Topological analysis of traffic pace via persistent homology

Daniel R Carmody\textsuperscript{1} and Richard B Sowers\textsuperscript{2,***}

\textsuperscript{1} Department of Mathematics, University of Illinois at Urbana-Champaign Urbana, IL 61801, United States of America
\textsuperscript{2} Department of Industrial and Enterprise Systems Engineering, Department of Mathematics, University of Illinois at Urbana-Champaign Urbana, IL 61801, United States of America
\textsuperscript{***} Author to whom any correspondence should be addressed.

E-mail: dcarmod2@illinois.edu and r-sowers@illinois.edu

Keywords: intelligent urban navigation, topological data analysis, smart city

Abstract
We develop a topological analysis of robust traffic pace patterns using persistent homology. We develop Rips filtrations, parametrized by pace, for a symmetrization of traffic pace along the (naturally) directed edges in a road network. Our symmetrization is inspired by recent work of Turner (2019 Algebr. Geom. Topol. 19 1135–1170). Our goal is to construct barcodes which help identify meaningful pace structures, namely connected components or ‘rings’. We develop a case study of our methods using datasets of Manhattan and Chengdu traffic speeds. In order to cope with the computational complexity of these large datasets, we develop an auxiliary application of the directed Louvain neighborhood-finding algorithm. We implement this as a preprocessing step prior to our main persistent homology analysis in order to coarse-grain small topological structures. We finally compute persistence barcodes on these neighborhoods. The persistence barcodes have a metric structure which allows us to both qualitatively and quantitatively compare traffic networks. As an example of the results, we find robust connected pace structures near Midtown bridges connecting Manhattan to the mainland.

1. Introduction

Today’s road networks are challenged with an excess of congestion and faced with an array of routing algorithms, apps, and other tools which can easily lead to unexpected and emergent behaviors. On the other hand, with more data now available, it is becoming possible to think of big-data approaches to understanding large-scale mobility problems and compare cities [AMSW+17, DM15, FPV+13, GCW16, ZOXY16, ZUZ14].

Our effort here is to understand the topology of pace. Road infrastructure, passenger travel demands, and real-time information systems all can interact to create complex behaviors on road networks. An important analytical challenge is to construct robust and coarse descriptions of traffic behavior which can be used to assess and compare these complex networks. Our interest here is to apply some recent methods of topological data analysis to understand emergent behaviors in the presence of slow roads. We hope that these techniques will lead to new ways of characterizing mobility patterns.

2. Outline

We begin in section \textsuperscript{3} with a summary of previous work and the history of topological data analysis. The background material begins in section \textsuperscript{4}, where we review the fundamental construction which allows us to pass from directed graphs to topological objects with higher-order structural information. We note that the assignment of a Rips complex to a directed network is an intermediate step in order to ultimately assign a collection

\textsuperscript{*} This material is based upon work supported by the National Science Foundation under Grant Nos. CMMI 1727785 and DMS 1345032. Part of this research was performed while the author was visiting the Institute for Pure and Applied Mathematics (IPAM), which is supported by the National Science Foundation (Grant No. DMS-1925919). The authors would like to thank the High Performance Computing Center of the MSFE program at the University of Illinois for computational support. The code for this work is at https://github.com/dcarmod2/TrafficViz.

© 2021 The Author(s). Published by IOP Publishing Ltd
of topological signatures to a directed network. Section 5 introduces the central computational tool of the paper: persistent homology. While traditional measures of slow roads or congestion take as input a directed network and output a real number, our algorithm takes as input a directed network and outputs an element of a more exotic metric space, the space of persistence barcodes (or diagrams). The metric on the space of barcodes, called the Wasserstein distance defined in section 7, naturally compares pace structures simultaneously over an entire family of thresholds. We note here that methods similar to persistent homology have already appeared in congestion analysis by a different name when studying the percolation critical threshold of a network [LFW+15].

We introduce the real-world datasets to which we apply persistent homology in section 6. To facilitate comparison, we look at both the grid-like traffic network of Manhattan, New York, USA and the ring-like traffic network of Chengdu, China. Because the persistent homology of a Rips complex has non-trivial computational overhead, we take a detour in section 7 to introduce the tool which we use to counteract the explosive size of the Rips complex. Here our methods differ from traditional methods in topological data analysis for reducing the size of the Rips complex. Traditional methods simply remove points based on their proximity to neighbors, while we use a community finding algorithm specifically designed to coarse-grain directed networks. In section 8 we demonstrate the salient features of our method on simple, hand-constructed networks. We aim here to give some intuition to the abstract notion of a homological cycle and discuss how persistent homology depends on the filtration we choose (we study two different filtrations in this work).

Section 9 demonstrates how to use persistent homology for exploratory data analysis of Manhattan traffic. We find several interesting pace structures, including intuitive results like identifying that connections to the mainland slow down Manhattan traffic. Section 10 contains a brief analysis of the persistent homology of Chengdu traffic followed by a demonstration of how to use our methods in order to quantitatively compare traffic in different cities. Appendix A contains a more technical discussion on the algebraic aspects of computing persistent homology elucidated with a simple example.

3. Topological data analysis

A recurring motif in the development of mathematics is the development of tools to identify objects which are invariant under different perspectives. This has become increasingly interesting in computational analyses of big data, where large datasets may represent some underlying structure. The field of topological data analysis has developed a number of methods for addressing challenges of this sort and doing so in robust ways.

Originally developed by Edelsbrunner, Letscher, and Zomorodian in [ELZ02], persistent homology is a method for identifying certain topological features of a dataset in the presence of ’topological noise’. Persistent homology seeks to associate a family of topological invariants, in the form of vector spaces, with a dataset, and to also identify a measure of significance of each invariant (see [CM17]) by means of perturbation with respect to a parameter.

The original use case of persistent homology was to point clouds which sampled points on the surface of an object. Using the Vietoris–Rips complexes, persistent homology allows one to reliably reconstruct the topology of the surface, even in the presence of sampling noise, and quantify how robust that reconstruction is to the sampling noise via a barcode [Ghr08].

Our interest here is to develop an application of persistent homology to analysis of traffic networks. As with point cloud data, we want to understand macroscopic topological structures from local data; the topological structures of interest here are pace networks where a driver may find it difficult, or alternately easy, to avoid slow roads. A simplified version of this was developed in [WSK+17], which ignored the effect of directionality on streets (see also [FP20] for another application of persistent homology to urban data). Our goal here is a treatment which properly addresses the effect of directionality. Similar methods have also appeared in [LFW+15], but the metric space of barcodes makes no appearance in that work.

To understand the development of our ideas in a meaningful way, we consider traffic in Manhattan from the dataset of [DW16, DMA+16]. This dataset is large and complex enough (section 6) to allow for a range of traffic behavior. We do find some locally coherent structures, particularly near connections between Manhattan and the mainland. Some of our conclusions are in section 9. In order to compare structures between cities, we also consider traffic from Chengdu using the dataset of [GZDG19a].

4. Rips complexes of directed networks

To associate topological signatures to a directed network, we first associate a family of simplicial complexes (topological spaces) to the network. While it is true that an (undirected) graph already has the structure of a topological space, there is a major problem with using graph topology alone. As topological spaces
graphs have no built-in notion of direction, and the topological invariants calculated by viewing a graph as a
topological space do not see the directed nature of the graph. For example, the null space and left null space
of a directed adjacency matrix carry information about weakly connected components and undirected cycles
respectively rather than their strong/directed counterparts. Defining the notion of a directed topological space
leads one down the complicated path of sheaves of preorders or topological spaces together with a collection of
distinguished directed paths, see for example [Gra03]. For this reason, we instead build simplicial complexes
(topological spaces) from a directed graph which during construction use the directed nature of the graph and
whose associated topological invariants change if one changes the direction of links in the graph.

Furthermore, rather than choosing a single threshold which we consider slow, we use the paradigm of
persistence [ELZ02] to bundle a collection of spaces into a single topological object which simultaneously
captures the connectivity of a network at many different pace thresholds. The idea of considering a family
of thresholds in order to get a notion of robust emergent features in traffic is not new, and has already
appeared in [LFW+15]. While loc. cit. considers emergent features arising from functionally connected com-
ponents of a traffic network, we consider both connected components and higher order topological informa-
tion (cycles). We discuss the relationship between our methods and those of [LFW+15] in more detail in
subsection 4.1.

Our starting point is a weighted directed graph D = (V, E, W), informally (at the moment) corresponding
to a road network with intersections V, links E, and traffic pace W. Roads are naturally directed, so E consists
of ordered pairs (v, v′) of distinct vertices which correspond to a link from v to v′. A two-way link between v
and v′ corresponds to two edges, (v, v′) and (v′, v). Each edge e ∈ E has a positive weight we denoting traffic
pace (time/distance; the reciprocal of speed) on the link.

Section 6 will give some summary statistics of pace, distance, and travel time along links in our dataset.

Pace is an appropriate indicator of road speeds, even though pace is not additive along links (the pace along a
path consisting of several links is not the sum of the paces along the links).

If the link between v and v′ is bi-directional, the paces w_{w(v,v')} and w_{w(v')} in the different directions need
to be the same. To avoid degeneracy, we assume that the graph is strongly connected—any destination vertex
can be reached from any origin vertex via a directed path.

Topological data analysis depends on a notion of ‘nearness’ of a collection of points. For traffic pace analysis,
two points are ‘near’ if there is a fast directed path joining them (by contrast, for point clouds, the notion of
nearness is simply Euclidean distance). More formally, for vertices v and v′, let

\[ d(v, v') \] \def \min \{ \sum_{1 \leq n < N} w_{(v_n, v_{n+1})} : \{(v_n, v_{n+1})\}_{1 \leq n < N} \in E, v_1 = v, v_N = v', N \in \mathbb{N} \} \] \( (1) \)

A sequence (v₁, v₂, ..., vₙ) such that \{(v₁, v₂, ..., vₙ)\}_{1 \leq n < N} \subset E is a path from v₁ to vₙ; d(v, v′) is the minimum
sum of paces over all paths from v to v′. For consistency, we also define

\[ d(v, v) \] \def \equiv 0 \] \( (2) \)

for all v ∈ V. Since pace is not additive along links in a road network (unlike trip time or length of a path),
d is a synthetic distance which is a reasonable indicator of slow roads. For any \{(v, v', v'')\} \subset V, d satisfies the
triangle inequality

\[ d(v, v') \leq d(v, v'') + d(v'', v') \]

(the cheapest cost from v to v′ is less than the cost of paths which go through v''). Also, since the \( w_e \)'s are
positive, d(v, v') = 0 if and only if v = v'. However, since the edges are directional, d(v, v') need not agree
with d(v', v).

To capture the directional nature of our network, we will build upon [Tur19] and consider symmetrizations
of distance given by

\[ \text{sym}_{−}(d)(v, v') \] \def \min \{ d(v, v'), d(v', v) \} (v, v') \in V \times V \] \] \( (3) \)

Note that \( \text{sym}_{−}(d) \) is a metric. Fix v and v' in V. Since d(v, v') = 0 if and only if v = v'; it follows that
\( \text{sym}⁺_{−}(d)(v, v') = 0 \) if and only if v = v'. Secondly, for v'' in V,

\[ \text{sym}⁺_{−}(d)(v, v') = \max \{ d(v, v'), d(v', v) \} \leq \max \{ d(v, v''), d(v'', v) \} \]

\[ \leq \max \{ \text{sym}⁺_{−}(d)(v, v') \} \].

3
\{\text{sym}_+(d)(v', v') + \text{sym}_+(d)(v', v)\} \\
= \text{sym}_+(d)(v, v'') + \text{sym}_+(d)(v'', v')

(the last equality following since both terms in the maximum are the same), which is the triangle inequality.

On the other hand, \text{sym}_-(d) is not a metric.

**Counterexample 4.1.** Consider the diagram

Then

\[
\text{sym}_-(d)(v_1, v_3) + \text{sym}_-(d)(v_3, v_2) = 1 + 1 < 5 = \text{sym}_-(d)(v_1, v_2)
\]

so the triangle inequality does not hold.

Given a pairwise notion of 'proximity' such as \text{sym}_+(d), we can construct Rips filtrations which describe how bigger and bigger ‘neighborhoods’ (defined according to \text{sym}_+(d)) fill up the entire space. For \text{sym}_+(d) these neighborhoods will correspond to nodes which are ‘close’ in terms of bi-directional travel times, while for \text{sym}_-(d), the nodes will be ‘close’ in at least one direction. The work of [Tur19] clarifies the structure of these Rips complexes built from \text{sym}_+(d). Symmetric notions of proximity simplify the construction of these neighborhoods, but this too can in fact be relaxed (Dowker [CM18] persistence diagrams, for example, might be appropriate to identify robust origin or destination regions). See also [FP19] for yet other examples of persistence in geospatial data.

An understanding of the topology of networks is often built upon simplices. A triangle (a two-simplex) with vertices \(v, v', v''\) is denoted as \([v, v', v'']\), a line (a one-simplex) with vertices \(v\) and \(v'\) is denoted as \([v, v']\), and a point (a zero-simplex) \(v\) is denoted as \([v]\). Subsimplices are naturally defined; \([v, v']\) is a subsimplex of \([v, v', v'']\). Higher-dimensional simplices can be similarly defined, but they will not play a role in our analysis.

**Definition 4.2 (Simplicial complex).** Given a vertex set \(V' \subset V\), an abstract simplicial complex \(\Delta\) with vertex set \(V'\) is a collection of simplices which is closed under the operation of taking nonempty subsets; i.e., if \(K \in \Delta\) and \(L\) is a subsimplex of \(K\), then \(L \in \Delta\).

In other words, the collection

\[\{[v, v', v''], [v, v'], [v, v''], [v'], [v'], [v']\}\]

is the simplicial complex formed by a triangle with vertices \(v, v', v''\).

**Definition 4.3 (Rips complex).** For \(\varepsilon > 0\), define the two-skeleton of the Rips complex as

\[R^2_{\varepsilon}(D) \overset{\text{def}}{=} \{[v_1, v_2, v_3, v_4]: N \leq 3, \max_{1 \leq i < j < k < N} \text{sym}_+(d)(v_i, v_j, v_k) \leq \varepsilon\} \tag{4}\]

This is clearly a simplicial complex; if \(\text{sym}_+(d)(v, v') < \varepsilon\) for all \(v\) and \(v'\) in some \(V_\varepsilon \subset V\), then \(\text{sym}_+(d)(v, v') < \varepsilon\) for all \(v\) and \(v'\) in any nonempty subset of \(V_\varepsilon\). Simplicial complexes are formally defined as finite subsets of a given vertex set; since \(V\) is itself finite, all subsets are also necessarily finite. Counterexample 4.1 in particular shows a distance-preserving embedding may not exist; this precludes using Cech complexes (which have nicer theoretical properties) instead of the Rips complexes.

Looking now at the Rips filtration \(R_\varepsilon(D) \overset{\text{def}}{=} \{R^0_{\varepsilon}(D); \varepsilon > 0\}\), we see that \(R^0_{\varepsilon}(D)\) becomes larger as \(\varepsilon\) increases; more pairs of points satisfy the distance criterion. If \(\varepsilon < \varepsilon'\), then there is a canonical inclusion from \(R_{\varepsilon}(D)\) into \(R_{\varepsilon'}(D)\). Understanding the geometry of this map (via algebraic topology) gives us a rigorous framework for ‘unfolding’ the topology of the network. We will formalize this in section 5.
We will be interested in ‘complex’ road networks, where $|V|$ is large (see our case study of section 6). While the Rips complex gives us an easily-implementable definition of ‘neighborhood’ (see [CM17] and [CdSO12] for a detailed discussion of Rips and Cech complexes), the total number of simplices in the Rips complex grows exponentially with the number of vertices [Zom10]. Nevertheless, our interest in connected sets of congested traffic, and ‘beltways’ of faster streets surrounding congested areas allows us to focus only on zero, one, and two-simplices of the Rips complex; these computations will be at most cubic in the number of vertices [Zom10].

4.1. A comparison to percolation methods
In [LFW+15], for a traffic network $G$, the authors assign to each link at a given time the percentage of maximal velocity, where maximal velocity is computed as the 95th percentile of velocity measurements on the link on a given day. They then vary a threshold $q$ from 0 to 1, and include only links whose percentage of maximal velocity is above the threshold $q$. After computing strongly connected components, the authors obtain a family of clusters which merge over time. The associated algebraic object keeping track of these merges is exactly the strongly connected persistence module from definition 26 of [Tur19].

Although [LFW+15] is similar in spirit to our work, there are some fundamental differences. We instead choose to work with the symmetrization of the path distance $\text{sym}_{\frac{1}{2}}(d)$ (another approach from [Tur19]) because we want to study higher dimensional homological information, namely data about cycles in a network. Rather than thresholding links in the original graph based on the weight of the link alone, we create a simplicial complex which has links (one-simplices) between nodes if a driver can travel at a high speed for the entire journey from one node to another.

5. Persistent homology

Although the homology of a simplicial complex or topological space has an abstract definition, the computation of homology with field coefficients comes down to calculating the row spaces and null spaces (image and kernel) of some matrices. For example, the null space of the edge-node adjacency matrix of an undirected graph is the zeroth cohomology group of the graph (which is equivalent to the zeroth homology in our setting), while the left null space is the first homology group. In the spirit of [LFW+15], we are more interested in the functional properties (i.e. traffic pace) of a traffic network rather than the structural properties of the underlying graph.

The key insight of persistent homology is that one can simultaneously compute the homology of a family of simplicial complexes while keeping track of how the relationships between the complexes induce relationships in homology in an algorithmic fashion. While spectral sequences have a long history of being used to solve such problems, spectral sequences do not in any practical sense yield an algorithm for computing homology. At the end of this section, we recall the metric structure on the collection of persistence diagrams, which allows one to use a topological notion of distance when comparing the persistent homology of Rips complexes associated to directed graphs.

The inclusion of Rips complexes into each other as $\varepsilon$ varies in (4) leads to quantitative barcodes which capture the stability of topological structures. Taking the homology of each complex with coefficients in the field $\mathbb{Z}/2\mathbb{Z}$, we get the persistence module

$$0 \to H_* (R^\infty_\varepsilon (D); \mathbb{Z}/2\mathbb{Z}) \to \cdots \to H_* (R^\infty_0 (D); \mathbb{Z}/2\mathbb{Z})$$

for $\varepsilon_1 < \varepsilon_2 < \cdots < \varepsilon_M$, which breaks into a direct sum of persistence intervals [ZC05], which are chain complexes of the form

$$0 \to \mathbb{Z}/2\mathbb{Z} \xrightarrow{id} \mathbb{Z}/2\mathbb{Z} \to \cdots \to \mathbb{Z}/2\mathbb{Z} \to 0,$$

where the copies of $\mathbb{Z}/2\mathbb{Z}$ correspond to the $\varepsilon_m$’s. In practice, the relevant values of $\varepsilon$ are the values of $\text{sym}_{\frac{1}{2}}(d)(v,v')$ as $v$ and $v'$ range over $V$. Persistent homology outputs a collection of persistence intervals which shed light on pace structures which are in some sense robust.

We can visualize this persistence interval in a number of ways. The two most common are the persistence barcode, which plots a horizontal bar starting at $\varepsilon_1$ and ending at $\varepsilon_m$ or as the point $(\varepsilon_1, \varepsilon_m)$ in a persistence diagram (a persistence diagram is a collection of (birth, death) pairs in the plane together with the diagonal $\Delta$). See [CM17] for details). Furthermore, we can compute an explicit collection of homology generators for each value of $\varepsilon$, and highlight these generators via a visualization of the one-skeleton (a graph) of the Rips complex $R^\varepsilon (\text{sym}_\varepsilon (d))$. See also appendix A for a detailed example of a persistent homology calculation.

Our goal is to use explicit generators of persistent homology to draw conclusions about traffic routing and pace. We view separations of the Rips complex into connected components as being due to traffic obstacles (high pace edges or inherent lack of connectivity in the traffic graph), and we capture the connected
components of the Rips complex at various pace thresholds using persistent $H_0$. The generators of $H_1(R^\infty(\text{sym}_+(d)))$ will capture bi-directional beltways around congested areas. Studying uni-directional traffic is slightly more involved. By the definition of $\text{sym}_+(d)$, the first persistent homology group $H_1(R^\infty(\text{sym}_1(d)))$ will capture fast routes around congested areas which are undirected cycles but not necessarily directed cycles. It follows that fast uni-directional cycles will (up to change of basis) be a subset of the generators returned. Furthermore, fast undirected cycles can give useful information about where to build bypasses in order to construct a proper beltway around slow roads: if the majority of the edges in an undirected cycle correspond to a single orientation, then building bypasses over the other edges yields a fast uni-directional beltway. Starting in section 9, we will develop a case study of Manhattan. As far as obtained generators which are optimal in some sense, because we are using $\mathbb{Z}/2\mathbb{Z}$ coefficients, the optimal generator problem is NP-hard (even for $H_1$; see [CF11]). For this reason we do not pursue optimal generators here. We believe that the tools of topological data analysis can be of use in comparing traffic behavior in different cities or environments (viz seasons, before-and-after comparisons). The Wasserstein distance [CM17] allows us to compare persistence diagrams and thus simultaneously compare topological signatures at various scales. The Wasserstein one-distance between persistence diagrams $dgm_1, dgm_2$ is defined as

$$W_1(dgm_1, dgm_2) = \inf_{\text{matching } m} \sum_{(p,q) \in m} ||p - q||_\infty.$$  

(7)

Here a matching $m$ of two persistence diagrams $dgm_1, dgm_2$ is a subset $m \subseteq dgm_1 \times dgm_2$ such that for every $p \in dgm_1 - \Delta$ and $q \in dgm_2 - \Delta$, the sets $(\{p\} \times dgm_2) \cap m$ and $(dgm_1 \times \{q\}) \cap m$ each contain a single pair.

6. Data and setup

To ground our efforts, we develop our ideas around the dataset of [DMA+16], which gives us hourly estimates of the travel times along different links in Manhattan. Our model of road networks is given by Open Street Map (circa February 26, 2020), which lists 4496 nodes (intersections) and 9720 links in Manhattan.

The dataset of [DMA+16] was obtained by reverse-engineering traffic speeds (and counts) from taxi origin-destination pairs, assuming that taxi drivers universally minimize travel times. From that dataset, we use hourly (24 h a day) traffic estimates in

$$[2012 - 01 - 01 00:00:00, 2013 - 01 - 01 00:00:00)$$  

(8)

(local time). We estimate travel time data by randomly sampling travel times from one business day per week [DW16] in the subinterval

$$[2012 - 1 - 00:00:00, 2012 - 12 - 26 23:00:00]$$

of (8). Our estimated travel time dataset then contains 14 243 122 total traffic speed estimates. Further restricting this data to only those links which begin and end in Manhattan, we obtain data for 5491 (roughly 56%) of the 9720 links in our graph. Links with no travel times are removed from the graph. Figure 39 shows how the estimated travel times and speeds (i.e., link distance, as given by Open Street Map, divided by estimated time) vary throughout the hours of the day. The geographic distribution of the pace (the reciprocal of speed) is given in figure 40.

These speed estimates serve as a proxy for actual travel speeds (which of course may differ from posted speed limits). The speed estimates of [DMA+16] may in fact be incomplete at any given time; a given link may have had no traffic, or any ‘optimizing’ taxi driver would have avoided it (due to, e.g., congestion). We resolve these deficiencies by averaging across days to get an hour-of-day estimate on each edge. In other words, if the dataset has estimates of the travel time of some link at 8 AM on two different days as 50.84 s and 30.0 s, we will define the travel time on that link to be 40.42 s. If travel times are stationary on a daily basis, this will approximate the true travel times.

Notwithstanding this averaging, consistently missing speed estimates may lead to disconnected components of Manhattan. Let the weighted directed graph $\text{ManG}_e \overset{\text{def}}{=} (V_e, E_e, W_e)$ denote the largest strongly connected component of the directed road network on Manhattan island after removing edges missing a speed estimate (figure 1), with $W_e$ being the collections of paces along the edges. Then $\text{ManG}_e$ then has 1087 intersections (the set $V_e$) and 2036 directed links (the set $E_e$).

In order to demonstrate how our algorithm changes in the face of differing graph topology, we also study the traffic network of Chengdu, China using the dataset [GZDG19b]. The dataset [GZDG19b] estimates traffic speeds at different time periods using GPS data from floating vehicles, ignoring links with few or no GPS
readings. After averaging traffic speeds along different time periods to obtain average speed and time estimates, we obtain a strongly connected directed graph \( \text{ChengG} \), with 1902 nodes and 5943 links shown in figure 2. New York City and Chengdu were ranked 52 and 65, respectively, in the worldwide TomTom traffic index rankings in 2019 [Tom].

7. Louvain algorithm

The complexity of the large road networks poses some significant computational challenges. For any sufficiently large \( \varepsilon \), \( R^\varepsilon(\text{sym}_N(d)) \) will have

\[
\sum_{n=1}^{\lfloor |V|/n \rfloor} \binom{|V|}{n} = 2^{|V|} - 1 = 2^{1087} - 1
\]

vertices; for \( \varepsilon > 0 \) sufficiently large, the Rips complex is the entire simplex on \( V \). Persistent homology has cubic complexity (it is essentially Gaussian elimination [ZC05]), requiring on the order of

\[
2^{3|V|} = 2^{3 \times 1087} \approx 10^{981}
\]

computations for \( \text{ManG} \), (in the worst case).

To approximate the theoretical calculations with tractable ones, we will first use the Louvain algorithm [BGLL08] to coarse-grain the graph into statistically similar neighborhoods. We note that the Louvain
algorithm has been extended to handle directed networks in [DP15]. The Louvain algorithm seeks to find a partition $P$ of $V$ which maximizes

$$Q(P) = \frac{1}{|W|} \sum_{(v,v') \in E} \left( w_{v,v'} - \frac{d_{in}^v d_{out}^{v'}}{|W|} \right),$$

where

$$|W| \overset{\text{def}}{=} \sum_{e \in E} w_e$$

and where

$$d_{in}^v \overset{\text{def}}{=} \sum_{v' \in V : (v,v') \in E} w_{v,v'}$$

$$d_{out}^v \overset{\text{def}}{=} \sum_{v' \in V : (v',v) \in E} w_{v,v'}$$

are, respectively, the weighted in- and out-degree of a vertex $v$, and $v \sim P v'$ if both $v$ and $v'$ are in the same set in the partition of $V$. We start by assigning each node in $V$ to its own community (i.e., begin with the $|V|$ communities in the trivial partition). Given a partition $P$ and a vertex $v$, we can update $P$ based on $v$ by trying to increase $Q(P)$ by moving $v$ to each of the different sets in $P$; if we can increase $Q(P)$ by doing so, we get a new partition. A pass through $V$ corresponds to sequentially updating the partitions based on each $v$ in a random ordering of $V$. Letting $P_n$ be the partition after the $n$th pass through $V$ (with $P_0$ being the trivial partition $V$ itself), we get a (stochastic) dynamical system on partitions on $V$. After $n^*$ passes, the partitions remain constant (i.e., the Louvain algorithm enters a fixed point). The (directed) Louvain algorithm is a greedy stochastic algorithm. It is believed to run in $O(|E|)$ time (see also [Tra15]).

Informally, we can split $d_{in}^v$ and $d_{out}^v$ into internal (intra-community) and external (inter-community) connections. The Louvain algorithm seeks to increase internal connections while decreasing external connections (see [BGLL08] for motivation).
7.1. Community simplification

If we have a partition \( P \) of \( V \), we construct a simplified graph \( \text{ManG}^P = (V^P, E^P, W^P) \). Let \( V^P \) be the partition \( P \) itself. For \( C \) and \( C' \) in \( P \) and \( v \in C \) and \( v' \in C' \), define

\[
\tilde{d}^{C \cup C'}(v, v') \overset{\text{def}}{=} \min \left\{ \sum_{n \in N} w_{(v_n, v_{n+1})} \mid \{v_n\}_{1 \leq n \leq N} \subseteq E, \right\}
\]

as the cheapest cost to go from \( v \) to \( v' \), where paths are restricted to \( C \cup C' \) (if no such paths exist, we set \( \tilde{d}^{C \cup C'}(v, v') \) to be \( \infty \)). The set \( E^P \) will consist of ordered pairs \((C, C')\) of distinct elements of \( P \) (i.e., points in \( V^P \)) such that \( \tilde{d}^{C \cup C'}(v, v') < \infty \) for some \( v \in C \) and \( v' \in C' \). For \((C, C') \in E^P\), we define the edge weight

\[
w_{(C, C')}^\infty \overset{\text{def}}{=} \frac{\sum_{(v, v') \in C \cup C'} \tilde{d}^{C \cup C'}(v, v') < \infty \tilde{d}^{C \cup C'}(v, v')} {\sum_{(v, v') \in C \cup C'} \tilde{d}^{C \cup C'}(v, v') < \infty} .
\]

This gives a measure of how quickly one can directly travel between two communities. We will see in our case study that this coarse-graining still allows us to capture interesting behavior.

Once we have weights between communities, we can copy the development of section 4 and construct a (directional) distance \( d^\infty \) between communities (analogous to (1) and (2)); note that (9) is a distance between points, whereas \( d^\infty \) is a distance between communities), a symmetrization \( \text{sym}_- (d^\infty) \) (analogous to (3)), \( R^\infty_- (\text{ManG}^P) \), and then persistence module and persistence intervals (as in (5) and (6)) on \( \text{ManG}^P \).

7.2. Louvain communities

The Louvain algorithm gives an evolving (dynamically defined) set of communities \( \{P_n; n = 0, 1, \ldots\} \), and we then have \( \text{ManG}^{P_n} \) and \( \text{ChengG}^{P_n} \) (with \( \text{ManG}^{P_0} = \text{ManG} \)) (see figures 3 and 4). The Louvain algorithm should coarse-grain various aspects of \( \text{ManG} \), and we will be able to draw some conclusions in our case study from how persistent homology of these communities (see section 9) evolves as we iterate the Louvain algorithm (and consider coarser and coarse \( P_n \); see also section 8. Tables 1 and 2 in the supplementary appendix show how the sizes of the graphs change as we increase the number of iterations.

8. Intuition and simulations

Consider the problem of finding detours around slow roads (where a slow road is one with high pace/low speed). For example, one might be planning a city bus route, where one needs to find a cyclical path which covers a decent portion of the city and travels along non-congested roads. For specificity, consider three vertices \( v_1, v_2, \) and \( v_3 \) in \( V \) (figure 5). The simplicial complex generated by the routes between these vertices is

\[
\{[v_1], [v_2], [v_3], [v_1, v_2], [v_2, v_3], [v_3, v_1] \} .
\]

If the various edge distances (generalized paces) \( d(v, v') \) for distinct \( v \) and \( v' \) in \( \{v_n\}_{n=1}^3 \) are low (i.e., are not congested), then (for the purposes of choosing a route) it does not matter how we route around the triangle...
Figure 6. The beltway graph $G$. Violet edges are slow (high pace) while green edges are fast (low pace).

Figure 7. Persistent homology after 0 (left), 1 (middle) and 2 (right) iterations of the Louvain algorithm on the beltway graph $G$ with $\text{sym}_+$ $(d)$.

Figure 8. Generators of persistent $H_1$ (left), the $H_1$ barcode (middle), and the $H_1$ impact (right) for the Rips filtration on the beltway graph $G$ with $\text{sym}_+$ $(d)$.

with vertex set $\{v_n\}_{n=1}^3$; any choice will yield a route with low pace (other metrics like total length may of course differ). To reflect that the choice does not matter, we fill in the triangle $[v_1,v_2,v_3]$; i.e., add the two-simplex $[v_1,v_2,v_3]$ to (10). This makes the space of choices topologically contractible. Essentially, we are filling in triangles of ‘fast’ routes. Since $\text{sym}_+$ $(d)$ is in fact a symmetric distance function, this leads to an equivalence relation on paths (see [BGK15]; two paths are equivalent if their difference is homologous to 0).

Persistent homology gives us a framework for identifying robustness of routing decisions by varying a pace parameter $\epsilon$ telling us when to fill in triangles. In our above example, we will fill in the triangle $v_1v_2v_3$ if

$$\max_{v,v' \in \{v_1,v_2,v_3\} \atop v \neq v'} d(v,v') < \epsilon \quad \text{(for } \text{sym}_+ (d))$$

and

$$\min_{v,v' \in \{v_1,v_2,v_3\} \atop v \neq v'} d(v,v') < \epsilon \quad \text{(for } \text{sym}_- (d))$$
Figure 9. Generators of persistent $\mathbb{H}_1$ (left), the $\mathbb{H}_1$ barcode (middle), and the $\mathbb{H}_1$ impact (right) for the Rips filtration on the beltway graph $G$ with $\text{sym}_-(d)$.

Figure 10. Uni-directional beltway graph $G'$. Violet edges are slow (high pace) while green edges are fast (low pace).

Figure 11. Generators of persistent $\mathbb{H}_1$ (left), the $\mathbb{H}_1$ barcode (middle), and the $\mathbb{H}_1$ impact (right) for the Rips filtration on uni-directional beltway graph $G'$ with $\text{sym}_-(d)$.

(i.e., fill in triangles with pace in some way smaller than $\varepsilon$). If $\varepsilon = 0$ no triangles are filled in (all roads are congested), and if $\varepsilon \geq \text{max}_{v,v' \in V} d(v,v')$, all triangles are filled in (no roads are congested according to $\varepsilon$). As $\varepsilon$ increases, meaningful cyclical route will appear and merge, with the longest lasting and fastest detours being the most impactful for routing around slow roads (see the below definition 8.2 of impact). Mathematically, as $\varepsilon$ increases, we obtain a simplicial complex whose first homology group $\mathbb{H}_1$ will have generators which represent cyclical detours around congested areas. In other words, adding two-simplices allows us to focus on the topological features which correspond to meaningful choices.

In this section, we will consider several carefully selected synthetic road networks to experimentally verify that several aspects of our approach work as intended. We begin by giving an explicit description of one-cycles in the Rips complex (with $\mathbb{Z}/2\mathbb{Z}$ coefficients). While in general a cycle is defined as an element in the kernel of some linear transformation, the following lemma should hopefully give the reader a visual image of the abstract notion of a cycle as a collection of loops in a graph. Recall that the first homology group $\mathbb{H}_1(R)$
of a simplicial complex $\mathcal{R}$ is the quotient group of the cycles (elements in the kernel of the boundary map $\partial_1 : \mathcal{R}_1 \rightarrow \mathcal{R}_0$) by the boundaries (elements in the image of the boundary map $\partial_2 : \mathcal{R}_2 \rightarrow \mathcal{R}_1$). See appendix A for more details.

**Lemma 8.1.** Irreducible one-cycles (i.e., irreducible elements in the kernel of the one-dimensional boundary map $d_1$) in the one-skeleton of the Rips complex (i.e., the collection of all zero- and one-dimensional simplices) are represented by connected two-regular subgraphs of said one-skeleton.

**Proof.** Every two-regular subgraph gives rise to a cycle by definition of the boundary map and the fact that we’re using $\mathbb{Z}/2\mathbb{Z}$ coefficients. By the handshaking lemma, the edge-node adjacency matrix $M$ of a two-regular graph is square. Because the graph is connected, $\ker(M^T)$ has rank one, and thus the rank-nullity
Figure 15. The generators of persistent $H_1$ for the Rips filtration on $\text{ManG}^{\mathbb{Z}_2}$ for $\text{sym}_+ (d)$. These can be compared to posted Manhattan road speeds (see [Cit16]).

Figure 16. A lift of the cycles in the Rips complex on $\text{ManG}^{\mathbb{Z}_2}$ to $\text{ManG}$ for $\text{sym}_+ (d)$. The left and right figures differ in that they are lifts of the same cycles but with different orientations (e.g. clockwise vs counter-clockwise).

Theorem implies that $\ker(M)$ also has rank one. The kernel is thus generated by $(1 \ldots 1)^T$, so that the cycle is irreducible.

Assume now that we have an irreducible cycle. By definition the degree of every vertex in the cycle is even, and by irreducibility the subgraph corresponding to the cycle is connected. Say for the sake of contradiction that some node has degree $\geq 4$. By the handshaking lemma, $|E| \geq |V| + 1$. The edge-node adjacency matrix $M$ of the cycle thus has kernel of rank at least one. We claim it must have a kernel of rank at least two, contradicting irreducibility. By definition of $M$ and the fact that we use $\mathbb{Z}/2\mathbb{Z}$ coefficients, the vector $(1 \ldots 1)$ is in the left kernel of $M$. Thus $M^T$ has rank at most $|V| - 1$. It follows that $M$ has rank at most $|V| - 1$. \hfill \Box

We want to slightly modify the notion of persistence (the length of a bar in the persistence barcode) to capture the length of a bar relative to its starting point (pace, in this case). While the persistence intervals of (6) are scale-specific (i.e., lengths will change, for example, if distance is measured in kilometers as opposed to miles), we want to capture the fact that the persistence interval $(20, 30)$ lasts 10% of its original starting point (of 20).
Definition 8.2. For a persistence interval $z = (a, b)$,

\[ \text{impact}(z) = \log \text{death}(z) - \log \text{birth}(z) = \log \frac{\text{death}(z)}{\text{birth}(z)} = \log \frac{b}{a}, \]

i.e., the logarithm of the persistence, to be the impact of the cycle.

8.1. Simulations

Figure 6 shows a strongly connected graph $G$ with a bi-directional outer beltway and uni-directional inner beltway. The pace along the edges of the outer green beltway is 1, while the pace along the inner violet edges is 20, making it more efficient to travel along the outer beltway. The barcode corresponding to the Rips filtration of $\text{sym}_1(d)$ (as in section 6) is shown in the left panel of figure 7. The $H_1$ portion of the barcode is expanded and shown with a corresponding choice of homology generator and the $H_1$ impact in figure 8. As expected, the outer beltway corresponds to a generator of the persistent homology which is born early and dies early (the top gray bar). While this generator seems insignificant (i.e. not persistent) in the original barcode, it is clearly the highest impact cycle as shown by the right panel. This supports the intuition that high impact cycles should correspond to large, speedy beltways.

Informally, the Louvain algorithm should allow us to coarse-grain a graph (and thus allow more tractable calculations) without losing too much of the information available in persistent homology. The Louvain
algorithm also reduces visual ‘noise’ by condensing nodes with short lifespans in the \( H_0 \) barcode. We can see this more clearly by constructing in figure 7 the persistence barcode after every pass of the Louvain algorithm through \( V \).

By contrast, figure 9 shows the persistent homology of figure 6 using the Rips filtration of \( \text{sym}_-(d) \) (rather than \( \text{sym}_+(d) \)). We still observe the speedy outer beltway as a high impact cycle, but now we also observe the slow inner beltway as a cycle with fairly large impact.

To get some more perspective on \( \text{sym}_-(d) \), let us modify figure 6 by making the outer beltway uni-directional (clockwise), giving us \( G' \) of figure 10. The graph is still strongly connected, and the persistence diagram of \( \text{sym}_-(d) \) is shown in figure 11. We now only detect the bi-directional inner beltway with \( \text{sym}_+(d) \), as can be seen in figure 12. This is because the outer beltway dies (is in the image of the boundary operator) as soon as it is born. Informally, if a generator has high impact in the \( \text{sym}_-(d) \) but low impact in \( \text{sym}_+(d) \), the generator is likely to be uni-directional.

Another way to think about the difference between \( \text{sym}_-(d) \) and \( \text{sym}_+(d) \) is to consider a bi-directional cycle graph with uniform edge weights and then make a single edge uni-directional. With \( \text{sym}_-(d) \), the distance matrix will remain unchanged. However, with \( \text{sym}_+(d) \), the distance between the vertices bounding the uni-directional edge will increase drastically, as one will have to travel every other edge in the cycle to get back to the adjacent vertex. This implies that the bar corresponding to the cycle in the \( \text{sym}_+(d) \) filtration will have length zero, since every other pair of vertices has distance strictly less than the maximum distance between the vertices bounding the uni-directional edge (hence the cycle will already be a boundary when it is born); see appendix A.

9. Persistent homology of Manhattan traffic

With our goal of using persistence to understand the topology of pace, let us look at the data of section 6. In agreement with the simulations of section 8:

- \( H_1(R_+(\text{ManG}^{\text{P}})) \) captures bi-directional emergent beltways in traffic; see subsection 9.1
Figure 21. The generators of persistent $H_1$ for the Rips filtration on $\text{ManG}^{2^\infty}$ for $\text{sym}^{-\infty}(d)$. These can be compared to posted Manhattan road speeds (see [Cit16]).

Figure 22. A lift of the cycles in the Rips complex on $\text{ManG}^{2^\infty}$ to $\text{ManG}$ for $\text{sym}^{-\infty}(d)$. The left and right figures differ in that they are lifts of the same cycles but with different orientations (e.g. clockwise vs counter-clockwise).

- $H_1(\text{R}^{-\infty}(\text{ManG}^{p^n}))$ captures uni-directional emergent beltways in traffic; see subsection 9.2
- High impact cycles are preserved by our Louvain preprocessing step (see figure 19)

Several conclusions stem from this analysis:
- Entry and exit points in Manhattan (bridges and tunnels) cause traffic slowdowns at the entry points,
- Traffic seems to obey the lower posted speed limits near Thompson Park creating a bi-directional beltway,
- There is an uni-directional emergent beltway around lower Manhattan, and
- 59th street is so slow that it acts as a barrier between lower and central Manhattan.

9.1. Bi-directional cycles: persistent homology $\mathcal{R}_+(\text{ManG}^{p^n})$

The Louvain algorithm takes five passes over $\text{ManG}$ before converging, giving us five successively coarser partitions of the starting graph. From each of these partitions we can construct a Rips complex $\mathcal{R}_+(\text{ManG}^{p^n})$, $n = 0, \ldots, 4$. In order to compute these Rips complexes in a reasonable time, we focus on $\mathcal{R}_+(\text{ManG}^{p^n})$ for $n \in \{2, 3, 4\}$. Figure 13 shows the persistence barcodes for $\mathcal{R}_+(\text{ManG}^{p^n})$ for $n \in \{2, 3, 4\}$. We see that none of the cycles (corresponding to green bars in figure 13) survive more extreme preprocessing by the Louvain algorithm, indicating that the beltways found are probably low impact. We confirm this in figure 14, which plots the $H_1$ barcode of $\mathcal{R}_+(\text{ManG}^{p^n})$ together with the shape and impact of the cycles, coded by color. 'Low'
impact is admittedly subjective, although interpreting it to be an impact bar whose length is less than

$$\frac{1}{4} \log \left( \text{maximal finite death time of generators of } H_0 \right)$$

will give us a consistent definition for our results. A more rigorous definition of low impact could result from a detailed analysis of the impact of persistent $H_1$ of random directed graphs with a fixed (positive real valued) degree distribution. We will not pursue this here.

Figure 15 positions the generators of persistent $H_1$ for $R_+(\text{ManG}^{sym})$ on a map. The position of node $n$ in each cycle in this condensed graph is located at the centroid of the communities in $P_n$. The green generator near Alphabet City surrounds Thompson Park, where the posted road speeds are 5 mph slower than the surrounding roads [Cit16]. This cycle is a small beltway around Thompson Park due to the slower speeds near the park. From figure 1, we see that our traffic graph does not contain enough data from the West Village to identify the slower road speeds in that area.

Finally, we want to decompress the data to see what these cycles and connected components look like in the original graph. Figure 16 shows cycles in $\text{ManG}$, produced as follows: we find the closest node (ties broken randomly) in $\text{ManG}$ to each node in the Rips complex, then compute a shortest path (ties broken randomly) between each of these nodes (and hence we first choose an orientation of the cycle since shortest paths are not symmetric). This cycle should be thought of as the path a fast driver would take if they were to start from home and proceed to make trips where they pick a passenger up at the same location they drop the previous passenger off, then return home at the end of the day. As such these cycles may have inner loops or may require
Bridges and tunnels seem to divide traffic into connected components.

Figure 26. Persistent homology after 0 (left), 1 (middle) and 2 (right) iterations of the Louvain algorithm on the ChengG\textsuperscript{P2} with \textup{sym}\_\textup{−}(d). Red bars are $H_0$ generators while green bars are $H_1$ generators.

the driver to backtrack as they move to the next location. The condensed graph can be viewed as a quotient space of the original graph, and as such a lift of a cycle to the original graph is not unique.

Figure 17 shows the two most persistent connected components in the Manhattan traffic graph. We can infer that crossing 59th street is incredibly slow compared to average Manhattan traffic. Figure 18 shows the connected components in the Manhattan traffic graph earlier in the filtration. We see that the locations where bridges and tunnels enter Manhattan seem to cause divides between connected components. Unwinding the definition of our filtration, this indicates that the roads which cross entryways are relatively slow. In other words, traffic entering Manhattan island obstructs traffic crossing in an orthogonal direction.

9.2. Uni-directional cycles: persistent homology of $\mathcal{R}_−(\text{ManG}^\text{P}_n)$

Figure 19 shows the persistence barcodes for $\mathcal{R}_−(\text{ManG}^\text{P}_n)$ for $n \in \{2, 3, 4\}$. There are two major differences from the persistence barcode using $\mathcal{R}_+(\text{ManG}^\text{P}_n)$. First, there are 3 rather than 2 connected components which survive longest, and second there is a cycle (a generator of $H_1$) even after 4 applications of the Louvain preprocessing step. This should indicate that this cycle has higher impact.

We confirm this in figure 20, which plots the shape of the cycles together with their persistence and impact (color coded). The cycle of highest impact appears to be the dark purple cycle. We can see from figure 21 that it corresponds to the area surrounding much of downtown Manhattan. The fact that this cycle is not a generator of the persistent homology of $\mathcal{R}_+(\text{ManG}^\text{P}_n)$ indicates that it is not bi-directional. Lifting both orientations (clockwise/counterclockwise) of the cycle to the original graph as in figure 22; we see that the counter-clockwise direction appears as to follow a circular pattern, while the clockwise direction follows an incredibly convoluted path with a lot of backtracking. Unfortunately, investigating ManG as shown in figure 1, we see that the reason for the difference in the counterclockwise and clockwise orientations is a combination
missing travel time data along Park Avenue and the fact that the east–west streets in this region of Manhattan
are one-way.

Recall that homology generators of \( H_1(\mathcal{R}_-(\text{ManG}^P_{n})) \) can correspond to zig-zags rather than oriented
cycles. That is, edges in a homology generator can switch between clockwise or counter-clockwise orientation
as we move around the cycle depending on which direction is faster. It is thus necessary to check the homol-
ogy generators to see how closely they resemble an honest uni-directional beltway. We isolate the cyclic route
corresponding to the highest impact cycle in figure 23 in order to demonstrate how important the orientation
of a cycle can be. The red points correspond to locations that the cycle is required to visit; specifically, they
correspond to the nodes in \( \text{ManG} \) which are closest to the nodes in the highest impact cycle in \( \text{ManG}^P_{n} \). From
this representation we can immediately see which stops along the cycle are significantly ‘out of the way’. For
example in the path with counter-clockwise orientation (right panel in figure 23) we see that the corner near
the intersection of 9th Avenue and 30th street is hard to reach from the intersection of 46th street and 9th
Avenue and consequently should be left out of any planned route going counter-clockwise around the city.

The paths in figure 23 allow us to determine where to add a bypass in order to obtain a proper uni-directional
beltway.

Figure 24 shows the three most persistent connected components in \( \mathcal{R}_-(\text{ManG}^P_{n}) \). This is similar to the
persistent connected components in \( \mathcal{R}_+(\text{ManG}^P_{n}) \) with the exception that the black and light green connected
components merge at almost the same time as the light green and dark green connected components. Compar-
ing the filtration levels in figure 24 to the filtration levels in figure 17, we see that crossing 59th street is
much faster in one direction than the other. Because the only roads in our data connecting these components
are Lexington Ave (one way north \( \rightarrow \) south) and York Ave which is close to the entrance of the Queensbor-
ough bridge, it is faster in our graph to travel north to south than south to north across 59th St. Figure 25
shows the connected components at an earlier stage in the filtration. Similarly to the connected components
of \( \mathcal{R}_+(\text{ManG}^P_{n}) \), we see that entries into the city seem to split connected components.
10. Using persistent homology to compare traffic across cities: a study of Chengdu

Our goals now are to demonstrate how persistent homology is affected by the topology of the underlying traffic network and to show how persistent homology provides a framework for comparing traffic across cities. We use traffic data from Chengdu, China described in section 6 in order to compare the grid-like network of Manhattan to a more ring-like network. We focus on the filtration $\text{sym}_-(d)$ in order to streamline the discussion and highlight methods of comparison.

We produce the same figures as for Manhattan, with the exception of the impact barcode in order to increase the visibility of the plotted cycles. Figure 26 contains the total persistence barcode of the Chengdu traffic network at three different levels of granularity. As expected, in a traffic network with structural beltways, we find more prominent emergent functional beltways which survive coarse-graining of the graph. Because the pace distribution of Chengdu is more concentrated than the distribution for Manhattan (figure 27), one expects the bars in its persistence barcodes to be shorter.
Figure 32. Some components of Chengdu traffic (left). The three most persistent connected components recolored for contrast (right).

Figure 33. The Wasserstein one-distance between the Chengdu and Manhattan traffic networks at various levels of granularity.

Figure 28 shows the shape of the persistent $H_1$ generators of $\text{ChengG}^P$. Compared to Manhattan, there are significantly more emergent beltways, and the persistent emergent beltways connect far more nodes in the Rips complex. Indeed we see from figure 29 that the cycles cover large portions of the city, indicating that the structural beltways are working as intended. The fact that the two orientations of lifts of the generating cycles to the traffic network in figure 30 are so similar indicates that the generating cycles in the barcode are bi-directional. We can see this in particular when examining a lift of the most persistent cycle with two different orientations in figure 31. Finally, figure 32 shows that there are two persistent components of Chengdu traffic near the Jinjiang theater and Dongguang street residential district.

Figure 33 shows the Wasserstein one-distance between the both the $H_0$ and $H_1$ barcodes of Chengdu and Manhattan at each partition level. Note that the distances in the $H_0$ confusion matrix are globally higher than the distances in the $H_1$ confusion matrix because there are more bars in the $H_0$ barcode: see formula (7). Just by looking at the networks of Manhattan and Chengdu, we might expect a significant difference in the $H_1$ barcode, since this measures emergent ring structures. Indeed, we can see that the differences between the $H_1$ barcodes for Manhattan and Chengdu are greater than any of the differences within a single city occurring because of changes in granularity.

11. Conclusions

By computing the persistence barcode of traffic networks, we provide a new means of both qualitative and quantitative comparison of city traffic at various levels of granularity. First, we show that coarse-graining a traffic network using the Louvain algorithm preserves important topological features by studying two hand-crafted networks. We demonstrate how to read data from persistent homology generators in section 9, showing the disruptive effect that bridges and tunnels have on Manhattan traffic and identifying a surprising emergent beltway. We confirm that significant emergent beltways in traffic survive our coarse-graining process by computing the barcode of the Chengdu traffic network. Finally, we show how to extend the ideas of [LFW+15] to embed traffic networks into a metric space of persistence barcodes to allow for quantitative comparison between cities. We see that the structural beltways of Chengdu allow for significantly more...
emergent beltways than in Manhattan, and the Wasserstein distance between $H_1$ barcodes quantitatively verifies this intuition.

**Data availability statement**

The data that support the findings of this study are openly available at the following URL/DOI: https://doi.org/10.13012/B2IDB-4900670_V1.

**Appendix A. Persistent homology: an example**

To explain some of our topological tools for the nonspecialist, let us consider a simple weighted directed graph $D$ (as in section 4).

Consider figure 34, where the matrix form of the weights are

\[
\begin{pmatrix}
1 & 1 & 1 \\
1 & 1 & 1 \\
1 & 1 & 1 \\
1 & 1 & 1 \\
\end{pmatrix},
\]

i.e., $w_{(1,2)} = 1$, and where $\bullet$ denotes either a pair of vertices not connected by an edge (i.e., there is no edge from node 1 to node 3) or a diagonal pair of vertices. The matrix form of the distance function $d$ as in (1) and (2) is then

\[
\begin{pmatrix}
0 & 1 & 2 & 1 \\
3 & 0 & 1 & 2 \\
2 & 1 & 0 & 3 \\
1 & 2 & 1 & 0 \\
\end{pmatrix},
\]

i.e., $d(2, 1) = 3$. Symmetrizing, we get the matrix forms of $\text{sym}_+(d)$ and $\text{sym}_-(d)$ of (3) as

\[
\begin{pmatrix}
0 & 1 & 2 & 1 \\
1 & 0 & 1 & 2 \\
2 & 1 & 0 & 1 \\
1 & 2 & 1 & 0 \\
\end{pmatrix} \quad \text{and} \quad \begin{pmatrix}
0 & 3 & 2 & 1 \\
3 & 0 & 1 & 2 \\
2 & 1 & 0 & 3 \\
1 & 2 & 3 & 0 \\
\end{pmatrix}.
\]

Let us fix a finite ordered set

\[F \overset{\text{def}}{=} \{0, 2.5, 3.5\}\]

of filtration values. The choice of the filtration is up to the user and controls the resolution of the observed homological features.

Let us construct the Rips complexes $\mathcal{R}_\varepsilon(d)$ for each $\varepsilon \in F$. For $\varepsilon = 0$, $\mathcal{R}_0^0(d)$ is a simplicial complex with four zero-simplices $[1], [2], [3], [4]$ corresponding to the four nodes of the graph and no $n$-simplices for $n \geq 1$. Next, $\mathcal{R}_0^2(d)$ has these same four zero-simplices, but also has four one-simplices $[0, 3], [1, 2], [0, 2], [1, 3]$. Finally, $\mathcal{R}_0^3(d)$ is a contractible simplicial complex with four zero-simplices, six one-simplices, four two-simplices, and one three-simplex. In other words, $\mathcal{R}_0^3(d)$ is the clique complex on a complete graph with four vertices (figures 36–38).
We summarize the filtered Rips complex in figure 35. As usual, the dimension of a simplex is one less than the length of the tuple describing the simplex. The persistence degree is the index of the start filtration in the ordered filtration set; i.e., the degree of the grading of the simplex as a basis element of the graded $\mathbb{Z}/2\mathbb{Z}$-module associated to our Rips complex as in [ZC05].

In detail, let $C_i(R^\varepsilon_+(D); \mathbb{Z}/2\mathbb{Z})$ denote the $\mathbb{Z}/2\mathbb{Z}$-vector space with basis given by the $i$-simplices in $R^\varepsilon_+(D)$. For example, $C_0(R^0_+(D); \mathbb{Z}/2\mathbb{Z})$ is the four dimensional vector space $\mathbb{Z}/2\mathbb{Z}^4$ where a basis is given by the vertices of the graph. There is a linear (over $\mathbb{Z}/2\mathbb{Z}$) boundary map $\partial_i: C_i(R^\varepsilon_+(D); \mathbb{Z}/2\mathbb{Z}) \rightarrow C_{i-1}(R^\varepsilon_+(D); \mathbb{Z}/2\mathbb{Z})$ defined on basis elements by

$$\partial_i([x_0, \ldots, x_i]) \triangleq \sum_{j=0}^i [x_0, \ldots, \hat{x}_j, \ldots, x_i].$$

We have disregarded the usual alternating sign in the definition of the boundary map since we are considering $\mathbb{Z}/2\mathbb{Z}$ coefficients. If we compute the kernel of $\partial_i$ mod the image of $\partial_{i+1}$ to all the simplices in figure 35 we recover the homology of $R^\varepsilon_+(D)$. To deal with the filtration, the authors of [ZC05] note that each $C_i(R^\varepsilon_+(D); \mathbb{Z}/2\mathbb{Z})$ can be decomposed as

$$C_i(R^\varepsilon_+(D); \mathbb{Z}/2\mathbb{Z}) = \bigcup_{\varepsilon' \in F \leq \varepsilon} C_i(R^\varepsilon'_+(D); \mathbb{Z}/2\mathbb{Z}).$$

Since $C_i(R^\varepsilon_+(D); \mathbb{Z}/2\mathbb{Z}) \subseteq C_i(R^{\varepsilon+1}_+(D); \mathbb{Z}/2\mathbb{Z})$, for each $i$ we get a graded $\mathbb{Z}/2\mathbb{Z}[t]$-module

$$\bigoplus_{\varepsilon'_j} C_i(R^\varepsilon'_+(D); \mathbb{Z}/2\mathbb{Z}),$$

where the action of $t$ on $c_i \in C_i(R^\varepsilon_+(D); \mathbb{Z}/2\mathbb{Z})$ is

$$t \cdot c_i = c_i \in C_i(R^{\varepsilon+1}_+(D); \mathbb{Z}/2\mathbb{Z}).$$

That is, multiplication by $t$ moves simplices into a higher filtration, keeping track of the fact that if the distance between nodes is less than $\varepsilon_j$, it is also less than $\varepsilon_{j+1}$. As the boundary map $\partial_i$ can only decrease filtration, we can promote $\partial_i$ to a graded map by defining

$$\tilde{\partial}_i([x_0, \ldots, x_i]) = \sum_{j=0}^i t^{\deg([x_0, \ldots, x_j]) - \deg([x_0, \ldots, \hat{x}_j, \ldots, x_i])} [x_0, \ldots, \hat{x}_j, \ldots, x_i].$$
for \([x_0, \ldots, x_i] \in C_i(\mathbb{R}_ε^j(D); \mathbb{Z}/2\mathbb{Z})\), so that \(\tilde{\partial}_i([x_0, \ldots, x_i]) \in C_i(\mathbb{R}_ε^j(D); \mathbb{Z}/2\mathbb{Z})\). Here the degree deg of a simplex is the index of its filtration in the ordered filtration set. With a graded boundary map in hand, one can compute a homogeneous basis of its kernel and image, and use these to determine the persistence intervals (the fact that persistence intervals are well defined relies on the structure theorem for finitely generated modules over a principal ideal domain).

We now proceed to compute the persistent homology of the Rips complex defined by figure 35. Because there are no \(-1\)-simplices, \(\tilde{\partial}_0 = 0\).

\[
\tilde{\partial}_1 = \begin{bmatrix}
4 & [1,4] & [2,3] & [2,4] & [1,3] & [1,2] & [3,4] \\
4 & t & 0 & t & 0 & 0 & t^2 \\
3 & 0 & t & 0 & t & 0 & t^2 \\
2 & 0 & t & t & 0 & t^2 & 0 \\
1 & t & 0 & 0 & t & t^2 & 0
\end{bmatrix}
\]

We can perform column operations to put the matrix in column echelon form:

\[
\tilde{\partial}_1 = \begin{bmatrix}
4 & [2,4] & [2,3] & [1,3] & [2,3] & [1,4] - [1,3] & [2,4] - [2,3] & [2,3] - [2,4] & [1,2] - t(2,3) - \ell(1,3) & [3,4] - t(2,4) - \ell(2,3) \\
4 & t & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
3 & 0 & t & 0 & 0 & 0 & 0 & 0 & 0 \\
2 & t & t & t & 0 & 0 & 0 & 0 & 0 \\
1 & 0 & 0 & t & 0 & 0 & 0 & 0 & 0
\end{bmatrix}
\]
From this representation we can read off three pieces of information: a basis for the kernel, $Z_1$, a basis for the image, $B_0$, and the filtration at which the generators of $Z_0$ are killed. In particular, by looking at the degree of the pivots, we read off three persistence intervals $(0, 2.5)$ corresponding to the generators $[1], [3], [4]$ of $H^0$, and one interval $(0, \infty)$ corresponding to the generator in the non-pivot row, $[2]$.

Next, we calculate persistent $H_1$ by applying similar techniques to $\tilde{\partial}_2$.

$$\tilde{\partial}_2 = \begin{bmatrix}
[3, 4] & [1, 2, 3] & [1, 2, 4] & [1, 3, 4] & [2, 3, 4] \\
[1, 2] & 0 & 0 & 1 & 1 \\
[1, 4] & 1 & 1 & 0 & 0 \\
[1, 3] & 0 & t & t & 0 \\
[2, 4] & 0 & t & 0 & t \\
[2, 3] & 0 & 0 & 0 & t \\
\end{bmatrix}.$$  

Because $\tilde{\partial}_1 \circ \tilde{\partial}_2 = 0$, some linear algebraic calculations as in [ZC05] tells us that we can represent $\tilde{\partial}_2$ with respect to the basis of $Z_1$ as

$$\begin{bmatrix}
3, 4 - t[2, 4] - t[2, 3] & [1, 2, 3] & [1, 2, 4] & [1, 3, 4] & [2, 3, 4] \\
1, 2 - t[2, 3] - t[1, 3] & 0 & 0 & 1 & 1 \\
1, 4 - [1, 3] - [2, 3] - [2, 4] & 1 & 1 & 0 & 0 \\
\end{bmatrix}.$$  

Again, using column operations to reduce, we have

$$\begin{bmatrix}
3, 4 - t[2, 4] - t[2, 3] & [1, 2, 3] & [1, 2, 4] - [1, 2, 3] & [1, 3, 4] - [2, 3, 4] - [1, 2, 4] + [1, 2, 3] \\
1, 2 - t[2, 3] - t[1, 3] & 0 & 0 & 0 \\
1, 4 - [1, 3] - [2, 3] - [2, 4] & 0 & 1 & 0 & 0 \\
\end{bmatrix}.$$  

Now again we read off the persistence intervals by looking at the filtration of the generator of $Z_1$ corresponding to a given row and the degree of the pivot in that row. We see that the $H_1$ persistence intervals are $(2.5, 3.5)$, $(3.5, 3.5)$, $(3.5, 3.5)$.

Abstractly, the process can continue to simplices of higher degree (vis $H_2$), but we only use $H_0$ and $H_1$ in our efforts.

Appendix B. Supplementary figures

See figures 39 and 40 and tables 1 and 2.

![Figure 39. Estimated travel times (left) and speeds (right) in Manhattan.](image-url)
Figure 40. Pace in Manhattan (slower links have a higher pace value and are in red).

Table 1. Size of the Manhattan traffic network at various levels of granularity.

|          | ManG | ManG2 | ManG3 | ManG4 | ManG5 |
|----------|------|-------|-------|-------|-------|
| #Vertices| 4496 | 1087  | 56    | 21    | 18    |
| #Edges   | 9720 | 2036  | 226   | 69    | 55    |

Table 2. Size of the Chengdu traffic network at various levels of granularity.

|          | ChengG | ChengG2 | ChengG3 | ChengG4 |
|----------|--------|---------|---------|---------|
| #Vertices| 1902   | 71      | 23      | 22      |
| #Edges   | 9720   | 443     | 139     | 132     |

ORCID iDs

Daniel R Carmody https://orcid.org/0000-0003-3306-859X

References

[AMSW+17] Alonso-Mora J, Samaranayake S, Wallar A, Frazzoli E and Rus D 2017 On-demand high-capacity ride-sharing via dynamic trip-vehicle assignment Proc. Natl Acad. Sci. 114 462–7

[BGK15] Bhattacharya S, Ghrist R and Kumar V 2015 Persistent homology for path planning in uncertain environments IEEE Trans. Robot. 31 578–90

[BGLL08] Blondel V D, Guillaume J-L, Lambiotte R and Lefebvre E 2008 Fast unfolding of communities in large networks J. Stat. Mech. P10008

[CGSO12] Chazal F, de Silva V and Oudot S 2012 Persistence stability for geometric complexes Geometriae Dedicata 173 193–214

[CF11] Chen C and Freedman D 2011 Hardness results for homology localization Discrete Comput. Geom. 45 425–48

[Cit16] City of New York 2016 Current pre-vision zero speed limit maps.pdf https://www1.nyc.gov/html/dot/downloads/pdf/current-pre-vision-zero-speed-limit-maps.pdf

[CM17] Chazal F and Michel B 2017 An introduction to topological data analysis: fundamental and practical aspects for data scientists (arXiv:1710.04019)

[CM18] Chowdhury S and Mémoli F 2018 A functorial Dowker theorem and persistent homology of asymmetric networks J. Appl. Comput. Topol. 2 115–75

[DM15] Deri J A and Moura J M F 2015 Taxi data in New York city: a network perspective 2015 49th Asilomar Conf. on Signals, Systems and Computers pp 1829–33

[DMA+16] Donovan B, Mori A, Agrawal N, Meng Y, Lee J and Work D 2016 New York city hourly traffic estimates (2010–2013) https://doi.org/10.13012/B2IDB-4900670_V1

[DP15] Duguet N and Perez A 2015 Directed Louvain: maximizing modularity in directed networks (Orléans: Université d’Orléans) p 11

[DW16] Donovan B and Work D 2016 New York city taxi trip data (2010–2013) https://doi.org/10.13012/J8PN93HI8
[ELZ02] Edelsbrunner H, Letscher D and Zomorodian A 2002 Topological persistence and simplification Discrete Comput. Geom. 28 511–33

[FP19] Feng M and Porter M A 2019 Persistent homology of geospatial data: a case study with voting (arXiv:1902.05911)

[FP20] Feng M and Porter M A 2020 Spatial applications of topological data analysis: cities, snowflakes, random structures, and spiders spinning under the influence Phys. Rev. Res. 2 033426

[FPV+13] Ferreira N, Peco I, Vo H T, Freire J and Silva C T 2013 Visual exploration of big spatio-temporal urban data: a study of New York city taxi trips IEEE Trans. Visual. Comput. Graph. 19 2149–58

[GCW16] Guan X, Chen C and Work D 2016 Tracking the evolution of infrastructure systems and mass responses using publicly available data PloS One 11 e0167267

[Ghr08] Ghrist R 2008 Barcodes: the persistent topology of data Bull. Am. Math. Soc. 45 61–75

[Gra03] Grandis M 2003 Directed homotopy theory, I Cah. Topol. Géom Différentielle Catégoriques 44 281–316

[GZDG19a] Guo F, Zhang D, Dong Y and Guo Z 2019 Urban link travel speed dataset from a megacity road network (https://doi.org/10.6084/m9.figshare.7140209.v4)

[GZDG19b] Guo F, Zhang D, Dong Y and Guo Z 2019 Urban link travel speed dataset from a megacity road network Sci. Data 6 61

[LFW+15] Li D, Fu B, Wang Y, Lu G, Berezin Y, Stanley H E and Havlin S 2015 Percolation transition in dynamical traffic network with evolving critical bottlenecks Proc. Natl Acad. Sci. USA 112 669–72

[Tom] TomTom. 2019 (https://www.tomtom.com/en_gb/traffic-index/ranking/)

[Tra15] Traag V A 2015 Faster unfolding of communities: speeding up the Louvain algorithm Phys. Rev. E 92 032801

[Tur19] Turner K 2019 Rips filtrations for quasimetric spaces and asymmetric functions with stability results Algebr. Geom. Topol. 19 1135–70

[WSK+17] Wu Y, Shindnes G, Karve V, Yager D, Work D B, Chakraborty A and Sowers R B 2017 Congestion barcodes: exploring the topology of urban congestion using persistent homology 2017 IEEE 20th Int. Conf. on Intelligent Transportation Systems (ITSC) (IEEE) pp 1–6

[ZC05] Zomorodian A and Carlsson G 2005 Computing persistent homology Discrete Comput. Geom. 33 249–74

[Zom10] Zomorodian A 2010 Fast construction of the Vietoris–Rips complex Comput. Graph. 34 263–71

[ZOXY16] Zhu Y, Ozbay K, Xie K and Yang H 2016 Using big data to study resilience of taxi and subway trips for Hurricanes Sandy and Irene Transp. Res. Rec. 2599 70–80

[ZUZ14] Zhan X, Ukkusuri S V and Zhu F 2014 Inferring urban land use using large-scale social media check-in data Netw. Spatial Econ. 14 647–67