

Optimal transport for graph data

Barycenters and dictionary learning

R. Flamary - CMAP, École Polytechnique, Institut Polytechnique de Paris

March 31 2022

London School of Economics

Collaborators



N. Courty



A. Rakotomamonjy



D. Tuia



A. Habrard



M. Perrot



M. Ducoffe



M. Cuturi



K. Lounici



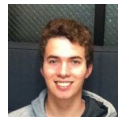
A. Ferrari



C. Févotte



V. Emiya



V. Seguy



B. Damodaran



T. Vayer



L. Chapel



R. Tavenard



K. Fatras



I. Redko



H. Janati



T. Séjourné



H. Tran



G. Gasso



M. Corneli



C. Vincent-Cuaz

Optimal Transport and divergences between graphs

Discrete Optimal Transport (OT)

Gromov-Wasserstein divergence and applications on graphs

Fused Gromov-Wasserstein and applications on attributed graphs

Online Graph Dictionary Learning

Linear modeling and unmixing of graphs

Learning a dictionary of graphs

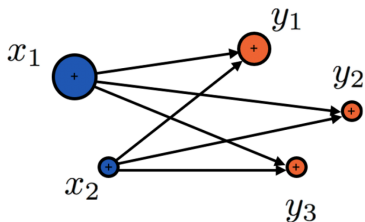
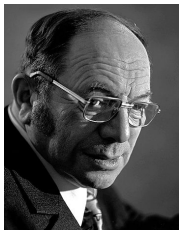
Numerical experiments

Semi-relaxed Gromov Wasserstein distance

Semi-relaxed GW problem and solver

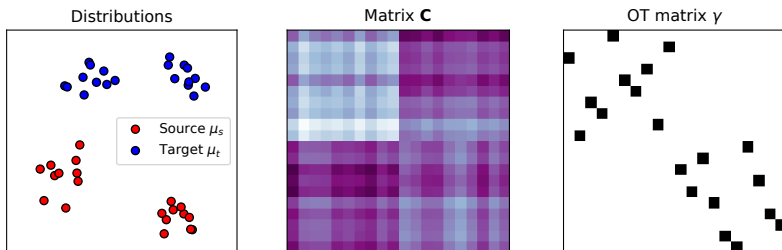
Numerical experiments with srGW

Optimal Transport and divergences between graphs



- Problem introduced by Gaspard Monge in his memoire [Monge, 1781].
- How to move mass while minimizing a cost (mass + cost)
- Monge formulation seeks for a mapping between two mass distribution.
- Reformulated by Leonid Kantorovich (1912–1986), Economy nobelist in 1975
- Focus on where the mass goes, allow splitting [Kantorovich, 1942].
- Applications originally for resource allocation problems

Optimal transport between discrete distributions



Kantorovitch formulation : OT Linear Program

When $\mu_s = \sum_{i=1}^{n_s} a_i \delta_{\mathbf{x}_i^s}$ and $\mu_t = \sum_{i=1}^{n_t} b_i \delta_{\mathbf{x}_i^t}$

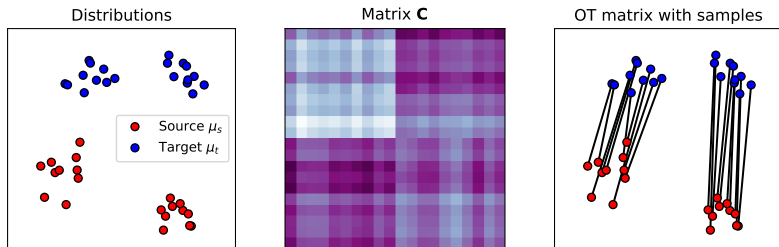
$$W_p^p(\mu_s, \mu_t) = \min_{\mathbf{T} \in \Pi(\mu_s, \mu_t)} \left\{ \langle \mathbf{T}, \mathbf{C} \rangle_F = \sum_{i,j} T_{i,j} c_{i,j} \right\}$$

where \mathbf{C} is a cost matrix with $c_{i,j} = c(\mathbf{x}_i^s, \mathbf{x}_j^t) = \|\mathbf{x}_i^s - \mathbf{x}_j^t\|^p$ and the constraints are

$$\Pi(\mu_s, \mu_t) = \left\{ \mathbf{T} \in (\mathbb{R}^+)^{n_s \times n_t} \mid \mathbf{T} \mathbf{1}_{n_t} = \mathbf{a}, \mathbf{T}^T \mathbf{1}_{n_s} = \mathbf{b} \right\}$$

- $W_p(\mu_s, \mu_t)$ is called the Wasserstein distance (EMD for $p = 1$).
- Entropic regularization solved efficiently with Sinkhorn [Cuturi, 2013b].
- Classical OT needs distributions lying in the same space \rightarrow Gromov-Wasserstein.

Optimal transport between discrete distributions



Kantorovitch formulation : OT Linear Program

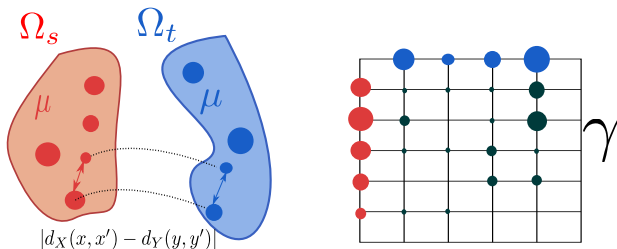
When $\mu_s = \sum_{i=1}^{n_s} a_i \delta_{\mathbf{x}_i^s}$ and $\mu_t = \sum_{i=1}^{n_t} b_i \delta_{\mathbf{x}_i^t}$

$$W_p^p(\mu_s, \mu_t) = \min_{\mathbf{T} \in \Pi(\mu_s, \mu_t)} \left\{ \langle \mathbf{T}, \mathbf{C} \rangle_F = \sum_{i,j} T_{i,j} c_{i,j} \right\}$$

where \mathbf{C} is a cost matrix with $c_{i,j} = c(\mathbf{x}_i^s, \mathbf{x}_j^t) = \|\mathbf{x}_i^s - \mathbf{x}_j^t\|^p$ and the constraints are

$$\Pi(\mu_s, \mu_t) = \left\{ \mathbf{T} \in (\mathbb{R}^+)^{n_s \times n_t} \mid \mathbf{T} \mathbf{1}_{n_t} = \mathbf{a}, \mathbf{T}^T \mathbf{1}_{n_s} = \mathbf{b} \right\}$$

- $W_p(\mu_s, \mu_t)$ is called the Wasserstein distance (EMD for $p = 1$).
- Entropic regularization solved efficiently with Sinkhorn [Cuturi, 2013b].
- Classical OT needs distributions lying in the same space \rightarrow Gromov-Wasserstein.



Inspired from Gabriel Peyré

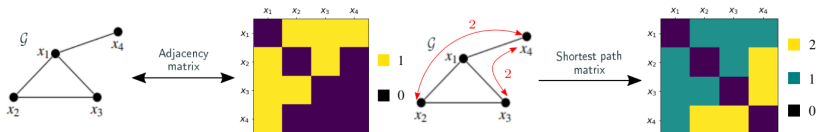
GW for discrete distributions [Memoli, 2011]

$$\mathcal{GW}_p(\mu_s, \mu_t) = \left(\min_{T \in \Pi(\mu_s, \mu_t)} \sum_{i,j,k,l} |D_{i,k} - D'_{j,l}|^p T_{i,j} T_{k,l} \right)^{\frac{1}{p}}$$

with $\mu_s = \sum_i a_i \delta_{\mathbf{x}_i^s}$ and $\mu_t = \sum_j b_j \delta_{\mathbf{x}_j^t}$ and $D_{i,k} = \|\mathbf{x}_i^s - \mathbf{x}_k^s\|$, $D'_{j,l} = \|\mathbf{x}_j^t - \mathbf{x}_l^t\|$

- Distance between metric measured spaces : across different spaces.
- Search for an OT plan that preserve the pairwise relationships between samples.
- Invariant to isometry in either spaces (e.g. rotations and translation).
- Entropy regularize GW proposed in [Peyré et al., 2016].

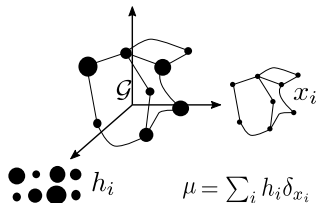
Gromov-Wasserstein between graphs



Model the graph structure

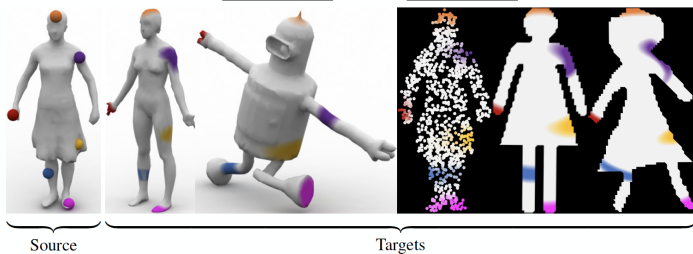
- A graph \mathcal{G} : node set $\{x_i\}_{i \in [N]}$ (implicit) & edge set $\{(x_i, x_j) | x_i \rightarrow x_j\}$.
- Encoded as a node relationship matrix D e.g. adjacency (task-driven choice).

OT context: Graph as a distribution

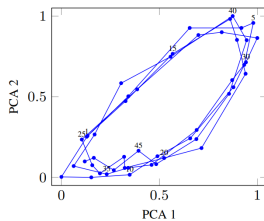
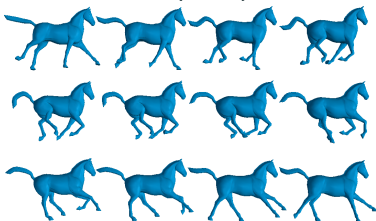


- \mathcal{G} modeled as a discrete distribution $\mu = \sum_i h_i \delta_{x_i}$ summarized by (D, h) .
- D : node relationship matrix.
- h : vector of probability masses specifying node relative importance (uniform by default).

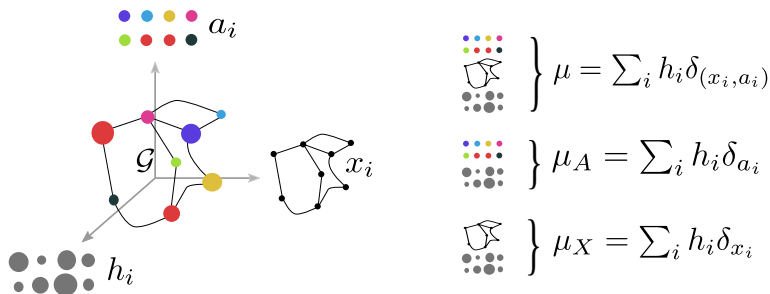
Shape matching between 3D and 2D surfaces



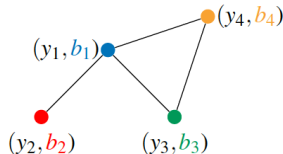
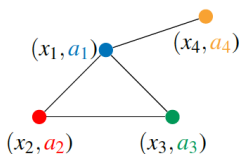
Multidimensional scaling (MDS) of shape collection



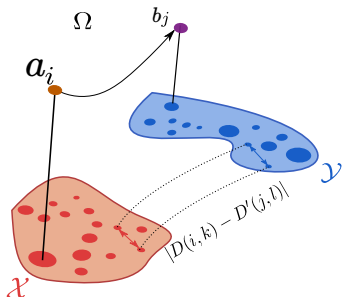
Attributed graphs as distributions



- Joint distribution μ in the feature/structure space.
 - Nodes are weighted by their mass h_i .
 - Structure encoded by x_i (no common metric between two different graphs).
 - Features values a_i can be compared through the common metric.
- Importance of the joint modeling:



Fused Gromov-Wasserstein distance



Fused Gromov Wasserstein distance [Vayer et al., 2020]

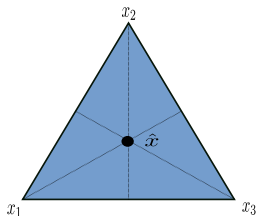
$$\mu_s = \sum_{i=1}^n h_i \delta_{x_i, a_i} \text{ and } \mu_t = \sum_{j=1}^m g_j \delta_{y_j, b_j}$$

$$\mathcal{FGW}_{p,q,\alpha}(D, D', \mu_s, \mu_t) = \left(\min_{T \in \Pi(\mu_s, \mu_t)} \sum_{i,j,k,l} ((1-\alpha)C_{i,j}^q + \alpha |D_{i,k} - D'_{j,l}|^q)^p T_{i,j} T_{k,l} \right)^{\frac{1}{p}}$$

with $D_{i,k} = \|x_i - x_k\|$ and $D'_{j,l} = \|y_j - y_l\|$ and $C_{i,j} = \|a_i - b_j\|$

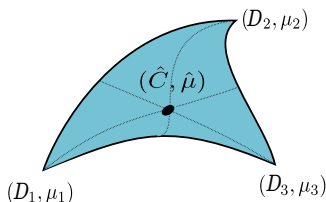
- Parameters $q > 1, \forall p \geq 1$.
- $\alpha \in [0, 1]$ is a trade off parameter between structure and features.

Euclidean barycenter



$$\min_x \sum_k \lambda_k \|x - x_k\|^2$$

FGW barycenter



$$\min_{D \in \mathbb{R}^{n \times n}, \mu} \sum_i \lambda_i \mathcal{FGW}(D_i, D, \mu_i, \mu)$$

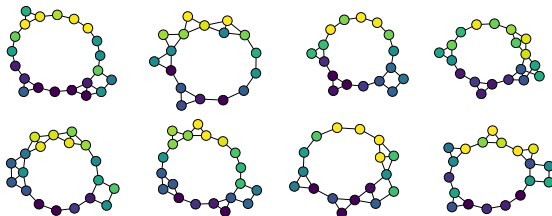
FGW barycenter $p = 1, q = 2$

- Estimate FGW barycenter using Frechet means (similar to [Peyré et al., 2016]).
- Barycenter optimization solved via block coordinate descent (on $\mathbf{T}, D, \{a_i\}_i$).
- Can chose to fix the structure (D) or the features $\{a_i\}_i$ in the barycenter.
- a_{ii} , and D updates are weighted averages using \mathbf{T} .

Noiseless graph



Noisy graphs samples



Barycenter of noisy graphs

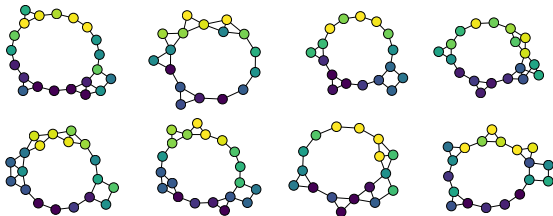
- We select a clean graph, change the number of nodes and add label noise and random connections.
- We compute the barycenter on $n = 15$ and $n = 7$ nodes.
- Barycenter graph is obtained through thresholding of the D matrix.

FGW barycenter on labeled graphs

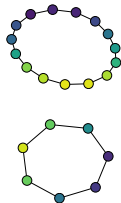
Noiseless graph



Noisy graphs samples



Barycenter



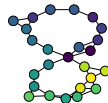
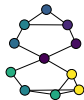
Barycenter of noisy graphs

- We select a clean graph, change the number of nodes and add label noise and random connections.
- We compute the barycenter on $n = 15$ and $n = 7$ nodes.
- Barycenter graph is obtained through thresholding of the D matrix.

Noiseless graph



Noisy graphs samples



Barycenter of noisy graphs

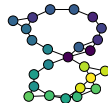
- We select a clean graph, change the number of nodes and add label noise and random connections.
- We compute the barycenter on $n = 15$ and $n = 7$ nodes.
- Barycenter graph is obtained through thresholding of the D matrix.

FGW barycenter on labeled graphs

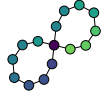
Noiseless graph



Noisy graphs samples



Barycenter

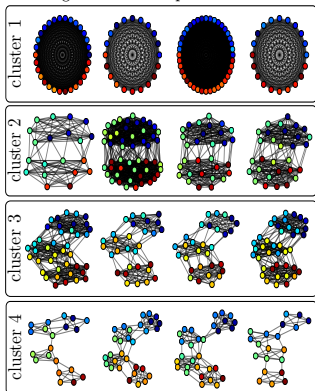


Barycenter of noisy graphs

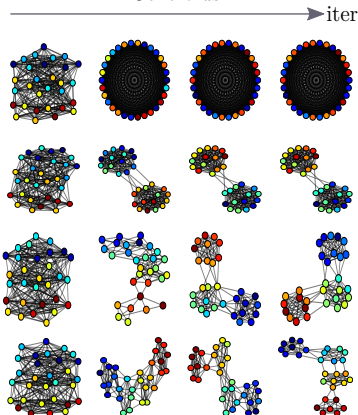
- We select a clean graph, change the number of nodes and add label noise and random connections.
- We compute the barycenter on $n = 15$ and $n = 7$ nodes.
- Barycenter graph is obtained through thresholding of the D matrix.

FGW for graphs based clustering

Training dataset examples



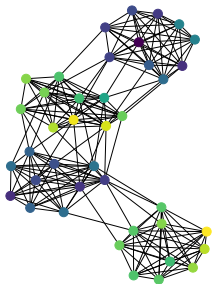
Centroids



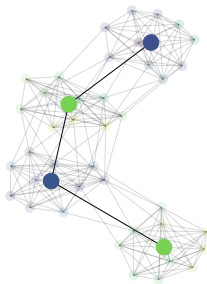
- Clustering of multiple real-valued graphs. Dataset composed of 40 graphs (10 graphs \times 4 types of communities)
- k -means clustering using the FGW barycenter

FGW barycenter for community clustering

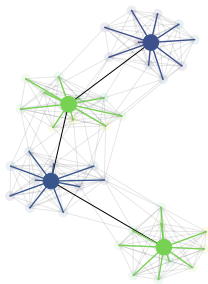
Graph with communities



Approximate Graph



Clustering with transport matrix

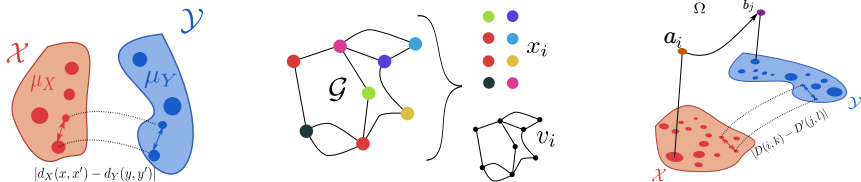


Graph approximation and community clustering

$$\min_{\mathbf{D}, \mu} \mathcal{FGW}(\mathbf{D}, \mathbf{D}_0, \mu, \mu_0)$$

- Approximate the graph (\mathbf{D}_0, μ_0) with a small number of nodes.
- Can be seen as a FGW (compressed) barycenter for one graph.
- OT matrix give the clustering affectation.
- Works for single and multiple modes in the clusters.

GW and FGW for graph modeling



Gromov-Wasserstein distance [Memoli, 2011]

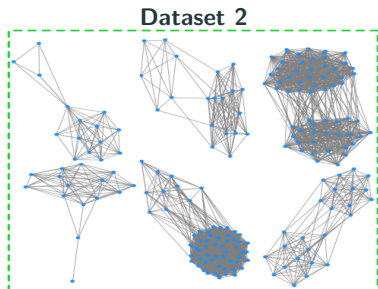
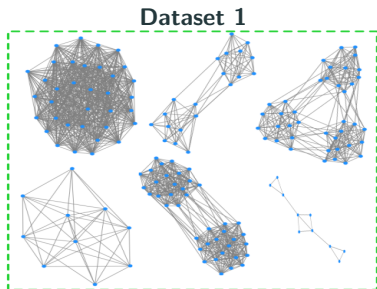
- Divergence between distributions across metric spaces.
- Can be used to measure similarity between graphs seen as distribution their pairwise node relationship.

Fused Gromov-Wasserstein distance [Vayer et al., 2018]

- Model labeled structured data as joint structure/labels distributions.
 - New versatile method for comparing structured data based on Optimal Transport
 - New notion of barycenter of structured data such as graphs or time series
1. How to use GW/FGW to model data variability in a dataset of graphs?
 2. How to handle the sensitivity to the weights (when no weights are provided) ?

Online Graph Dictionary Learning

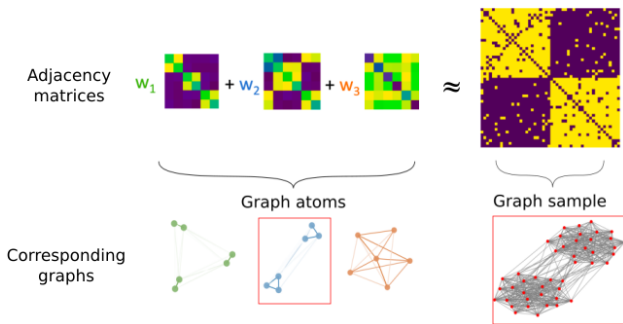
Datasets of graphs



SBM with balanced communities $\{1, 2, 3\}$.

Two communities of variable proportions.

- We have access to **large datasets of graphs** with variable number of nodes.
- How to model the variability of those graphs?
- A natural formulation is to use **factorization**.
- We propose to use a **linear** model for representing the graph associated to and estimation of the linear basis : **Dictionary learning**.

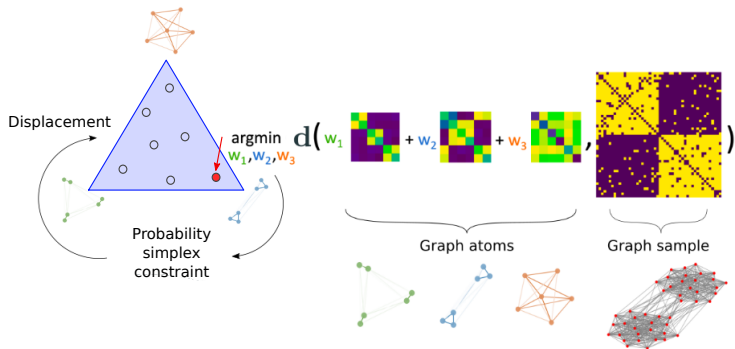


Linear modeling of graphs

$$D \approx \sum_{s \in [S]} w_s \overline{D}_s \quad (1)$$

- Approximate a given graph structure D as a non-negative weighted sum of template graphs \overline{D}_s .
- $w \in \Sigma_S$ are the weights in the simplex.
- $\{\overline{D}_s\}_s$ is the dictionary of templates that all have the same order (nb. of nodes).

Gromov-Wasserstein Linear unmixing

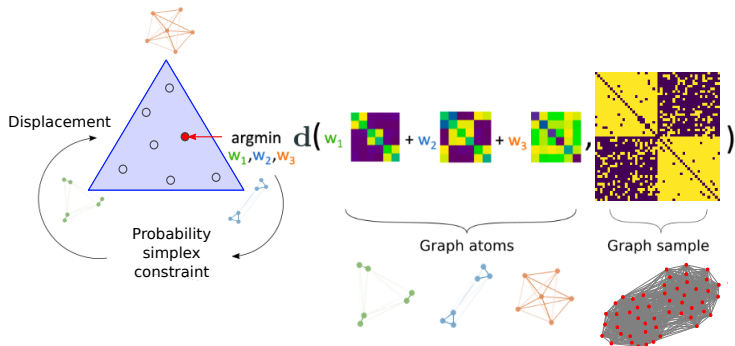


Sparse linear unmixing with Gromov-Wasserstein [Vincent-Cuaz et al., 2021]

$$\min_{\mathbf{w} \in \Sigma_S} \mathcal{GW}_2^2 \left(\sum_{s \in [S]} w_s \overline{D}_s, D \right) \quad (2)$$

- Estimate the linear (vector) representation on the simplex \mathbf{w} minimizing the GW distance *w.r.t.* the target graph D (non-negative unmixing).
- \mathbf{w} is a vector embedding of the graph D in the dictionary.

Gromov-Wasserstein Linear unmixing

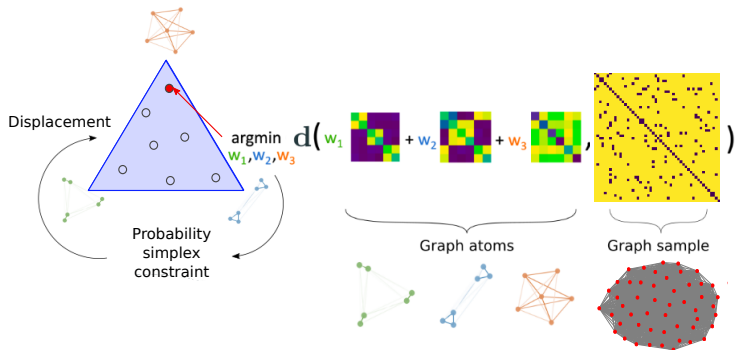


Sparse linear unmixing with Gromov-Wasserstein [Vincent-Cuaz et al., 2021]

$$\min_{\mathbf{w} \in \Sigma_S} \mathcal{GW}_2^2 \left(\sum_{s \in [S]} w_s \overline{D}_s, D \right) \quad (2)$$

- Estimate the linear (vector) representation on the simplex \mathbf{w} minimizing the GW distance *w.r.t.* the target graph D (non-negative unmixing).
- \mathbf{w} is a vector embedding of the graph D in the dictionary.

Gromov-Wasserstein Linear unmixing



Sparse linear unmixing with Gromov-Wasserstein [Vincent-Cuaz et al., 2021]

$$\min_{\mathbf{w} \in \Sigma_S} \mathcal{GW}_2^2 \left(\sum_{s \in [S]} w_s \overline{D}_s, D \right) \quad (2)$$

- Estimate the linear (vector) representation on the simplex \mathbf{w} minimizing the GW distance *w.r.t.* the target graph D (non-negative unmixing).
- \mathbf{w} is a vector embedding of the graph D in the dictionary.

GDL optimization problem

$$\min_{\{\mathbf{w}^{(k)}\}_{k \in [K]}, \{\overline{\mathbf{D}}_s\}_{s \in [S]}} \sum_{k=1}^K \mathcal{GW}_2^2 \left(\mathbf{D}^{(k)}, \sum_{s \in [S]} w_s^{(k)} \overline{\mathbf{D}}_s \right) - \lambda \|\mathbf{w}^{(k)}\|_2^2 \quad (3)$$

- On a dataset of K undirected graphs $\{\mathbf{D}^{(k)} \in S_{N^{(k)}}(\mathbb{R})\}_{k \in [K]}$.
- We want to estimate simultaneously the unmixing $\mathbf{w}^{(k)}$ of each graphs and the optimal dictionary $\{\overline{\mathbf{D}}_s\}_{s \in [S]}$.
- Very similar to classical DL (Non-negative Matrix Factorization) approach but with GW as a data fitting term.
- We propose to solve it an adaptation of the online algorithm [Mairal et al., 2009]

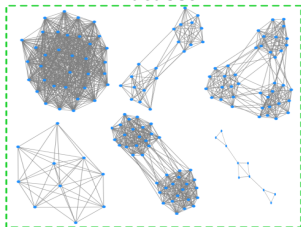
Stochastic/Online update [Vincent-Cuaz et al., 2021]

- 1: Sample a minibatch of graphs $\mathcal{B} := \{\mathbf{D}^{(k)}\}_{k \in \mathcal{B}}$.
- 2: Compute $\{(\mathbf{w}^{(k)}, \mathbf{T}^{(k)})\}_{k \in [B]}$ from solving B independent unmixings.
- 3: Compute the gradient $\tilde{\nabla}_{\overline{\mathbf{D}}_s}$ on the minibatch with fixed $\{(\mathbf{w}^{(k)}, \mathbf{T}^{(k)})\}_{k \in [B]}$.
- 4: Projected gradient step, $\forall s \in [S], \overline{\mathbf{D}}_s \leftarrow Proj_{S_{N^{(k)}}(\mathbb{R})}(\overline{\mathbf{D}}_s - \eta_C \tilde{\nabla}_{\overline{\mathbf{D}}_s})$

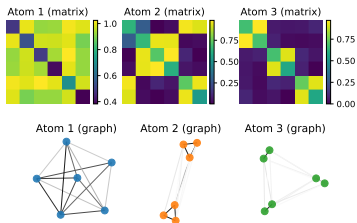
Experiments - Unsupervised representation learning

- Stochastic block model with $\{1, 2, 3\}$ blocks

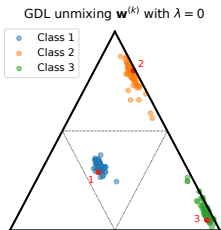
Dataset



Learned atoms



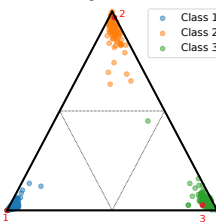
Embedding space



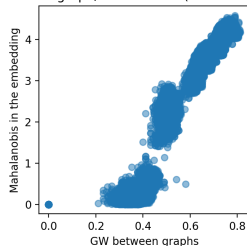
Examples



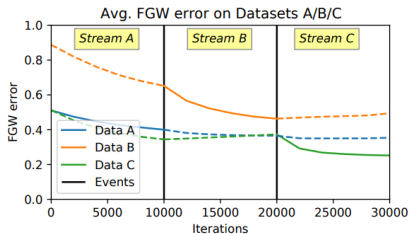
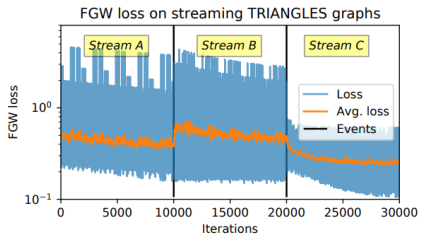
GDL unmixing $\mathbf{w}^{(k)}$ with $\lambda = 0.001$



GW graph/Mahalanobis (corr=0.96)



- **Streaming graphs:** Stochastic update for each new incoming graph
- Dataset : **TRIANGLES**
 - 30.000+ labeled graphs
 - 10 classes
- **Simulated stream:** data A (4 classes) → data B (3 classes) → data C (3 classes)



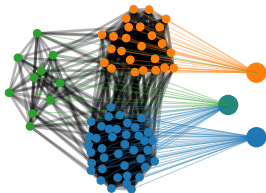
Semi-relaxed Gromov Wasserstein distance

Nodes weights are important

Uniform weights graph partitioning with GW

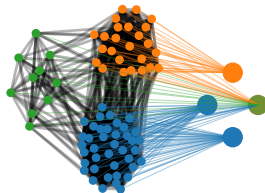
$$\text{GW}(\mathbf{C}, \mathbf{h}, \mathbf{I}_3, \bar{\mathbf{h}}) = 0.235$$

(ami=0.66)



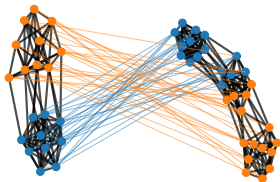
$$\text{GW}(\mathbf{C}, \mathbf{h}, \mathbf{I}_4, \bar{\mathbf{h}}) = 0.274$$

(ami=0.54)



All mass needs to be transported: sub-structures are lost

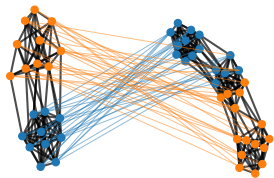
$$\text{GW}(\mathbf{C}, \mathbf{h}, \bar{\mathbf{C}}, \bar{\mathbf{h}}) = 0.219$$



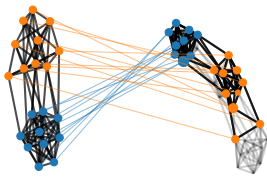
Relax the weights (half of them)!

Semi-relaxed Gromov-Wasserstein divergence

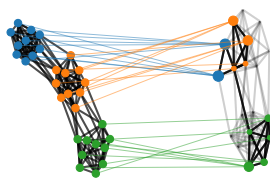
$$\text{GW}(\mathbf{C}, \mathbf{h}, \bar{\mathbf{C}}, \bar{\mathbf{h}}) = 0.219$$



$$\text{srGW}(\mathbf{C}, \mathbf{h}, \bar{\mathbf{C}}) = 0.05$$



$$\text{srGW}(\bar{\mathbf{C}}, \bar{\mathbf{h}}, \mathbf{C}) = 0.113$$



Semi-relaxed GW divergence [Vincent-Cuaz et al., 2022]:

$$\text{srGW}_2^2(\mathbf{D}, \mathbf{h}, \bar{\mathbf{D}}) := \min_{\bar{\mathbf{h}} \in \Sigma_{\bar{N}}} \text{GW}_2^2(\mathbf{D}, \mathbf{h}, \bar{\mathbf{D}}, \bar{\mathbf{h}})$$

- Match \mathcal{G} and $\bar{\mathcal{G}}$ while reweighing nodes of $\bar{\mathcal{G}}$ so that the formed graph $(\bar{\mathbf{D}}, \bar{\mathbf{h}})$ is at minimal GW distance from \mathcal{G} .
- **Equivalent problem easier to solve:**

$$\text{srGW}_2^2(\mathbf{D}, \mathbf{h}, \bar{\mathbf{D}}) = \min_{T \mathbf{1}_{\bar{N}} = \mathbf{h}} \sum_{ijkl} (C_{ij} - \bar{C}_{kl})^2 T_{ik} T_{jl} \quad \text{with} \quad T \in \mathbb{R}_+^{N \times \bar{N}'}$$

- second marginal of T is $\bar{\mathbf{h}}$ (can be recovered a posteriori).

- **Vanilla srGW**: solved using Conditional gradient with optimal step size

Algorithm 1 srGW - CG iteration

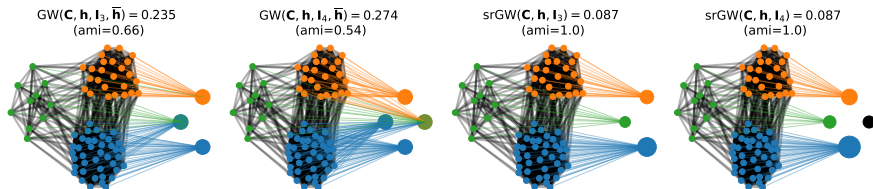
- 1: $\mathbf{G}^{(t)} \leftarrow$ gradient w.r.t \mathbf{T} . $O(N^2\bar{N} + N\bar{N}^2)$
 - 2: $\mathbf{X}^{(t)} \leftarrow \min_{\mathbf{X} \mathbf{1}_m = \mathbf{h}} \langle \mathbf{X}, \mathbf{G}^{(t)} \rangle$ $O(N\bar{N}) + \text{GPU} \parallel O(N^2\bar{N} + N\bar{N}^2)$
 - 3: $\mathbf{T}^{(t+1)} \leftarrow$ exact-line search. $O(N^2\bar{N} + N\bar{N}^2)$
-

Algorithm 2 GW - CG iteration

- 1: $\mathbf{G}^{(t)} \leftarrow$ gradient w.r.t \mathbf{T} .
 - 2: $\mathbf{X}^{(t)} \leftarrow W_{\mathbf{G}^{(t)}}(\mathbf{h}, \bar{\mathbf{h}})$
 - 3: $\mathbf{T}^{(t+1)} \leftarrow$ exact-line search.
-

- **Entropic regularized srGW_e** [Cuturi, 2013a, Peyré et al., 2016]:
 - Dense \mathbf{T}^* and $\bar{\mathbf{h}}$ informally taking uncertainty into account.
 - Solved with mirror descent much more efficient than GW.
 - One Bregman projection (softmax) instead of solving a Sinkhorn at each iteration.
- **Sparsity promoting regularization srGW_g**:
 - compress the localization over a few nodes of $\bar{\mathbf{D}}$ using group-lasso on $\bar{\mathbf{h}}$.
 - Solve with Majorization Minimization [Courty et al., 2014].

srGW for graph partitioning



- $\bar{\mathbf{h}}$ efficiently estimates cluster proportions.
- Recover the true number of clusters (3).
- Benchmark on real datasets:
 - srGW / GW using Adjacency & Heat kernels on Laplacian [Chowdhury and Needham, 2021].
 - srGW outperforms unsupervised graph partitioning SOTA on 4 datasets out of 6.
 - Entropic regularization useful for sparse real-world graphs.

Learn Optimal target structure

$$\min_{\overline{D}} \frac{1}{I} \sum_{i \leq I} \text{srGW}(\mathbf{D}_i, \mathbf{h}_i, \overline{D})$$

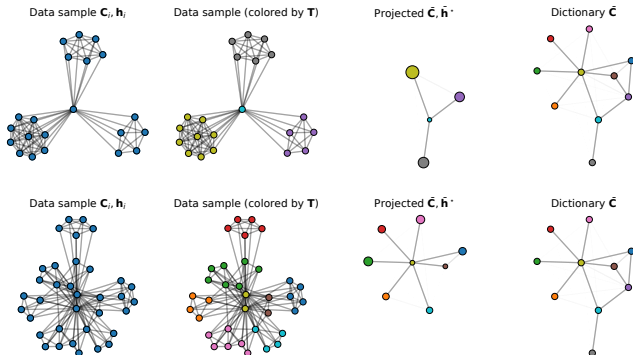
- For graphs $\{(\mathbf{D}_i, \mathbf{h}_i)\}_{i \leq I}$, learn a target structure \overline{D} minimizing on average all srGW divergences.
- $\{(\mathbf{D}_i, \mathbf{h}_i)\}$ embedded as $\{\overline{\mathbf{h}}_i\} = \{\mathbf{T}_i^{*T} \mathbf{1}\}$ where $\mathbf{T}_i^* \leftarrow \text{srGW}(\mathbf{D}_i, \mathbf{h}_i, \overline{D})$.
- Embedded graphs $\{(\overline{D}, \overline{\mathbf{h}}_i)\}$ leverage information from every subgraphs of the atom \overline{D} .
- **Online stochastic solver** scaling to large datasets [Mairal et al., 2009].

Unmixing time on the dictionary

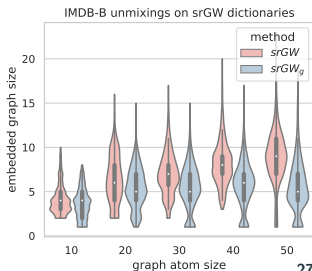
- Average timings in ms.
- srGW 100 – 1000 times faster than competitors.
- Can be executed on GPU.

	NO ATTRIBUTE			
	IMDB-B		IMDB-M	
	(-)	(+)	(-)	(+)
srGW (ours)	1.51	2.62	0.83	1.59
<i>srGW_g</i>	1.95	6.11	1.06	5.53
GWF-f	219	651	103	373
GDL	108	236	43.8	152

srGW Dictionary Learning on IMDB-B

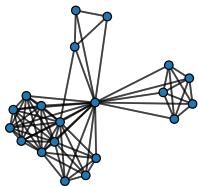


- Different local patterns depending on the dictionary size \bar{N} ,
e.g. clusters, hubs, subclusters etc.

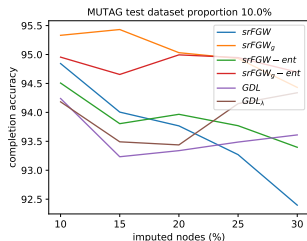
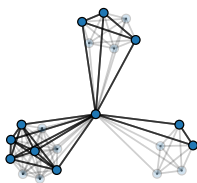


Completion of graphs

fully observed graph



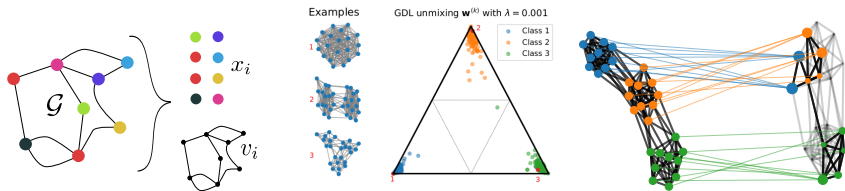
partially observed graph



- 1) Learn a srGW dictionary \overline{D} on fully observed graphs
- 2) For a partially observed graph D_{obs} , complete its full structure \tilde{D} solving for:

$$\min_{D_{imp}} \text{srGW}(\tilde{D}, h, \overline{D}), \text{ where } \tilde{D} = \begin{bmatrix} D_{obs} & \vdots \\ \dots & D_{imp} \end{bmatrix},$$

- 3) Recover Adjacency matrix of \tilde{D} by thresholding if you learned on adjacency matrices.



Gromov-Wasserstein family for graph modeling

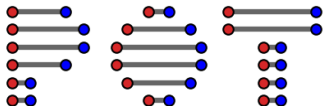
- Graphs modelled as distributions, \mathcal{GW} can measure their similarity.
- Extensions of GW for labeled graphs and Fréchet means can be computed.
- Nonlinear and linear dictionaries of graphs using \mathcal{GW} provide a good modeling.
- Relaxing the marginal constraints can sometimes better model the graphs.

Open questions and future works

- Stability of the \mathcal{GW} plan to perturbations of D (related to the GDL upper bound).
- Use \mathcal{GW} as a "kernel" for structured prediction ([Brogat-Motte et al., 2022]).
- Using GW/FGW/srGW in Graph Neural Networks (pooling, representations).

Thank you

Python code available on GitHub:



<https://github.com/PythonOT/POT>

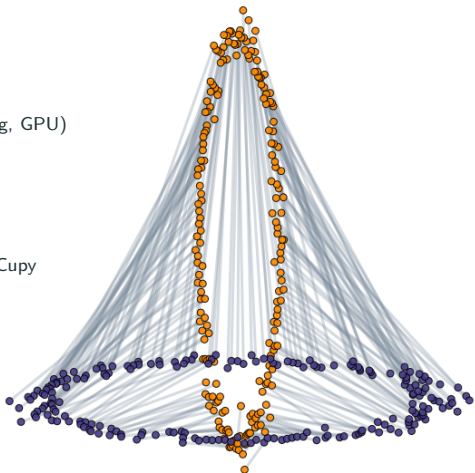
- OT LP solver, Sinkhorn (stabilized, ϵ -scaling, GPU)
- Domain adaptation with OT.
- Barycenters, Wasserstein unmixing.
- Gromov Wasserstein.
- Solvers for Numpy/Pytorch/Jax/tensorflow/Cupy

Tutorial on OT for ML:

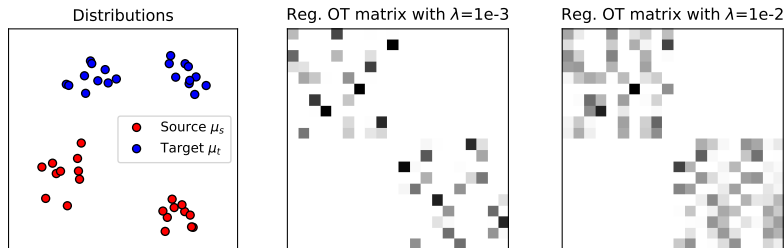
<http://tinyurl.com/otml-isbi>

Papers available on my website:

<https://remi.flamary.com/>



Entropic regularized optimal transport

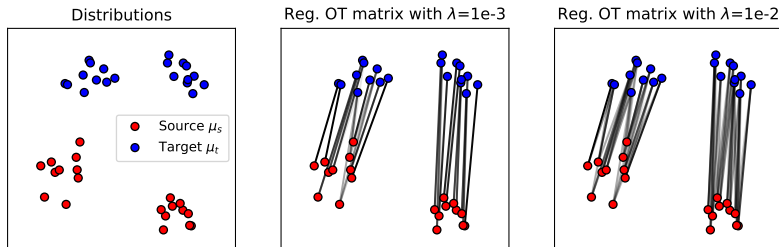


Entropic regularization [Cuturi, 2013b]

$$W_\epsilon(\mu_s, \mu_t) = \min_{\mathbf{T} \in \Pi(\mu_s, \mu_t)} \langle \mathbf{T}, \mathbf{C} \rangle_F + \epsilon \sum_{i,j} T_{i,j} \log T_{i,j}$$

- Regularization with the negative entropy $-H(\mathbf{T})$.
- Looses sparsity, but strictly convex optimization problem [Benamou et al., 2015].
- Can be solved with the very efficient Sinkhorn-Knopp matrix scaling algorithm.
- Loss and OT matrix are differentiable and have better statistical properties [Genevay et al., 2018].

Entropic regularized optimal transport



Entropic regularization [Cuturi, 2013b]

$$W_\epsilon(\mu_s, \mu_t) = \min_{\mathbf{T} \in \Pi(\mu_s, \mu_t)} \langle \mathbf{T}, \mathbf{C} \rangle_F + \epsilon \sum_{i,j} T_{i,j} \log T_{i,j}$$

- Regularization with the negative entropy $-H(\mathbf{T})$.
- Looses sparsity, but strictly convex optimization problem [Benamou et al., 2015].
- Can be solved with the very efficient Sinkhorn-Knopp matrix scaling algorithm.
- Loss and OT matrix are differentiable and have better statistical properties [Genevay et al., 2018].

$$\mathcal{FGW}_{p,q,\alpha}(D, D', \mu_s, \mu_t) = \left(\min_{T \in \Pi(\mu_s, \mu_t)} \sum_{i,j,k,l} ((1-\alpha)C_{i,j}^q + \alpha |D_{i,k} - D'_{j,l}|^q)^p T_{i,j} T_{k,l} \right)^{\frac{1}{p}}$$

Metric properties [Vayer et al., 2020]

- \mathcal{FGW} defines a metric over structured data with **measure and features preserving isometries** as invariants.
- \mathcal{FGW} is a metric for $q = 1$ a semi metric for $q > 1$, $\forall p \geq 1$.
- The distance is nul *iff* :
 - There exists a Monge map $T \# \mu_s = \mu_t$.
 - Structures are equivalent through this Monge map (isometry).
 - Features are equal through this Monge map.

Bounds and convergence to finite samples [Vayer et al., 2020]

- $\mathcal{FGW}(\mu_s, \mu_t)$ is lower bounded by $(1 - \alpha)\mathcal{W}(\mu_A, \mu_B)^q$ and $\alpha\mathcal{GW}(\mu_X, \mu_Y)^q$
- Convergence of finite samples when $\mathcal{X} = \mathcal{Y}$ with $d = \text{Dim}(\mathcal{X}) + \text{Dim}(\Omega)$:

$$\mathbb{E}[\mathcal{FGW}(\mu, \mu_n)] = O\left(n^{-\frac{1}{d}}\right)$$

Solving the Gromov Wasserstein optimization problem

Optimization problem

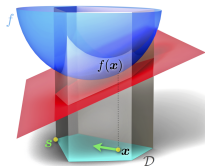
$$\mathcal{GW}_p^p(\mu_s, \mu_t) = \min_{\mathbf{T} \in \Pi(\mu_s, \mu_t)} \sum_{i,j,k,l} |D_{i,k} - D'_{j,l}|^p T_{i,j} T_{k,l}$$

with $\mu_s = \sum_i a_i \delta_{\mathbf{x}_i^s}$ and $\mu_t = \sum_j b_j \delta_{\mathbf{x}_j^t}$ and $D_{i,k} = \|\mathbf{x}_i^s - \mathbf{x}_k^s\|$, $D'_{j,l} = \|\mathbf{x}_j^t - \mathbf{x}_l^t\|$

- Quadratic Program (Wasserstein is a linear program).
- Nonconvex, NP-hard, related to Quadratic Assignment Problem (QAP).
- Large problem and non convexity forbid standard QP solvers.

Optimization algorithms

- Local solution with conditional gradient algorithm (Frank-Wolfe) [Frank and Wolfe, 1956].
- Each FW iteration requires solving an OT problems.
- Gromov in 1D has a close form (solved in discrete with a sort) [Vayer et al., 2019].
- With entropic regularization, one can use mirror descent [Peyré et al., 2016] or fast low rank approximations [Scetbon et al., 2021].



Optimization Problem

$$\mathcal{GW}_{p,\epsilon}^p(\mu_s, \mu_t) = \min_{\mathbf{T} \in \Pi(\mu_s, \mu_t)} \sum_{i,j,k,l} |D_{i,k} - D'_{j,l}|^p T_{i,j} T_{k,l} + \epsilon \sum_{i,j} T_{i,j} \log T_{i,j} \quad (4)$$

with $\mu_s = \sum_i a_i \delta_{\mathbf{x}_i^s}$ and $\mu_t = \sum_j b_j \delta_{\mathbf{x}_j^t}$ and $D_{i,k} = \|\mathbf{x}_i^s - \mathbf{x}_k^s\|$, $D'_{j,l} = \|\mathbf{x}_j^t - \mathbf{x}_l^t\|$

- Smoothing the original GW with a convex and smooth entropic term.

Solving the entropic GW [Peyré et al., 2016]

- Problem (4) can be solved using a KL mirror descent.
- This is equivalent to solving at each iteration t

$$\mathbf{T}^{(t+1)} = \min_{\mathbf{T} \in \mathcal{P}} \left\langle \mathbf{T}, \mathbf{G}^{(t)} \right\rangle_F + \epsilon \sum_{i,j} T_{i,j} \log T_{i,j}$$

Where $G_{i,j}^{(t)} = 2 \sum_{k,l} |D_{i,k} - D'_{j,l}|^p T_{k,l}^{(t)}$ is the gradient of the GW loss at previous point $\mathbf{T}^{(k)}$.

- Problem above solved using a Sinkhorn-Knopp algorithm of entropic OT.
- Very fast approximation exist for low rank distances [Scetbon et al., 2021].

Optimization problem

$$\min_{\mathbf{w} \in \Sigma_S} \mathcal{GW}_2^2 \left(\sum_{s \in [S]} w_s \overline{\mathbf{D}}_s, \mathbf{D} \right) - \lambda \|\mathbf{w}\|_2^2$$

- Non-convex Quadratic Program *w.r.t.* \mathbf{T} and \mathbf{w} .
- GW for fixed \mathbf{w} already have an existing Frank-Wolfe solver.
- We proposed a Block Coordinate Descent algorithm

BCD Algorithm for sparse GW unmixing [Tseng, 2001]

- 1: **repeat**
 - 2: Compute OT matrix \mathbf{T} of $\mathcal{GW}_2^2(\mathbf{D}, \sum_s w_s \overline{\mathbf{D}}_s)$, with FW [Vayer et al., 2018].
 - 3: Compute the optimal \mathbf{w} given \mathbf{T} with Frank-Wolfe algorithm.
 - 4: **until** convergence
- Since the problem is quadratic optimal steps can be obtained for both FW.
 - BCD convergence in practice in a few tens of iterations.

Approximating GW in the linear embedding

GW Upper bound [Vincent-Cuaz et al., 2021]

Let two graphs of order N in the linear embedding $\left(\sum_s w_s^{(1)} \overline{\mathbf{D}}_s\right)$ and $\left(\sum_s w_s^{(2)} \overline{\mathbf{D}}_s\right)$, the \mathcal{GW} divergence can be upper bounded by

$$\mathcal{GW}_2 \left(\sum_{s \in [S]} w_s^{(1)} \overline{\mathbf{D}}_s, \sum_{s \in [S]} w_s^{(2)} \overline{\mathbf{D}}_s \right) \leq \|\mathbf{w}^{(1)} - \mathbf{w}^{(2)}\|_M \quad (5)$$

with M a PSD matrix of components $M_{p,q} = \langle \mathbf{D}_h \overline{\mathbf{D}}_p, \overline{\mathbf{D}}_q \mathbf{D}_h \rangle_F$, $\mathbf{D}_h = \text{diag}(\mathbf{h})$.

Discussion

- The upper bound is the value of GW for a transport $T = \text{diag}(\mathbf{h})$ assuming that the nodes are already aligned.
- The bound is exact when the weights $\mathbf{w}^{(1)}$ and $\mathbf{w}^{(2)}$ are close.
- Solving \mathcal{GW} with FW is $O(N^3 \log(N))$ at each iterations.
- Computing the Mahalanobis upper bound is $O(S^2)$: very fast alternative to GW for nearest neighbors retrieval.

GDL on labeled graphs

- For datasets with labeled graphs, one can learn simultaneously a dictionary of the structure $\{\overline{\mathbf{D}}_s\}_{s \in [S]}$ and a dictionary on the labels/features $\{\overline{\mathbf{F}}_s\}_{s \in [S]}$.
- Data fitting is Fused Gromov-Wasserstein distance \mathcal{FGW} , same stochastic algorithm.

Dictionary on weights

$$\min_{\substack{\{(\mathbf{w}^{(k)}, \mathbf{v}^{(k)})\}_k \\ \{(\overline{\mathbf{D}}_s, \overline{\mathbf{h}}_s)\}_s}} \sum_{k=1}^K \mathcal{GW}_2^2 \left(\mathbf{D}^{(k)}, \sum_s w_s^{(k)} \overline{\mathbf{D}}_s, \mathbf{h}^{(k)}, \sum_s v_s^{(k)} \overline{\mathbf{h}}_s \right) - \lambda \|\mathbf{w}^{(k)}\|_2^2 - \mu \|\mathbf{v}^{(k)}\|_2^2$$

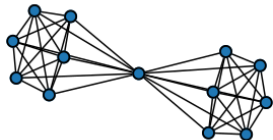
- We model the graphs as a linear model on the structure and the node weights

$$(\mathbf{D}^{(k)}, \mathbf{h}^{(k)}) \longrightarrow \left(\sum_s w_s^{(k)} \overline{\mathbf{D}}_s, \sum_s v_s^{(k)} \overline{\mathbf{h}}_s \right)$$

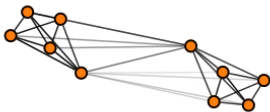
- This allows for sparse weights \mathbf{h} so embedded graphs with different order.
- We provide in [Vincent-Cuaz et al., 2021] subgradients of GW *w.r.t.* the mass \mathbf{h} .

Experiments - Unsupervised representation learning

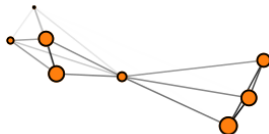
Graph from dataset



Model unif. \mathbf{h} (GW=0.09)



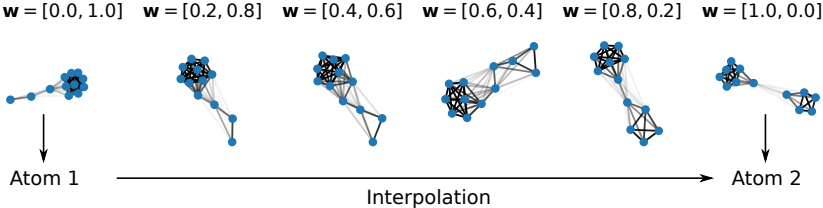
Model est. $\tilde{\mathbf{h}}$ (GW=0.08)



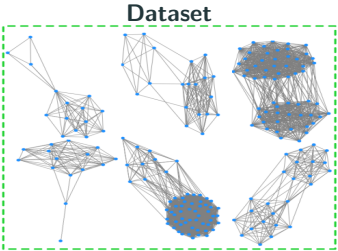
Comparison of fixed and learned weights dictionaries

- Graph taken from the IMBD dataset.
- Show original graph and representation after projection on the embedding.
- Uniform weight \mathbf{h} has a hard time representing a central node.
- Estimated weights $\tilde{\mathbf{h}}$ recover a central node.
- In addition some nodes are discarded with 0 weight (graphs can change order).

Experiments - Unsupervised representation learning



Learned Dictionary: Interpolation ~ 1D Manifold



- Stochastic block model with 2 blocks and varying proportions of block size.
- GDL with 2 atoms can recover the extreme points.
- Linear interpolation recover a continuous variation of proportion.

Table 1. Clustering: Rand Index computed for benchmarked approaches on real datasets.

models	no attribute		discrete attributes		real attributes			
	IMDB-B	IMDB-M	MUTAG	PTC-MR	BZR	COX2	ENZYMES	PROTEIN
GDL(ours)	51.64(0.59)	55.41(0.20)	70.89(0.11)	51.90(0.54)	66.42(1.96)	59.48(0.68)	66.97(0.93)	60.49(0.71)
GWF-r	51.24 (0.02)	55.54(0.03)	-	-	52.42(2.48)	56.84(0.41)	72.13(0.19)	59.96(0.09)
GWF-f	50.47(0.34)	54.01(0.37)	-	-	51.65(2.96)	52.86(0.53)	71.64(0.31)	58.89(0.39)
GW-k	50.32(0.02)	53.65(0.07)	57.56(1.50)	50.44(0.35)	56.72(0.50)	52.48(0.12)	66.33(1.42)	50.08(0.01)
SC	50.11(0.10)	54.40(9.45)	50.82(2.71)	50.45(0.31)	42.73(7.06)	41.32(6.07)	70.74(10.60)	49.92(1.23)

Clustering Experiments on real datasets





- Different data fitting losses:
 - Graphs without node attributes : Gromov-Wasserstein.
 - Graphs with node attributes (discrete and real): Fused Gromov-Wasserstein.
- We learn a dictionary on the dataset and perform K-means in the embedding using the Mahalanobis distance approximation.
- Compared to GW Factorization (GWF) [Xu, 2020] and spectral clustering.
- Similar performance for supervised classification (using GW in a kernel).

Clustering of datasets of graphs

Table 1: Embedding computation times (in ms) averaged over whole datasets on learned dictionaries. (–) (resp. (+)) denotes the fastest (resp. slowest)

	NO ATTRIBUTE				DISCRETE ATTRIBUTES				REAL ATTRIBUTES							
	IMDB-B		IMDB-M		MUTAG		PTC-MR		BZR		COX2		ENZYMES		PROTEIN	
	(-)	(+)	(-)	(+)	(-)	(+)	(-)	(+)	(-)	(+)	(-)	(+)	(-)	(+)	(-)	(+)
srGW (ours)	1.51	2.62	0.83	1.59	0.86	1.83	0.40	1.01	0.43	0.79	0.51	0.90	0.62	0.95	0.46	0.60
srGW _g	1.95	6.11	1.06	5.53	3.68	5.98	1.65	3.38	0.89	2.88	0.97	4.60	1.35	4.73	1.57	2.96
GWf-f	219	651	103	373	236	495	191	477	181	916	129	641	93	627	78	322
GDL	108	236	43.8	152	102	514	100	509	73.2	532	48.7	347	38	301	29	151

- srGW unmixings clustered using Kmeans algorithm: **perform consistently better** than SOTA OT based clustering methods over 8 datasets (including graph with features).
- **Unmixing runtimes:** 100 to 1000 times faster than fastest competitor GDL.
- **Denoising beneficial to supervised classification:** embedded graphs by srGW enhances and speeds up supervised classification performances while endowing a SVM with a GW kernel.

-  Benamou, J.-D., Carlier, G., Cuturi, M., Nenna, L., and Peyré, G. (2015).
Iterative Bregman projections for regularized transportation problems.
SISC.
-  Brogat-Motte, L., Flamary, R., Brouard, C., Rousu, J., and d'Alché Buc, F. (2022).
Learning to predict graphs with fused gromov-wasserstein barycenters.
In *International Conference in Machine Learning (ICML)*.
-  Chowdhury, S. and Needham, T. (2021).
Generalized spectral clustering via gromov-wasserstein learning.
In *International Conference on Artificial Intelligence and Statistics*, pages 712–720. PMLR.
-  Courty, N., Flamary, R., and Tuia, D. (2014).
Domain adaptation with regularized optimal transport.
In *European Conference on Machine Learning and Principles and Practice of Knowledge Discovery in Databases (ECML PKDD)*.



Cuturi, M. (2013a).

Sinkhorn distances: Lightspeed computation of optimal transport.

In *NIPS*, pages 2292–2300.



Cuturi, M. (2013b).

Sinkhorn distances: Lightspeed computation of optimal transportation.

In *Neural Information Processing Systems (NIPS)*, pages 2292–2300.



Frank, M. and Wolfe, P. (1956).

An algorithm for quadratic programming.

Naval research logistics quarterly, 3(1-2):95–110.



Genevay, A., Chizat, L., Bach, F., Cuturi, M., and Peyré, G. (2018).

Sample complexity of sinkhorn divergences.

arXiv preprint arXiv:1810.02733.



Kantorovich, L. (1942).

On the translocation of masses.

C.R. (Doklady) Acad. Sci. URSS (N.S.), 37:199–201.



Mairal, J., Bach, F., Ponce, J., and Sapiro, G. (2009).

Online dictionary learning for sparse coding.

In *Proceedings of the 26th annual international conference on machine learning*, pages 689–696.



Memoli, F. (2011).

Gromov wasserstein distances and the metric approach to object matching.





Foundations of Computational Mathematics, pages 1–71.









Monge, G. (1781).

Mémoire sur la théorie des déblais et des remblais.

De l'Imprimerie Royale.

-  Peyré, G., Cuturi, M., and Solomon, J. (2016).
Gromov-wasserstein averaging of kernel and distance matrices.
In *ICML*, pages 2664–2672.
-  Scetbon, M., Peyré, G., and Cuturi, M. (2021).
Linear-time gromov wasserstein distances using low rank couplings and costs.
arXiv preprint arXiv:2106.01128.
-  Solomon, J., Peyré, G., Kim, V. G., and Sra, S. (2016).
Entropic metric alignment for correspondence problems.
ACM Transactions on Graphics (TOG), 35(4):72.
-  Tseng, P. (2001).
Convergence of a block coordinate descent method for nondifferentiable minimization.
Journal of optimization theory and applications, 109(3):475–494.

-  Vayer, T., Chapel, L., Flamary, R., Tavenard, R., and Courty, N. (2018).
Fused gromov-wasserstein distance for structured objects: theoretical foundations and mathematical properties.
-  Vayer, T., Chapel, L., Flamary, R., Tavenard, R., and Courty, N. (2020).
Fused gromov-wasserstein distance for structured objects.
Algorithms, 13 (9):212.
-  Vayer, T., Flamary, R., Tavenard, R., Chapel, L., and Courty, N. (2019).
Sliced gromov-wasserstein.
In *Neural Information Processing Systems (NeurIPS)*.
-  Vincent-Cuaz, C., Flamary, R., Corneli, M., Vayer, T., and Courty, N. (2022).
Semi-relaxed gromov wasserstein divergence with applications on graphs.
In *International Conference on Learning Representations (ICLR)*.

-  Vincent-Cuaz, C., Vayer, T., Flamary, R., Corneli, M., and Courty, N. (2021).
Online graph dictionary learning.
In *International Conference on Machine Learning (ICML)*.
-  Xu, H. (2020).
Gromov-wasserstein factorization models for graph clustering.
In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 34,
pages 6478–6485.