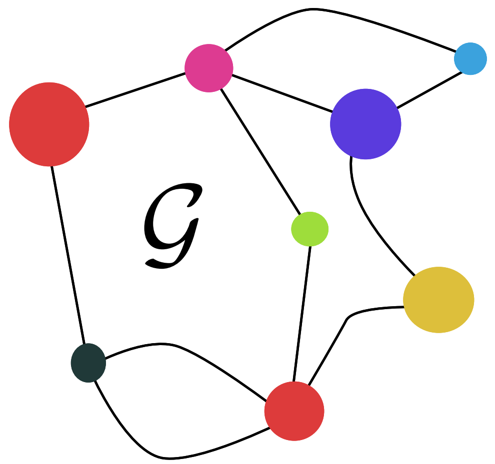
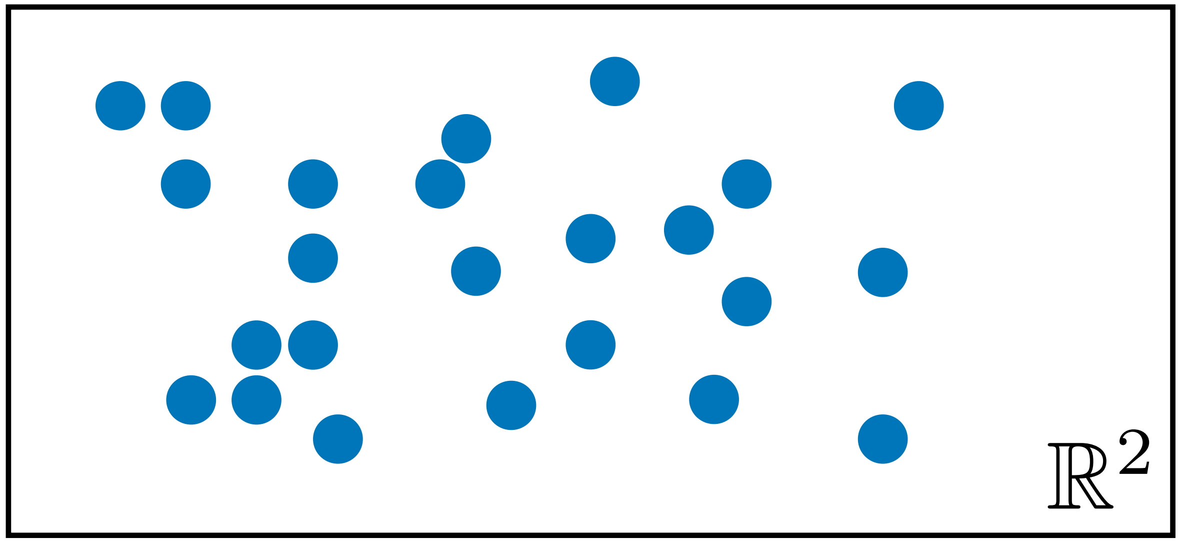
Optimal Transport for graph data

Titouan Vayer



In short:

Machine Learning: Learn to make decision from **data**



In short:

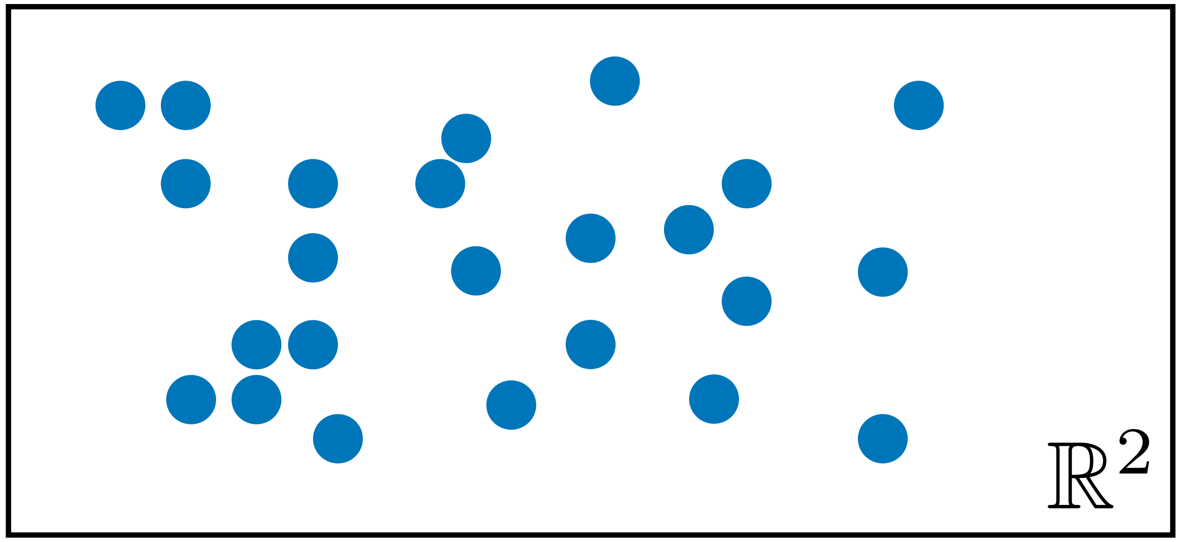
Mathematical representation

Tools which build upon this representation

Machine Learning: Learn to make decision from **data**

How to represent data?

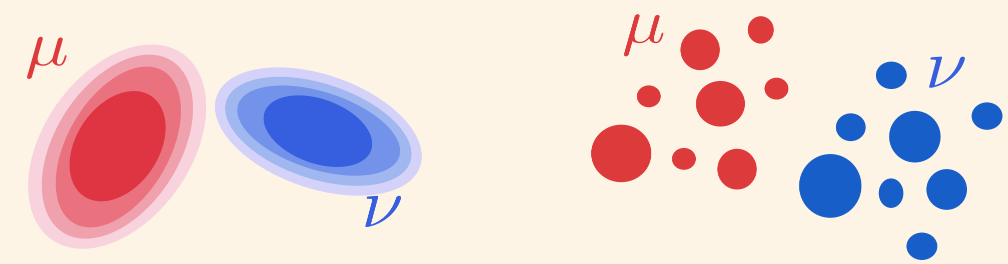
How to operate on them?



In short:

Mathematical representation

As probability distributions



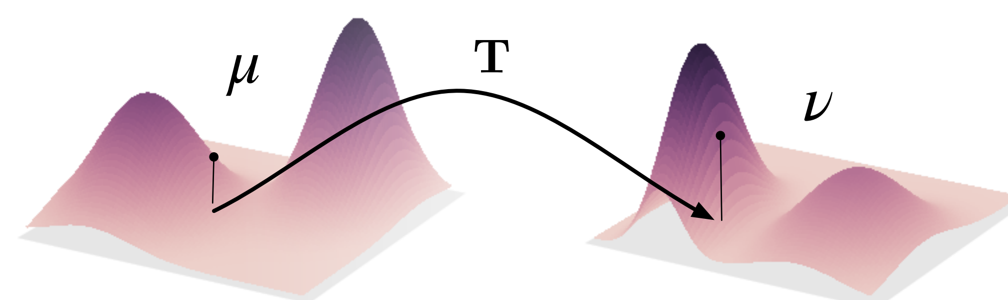
Machine Learning: Learn to make decision from **data**

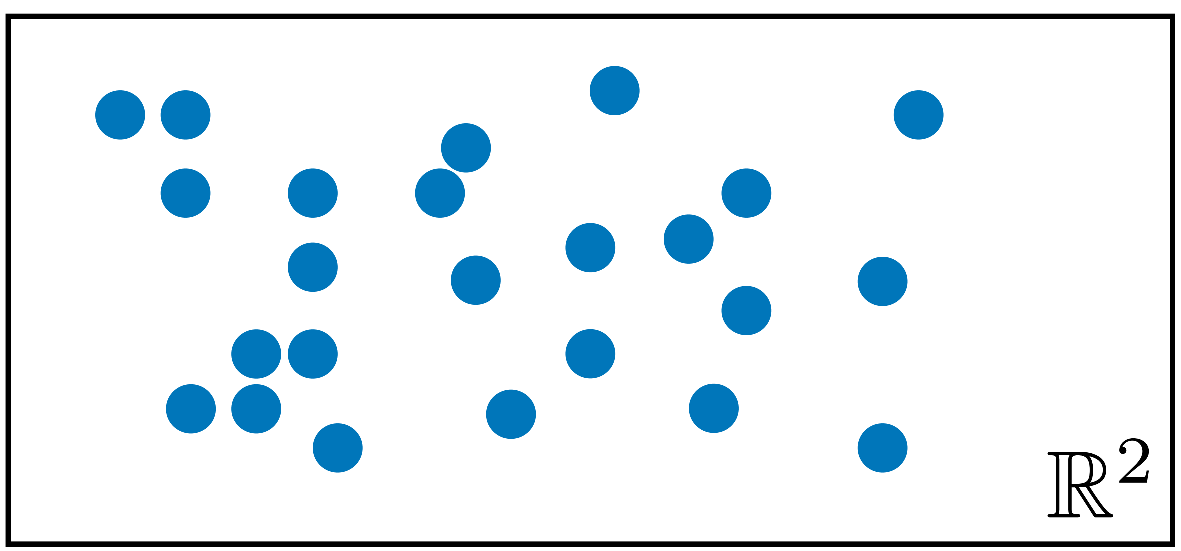
How to represent data?

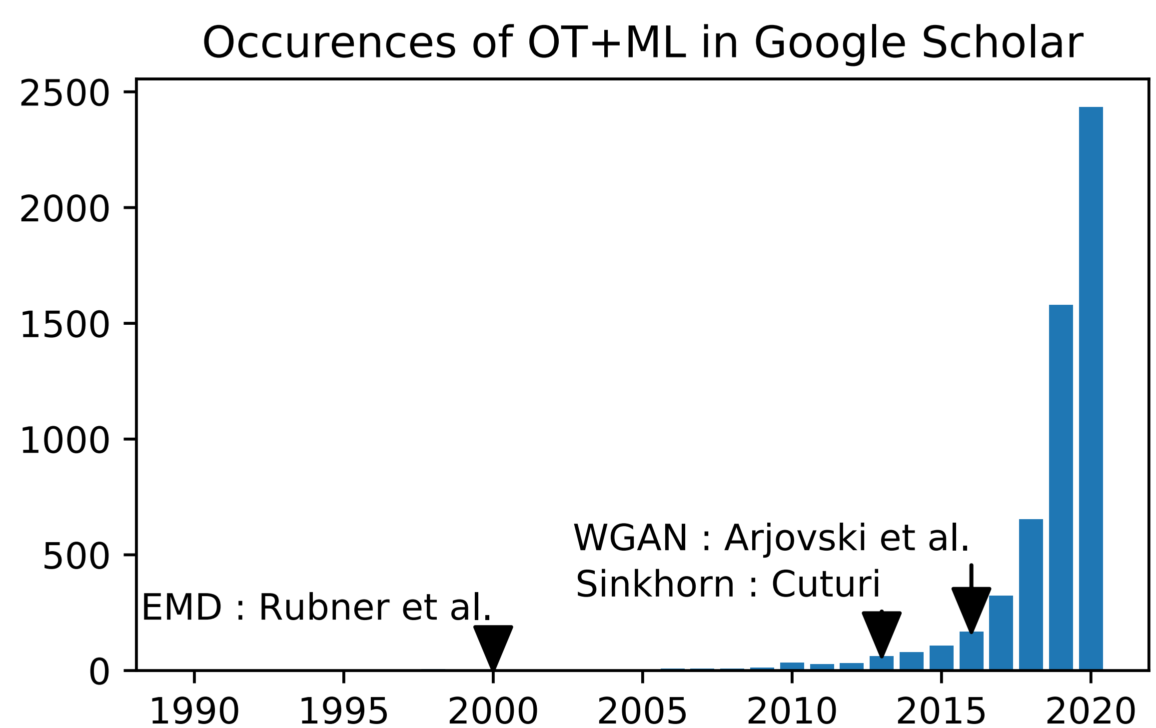
How to operate on them?

Tools which build upon this representation

Optimal Transport theory







In short:

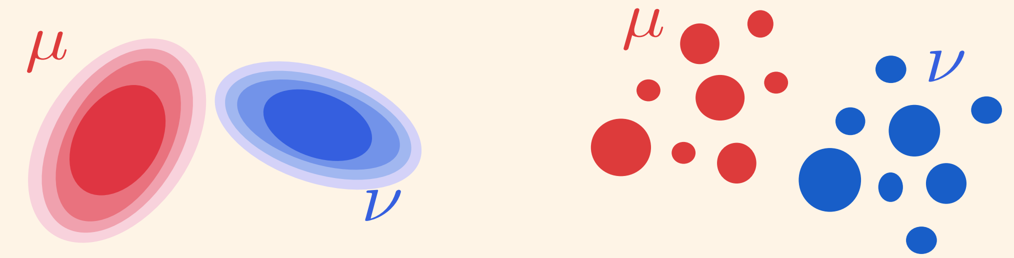
Machine Learning: Learn to make decision from **data**

How to represent data?

How to operate on them?

Mathematical representation

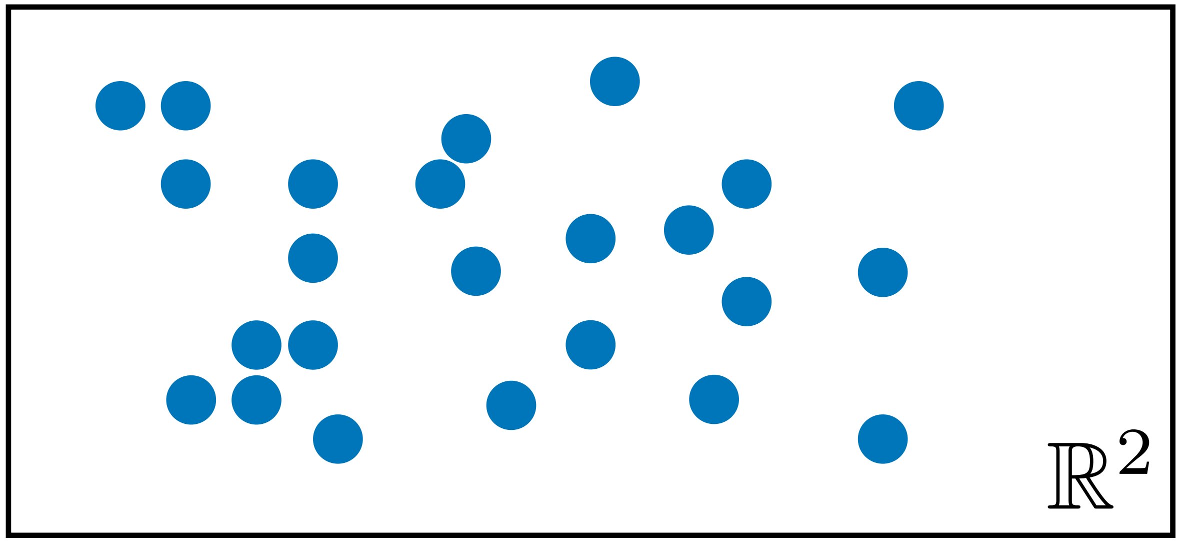
As probability distributions



Tools which build upon this representation

Optimal Transport theory

Particularly challenging: highly structured data, heterogeneous spaces



In short:

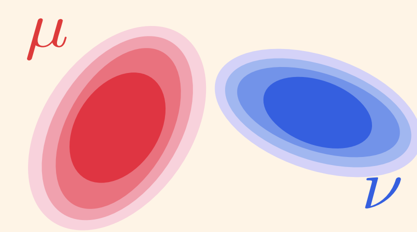
Machine Learning: Learn to make decision from **data**

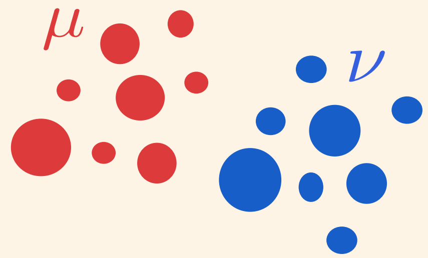
How to represent data?

How to operate on them?

Mathematical representation

As probability distributions

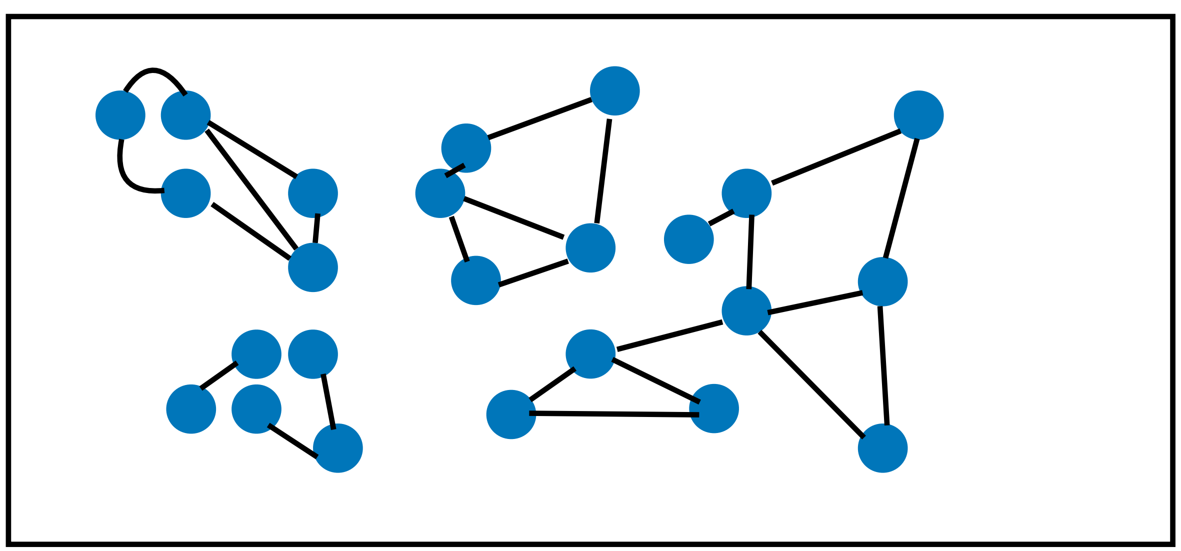


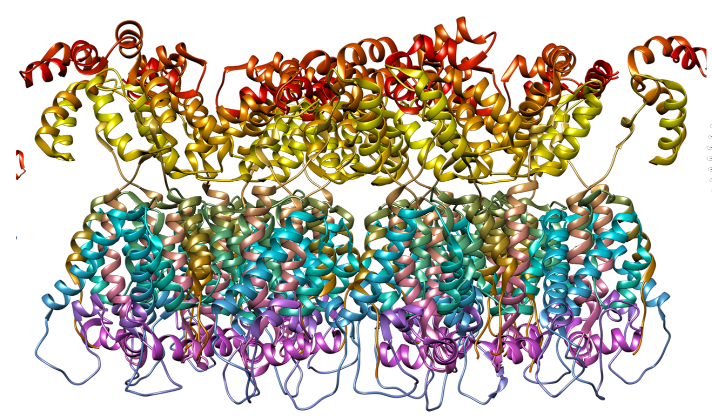


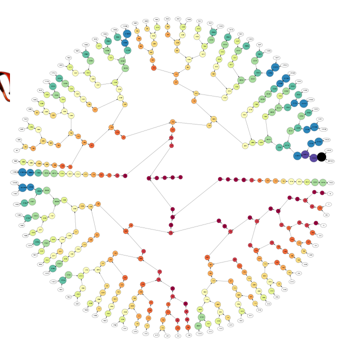
Tools which build upon this representation

Optimal Transport theory

Particularly challenging: **highly structured data**, heterogeneous spaces







graphs

molecules, sequences..

In short:

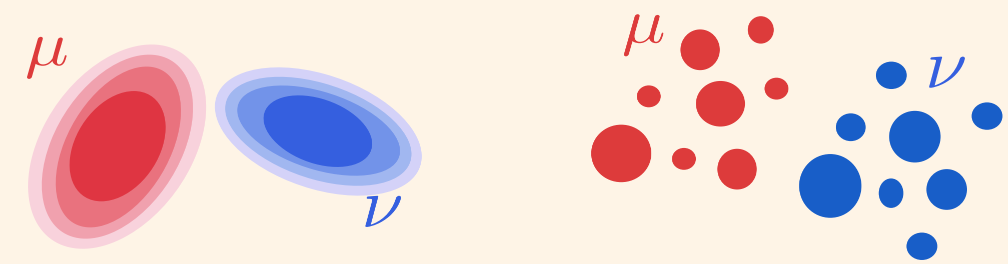
Machine Learning: Learn to make decision from **data**

How to represent data?

How to operate on them?

Mathematical representation

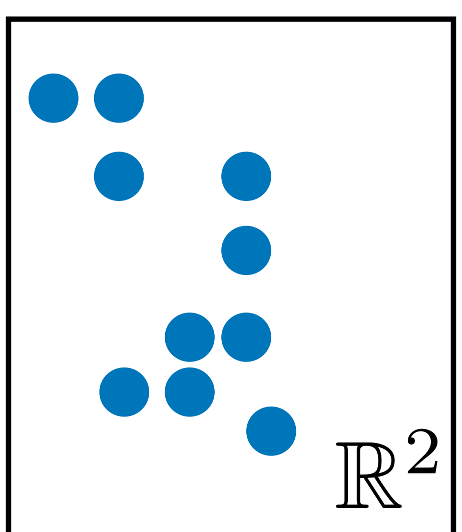
As probability distributions

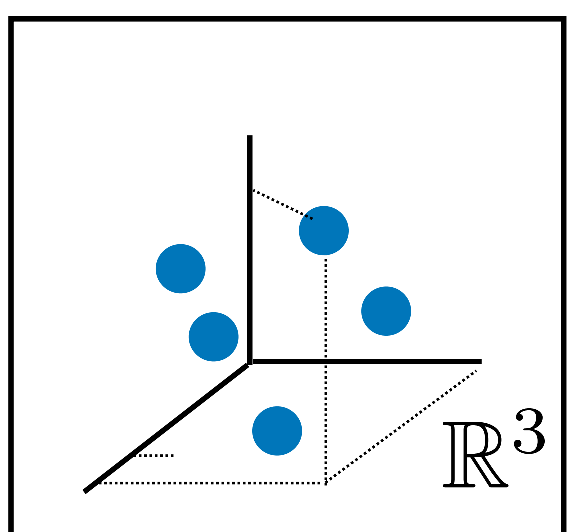


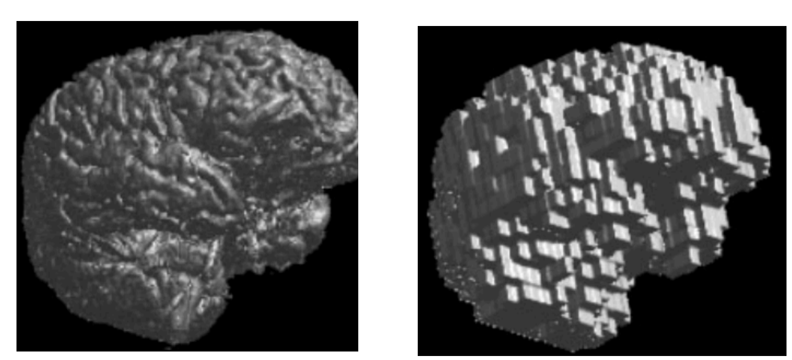
Tools which build upon this representation

Optimal Transport theory

Particularly challenging: highly structured data, **heterogeneous spaces**







high & low resolution images

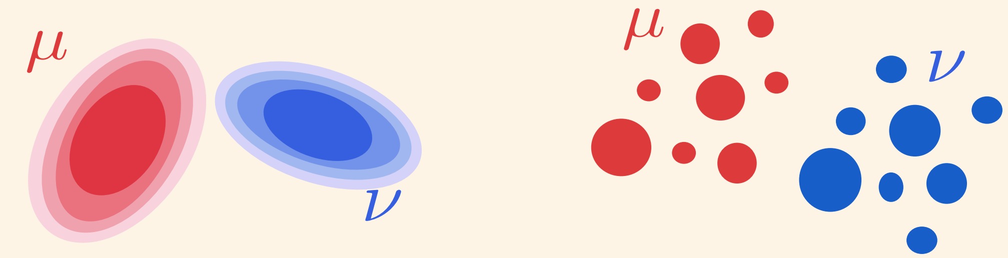
In short:

Machine Learning: Learn to make decision from **data**

- **How to represent data?**
- **How to operate on them?**

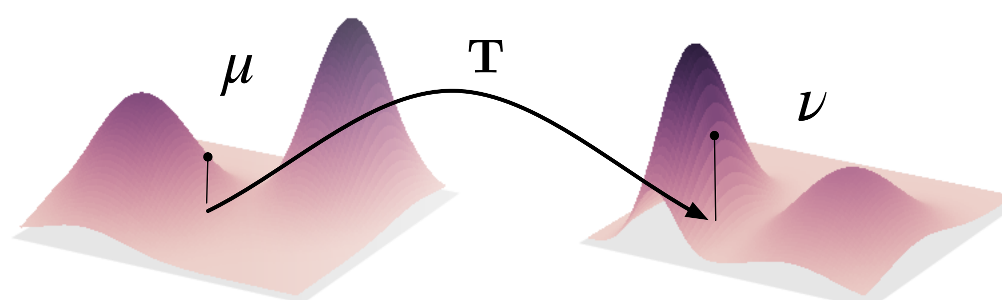
Mathematical representation

As probability distributions

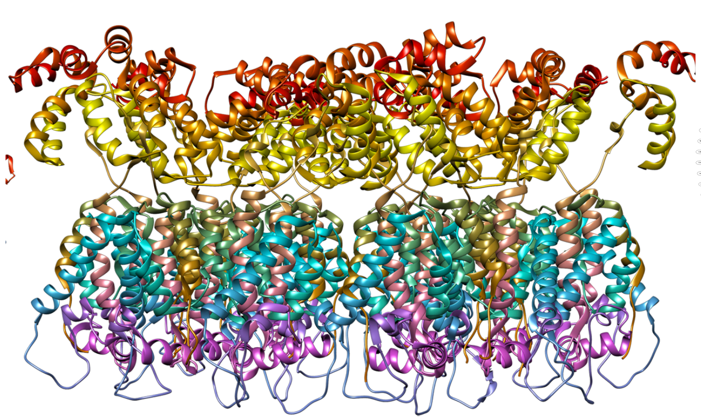


Tools which build upon this representation

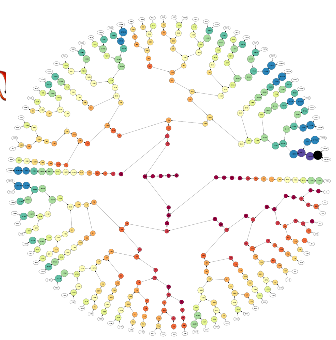
Optimal Transport theory



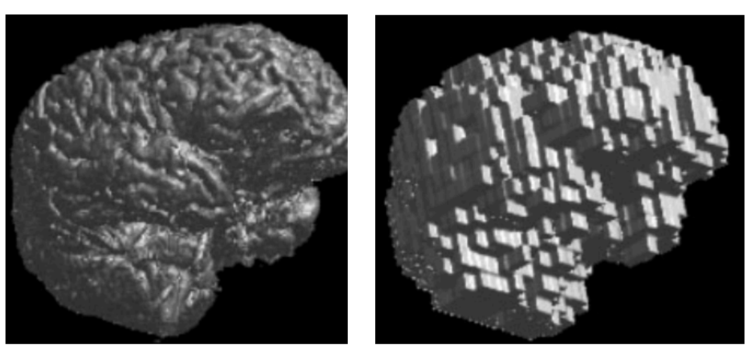
Particularly challenging: highly structured data, heterogeneous spaces



molecules, sequences..



graphs

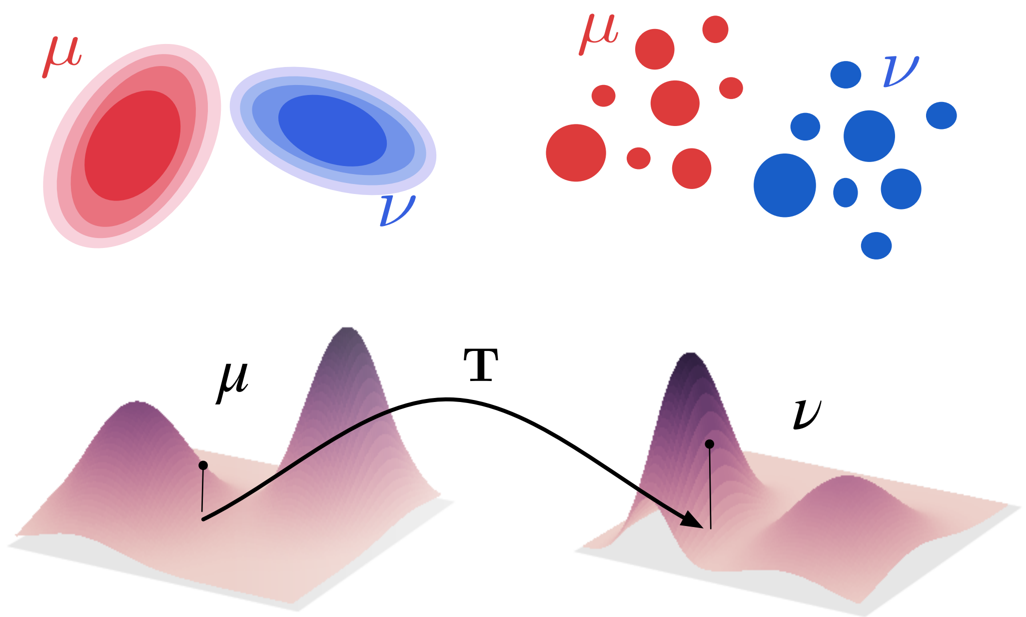


high & low resolution images

Use + Develop the Optimal transport theory in this challenging scenario

- **Applicability**
- **Mathematical foundations**

From linear Optimal Transport to Gromov-Wasserstein



From linear Optimal Transport…

What is it?

Input:

$$\mu \in \mathcal{P}(\mathcal{X}), \nu \in \mathcal{P}(\mathcal{Y})$$

Two probability distributions

Output:

Geometric notion of distance between these distributions

Find correspondences/relations between the samples

From linear Optimal Transport…

Why do we care about probability distributions?

Measure and probability distributions are at the core of Machine learning

From linear Optimal Transport…

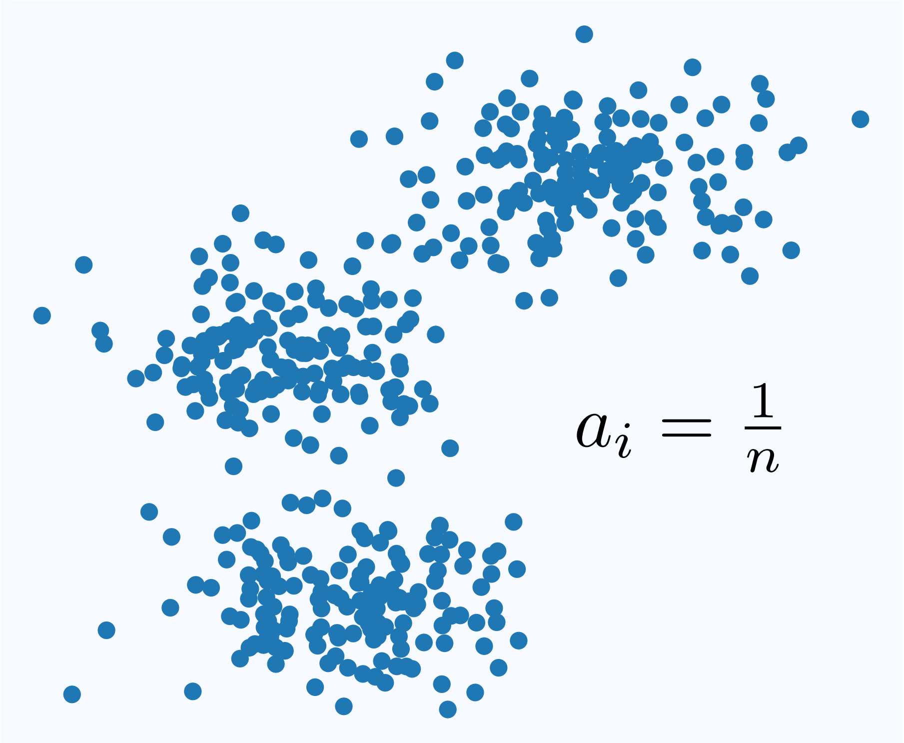
Why do we care about probability distributions?

Measure and probability distributions are at the core of Machine learning

A point of view on the data

Data: $(\mathbf{x}_i)_{i\in[\![n]\!]}$; $\mathbf{x}_i \in \mathbb{R}^d$ ⟶ A probability distribution describing the data

Lagrangian: $\sum_{i=1}^n a_i \delta_{x_i}$



$a_i = \frac{1}{n}$

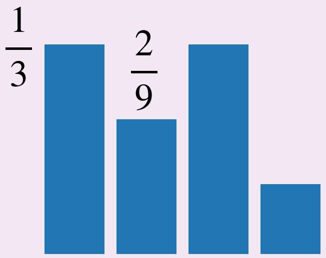
(point clouds)

$\delta_{\mathbf{x}_i}(\mathbf{x}) = 1$ if $\mathbf{x} = \mathbf{x}_i$ else 0

Probability simplex

$\mathbf{a} = (a_i)_{i\in[\![n]\!]} \in \Sigma_n$

$a_i \geq 0, \sum_{i=1}^n a_i = 1$



From linear Optimal Transport…

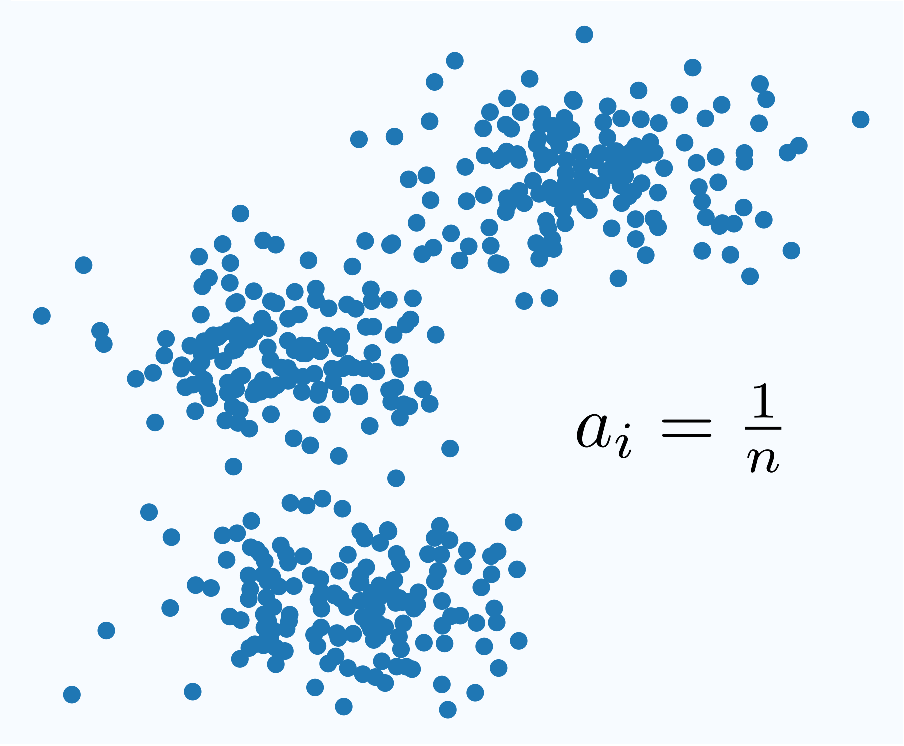
Why do we care about probability distributions?

Measure and probability distributions are at the core of Machine learning

A point of view on the data

Data: $(\mathbf{x}_i)_{i\in[\![n]\!]}$; $\mathbf{x}_i \in \mathbb{R}^d$ ⟶ A probability distribution describing the data

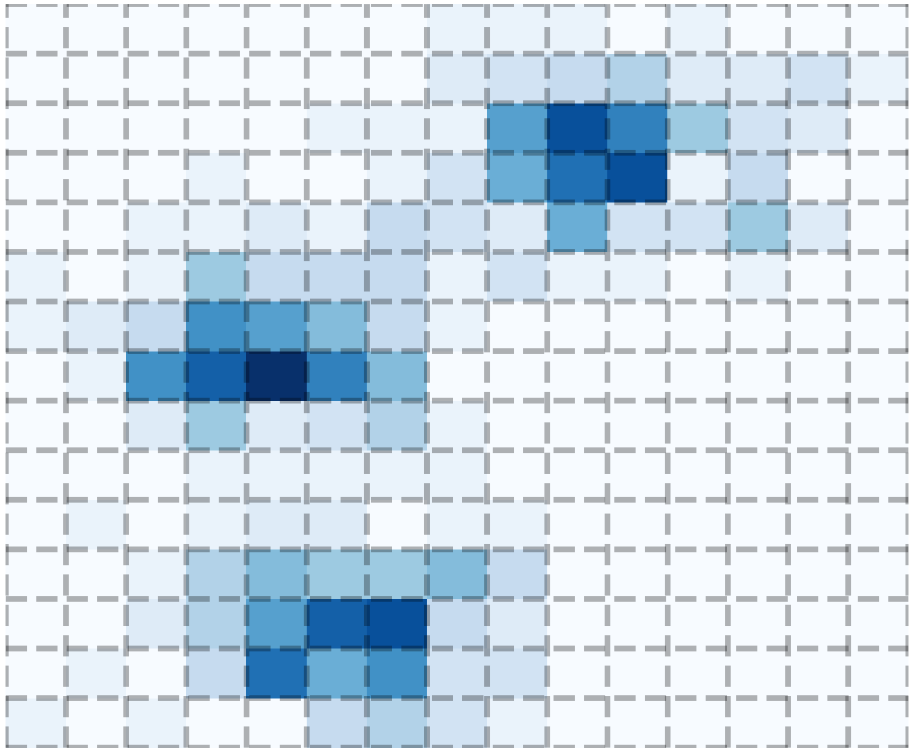
Lagrangian: $\sum_{i=1}^n a_i \delta_{x_i}$



(point clouds)

$\delta_{\mathbf{x}_i}(\mathbf{x}) = 1$ if $\mathbf{x} = \mathbf{x}_i$ else 0

Eulerian: $\sum_{i=1}^N a_i \delta_{\hat{x}_i}$



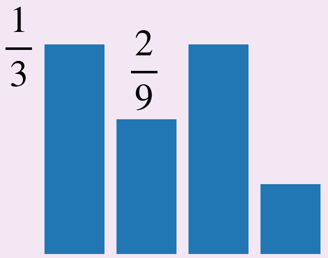
(histograms)

$\hat{x}_i$ fixed position (grid)

Probability simplex

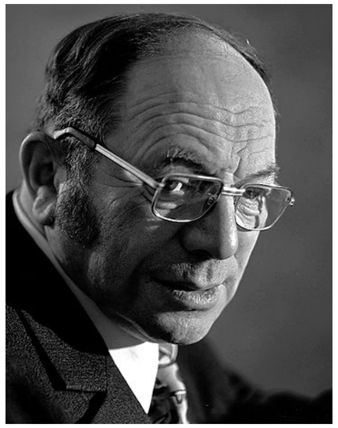
$\mathbf{a} = (a_i)_{i\in[\![n]\!]} \in \Sigma_n$

$a_i \geq 0, \sum_{i=1}^n a_i = 1$



From linear Optimal Transport…

Formulation



Two probability distributions

$\mu \in \mathcal{P}(\mathcal{X}), \nu \in \mathcal{P}(\mathcal{Y})$

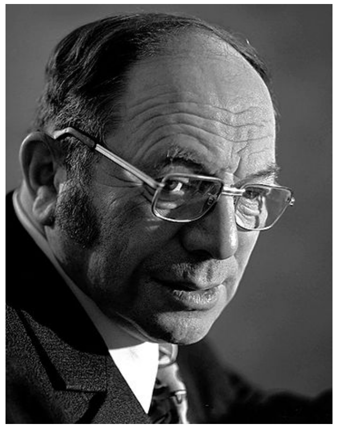
A cost function

$c(x, y) : \mathcal{X} \times \mathcal{Y} \to \mathbb{R}$

Optimal Transport

From linear Optimal Transport…

Kantorovitch Formulation



Two probability distributions

$\mu \in \mathcal{P}(\mathcal{X}), \nu \in \mathcal{P}(\mathcal{Y})$

A cost function

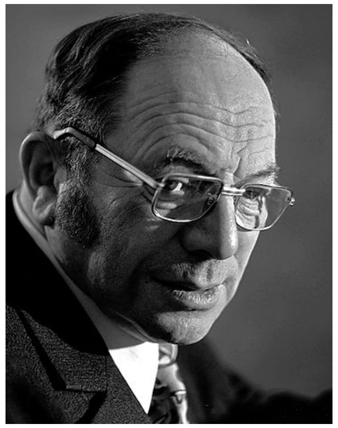
$c(x, y) : \mathcal{X} \times \mathcal{Y} \to \mathbb{R}$

Optimal Transport

All the mass of μ is **transported** to ν by a **transport plan** $\pi \in \mathcal{P}(\mathcal{X} \times \mathcal{Y})$

From linear Optimal Transport…

Kantorovitch Formulation



Two probability distributions

$\mu \in \mathcal{P}(\mathcal{X}), \nu \in \mathcal{P}(\mathcal{Y})$

A cost function

$c(x, y) : \mathcal{X} \times \mathcal{Y} \to \mathbb{R}$

Optimal Transport

All the mass of μ is **transported** to ν by a **transport plan** $\pi \in \mathcal{P}(\mathcal{X} \times \mathcal{Y})$

We want to find the plan that **minimizes the overall cost** of moving all the points

From linear Optimal Transport…

Kantorovich Formulation: an example

Two probability distributions

$$\mu = \sum_{i=1}^{n} a_i \delta_{x_i} \quad \nu = \sum_{i=j}^{m} b_j \delta_{y_j}$$

A cost function

$$c(x, y) : \mathcal{X} \times \mathcal{Y} \rightarrow \mathbb{R}$$

Bakeries = quantity of breads

loc: x_i quantity: a_i

Cafés = demand of breads

loc: y_j demand: b_j

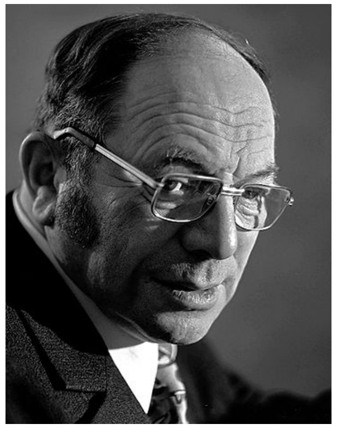
Distance between bakeries and cafés

$$c(x_i, y_j)$$

We want to route all the breads from bakeries to cafés the cheapest way

From linear Optimal Transport...

Kantorovitch Formulation: an example



Two probability distributions

$\mu = \sum_{i=1}^{n} a_i \delta_{x_i} \quad \nu = \sum_{i=j}^{m} b_j \delta_{y_j}$

A cost function

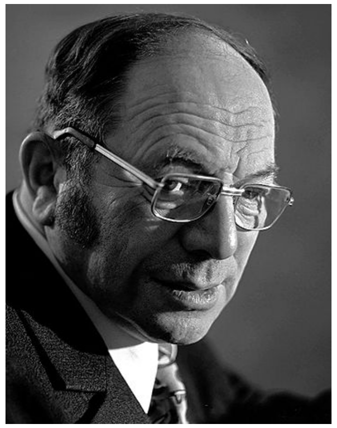
$c(x, y) : \mathcal{X} \times \mathcal{Y} \to \mathbb{R}$

Kantorovitch formulation

$$\min_{\boldsymbol{\pi} \in \Pi(\mathbf{a}, \mathbf{b})} \sum_{i,j=1}^{m,n} c(x_i, y_j) \pi_{ij}$$

From linear Optimal Transport…

Kantorovitch Formulation: an example



Two probability distributions

$\mu = \sum_{i=1}^{n} a_i \delta_{x_i} \quad \nu = \sum_{i=j}^{m} b_j \delta_{y_j}$

A cost function

$c(x, y) : \mathcal{X} \times \mathcal{Y} \to \mathbb{R}$

Kantorovitch formulation

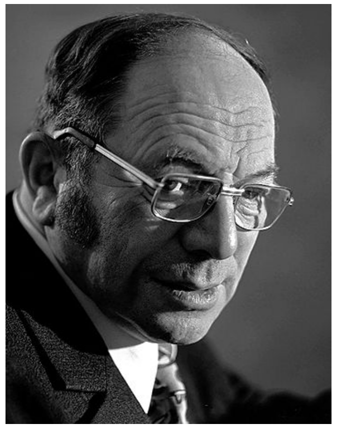
$$\min_{\boldsymbol{\pi} \in \Pi(\mathbf{a}, \mathbf{b})} \sum_{i,j=1}^{m,n} c(x_i, y_j) \pi_{ij}$$

Set of couplings/ transport plans

$\Pi(\mathbf{a}, \mathbf{b})$

From linear Optimal Transport…

Kantorovitch Formulation: an example



Two probability distributions

$\mu = \sum_{i=1}^{n} a_i \delta_{x_i} \quad \nu = \sum_{i=j}^{m} b_j \delta_{y_j}$

A cost function

$c(x, y) : \mathcal{X} \times \mathcal{Y} \to \mathbb{R}$

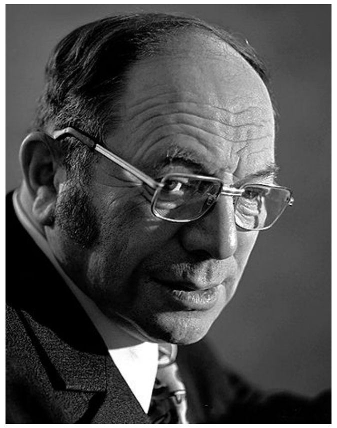
Kantorovitch formulation

$$\min_{\boldsymbol{\pi} \in \Pi(\mathbf{a}, \mathbf{b})} \sum_{i,j=1}^{m,n} c(x_i, y_j) \pi_{ij}$$

How much is shifted from x_i to y_j

From linear Optimal Transport…

Kantorovitch Formulation: an example



Two probability distributions

$\mu = \sum_{i=1}^{n} a_i \delta_{x_i} \quad \nu = \sum_{i=j}^{m} b_j \delta_{y_j}$

A cost function

$c(x,y) : \mathcal{X} \times \mathcal{Y} \to \mathbb{R}$

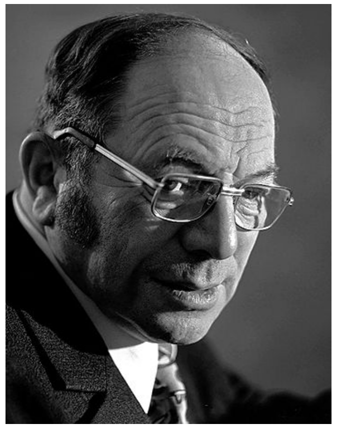
Kantorovitch formulation

$$\min_{\boldsymbol{\pi} \in \Pi(\mathbf{a},\mathbf{b})} \sum_{i,j=1}^{m,n} c(x_i, y_j)\pi_{ij}$$

Cost of moving masses from x_i to y_j

From linear Optimal Transport...

Kantorovich Formulation: an example



Two probability distributions

$$\mu = \sum_{i=1}^{n} a_i \delta_{x_i} \quad \nu = \sum_{i=j}^{m} b_j \delta_{y_j}$$

A cost function

$$c(x, y) : \mathcal{X} \times \mathcal{Y} \to \mathbb{R}$$

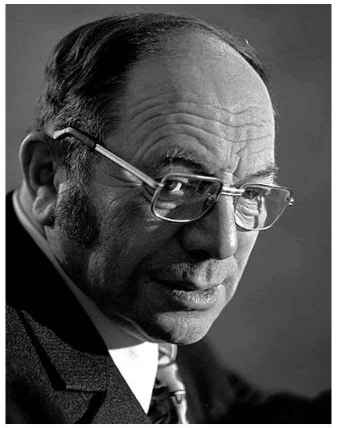
Kantorovich formulation

$$\min_{\boldsymbol{\pi} \in \Pi(\mathbf{a}, \mathbf{b})} \sum_{i,j=1}^{m,n} c(x_i, y_j) \pi_{ij}$$

Total cost

From linear Optimal Transport…

Kantorovitch Formulation: an example



Two probability distributions

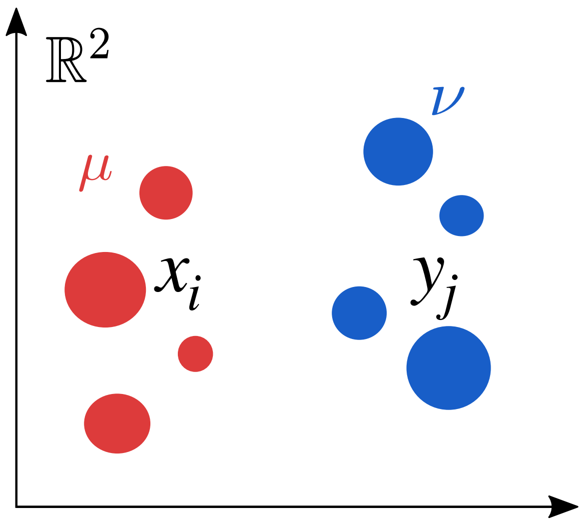
$$\mu = \sum_{i=1}^{n} a_i \delta_{x_i} \quad \nu = \sum_{i=j}^{m} b_j \delta_{y_j}$$

A cost function

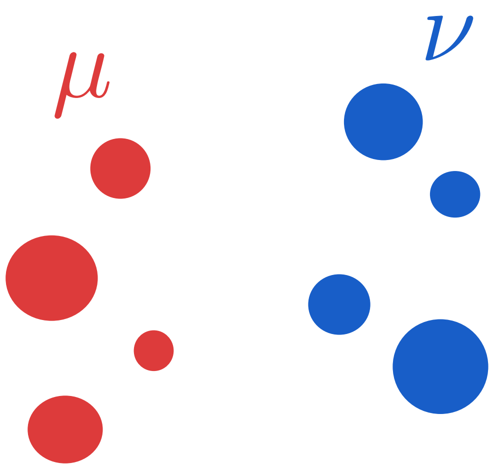
$$c(x, y) : \mathcal{X} \times \mathcal{Y} \to \mathbb{R}$$

Kantorovitch formulation

$$\min_{\boldsymbol{\pi} \in \Pi(\mathbf{a}, \mathbf{b})} \sum_{i,j=1}^{m,n} c(x_i, y_j) \pi_{ij}$$

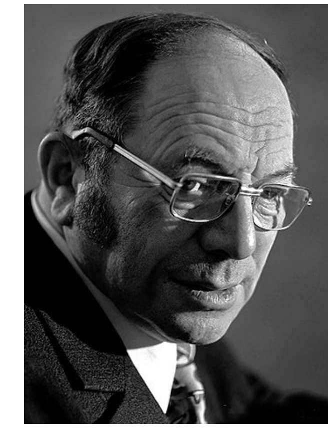


$$\Pi(\mathbf{a}, \mathbf{b}) = \{\boldsymbol{\pi} \in \mathbb{R}_+^{n \times m} \mid \forall (i,j), \sum_{j=1}^{m} \pi_{ij} = a_i, \ \sum_{i=1}^{n} \pi_{ij} = b_j\}$$



From linear Optimal Transport…

Kantorovitch Formulation: an example



Two probability distributions

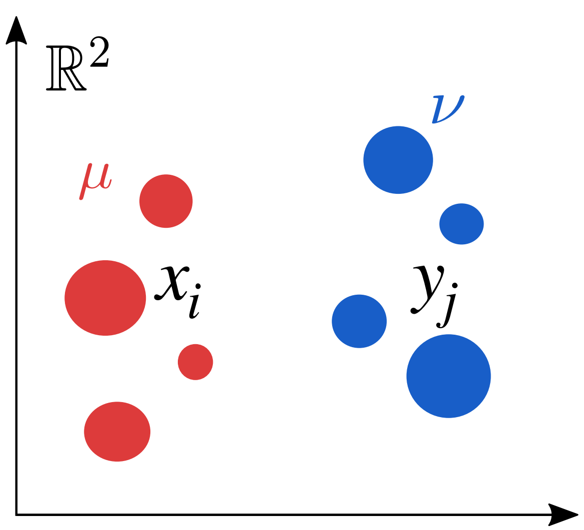
$\mu = \sum_{i=1}^{n} a_i \delta_{x_i} \quad \nu = \sum_{i=j}^{m} b_j \delta_{y_j}$

A cost function

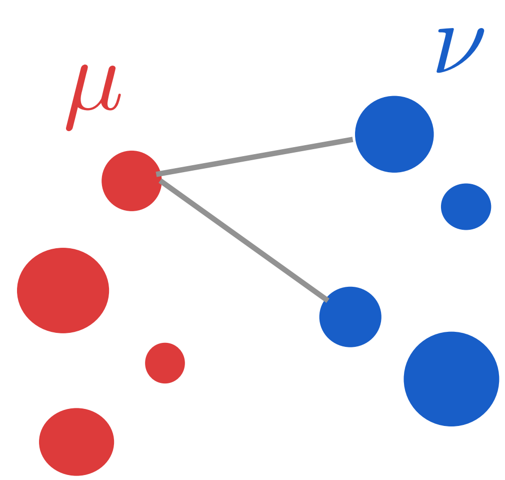
$c(x, y) : \mathcal{X} \times \mathcal{Y} \to \mathbb{R}$

Kantorovitch formulation

$$\min_{\boldsymbol{\pi} \in \Pi(\mathbf{a}, \mathbf{b})} \sum_{i,j=1}^{m,n} c(x_i, y_j) \pi_{ij}$$



$\Pi(\mathbf{a}, \mathbf{b}) = \{\boldsymbol{\pi} \in \mathbb{R}_+^{n \times m} \mid \forall (i, j), \sum_{j=1}^{m} \pi_{ij} = a_i, \ \sum_{i=1}^{n} \pi_{ij} = b_j\}$



From linear Optimal Transport…

Kantorovitch Formulation: an example

Two probability distributions

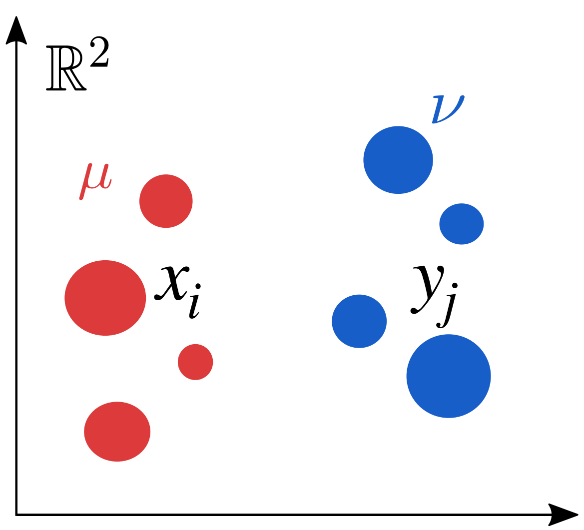
$$\mu = \sum_{i=1}^{n} a_i \delta_{x_i} \quad \nu = \sum_{i=j}^{m} b_j \delta_{y_j}$$

A cost function

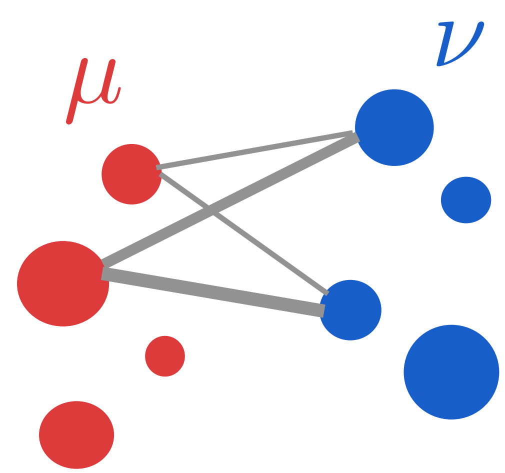
$$c(x, y) : \mathcal{X} \times \mathcal{Y} \to \mathbb{R}$$

Kantorovitch formulation

$$\min_{\boldsymbol{\pi} \in \Pi(\mathbf{a}, \mathbf{b})} \sum_{i,j=1}^{m,n} c(x_i, y_j) \pi_{ij}$$



$$\Pi(\mathbf{a}, \mathbf{b}) = \{\boldsymbol{\pi} \in \mathbb{R}_+^{n \times m} \mid \forall (i,j), \sum_{j=1}^{m} \pi_{ij} = a_i, \ \sum_{i=1}^{n} \pi_{ij} = b_j\}$$



From linear Optimal Transport...

Kantorovitch Formulation: an example

Two probability distributions

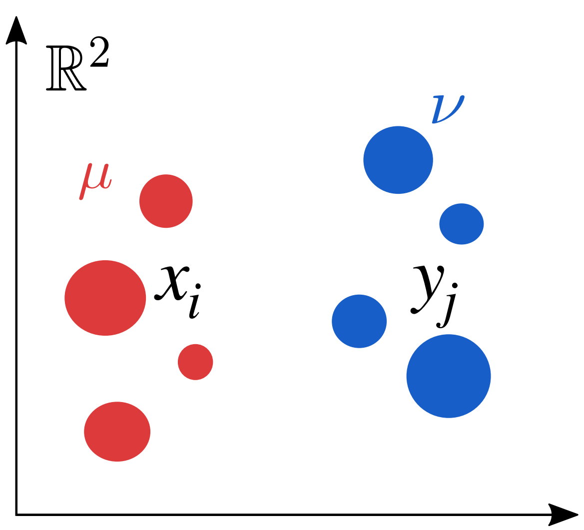
$$\mu = \sum_{i=1}^{n} a_i \delta_{x_i} \quad \nu = \sum_{i=j}^{m} b_j \delta_{y_j}$$

A cost function

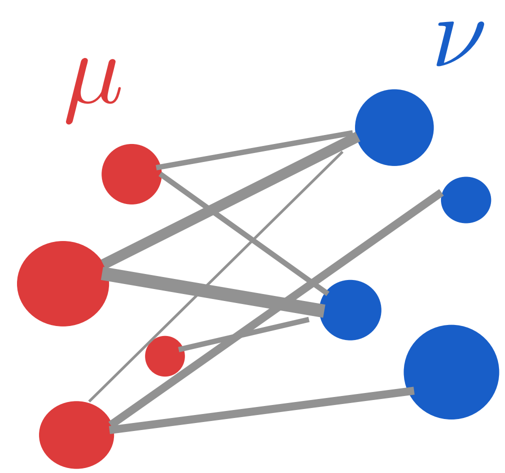
$$c(x,y) : \mathcal{X} \times \mathcal{Y} \to \mathbb{R}$$

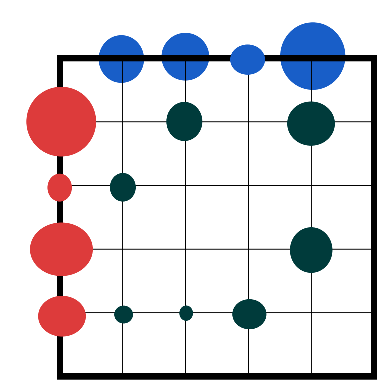
Kantorovitch formulation

$$\min_{\boldsymbol{\pi} \in \Pi(\mathbf{a},\mathbf{b})} \sum_{i,j=1}^{m,n} c(x_i, y_j)\pi_{ij}$$



$$\Pi(\mathbf{a},\mathbf{b}) = \{\boldsymbol{\pi} \in \mathbb{R}_+^{n\times m} \mid \forall (i,j), \sum_{j=1}^{m} \pi_{ij} = a_i, \ \sum_{i=1}^{n} \pi_{ij} = b_j\}$$

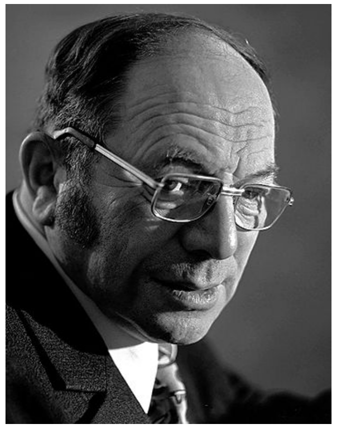




$$\pi \in \mathbb{R}_+^{n\times m}$$

From linear Optimal Transport…

Kantorovitch Formulation: general case



Two probability distributions

$\mu \in \mathcal{P}(\mathcal{X}), \nu \in \mathcal{P}(\mathcal{Y})$

A cost function

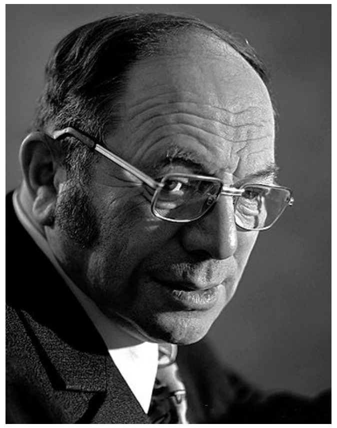
$c(x,y) : \mathcal{X} \times \mathcal{Y} \rightarrow \mathbb{R}$

Kantorovitch formulation

$$\mathcal{T}_c(\mu, \nu) = \inf_{\pi \in \Pi(\mu,\nu)} \int_{\mathcal{X}\times\mathcal{Y}} c(x,y) \mathrm{d}\pi(x,y)$$

From linear Optimal Transport...

Wasserstein distance



Two probability distributions

$\mu \in \mathcal{P}(\Omega), \nu \in \mathcal{P}(\Omega)$

A distance

$d : \Omega \times \Omega \to \mathbb{R}_+$

Example: $\Omega = \mathbb{R}^d$

Wasserstein distance

$$W_p^p(\mu, \nu) = \inf_{\pi \in \Pi(\mu, \nu)} \int_{\Omega \times \Omega} d^p(x, y) \mathrm{d}\pi(x, y)$$

Result:

$\mathcal{P}(\Omega)$ is a metric space

$W_p(\mu, \nu) = 0 \iff \mu = \nu$

…to Gromov-Wasserstein

What if ?

Data are in Incomparable spaces

Two probability distributions

$$\mu \in \mathcal{P}(\mathcal{X}), \nu \in \mathcal{P}(\mathcal{Y}) \text{ with } \mathcal{X}, \mathcal{Y} \not\subseteq \Omega$$

A cost function ?????

$$c(x, y) : \mathcal{X} \times \mathcal{Y} \rightarrow \mathbb{R}$$

⇨ Not straightforward to find a suitable cost (e.g. no distance available)

...to Gromov-Wasserstein

What if ?

Data are in Incomparable spaces

Two probability distributions

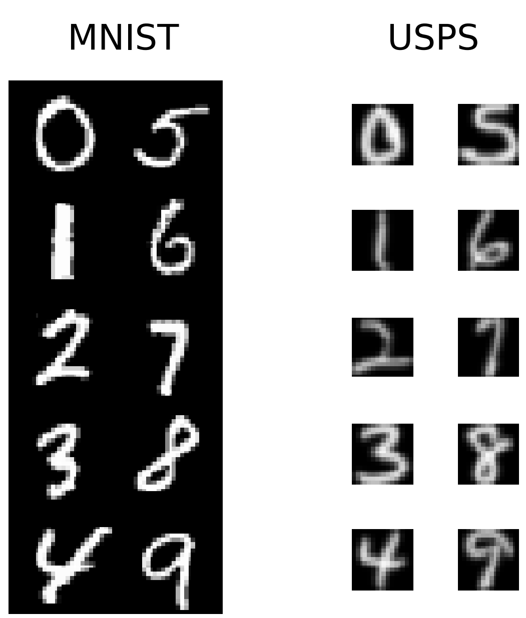
$$\mu \in \mathcal{P}(\mathcal{X}), \nu \in \mathcal{P}(\mathcal{Y}) \text{ with } \mathcal{X}, \mathcal{Y} \not\subseteq \Omega$$

A cost function ?????

$$c(x, y) : \mathcal{X} \times \mathcal{Y} \to \mathbb{R}$$

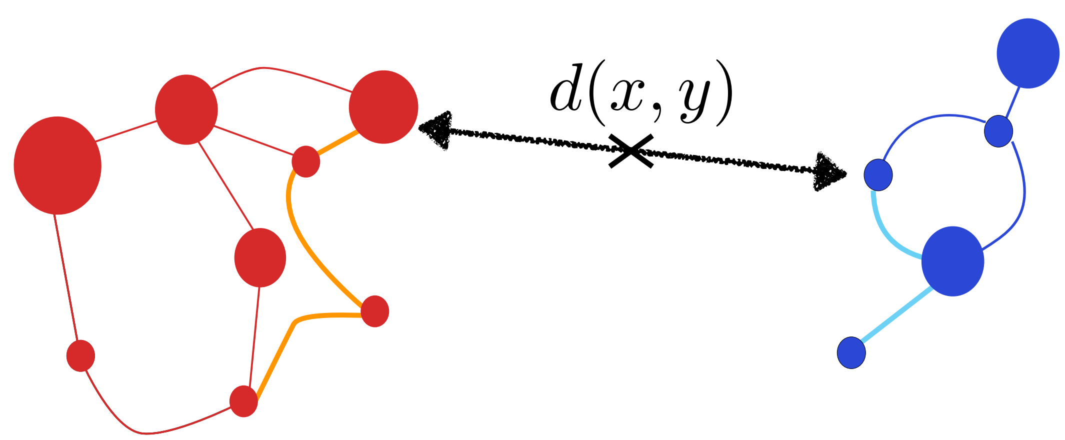
⇨ Not straightforward to find a suitable cost (e.g. no distance available)

Different Euclidean spaces



Example: $\mathcal{X} = \mathbb{R}^{28*28}, \mathcal{Y} = \mathbb{R}^{16*16}$

Samples = nodes of different graphs



Example: $\mathcal{X} = \text{Graph 1}, \mathcal{Y} = \text{Graph 2}$

…to Gromov-Wasserstein

Gromov-Wasserstein distance


Two probability distributions

$$\mu \in \mathcal{P}(\mathcal{X}), \nu \in \mathcal{P}(\mathcal{Y})$$

Two « intra-domain » costs

$$c_{\mathcal{X}} : \mathcal{X} \times \mathcal{X} \to \mathbb{R}$$
$$c_{\mathcal{Y}} : \mathcal{Y} \times \mathcal{Y} \to \mathbb{R}$$

Gromov-Wasserstein distance

$$GW_p^p(c_{\mathcal{X}}, c_{\mathcal{Y}}, \mu, \nu) = \inf_{\pi \in \Pi(\mu,\nu)} \int_{\mathcal{X}\times\mathcal{Y}} \int_{\mathcal{X}\times\mathcal{Y}} \left| c_{\mathcal{X}}(x,x') - c_{\mathcal{Y}}(y,y') \right|^p \mathrm{d}\pi(x,y)\mathrm{d}\pi(x',y')$$

…to Gromov-Wasserstein

Gromov-Wasserstein distance

Two probability distributions

$$\mu \in \mathcal{P}(\mathcal{X}), \nu \in \mathcal{P}(\mathcal{Y})$$

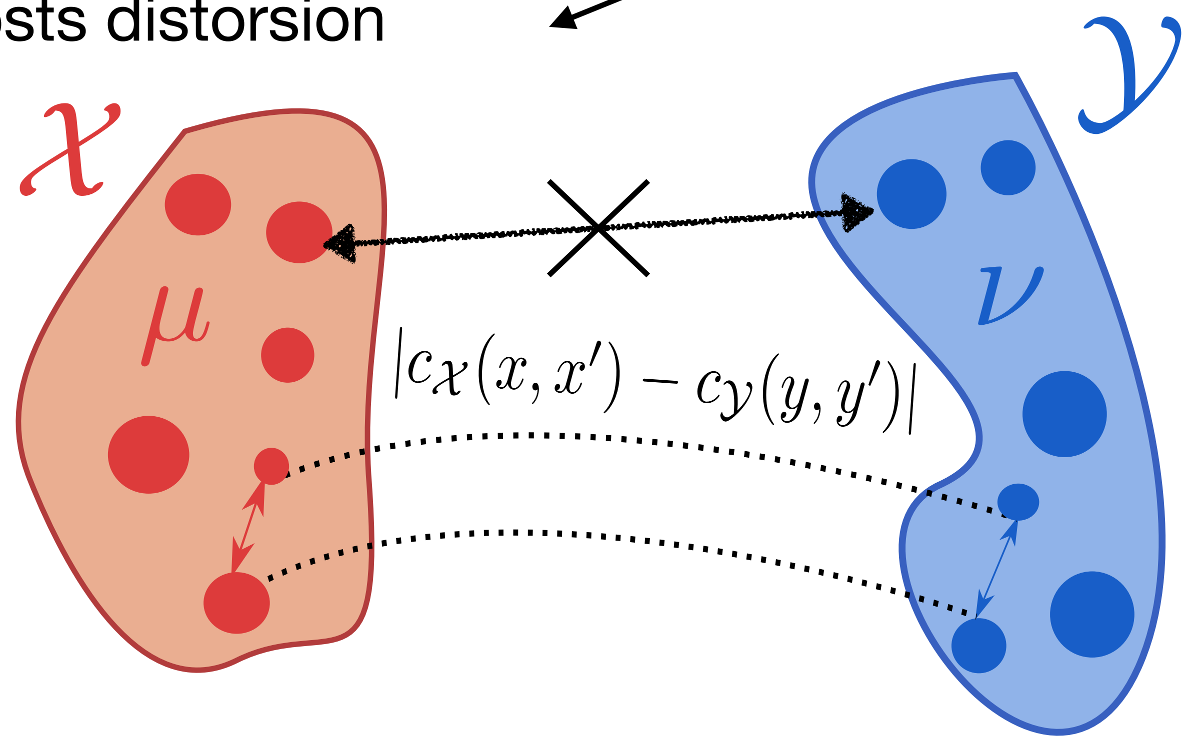
Two « intra-domain » costs

$$c_{\mathcal{X}} : \mathcal{X} \times \mathcal{X} \rightarrow \mathbb{R}$$
$$c_{\mathcal{Y}} : \mathcal{Y} \times \mathcal{Y} \rightarrow \mathbb{R}$$

Gromov-Wasserstein distance

$$GW_p^p(c_{\mathcal{X}}, c_{\mathcal{Y}}, \mu, \nu) = \inf_{\pi \in \Pi(\mu,\nu)} \int_{\mathcal{X}\times\mathcal{Y}} \int_{\mathcal{X}\times\mathcal{Y}} \left| c_{\mathcal{X}}(x,x') - c_{\mathcal{Y}}(y,y') \right|^p \mathrm{d}\pi(x,y)\mathrm{d}\pi(x',y'))$$

Measure the costs distortion



...to Gromov-Wasserstein

Gromov-Wasserstein distance

Two probability distributions

$$\mu \in \mathcal{P}(\mathcal{X}), \nu \in \mathcal{P}(\mathcal{Y})$$

Two « intra-domain » costs

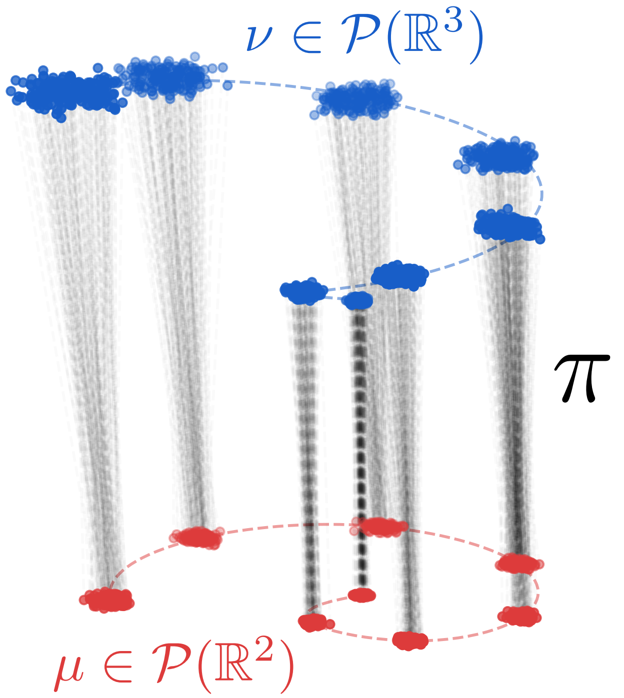
$$c_{\mathcal{X}} : \mathcal{X} \times \mathcal{X} \to \mathbb{R}$$
$$c_{\mathcal{Y}} : \mathcal{Y} \times \mathcal{Y} \to \mathbb{R}$$

Gromov-Wasserstein distance

$$GW_p^p(c_{\mathcal{X}}, c_{\mathcal{Y}}, \mu, \nu) = \inf_{\pi \in \Pi(\mu,\nu)} \int_{\mathcal{X}\times\mathcal{Y}} \int_{\mathcal{X}\times\mathcal{Y}} \left| c_{\mathcal{X}}(x,x') - c_{\mathcal{Y}}(y,y') \right|^p \mathrm{d}\pi(x,y)\mathrm{d}\pi(x',y'))$$

The transportation problem is not linear anymore but **quadratic**

Associate pair of points with similar costs in each space



...to Gromov-Wasserstein

Gromov-Wasserstein distance

Gromov-Wasserstein distance

$$GW_p^p(c_{\mathcal{X}}, c_{\mathcal{Y}}, \mu, \nu) = \inf_{\pi \in \Pi(\mu, \nu)} \int_{\mathcal{X}\times\mathcal{Y}} \int_{\mathcal{X}\times\mathcal{Y}} \left|c_{\mathcal{X}}(x, x') - c_{\mathcal{Y}}(y, y')\right|^p \mathrm{d}\pi(x, y)\mathrm{d}\pi(x', y')$$

A distance w.r.t isomorphism

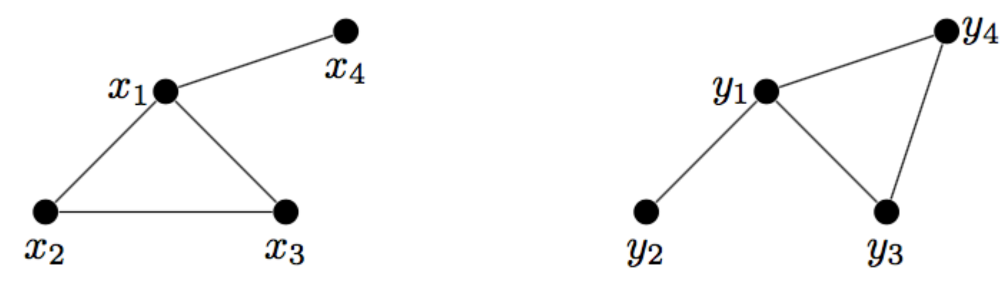
GW is a distance on the "space of all spaces":

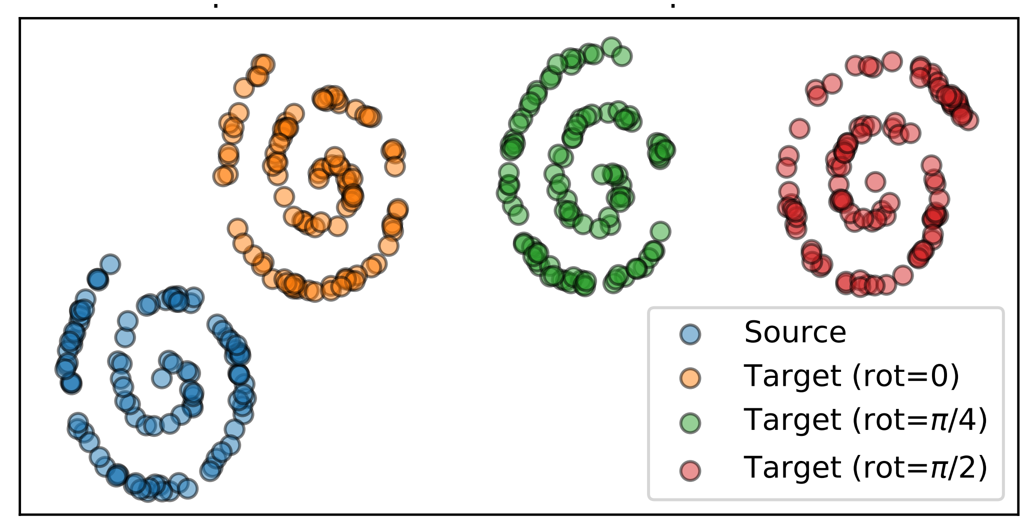
$$\mathbb{X} = \{(\mathcal{X}, d_{\mathcal{X}}, \mu \in \mathcal{P}(\mathcal{X})); d_{\mathcal{X}} \text{ metric } \} \text{ (mm-spaces)}$$

- $GW_p(d_{\mathcal{X}}, d_{\mathcal{Y}}, \mu, \nu) = 0$ iff $\exists \phi : \mathcal{X} \to \mathcal{Y}$

  ϕ is a isometry $d_{\mathcal{X}}(x, x') = d_{\mathcal{Y}}(\phi(x), \phi(x'))$

Isometry: permutations, rotations, translations,…





...to Gromov-Wasserstein

Gromov-Wasserstein distance

Gromov-Wasserstein distance

$$GW_p^p(c_{\mathcal{X}}, c_{\mathcal{Y}}, \mu, \nu) = \inf_{\pi \in \Pi(\mu,\nu)} \int_{\mathcal{X}\times\mathcal{Y}} \int_{\mathcal{X}\times\mathcal{Y}} \left| c_{\mathcal{X}}(x,x') - c_{\mathcal{Y}}(y,y') \right|^p \mathrm{d}\pi(x,y)\mathrm{d}\pi(x',y')$$

A distance w.r.t isomorphism

GW is a distance on the "space of all spaces":

$$\mathbb{X} = \{(\mathcal{X}, d_{\mathcal{X}}, \mu \in \mathcal{P}(\mathcal{X})); d_{\mathcal{X}} \text{ metric } \} \text{ (mm-spaces)}$$

- $GW_p(d_{\mathcal{X}}, d_{\mathcal{Y}}, \mu, \nu) = 0$ iff $\exists \phi : \mathcal{X} \rightarrow \mathcal{Y}$

  ϕ is a isometry $d_{\mathcal{X}}(x,x') = d_{\mathcal{Y}}(\phi(x), \phi(x'))$

  ϕ is measure-preserving: $\phi\#\mu = \nu$

Push-forward $\phi\#\mu$

$$\mu = \sum_{i=1}^n a_i \delta_{x_i} \underset{\phi\#\mu}{\rightarrow} \sum_{i=1}^n a_i \delta_{\phi(x_i)}$$

...to Gromov-Wasserstein

Gromov-Wasserstein distance

Gromov-Wasserstein distance

$$GW_p^p(c_{\mathcal{X}}, c_{\mathcal{Y}}, \mu, \nu) = \inf_{\pi \in \Pi(\mu,\nu)} \int_{\mathcal{X}\times\mathcal{Y}} \int_{\mathcal{X}\times\mathcal{Y}} \left|c_{\mathcal{X}}(x,x') - c_{\mathcal{Y}}(y,y')\right|^p \mathrm{d}\pi(x,y)\mathrm{d}\pi(x',y')$$

A distance w.r.t isomorphism

GW is a distance on the "space of all spaces":

$$\mathbb{X} = \{(\mathcal{X}, d_{\mathcal{X}}, \mu \in \mathcal{P}(\mathcal{X})); d_{\mathcal{X}} \text{ metric } \} \text{ (mm-spaces)}$$

- $GW_p(d_{\mathcal{X}}, d_{\mathcal{Y}}, \mu, \nu) = 0$ iff $\exists \phi : \mathcal{X} \to \mathcal{Y}$

  ϕ is a isometry $d_{\mathcal{X}}(x,x') = d_{\mathcal{Y}}(\phi(x), \phi(x'))$

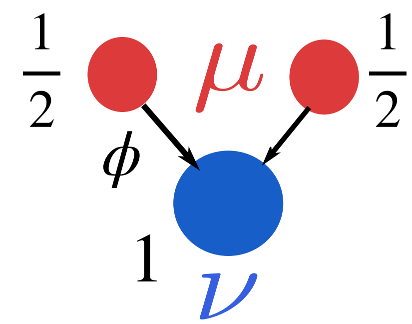
  ϕ is measure-preserving: $\phi\#\mu = \nu$

(Weights are compatible)

Push-forward $\phi\#\mu$

$$\mu = \sum_{i=1}^n a_i \delta_{x_i} \underset{\phi\#\mu}{\rightarrow} \sum_{i=1}^n a_i \delta_{\phi(x_i)}$$

Compatible



$$\frac{1}{2} + \frac{1}{2} \to 1$$

…to Gromov-Wasserstein

Gromov-Wasserstein distance

Gromov-Wasserstein distance

$$GW_p^p(c_{\mathcal{X}}, c_{\mathcal{Y}}, {\color{red}\mu}, {\color{blue}\nu}) = \inf_{\pi \in \Pi({\color{red}\mu}, {\color{blue}\nu})} \int_{\mathcal{X}\times\mathcal{Y}} \int_{\mathcal{X}\times\mathcal{Y}} \left| c_{\mathcal{X}}(x, x') - c_{\mathcal{Y}}(y, y') \right|^p \mathrm{d}\pi(x, y)\mathrm{d}\pi(x', y'))$$

A distance w.r.t isomorphism

GW is a distance on the "space of all spaces":

$$\mathbb{X} = \{(\mathcal{X}, d_{\mathcal{X}}, \mu \in \mathcal{P}(\mathcal{X})); d_{\mathcal{X}} \text{ metric } \} \text{ (mm-spaces)}$$

- $GW_p(d_{\mathcal{X}}, d_{\mathcal{Y}}, {\color{red}\mu}, {\color{blue}\nu}) = 0$ iff $\exists \phi : \mathcal{X} \to \mathcal{Y}$

  ϕ is a isometry $d_{\mathcal{X}}(x, x') = d_{\mathcal{Y}}(\phi(x), \phi(x'))$

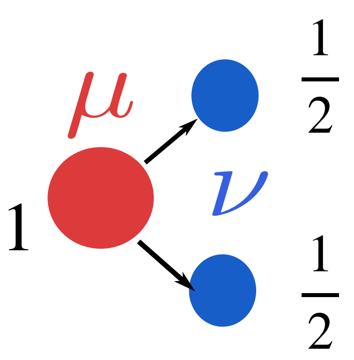
  ϕ is measure-preserving: $\phi \# {\color{red}\mu} = {\color{blue}\nu}$

(Weights are compatible)

Push-forward $\phi \# {\color{red}\mu}$

$${\color{red}\mu} = \sum_{i=1}^n a_i \delta_{x_i} \underset{\phi \# {\color{red}\mu}}{\rightarrow} \sum_{i=1}^n a_i \delta_{\phi(x_i)}$$

Not compatible



$$1 \nrightarrow (\frac{1}{2}, \frac{1}{2})$$

...to Gromov-Wasserstein

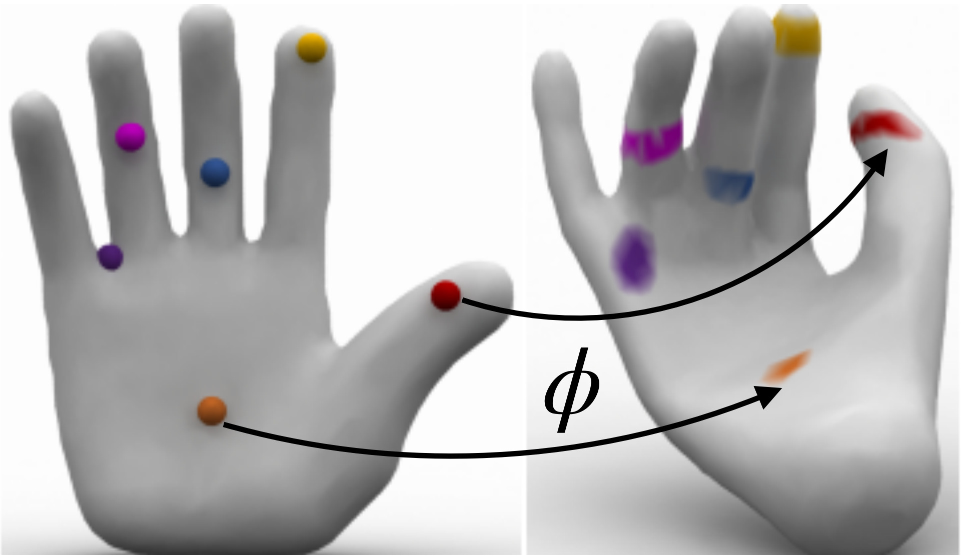
Gromov-Wasserstein distance

Gromov-Wasserstein = a bending invariant distance

- $GW_p(d_\mathcal{X}, d_\mathcal{Y}, \mu, \nu) = 0$ iff $\exists \phi : \mathcal{X} \to \mathcal{Y}$

  ϕ is a isometry $d_\mathcal{X}(x, x') = d_\mathcal{Y}(\phi(x), \phi(x'))$

  ϕ is measure-preserving $\phi \# \mu = \nu$



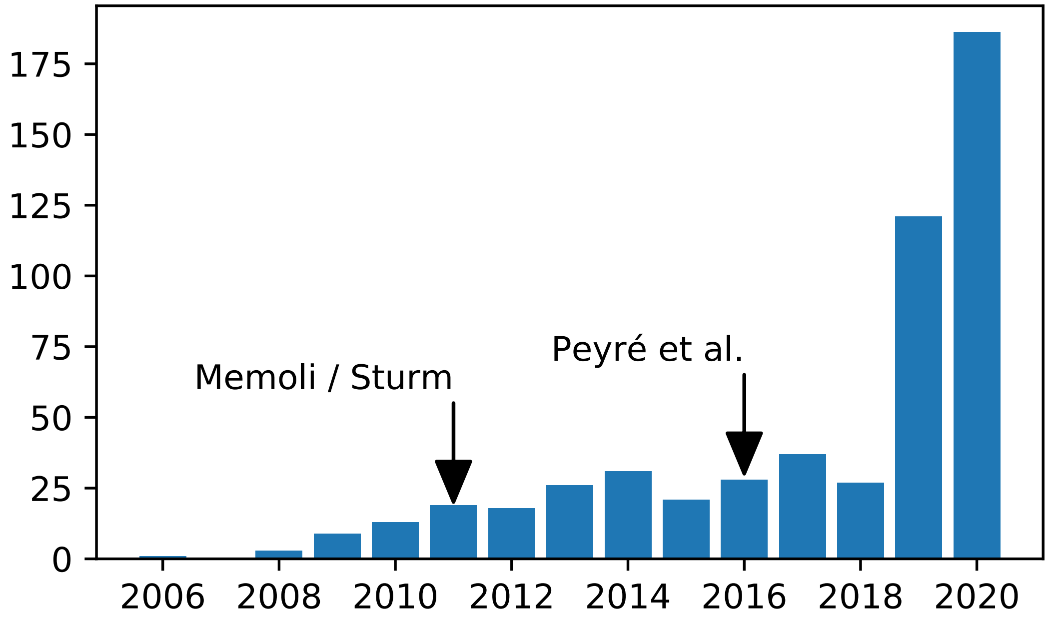
[Solomon 2016]

Applications for geometric data

Barycenter of relational data [Peyré 2016], Point clouds/meshes [Ezuz 2017]

Shape comparison [Mémoli 2011, Solomon 2016]

Graphs [Xu 2019, Fey 2020], biology [Demetci 2020], generative modeling [Bunne 2019]



Solving OT

Solving OT

A linear problem

Discrete probability measures

$\mu = \sum_{i=1}^{n} a_i \delta_{x_i}$ $\nu = \sum_{j=1}^{m} b_j \delta_{y_j}$

Linear Program:

$$\min_{\boldsymbol{\pi} \in \Pi(\mathbf{a}, \mathbf{b})} \sum_{ij} c_{i,j} \pi_{i,j} = \min_{\boldsymbol{\pi} \in \Pi(\mathbf{a}, \mathbf{b})} \langle \mathbf{C}, \boldsymbol{\pi} \rangle$$

Simplex, Network flow, Hungarian algorithms $\sim O(n^3 \log(n))$

Solving OT

A linear problem

Discrete probability measures

$\mu = \sum_{i=1}^{n} a_i \delta_{x_i} \quad \nu = \sum_{j=1}^{m} b_j \delta_{y_j}$

Linear Program:

$$\min_{\boldsymbol{\pi} \in \Pi(\mathbf{a},\mathbf{b})} \sum_{ij} c_{i,j}\pi_{i,j} = \min_{\boldsymbol{\pi} \in \Pi(\mathbf{a},\mathbf{b})} \langle \mathbf{C}, \boldsymbol{\pi} \rangle$$

Simplex, Network flow, Hungarian algorithms $\sim O(n^3 \log(n))$

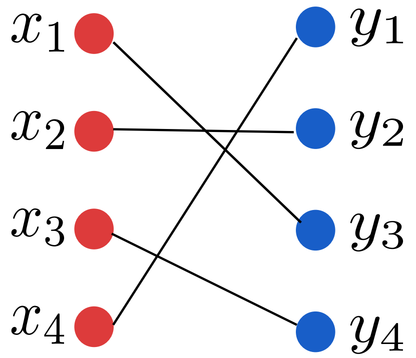
Uniform weights

$$\mathbf{a} = \mathbf{b} = \frac{\mathbf{1}_n}{n}$$

Monge Problem

$$\min_{\sigma \in S_n} \sum_{i=1}^{n} c_{i,\sigma(i)}$$

One-to-one



Solving OT

A linear problem

Discrete probability measures

$\mu = \sum_{i=1}^n a_i \delta_{x_i} \quad \nu = \sum_{j=1}^m b_j \delta_{y_j}$

Linear Program:

$$\min_{\boldsymbol{\pi} \in \Pi(\mathbf{a},\mathbf{b})} \sum_{ij} c_{i,j} \pi_{i,j} = \min_{\boldsymbol{\pi} \in \Pi(\mathbf{a},\mathbf{b})} \langle \mathbf{C}, \boldsymbol{\pi} \rangle$$

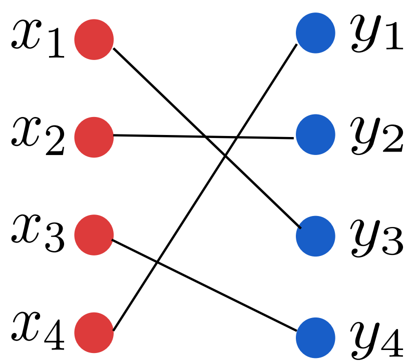
Simplex, Network flow, Hungarian algorithms $\sim O(n^3 \log(n))$

Uniform weights

$$\mathbf{a} = \mathbf{b} = \frac{\mathbf{1}_n}{n}$$

Monge Problem

$$\min_{\sigma \in S_n} \sum_{i=1}^n c_{i,\sigma(i)}$$



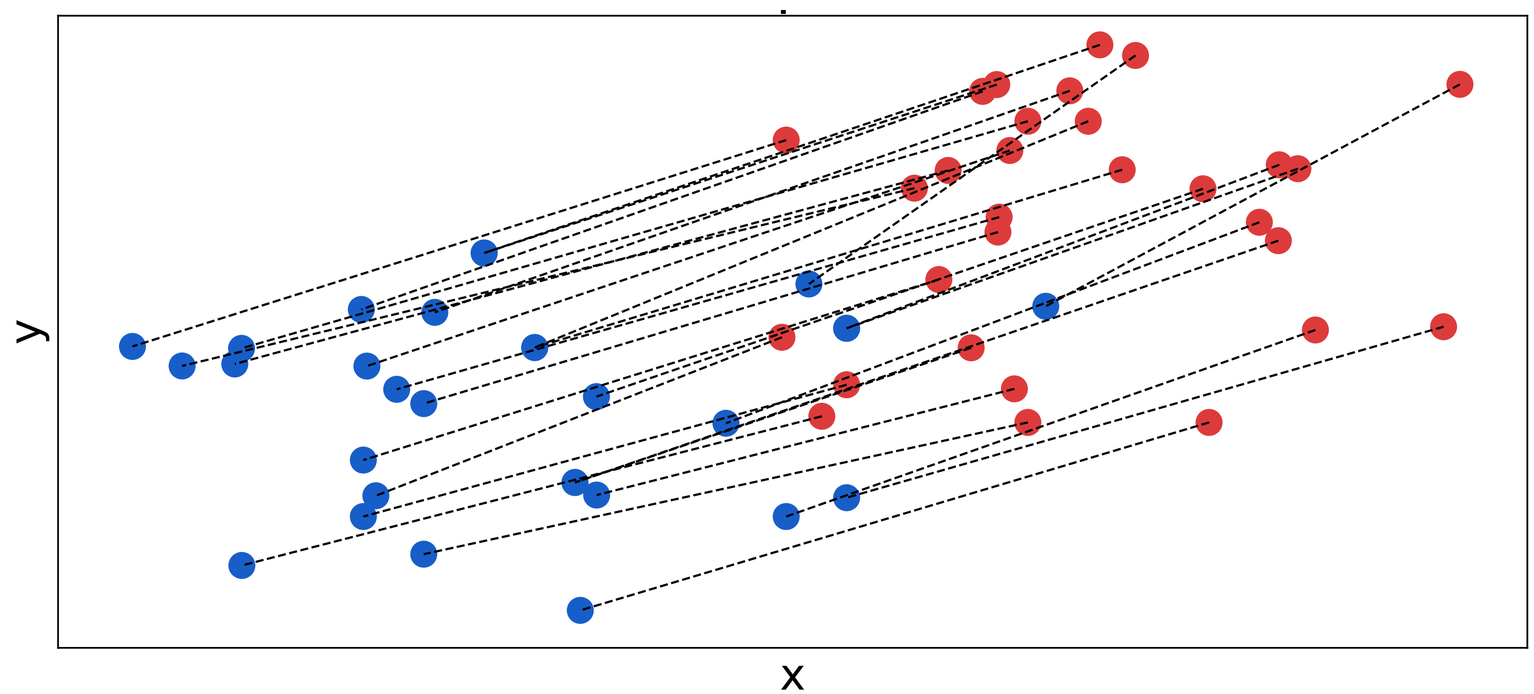
One-to-one

Fundamental theorem LP:

$$\boldsymbol{\pi}^* \leftrightarrow \sigma^* \in S_n$$

Optimal coupling is a permutation

Solves the Monge Problem



Solving OT

Entropic regularization

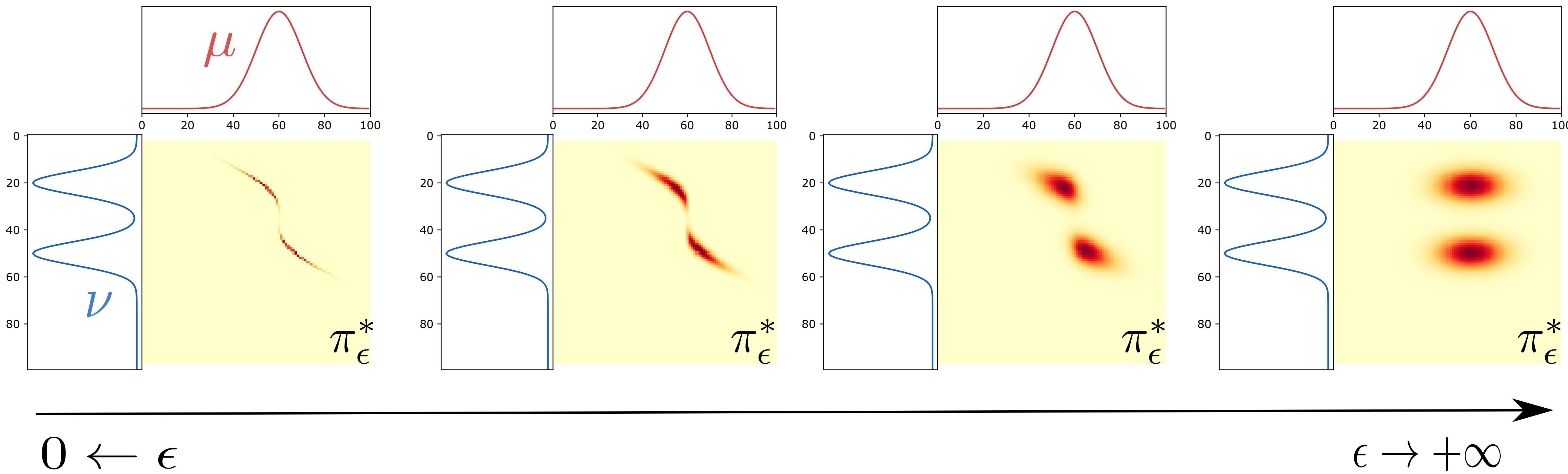
Discrete probability measures

$\mu = \sum_{i=1}^{n} a_i \delta_{x_i}$ $\nu = \sum_{j=1}^{m} b_j \delta_{y_j}$

Strongly convex problem:

$$\min_{\boldsymbol{\pi} \in \Pi(\mathbf{a},\mathbf{b})} \langle \mathbf{C}, \boldsymbol{\pi} \rangle - \varepsilon H(\boldsymbol{\pi})$$

- Entropy term $H(\boldsymbol{\pi}) = -\sum_{ij} (\log(\pi_{ij}) - 1)\pi_{ij}$
- Sinkhorn-Knopp algorithm: 1) fast 2) based on matrix multiplication
- τ approximate solution $\sim O(n^2 \log(n) \tau^{-3})$



Solving OT

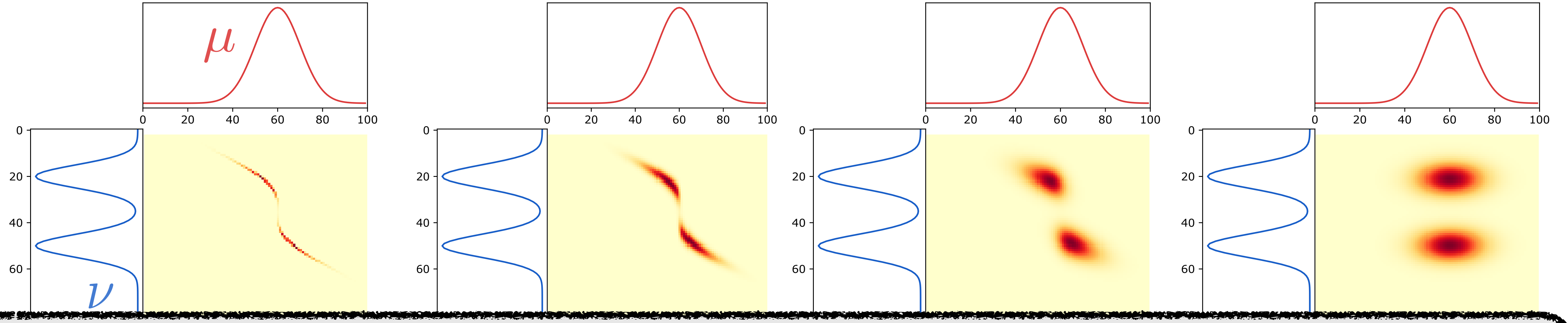
Entropic regularization

Discrete probability measures

$\mu = \sum_{i=1}^{n} a_i \delta_{x_i}$ $\nu = \sum_{j=1}^{m} b_j \delta_{y_j}$

Strongly convex problem: $$\min_{\boldsymbol{\pi} \in \Pi(\mathbf{a},\mathbf{b})} \langle \mathbf{C}, \boldsymbol{\pi} \rangle - \varepsilon H(\boldsymbol{\pi})$$

- Entropy term $H(\boldsymbol{\pi}) = -\sum_{ij} (\log(\pi_{ij}) - 1)\pi_{ij}$
- Sinkhorn-Knopp algorithm: 1) fast 2) based on matrix multiplication
- τ approximate solution $\sim O(n^2 \log(n) \tau^{-3})$



Linear OT: costly but solvable in practice

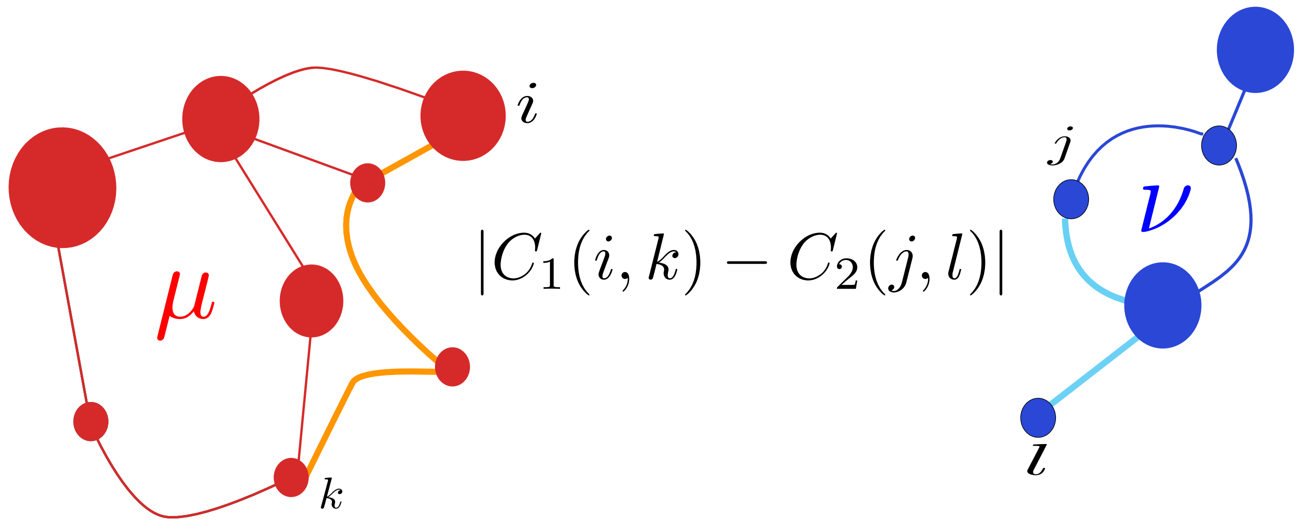
Solving OT

A quadratic problem (QP)

Discrete probability measures

$$\mu = \sum_{i=1}^{n} a_i \delta_{x_i} \quad \nu = \sum_{j=1}^{m} b_j \delta_{y_j}$$

$$\mathcal{X}, \mathcal{Y} \not\subset \Omega$$



$$\min_{\pi \in \Pi(\mathbf{a},\mathbf{b})} \sum_{ijkl} |C_1(i,k) - C_2(j,l)|^p \pi_{ij} \pi_{kl}$$

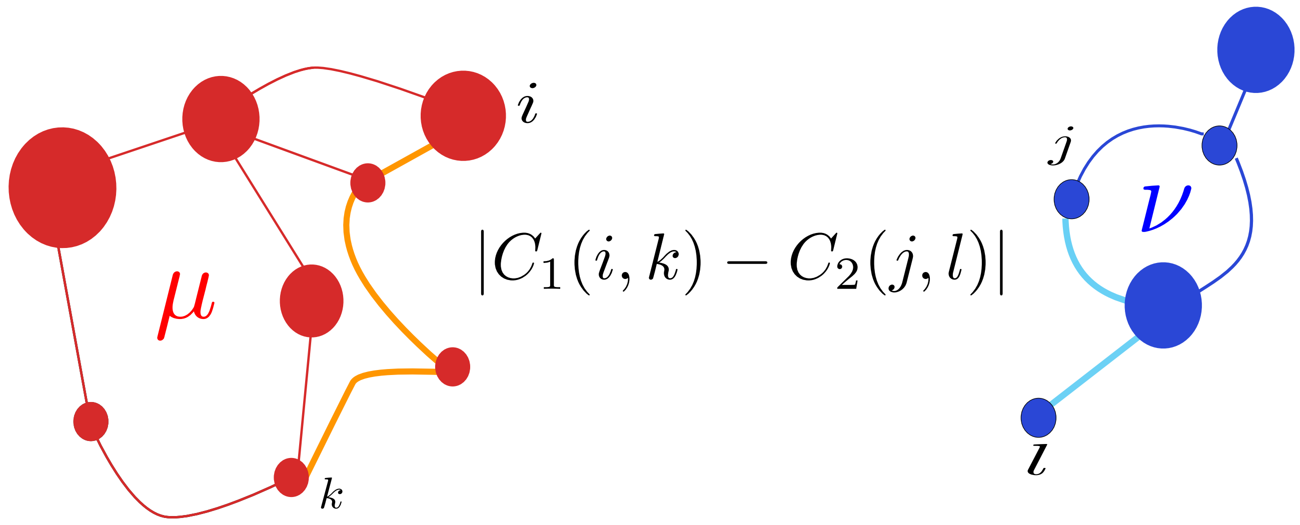
Solving OT

A quadratic problem (QP)

Discrete probability measures

$\mu = \sum_{i=1}^{n} a_i \delta_{x_i}$ $\nu = \sum_{j=1}^{m} b_j \delta_{y_j}$

$\mathcal{X}, \mathcal{Y} \not\subset \Omega$



$$\min_{\boldsymbol{\pi} \in \Pi(\mathbf{a},\mathbf{b})} \sum_{ijkl} |C_1(i,k) - C_2(j,l)|^p \pi_{ij} \pi_{kl}$$

Non convex QP: NP-hard in general (graph matching problem)

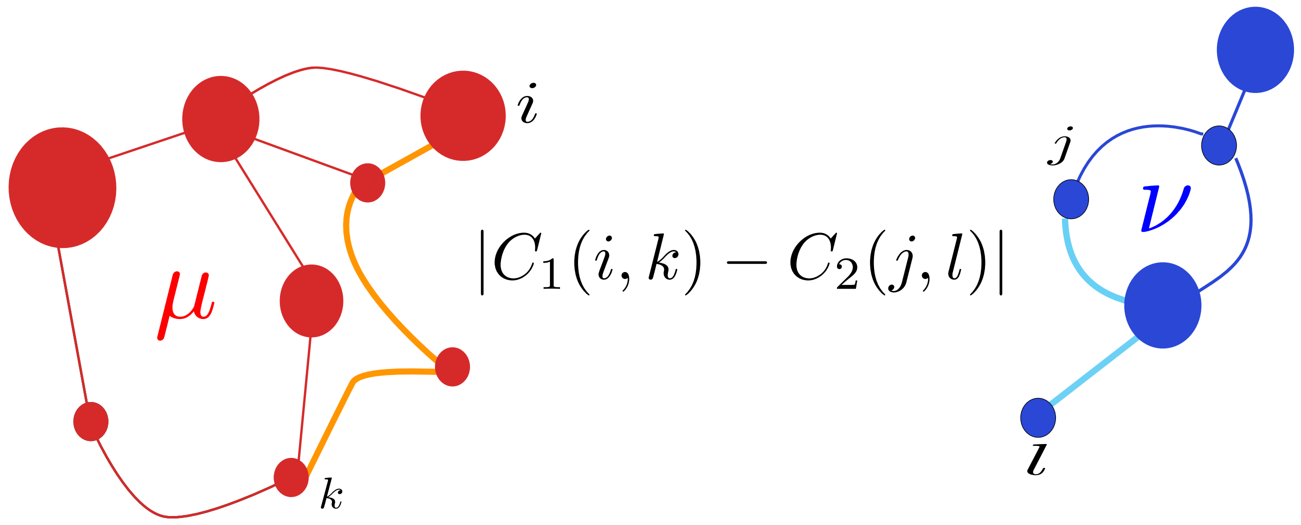
Solving OT

A quadratic problem (QP)

Discrete probability measures

$\mu = \sum_{i=1}^{n} a_i \delta_{x_i}$ $\nu = \sum_{j=1}^{m} b_j \delta_{y_j}$

$\mathcal{X}, \mathcal{Y} \not\subset \Omega$



$$\min_{\boldsymbol{\pi} \in \Pi(\mathbf{a},\mathbf{b})} \sum_{ijkl} |C_1(i,k) - C_2(j,l)|^p \pi_{ij}\pi_{kl} - \varepsilon H(\boldsymbol{\pi})$$

- Non convex QP: NP-hard in general
- With entropic regularization [Peyré 2016, Solomon 2016]
- Can be solved using projected gradient descent under KL geometry
- Each gradient step: Sinkhorn algorithm

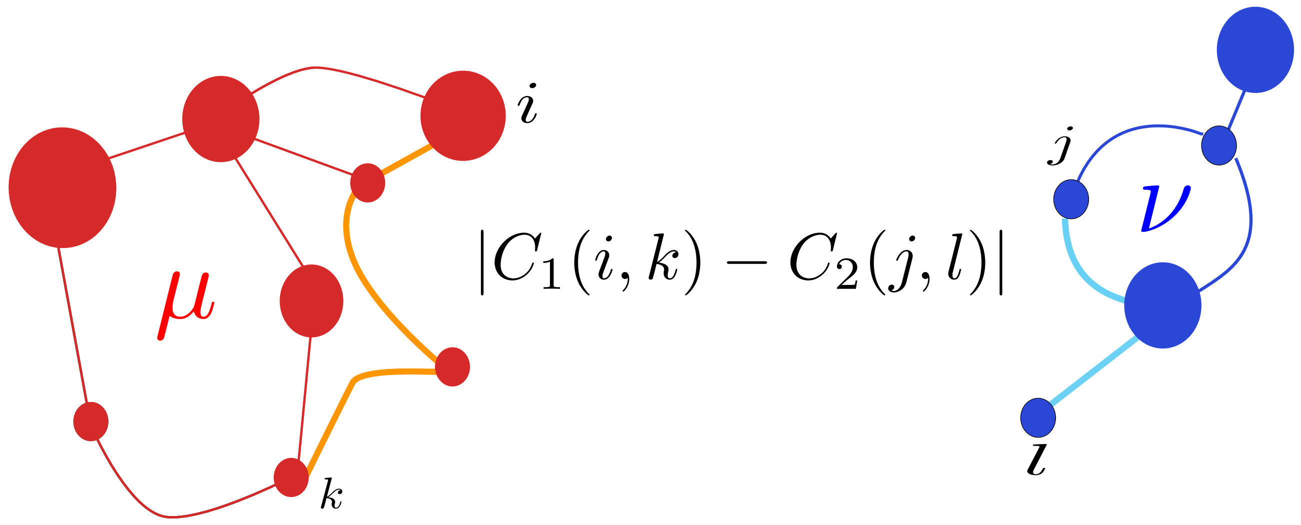
Solving OT

A quadratic problem (QP)

Discrete probability measures

$\mu = \sum_{i=1}^{n} a_i \delta_{x_i}$ $\nu = \sum_{j=1}^{m} b_j \delta_{y_j}$

$\mathcal{X}, \mathcal{Y} \not\subset \Omega$



$$\min_{\boldsymbol{\pi} \in \Pi(\mathbf{a},\mathbf{b})} \sum_{ijkl} |C_1(i,k) - C_2(j,l)|^p \pi_{ij}\pi_{kl} - \varepsilon H(\boldsymbol{\pi})$$

- Non convex QP: NP-hard in general
- With entropic regularization [Peyré 2016, Solomon 2016] $\sim O(n_{iter} * n^2 \log(n))$
- Can be solved using projected gradient descent under KL geometry
- Each gradient step: Sinkhorn algorithm

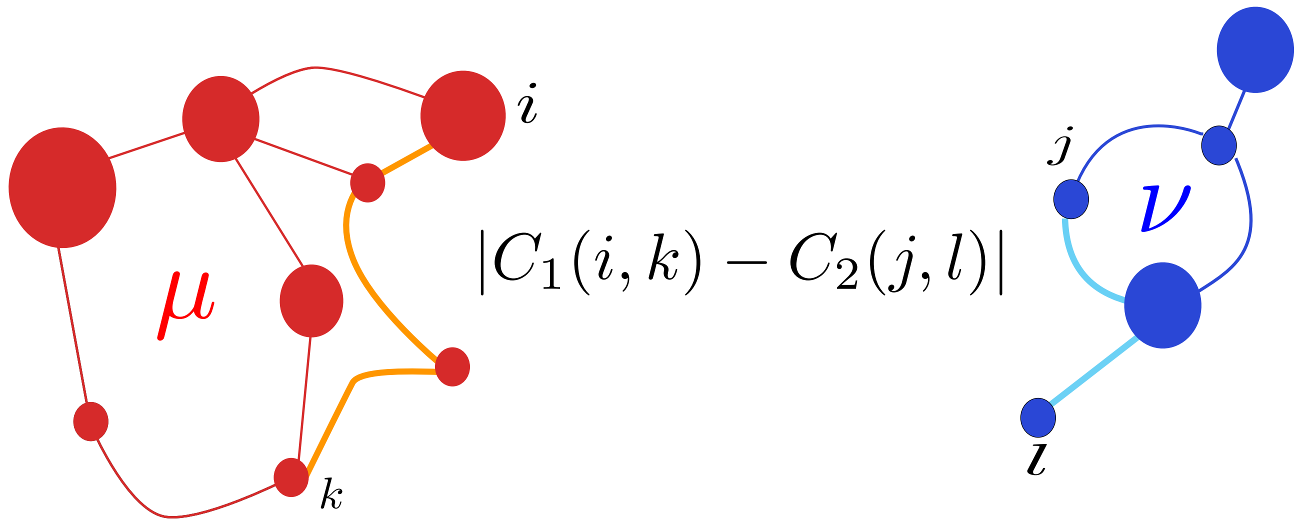
Solving OT

A quadratic problem (QP)

Discrete probability measures

$\mu = \sum_{i=1}^{n} a_i \delta_{x_i}$ $\nu = \sum_{j=1}^{m} b_j \delta_{y_j}$

$\mathcal{X}, \mathcal{Y} \not\subset \Omega$



$$\min_{\boldsymbol{\pi} \in \Pi(\mathbf{a},\mathbf{b})} \sum_{ijkl} |C_1(i,k) - C_2(j,l)|^p \pi_{ij} \pi_{kl}$$

- Non convex QP: NP-hard in general
- With entropic regularization [Peyré 2016, Solomon 2016] $\sim O(n_{iter} * n^2 \log(n))$
- Can be solved using projected gradient descent under KL geometry
- Each gradient step: Sinkhorn algorithm

Hard to solve and even to approximate…

Solving OT

Computing GW

Solving FGW: a non convex QP

$$\min_{\pi \in \Pi(\mathbf{a},\mathbf{b})} \sum\nolimits_{ijkl} |C_1(i,k) - C_2(j,l)|^p \pi_{ij}\pi_{kl}$$

Quadratic function over polytope -> Conditional Gradient algorithm (a.k.a Frank-Wolfe)

Non convex but converges to a **local optimal solution** [Lacoste-Julien 2016]

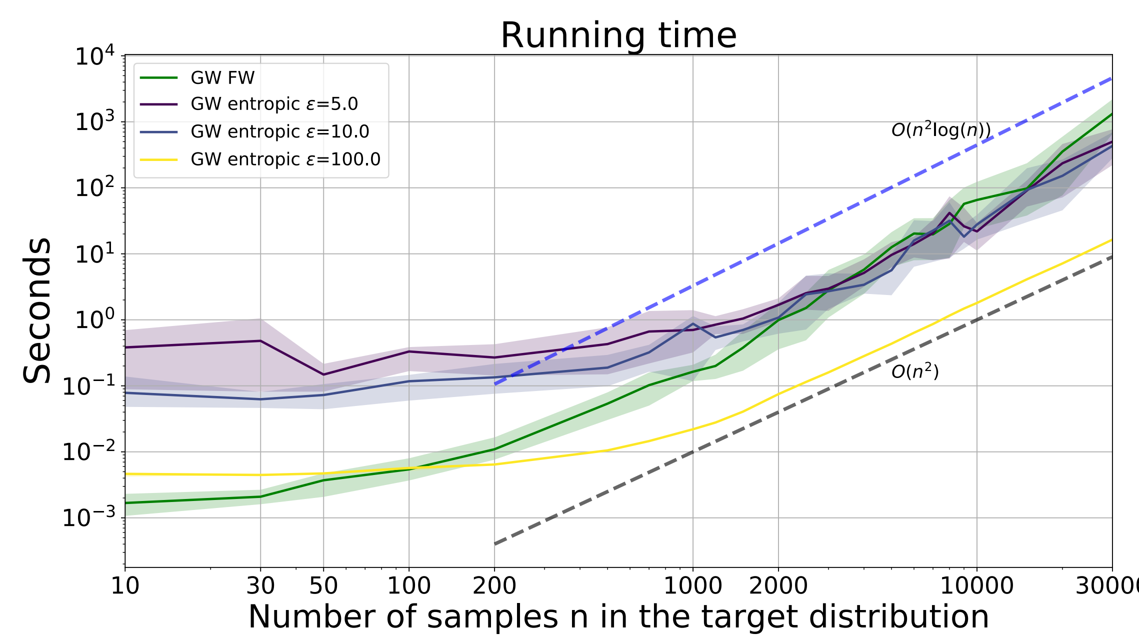
Find a **sparse** solution. FW gap = $O(\frac{1}{\sqrt{n_{iter}}})$

Algorithm 1 Conditional Gradient (CG) for *FGW*

$\pi^{(0)} \leftarrow \mathbf{h}\mathbf{g}^\top$
for $i = 1, \dots,$ **do**
  $\mathbf{G} \leftarrow$ Gradient from GW loss *w.r.t.* $\boldsymbol{\pi}^{(i-1)}$
  $\tilde{\boldsymbol{\pi}}^{(i)} \leftarrow$ Solve OT with ground loss $\mathbf{G}$
  $\tau^{(i)} \leftarrow$ Line-search for GW loss with $\tau \in (0,1)$ (closed-form)
  $\boldsymbol{\pi}^{(i)} \leftarrow (1-\tau^{(i)})\boldsymbol{\pi}^{(i-1)} + \tau^{(i)}\tilde{\boldsymbol{\pi}}^{(i)}$
end for

Complexity

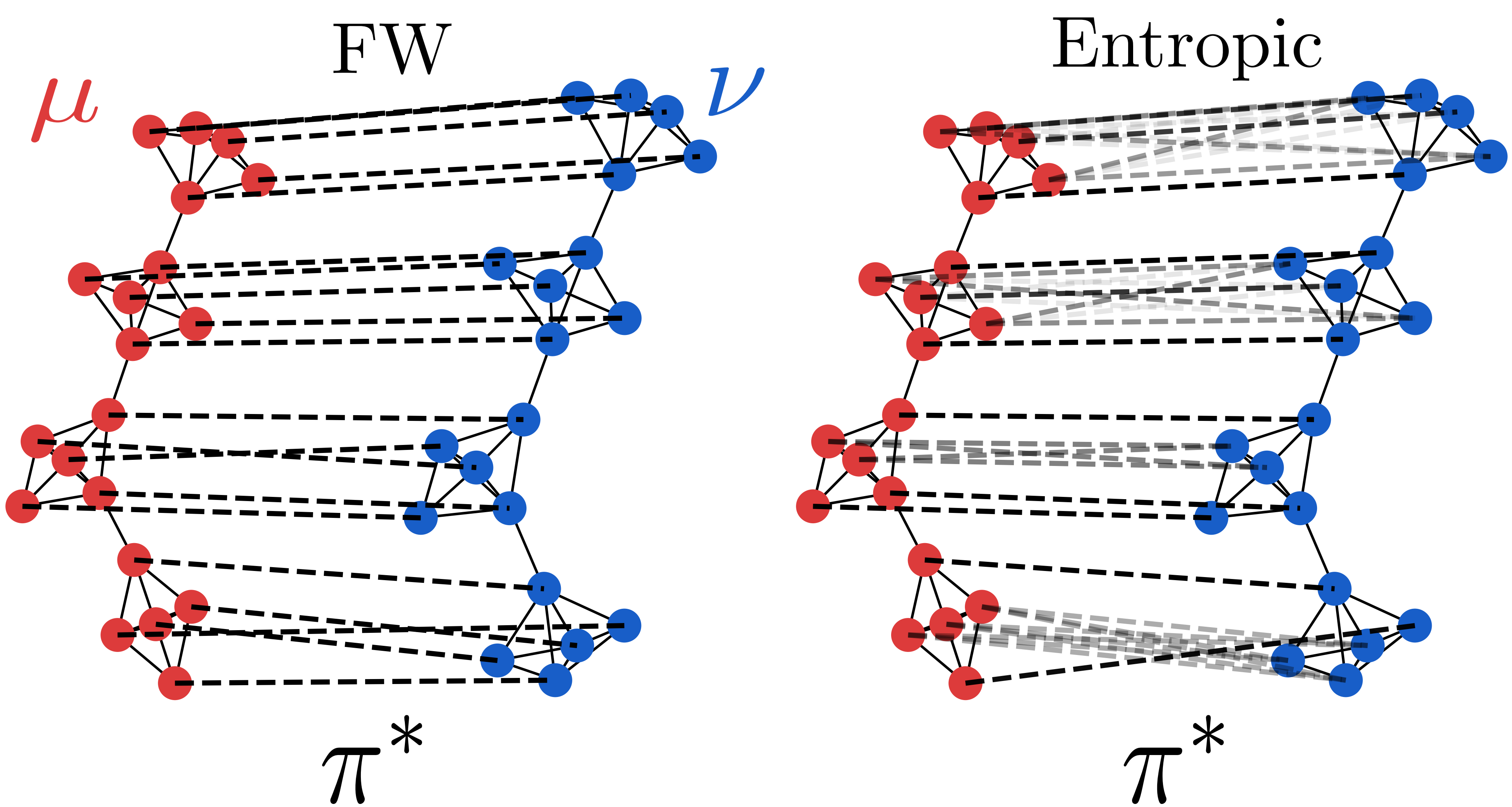
$O(n_{iter}\ n^3)$



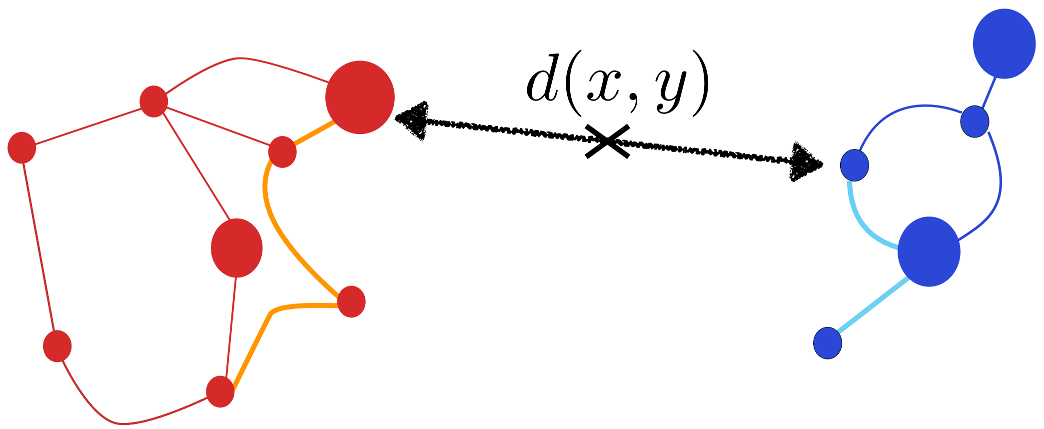
...to Gromov-Wasserstein

An example on graphs

> $\mathbf{C}_1, \mathbf{C}_2$ are the shortest path distance in each graph



Optimal transport for structured data



Optimal Transport for structured data

Motivations

Motivation: Is the Optimal transport framework suited for structured data ?

Problem 1: How do we model structured data ?

As probability distributions!

Problem 2: How do we compare structured data ?

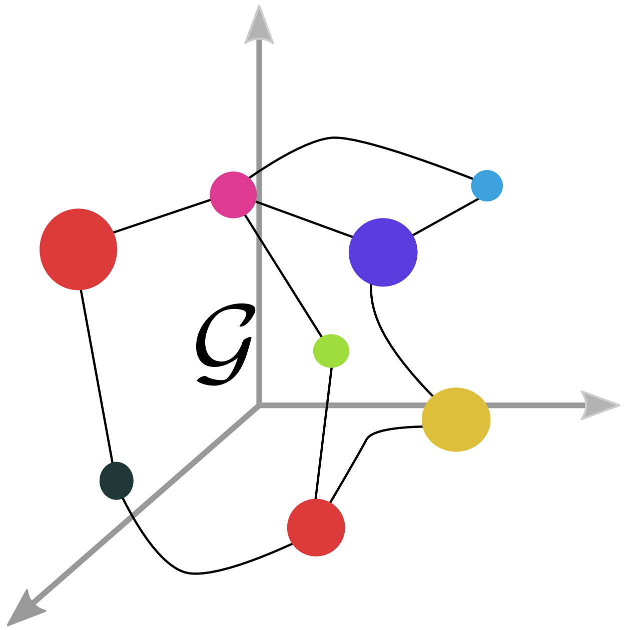
Based on the theories of Wasserstein and Gromov-Wasserstein

Optimal Transport for structured data

Structured data as probability distribution

Discrete case

- Structured data can be seen as a labeled graph
- Combines a feature **and** a structure information



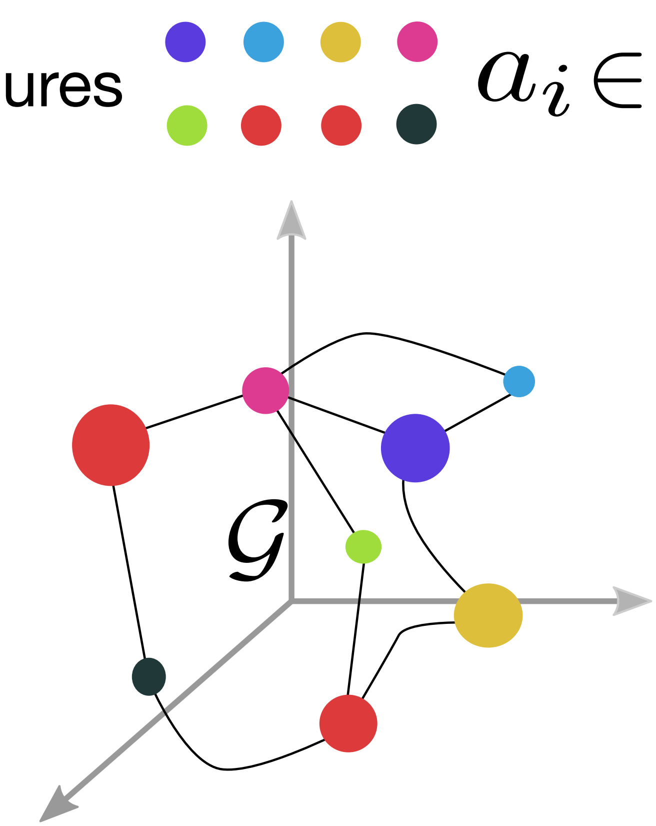
Optimal Transport for structured data

Structured data as probability distribution

Discrete case

- Structured data can be seen as a labeled graph
- Combines a feature **and** a structure information

Features $a_i \in \Omega$

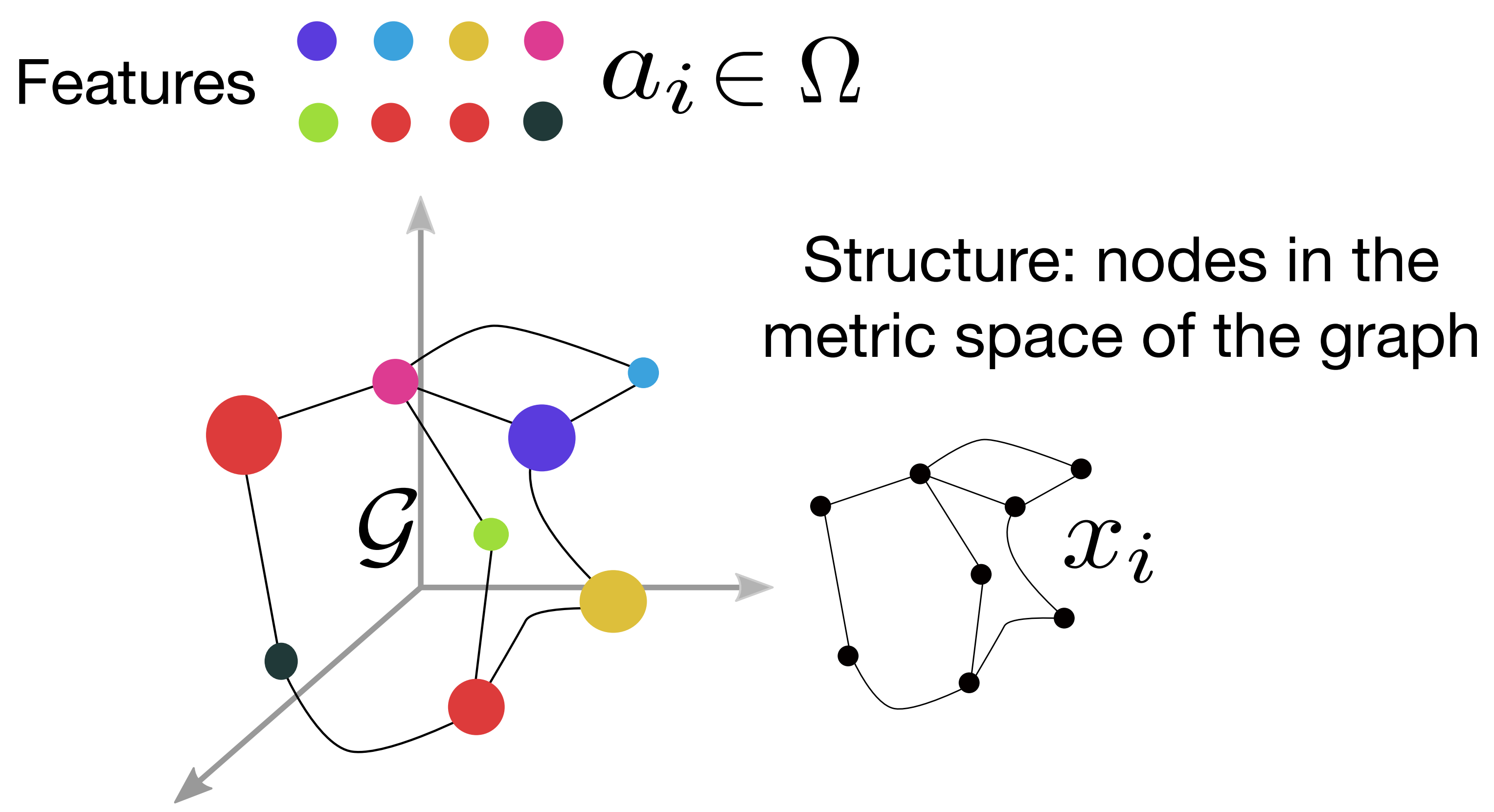


Optimal Transport for structured data

Structured data as probability distribution

Discrete case

- Structured data can be seen as a labeled graph
- Combines a feature **and** a structure information

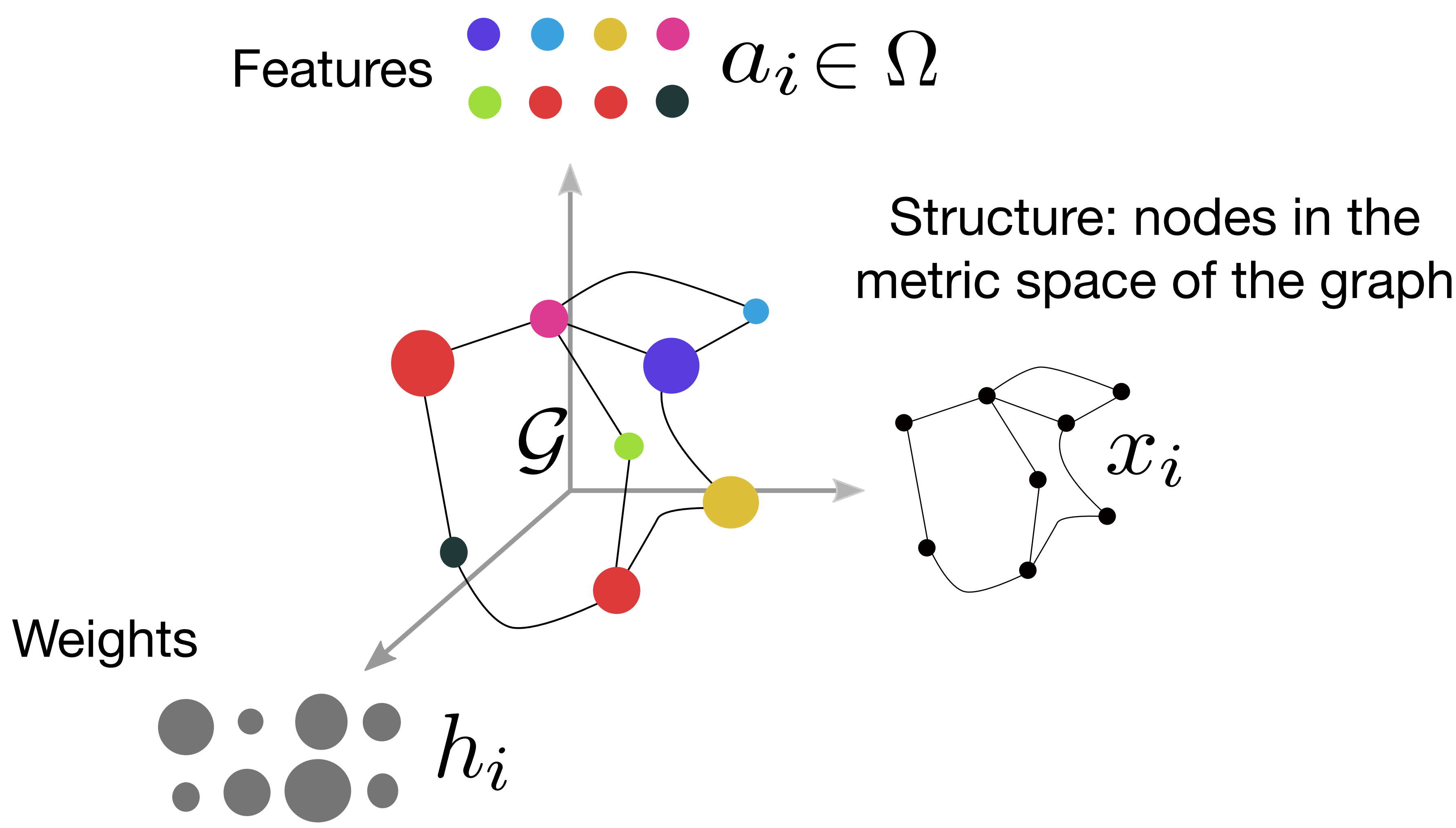


Optimal Transport for structured data

Structured data as probability distribution

Discrete case

- Structured data can be seen as a labeled graph
- Combines a feature **and** a structure information
- Add weights that encodes the relative importance of the nodes

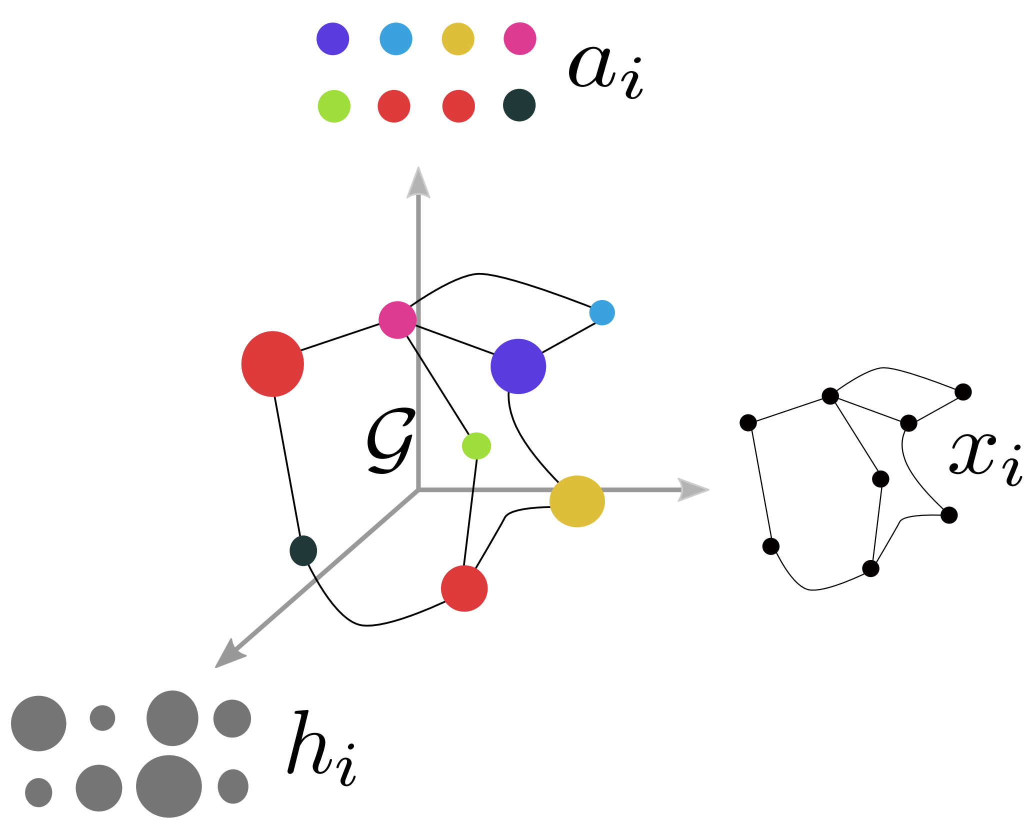


Optimal Transport for structured data

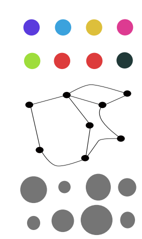
Structured data as probability distribution

Discrete case

- Structured data can be seen as a labeled graph
- Combines a feature **and** a structure information
- Add weights that encodes the relative importance of the nodes

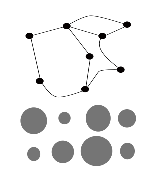


Form a probability measure



$$\mu = \sum_i h_i \delta_{(x_i, a_i)}$$

$$\mu_A = \sum_i h_i \delta_{a_i}$$



$$\mu_X = \sum_i h_i \delta_{x_i}$$

Optimal Transport for structured data

Fused Gromov-Wasserstein distance

Two structured data

$$\mu = \sum_i h_i \delta_{(x_i, a_i)}, \nu = \sum_j g_j \delta_{(y_j, b_j)}$$

Two matrices describing structures

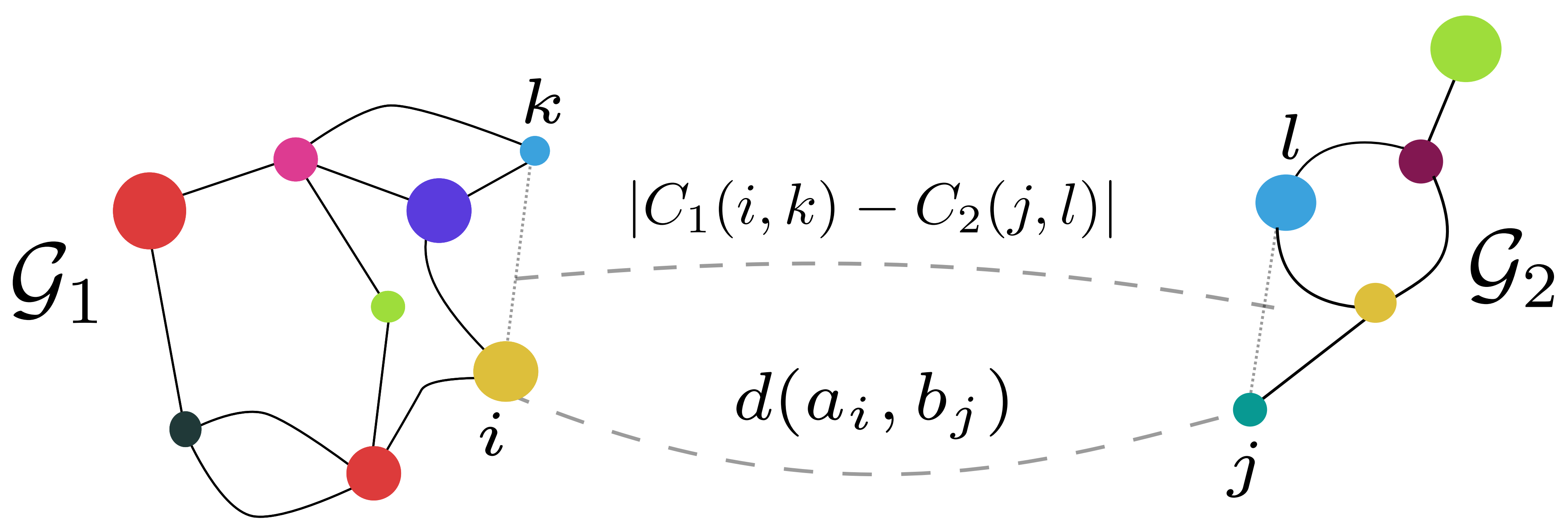
$$\mathbf{C}_1, \mathbf{C}_2$$

A distance between labels

$$d : \Omega \times \Omega \to \mathbb{R}_+$$

Fused Gromov-Wasserstein distance

$$FGW(\mathbf{M_{AB}}, \mathbf{C}_1, \mathbf{C}_2, \mathbf{h}, \mathbf{g}) = \min_{\boldsymbol{\pi} \in \Pi(\mathbf{h}, \mathbf{g})} \sum_{i,j,k,l} (1-\alpha) d(a_i, b_j)^q + \alpha |C_1(i,k) - C_2(j,l)|^q \pi_{i,j} \pi_{k,l}$$



Optimal Transport for structured data

Fused Gromov-Wasserstein distance

Two structured data

$$\mu = \sum_i h_i \delta_{(x_i, a_i)}, \nu = \sum_j g_j \delta_{(y_j, b_j)}$$

Two matrices describing structures

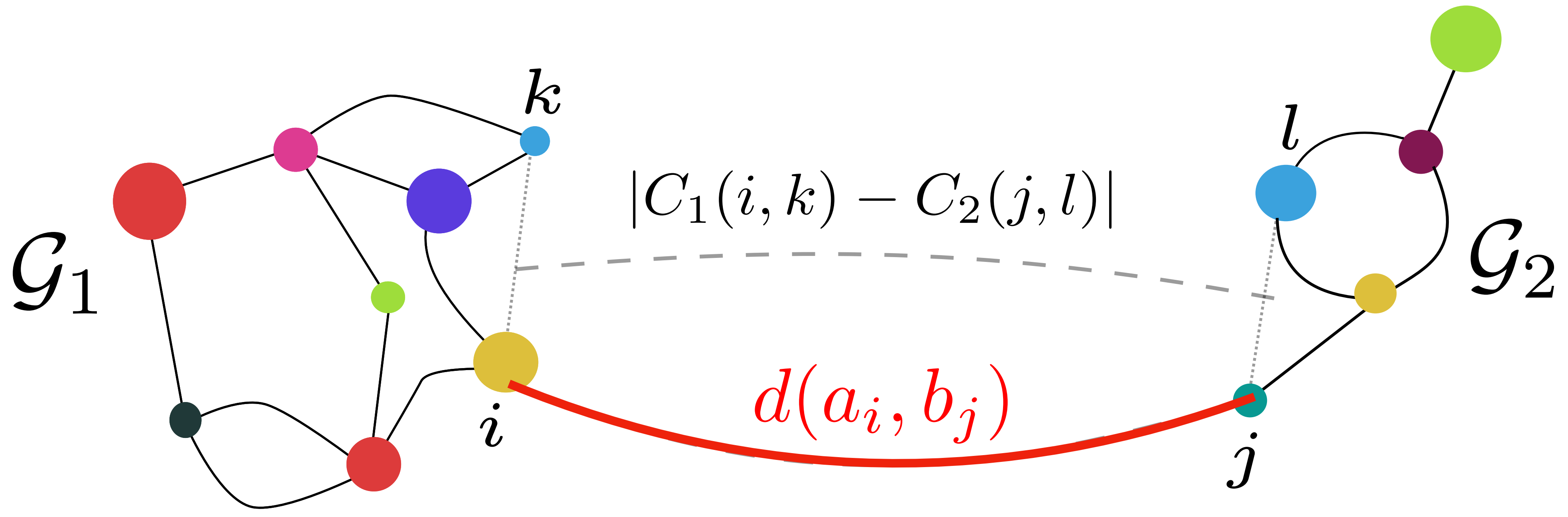
$$\mathbf{C}_1, \mathbf{C}_2$$

A distance between labels

$$d : \Omega \times \Omega \to \mathbb{R}_+$$

Fused Gromov-Wasserstein distance

$$FGW(\mathbf{M_{AB}}, \mathbf{C}_1, \mathbf{C}_2, \mathbf{h}, \mathbf{g}) = \min_{\boldsymbol{\pi} \in \Pi(\mathbf{h}, \mathbf{g})} \sum_{i,j,k,l} (1-\alpha) d(a_i, b_j)^q + \alpha |C_1(i,k) - C_2(j,l)|^q \pi_{i,j} \pi_{k,l}$$



Optimal Transport for structured data

Fused Gromov-Wasserstein distance

Two structured data

$\mu = \sum_i h_i \delta_{(x_i, a_i)}, \nu = \sum_j g_j \delta_{(y_j, b_j)}$

Two matrices describing structures

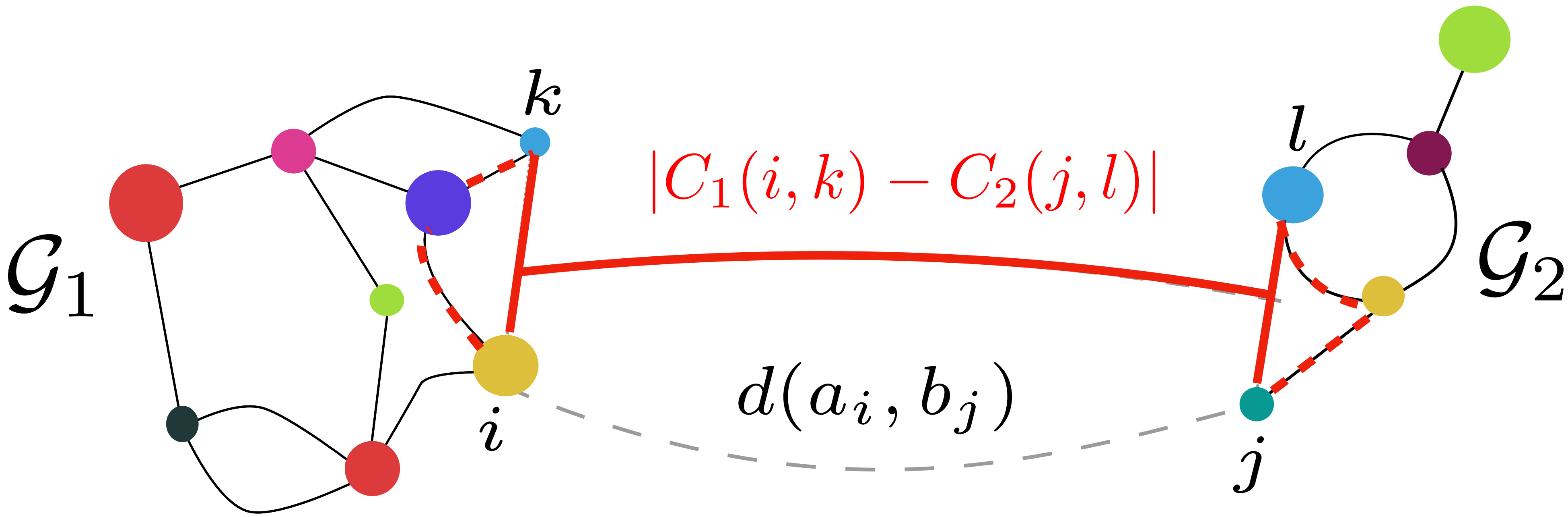
$\mathbf{C}_1, \mathbf{C}_2$

A distance between labels

$d : \Omega \times \Omega \rightarrow \mathbb{R}_+$

Fused Gromov-Wasserstein distance

$$FGW(\mathbf{M_{AB}}, \mathbf{C}_1, \mathbf{C}_2, \mathbf{h}, \mathbf{g}) = \min_{\boldsymbol{\pi} \in \Pi(\mathbf{h}, \mathbf{g})} \sum_{i,j,k,l} (1-\alpha) d(a_i, b_j)^q + \alpha |C_1(i,k) - C_2(j,l)|^q \pi_{i,j} \pi_{k,l}$$



Optimal Transport for structured data

Fused Gromov-Wasserstein distance

Two structured data

$\mu = \sum_i h_i \delta_{(x_i, a_i)}, \nu = \sum_j g_j \delta_{(y_j, b_j)}$

Two matrices describing structures

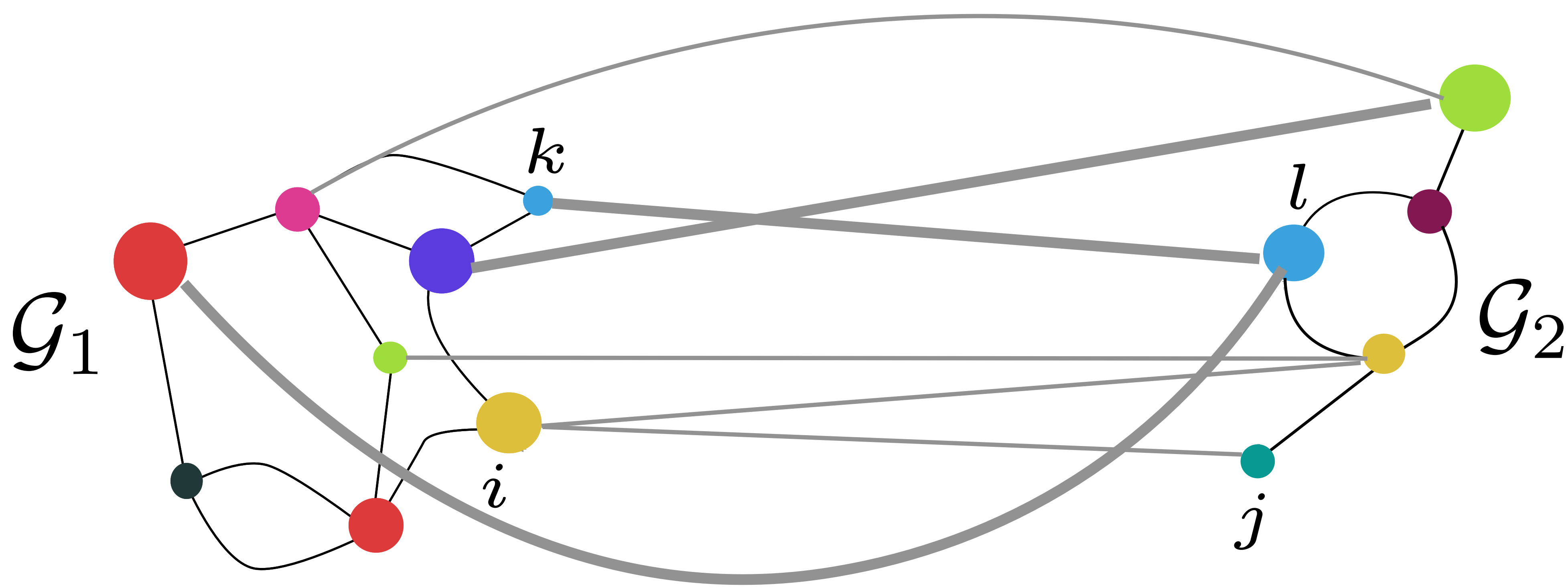
$\mathbf{C}_1, \mathbf{C}_2$

A distance between labels

$d : \Omega \times \Omega \to \mathbb{R}_+$

Fused Gromov-Wasserstein distance

$$FGW(\mathbf{M_{AB}}, \mathbf{C}_1, \mathbf{C}_2, \mathbf{h}, \mathbf{g}) = \min_{\boldsymbol{\pi} \in \Pi(\mathbf{h}, \mathbf{g})} \sum_{i,j,k,l} (1-\alpha) d(a_i, b_j)^q + \alpha |C_1(i,k) - C_2(j,l)|^q \pi_{i,j} \pi_{k,l}$$

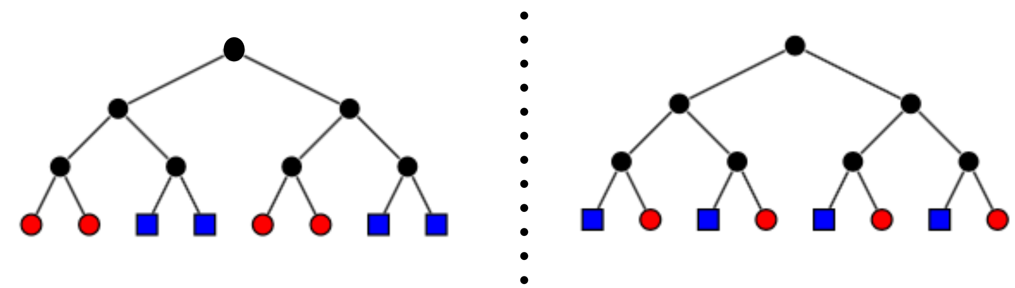


$\boldsymbol{\pi}$ provides a soft assignment of the nodes

Optimal Transport for structured data

Fused Gromov-Wasserstein distance: example

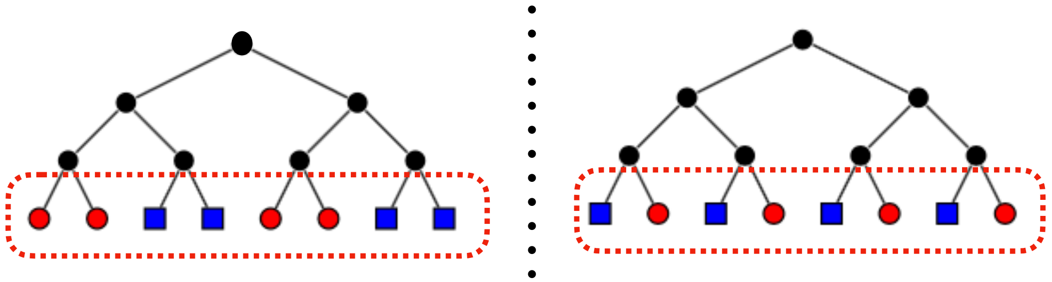
Consider two trees



Optimal Transport for structured data

Fused Gromov-Wasserstein distance: example

Consider two trees

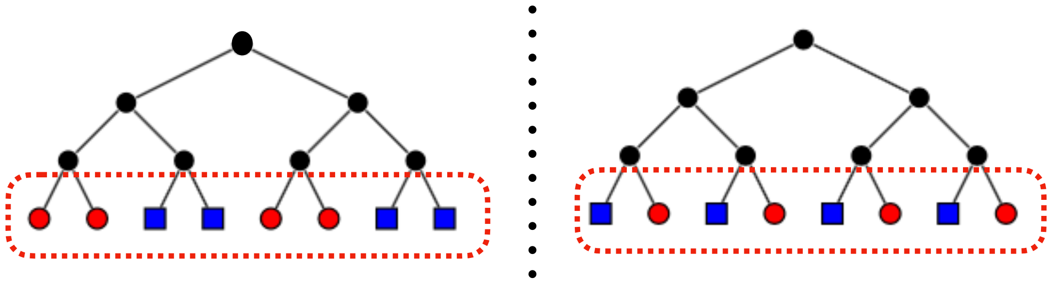


We want to compare the leaves of the trees

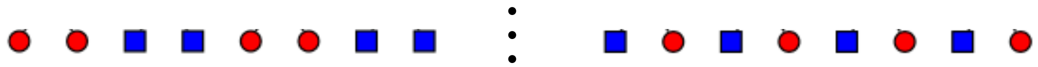
Optimal Transport for structured data

Fused Gromov-Wasserstein distance: example

Consider two trees

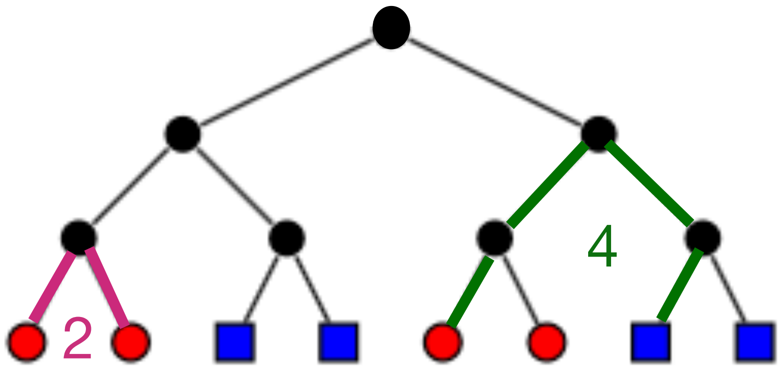


We want to compare the leaves of the trees



Features: blue or red

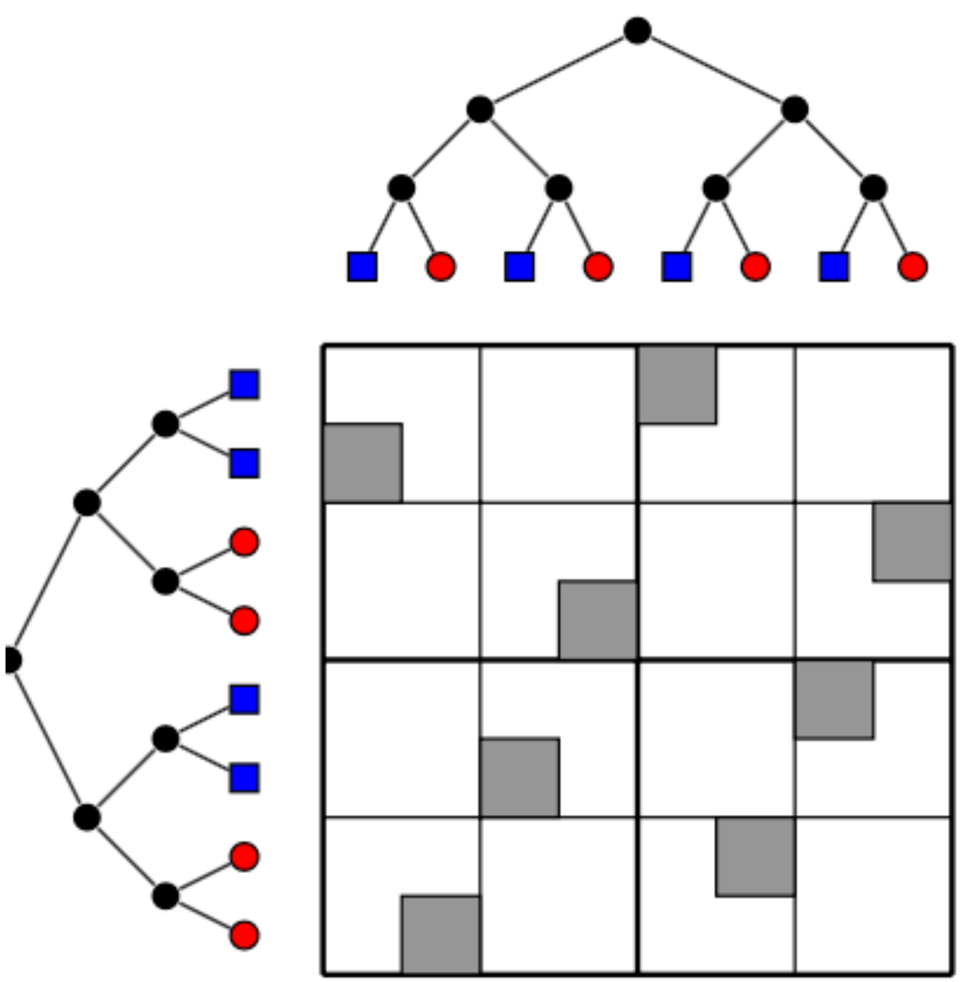
Structures : shortest path between the leaves



Optimal Transport for structured data

Fused Gromov-Wasserstein distance: example

Wasserstein distance (features only)

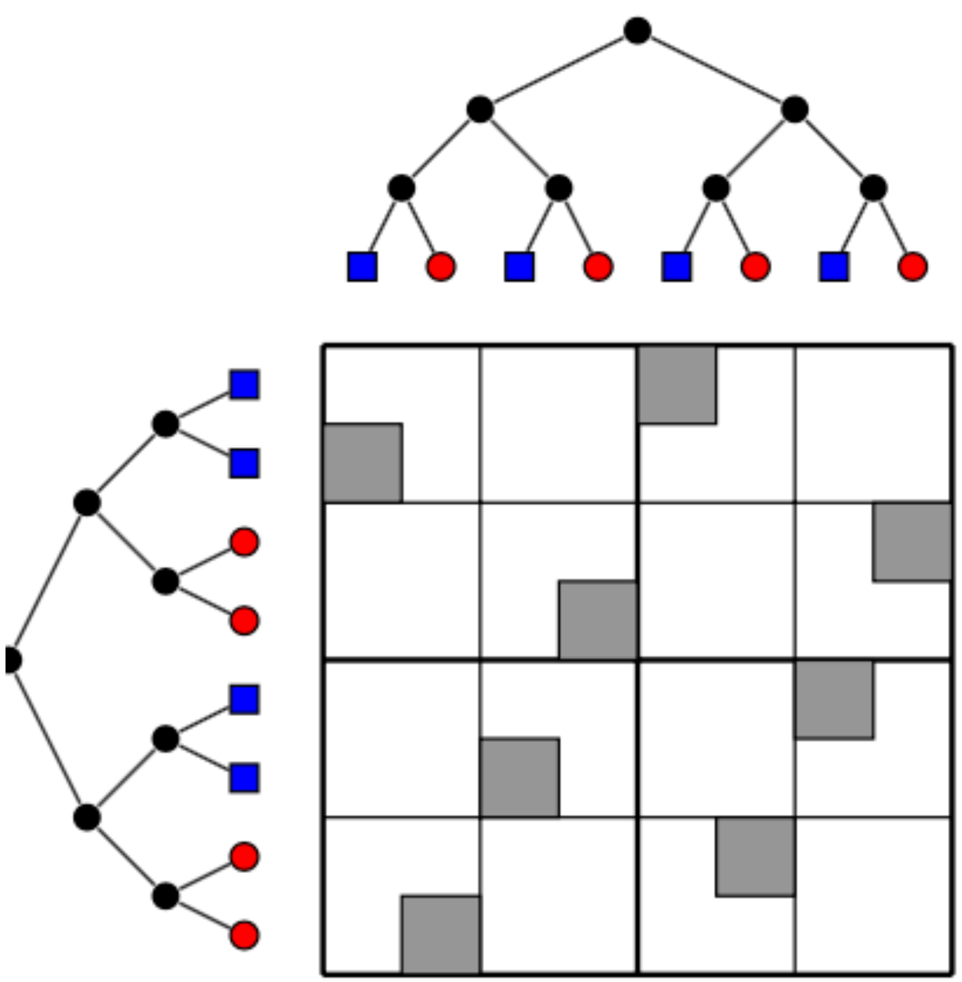


$W = 0$

Optimal Transport for structured data

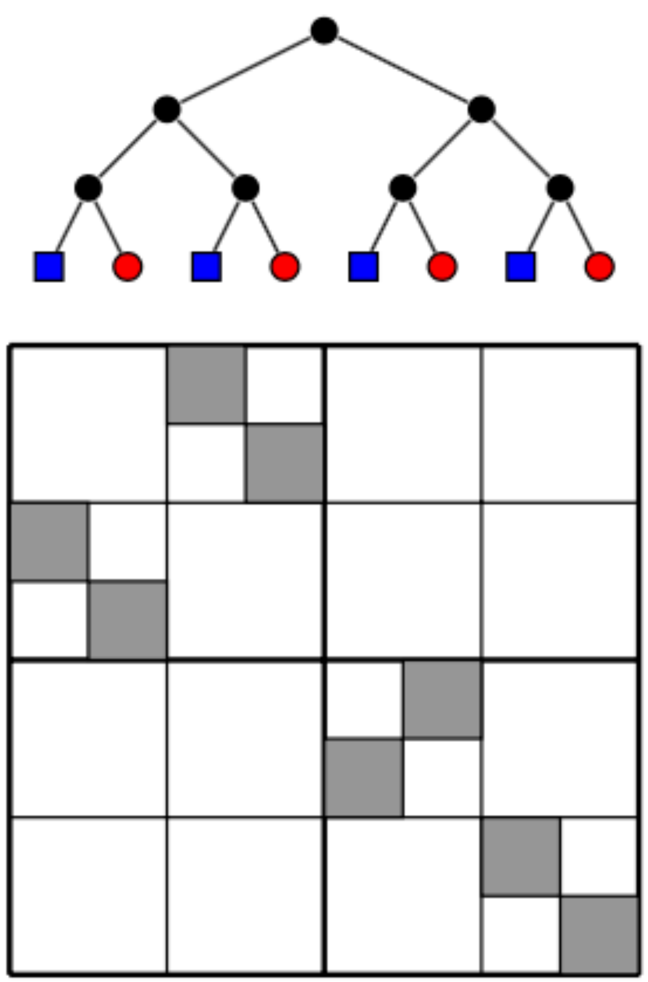
Fused Gromov-Wasserstein distance: example

Wasserstein distance (features only)



$W = 0$

Gromov-Wasserstein distance (structures only)

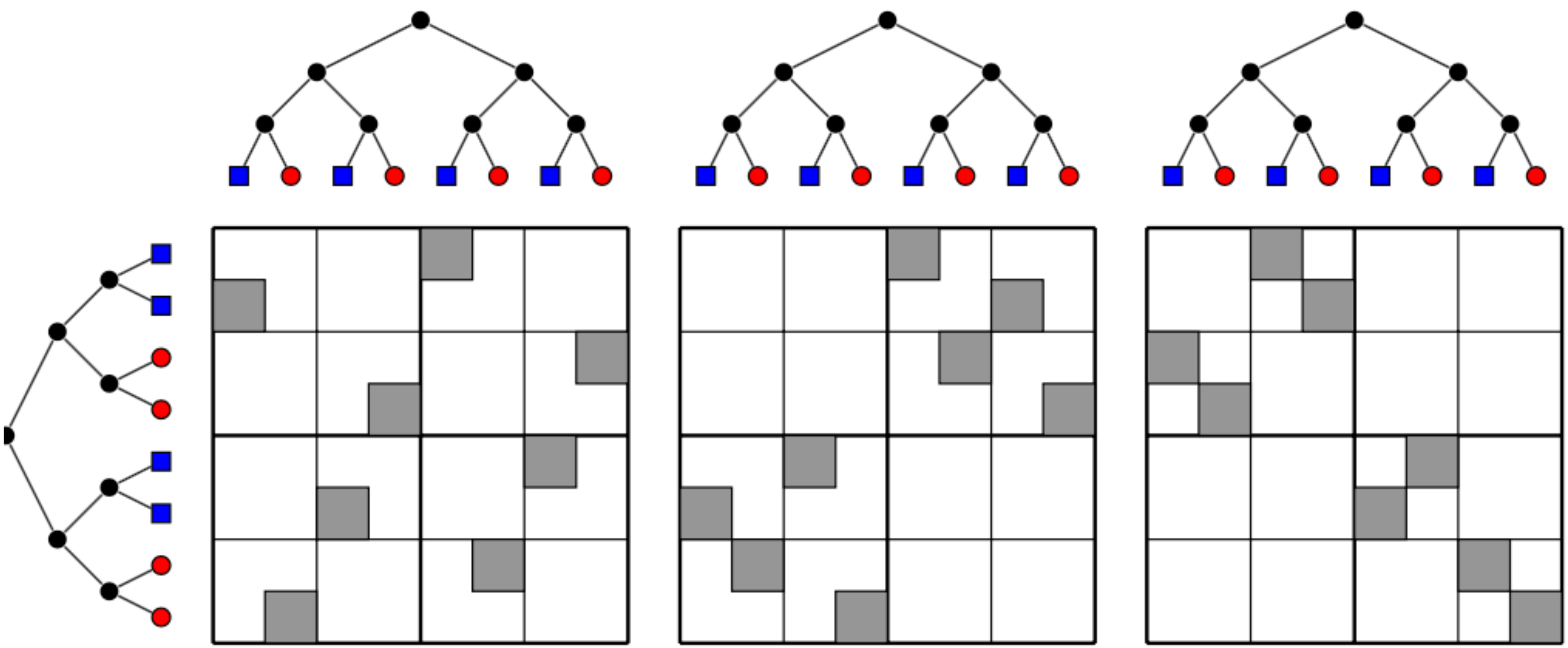


$GW = 0$

Optimal Transport for structured data

Fused Gromov-Wasserstein distance: example

Wasserstein distance (features only) | **FGW** | **Gromov-Wasserstein distance (structures only)**



$W = 0$ | $FGW > 0$ | $GW = 0$

Optimal Transport for structured data

Fused Gromov-Wasserstein distance

A distance w.r.t strong isomorphism

- $FGW \geq 0$ and satisfies the triangle inequality
- $\mathbf{C}_1, \mathbf{C}_2$ distances. $FGW = 0$ iff $\exists \sigma$ permutations of the nodes

$$\text{(conservation of the weights) } h_i = g_{\sigma(i)}$$
$$\text{(conservation of the features) } a_i = b_{\sigma(i)}$$
$$\text{(conservation of the structures) } C_1(i,k) = C_2(\sigma(i), \sigma(k))$$

Optimal Transport for structured data

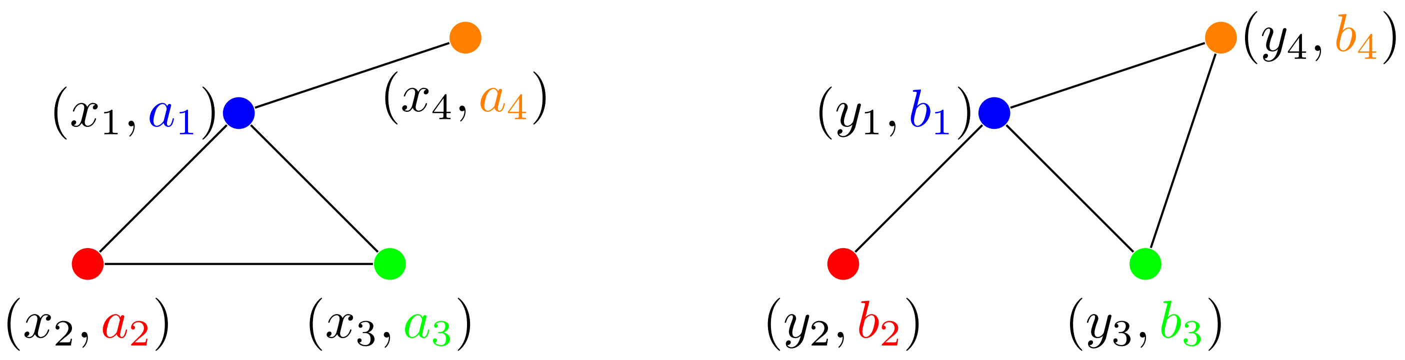
Fused Gromov-Wasserstein distance

A distance w.r.t strong isomorphism

- $FGW \geq 0$ and satisfies the triangle inequality
- $\mathbf{C}_1, \mathbf{C}_2$ distances. $FGW = 0$ iff $\exists \sigma$ permutations of the nodes

$$\text{(conservation of the weights) } h_i = g_{\sigma(i)}$$
$$\text{(conservation of the features) } a_i = b_{\sigma(i)}$$
$$\text{(conservation of the structures) } C_1(i,k) = C_2(\sigma(i), \sigma(k))$$

Same weights, same labels at the same place up to a permutation



Isometric + same features but not strongly isomorphic

Optimal Transport for structured data

Fused Gromov-Wasserstein distance

A distance w.r.t strong isomorphism

- $FGW \geq 0$ and satisfies the triangle inequality
- $\mathbf{C}_1, \mathbf{C}_2$ distances. $FGW = 0$ iff $\exists \sigma$ permutations of the nodes

$$\text{(conservation of the weights) } h_i = g_{\sigma(i)}$$
$$\text{(conservation of the features) } a_i = b_{\sigma(i)}$$
$$\text{(conservation of the structures) } C_1(i,k) = C_2(\sigma(i), \sigma(k))$$

Other properties

Interpolates GW between the structures and W between the features

Extends to the continuous setting: geodesic properties + sample complexity

Optimal Transport for structured data

Computing FGW (and GW!)

Solving FGW: a non convex QP

$$\min_{\boldsymbol{\pi}\in\Pi(\mathbf{h},\mathbf{g})} \sum_{i,j,k,l}(1-\alpha)d(a_i,b_j)^q+\alpha|C_1(i,k)-C_2(j,l)|^q\pi_{i,j}\pi_{k,l}$$

Quadratic function over polytope -> Conditional Gradient algorithm (a.k.a Frank-Wolfe)

Non convex but converges to a **local optimal solution** [Lacoste-Julien 2016]

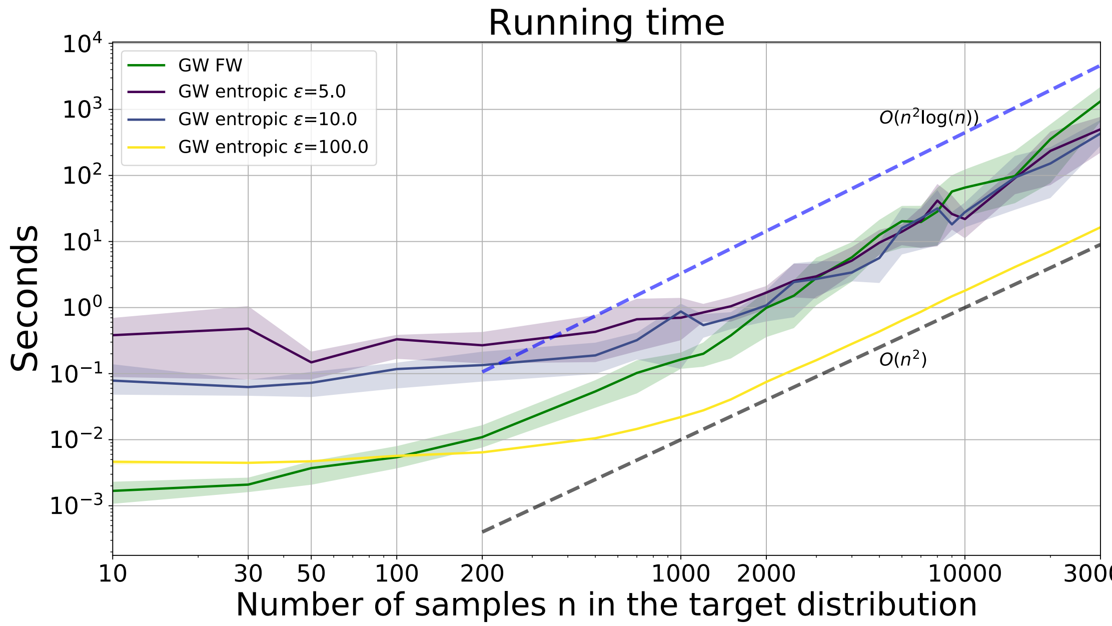
Find a **sparse** solution. FW gap = $O(\frac{1}{\sqrt{n_{iter}}})$

Algorithm 1 Conditional Gradient (CG) for FGW

1: $\pi^{(0)} \leftarrow \mathbf{h}\mathbf{g}^\top$
2: **for** $i = 1, \dots,$ **do**
3: $\quad \mathbf{G} \leftarrow$ Gradient from GW loss $w.r.t.$ $\boldsymbol{\pi}^{(i-1)}$
4: $\quad \tilde{\boldsymbol{\pi}}^{(i)} \leftarrow$ Solve OT with ground loss $\mathbf{G}$
5: $\quad \tau^{(i)} \leftarrow$ Line-search for GW loss with $\tau \in (0,1)$ (closed-form)
6: $\quad \boldsymbol{\pi}^{(i)} \leftarrow (1-\tau^{(i)})\boldsymbol{\pi}^{(i-1)} + \tau^{(i)}\tilde{\boldsymbol{\pi}}^{(i)}$
7: **end for**

Complexity

$$O(n_{iter}\ n^3)$$



FGW in action

Optimal Transport for structured data

FGW in action

Graph classification

A set of labeled graphs $(\mathcal{G}_i, y_i)$. Structure matrices shortest path

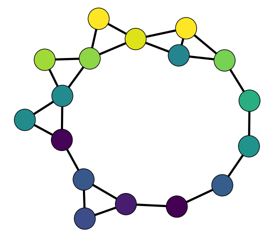
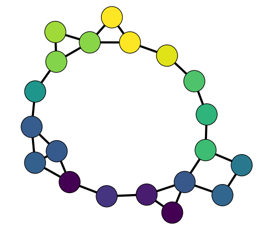
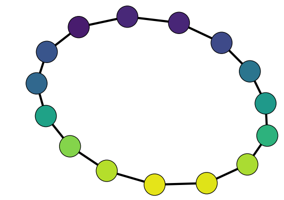
Linear classifier: SVM on the indefinite kernel $e^{-\frac{1}{\beta}FGW(\mathcal{G}_i,\mathcal{G}_j)}$

Compare with graph kernel approaches + GCN on benchmark datasets

	Labeled Graphs			Social Graphs	Vector attributes Graph		
Dataset	MUTAG	PTC	NCI1	IMDB-B	SYNTHETIC	PROTEIN	CUNEIFORM
WL	86.21±8.15	62.17±7.80	85.13±1.61	Unapplicable(U)	U	U	U
GK	82.42±8.40	56.46±8.03	60.78±2.48	56.00±3.61	41.13±4.68	U	U
RW	79.47±8.17	55.09±7.34	58.63±2.44	U	U	U	U
SP	85.79±2.51	58.53±2.55	73.00±0.51	55.80±2.93	38.93±5.12	U	U
HOPPER	U	U	U	U	90.67±4.67	71.96±3.22	32.59±8.73
PROPA	U	U	U	U	64.67±6.70	61.34±4.38	12.59± 6.67
PSCN $k = 10$	83.47±10.26	58.34±7.71	70.65±2.58	U	**100.00±0.00**	67.95±11.28	25.19±7.73
FGW	**88.42±5.67**	**65.31±7.90**	**86.42±1.63**	**63.80±3.49**	**100.00±0.00**	**74.55±2.74**	**76.67±7.04**

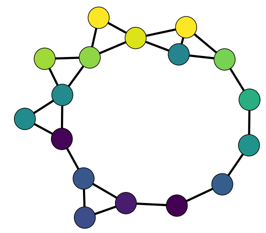
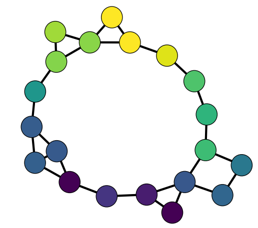
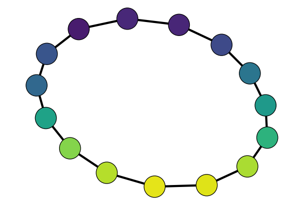
Optimal Transport for structured data

FGW barycenter

Making sense of: $\frac{1}{2}\Big($  $+$  $\Big) =$ 

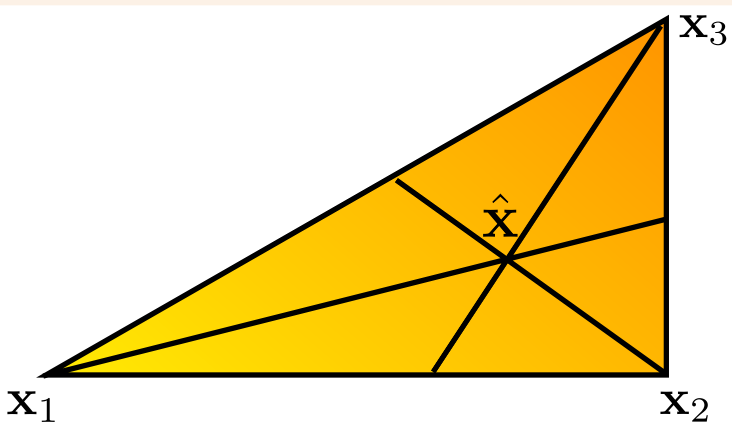
Optimal Transport for structured data

FGW barycenter

Making sense of: $\frac{1}{2}($  $+$  $) =$ 

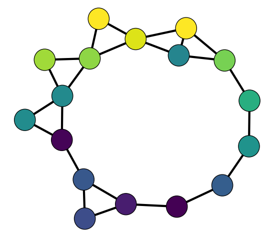
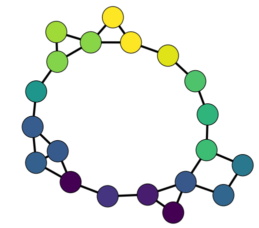
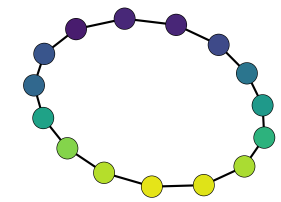
Euclidean Barycenter: $(\mathbb{R}^d, \|.\|_2)$

$$\inf_{\mathbf{x} \in \mathbb{R}^d} \sum_{i=1}^{n} \lambda_i \|\mathbf{x} - \mathbf{x}_i\|_2^2$$



Optimal Transport for structured data

FGW barycenter

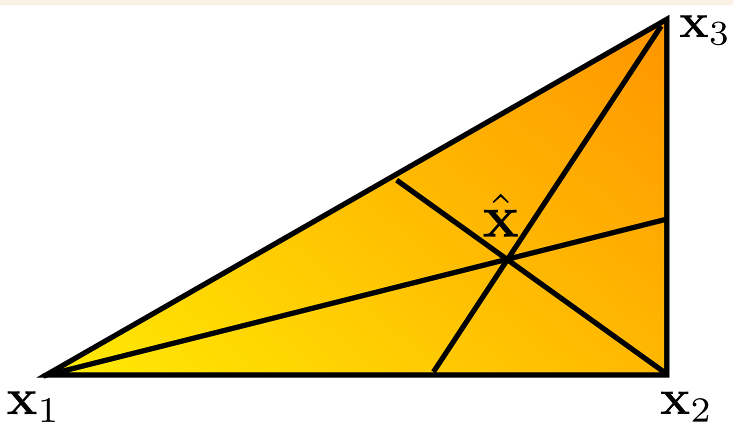
Making sense of: $\frac{1}{2}($  $+$  $) =$ 

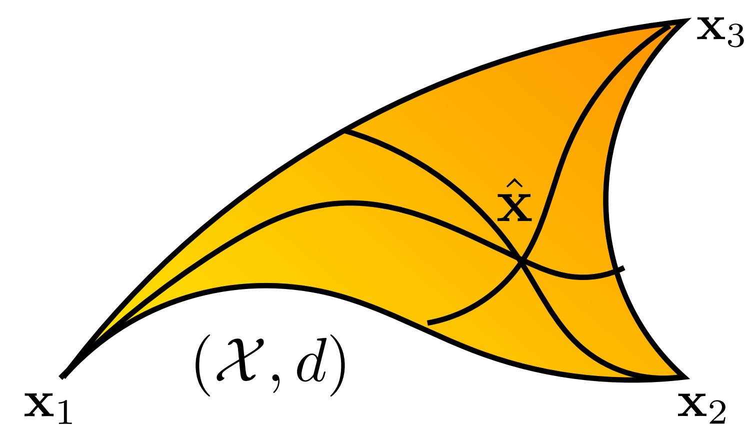
Euclidean Barycenter: $(\mathbb{R}^d, \|.\|_2)$

$$\inf_{\mathbf{x}\in\mathbb{R}^d} \sum_{i=1}^n \lambda_i \|\mathbf{x} - \mathbf{x}_i\|_2^2$$

Fréchet Barycenter: $(\mathcal{X}, d)$ metric space

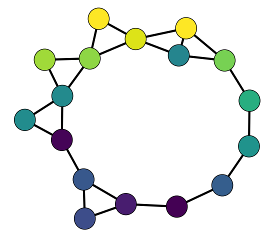
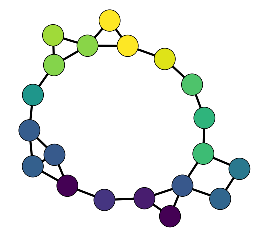
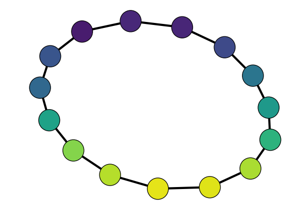
$$\inf_{x\in\mathcal{X}} \sum_{i=1}^n \lambda_i d(x, x_i)^p$$





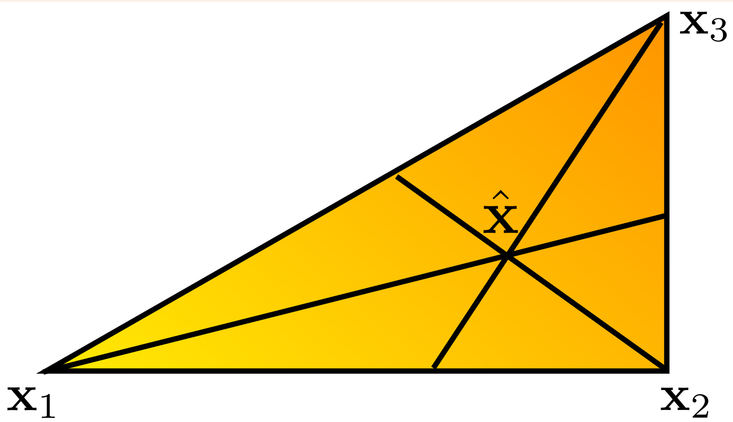
Optimal Transport for structured data

FGW barycenter

Making sense of: $\frac{1}{2}($  $+$  $) =$ 

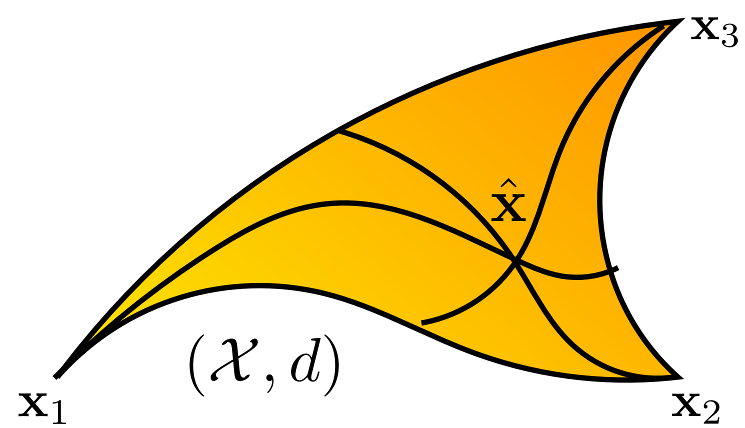
Euclidean Barycenter: $(\mathbb{R}^d, \|.\|_2)$

$$\inf_{\mathbf{x}\in\mathbb{R}^d} \sum_{i=1}^n \lambda_i \|\mathbf{x} - \mathbf{x}_i\|_2^2$$



Fréchet Barycenter: $(\mathcal{X}, d)$ metric space

$$\inf_{x\in\mathcal{X}} \sum_{i=1}^n \lambda_i d(x, x_i)^p$$



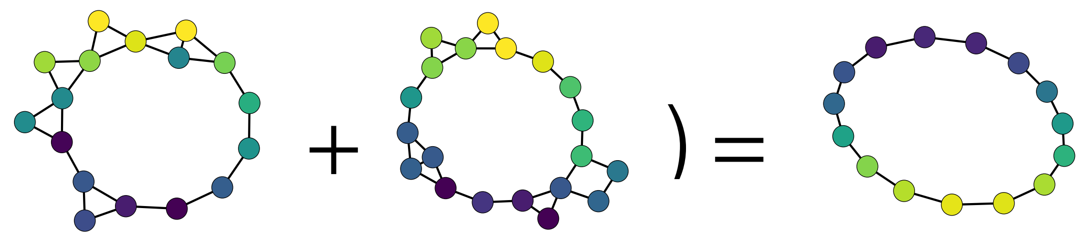
FGW barycenter

$$\min_{\mu} \sum_{k=1}^K \lambda_k FGW_{q,\alpha}(\mu, \mu_k)$$

- Barycenter of labeled graphs, relational data with attributes
- Consider feature space $\Omega = (\mathbb{R}^d, \|.\|_2^2)$ structured data $(\mathbf{C}_k, \mathbf{B}_k, \mathbf{h}_k)_{k=1}^K$

Optimal Transport for structured data

FGW barycenter

Making sense of: $\frac{1}{2}\Big($  $\Big) =$

FGW barycenter

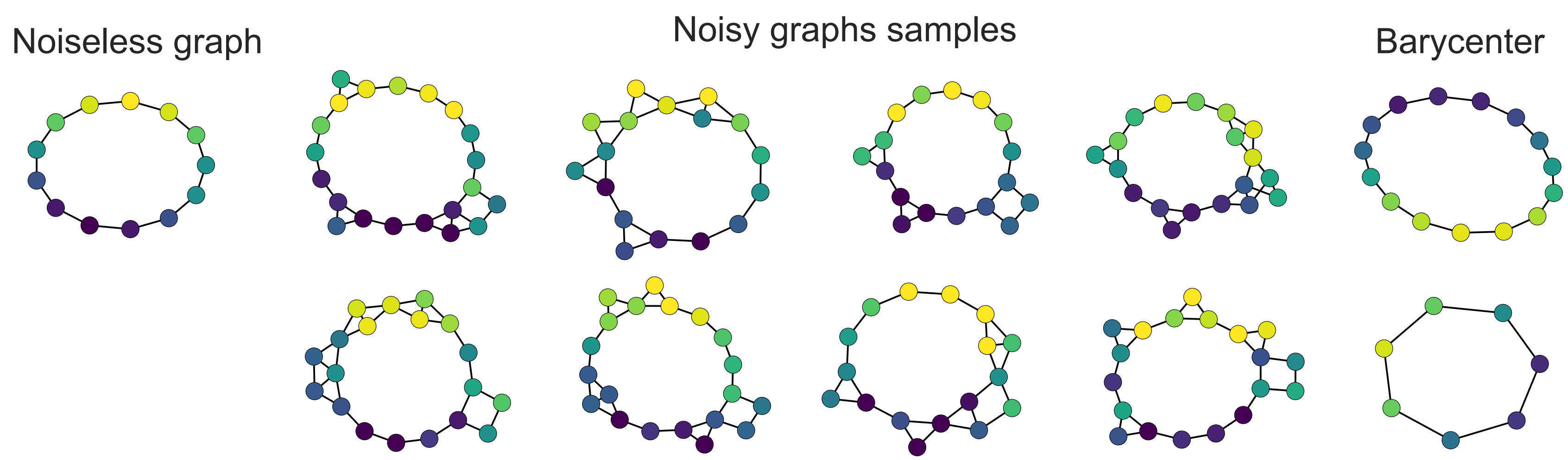
$$\min_{\mu} \sum_{k=1}^{K} \lambda_k FGW_{q,\alpha}(\mu, \mu_k)$$

Barycenter of labeled graphs, relational data with attributes

Consider feature space $\Omega = (\mathbb{R}^d, \|.\|_2^2)$ structured data $(\mathbf{C}_k, \mathbf{B}_k, \mathbf{h}_k)_{k=1}^{K}$

Algorithm 1 FGW barycenter

Initialize $\mathbf{C} \leftarrow \mathbf{C}_0, \mathbf{A} \leftarrow \mathbf{A}_0$.
while not converged **do**
  for $k = 1 \dots K$ **do**
    $\boldsymbol{\pi}_k \leftarrow FGW(\mathbf{M}_{\mathbf{AB}_k}, \mathbf{C}, \mathbf{C}_k, \mathbf{h}, \mathbf{h}_k)$
  end for
  $\mathbf{C} \leftarrow \frac{1}{\mathbf{h}\mathbf{h}^T} \sum_{k=1}^{K} \lambda_k \boldsymbol{\pi}_k^T \mathbf{C}_k \boldsymbol{\pi}_k$
  $\mathbf{A} \leftarrow \sum_{k=1}^{K} \lambda_k \mathbf{B}_k \boldsymbol{\pi}_k^T \mathrm{diag}(\frac{1}{\mathbf{h}})$
end while



Optimal Transport for structured data

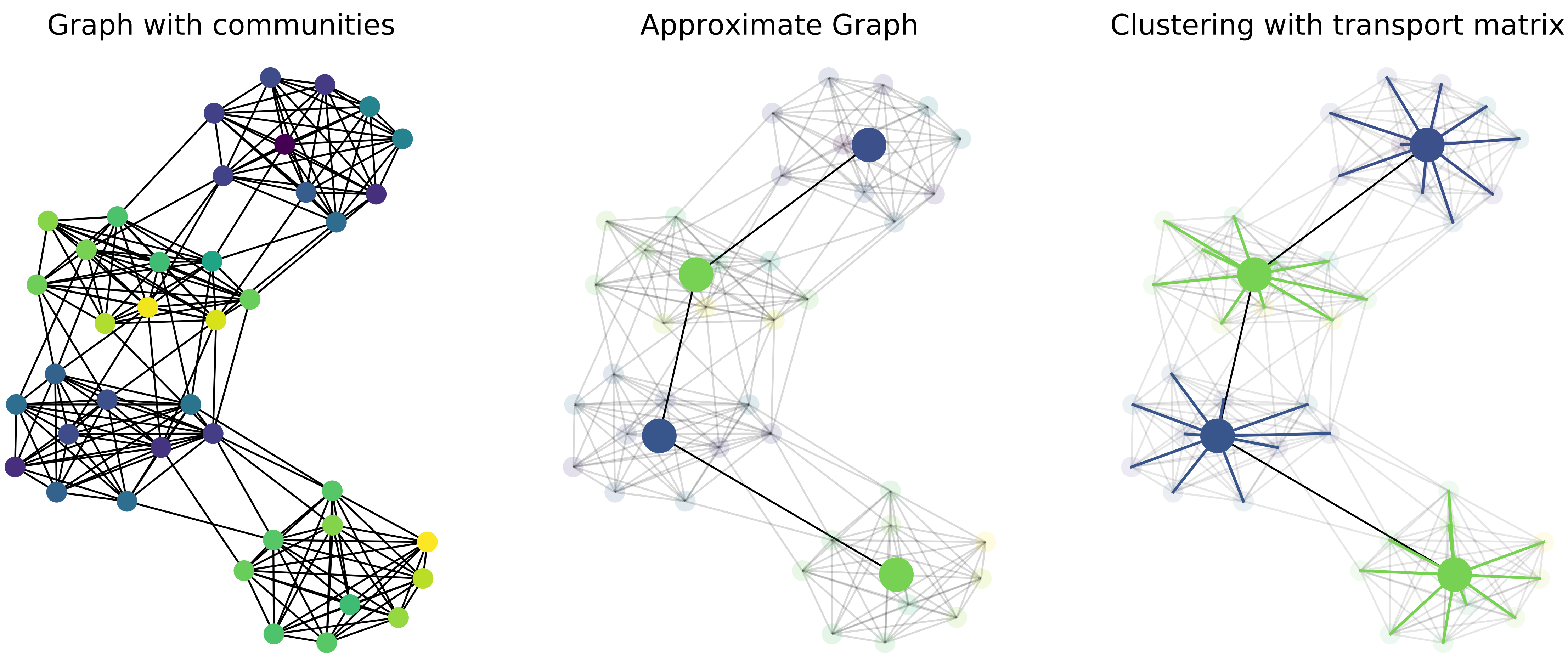
Summarization of graph

FGW coarsening

$$\min_{\mu} FGW(\mu, \nu) = \min_{\mathbf{A}, \mathbf{C}_1} FGW(\mathbf{M}_{\mathbf{AB}}, \mathbf{C}_1, \mathbf{C}_2, \mathbf{h}, \mathbf{g})$$

Given a labeled graph we look for the closest graph w.r.t FGW with fewer nodes

Projection w.r.t FGW -> barycenter problem with $K = 1$



Optimal Transport for structured data

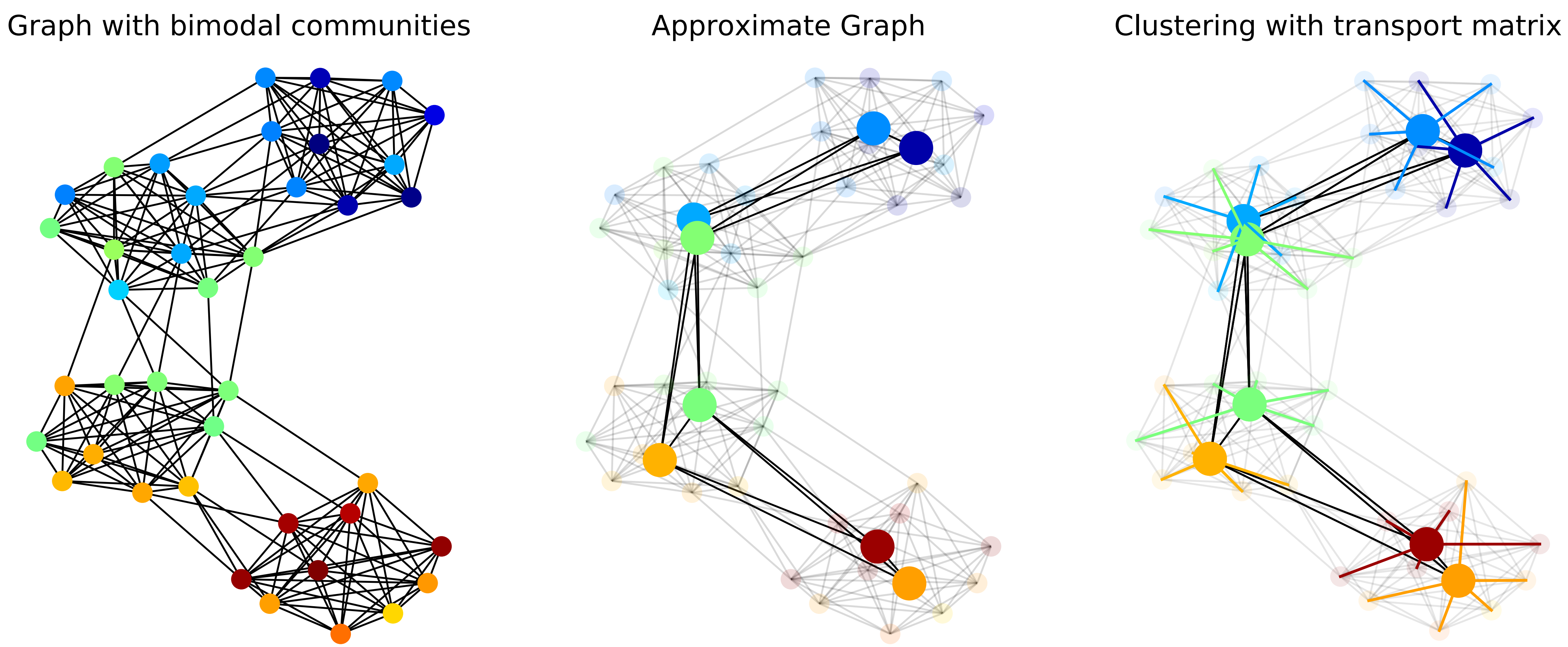
Summarization of graph

FGW coarsening

$$\min_{\mu} FGW(\mu, \nu) = \min_{\mathbf{A}, \mathbf{C}_1} FGW(\mathbf{M}_{\mathbf{AB}}, \mathbf{C}_1, \mathbf{C}_2, \mathbf{h}, \mathbf{g})$$

Given a labeled graph we look for the closest graph w.r.t FGW with fewer nodes

Projection w.r.t FGW -> barycenter problem with $K = 1$



Optimal Transport for structured data

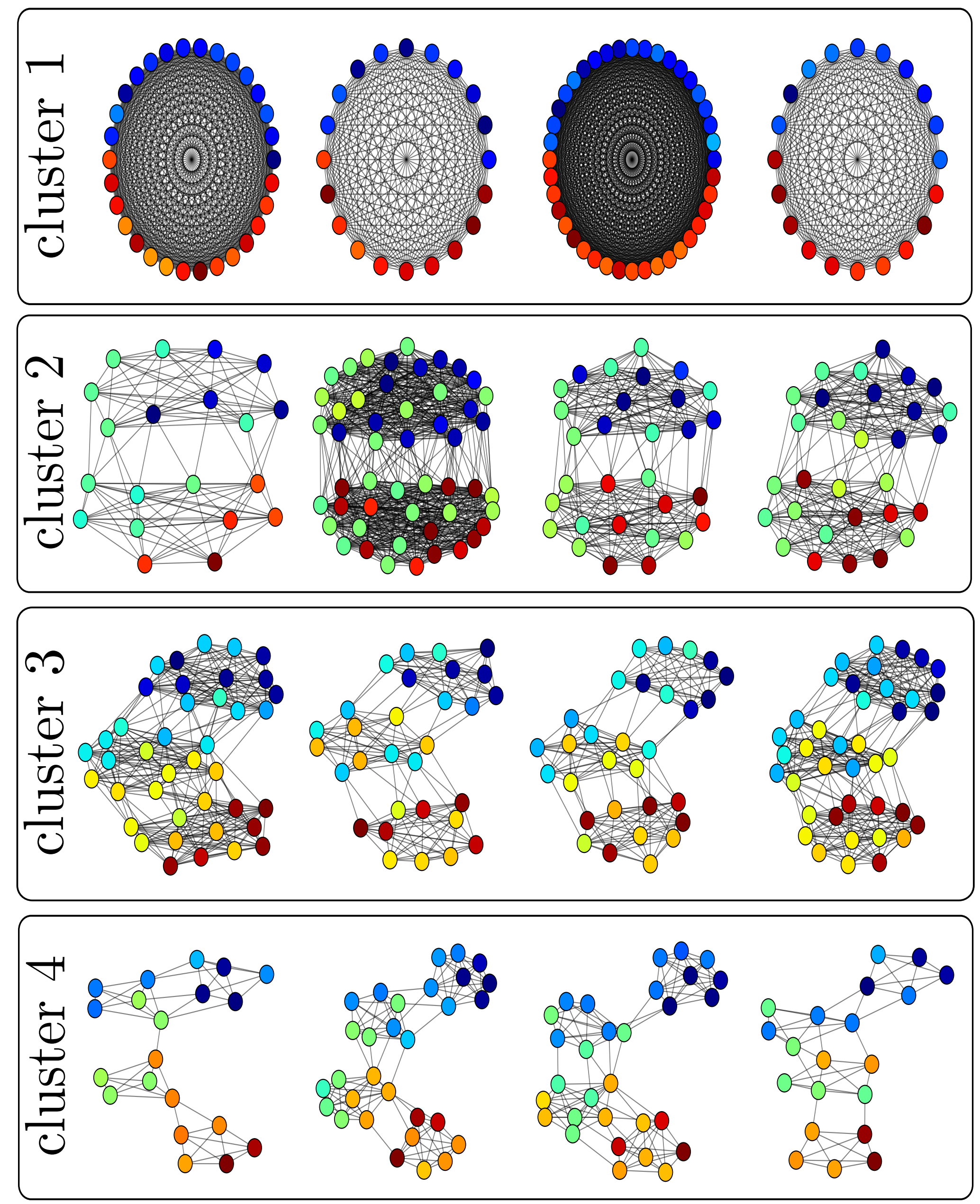
FGW clustering

Given a set of labeled graphs -> k-means using FGW barycenter

Algorithm 1 FGW clustering

Number of clusters K. Labeled graphs $(\mathbf{C}_i, \mathbf{B}_i, \mathbf{h}_i)_{i \in [\![N]\!]}$
Initialize centroids $\forall k \in [\![K]\!], \mathbf{C}_k \leftarrow \mathbf{C}_0, \mathbf{A}_k \leftarrow \mathbf{A}_0$.
while not converged **do**
    Calculate $N \times K$ FGW distances.
    for $i = 1 \dots N$ **do**
        Assign $(\mathbf{C}_i, \mathbf{B}_i, \mathbf{h}_i)$ to a cluster $k \in [\![K]\!]$
    end for
    for $k = 1 \dots K$ **do**
        $\mathbf{C}_k, \mathbf{A}_k \leftarrow$ `FGW barycenter`$((\mathbf{C}_i, \mathbf{B}_i, \mathbf{h}_i)_{i \in \text{cluster } k})$
    end for
end while

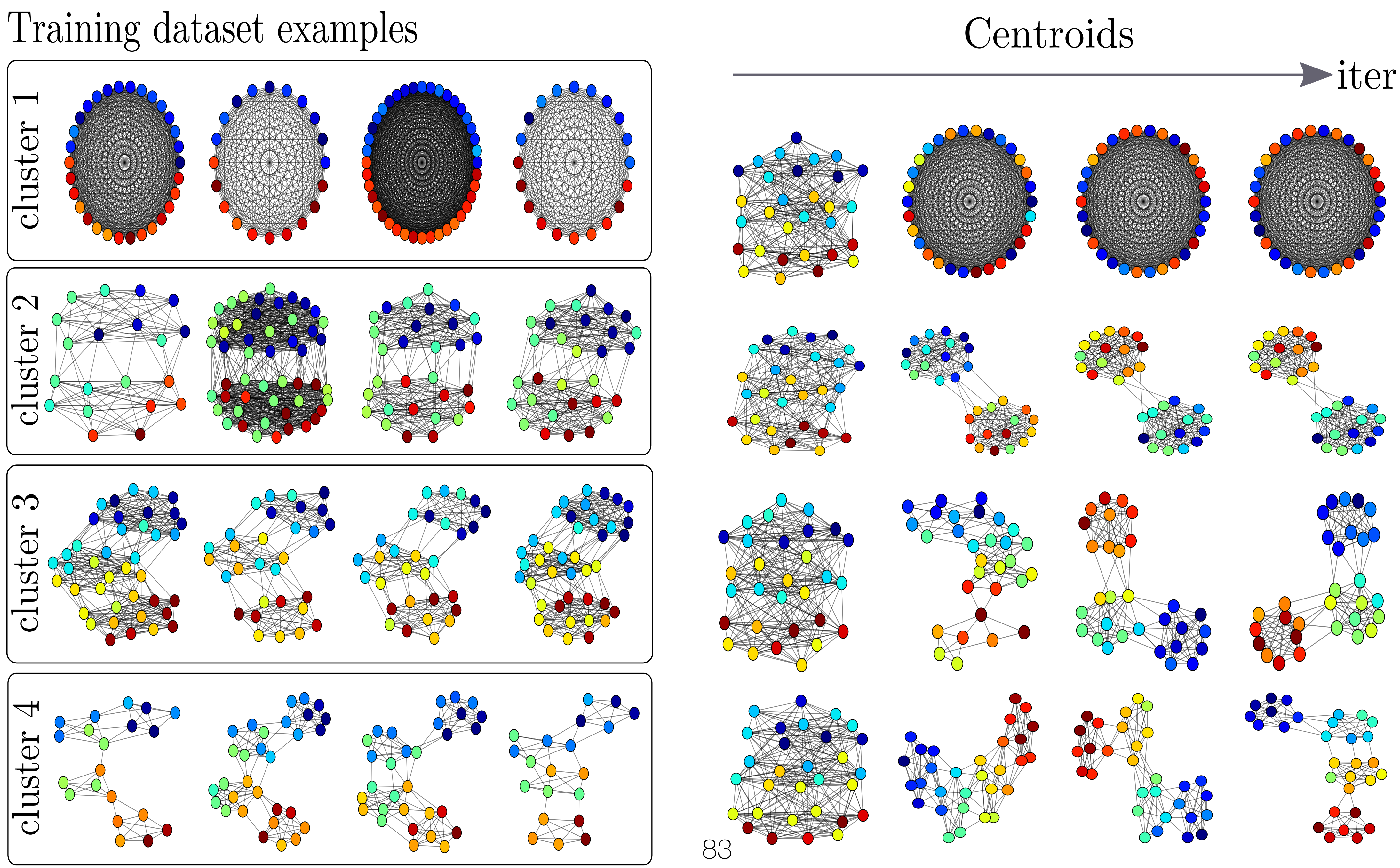
Training dataset examples



Optimal Transport for structured data

FGW clustering

Given a set of labeled graphs -> k-means using FGW barycenter



The POT library

Rémi Flamary, Nicolas Courty, Alexandre Gramfort, Mokhtar Z. Alaya, Aurélie Boisbunon, Stanislas Chambon, Laetitia Chapel, Adrien Corenflos, Kilian Fatras, Nemo Fournier, Léo Gautheron, Nathalie T.H. Gayraud, Hicham Janati, Alain Rakotomamonjy, Ievgen Redko, Antoine Rolet, Antony Schutz, Vivien Seguy, Danica J. Sutherland, Romain Tavenard, Alexander Tong, Titouan Vayer; 22(78):1−8, 2021.

Abstract

Optimal transport has recently been reintroduced to the machine learning community thanks in part to novel efficient optimization procedures allowing for medium to large scale applications. We propose a Python toolbox that implements several key optimal transport ideas for the machine learning community. The toolbox contains implementations of a number of founding works of OT for machine learning such as Sinkhorn algorithm and Wasserstein barycenters, but also provides generic solvers that can be used for conducting novel fundamental research. This toolbox, named POT for Python Optimal Transport, is open source with an MIT license.

Python library on Optimal Transport

- OT LP solver, Sinkhorn
- Barycenters, Domain adaptation
- Gromov, FGW, graphs OT…

Url: https://github.com/PythonOT/POT