Comparing Embedded and Immersed Graphs

Carola Wenk
Department of Computer Science
Tulane University
Collaborators:
(Highly incomplete list)

- Mahmuda Ahmed
  Google

- Kevin Buchin
  TU Eindhoven

- Maike Buchin
  U. Bochum

- Brittany Fasy
  Montana State U.

- Erin Chambers
  Saint Louis U.

- Dieter Pfoser
  George Mason U.

- Elena Fasano
  Università di Roma Tor Vergata

- Sophia Karagiorgou
  Ubitech

- Rodrigo Silveira
  U. Poli. de Catalunia

- Pan Fan
  Tulane U.

- Majid Mirzanezhad
  Tulane U.

- Liz Munch
  Michigan State U.

- Erfan Hosseini
  Tulane U.

- Mahmuda Ahmed
  Google

- Matthieu Simeoni
  Università di Roma Tor Vergata

- Britta Seeger
  U. Bochum

- Onion Center

Websites:

- cs.tu-berlin.de/~carola
- mapconstruction.org

Book:

- Map Construction Algorithms

Grant support:

NSF CCF-1618469
and CCF-1637576
Outline

1. 1D embedded data: Curves and embedded & immersed graphs
2. Hausdorff and Fréchet-like distances:
   - Hausdorff distance
   - Fréchet distance
   - Path-based distance
   - Traversal distance
   - Strong/weak graph distance
   - Contour tree distance
3. Local persistent homology distance and local signatures
4. Other distances
   - Edit distance for geometric graphs
   - Shortest path sampling distance
   - Point sampling distance
1. 1D Embedded Data
1D Embedded Data
embedded in abient (usually Euclidean) space

- Want to compare such 1D embedded data
  ⇒ Geometric shapes
  - There are lots of distance measures and algorithms for comparing curves, and some for trees. But not so many for embedded (geometric) graphs.
  - Graphs are the most general 1D shapes.
Curves

• A curve is a continuous map $f: [0,1] \rightarrow \mathbb{R}^d$

• Many different curves can have the same image.

• We can reparameterize curves: $f \circ \alpha: [0,1] \rightarrow \mathbb{R}^d$, where $\alpha: [0,1] \rightarrow [0,1]$ is a reparameterization.
Polygons, Curves, & Trajectories

- Polygonal curves consist of a finite number of line segments and vertices. They can be specified by a sequence of points \( \langle p_0, ..., p_{n-1} \rangle \).

- We typically endow a polygonal curve with its arc-length parameterization \( f: [0,1] \rightarrow \mathbb{R}^d \). On each edge \( p_i p_{i+1} \) this is a linear function, hence a piecewise linear function overall.

- A (geospatial) trajectory is a sequence of time-stamped position samples.
Embedded/Immersed Graphs

- Graph $G = (V, E)$ with a set of vertices $V$ and edges $E$.
- Road network: Planar embedded

- Can consider $G$ as a topological space (e.g., 1D simplicial complex)
- **Embedded graph**: Have a continuous function $\phi: G \rightarrow \mathbb{R}^d$, $d \geq 2$, that is homeomorphic onto its image.
- **Immersed graph**: $\phi: G \rightarrow \mathbb{R}^d$ is only **locally** homeomorphic onto its image.
Embedded/Immersed Graphs

- **Embedded graph**: Have a continuous function \( \phi: G \to \mathbb{R}^d \), \( d \geq 2 \), that is homeomorphic onto its image.

- **Immersed graph**: \( \phi: G \to \mathbb{R}^d \) is only **locally** homeomorphic onto its image.

=> Each vertex is mapped to a point and edges are mapped to curves in \( \mathbb{R}^d \) in such a way that the graph structure is maintained.

  - Homeomorphism: A continuous, bijective map whose inverse is continuous.

**Embedding:**
all edge-curves are non-crossing (every crossing is a vertex)

**Immersion:**
“Bridges” are allowed

- \( \mathbb{R}^2 \): planar graphs vs. plane (= planar embedded) graphs
- Assume edge curves are piecewise linear, and may ignore deg-2 vertices
Immersed Graph Comparison

Given two immersed graphs $G = (V_G, E_G, \phi_G)$ and $H = (V_H, E_H, \phi_H)$, we want to compare them.

- How similar / different are they?
- What does it mean to be similar?
  - Depends on the application.
  - Graph isomorphism?

Here: Assume $G$ and $H$ are embedded in the same space and aligned.

1. Define different distances between $G$ and $H$, and study their properties and computational complexities.
2. Compute correspondences between portions of $G$ and $H$.
3. Consider local distance signatures (heatmaps).
Graph Isomorphism

• An isomorphism of $G = (V_G, E_G)$ and $H = (V_H, E_H)$ is a
  - bijective map $f: V_G \rightarrow V_H$ for which holds
  - $\{u, v\} \in E_G \iff \{f(u), f(v)\} \in E_H$

  Can be computed in linear time for planar graphs [HW74]

• Subgraph isomorphism: An isomorphism between $G$ and a subgraph of $H$
  - NP-complete
  - Can be computed in linear time if $G$ and $H$ are planar and $G$ has constant complexity [E95]

• Isomorphisms are bijective (1-to-1). However, we may want to allow 1-to-many assignments.
• We may also want to allow partial matchings.
• Isomorphisms are combinatorial in nature and don’t take the embeddings/immersions into account.

Compare Reconstructed Roadmaps

GPS Trajectory Data

Reconstructed Roadmaps
Compare Reconstructed Roadmaps
Compare Reconstructed Roadmaps

• How can one measure the quality of constructed maps?
• Surprisingly, there is no applicable ground truth map:
  – Professional maps
  – Do not cover the same area and the same details as a given input set of trajectories

⇒ Compare two immersed graphs
2. Hausdorff and Fréchet-Like Distances
Hausdorff Distance

- Directed Hausdorff distance
  \[ \delta_H(A, B) = \max_{a \in A} \min_{b \in B} || a - b || \]
- Undirected Hausdorff-distance
  \[ \delta_H(A, B) = \max(\delta_H(A, B), \delta_H(B, A)) \]
- Can be computed in polynomial time; O(N log N) in the plane

**Con:** When applied to graph comparison, \( \delta_H \) only compares the geometry but not the topology

**Pro:** \( \delta_H \) allows for partial comparison of one graph

- \( \delta_H \) is a metric on the set of compact subsets of \( \mathbb{R}^d \)
Metric Properties

Definition 1 (Key Properties of Dissimilarity Functions). Let $\mathbb{X}$ be a set. Consider a function $d: \mathbb{X} \times \mathbb{X} \rightarrow \mathbb{R}_{\geq 0}$. We define the following properties:

1. Identity: $d(x, x) = 0$.
2. Symmetry: for all $x, y \in \mathbb{X}$, $d(x, y) = d(y, x)$.
3. Separability: for all $x, y \in \mathbb{X}$, $d(x, y) = 0$ implies $x = y$.
4. Subadditivity (Triangle Inequality): for all $x, y, z \in \mathbb{X}$, $d(x, y) \leq d(x, z) + d(z, y)$.

- **Metric**: Fulfills 1.-4.
- **Directed**: Does not fulfill 2.
- **Pseudo-metric**: 1., 2., 4.
- **Semi-metric**: 1., 2., 3.
- **Quasi-metric**: 1., 3., 4.

$\Rightarrow \overrightarrow{\delta_H}$ is a directed pseudo-metric
Fréchet Distance for Curves

\[ \delta_F(f,g) = \inf_{\alpha,\beta:[0,1] \rightarrow [0,1]} \max_{t \in [0,1]} ||f(\alpha(t)) - g(\beta(t))|| \]

where \( \alpha \) and \( \beta \) range over continuous monotone increasing reparameterizations only.

- Man and dog walk on one curve each
- They hold each other at a leash
- They are only allowed to go forward
- \( \delta_F \) is the minimal possible leash length

Free Space Diagram

- Let $\varepsilon > 0$ fixed (eventually solve decision problem)
- $F_\varepsilon(f,g) = \{ (s,t) \in [0,1]^2 | \| f(s) - g(t) \| \leq \varepsilon \}$ \textit{white points} \textit{free space} of $f$ and $g$
- The free space in one cell is an ellipse.
Monotone path encodes reparametrizations of $f$ and $g$

$\delta_F(f,g) \leq \varepsilon$ iff there is a monotone path in the free space from $(0,0)$ to $(1,1)$

Such a path can be computed using DP in $O(mn)$ time
- Monotone path encodes reparametrizations of $f$ and $g$
- $\delta_F(f,g) \leq \varepsilon$ iff there is a monotone path in the free space from $(0,0)$ to $(1,1)$
- Such a path can be computed using DP in $O(mn)$ time
Weak Fréchet Distance

- Weak Fréchet distance $\delta_{wf}(f,g)$: Allow any continuous reparameterizations $\alpha$ and $\beta$
  $\Rightarrow$ Any continuous path in free space (not necessarily monotone)
- $\delta_{h}(f,g) \leq \delta_{wf}(f,g) \leq \delta_{F}(f,g)$
Map-Matching

**Given:** A graph $G$, a curve $l$, and a distance parameter $\varepsilon$.

**Task:** Find a path $\pi$ in $G$ such that $\delta_F(l,\pi) \leq \varepsilon$

Compute free space surface.
And find path $\pi'$ in it

Such a path can be computed using DP in $O(mn)$ time
Fréchet Distance, General

Let $A, B \subseteq \mathbb{R}^k$ be two oriented manifolds. And let $f: A \to \mathbb{R}^d$ and $g: B \to \mathbb{R}^d$ be two immersions. Then

$$\delta_F(f, g) = \inf_\alpha \max_{t \in A} \|f(t) - g(\alpha(t))\|,$$

where $\alpha: A \to B$ ranges over all orientation-preserving homeomorphisms.

- The Fréchet distance is a pseudo-metric (separability is not fulfilled, since shapes with different parameterizations can have distance 0).
- Originally defined for oriented manifolds, but can be generalized even further.
Fréchet Distance, Immersed Graphs

Let $G = (V_G, E_G, \phi_G)$ and $H = (V_H, E_H, \phi_H)$ be two immersed graphs.

- We can apply the Fréchet distance definition in principle on the maps $\phi_G$ and $\phi_G$.
- Drop the "orientation-preserving" requirement.
- Equivalent definition:
  \[
  \delta_F(G, H) = \inf \max_{\alpha} \delta_F(e, \alpha(e)),
  \]
  where $\alpha$ ranges over all edge mappings corresponding to isomorphisms of $G$ and $H$.

- Is graph-isomorphism hard. Can be computed in poly time for trees and for graphs of bounded tree-width. [BKN20]
- For planar graphs, can enumerate orientation-preserving isomorphisms in polynomial time. [FW21]

Path-Based Distance

• Directed Hausdorff distance on path-sets:

\[
\overrightarrow{d}_{G,H}(\pi_G, \pi_H) = \max_{p_G \in \pi_G} \min_{p_H \in \pi_H} \delta_F(p_G, p_H)
\]

• \(\pi_G\) path-set in G, and \(\pi_H\) path-set in H

• Asymmetry of distance definition is desirable, if G is a reconstructed map and H a ground-truth map.

Fréchet distance

Path-Based Distance

- Ideally, $\pi_G$ and $\pi_H$ are the set of all paths in $G$ and $H$

$$\vec{d}_{G,H}(\pi_G, \pi_H) = \max_{p_G \in \pi_G} \min_{p_H \in \pi_H} \delta_F(p_G, p_H)$$

- It is a directed pseudo-metric.

- One can use the set of paths of link-length three to approximate the overall distance in polynomial time, if vertices in $G$ are well-separated and have degree $\neq 3$.
  → Stitch link-length three paths together to form longer paths

Traversal Distance

Let $G = (V_G, E_G, \phi_G)$ and $H = (V_H, E_H, \phi_H)$ be two immersed graphs.

- Represent $G$ by traversals $f: [0,1] \rightarrow G$ (continuous, surjective) and $H$ by partial traversals $g: [0,1] \rightarrow H$:
  \[
  \vec{d}_T(G, H) = \inf_{f, g} \max_{t \in [0,1]} ||f(t) - g(t)||
  \]

- Can be computed in $O(mn \log mn)$ time using free space diagram.
- Is a directed distance, but fulfills neither separability nor triangle inequality.
- Concides with the weak Fréchet distance when $G$ and $H$ are polygonal curves.

Traversal Distance

Fig. 1: Example showing that the traversal distance violates the separability and triangle inequality. Assume all graphs lie on top of each other, i.e., \( G_1 \) and \( G_3 \) are subgraphs of \( G_2 \). Then \( \overrightarrow{d_T}(G_1, G_2) = 0, \overrightarrow{d_T}(G_2, G_3) = w/2 \), but \( G_1 \neq G_2 \) and \( \overrightarrow{d_T}(G_1, G_3) = w > w/2 \).

Strong and Weak Graph Distances

Let $G = (V_G, E_G, \phi_G)$ and $H = (V_H, E_H, \phi_H)$ be two immersed graphs.

- Define a graph mapping $s: G \rightarrow H$ as follows:
  - $s$ sends each $v \in V_G$ to a point $s(v) \in H$
  - $s$ sends each $e \in E_G$ to a simple path from $s(u)$ to $s(v)$ in $H$.

- Then the strong graph distance is
  \[ \delta \rightarrow (G, H) = \inf_{s: G \rightarrow H} \max_{e \in E_G} \delta_F(e, s(e)) \]

- The weak graph distance $\delta \rightarrow_w$ uses $\delta_{wF}$ instead of $\delta_F$.

- We have $d_T(G, H) \leq \delta \rightarrow_w(G, H) \leq \delta \rightarrow(G, H)$

- NP-hard to decide, but can be computed in poly time for trees, and the weak graph distance can be computed in poly time for planar embedded graphs.

Traversing and Graph Distance

- **Small traversal distance**

Traversal and Graph Distance

- Small traversal distance
- Large graph distance

- Small traversal distance
Traversals and Graph Distance

- Small traversal distance
- Large graph distance

- Small traversal distance
- Large graph distance
- Both small

Strong and Weak Graph Distances

(a) $G_1$ and $G_2$

(b) $G_1$ and $G_2$

(c) $G_1$ and $G_2$

[ABKSW21] H.A. Akitaya, M. Buchin, B. Kilgus, S. Sijben, C. Wenk, Distance Measures for Embedded Graphs, CGTA 95, 101743, 2021. 34/52
Contour Tree Distance

Let $G = (V_G, E_G, \phi_G)$ and $H = (V_H, E_H, \phi_H)$ be two connected immersed graphs.

- The contour tree distance is
  \[
  d_C(G, H) = \inf_{\tau} \sup_{(x,y) \in \tau} \|x - y\|,
  \]
  where $G$ ranges over all correspondences $\tau$ between $G$ and $H$ such that
  1. $\tau \subseteq G \times H$ is connected
  2. For each $x \in G$: The set $\tau \cap \{x\} \times H$ is non-empty and connected
  3. For each $y \in H$: The set $\tau \cap (G \times \{y\})$ is non-empty and connected

Contour Tree Distance

\[ d_c(G, H) = \inf_{\tau} \sup_{(x,y) \in \tau} \|x - y\|, \]

where \( G \) ranges over all correspondences \( \tau \) between \( G \) and \( H \) such that

1. \( \tau \subseteq G \times H \) is connected
2. For each \( x \in G \): The set \( \tau \cap (\{x\} \times H) \) is non-empty and connected
3. For each \( y \in H \): The set \( \tau \cap (G \times \{y\}) \) is non-empty and connected
Contour Tree Distance

- The contour tree distance is a metric.
- But it is NP-complete, already for trees.
- This distance seems to correspond to a symmetric version of the (strong or weak) graph distances.

3. Local Persistent Homology Distance and Local Signatures
Excursion into Computational Topology: Persistent Homology

• Develop topological descriptors to analyze point set shapes

• It looks like this shape contains two cycles. But how do we know?

• Let’s make the points thicker:

Adapted from Tamal Dey’s slides http://ww2.ii.uj.edu.pl/wsocm/slides/DEY.pdf
Persistent Homology

- \( f(x) = d(x, P) \): distance to point cloud \( P \)
- **Sublevel sets** \( f^{-1}[0, r] \) are union of balls
- Evolution of the sublevel sets with increasing radius \( r \)
  \( \Rightarrow \) The left hole **persistence** longer
- Growing union of balls are nested topological spaces
  \( \Rightarrow \) a **filtration**
  \( \Rightarrow \) persistent homology classes (groups)

Adapted from Tamal Dey’s slides [http://ww2.ii.uj.edu.pl/wsocm/slides/DEY.pdf](http://ww2.ii.uj.edu.pl/wsocm/slides/DEY.pdf)
• $f(x) = d(x, P)$: distance to point cloud $P$
• Sublevel sets $f^{-1}[0, r]$ are union of balls

$Dgm(f, P)$ is the **persistence diagram** of $P$
• Each point in $Dgm(f, P)$ is a pair of $r$-values: (birth, death)
• $\Rightarrow$ **Topological descriptor of $P$**

Adapted from Tamal Dey’s slides [http://ww2.ii.uj.edu.pl/wsocm/slides/DEY.pdf](http://ww2.ii.uj.edu.pl/wsocm/slides/DEY.pdf)
The **bottleneck distance** between two diagrams $Dgm_1$ and $Dgm_2$ is

$$d_b(Dgm_1, Dgm_2) = \inf_{\gamma \in \Gamma} \sup_{p \in Dgm_1} \|p - \gamma(p)\|_\infty$$

where $\Gamma$ is the set of all the bijections between $Dgm_1$ and $Dgm_2$ and

$$\|p - q\|_\infty = \max(|x_p - x_q|, |y_p - y_q|).$$

Local Persistent Homology Distance

• Consider a common local neighborhood of both maps.
• Consider the cycles of each graph inside this neighborhood.
• Now thicken each graph and track changes in the cycle structure using persistent homology

⇒ Use (bottleneck) distance between persistence diagrams to compare changing local cycle structure

Local Persistent Homology Distance

- **Local “signature”** that captures local topological similarity of graphs: 
  \[ \psi_r(x) = d(P_{1,x,r}, P_{2,x,r}) \]
  where \( d \) is the bottleneck distance between the two persistence diagrams

- Fixed radius:
  \[ d_{r}^{LH}(G_1, G_2) = \frac{1}{|X|} \int_{X} \psi_r(x) \, dx \]

- Local homology metric:
  \[ d^{LH}(G_1, G_2) = \frac{1}{r_1|X|} \int_{0}^{r_1} \omega(r) \int_{X} \psi_r(x) \, dx \, dr \]

Local Persistent Homology Distance

- Compared two reconstructed maps.
- Disk centers sampled 5m; disk radius 25m
- Local signature captures different topology (missing intersections) well

4. Other Distances
Geometric Edit Distance

- Geometric Edit Distance [CGKSS09]
  - Defined for straight-line embedded graphs.
  - Motivated by Chinese character comparison
  - Perform the following edit operations in this order: Edge deletion, vertex deletion, vertex translation, vertex insertion, edge insertion
  - Costs are proportional to edge lengths and to the distance a vertex has been translated.
  - Is a metric. But NP-hard.

Shortest Path Sampling Distance

- Shortest Path Sampling Distance [KP12] in $\mathbb{R}^2$:
  - Randomly sample $x, y \in \mathbb{R}^2$
  - Find nearest neighbors $x_G, y_G$ on $G$ and compute a shortest path $\pi_G$ from $x_G$ to $y_G$ in $G$.
  - Similarly, compute a shortest $\pi_H$ from $x_H$ to $y_H$ in $H$.
  - Compute $\delta_F(\pi_G, \pi_H)$.
  - Repeat for several random samples, and compare sets of resulting distances.

Point Sampling Distance

- In a local neighborhood of both graphs, traverse the graphs (from random seeds) and place point samples. (Only graph edges of length $\leq \tau$.)

- $\tau$: match_distance threshold
  $m = m(\tau)$: #samples in $G$
  $n = n(\tau)$: #samples in $H$
  $k = k(\tau) = \#$matched samples (1-1) within distance $\tau$

- **Precision:** $p = k/n$  **Recall:** $r = k/m$  **F-score:** $2pr/(p+r) = 2k/(n+m)$

**Point Sampling Distance**

\[ G = \text{OSM ground-truth: } m \text{ samples}; \quad H = \text{constructed map: } n \text{ samples} \]

\[ \frac{2k}{(n+m)} \quad p = \frac{k}{n} \]

Biagioni and Karagiorgou: F-score decreases, precision increases

→ More matched samples (k), more (unmatched) ground-truth samples (m)

### Chicago

![Graph showing F-score comparison between different generated maps in Chicago]
Point Sampling Distance

• Can also be used as a **local distance signature**.
• Lacks theoretical foundation but is practical.
• Does not work well if the reconstructed graph is compared with more a detailed ground-truth graph (e.g., OSM).
• Provides a **matching (1-to-1)** between a subset of points in $G$ and $H$

• What is a good matching?
• Can one define this continuously (and compute/approximate efficiently)?
Conclusion & Discussion

1. We’ve seen a lot of distances for immersed graphs.
   • Are they useful in practice? (Noisy input, runtimes)
   • What are their mathematical properties? (Metric, topological)
2. Would like to compute a correspondence / mapping between the two graphs efficiently.
   • An application: Merge multiple road networks
3. Optimize under transformations
4. Local signatures:
   • Useful to identify local differences
   • Compute global correspondence from local correspondences?