Comparing Temporal Graphs

with Time Warping

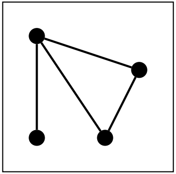
Malte Renken

Algorithmics and Computational Complexity,
TU Berlin, Germany

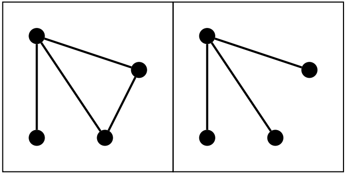
8. July 2019

Joint work with
Vincent Froese, Brijnesh Jain and Rolf Niedermeier.

Temporal Graphs

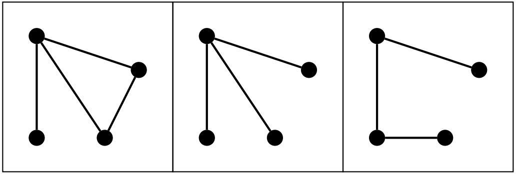


Temporal Graphs



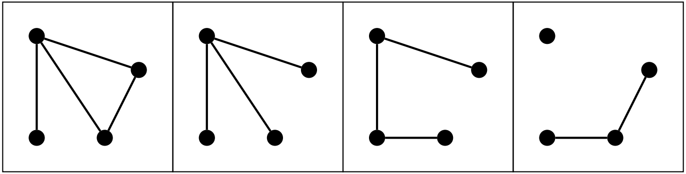
$t = 1$ $t = 2$

Temporal Graphs



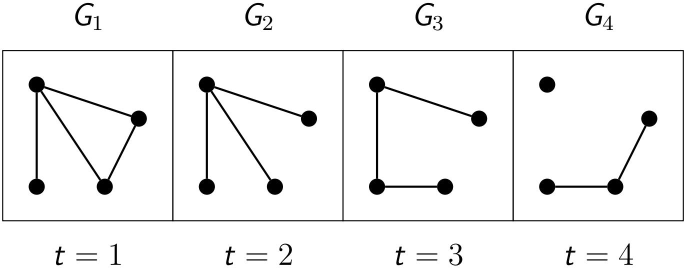
$t = 1$ $t = 2$ $t = 3$

Temporal Graphs

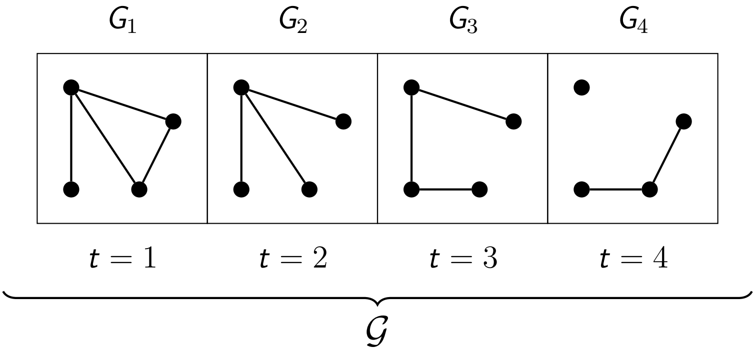


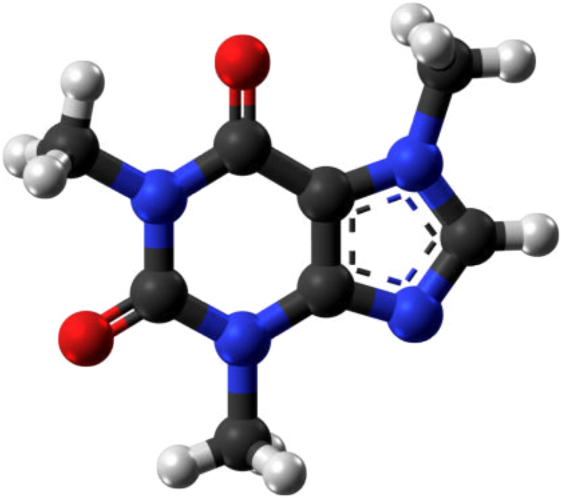
$t = 1$ $t = 2$ $t = 3$ $t = 4$

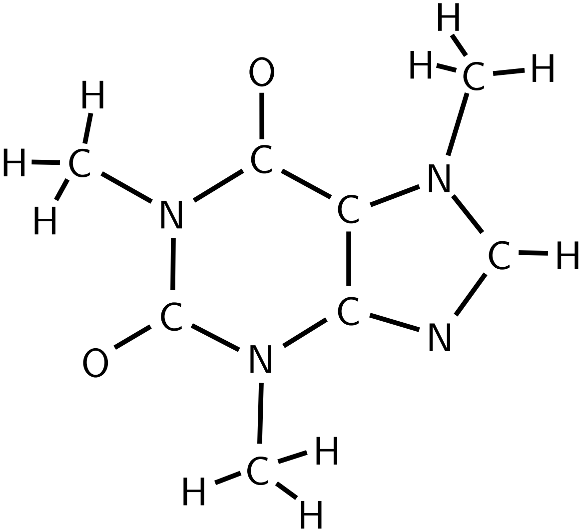
Temporal Graphs



Temporal Graphs






H
H
C
H
O
H
H
C
C
N
H
N
C
C
H
C
N
C
O
C
N
N
H
C
H
H

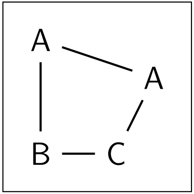
$$\overset{?}{\cong}$$

Main Question

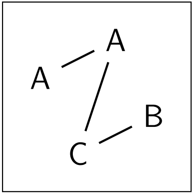
How can we measure the similarity / distance between two temporal graphs $\mathcal{G}$, $\mathcal{H}$?

Graph distance using vertex signatures

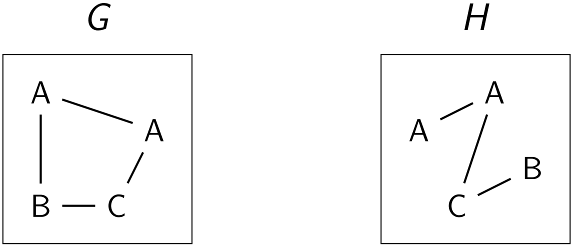
G



H



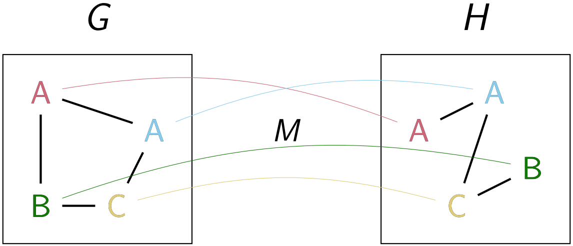
Graph distance using vertex signatures



Example vertex signatures:

$\sigma(v) = ($label of v, labels of v's neighbors$)$

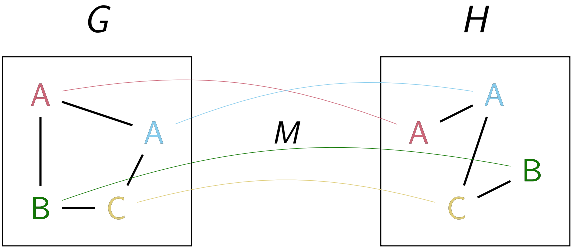
Graph distance using vertex signatures



Example vertex signatures:

$\sigma(v) = (\text{label of } v, \text{labels of } v\text{'s neighbors})$

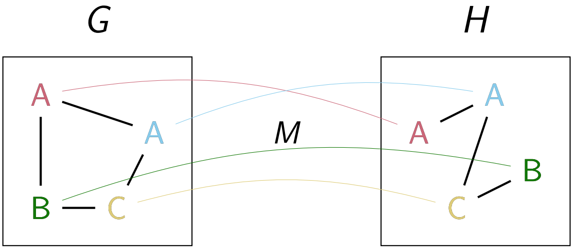
Graph distance using vertex signatures



Example vertex signatures:
$\sigma(v) = (\text{label of } v, \text{labels of } v\text{'s neighbors})$

$$\mathrm{dist}(G,H) = \underbrace{\sum_{(v,w)\in M} \|\sigma(v) - \sigma(w)\|}_{C_M(G,H)}$$

Graph distance using vertex signatures



Example vertex signatures:
$\sigma(v) = (\text{label of } v, \text{labels of } v\text{'s neighbors})$

$$\operatorname{dist}(G, H) = \min_{M \in \mathcal{M}} \underbrace{\sum_{(v,w) \in M} \|\sigma(v) - \sigma(w)\|}_{C_M(G,H)}$$

all maximal matchings between the two vertex sets

Graph distance using vertex signatures

$$\mathrm{dist}(G, H) = \min_{M \in \mathcal{M}} C_M(G, H)$$

Graph distance using vertex signatures

$$\mathrm{dist}(G, H) = \min_{M \in \mathcal{M}} C_M(G, H)$$

- Jouili and Tabbone (GbRPR 2009).

Graph distance using vertex signatures

$$\mathrm{dist}(G, H) = \min_{M \in \mathcal{M}} C_M(G, H)$$

- Jouili and Tabbone (GbRPR 2009).
- Computation in cubic time using Jonker-Volgenant (or Hungarian).

Time Warping

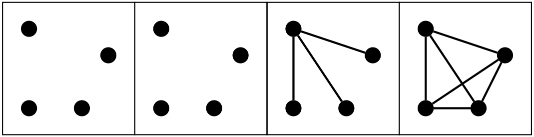
Problem: The two temporal graphs may have different lifetimes.

Even worse, they can have different (non-homogeneous) time scales.

Time Warping

Problem: The two temporal graphs may have different lifetimes.

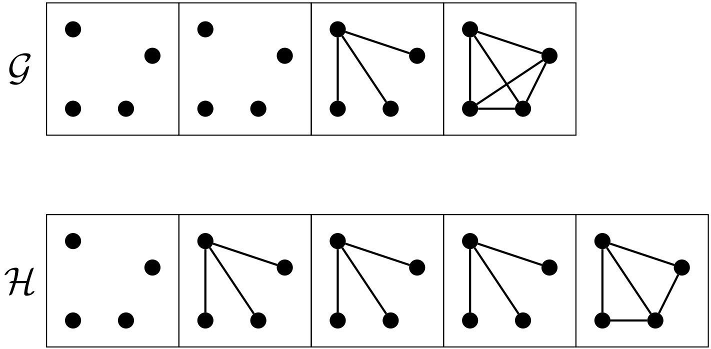
Even worse, they can have different (non-homogeneous) time scales.

$\mathcal{G}$ 

Time Warping

Problem: The two temporal graphs may have different lifetimes.

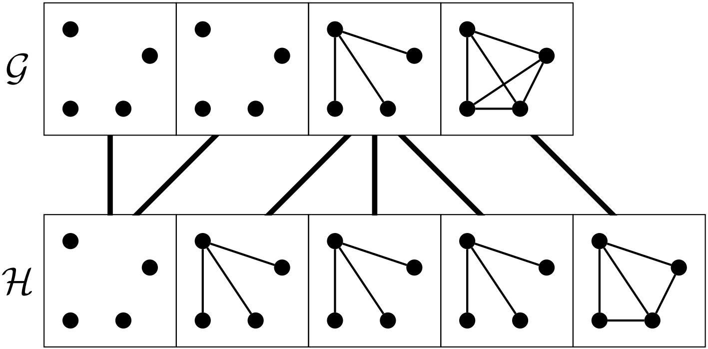
Even worse, they can have different (non-homogeneous) time scales.



Time Warping

Problem: The two temporal graphs may have different lifetimes.

Even worse, they can have different (non-homogeneous) time scales.



Solution: **Time warping** — assign each layer to the other one it resembles most (no crossings allowed!).

dtgw-distance

$$\operatorname{dist}(G, H) = \min_{M \in \mathcal{M}} C_M(G, H)$$

dtgw-distance

$$\text{dtgw-dist}(\mathcal{G}, \mathcal{H}) = \min_{W \in \mathcal{W}} \min_{M \in \mathcal{M}} \sum_{(t,u) \in W} C_M(G_t, H_u)$$

all time warpings between the two layer sets

dtgw-distance

$$\text{dtgw-dist}(\mathcal{G}, \mathcal{H}) = \min_{W \in \mathcal{W}} \min_{M \in \mathcal{M}} \sum_{(t,u) \in W} C_M(G_t, H_u)$$

all time warpings between the two layer sets

Good news: Time warping can be solved by a dynamic program in quadratic time ...

dtgw-distance

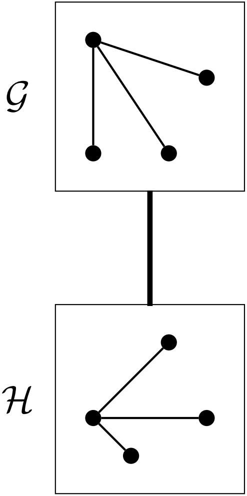
$$\text{dtgw-dist}(\mathcal{G},\mathcal{H}) = \min_{W \in \mathcal{W}} \min_{M \in \mathcal{M}} \sum_{(t,u) \in W} C_M(G_t, H_u)$$

all time warpings between the two layer sets

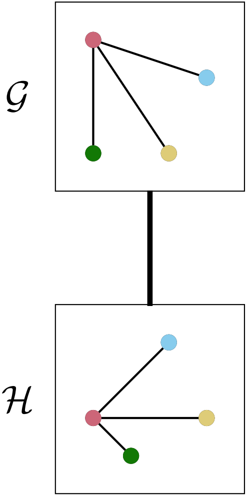
Good news: Time warping can be solved by a dynamic program in quadratic time ...

Bad news: ... if all pairwise distances are known in advance.

Pairwise distances

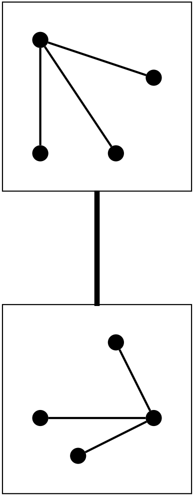


Pairwise distances



Pairwise distances

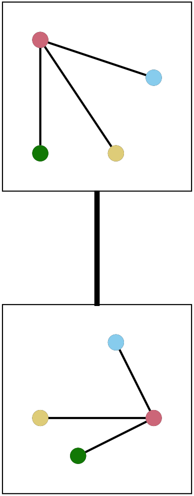
$\mathcal{G}$



$\mathcal{H}$

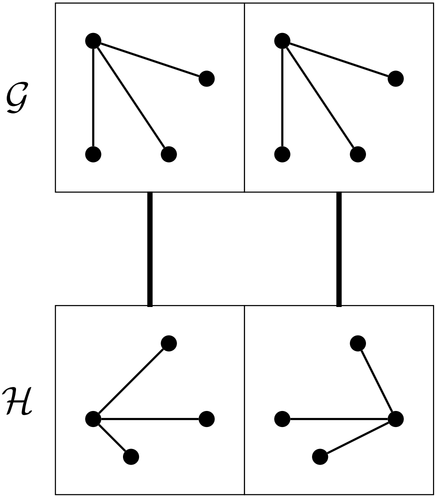
Pairwise distances

$\mathcal{G}$

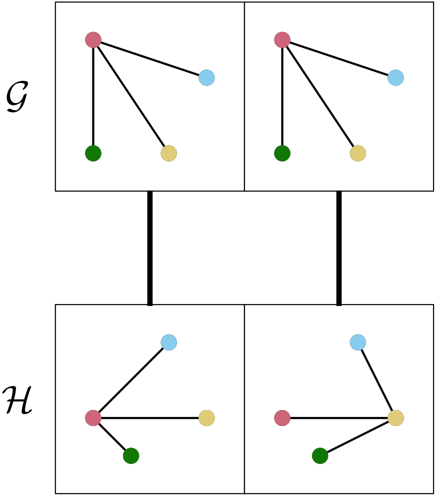


$\mathcal{H}$

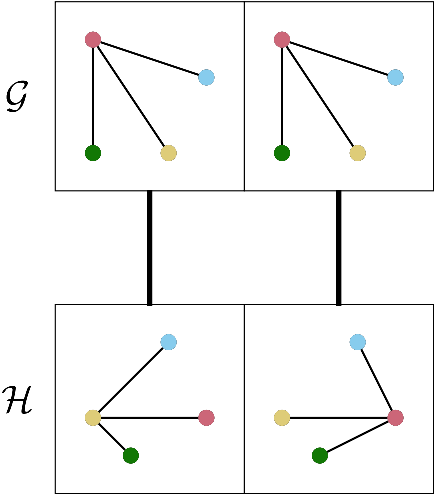
Pairwise distances

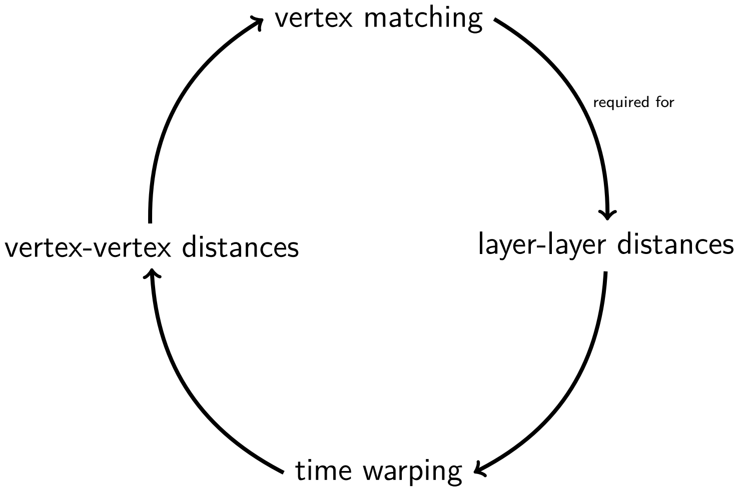


Pairwise distances



Pairwise distances




vertex matching
required for
layer-layer distances
time warping
vertex-vertex distances

Hardness

- Computing $\mathrm{dtgw\text{-}dist}(\mathcal{G}, \mathcal{H})$ is NP-hard ...

Hardness

- Computing $\mathrm{dtgw\text{-}dist}(\mathcal{G}, \mathcal{H})$ is NP-hard ...
- ... and probably[1] impossible in $2^{o(\#\mathrm{vertices} + \#\mathrm{layers} + \mathrm{dtgw\text{-}dist})}$

[1] assuming ETH

Hardness

- Computing $\mathrm{dtgw\text{-}dist}(\mathcal{G}, \mathcal{H})$ is NP-hard ...
- ... and probably[1] impossible in $2^{o(\#\mathrm{vertices} + \#\mathrm{layers} + \mathrm{dtgw\text{-}dist})}$
- ... even if you limit the time warping to be "nice"

[1] assuming ETH

Hardness

- Computing $\mathrm{dtgw\text{-}dist}(\mathcal{G}, \mathcal{H})$ is NP-hard ...
- ... and probably[1] impossible in $2^{o(\#\mathrm{vertices} + \#\mathrm{layers} + \mathrm{dtgw\text{-}dist})}$
- ... even if you limit the time warping to be "nice"
- ... even if your graphs have maximum degree one.

[1] assuming ETH

Hardness

- Computing $\mathrm{dtgw\text{-}dist}(\mathcal{G}, \mathcal{H})$ is NP-hard ...
- ... and probably[1] impossible in $2^{o(\#\mathrm{vertices} + \#\mathrm{layers} + \mathrm{dtgw\text{-}dist})}$
- ... even if you limit the time warping to be "nice"
- ... even if your graphs have maximum degree one.

But ...

[1] assuming ETH

Hardness

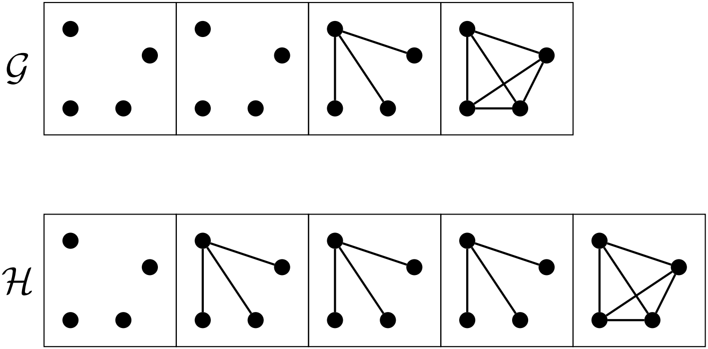
- Computing $\mathrm{dtgw\text{-}dist}(\mathcal{G}, \mathcal{H})$ is NP-hard ...
- ... and probably[1] impossible in $2^{o(\#\mathrm{vertices} + \#\mathrm{layers} + \mathrm{dtgw\text{-}dist})}$
- ... even if you limit the time warping to be "nice"
- ... even if your graphs have maximum degree one.

But ...

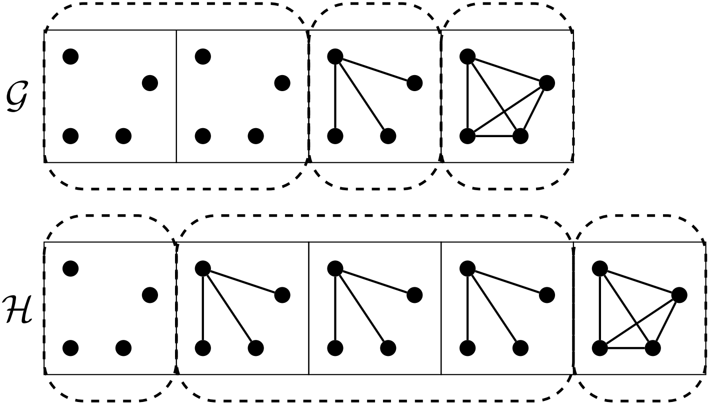
- ... you can check in polynomial time whether $\mathrm{dtgw\text{-}dist}(\mathcal{G}, \mathcal{H}) = 0$.

[1] assuming ETH

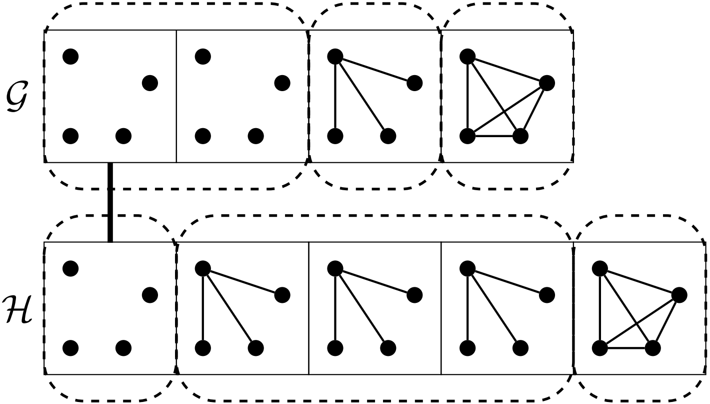
Testing for $\text{dtgw-dist}(\mathcal{G}, \mathcal{H}) = 0$



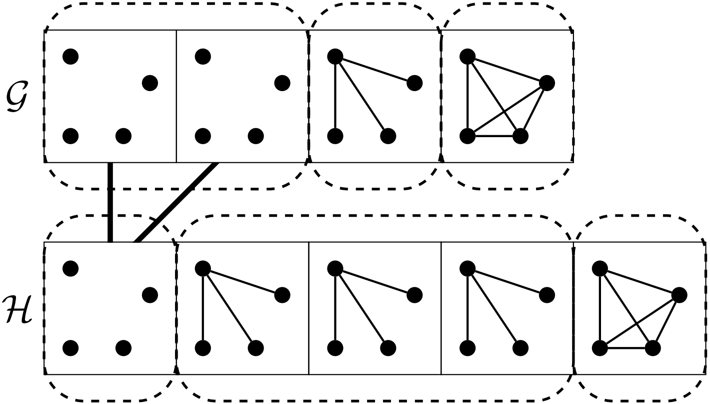
Testing for $\mathrm{dtgw\text{-}dist}(\mathcal{G}, \mathcal{H}) = 0$



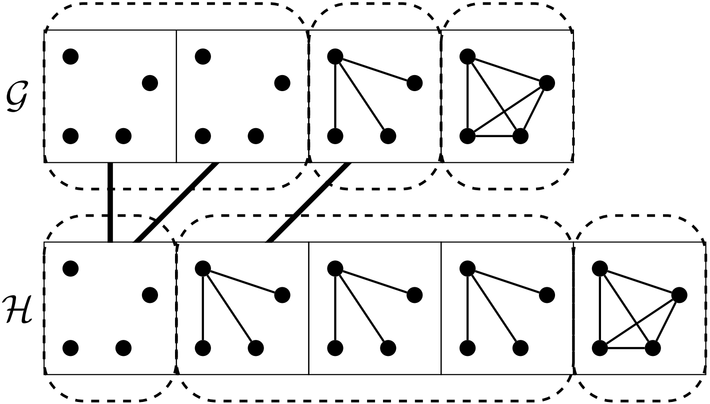
Testing for $\text{dtgw-dist}(\mathcal{G}, \mathcal{H}) = 0$



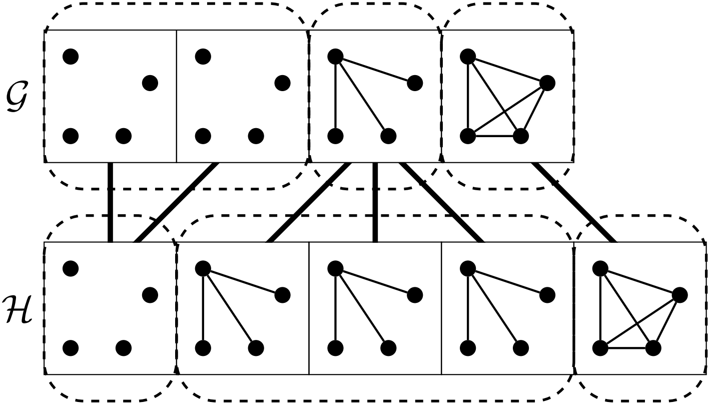
Testing for $\text{dtgw-dist}(\mathcal{G}, \mathcal{H}) = 0$



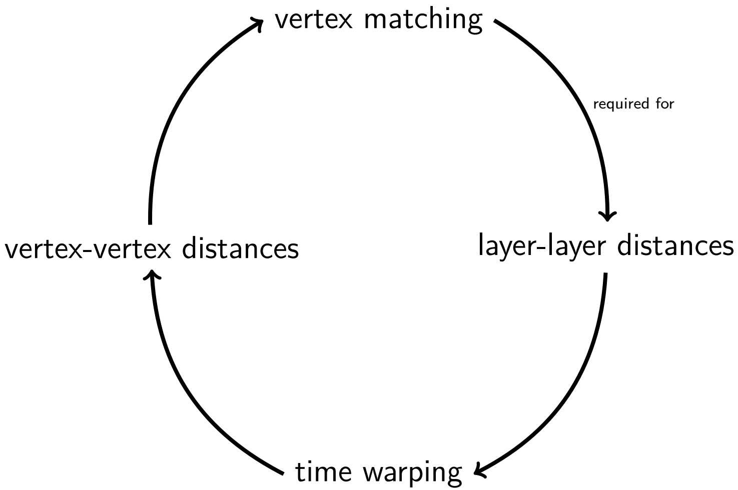
Testing for $\mathrm{dtgw\text{-}dist}(\mathcal{G}, \mathcal{H}) = 0$



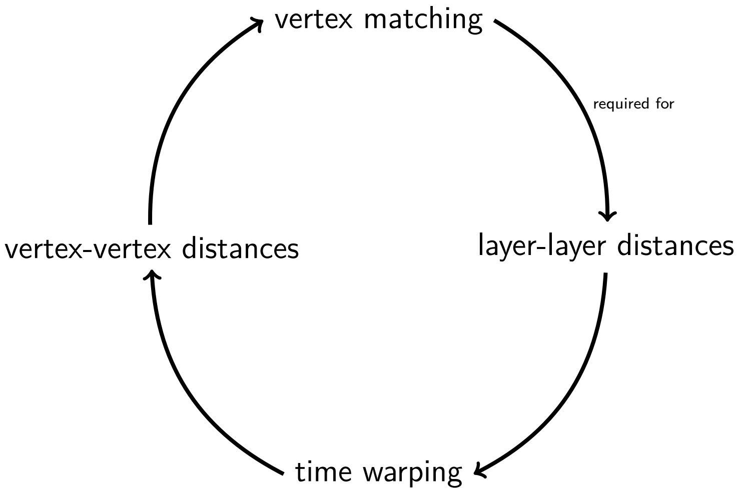
Testing for $\mathrm{dtgw\text{-}dist}(\mathcal{G}, \mathcal{H}) = 0$



Heuristic


vertex matching
required for
layer-layer distances
time warping
vertex-vertex distances

Heuristic



Make an educated guess on one of these and start cycling.

Heuristic

Make an educated guess on one of these and start cycling.

Heuristic

Make an educated guess on one of these and start cycling.

- Usually reaches a stable state after 2–5 rounds.

Heuristic

Make an educated guess on one of these and start cycling.

- Usually reaches a stable state after 2–5 rounds.
- Depends surprisingly little on your initial guess.

Heuristic

Make an educated guess on one of these and start cycling.

- Usually reaches a stable state after 2–5 rounds.
- Depends surprisingly little on your initial guess.
- Seems to produce good results.

Experiment: Noise

1. Take a couple of real world temporal graphs[2].

[2] sociopatterns.org

Experiment: Noise

1. Take a couple of real world temporal graphs[2].
2. Make multiple copies of each.

[2] sociopatterns.org

Experiment: Noise

1. Take a couple of real world temporal graphs[2].
2. Make multiple copies of each.
3. Add up to 30% noise (change edges, delete layers).

[2] sociopatterns.org

Experiment: Noise

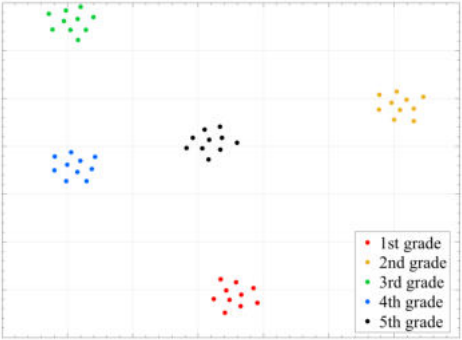
1. Take a couple of real world temporal graphs[2].
2. Make multiple copies of each.
3. Add up to 30% noise (change edges, delete layers).
4. Compute all pairwise distances with $\sigma(v) = \deg(v)$.

[2] sociopatterns.org

Experiment: Noise

1. Take a couple of real world temporal graphs[2].
2. Make multiple copies of each.
3. Add up to 30% noise (change edges, delete layers).
4. Compute all pairwise distances with $\sigma(v) = \deg(v)$.

Result



[2] sociopatterns.org

Experiment: De-anonymization

1. Vertices: People at a conference.

Experiment: De-anonymization

1. Vertices: People at a conference.
2. Two temporal graphs:
    2.1 A: edges = face-to-face contacts
    2.2 B: edges = proximity

Experiment: De-anonymization

1. Vertices: People at a conference.
2. Two temporal graphs:
   2.1 A: edges = face-to-face contacts
   2.2 B: edges = proximity
3. Compute dtgw-distance.

Experiment: De-anonymization

1. Vertices: People at a conference.
2. Two temporal graphs:
   2.1 A: edges = face-to-face contacts
   2.2 B: edges = proximity
3. Compute dtgw-distance.
4. Look at the vertex matching.

Experiment: De-anonymization

1. Vertices: People at a conference.
2. Two temporal graphs:
   2.1 A: edges = face-to-face contacts
   2.2 B: edges = proximity
3. Compute dtgw-distance.
4. Look at the vertex matching.

Result

- 86% of people correctly identified, using only $\sigma(v) = \deg(v)$.

Experiment: De-anonymization

1. Vertices: People at a conference.
2. Two temporal graphs:
   2.1 A: edges = face-to-face contacts
   2.2 B: edges = proximity
3. Compute dtgw-distance.
4. Look at the vertex matching.

Result

- 86% of people correctly identified, using only $\sigma(v) = \deg(v)$.
- Robust against misaligned times.

DTGW-distance

DTGW-distance

- ...can help you compare structures that change over time (aka temporal graphs).

DTGW-distance

- ...can help you compare structures that change over time (aka temporal graphs).
- ...is very hard in theory but mostly easy in practice.

DTGW-distance

- ...can help you compare structures that change over time (aka temporal graphs).
- ...is very hard in theory but mostly easy in practice.

Open questions

DTGW-distance

- ...can help you compare structures that change over time (aka temporal graphs).
- ...is very hard in theory but mostly easy in practice.

Open questions

- Is "$\mathrm{dtgw\text{-}dist}(\mathcal{G}, \mathcal{H}) \leq d$" decidable in $f(d) \cdot \mathrm{poly}(\mathcal{G}, \mathcal{H})$?

DTGW-distance

- ...can help you compare structures that change over time (aka temporal graphs).
- ...is very hard in theory but mostly easy in practice.

Open questions

- Is "$\mathrm{dtgw\text{-}dist}(\mathcal{G}, \mathcal{H}) \leq d$" decidable in $f(d) \cdot \mathrm{poly}(\mathcal{G}, \mathcal{H})$?
- Can you find approximation algorithms with guaranteed approximation quality?

DTGW-distance

- ...can help you compare structures that change over time (aka temporal graphs).
- ...is very hard in theory but mostly easy in practice.

Open questions

- Is "$\mathrm{dtgw\text{-}dist}(\mathcal{G}, \mathcal{H}) \leq d$" decidable in $f(d) \cdot \mathrm{poly}(\mathcal{G}, \mathcal{H})$?
- Can you find approximation algorithms with guaranteed approximation quality?
- Which vertex signatures work best in different settings?