Comparing Directed and Weighted Road Maps

Alyson Bittner, Brittany Terese Fasy, Maia Grudzien, Sayonita Ghosh Hajra, Jici Huang, Kristine Pelatt, Courtney Thatcher, Altansuren Tumurbaatar, and Carola Wenk

Abstract With the increasing availability of GPS trajectory data, map construction algorithms have been developed that automatically construct road maps from this data. In order to assess the quality of such (constructed) road maps, the need for meaningful road map comparison algorithms becomes increasingly important.

A. Bittner
Department of Mathematics, University at Buffalo (SUNY), Buffalo, NY, USA
e-mail: alysonbi@buffalo.edu

B. T. Fasy (✉)
Gianforte School of Computing and Department of Mathematical Sciences, Montana State University, Bozeman, MT, USA
e-mail: brittany@cs.montana.edu

M. Grudzien
School of Computing, Montana State University, Bozeman, MT, USA

S. Ghosh Hajra
Department of Mathematics, Hamline University, St Paul, MN, USA
e-mail: sghoshhajra01@hamline.edu

J. Huang
Gianforte School of Computing, Montana State University, Bozeman, MT, USA
e-mail: jici.huang@msu.montana.edu

K. Pelatt
Department of Mathematics, St. Catherine University, St Paul, MN, USA
e-mail: kepelatt@stkate.edu

C. Thatcher
Department of Mathematics and Computer Science, University of Puget Sound, Tacoma, WA, USA
e-mail: cthatcher@pugetsound.edu

A. Tumurbaatar
Department of Mathematics and Statistics, Washington State University, Pullman, WA, USA
e-mail: altaa.tumurbaatar@wsu.edu

C. Wenk
Department of Computer Science, Tulane University, New Orleans, LA, USA
e-mail: cwenk@tulane.edu

E. W. Chambers et al. (eds.), *Research in Computational Topology*, Association for Women in Mathematics Series 13, https://doi.org/10.1007/978-3-319-89593-2_4

Indeed, different approaches for map comparison have been recently proposed; however, most of these approaches assume that the road maps are modeled as undirected embedded planar graphs.

In this paper, we study map comparison algorithms for more realistic models of road maps: directed roads as well as weighted roads. In particular, we address two main questions: how close are the graphs to each other, and how close is the information presented by the graphs (i.e., traffic times, trajectories, and road type)? We propose new road network comparisons and give illustrative examples. Furthermore, our approaches do not only apply to road maps but can be used to compare other kinds of graphs as well.

1 Introduction

Road maps have traditionally been constructed by performing costly and time-consuming land surveys. With increasing availability of GPS trajectory data (e.g., as in [25]), map construction algorithms have been developed that automatically construct road maps from such data sources. In order to assess the quality of such (constructed) road maps, meaningful comparison of different road maps is important. Indeed, different approaches for map comparison have been recently proposed in the literature [1, 2, 7, 20]; however, most of these approaches assume that the road maps are modeled as undirected embedded graphs.

In this paper, we use topology to inform graph comparison. As higher-ordered topological features are not present in graphs and connectivity of graphs is well-studied, we focus on the role of one-cycles (i.e., loops) in graphs. One-cycles play a fundamental role in the combinatorics of the road network (e.g., in defining the options available for navigation). In some cases, we may focus on the one-cycles defined by city blocks; that is, loops in planar embedded graphs that contain no graph edges inside the loop. More generally, we can allow for any set of (directed) one-cycles.

After presenting related work in Sect. 2, we investigate several suitable filtrations on graphs in Sect. 3, and we present our approach for comparing two different graphs in Sect. 4. Our goal is to be able to determine how close the graphs are to each other and to quantify the similarity of information presented in the graphs (such as traffic times or road type).

2 Related Work

When considering graph comparison from a graph-theoretical point of view, the obvious approaches to model the problem are NP-hard, such as subgraph isomorphism or graph edit distance [11, 14, 26]. However, these generally require one-to-one mappings of the graphs and do not take any geometric embedding into

account. For comparing road maps, distance measures have been proposed that generally model the maps as embedded undirected graphs; see [3, 4, 7] for surveys as well as experimental comparisons and quality assessments. The (directed or undirected) Hausdorff distance can be used to compare the sets of points covered by each graph [5], but this does not take any structural information of the graphs into account. Other distance measures have been proposed that compare random samples of shortest paths [20, 22] or all paths [2] in the street maps. In order to compare more general topological information, Biagioni and Eriksson developed a sampling-based distance measure [7] and Ahmed et al. introduced the local persistent homology distance [1]. Still, few of these algorithms have theoretical guarantees, and most do not deal with weighted or directed graphs. This paper fills the gap by proposing ways we can compare weighted and directed graphs using topological information found via the intrinsic distance measure in the graph. The information extracted can be local or global in nature; see the swatch filtration in Sect. 3.1 and the AMD filtration in Sect. 3.2, respectively.

2.1 Weighted and Directed Graph Comparison

Any graph comparison approach has to map features from one graph to the other in order to define a distance measure. An example is subgraph isomorphism, which is NP-hard in general but can be solved in polynomial time for planar graphs [16]. A connectivity-based dissimilarity measure is used in [30], where the authors generalize a standard distance metric in graph theory to the context of directed weighted graphs. In the case where there is a bijection ψ between the nodes in the two graphs G_1 and G_2, one can define a metric dependent on ψ by comparing the distance between nodes i, j in G_1 and nodes $\psi(i), \psi(j)$ in G_2. Explicitly, this metric, called the *total distortion*, is defined as follows:

$$d_\psi(G_1, G_2) = \Sigma_{i,j \in G_1} |d_{G_1}(i, j) - d_{G_2}(\psi(i), \psi(j))|$$

To obtain a distance between the graphs not dependent on the bijection, we consider the minimum distance over all bijections ψ:

$$\nu(G_1, G_2) = \min_\psi d_\psi(G_1, G_2)$$

Since these paths are now taking into account the distance as defined in the directed weighted graph setting, this metric encodes that data as well.

Another approach is to use random walks, as in [21, 29], where two graphs G_1 and G_2 are compared by computing random walks of a fixed length in the product space $G_1 \times G_2$, and then computing the so-called *label sequence kernel* (here, labels are referring to the edge labels), which can formally be expressed as:

$$\kappa(G_1, G_2) := \sum_{h_1} \sum_{h_2} \kappa(h_1, h_2) p(h_1|G_1) p(h'|G'),$$

where $\kappa(h_1, h_2)$ is an inner-product kernel between weighted edges h_1 and h_2 in G_1 and G_2, and $p(h|G)$ is the probability of h being a random walk in G.

2.2 Persistence-Based Comparisons

Persistent homology is a tool that describes the structure of a point cloud at multiple scales (more generally, of a filtered topological space). Moreover, the persistence diagram (a multi-set of birth–death pairs) can be computed in matrix multiplication time using basic linear algebra. Next, we briefly introduce the relevant concepts. For more details on the fundamentals of persistent homology, we refer the reader to [8, 10, 15, 17, 23], and to [19, 24] for algebraic topology.

Intuitively, the zero- and one-dimensional homology groups, denoted as $\mathsf{H}_0(\mathbb{X})$ and $\mathsf{H}_1(\mathbb{X})$, respectively, correspond to path-connected components and loops of $\mathbb{X}$. (In this paper, we are not interested in two- and higher-dimensional homology, as the data we consider are graphs). In some examples, we work with the *relative homology* groups $\mathsf{H}_k(\mathbb{X}, \mathbb{A})$ (for $\mathbb{A} \subseteq \mathbb{X}$), which is equivalent to $H_k(\mathbb{X}/\mathbb{A})$ when $\mathbb{X}$ is a simplicial complex and $\mathbb{A}$ is a subcomplex of $\mathbb{X}$ [19, Prop. A5]. The common setting for persistent homology is when we have a finite sequence of topological spaces connected by inclusions, called a discrete *filtration*. If $\mathbb{X}$ is embedded in $\mathbb{R}^2$, we could, for example, use the height filtration where $\mathbb{X}_t = \{(x, y) \in \mathbb{X}\ s.t.\ y \leq t\}$. Here, t is the filtration parameter and is colloquially referred to as *time*. Applying the homology functor to this sequence, we obtain a sequence of vector spaces connected by homomorphisms $f_k^{i,j} : \mathsf{H}_k(\mathbb{X}_i) \to \mathsf{H}_k(\mathbb{X}_j)$ for $i \leq j$. Persistent homology tracks the so-called *birth* and *death* events of $\mathbb{X}_t$ as t ranges from $-\infty$ to ∞. These birth and death times can be paired, resulting in a set of birth–death pairs that can be plotted in $\mathbb{R}^2$ in a *persistence diagram*, or plotted as stacked intervals in a *barcode plot*.

For two diagrams $\mathcal{D}_1$ and $\mathcal{D}_2$, a well-defined and commonly used distance is the *bottleneck distance*:

$$W_\infty(\mathcal{D}_1, \mathcal{D}_2) := \inf_{f:\ \mathcal{D}_1 \to \mathcal{D}_2} \sup_{x \in \mathcal{D}_1} ||x - f(x)||, \tag{1}$$

where f is a bijection between $\mathcal{D}_1$ and $\mathcal{D}_2$ [15, Ch. VII]. This distance between diagrams is Lipschitz-stable both for general filtrations [9, 13] and for the local homology filtrations [1, 6]. More generally, one may use the Wasserstein distance between diagrams, $W_p^p(\mathcal{D}_1, \mathcal{D}_2) := \inf_f \sum_{x \in \mathcal{D}_1} ||x - f(x)||^p$, under which the space of persistence diagrams is complete and separable but not a Hilbert space (which is needed for many machine learning and statistical tools).

For graphs (or any subset of a metric space), one approach is to utilize local persistent homology to generate a distance metric [1]. The *local persistent homology distance* computes bottleneck distance between the local persistence diagrams of G and G' over a common finite open cover $\mathcal{U} = \{U_1, U_2, \ldots, U_n\}$ of the domain of interest (e.g., over disks of a fixed radius centered at lattice points) [1]. Here, the local persistence diagram over U_i, denoted $\mathcal{D}(G, U_i)$, corresponds to the persistence module $\{H_k(G_t, G_t - U_i)\}_t$ and captures the local homology of the road networks as seen from the set U_i. A measure of distance is obtained by taking the weighted average of local distances over $\mathcal{U}$:

$$d_{lph}(G, G') := \sum_{U \in \mathcal{U}} \omega_U W_\infty(\mathcal{D}(G, U), \mathcal{D}(G', U)),$$

where ω_U is a weight function. In Sect. 4.1, we extend this definition to a set of paired persistence diagrams.

Recently, the TDA community has made progress in using a persistence-based approach for comparing weighted digraphs. In [28], Turner defined two filtrations of directed graphs. In particular, geodesic paths are used to define an *ordered tuple (OT) filtration*. Persistent homology with the OT filtration has the same persistent homology as the Rips filtration, when the underlying graph is undirected. In another setting, directed graph homology introduced in [18] has been extended to the persistence setting in [12].

For the comparison of weighted graphs, we introduce the idea to use correspondence between cycles. When comparing different data on the same graph, constructing such correspondences is trivial. In applications that compare road maps, correspondences may be obtained via mappings between city blocks or landmarks. While, as a general theoretical problem, constructing the best correspondence might be difficult, this work is motivated by a special setting: detecting and quantifying changes in road networks. In this case, the graph changes slowly over time in predictable ways (roads opening / closing is common, but roads moving or shifting is uncommon); thus, most underlying road network cycles stay the same.

3 Filtrations

In this section, we introduce three different filtrations from which we can compute persistence diagrams in order to compare two directed and / or weighted graphs in Sect. 4. Some of the filtrations below require an additional parameter choice (e.g., choice of basepoint), which has some choice involved; see Sect. 5.

3.1 Swatch Filtration

Let $G = (V, E)$ be a graph with weight function $f : E \to \mathbb{R}_{\geq 0}$ defined over the edges. To define the *swatch filtration*, we start with a choice of initial vertex (or root) $v \in V$ and define $G_t = G_t(f, v)$ to be the subgraph of G induced by the set of all (possibly directed) paths in G with initial vertex v and length less than or equal to t. The length of a (directed) path $p = (e_1, \ldots, e_n)$ in G with respect to f is defined to be $\ell_f(p) := \Sigma_{i=1}^n f(e_i)$. If G is a directed graph without an accompanying function, we assume that the unit weights and so the length of a directed path is simply the number of edges in that path. We note here that these subgraphs G_t are the so-called *swatches* defined in [27]; we propose using the radius threshold t as a persistence

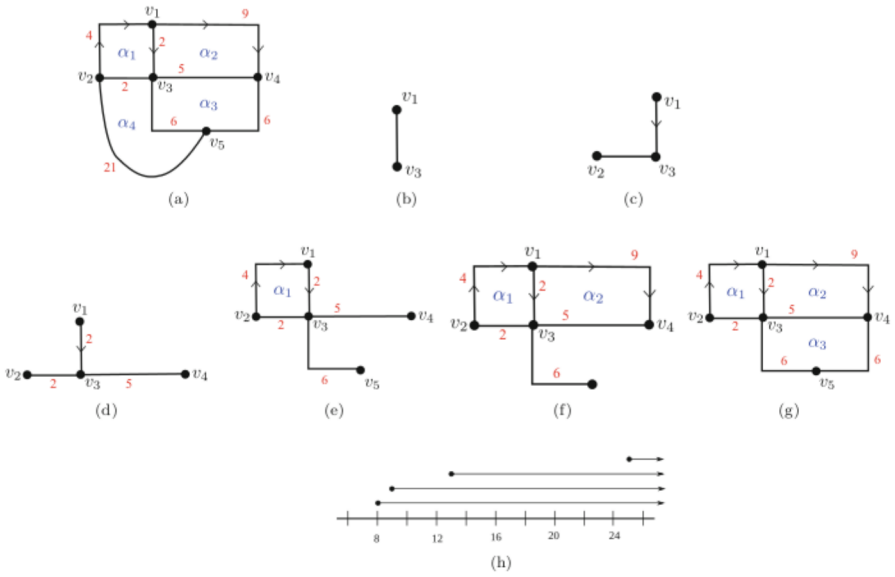

Fig. 1 Example of a swatch filtration of the directed weighted graph G using v_1 as the root node. The faces of the planar graph are labeled with the cycle that bounds the face. Notice that $\cup G_t = G$ for $t > 25$, as the set of all paths of length at most 25 covers G. (**a**) Weighted digraph G. (**b**) $\cup G_t$ for $2 < t < 4$. (**c**) $\cup G_t$ for $4 < t < 7$. (**d**) $\cup G_t$ for $7 < t < 8$. (**e**) $\cup G_t$ for $8 < t < 9$. (**f**) $\cup G_t$ for $9 < t < 13$. (**g**) $\cup G_t$ for $13 < t < 25$. (**h**) Resulting barcode for G

parameter in order to define a persistence barcode for each $v \in V$. We note that this diverges from [27], where histograms of swatches of a fixed radius (called *clothes*) were used to compare dynamical processes.

Example 1 (Swatch Filtration) In Fig. 1, we show an example of the swatch filtration with the choice of v_1 as the initial vertex. Notice that the homology of this graph is generated by the following cycles: α_1, α_2, α_3, and α_4. These cycles are minimal in this embedding, as each has an empty interior. In subfigures (a)–(g), we see the union of paths in G_t, denoted by $\cup G_t$. The first value of t for which the cycle α_1 appears is $t = 8$, so the birth time $b(\alpha_1) = 8$, as shown in the barcode in Fig. 1h. Notice that since no two-cells are introduced, no one-cycles ever die in this filtration.

The swatch diagram has two potential use-cases. First, a natural choice for the initial vertex v_1 might exist (e.g., the vertex represents a landmark such as the main train station or a baseball stadium). In this case, the swatch diagram can be used to compare the road networks *from the perspective of the landmark*. On the other hand, a choice of a landmark is not always clear, and different landmarks can create vastly different swatch filtrations; see Example 2.

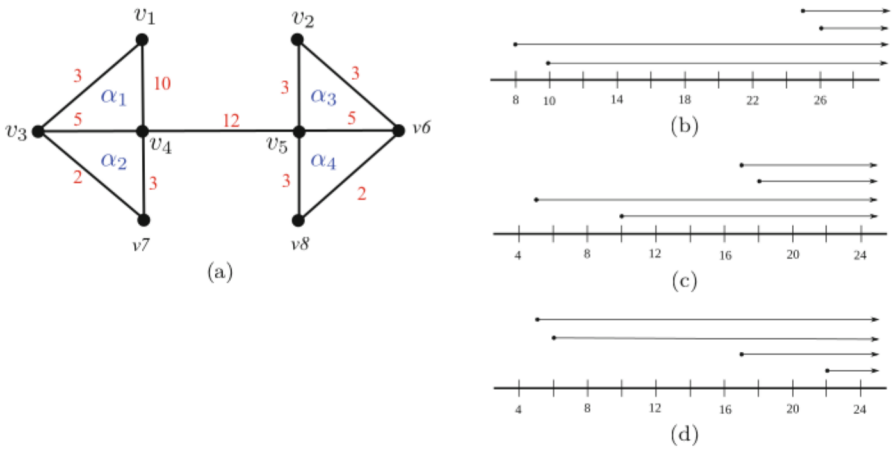

Fig. 2 The swatch filtrations constructed from different initial vertices (basepoints) have different barcodes. The edges are not drawn to scale. (**a**) Undirected graph G with edge weights. (**b**) Barcode using v_1 as initial vertex. (**c**) Barcode using v_4 as initial vertex. (**d**) Barcode using v_5 as initial vertex

Example 2 (Different Basepoints) The birth times of the cycles depend on the choice of initial vertex, which means that the distance based on the swatch filtration is sensitive to choice of basepoint. In this example, we calculate the birth times for the cycles in Fig. 2, starting at three different vertices: v_1, v_4, and v_5, resulting in three different barcodes.

3.2 Average Minimum Distance

To avoid the arbitrary choice of an initial vertex as we encountered above, we consider a second filtration using the average minimum distance (AMD) to each of the other vertices.

Definition 1 (Average Minimum Distance) The graph geodesic distance (or the minimum distance) from vertex v_i to vertex v_j is defined as: $d(v_i, v_j) := \min_p\{\ell(p)\}$, where p ranges over all possible (directed) paths between v_i and v_j. The degree-k AMD, $\text{AMD}_k \colon V \to \mathbb{R}$, of vertex v_i is the average of the minimum distances to all of the other vertices:

$$\text{AMD}_k^k(v_i) := \frac{1}{|V|-1} \sum_{v_j \neq v_i} (d(v_i, v_j))^k.$$

We define G_0 to be the empty set and G_t to be the subgraph of G induced by the set of all vertices of AMD less than or equal to t.

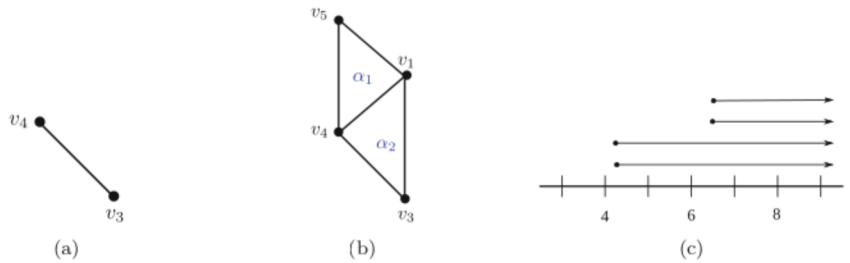

Fig. 3 Example of an average minimum distance (AMD) filtration for the graph found in Fig. 1. We calculate the AMD for v_1 as $(7+3+3+4)/4 = 4.25$, so v_1 appears in G_t when $t \geq 4.25$. Likewise, we have $\text{AMD}_1(v_2) = 6.5$, $\text{AMD}_1(v_3) = 3.0$, $\text{AMD}_1(v_4) = 3.0$, and $\text{AMD}_1(v_5) = 4.25$. Notice that $G_t = G$ for all $t > 6.5$. (**a**) G_t for $3 < t < 4.25$. (**b**) G_t for $4.25 < t < 6.5$. (**c**) Resulting barcode

We note here that AMD_k is related to the eccentricity function:

$$\text{ECC}(v_i) := \frac{1}{|V|-1} \max_{v_j \neq v_i} d(v_i, v_j) = \lim_{k \to \infty} \text{AMD}_k(v_i).$$

In other words, the eccentricity of vertex v_i is the maximum geodesic distance from v_i to any other vertex in the graph. On the other hand, the function $\text{AMD}_k(v_i)$ is the weighted average of the minimum distances to every other vertex in the graph, and limits to the eccentricity.

Example 3 (AMD Filtration) In Fig. 3, we use the AMD filtration of the graph G from Fig. 1. The AMDs of the vertices in G are $\text{AMD}_1(v_1) = 5.25$, $\text{AMD}_1(v_2) = 5.25$, $\text{AMD}_1(v_3) = 4.75$, $\text{AMD}_1(v_4) = 7.25$, and $\text{AMD}_1(v_5) = 8$.

3.3 Killing Cycles

The filtrations described above do not include higher-dimensional cells; hence, every cycle *lives forever*. In order to decrease the complexity of the homology groups, we need to either delete edges or add two-cells. We construct a cone vertex c at time zero with an edge connecting it to the initial vertex for the swatch filtration (or the first vertex added for the AMD filtration). Then, for each edge $e = (v_i, v_j) \in V \times V$, we add the (unoriented) edges (c, v_i) and (c, v_j) along with the triangle $\Delta = (c, v_i, v_j)$ at the parameter value $\max\{c\, \ell_f(e), \text{appear}(e)\}$, for some constant $c \in R_{\geq 0}$ and where $\text{appear}(e)$ is the first parameter containing e in the filtration. So, the additional simplices we add at parameter t in the filtration are:

$$C_t = \{c\} \cup \{(c, v), (c, w), (c, v, w) \mid (v, w) \in G_t \text{ and } c\, \ell_f(v, w) \leq t\}.$$

Thus, we are considering the filtrations of the form $\{\overline{G}_t := G_t \cup C_t\}$, where G_t is either the swatch or AMD filtration. The effect of adding these simplices is as follows: In the swatch filtration, cycles must be longer for larger values of t in order to be seen from the initial vertex, which intuitively makes sense. In the AMD filtration, the lengths of the cycles in $\overline{G}_t$ will be at least $\frac{2t}{c}$.

4 Comparing Data on Graphs

Given two road networks as directed graphs, both embedded in a compact subset of $\mathbb{R}^2$, we want to define a topology-based distance metric between the two networks. See Sect. 2 for related work and a discussion on the difficulty of this problem. We start with a simplified problem: to develop a method for measuring distance between a graph $G = (V, E)$ with two different annotations, such as two different weight functions on the edges $f, g : E \to \mathbb{R}$ or two different directed graphs with the same underlying undirected graph.

4.1 Paired Analysis

For each $v \in V$, we compute two persistence diagrams $\mathcal{D}_v(f)$ and $\mathcal{D}_v(g)$, corresponding to annotations f and g, respectively. Thus, we can mirror the approach of [1] for aggregating paired diagrams (see Sect. 2) in order to compute a distance between annotations f and g:

$$d_{\text{pair}}(f, g) := \sum_{v \in V} \omega_v W_\infty \left(\mathcal{D}_v(f), \mathcal{D}_v(g)\right),$$

where ω_v is a weight function on the vertices of G, assumed to be the uniform weight $1/|V|$, unless otherwise stated. Since the underlying graphs are exactly the same, we can construct an even more sensitive distance between filtrations $G_t(f, v)$ and $G_t(g, v)$, which we describe next.

4.2 Birth–Birth Diagrams

Using the swatch and AMD filtrations from the previous section, we notice that these filtrations depend only on the intrinsic information on the graphs, whether this information is given by functions (such as f, g) or by direction assignments to the edges. Furthermore, each complex in the swatch and AMD filtrations is a subgraph of G, and all one-cycles have infinite persistence. Rather than looking at persistent homology as the traditional birth–death diagram, we designate a special subset of

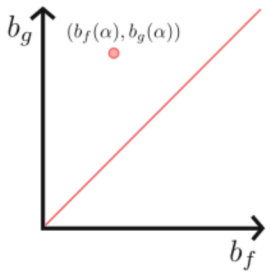

Fig. 4 A birth–birth diagram

cycles $A \subseteq H_1(G)$ of interest (e.g., A could be a set of generating cycles). For each cycle in A, we have two birth times (one for each filtration). We use the two birth times of the cycles to develop a distance measure between the two different annotations, and we visualize this information in a birth–birth diagram; see Fig. 4.

More formally, let $G = (V, E)$ be a directed graph, and let $f, g \colon E \to \mathbb{R}_{\geq 0}$ be two annotations that we wish to compare. We filter the paths p in G by this length $\ell_f(p)$ (or, alternatively, by one of the filtrations defined in the previous section), and say that the minimum distance necessary to complete a cycle $\alpha = (v_1, \ldots, v_n)$ is computed by considering all possible ways to break the cycle into two connected subpaths. More explicitly, we compute this minimum as:

$$b_f(\alpha) := \min_{v_i, v_j \in \alpha} \left\{ \max\{\ell_f(\alpha_{i,j}), \ell_f(\alpha_{j,i})\} \right\},$$

where $\alpha_{i,j}$ is the subpath of α from v_i to v_j. (Using either the swatch or AMD filtration, we can define $b_f(\alpha)$ to be the minimum parameter t such that $\alpha \in G_t$.) We note that $b_f(\alpha)$ is well-defined.

To compare the intrinsic information on a graph G given two annotations $f, g \colon E \to \mathbb{R}$, we begin by choosing a subset of cycles $A \subseteq H_1(G)$; for example, we could use a set of generating cycles. For each $\alpha \in A$, we compute the birth times $b_f(\alpha)$ and $b_g(\alpha)$. Thus, we have two paired sets of birth times for each cycle in A. We visualize this information in what we call the *birth–birth diagram*, where the birth for the filtration corresponding to f is the x-coordinate and for g the y-coordinate; see Fig. 4. We then define the distance between the two annotated graphs to be the total L_∞-distance between points in the birth–birth diagram and the diagonal $y = x$; in other words, we compute

$$D\left((G, f), (G, g)\right) = \sum_{\alpha_i \in A} ||b_f(\alpha_i) - b_g(\alpha_i))||_\infty.$$

We note here that sometimes it may be convenient to first normalize the parameters so that $t = 1$ corresponds to the end of both filtrations.

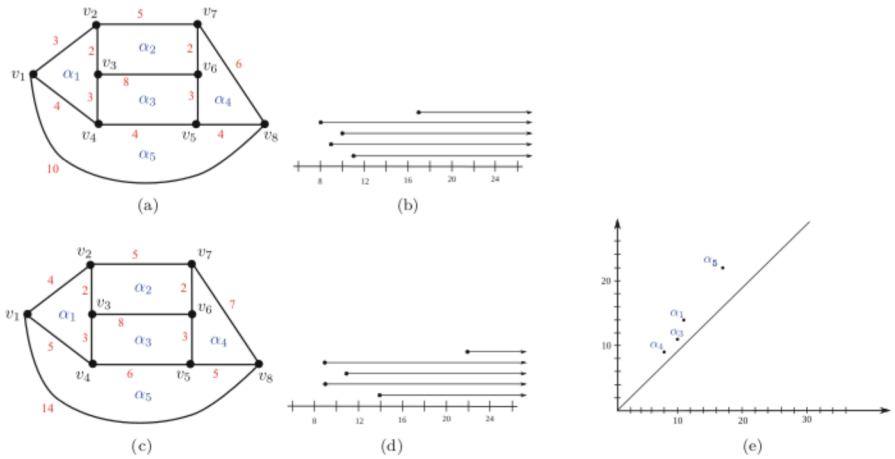

Fig. 5 Example of two swatch filtrations derived from two different weights on the same underlying graph. Here, we use v_6 as the root vertex to construct both filtrations. The distance between these two graphs is $10 = 3 + 0 + 1 + 1 + 5$. (**a**) Weighted graph G. (**b**) Resulting barcode for G. (**c**) Weighted graph G'. (**d**) Resulting barcode for G'. (**e**) Birth–birth diagram

Notice that this construction considers a single graph with two different annotations. These techniques can be extended to analyze two annotations on different graphs (G, f) and (G', f'). To do so, rather than starting with $A \subseteq H_1(G)$, we start with a correspondence $C \subset H_1(G) \times H_1(G')$. Given such a correspondence between cycles, we proceed as above to plot a point in the birth–birth diagram for every $(\alpha, \alpha') \in C$.

4.3 Examples

This section provides enhanced examples to clarify some of the concepts presented above.

Example 4 (Birth–Birth Diagram for Two Weighted Graphs) In Fig. 5, we give an example of two swatch filtrations from the same undirected graph but with different weight functions, and the corresponding birth–birth diagram. Here, the filtrations use v_6 as the root vertex.

Example 5 (Birth–Birth Diagram for Two Directed Graphs) In Fig. 6, we give an example of two filtrations from the same underlying graph with the direction of one edge changed, and the corresponding birth–birth diagram.

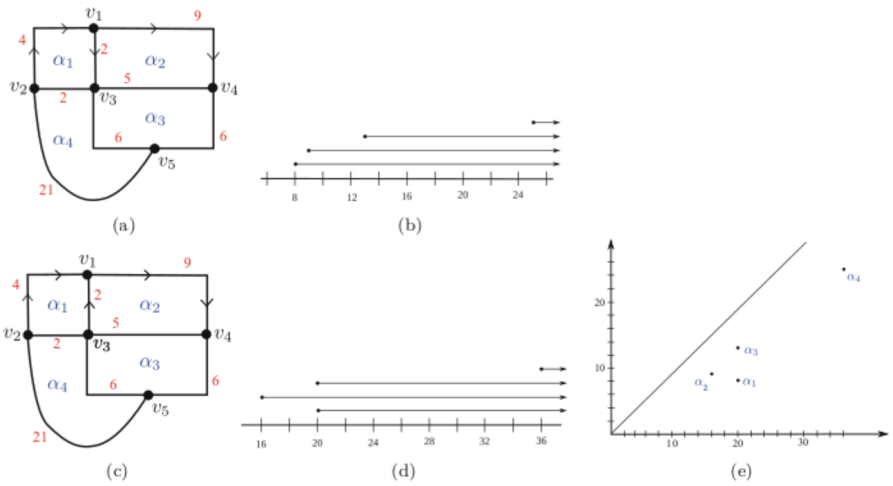

Fig. 6 Example of two swatch filtrations starting at v_1 on two weighted directed graphs, and the resulting birth–birth diagram. The graphs G and G' are identical, except for the direction of the edge (v_1, v_3). (**a**) Weighted digraph G. (**b**) Resulting barcode for G. (**c**) Weighted digraph G'. (**d**) Resulting barcode for G'. (**e**) Birth–birth diagram

5 Discussion

In this paper, we present a framework for comparing weighted and directed graphs. We have introduced new filtrations for directed graphs that capture the complexity of potential paths in a graph by focusing on cycles in the graph. We note that the set of filtrations presented is not exhaustive but is a practical starting point for using topological techniques for comparing weighted directed graphs. Future work will provide an analysis of these techniques on real data sets.

The swatch filtration presented has a persistence module corresponding to each vertex. In addition, if we allow for the killing cycles enhancement (as described in Sect. 3.3), we have an additional parameter c to account for. The parameter c can be optimally chosen by cross-validation; however, a single vertex might not be able to reveal all necessary information through its corresponding swatch filtration. So, we suggest randomly sampling the vertices to have a basepoint set when the set of vertices is too large to consider all possibilities.

The contribution of this paper that we suspect will have the most impact is the birth–birth diagram, which can be used in settings outside of directed, weighted graphs. In fact, it can be applied in any setting where we have multiple functions defined over the same domain (e.g., temperatures at different times, observations before and after a stimulus in a scientific experiment, etc.). Of course, in other settings, the features of interest we wish to match might not be one-cycles.

We note that the theory presented here is not limited to these filtrations. For example, we ask: what topological properties of a given directed graph embedded

in $\mathbb{R}^2$ during the filtration can be uncovered? Can we incorporate the embedding rather than just the intrinsic distances? Moreover, empirical study of these filtrations will determine if they can be used in practice to compare different weighted (directed) graphs. As a first step, we will investigate how these methods can be used to compare traffic patterns across large timescales, where the road networks might change over time (bridges added, new exits on highways, etc.). We also note that the assumptions do not require the graphs to be planar; hence, they provide an obvious advantage over most of the existing methods to compare road networks. Analyzing this advantage will be part of our future research.

Acknowledgements This paper is the product of a working group of WinCompTop 2016, sponsored by NSF DMS 1619908, Microsoft Research, and the Institute for Mathematics and Its Applications (IMA) in Minneapolis, MN. In addition, part of this research was conducted under NSF CCF 618605 (Fasy) and NSF CCF 1618469 (Wenk).

References

1. M. Ahmed, B.T. Fasy, C. Wenk, Local persistent homology based distance between maps, in *Proceedings of 22nd ACM SIGSPATIAL International Conference on Advances in Geographic Information Systems* (ACM, New York, 2014), pp. 43–52
2. M. Ahmed, B.T. Fasy, K.S. Hickmann, C. Wenk, Path-based distance for street map comparison. ACM Trans. Spatial Algorithms Syst. **1**, article 3, 28 pages (2015)
3. M. Ahmed, S. Karagiorgou, D. Pfoser, C. Wenk, A comparison and evaluation of map construction algorithms using vehicle tracking data. GeoInformatica **19**(3), 601–632 (2015)
4. M. Ahmed, S. Karagiorgou, D. Pfoser, C. Wenk, *Map Construction Algorithms* (Springer, Berlin, 2015)
5. H. Alt, L.J. Guibas, Discrete geometric shapes: matching, interpolation, and approximation - a survey, in *Handbook of Computational Geometry*, ed. by J.-R. Sack, J. Urrutia (Elsevier, North-Holland, 1999), pp. 121–154
6. P. Bendich, E. Gasparovic, J. Harer, R. Izmailov, L. Ness, Multi-scale local shape analysis for feature selection in machine learning applications, in *Proceedings of International Joint Conference on Neural Networks* (2015)
7. J. Biagioni, J. Eriksson, Inferring road maps from global positioning system traces: survey and comparative evaluation. Transp. Res. Rec. J. Transp. Res. Board **2291**, 61–71 (2012)
8. G. Carlsson, Topology and data. Bull. Am. Math. Soc. **46**(2), 255–308 (2009)
9. F. Chazal, D. Cohen-Steiner, M. Glisse, L.J. Guibas, S.Y. Oudot, Proximity of persistence modules and their diagrams, in *Proceedings of 25th Annual Symposium on Computational Geometry* (2009), pp. 237–246
10. F. Chazal, V. De Silva, M. Glisse, S. Oudot, *The Structure and Stability of Persistence Modules* (Springer, Berlin, 2016)
11. O. Cheong, J. Gudmundsson, H.-S. Kim, D. Schymura, F. Stehn, Measuring the similarity of geometric graphs, in *Proceedings of International Symposium on Experimental Algorithms* (2009), pp. 101–112
12. S. Chowdhury, F. Mémoli, Persistent homology of directed networks, in *Proceedings of 50th Asilomar Conference on Signals, Systems and Computers* (IEEE, New York, 2016), pp. 77–81
13. D. Cohen-Steiner, H. Edelsbrunner, J. Harer, Stability of persistence diagrams, in *Proceedings of 21st Annual Symposium on Computational Geometry* (2005), pp. 263–271

14. D. Conte, P. Foggia, C. Sansone, M. Vento, Thirty years of graph matching in pattern recognition. Int. J. Pattern Recognit. Artif. Intell. **18**(3), 265–298 (2004)
15. H. Edelsbrunner, J. Harer, *Computational Topology: An Introduction* (AMS, Providence, 2010)
16. D. Eppstein, Subgraph isomorphism in planar graphs and related problems. J. Graph Algorithms Appl. **3**(3), 1–27 (1999)
17. R. Ghrist, Barcodes: the persistent topology of data. Bull. Am. Math. Soc. **45**, 61–75 (2008)
18. A. Grigor'yan, Y. Lin, Y. Muranov, S.-T. Yau, Homologies of path complexes and digraphs (2012, Preprint). arXiv:1207.2834
19. A. Hatcher, *Algebraic Topology* (Cambridge University Press, Cambridge, 2002). Electronic Version
20. S. Karagiorgou, D. Pfoser, On vehicle tracking data-based road network generation, in *Proceedings of 20th ACM SIGSPATIAL International Conference on Advances in Geographic Information Systems* (2012), pp. 89–98
21. H. Kashima, K. Tsuda, A. Inokuchi, Kernels for graphs, in *Kernel Methods in Computational Biology*, ed. by B. Schölkopf, K. Tsuda, J.-P. Vert (MIT Press, Cambridge, 2004), pp. 155–170
22. J. Mondzech, M. Sester, Quality analysis of Openstreetmap data based on application needs. Cartographica **46**, 115–125 (2011)
23. E. Munch, A user's guide to topological data analysis. J. Learn. Anal. **4**(2), 47–61 (2017)
24. J.R. Munkres, *Elements of Algebraic Topology* (Addison-Wesley, Redwood City, 1984)
25. Open street map. http://www.openstreetmap.org
26. R.C. Read, D.G. Corneil, The graph isomorphism disease. J. Graph Theory **1**(4), 339–363 (1977)
27. B. Schweinhart, J.K. Mason, R.D. MacPherson, Topological similarity of random cell complexes and applications. Phys. Rev. E **93**(6), 062111 (2016)
28. K. Turner, Generalizations of the Rips filtration for quasi-metric spaces with persistent homology stability results (2016, Preprint). arXiv:1608.00365
29. S.V.N. Vishwanathan, N.N. Schraudolph, R. Kondor, K.M. Borgwardt, Graph kernels. J. Mach. Learn. Res. **11**(Apr), 1201–1242 (2010)
30. Y. Xu, S.M. Salapaka, C.L. Beck, A distance metric between directed weighted graphs, in *2013 IEEE 52nd Annual Conference on Decision and Control* (IEEE, New York, 2013), pp. 6359–6364