that touch $\mathcal{E}^*$ in points $s^* = (x^*, F_1^*(x^*))$ with $x \in \xi$, take the one that minimizes the vertical distance to $\mathcal{E}^*$. This vertical distance is $r_{\mathrm{C}}(\xi)$.

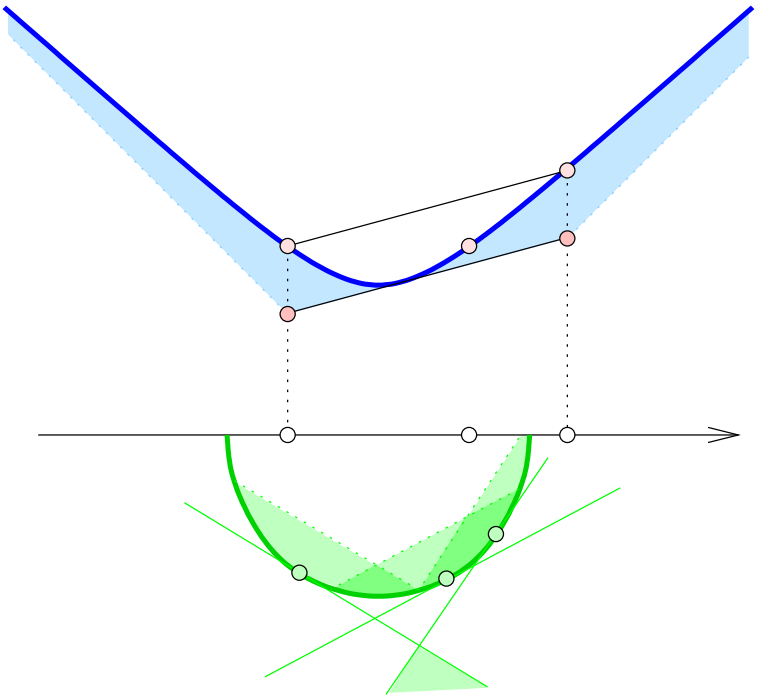

Figure 15: The radius is chosen so that the balls centered at x and at y touch inside a third ball. The Čech complex consists of a triangle and its faces.

Discrete Morse function. A combinatorial property of r_{C} with important structural consequences is the following:

Result 4.9 (C-Gdmf Lemma)**.** *Letting $X \subseteq \mathbb{R}^{n-1}$ be finite, the Čech radius function, $r_{\mathrm{C}} \colon \Delta(X) \to \mathbb{R}$, is a generalized discrete Morse function.*

Proof. It is clear that $r_{\mathrm{C}}(\xi) \leqslant r_{\mathrm{C}}(\eta)$ whenever $\xi \subseteq \eta$. Assuming the points are in general position, we will show that for every radius value, the collection of simplices in $\Delta(X)$ sharing this value is an interval in the face lattice. This implies that r_{C} is a generalized discrete Morse function; see Section 4.2.

□

4.4.3 Delaunay Construction

We first give a geometric interpretation of the Voronoi diagram, and its dual, Delauney triangulation. Then we construct a filtration using the Delaunay complex with an appropriate radius function.

Voronoi diagram. Referring to Figure 16, we have the following geometric interpretation:

[V_1] Find the $(n-1)$ planes that touch $\mathcal{E}^*$ at the points $s^* = (x^*, F_1^*(x^*))$ with $x \in X$, and intersect the closed half-spaces bounded from below by these planes. The Voronoi diagram is

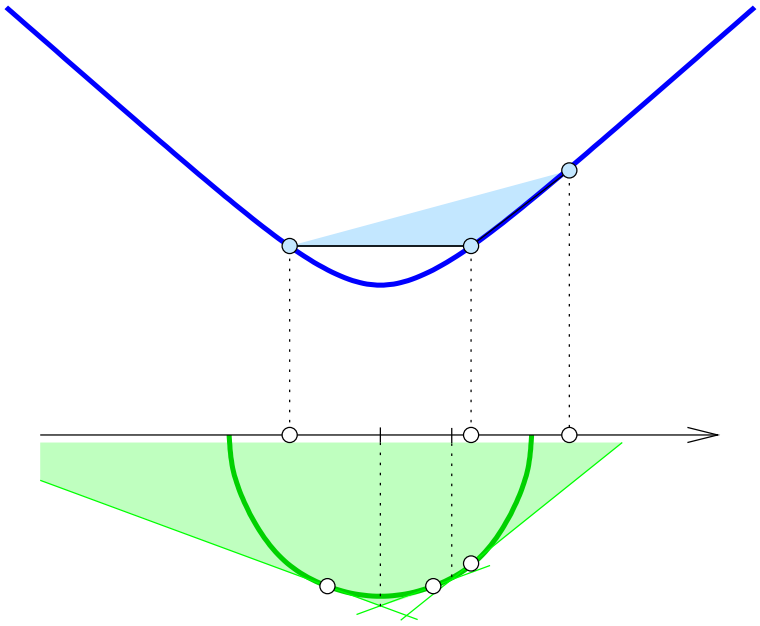

Figure 16: *Top:* the Delaunay triangulation is the vertical projection of the lower boundary complex of the convex hull. *Bottom:* the conjugate image of the Voronoi diagram is the vertical projection of the $(n-1)$-faces in the boundary of the intersection of halfspaces.

the conjugate image of the projection of the $(n-1)$-dimensional faces of this convex polyhedron to $\mathbb{R}^{n-1}$.

Delaunay triangulation. Referring to Figure 16, we have the following geometric interpretation of the Delaunay triangulation:

[V_2] Assuming general position, the convex hull of the points $s = (x, F_1(x))$ with $x \in X$ is a simplicial polyhedron. Projecting the lower faces to $\mathbb{R}^{n-1}$ gives the Delaunay triangulation.

Delaunay complex. Referring to Figure 16, we get two geometric interpretations:

[D_1] Moving the convex hull vertically down by r, the simplices in $\mathrm{Del}_r(X)$ correspond to the lower faces whose affine hulls have only points on or below $\mathcal{E}$.

[D_2] Moving the intersection of half-spaces vertically up by r, the simplices in $\mathrm{Del}_r(X)$ correspond to the faces that have points on or above $\mathcal{E}^*$.

Delaunay radius function. For each $\xi \in \mathrm{Del}(X)$, there is a smallest radius for which ξ belongs to the Delaunay complex:

$$r_{\mathrm{D}}(\xi) = \min\{r \mid \xi \in \mathrm{Del}_r(X)\}. \tag{10}$$

We call $r_{\mathrm{D}}\colon \mathrm{Del}(X) \to \mathbb{R}$, the *Delaunay radius function* of X. Using [D_1] and referring to Figure 16, we have the following geometric interpretation:

[D_3] Consider the affine hull of the points $s = (x, F_1(x))$ with $x \in \xi$. Then $r_{\mathrm{D}}(\xi)$ is the maximum vertical distance between a point of the affine hull and the point of $\mathcal{E}$ vertically below it.

Discrete Morse function. A combinatorial property of r_{D} with important structural consequences is the following:

Result 4.10 (D-Gdmf Lemma)**.** *Letting $X \subseteq \mathbb{R}^{n-1}$ be finite, the Delaunay radius function, $r_{\mathrm{D}}\colon \mathrm{Del}(X) \to \mathbb{R}$, is a generalized discrete Morse function.*

Proof. It is clear that $r_{\mathrm{D}}(\xi) \leqslant r_{\mathrm{D}}(\eta)$ whenever $\xi \subseteq \eta$. Assuming the points are in general position, we will show that for every radius value, the collection of simplices in $\mathrm{Del}(X)$ sharing this value is an interval in the face lattice. This implies that r_{D} is a generalized discrete Morse function; see Section 4.2.

□

4.4.4 Comparison

We aim at comparing the radius functions with the measure R_{gm} on simplices as a possible choice for generalizing beyond pairs.

Conjugate measures. Using the conjugate distance instead is equivalent to switching the roles of X and X^*. We therefore construct $\mathrm{Cech}_r(X^*)$, $\mathrm{Vor}(X^*)$, $\mathrm{Del}(X^*)$, $\mathrm{Del}_r(X^*)$ and the corresponding radius functions, $r^*_{\mathrm{C}}\colon \Delta(X^*) \to \mathbb{R}$ and $r^*_{\mathrm{D}}\colon \mathrm{Del}(X^*) \to \mathbb{R}$. Casting them into the space of the term-vectors gives

$$R^*_{\mathrm{C}}(\xi^*) = e^{-r^*_{\mathrm{C}}(\xi^*)/\sqrt{n}}, R^*_{\mathrm{D}}(\xi^*) = e^{-r^*_{\mathrm{D}}(\xi^*)/\sqrt{n}}. \tag{11}$$

To have a complete set of measures, we add the conversion of the geometric mean of the conjugate points to our set:

$$R^*_{\mathrm{gm}}(\xi^*) = e^{-r^*_{\mathrm{gm}}(\xi^*)/\sqrt{n}}. \tag{12}$$

Ranking. We claim inequalities between the measures. In contrast to R_{gm} and R_{C}, the converted Delaunay radius functions are not defined on all simplices. We therefore set $R_{\mathrm{D}}(\xi) = 0$ for all simplices $\xi \notin \mathrm{Del}(X)$, and similarly for the conjugate version.

Result 4.11 (Ranking Lemma)**.** *Let $X \subseteq \mathbb{R}^{n-1}$ be finite and $\xi \subseteq X$.*

$$R_{\mathrm{D}}(\xi) \quad \leqslant R_{\mathrm{C}}(\xi) \quad \leqslant R_{\mathrm{gm}}(\xi), R^*_{\mathrm{D}}(\xi^*) \leqslant R^*_{\mathrm{C}}(\xi^*) \leqslant R^*_{\mathrm{gm}}(\xi^*). \tag{13}$$

Proof. We use geometric interpretations of $R_{\mathrm{gm}}, R_{\mathrm{C}}, R_{\mathrm{D}}$ for a simplex $\xi \subseteq X$ to prove the first set of inequalities. Set $k+1 = \operatorname{card}\xi$ and suppose first that the k-simplex belongs to the Delaunay triangulation of X. Let σ be the set of points $s = (x, F_1(x))$ with $x \in \xi$, and let $\Sigma = \operatorname{aff}\sigma$. Assuming the points are in general position, Σ is a k-plane with $k < n$. We are interested in three not necessarily distinct points in Σ. To define them, we note that for every point $a \in \Sigma$ there is a unique radius $R = R(a)$ such that $a \in E^{-1}(R^2)$.

- s_{gm} is the barycenter of σ;

- s_{C} is the point of $\operatorname{conv}\sigma$ that minimizes $R(s_{\mathrm{C}})$;
- s_{D} is the point of Σ that minimizes $R(s_{\mathrm{D}})$.

We have $R(s_{\mathrm{C}}) \leqslant R(s_{\mathrm{gm}})$ because s_{gm} is a point of $\operatorname{conv}\sigma$, and $R(s_{\mathrm{D}}) \leqslant R(s_{\mathrm{C}})$ because $\operatorname{conv}\sigma \subseteq \Sigma$. To see the first set of inequalities, it remains to notice that the R-values of these points coincide with the measures of the simplex:

- $R_{\mathrm{gm}}(\xi) = R(s_{\mathrm{gm}})$ by definition of R_{gm};
- $R_{\mathrm{C}}(\xi) = R(s_{\mathrm{C}})$ by $[\mathrm{C}_1]$;
- $R_{\mathrm{D}}(\xi) = R(s_{\mathrm{D}})$ by $[\mathrm{D}_3]$.

By setting $R_{\mathrm{D}}(\xi)$ to the smallest possible value whenever it is not otherwise defined, we make sure that (13) also holds for non-Delaunay simplices. Finally, we get (**??**) by the symmetric argument. □

4.5 Example application.

In this section we show how the above theory can be used in the context of analyzing text documents.

Vector space model. The *vector space model* introduced in [55] turns a corpus of documents into a geometric object, namely a set of points in $\mathbb{R}^n$. Specifically, each document is represented as a sequence of real numbers, called a *term-vector*. Each coordinate corresponds to one of n words or *terms* and depends on the frequency within the document and within the corpus. Intuitively, the prominent terms determine the *topic* of the document.

Cosine similarity measure. A popular notion for comparing term-vectors is the *cosine similarity measure*: $R_{\cos}(p^0, p^1) = \sum_{j=1}^{n} p_j^0 p_j^1$, in which $p^0 = (p_1^0, p_2^0, \ldots, p_n^0)$ and $p^1 = (p_1^1, p_2^1, \ldots, p_n^1)$ are two L_2-normalized term-vectors. This measure suffices to answer simple queries, such as finding all documents with similarity above some threshold to a query document. More challenging questions require more sophisticated measures. In particular, we are interested in capturing the global topology of a corpus, or of a subset of the documents embedded within the corpus. For example, we may ask whether there is evidence for significant holes in the collection. To approach this and similar questions, we extend the cosine similarity measure from pairs to tuplets of size $k+1$, which we refer to as *k-simplices*. A straightforward idea would be to map $k+1$ normalized term-vectors to the sum of the componentwise products:

$$R_{\cos}(p^0, p^1, \ldots, p^k) = \sum_{j=1}^{n} \prod_{i=0}^{k} p_j^i. \tag{14}$$

For $k = 1$, this agrees with the cosine similarity measure, but it lacks a useful geometric interpretation. A geometrically more appealing option is the Euclidean length q q of the geometric mean:

$$R_{\mathrm{gm}}(p^0, p^1, \ldots, p^k) = \left(\sum_{j=1}^{n} \left(\prod_{i=0}^{k} p_j^i \right)^{\frac{2}{k+1}} \right)^{\frac{1}{2}}, \tag{15}$$

which for $k = 1$ is the square root of the cosine similarity measure. While this is an attractive option, it has subtle drawbacks, which will become apparent when we discuss the emerging geometric structure. In a nutshell, the flaw of R_{gm} is its sole dependence on the documents in the simplex, not taking into account its surrounding. We say the measure of a simplex is *context-sensitive* if it adapts to term-vectors in the local neighborhood. Our declared goal is to introduce such measures and to demonstrate how context-sensitivity can be used to enable topological analysis tools.

4.6 Discussion

This shows that simplicial complexes can be built based on divergences. This may represent a more informative approach to topological analysis for textual data, although the applications are more widespread. This can represent a "personalized" view of topology and persistence, where rather than consider isotropic balls around points, we consider non-symmetric balls depending on information theoretic measures. Although this work is in its early stages, we believe it represents an important generalization to the classes of filtrations used in persistent homology.

5 Improved Runtime Analysis Using Nested Dissection

The relevant information in persistent homology can be read off from a (generalized) LU-decomposition of the *boundary matrix*. All persistent homology algorithms used in practice are based on matrix reduction through row or column operations on an initially sparse boundary matrix. Due to matrix fill-in, a cubic complexity in the size of the matrix can be achieved on worst-case examples [26]. On the other hand, algorithms often show near-linear asymptotic behavior on most realistic inputs. It is likely that the structure of realistic examples keeps the matrices sparse during the computation, but how can we quantify this?

Generalized nested dissection [22] is a technique to reduce the effect of fill-in in Gaussian elimination and related matrix reduction problems for instances which have certain structure. Originally, the method was developed for symmetric matrices. Nested dissection interprets the matrix as a graph and looks for a *separator*, a subset of vertices C whose removal splits the graph into two disconnected parts A and B. The missing connections between A and B correspond to areas of the matrix in which no fill-in will happen during the reduction, if the pivots are chosen in a suitable order. If the graph is guaranteed to have a "good" separator (C is small and both A and B are not too small), it can be shown that this strategy yields matrices whose fill-in is asymptotically subquadratic, and matrix reduction can be performed in subcubic time.

A natural question is: can nested dissection be applied in the context of computing persistent homology? There are several obstructions for combining these two techniques. The first is that nested dissection is based on graph separators, whereas persistent homology operates on simplicial complexes. We therefore require a separator theory on simplicial complexes that not only gives the existence of separators, but also efficient algorithms to find them in common instances. The second obstruction is that nested dissection was originally designed for symmetric square matrices, whereas the boundary matrices in persistent homology are asymmetric. While the asymmetric case has been studied (see related work below), those techniques must be adapted to the case of persistent homology. Finally, the third obstruction is that nested dissection chooses a pivot order in the matrix reduction process to guarantee a fast running time, whereas persistent homology does not permit arbitrary column swaps without changing the outcome. While the effect of a column swap is well understood for persistent homology [8], there is no previous evidence that changing the pivot order is beneficial in terms of algorithmic complexity.

Contributions We show how to overcome all the obstructions mentioned above and apply the theory of nested dissection in the context of persistent homology computation.

1. We extend the theory of geometric separators to simplicial complexes, proving that they exist in a widely-used class of geometric complexes and can be computed efficiently. We introduce an analogous notion of a *β-separable complex* where every subcomplex has vertex separators of size $O(n^\beta)$ that separate the complex into pieces no bigger than αn for some fixed $\alpha, \beta < 1$.
2. We analyze a simple variant of the classic persistence algorithm and show that its fill-in can be bounded by the same techniques as used in nested dissection. In particular, if the columns are ordered according to a nested dissection ordering of $O(n^\beta)$-size separators, then the fill-in is $O(n^{2\beta})$ and the running time is $O(n^{3\beta})$.
3. We use the Vineyards algorithm from [8] to transform the persistent homology of the nested dissection order to that of an arbitrary order in $O(n^{2+\beta})$. The worst-case of $O(n^3)$ was the best previously known.

4. Using the framework of the output-sensitive persistence algorithm from [7], we show for β-separable filtered simplicial complexes with only a constant number of ε-persistent features, the persistent homology can be computed in $O(n^{\beta\omega})$ time, where $\omega \leqslant 2.373$ is the matrix multiplication exponent [21]. The best-known bound for arbitrary input complexes is $\tilde{O}(n^2)$ [7].

Related Work Since the first algorithm for persistent homology by Edelsbrunner et al. [13], many variants have been proposed. The only unconditional subcubic complexity bound by Milosavljević et al. [25] employs fast matrix multiplication to run in $O(n^\omega)$ time. Edelsbrunner and Parsa show that computing Betti numbers, and thus also persistent homology, is at least as hard as computing ranks of sparse $n \times n$ matrices [14]. Chen & Kerber [7] give an output-sensitive algorithm whose running time depends only on the running time of matrix rank computations and the number of features that persist longer than some threshold; it is also the only variant with subquadratic worst-case space complexity. Both variants lack a proof of usefulness in application scenarios. The currently fastest approaches in practice are based on the cohomological persistence algorithm by De Silva et al. [11] and on heuristics of the standard reduction algorithm that exploit the structure of the boundary matrices [6]. Some algorithmic variants are analyzed in terms of additional parameters in addition to the input size, including the total (index) persistence of the filtration [6], the number of critical cells in a cubical complex [3], or the maximum Betti number among all the complexes in the input filtration [9, 2]; while these results partially explain the excellent behavior of persistent homology in practice, the worst-case for all these variants remains cubic in the input size. Several open source software libraries implement variations of these algorithms [4, 10, 24].

Bounding fill-in during the Gaussian elimination of sparse matrices is in general, known to be NP-hard [30]. A natural question arose was if it could be bounded in special cases.. Nested dissection was first presented for regular grids by George [17]. It was generalized to arbitrary, recursively separable graphs by Lipton et al. [22], whose analysis is based on the work of Rose et al. [28] which we also employ in our analysis. An alternative analysis was later given by Gilbert & Tarjan [18] who introduced some insights that allow one to weaken the condition on the recursive separators. Most of the work on nested dissection has been limited to the case of real, symmetric, positive definite matrices. Grigori et al. [16] consider a non-symmetric version, but do not prove explicit bounds on the fill or the work. Another work in the same spirit, though not explicitly about nested dissection, is the work of Carlsson & de Silva [5] on geometric sparsity in solving linear systems. Yuster [31] and Alon & Yuster [1] eliminates the conditions on symmetry and positive-definiteness while also allowing computations over arbitrary fields.

5.1 Separating Graphs and Nested Dissection

Nested dissection is a method for ordering pivots in Gaussian elimination of sparse matrices to reduce fill-in and running time. The idea is to treat the matrix as a graph and order the pivots based on recursive decomposition of the graph. The main definition is that of a *graph separator*:

Definition 5.1. *A graph $G = (V, E)$ with $|V| = n$ has a $(f(n), \alpha)$-**separation** with $\alpha \in (1/2, 1)$ if V can be partitioned into 3 parts X, Y, and Z such that*

$$|X|, |Y| \leqslant \alpha n, \text{ and } |Z| \leqslant f(n),$$

*and also no edge of E has one endpoint in X and one end point in Y. The set Z is referred to as a **separator of the graph** G.*

The quality of a separator depends on the growth of f. Throughout paper, we will exclusively consider functions of the form $f(n) = \gamma n^\beta$ with $\beta \in (0, 1)$ and $\gamma > 0$.

Definition 5.2. *A **graph** G **is** β-**separable** if for some fixed α and γ, every subgraph has a $(\gamma n^\beta, \alpha)$-separation.*

The requirement that the separations exist for all subgraphs allows us to recursively separate the graph into smaller pieces for divide-and-conquer. This is the reason for the name *nested dissection*. For brevity in some of the statements, we assume henceforth that $\beta > 1/2$; while some types of graphs are 1/2-separable (in particular, planar graphs [23]), larger values of β seem more common in the situations that we are considering.

$\mathcal{V}$-paths and separator trees The following concepts and results are taken from [22]. Let $G = (V, E)$ be a graph and π be a numbering of its vertices, that is, a bijective map $V \to [n]$ with $[n] := \{1, \ldots, n\}$. We call a path between v_i and v_j with $i < j$ in G a *$\mathcal{V}$-path*, if no vertex is repeated and except for i and j, each vertex on the path has an index smaller than i. When the ordering is fixed, we will sometimes identify a vertex of G and its index.

The importance of $\mathcal{V}$-paths is as follows: If A is a symmetric positive definite matrix with Cholesky decomposition $A = LL^T$, and G is the adjacency graph of A, the non-zero entries of L are in one-to-one correspondence to $\mathcal{V}$-paths [22, Lemma 1] (attributed to [28]), if the elimination process is performed according to the ordering π.

To bound the number of $\mathcal{V}$-paths, assume that G is a β-separable graph. We say that G is in *nested dissection ordering*, if its vertices are sorted according to the following *Numbering Algorithm* [22, Sec.2]. It assumes that ℓ vertices are already ordered with indices larger than b and it numbers the remaining vertices consecutively from a to b. If G has at most $n_0 = (\gamma/(1-\alpha))^{1/(1-\beta)}$ vertices, we number the vertices arbitrary from a to b. Otherwise, we split the vertex set into (A, B, C), where C is the separator of size $O(n^\beta)$ and there is no edge connecting a vertex of A and a vertex of B. Let i, j, k denote the unnumbered vertices of A, B, C, respectively. We number the remaining vertices of C from $b - k + 1$ to b. We apply the numbering algorithm recursively on the subgraph induced by $B \cup C$ to assign the range $b - k - j + 1$ to $b - k$. Finally, we apply the numbering algorithm recursively on the subgraph induced by $A \cup C$ to assign the range 1 to $b - k - j$. This concludes the description of the numbering algorithm.

The numbering algorithm gives rise to a *separator tree*: it is a rooted binary tree where each node corresponds to a recursive call of the algorithm. Each node contains the subset of vertices that are numbered in the corresponding recursive call. In particular, the root contains precisely the vertices of the first separator, and each internal node contains a subset of the separator constructed for the corresponding point set (it is only a subset because part of the separator may already have been numbered in a preceding call). The leaves contain at most n_0 vertices. Moreover, all vertices stored in a node have larger indices than all vertices stored in a successor of that node in the separator tree.

Lemma 5.3. *An internal node of the separator tree on level ℓ contains at most $\gamma(1-\varepsilon)^{\beta\ell} n^\beta$ vertices with $\varepsilon = (1 - \alpha - \gamma/(n_0 + 1)^{1-\beta})$.*

Proof. The claim follows by induction, showing that the algorithm recurses on at most $(1 - \varepsilon)^\ell n$ vertices; see [22, Theorem 2] for further details. □

The next two lemmas bound the number of $\mathcal{V}$-paths for a nested dissection ordering π locally (Lemma 5.4) and globally (Lemma 5.5).

Lemma 5.4. *For any vertex j, the number of $\mathcal{V}$-paths with lower endpoint j is $O(n^\beta)$.*

Proof. All such $\mathcal{V}$-paths are formed with vertices stored in the same or a predecessor node in the separator tree, and their size is bounded by a geometric sum, similar to [18, Lemma 5]. □

Lemma 5.5. *Let G be a β-separable graph and let its vertices be in nested dissection ordering π. Then, the number of $\mathcal{V}$-paths is $O(n^{2\beta})$.*

Proof. The number of $\mathcal{V}$-paths is bounded in [22, Theorem 2] to bound the fill-in of a Cholesky decomposition. More precisely, the result is exposed for $\beta = 1/2$, but extends to $\beta > 1/2$ as mentioned in [22, Theorem 6-9]. □

5.2 Separators on Simplicial Complexes

Given a symmetric, $n \times n$ matrix M, the **graph of** M is the n vertex graph with edges corresponding to the nonzero entries of M, i.e. $\mathcal{G}_M = ([n], \{(i,j) : M_{ij} \neq 0\})$. With this definition, the notions of separators and β-separability carry over to symmetric matrices in a natural way. For simplicial complexes, we can define a symmetric graph for a fixed dimension as follows:

Definition 5.6. *The p-**skeleton graph of a simplicial complex** $\mathcal{K}$ is the graph whose vertices are the p-simplices of $\mathcal{K}$ and whose edges are the pairs of p-simplices that have a common $(p-1)$-dimensional face. It is denoted $\mathcal{G}_{\mathcal{K}^{(p)}}$.*

Recall the notion of the pth boundary matrix ∂_p of $\mathcal{K}$. Clearly, $\partial_p^\top \partial_p$ is a symmetric $n_p \times n_p$-matrix, and it is not hard to see that $\mathcal{G}_{\mathcal{K}^{(p)}} = \mathcal{G}_{\partial_p^\top \partial_p}$, where the matrix multiplication is performed over an arbitrary base field $\mathbb{F}$. We remark that the same holds true when we include a diagonal matrix R with non-zero diagonal entries in the symmetrization as in $\partial_p^\top R \partial_p$.

Definition 5.7. *A **simplicial complex** $\mathcal{K}$ **is** β-**separable** if $\mathcal{G}_{\mathcal{K}^{(p)}}$ is β-separable for all $p \in \mathbb{N}$.*

A natural question is: how common are β-separable complexes? We show that if vertices have bounded degree, separability of the 1-skeleton implies separability for the complex.

Theorem 5.8. *Let $\mathcal{K}$ be a d-dimensional simplicial complex for some constant d and let G be its 1-skeleton. If G is β-separable and has maximum degree $\Delta = O(1)$, then $\mathcal{K}$ is β-separable as well.*

$$|X_T| = m - |Y_T \cup Z_T| \leqslant m - \frac{1}{p+1}|Y_V \cup Z_V| \leqslant m - \frac{1}{p+1}(1-\alpha)n \leqslant m\left(1 - \frac{1-\alpha}{\Delta_p}\right) = \alpha' m.$$

There is naturally some degradation in the quality of the separator in terms of α and γ as the dimension increases, but this degradation does not affect the exponent in the separator size.

Geometric separators Our results only assume β-separability. A good example, which is used in TDA, is the Delaunay triangulation of well-spaced points [19, 29]. We show that for d-dimensional inputs, this complex is β-separable with $\beta = 1 - 1/d$.

We recall some notions from computational geometry. Given a finite set of points $P \subset \mathbb{R}^d$, the *Voronoi diagram* splits $\mathbb{R}^d$ into (closed) polytopes, called *Voronoi regions*, where the Voronoi region V_q of q consists of all points in $\mathbb{R}^d$ that are at least as close to q as to any other point of P. For V_q,

let the *in-radius* $r(q)$ be the maximal radius of a ball centered at q that is contained in V_q and let the *out-radius* $R(q)$ be the minimal radius of a ball centered at q that contains all boundary vertices of V_q. We say that P is *τ-well-spaced* if $R(q)/r(q) \leqslant \tau$ for each $q \in P$. A relatively simple packing argument shows that for a set of τ-well-spaced points, each Voronoi region intersects $k(\tau, d)$ other Voronoi regions, where $k(\tau, d)$ is a constant independent of n.

The dual of the Voronoi diagram is the *Delaunay triangulation*, Del_P. We define it as a simplicial complex with vertex set P, where a simplex σ is in Del_P if the corresponding Voronoi regions intersect. When no $k+3$ points of P lie on a common k-sphere for $k < d$, the Delaunay triangulation is d-dimensional and has a natural embedding in $\mathbb{R}^d$. If this condition is not met, there are known ways to perturb the points to make it so.

Miller et al. [27] studied separators on the 1-skeleton of Del_P for well-spaced sets P arising in finite element analysis. Their algorithm and Theorem 5.8 imply the following theorem (see the full version for details).

Theorem 5.9. *If P is a set of τ-well-spaced points for some constant τ, then, Del_P is β-separable with $\beta = 1 - 1/d$, and the separator tree can be computed in expected $O(n)$ time.*

5.3 Fill and Work in a Persistence Algorithm

We show that the technique of nested dissection also applies to the problem of persistence computation via matrix reduction. This gives a method computing the homology of a large class of geometric simplicial complexes faster than matrix multiplication time.

Matrix Reduction We present a simple algorithm for computing persistence that we refer to as *the persistence algorithm* in this work (Algorithm 1). The reduction strategy resembles the annotation algorithm [9, 2] as well as the reduction implicit in the fast matrix multiplication algorithm [25].

Algorithm 1 The persistence algorithm

```
1:  procedure MATRIX_REDUCTION(∂)
2:      R ← ∂
3:      U ← I
4:      for j = 1, ..., n do
5:          if R_j ≠ 0 then
6:              i ← low_R(j)
7:              for each k such that k > j and R_ik ≠ 0 do
8:                  U_jk ← R_ik / R_jk
9:                  R_k ← R_k − U_jk R_j
10:     return R
```

The algorithm processes columns from left to right, finding the lowest nonzero element and zeroing out the rest of the row by column operations. Upon termination, R will be reduced and U will be unit upper triangular. The algorithm runs in $O(n^3)$ time on an $n \times n$ boundary matrix.

Define the *fill*, denoted $\mathrm{fill}(M_j)$ ($\mathrm{fill}(M^j)$), as the number of non-zero entries of the jth column (jth row) of matrix M, and let $\mathrm{fill}(M)$ be the total number of non-zero entries of M. By a careful choice of the data structures in Algorithm 1, we can bound the running time of the algorithm in terms of the fills of R and U. A major difference between this analysis and the classic setting of

nested dissection is that we use the sparsity of the input matrix ∂ to bound the fill even thought the matrix is not symmetric.

Recall the notion of pth skeleton graphs (Definition 5.6) and let $\mathcal{G} := \mathcal{G}_{\mathcal{K}^{(p)}}$. The major insight with respect to Algorithm 1 is that the fill-in of U is at most the number of $\mathcal{V}$-paths of $\mathcal{G}$.

Lemma 5.10. *If $U_{jk} \neq 0$, there exists a $\mathcal{V}$-path between σ_j and σ_k in $\mathcal{G}$.*

Proof. We prove by induction that after the jth iteration of Algorithm 1, for all entries $U_{\ell k} \neq 0$ with $\ell \leqslant j$ there exists a $\mathcal{V}$-path between σ_ℓ and σ_k. Assume that in the jth iteration, the algorithm encounters a non-zero column with lowest index i, and assume that another column $k > j$ has a non-zero entry in row i. Since the algorithm sets U_{jk} to a non-zero entry, we have to show that there exists a $\mathcal{V}$-path from σ_j to σ_k.

When the jth iteration starts, both the jth and the kth column of R are linear combinations of columns of the input matrix ∂. More precisely, we can write $L_j = \partial_j + \sum_{\ell<j} a_\ell \partial_\ell$ with some coefficients a_ℓ. Moreover, for any $a_\ell \neq 0$, column ℓ was added to j and consequently, $U_{\ell j} \neq 0$. By induction, there exists a $\mathcal{V}$-path between ℓ and j. The same statements are true for k and its linear combination $L_k = \partial_k + \sum_{\ell<j} b_\ell \partial_\ell$ Because the ith row of column j is not zero, some simplex σ_1 contributing to the linear combination L_j has the $(p-1)$-simplex τ associated with the i-row in its boundary, and there is a $\mathcal{V}$-path from σ_j to σ_1. Similarly, there is a simplex σ_2 with boundary simplex τ and a $\mathcal{V}$-path to σ_k. Because of τ, σ_1 and σ_2 are connected in $\mathcal{G}$, and the composition yields a $\mathcal{V}$-path between σ_j and σ_k. □

Let $\mathcal{K}$ be a β-separable simplicial complex (Definition 5.7). We call a filtration of $\mathcal{K}$ a *nested dissection filtration*, if for any dimension p, the filtration orders the p-simplices such that (the graph of) $\partial_p^T \partial_p$ is in nested dissection ordering, where ∂_p is the pth boundary matrix of $\mathcal{K}$.

Lemma 5.11. *For a β-separable simplicial complex with nested dissection filtration and any dimension p, let U and R be the result of Algorithm 1. Then, for any row of U, $\mathrm{fill}(U^j) = O(n^\beta)$. Moreover, $\mathrm{fill}(U)$ and $\mathrm{fill}(R)$ are in $O(n^{2\beta})$.*

Proof. The bound on $\mathrm{fill}(U^j)$ follows from Lemma 5.5 and Lemma 5.10. Similarly the bound on $\mathrm{fill}(U)$ follows by combining Lemma 5.4 and Lemma 5.10.

For R, observe that $R = \partial U$, and ∂ has at most $d+1$ non-zero entries per column. Let u be the number of non-zero entries of U away from the diagonal. Every column of R can be formed through a linear combination of columns of ∂. In total, the number of additions is u, and each such addition turns at most $d+1$ entries non-zero. Thus, the number of non-zero entries of R is bounded by $(d+1)n + (d+1)u = O(u)$. □

Theorem 5.12. *For a β-separable simplicial complex with a nested dissection filtration and any dimension p, Algorithm 1 runs in $O(n^{3\beta})$.*

Proof. Algorithm 1 can be implemented with hash tables to run in $O(n+\mathrm{fill}(R)+\sum_{j=1}^{n} \mathrm{fill}(U^j)\mathrm{fill}(R_j))$ time in expectation (see the full version for details). The summation of $\mathrm{fill}(U^j)\mathrm{fill}(R_j)$ terms is just the cost of an iteration of the inner loop. So, it follows that

$$\begin{aligned} O(n + \mathrm{fill}(R) + \sum_{j=1}^{n} \mathrm{fill}(U^j)\mathrm{fill}(R_j)) &= O(n + \mathrm{fill}(R) + \max_i\{\mathrm{fill}(U^i)\} \sum_{j=1}^{n} \mathrm{fill}(R_j) \\ &= O(n + \mathrm{fill}(R)(1 + \max_i\{\mathrm{fill}(U^i)\})) \end{aligned}$$

Plugging in $\mathrm{fill}(U^j) = O(n^\beta)$ and the bound for $\mathrm{fill}(R)$ from Lemma 5.11 yields the result. □

5.4 Vineyards in Separable Complexes

In the preceding section, we showed that if the input filtration corresponds to a nested dissection order, then the nested dissection analysis can be used to bound the fill-in and running-time of a simple persistence algorithm. However, one is usually interested in a fixed filtration different from the nested filtration order. In this section, we consider how to compute the persistence of $\partial := \partial_p$ with respect to an arbitrary filtration f. Assuming the simplicial complex is β-separable, and a nested filtration ordering π of the simplices is known, we show that converting a reduced matrix for a nested dissection order into a reduced matrix for any other filtration can be done in $O(n^{2+\beta})$ time. Although this is worse than the best known complexity bound of $O(n^\omega)$ [25], it leads to an algorithm that is based on elementary reductions and yields a subcubic bound for a large class of instances.

We order the rows of ∂ in f-order and the columns of ∂ in π-order. Observe that the results of the preceding section hold for any *row* order of the matrix. Therefore, using Algorithm 1, the re-ordered boundary matrix can be reduced in $O(n^{3\beta})$ time. To get the columns from π into f-order, we employ the Vineyard algorithm of Cohen-Steiner et al. [8]. In general, each transposition of simplices requires linear time. Because $O(n^2)$ transpositions might be necessary from π to f, it seems that $O(n^3)$ is the best one could hope for. However, $O(n)$ time is not necessary for many transpositions. To show this, we must define the sequence of transpositions carefully. Recall the separator tree of nested dissection from Section 5.1. Let C denote the simplices stored in some tree node, and let A, B denote the simplices stored in the left and right subtree, respectively. In π, all columns of A precede all columns of B, which, in turn, precede all columns in C. We first bring A into f-order recursively, then we bring B into f-order. Next, we bring $A \cup B$ into f-order, by transposing the columns in B with predecessors until they end in the correct position. This step only transposes A-columns with B-columns—we call this an *A-B-swap*. Finally, we bring $(A \cup B) \cup C$ into f-order by transposing the columns in C with predecessors until they are in the correct position. To bound the cost, we give the following lemma.

Lemma 5.13. *An A-B-swap takes $O(1)$ time.*

Proof. Let $i < j$ be the indices of the simplices involved in the swap, let R, U be the matrices constructed in the persistent homology algorithm for nested dissection order, such that $R = \partial U$. Because A and B are separated, columns i and j are not added during the algorithm, so $U_{ij} = 0$. Inspecting the case distinction of the Vineyard algorithm [8, Sec.3], we observe that case 1 can be disregarded because it repairs only the effect of a row swap. Since U_{ij}, the only possible cases are 2.2, 3.2, and 4, and all these cases require only constant time. □

Let $\mathrm{Cost}(n)$ denote the cost function for the transpositions with $n = |A \cup B \cup C|$. The recursive calls cost $\mathrm{Cost}(|A|) + \mathrm{Cost}(|B|)$, and the A-B-swaps cost $O(n^2)$ at most. Finally, to move a column of C to the correct position, at most $n - 1$ transpositions are needed, and each such transposition costs $O(n)$ in the worst case. Since $|C| = O(n^\beta)$, the total cost for moving the C-columns is $O(n^{2+\beta})$. Therefore, the transposition cost satisfies the recurrence

$$\mathrm{Cost}(n) = cn^{2+\beta} + \mathrm{Cost}(n_1) + \mathrm{Cost}(n_2),$$

with $n_1 + n_2 \leqslant n$ and $n_1, n_2 \leqslant \alpha n$ for some $\alpha < 1$. A simple inductive proof shows that $\mathrm{Cost}(n) \leqslant c' n^{2+\beta}$ with $c' = c/(1 - \alpha^{2+\beta} - (1-\alpha)^{2+\beta})$. We can thus summarize as follows.

Theorem 5.14. *Given a β-separable complex $\mathcal{K}$ with a nested dissection ordering π and any filtration f on $\mathcal{K}$. There is a sequence of transpositions that transforms the columns from π-order to f-order for which the Vineyard algorithm only requires $O(n^{2+\beta})$ time.*

As the complexity of the vineyard transformation dominates the complexity of persistence computation for π-order, we arrive at the following running time bound.

Theorem 5.15. *Given a β-separable complex $\mathcal{K}$ with a known nested dissection ordering π and any filtration f on $\mathcal{K}$. There is a combination of Algorithm 1 and the Vineyard algorithm which computes persistent homology in $O(n^{2+\beta})$ time.*

5.5 The Output-Sensitive Algorithm

The output-sensitive algorithm from [7] reduces persistence computation to rank queries over submatrices of the boundary matrix. We will show in this section that for β-separable complexes, nested dissection can improve the asymptotic running time for the output-sensitive algorithm. When the number of highly persistent features is small, it gives an asymptotic improvement over the algorithm in the previous section.

Persistence via rank computation. We first revisit the algorithm from [7]. Recall our notation of ∂ for the boundary matrix in fixed dimension p. Assume for simplicity that ∂ is a $m \times n$ matrix with $m \leqslant n$. For row indices $i_1 \leqslant i_2$ and column indices $j_1 \leqslant j_2$, let $\mu_{i_1,i_2}^{j_1,j_2}$ denote the number of homology classes which are born in the index range $[i_1, i_2]$ and die in the index range $[j_1, j_2]$. The key observation is that μ can be computed with four rank computations of submatrices of ∂ using the following formula.

$$\mu_{i_1,i_2}^{j_1,j_2} = \operatorname{rk}\left(\partial_{i_1,j_2}^{i_1,j_2}\right) - \operatorname{rk}\left(\partial_{i_1+1,j_1-1}^{i_1+1,j_1-1}\right) - \operatorname{rk}\left(\partial_{i_2+1,j_2}^{i_2+1,j_2}\right) + \operatorname{rk}\left(\partial_{i_2+1,j_1-1}^{i_2+1,j_1-1}\right).$$

Moreover, if $\mu > 0$, the birth-death index pairs in the range can be computed by binary search, where the cost in each iteration corresponds (asymptotically) to the cost of computing μ. Therefore, the cost to compute one persistence pair is roughly bounded by $R(n) \log n$, where n is the size of the matrix and $R(n)$ is the cost to compute the rank of ∂.

Finally, the algorithm avoids the computation of low-persistence points. This is achieved by computing $\mu_{i_1,i_2}^{j_1,j_2}$ only if the birth and death range have sufficient difference in function value. For fixed $\Gamma > 0$ and $0 < \delta < 1$, the algorithm detects all persistence pairs of persistence at least Γ in time $O((1/\delta + C_{(1-\delta)\Gamma} \log n) R(n))$, where $C_{(1-\delta)\Gamma}$ is the number of persistence pairs of persistence at least $(1-\delta)\Gamma$. The main primitive computed by the algorithm is ranks of submatrices. In [7], deterministic and randomized variants are discussed. In the deterministic case, $R(n) = O(n^{\omega})$ [20], resulting in a complexity of $O(C_{(1-\delta)\Gamma} n^{2.373})$. More recently, Yuster [31] showed that the rank of matrices with β-separable graphs ordered according to a nested dissection ordering can be computed in $O(n^{\omega\beta})$ time with high probability. This method can be combined with the Chen-Kerber algorithm to give the following result for β-separable complexes. Details may be found in the full version.

Theorem 5.16. *For a filtration on a β-separable simplicial complex of size n, there is a Monte-Carlo algorithm with fixed success probability to compute all homology classes with persistence at least Γ in time*

$$\tilde{O}(C_{(1-\delta)\Gamma}(S(n) + n^{\omega\beta})),$$

where $\delta > 0$ is a fixed constant, $C_{(1-\delta)\Gamma}$ is the number of homology classes with persistence at least $(1-\delta)\Gamma$, $S(n)$ is the complexity of computing a nested dissection order,and $\tilde{O}$ means that we ignore logarithmic factors in n.

In particular, for a β-separable complex for β is smaller than $\frac{2}{\omega} \approx 0.843$ and if Γ is chosen such that $C_{(1-\delta)\Gamma}$ is small (e.g. $O(\log n)$), the algorithm computes persistence in subquadratic time.

5.6 Discussion

We have drawn a connection between nested dissection, a method for ordering pivots in Gaussian elimination, and persistent homology. This method takes advantage of additional geometric structure in the form of separators, which occurs in a natural class of geometric complexes to improve running time. We showed that, under this condition, a simple variant of the persistence algorithm provably runs in subcubic time. There are several open problems and directions which to consider:

- Can the running time bound for persistence be improved to $O(n^{3\beta})$ or $O(n^{\omega\beta})$ on β-separable filtrations? Can a single dissection ordering be used in the output sensitive algorithm, rather than computing separators for each submatrix?
- Approximation of persistence has been shown to improve space and time bounds. If we consider the L_∞ distance on filtrations, are there a large classes of filtrations which are ε-close to nested dissection orderings? That is, can we always find a nested dissection ordering which is ε-close to any filtration, or at least a general class of filtrations?

6 Future Work

The work on applications of the work developed in TOPOSYS will continue in a number of directions. Primarily, all works in progress, e.g. TDA using Bregman divergence will continue and as results and implementations become available they will be tested on available datasets as well as on newly available datasets. In terms of robotics applications, there is also continuing work on understanding high-level structure of the configuration spaces. Many of the questions would benefit from variants of multidimensional persistence and more efficient construction of complexes in medium to high dimensions.

Finally, we have developed a fairly good understanding of efficiently computing persistent homology. There are still numerous questions, including

- Does there exist a better distributed version of persistence computation, e.g. with theoretical guarantees? The work presented in Section 5 suggests this may be possible in some cases.
- How can we construct more efficient complexes to represent the space? This is perhaps the most important implementational question in the field today. Currently, the complex construction is the main bottleneck in computation. Simplicial complexes are large and require enumeration of all simplices. This simply becomes impossible for even small to medium size high dimensional data sets. A more efficient representation must be found - perhaps as an approximation. Current approximation techniques have constants which depend on ambient dimension, whereas for practical application we would require them to depend on intrinsic dimemension.
- An efficient implementation of eigenspace persistence will the investigation of other types of proximity information, such as links. It remains an open question as to qhat kind of information this type of analysis will reveal.
- Extension of TDA to divergences, allows for a much wider application of the techniques. Many types of data are more naturally represented by distributions rather than metrics. Doing this type of analysis directly on the distributions should effectivley "boost" the signal and allow us to find more structure in data than we can with metric-based methods.
- Is there topological information available in the links in data sources such as ArXiv and Wikipedia and can we find it using the persistence of maps?

As techniques further improve, in terms of complex construction and more informative metrics, we will revisit the data sets to examine if interestign structure can be detected. Furthermore, as the algorithm for and the implementation of the persistence of maps improves, we will revisit the Topological PaeRank breifly mentioned at the end of Section .

References

[1] Noga Alon and Raphael Yuster. Matrix sparsification and nested dissection over arbitrary fields. *J. ACM*, 60(4), 2013.

[2] Jean-Daniel Boissonnat, Tamal K. Dey, and Clément Maria. The compressed annotation matrix: An efficient data structure for computing persistent cohomology. In *Algorithms - ESA*

2013 - 21st Annual European Symposium, Sophia Antipolis, France, September 2-4, 2013. Proceedings, pages 695–706, 2013.

[3] Ulrich Bauer, Michael Kerber, and Jan Reininghaus. Clear and compress: Computing persistent homology in chunks. In Peer-Timo Bremer, Ingrid Hotz, Valerio Pascucci, and Ronald Peikert, editors, *Topological Methods in Data Analysis and Visualization III*, Mathematics and Visualization, pages 103–117. Springer International Publishing, 2014.

[4] Ulrich Bauer, Michael Kerber, Jan Reininghaus, and Hubert Wagner. PHAT - persistent homology algorithms toolbox. In *Mathematical Software - ICMS 2014 - 4th International Congress, Seoul, South Korea, August 5-9, 2014. Proceedings*, pages 137–143, 2014.

[5] Gunnar Carlsson and Vin de Silva. A geometric framework for sparse matrix problems. *Advances in Applied Mathematics*, 33(1):1–25, 2004.

[6] Chao Chen and Michael Kerber. Persistent homology computation with a twist. In *27th European Workshop on Computational Geometry*, 2011.

[7] Chao Chen and Michael Kerber. An output-sensitive algorithm for persistent homology. *Comput. Geom*, 46(4):435–447, 2013.

[8] David Cohen-Steiner, Herbert Edelsbrunner, and Dmitriy Morozov. Vines and vineyards by updating persistence in linear time. In *Proceedings of the Twenty-second Annual Symposium on Computational Geometry*, SCG '06, pages 119–126, New York, NY, USA, 2006. ACM.

[9] Tamal K. Dey, Fengtao Fan, and Yusu Wang. Computing topological persistence for simplicial maps. In *30th Annual Symposium on Computational Geometry, SOCG'14, Kyoto, Japan, June 08 - 11, 2014*, page 345, 2014.

[10] Dionysus. By Dmitriy Morozov (`http://www.mrzv.org/software/dionysus/`).

[11] Vin de Silva, Dmitriy Morozov, and Mikael Vejdemo-Johansson. Dualities in persistent (co)homology. *Inverse Problems*, 27, 2011.

[12] H. Edelsbrunner and J. Harer. *Computational Topology. An Introduction.* Amer. Math. Soc., 2010.

[13] Herbert Edelsbrunner, David Letscher, and Afra Zomorodian. Topological persistence and simplification. *Discrete & Computational Geometry*, 4(28):511–533, 2002.

[14] Herbert Edelsbrunner and Salman Parsa. On the computational complexity of betti numbers: Reductions from matrix rank. In *Proceedings of the Twenty-Fifth Annual ACM-SIAM Symposium on Discrete Algorithms, SODA 2014, Portland, Oregon, USA, January 5-7, 2014*, pages 152–160, 2014.

[15] Brittany Terese Fasy, Jisu Kim, Fabrizio Lecci, and Clément Maria. Introduction to the R package TDA. *CoRR*, abs/1411.1830, 2014.

[16] Laura Grigori, Erik G. Boman, Simplice Donfack, and Timothy A. Davis. Hypergraph-based unsymmetric nested dissection ordering for sparse LU factorization. *SIAM J. Scientific Computing*, 32(6):3426–3446, 2010.

[17] Alan George. Nested dissection of a regular finite element mesh. *SIAM Journal on Numerical Analysis*, 10(2):345–363, 1973.

[18] John Russell Gilbert and Robert Endre Tarjan. The analysis of a nested dissection algorithm. *Numerische Mathematik*, 50(4):377–404, 1987.

[19] Benoît Hudson, Gary L. Miller, Steve Y. Oudot, and Donald R. Sheehy. Topological inference via meshing. In *Proceedings of the 26th ACM Symposium on Computational Geometry*, pages 277–286, 2010.

[20] Oscar H. Ibara, Shlomo Moran, and Roger Hui. A generalization of the fast LUP matrix decomposition algorithm and applications. *Journal of Algorithms*, 3:45–56, 1982.

[21] François Le Gall. Powers of tensors and fast matrix multiplication. In *Proceedings of the 39th International Symposium on Symbolic and Algebraic Computation*, ISSAC '14, pages 296–303, New York, NY, USA, 2014. ACM.

[22] Richard J. Lipton, Donald J. Rose, and Robert Endre Tarjan. Generalized nested dissection. *SIAM Journal on Numerical Analysis*, 16(2):346–358, 1979.

[23] Richard J. Lipton and Robert Endre Tarjan. A separator theorem for planar graphs. *SIAM J. Appl. Math*, 36:177–189, 1979.

[24] Clément Maria, Jean-Daniel Boissonnat, Marc Glisse, and Mariette Yvinec. The gudhi library: Simplicial complexes and persistent homology. In *Mathematical Software - ICMS 2014 - 4th International Congress, Seoul, South Korea, August 5-9, 2014. Proceedings*, pages 167–174, https://project.inria.fr/gudhi/software/, 2014.

[25] Nikola Milosavljević, Dmitriy Morozov, and Primoz Skraba. Zigzag persistent homology in matrix multiplication time. In *Proceedings of the Twenty-seventh Annual Symposium on Computational Geometry*, SoCG '11, pages 216–225, New York, NY, USA, 2011. ACM.

[26] Dmitriy Morozov. Persistence algorithm takes cubic time in worst case. *BioGeometry News, Dept. Comput. Sci., Duke Univ*, 2005.

[27] Gary L. Miller, Shang-Hua Teng, William Thurston, and Stephen A. Vavasis. Geometric separators for finite-element meshes. *SIAM J. Sci. Comput.*, 19(2):364–386, 1998.

[28] Donald J. Rose, R. Endre Tarjan, and George S. Lueker. Algorithmic aspects of vertex elimination on graphs. *SIAM J. Comput.*, 5(2):266–283, 1976.

[29] Donald R. Sheehy. *Mesh Generation and Geometric Persistent Homology*. PhD thesis, Carnegie Mellon University, 2011.

[30] Mihalis Yannakakis. Computing the minimum fill-in is np-complete. *SIAM Journal on Algebraic Discrete Methods*, 2(1):77–79, 1981.

[31] Raphael Yuster. Matrix sparsification for rank and determinant computations via nested dissection. In *FOCS: IEEE Symposium on Foundations of Computer Science*, pages 137–145, 2008.

[32] Arindam Banerjee, Srujana Merugu, Inderjit S. Dhillon, and Joydeep Ghosh. Clustering with Bregman Divergences. *J. Mach. Learn. Res.* **6**, 2005, 1705–1749.

[33] G. Carlsson. Topology and data. Bulletin of the American Mathematical Society 46.2 (2009): 255-308.

[34] J. M. Chan, G. Carlsson and R. Rabadan. Topology of viral evolution. Proceedings of the National Academy of Sciences 110.46 (2013): 18566-18571.

[35] P. Dlotko. Persistence Landscapes Toolbox (`www.math.upenn.edu/dlotko`) (2014).

[36] V. Nanda. Perseus (`www.sas.upenn.edu/?vnanda/perseus`) (2014).

[37] D. Paolotti et al. Web?based participatory surveillance of infectious diseases: the Influenzanet participatory surveillance experience. Clinical Microbiology and Infection 20.1 (2014): 17-21.

[38] J. A. Perea and J. Harer. Sliding windows and persistence: An application of topological methods to signal analysis. Foundations of Computational Mathematics 15.3 (2013): 799-838.

[39] J. A. Perea et al. SW1PerS: Sliding windows and 1-persistence scoring; discovering periodicity in gene expression time series data. BMC bioinformatics 16.1 (2015): 257.

[40] J. Pita Costa and P. ?kraba. A topological data analysis approach to epidemiology. European Conference of Complexity Science (2014).

[41] J. Pita Costa and P. ?kraba. Topological epidemiological data analysis. ACML Health (2015).

]

[42] E. Yom?Tov et al. Estimating the Secondary Attack Rate and Serial Interval of Influenza?like Illnesses using Social Media. Influenza and other respiratory viruses. DOI:10.1111/irv.12321 (2015)

[43] U. Bauer and H. Edelsbrunner. The Morse theory of Čech and Delaunay complexes. Manuscript, IST Austria, Klosterneuburg, Austria, 2014.

[44] K. Borsuk. On the imbedding of systems of compacta in simplicial complexes. *Fund. Math.* **35** (1948), 217–234.

[45] L. M. Bregman. The relaxation method of finding the common point of convex sets and its applications to the solution of problems in convex programming. *USSR Comput. Math. and Math. Phys.* **7** (1967), 200–217.

[46] M. M. Cohen. *A Course in Simple-Homotopy Theory.* Graduate Texts in Mathematics **10**, Springer, New York, New York, 1973.

[47] H. Edelsbrunner and J. L. Harer. *Computational Topology. An Introduction.* Amer. Math. Soc., Providence, Rhode Island, 2010.

[48] H. Edelsbrunner, G. Jabłonski and M. Mrozek. The persistent homology of a self-map. *Found. Comput. Math.*, to appear.

[49] H. Edelsbrunner, D. Letscher and A. Zomorodian. Topological persistence and simplification. *Discrete Comput. Geom.* **28** (2002), 511–533.

[50] R. Forman. Morse theory for cell complexes. *Adv. Math.* **134** (1998), 90–145.

[51] F. Nielsen, J.-D. Boissonnat and R. Nock. Bregman Voronoi diagrams: properties, algorithms and applications. arXiv:0709.2196v1, 2007.

[52] F. Nielsen and R. Nock. On the smallest enclosing information disk. *In* "Proc. 18th Canad. Conf. Comput. Geom., 2006", [[pages?]]

[53] K. Onishi and H. Imai. Voronoi diagrams in statistical parametric space by Kullback-Leibler convergence. *In* "Proc. 13th Ann. Sympos. Comput. Geom., 1997", 463–465.

[54] K. Sadakane, H. Imai, K. Onishi, M. Inaba, F. Takeuchi and K. Imai. Voronoi diagrams by divergence with additive weights. *In* "Proc. 14th Ann. Sympos. Comput. Geom., 1998", 403–404.

[55] G. Salton, A. Wong and C. S. Yang, A vector space model for automatic indexing. *Comm. ACM* **18** (1975), 613-620.

[56] E. Welzl. Smallest enclosing disks (balls and ellipsoids). In *New Results and New Trends in Computer Science*, H. A. Maurer (ed.), Springer, LNCS **555** (1991), 359–370.