Optimal Transport for Machine Learning

10 years of least effort

Rémi Flamary, École polytechnique

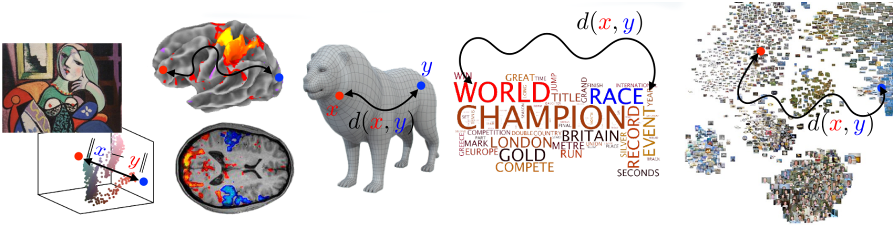
October 2nd 2023

Mathematical Foundations of AI, 2023, Paris

Distributions are everywhere in machine learning

- Images, vision, graphics, Time series, text, genes, proteins.
- Many datum and datasets can be seen as distributions.
- Important questions:
    - How to compare distributions?
    - How to use the geometry of distributions?
- Optimal transport provides many tools that can answer those questions.

Illustration from the slides of Gabriel Peyré.



Distributions are everywhere in machine learning

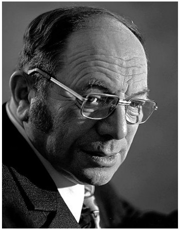
- Images, vision, graphics, Time series, text, genes, proteins.
- Many datum and datasets can be seen as distributions.
- Important questions:
  - How to compare distributions?
  - How to use the geometry of distributions?
- Optimal transport provides many tools that can answer those questions.

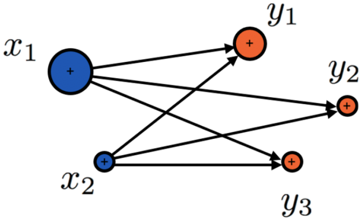
Illustration from the slides of Gabriel Peyré.

Table of content

Outline

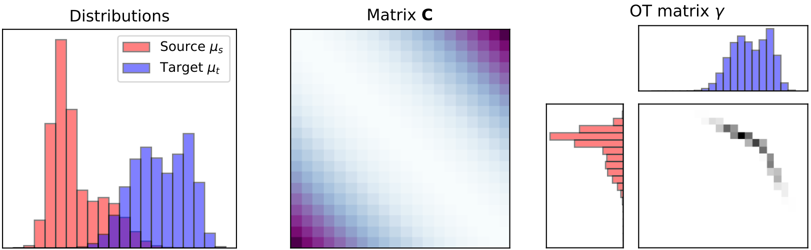
Optimal transport





- 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 $\mu_t = \sum_{i=1}^{n_t} b_i \delta_{\mathbf{x}_i^t}$

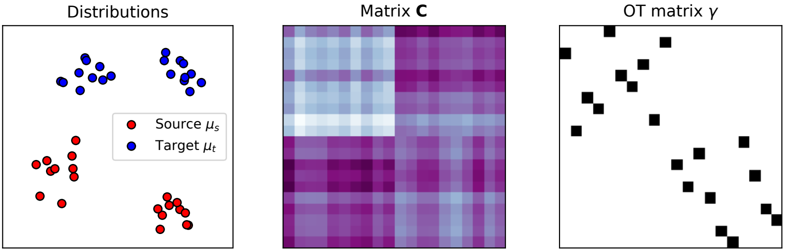
$$W_p^p(\mu_s, \mu_t) = \min_{\boldsymbol{T} \in \Pi(\mu_s, \mu_t)} \left\{ \langle \boldsymbol{T}, \mathbf{C} \rangle_F = \sum_{i,j} T_{i,j} c_{i,j} \right\}$$

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

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

- Solving the OT problem with network simplex is $O(n^3 \log(n))$ for $n = n_s = n_t$.
- $W_p(\mu_s, \mu_t)$ is called the Wasserstein distance (EMD for $p = 1$).

Optimal transport between discrete distributions



Kantorovitch formulation : OT Linear Program

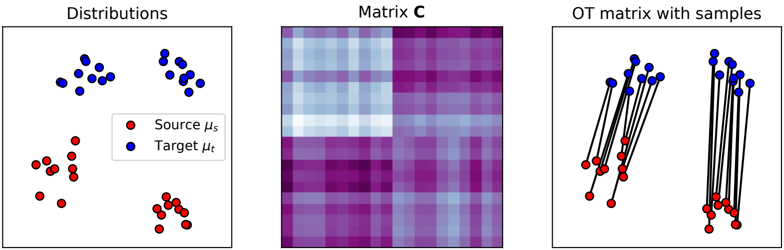
When $\mu_s = \sum_{i=1}^{n_s} a_i \delta_{\mathbf{x}_i^s}$ and $\mu_t = \sum_{i=1}^{n_t} b_i \delta_{\mathbf{x}_i^t}$

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

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

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

- Solving the OT problem with network simplex is $O(n^3 \log(n))$ for $n = n_s = n_t$.
- $W_p(\mu_s, \mu_t)$ is called the Wasserstein distance (EMD for $p = 1$).



Kantorovitch formulation : OT Linear Program

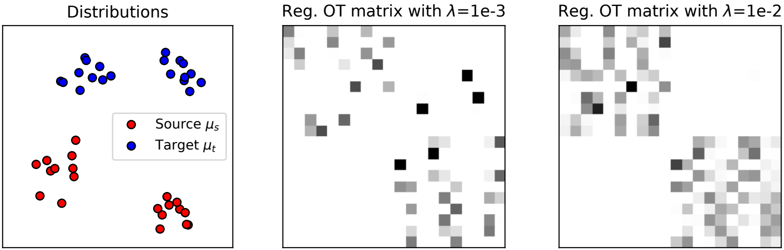
When $\mu_s = \sum_{i=1}^{n_s} a_i \delta_{\mathbf{x}_i^s}$ and $\mu_t = \sum_{i=1}^{n_t} b_i \delta_{\mathbf{x}_i^t}$

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

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

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

- Solving the OT problem with network simplex is $O(n^3 \log(n))$ for $n = n_s = n_t$.
- $W_p(\mu_s, \mu_t)$ is called the Wasserstein distance (EMD for $p = 1$).



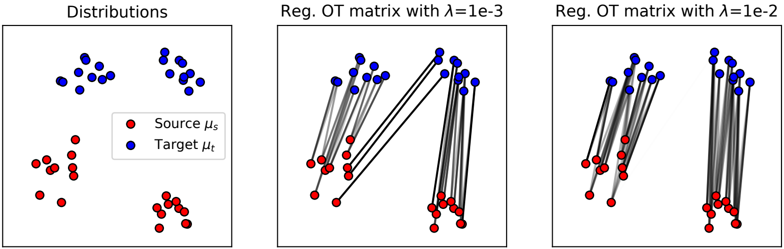
Entropic regularization [Cuturi, 2013]

$$\mathbf{T}_0^\lambda = \underset{\mathbf{T}\in\Pi(\mu_s,\mu_t)}{\arg\min} \quad \langle \mathbf{T}, \mathbf{C}\rangle_F + \lambda \sum_{i,j} T_{i,j}(\log T_{i,j} - 1)$$

- Regularization with the negative entropy of $\boldsymbol{T}$.
- Looses sparsity but smooth and strictly convex optimization problem.
- Can be solved efficiently with Sinkhorn's matrix scaling algorithm with $\mathbf{u}^{(0)} = \mathbf{1}, \mathbf{K} = \exp(-\mathbf{C}/\lambda)$ and $\mathbf{T} = \text{diag}(\mathbf{u}^\star)\mathbf{K}\text{diag}(\mathbf{v}^\star)$

$$\mathbf{v}^{(k)} = \mathbf{b} \oslash \mathbf{K}^\top \mathbf{u}^{(k-1)}, \quad \mathbf{u}^{(k)} = \mathbf{a} \oslash \mathbf{K}\mathbf{v}^{(k)}$$

Entropic regularized optimal transport



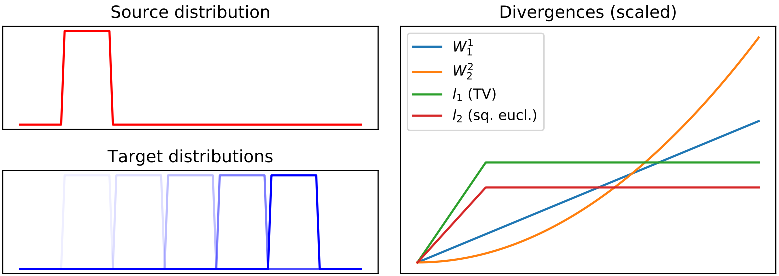
Entropic regularization [Cuturi, 2013]

$$\mathbf{T}_0^\lambda = \underset{\mathbf{T}\in\Pi(\mu_s,\mu_t)}{\arg\min} \quad \langle \mathbf{T}, \mathbf{C}\rangle_F + \lambda \sum_{i,j} T_{i,j}(\log T_{i,j} - 1)$$

- Regularization with the negative entropy of $\boldsymbol{T}$.
- Looses sparsity but smooth and strictly convex optimization problem.
- Can be solved efficiently with Sinkhorn's matrix scaling algorithm with $\mathbf{u}^{(0)} = \mathbf{1}, \mathbf{K} = \exp(-\mathbf{C}/\lambda)$ and $\mathbf{T} = \text{diag}(\mathbf{u}^\star)\mathbf{K}\text{diag}(\mathbf{v}^\star)$

$$\mathbf{v}^{(k)} = \mathbf{b} \oslash \mathbf{K}^\top \mathbf{u}^{(k-1)}, \quad \mathbf{u}^{(k)} = \mathbf{a} \oslash \mathbf{K}\mathbf{v}^{(k)}$$

Wasserstein distance

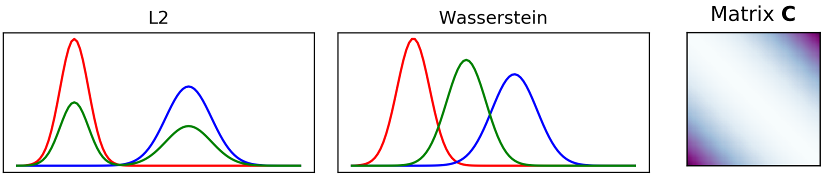


Wasserstein distance

$$W_p^p(\mu_s, \mu_t) = \min_{\gamma \in \mathcal{P}} \int_{\Omega_s \times \Omega_t} \|\mathbf{x} - \mathbf{y}\|^p \gamma(\mathbf{x}, \mathbf{y}) d\mathbf{x} d\mathbf{y} = \mathbb{E}_{(\mathbf{x},\mathbf{y})\sim\gamma}[\|\mathbf{x} - \mathbf{y}\|^p] \quad (1)$$

In this case we have $c(\mathbf{x}, \mathbf{y}) = \|\mathbf{x} - \mathbf{y}\|^p$

- A.K.A. Earth Mover's Distance (W_1^1) [Rubner et al., 2000].
- Useful between discrete distribution even without overlapping support.
- Smooth approximation can be computed with Sinkhorn [Cuturi, 2013].
- **Wasserstein barycenter**: $\overline{\mu} = \arg\min_\mu \sum_i w_i W_p^p(\mu, \mu_i)$

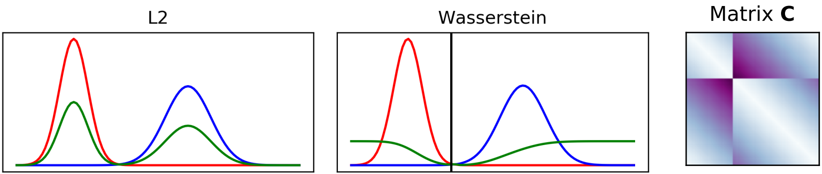


Wasserstein distance

$$W_p^p(\color{red}{\mu_s}, \color{blue}{\mu_t}) = \min_{\gamma \in \mathcal{P}} \int_{\Omega_s \times \Omega_t} \|\mathbf{x} - \mathbf{y}\|^p \gamma(\mathbf{x}, \mathbf{y}) d\mathbf{x} d\mathbf{y} = \mathbb{E}_{(\mathbf{x},\mathbf{y}) \sim \gamma}[\|\mathbf{x} - \mathbf{y}\|^p] \quad (1)$$

In this case we have $c(\mathbf{x}, \mathbf{y}) = \|\mathbf{x} - \mathbf{y}\|^p$

- A.K.A. Earth Mover's Distance (W_1^1) [Rubner et al., 2000].
- Useful between discrete distribution even without overlapping support.
- Smooth approximation can be computed with Sinkhorn [Cuturi, 2013].
- **Wasserstein barycenter**: $\overline{\mu} = \arg\min_\mu \sum_i w_i W_p^p(\mu, \mu_i)$

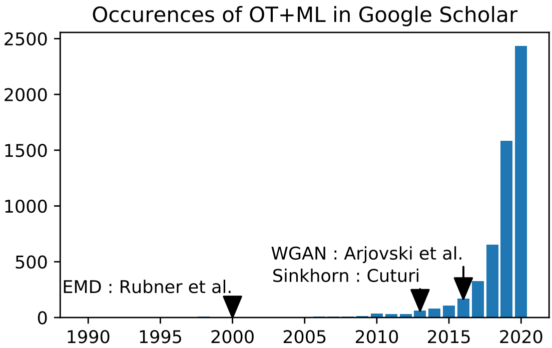


Wasserstein distance

$$W_p^p(\mu_s, \mu_t) = \min_{\gamma \in \mathcal{P}} \int_{\Omega_s \times \Omega_t} \|\mathbf{x} - \mathbf{y}\|^p \gamma(\mathbf{x}, \mathbf{y}) d\mathbf{x} d\mathbf{y} = \mathbb{E}_{(\mathbf{x},\mathbf{y}) \sim \gamma}[\|\mathbf{x} - \mathbf{y}\|^p] \quad (1)$$

In this case we have $c(\mathbf{x}, \mathbf{y}) = \|\mathbf{x} - \mathbf{y}\|^p$

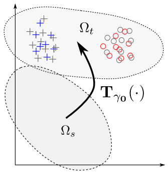
- A.K.A. Earth Mover's Distance (W_1^1) [Rubner et al., 2000].
- Useful between discrete distribution even without overlapping support.
- Smooth approximation can be computed with Sinkhorn [Cuturi, 2013].
- **Wasserstein barycenter**: $\overline{\mu} = \arg\min_\mu \sum_i w_i W_p^p(\mu, \mu_i)$

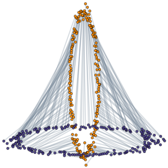


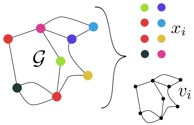
Short history of OT for ML

- Proposed in in image processing by [Rubner et al., 2000] (EMD).
- Entropic regularized OT allows fast approximation [Cuturi, 2013].
- Deep learning/ stochastic optimization [Arjovsky et al., 2017].
- Generative models with diffusion/Schrödinger bridges.

Three aspects of optimal transport







Transporting with optimal transport

- Learn to map between distributions.
- Estimate a smooth mapping from discrete distributions.
- Applications in domain adaptation.

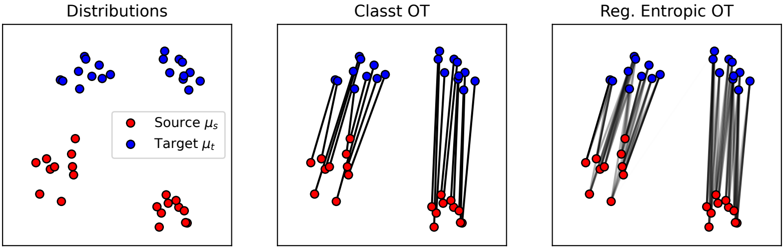
Divergence between histograms/empirical distributions

- Use the ground metric to encode complex relations between the bins of histograms for data fitting.
- OT losses are non-parametric divergences between non overlapping distributions.
- Used to train minimal Wasserstein estimators.

Divergence between structured objects and spaces

- Modeling of structured data and graphs as distribution.
- OT losses (Wass. or (F)GW) measure similarity between distributions/objects.
- OT find correspondance across spaces for adaptation.

Outline



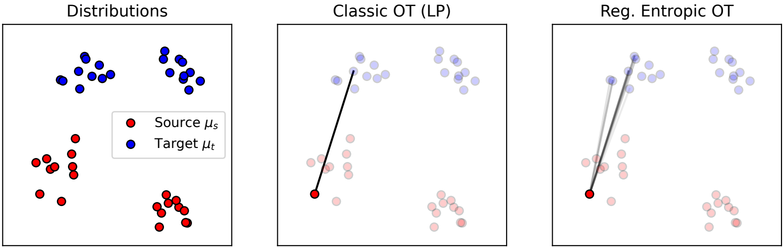
Mapping estimation

- Barycentric mapping using the OT matrix $\boldsymbol{T}$ [Ferradans et al., 2014].

$$\widehat{m}_{\boldsymbol{T}}(\mathbf{x}_i^s) = \arg\min_{\mathbf{x}} \quad \sum_j \boldsymbol{T}_{i,j} c(\mathbf{x}, \mathbf{x}_j^t)$$

- Smooth entropic mapping [Seguy et al., 2017, Pooladian and Niles-Weed, 2021].
- Linear Monge mapping when data supposed Gaussian [Flamary et al., 2019].
- Estimation for W_2 using input convex neural networks [Makkuva et al., 2020].

Mapping with optimal transport

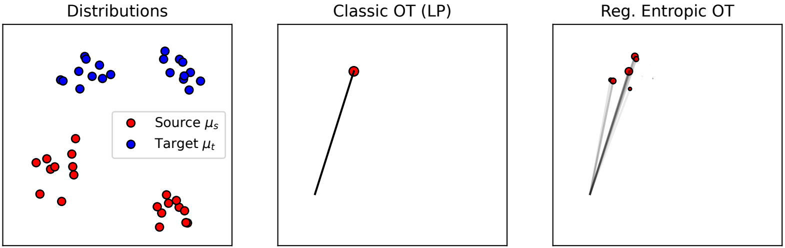


Mapping estimation

- Barycentric mapping using the OT matrix $\boldsymbol{T}$ [Ferradans et al., 2014].

$$\widehat{m}_{\boldsymbol{T}}(\mathbf{x}_i^s) = \arg\min_{\mathbf{x}} \quad \sum_j \boldsymbol{T}_{i,j} c(\mathbf{x}, \mathbf{x}_j^t)$$

- Smooth entropic mapping [Seguy et al., 2017, Pooladian and Niles-Weed, 2021].
- Linear Monge mapping when data supposed Gaussian [Flamary et al., 2019].
- Estimation for W_2 using input convex neural networks [Makkuva et al., 2020].



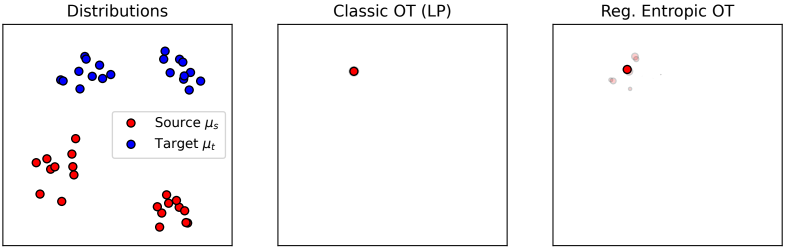
Mapping estimation

- Barycentric mapping using the OT matrix $\boldsymbol{T}$ [Ferradans et al., 2014].

$$\widehat{m}_{\boldsymbol{T}}(\mathbf{x}_i^s) = \arg\min_{\mathbf{x}} \sum_j \boldsymbol{T}_{i,j} c(\mathbf{x}, \mathbf{x}_j^t)$$

- Smooth entropic mapping [Seguy et al., 2017, Pooladian and Niles-Weed, 2021].
- Linear Monge mapping when data supposed Gaussian [Flamary et al., 2019].
- Estimation for W_2 using input convex neural networks [Makkuva et al., 2020].

Mapping with optimal transport



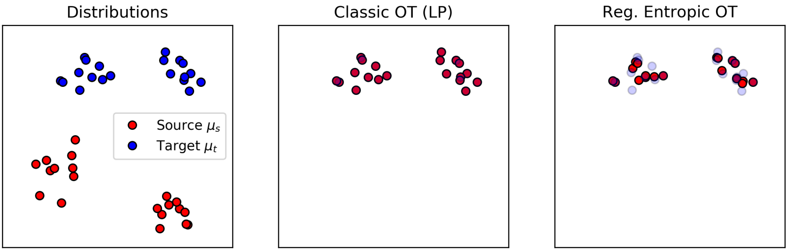
Mapping estimation

- Barycentric mapping using the OT matrix $\boldsymbol{T}$ [Ferradans et al., 2014].

$$\widehat{m}_{\boldsymbol{T}}(\mathbf{x}_i^s) = \arg\min_{\mathbf{x}} \quad \sum_j \boldsymbol{T}_{i,j} c(\mathbf{x}, \mathbf{x}_j^t)$$

- Smooth entropic mapping [Seguy et al., 2017, Pooladian and Niles-Weed, 2021].
- Linear Monge mapping when data supposed Gaussian [Flamary et al., 2019].
- Estimation for W_2 using input convex neural networks [Makkuva et al., 2020].

Mapping with optimal transport



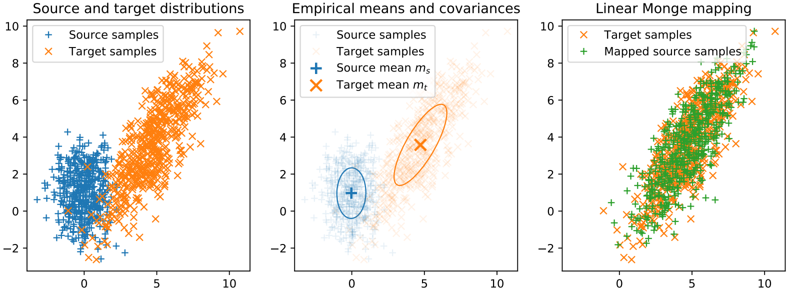
Mapping estimation

- Barycentric mapping using the OT matrix $\boldsymbol{T}$ [Ferradans et al., 2014].
- Smooth entropic mapping [Seguy et al., 2017, Pooladian and Niles-Weed, 2021].

$$\widehat{m}(\mathbf{x}) = \frac{\sum_j \mathbf{x}_j^t v_j \exp(-\|\mathbf{x} - \mathbf{x}_j^t\|^2/\lambda)}{\sum_j v_j \exp(-\|\mathbf{x} - \mathbf{x}_j^t\|^2/\lambda)}, \quad \text{with } \mathbf{v} \text{ sol. of Sinkhorn}$$

- Linear Monge mapping when data supposed Gaussian [Flamary et al., 2019].
- Estimation for W_2 using input convex neural networks [Makkuva et al., 2020].

Mapping with optimal transport

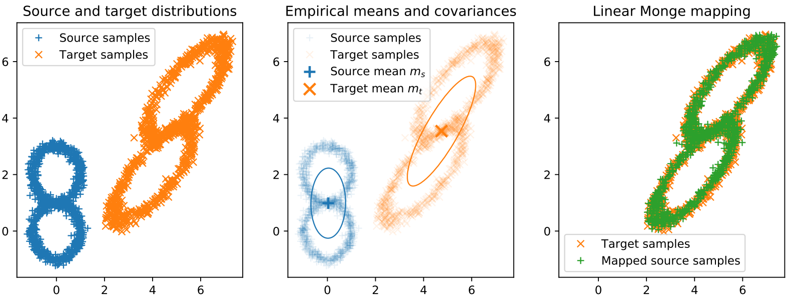


Mapping estimation

- Barycentric mapping using the OT matrix $\boldsymbol{T}$ [Ferradans et al., 2014].
- Smooth entropic mapping [Seguy et al., 2017, Pooladian and Niles-Weed, 2021].
- Linear Monge mapping when data supposed Gaussian [Flamary et al., 2019].

$$m(\mathbf{x}) = \mathbf{m}_2 + \mathbf{A}(\mathbf{x} - \mathbf{m}_1) \quad \text{with} \quad \mathbf{A} = \Sigma_1^{-1/2}(\Sigma_1^{1/2}\Sigma_2\Sigma_1^{1/2})^{1/2}\Sigma_1^{-1/2}$$

- Estimation for W_2 using input convex neural networks [Makkuva et al., 2020].



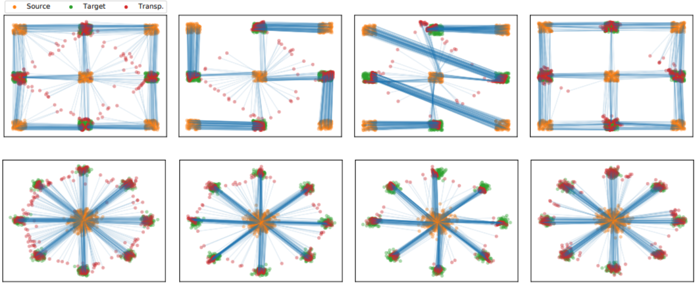
Mapping estimation

- Barycentric mapping using the OT matrix $\boldsymbol{T}$ [Ferradans et al., 2014].
- Smooth entropic mapping [Seguy et al., 2017, Pooladian and Niles-Weed, 2021].
- Linear Monge mapping when data supposed Gaussian [Flamary et al., 2019].

$$m(\mathbf{x}) = \mathbf{m}_2 + \mathbf{A}(\mathbf{x} - \mathbf{m}_1) \quad \text{with} \quad \mathbf{A} = \Sigma_1^{-1/2}(\Sigma_1^{1/2}\Sigma_2\Sigma_1^{1/2})^{1/2}\Sigma_1^{-1/2}$$

- Estimation for W_2 using input convex neural networks [Makkuva et al., 2020].

Mapping with optimal transport

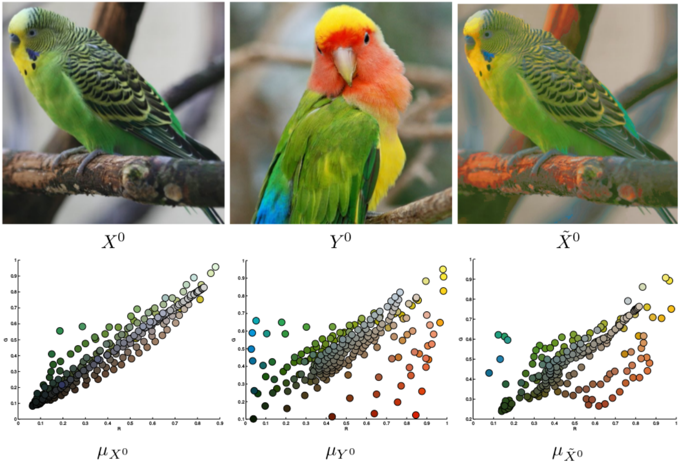


(a) Barycentric-OT (b) W1-LP (c) W2GAN (d) Our approach

Mapping estimation

- Barycentric mapping using the OT matrix $\boldsymbol{T}$ [Ferradans et al., 2014].
- Smooth entropic mapping [Seguy et al., 2017, Pooladian and Niles-Weed, 2021].
- Linear Monge mapping when data supposed Gaussian [Flamary et al., 2019].
- Estimation for W_2 using input convex neural networks [Makkuva et al., 2020].

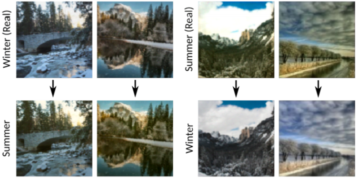
Pixels as empirical distribution [Ferradans et al., 2014]



Histogram matching in images

Image colorization [Ferradans et al., 2014]

OT mapping for Image-to-Image translation



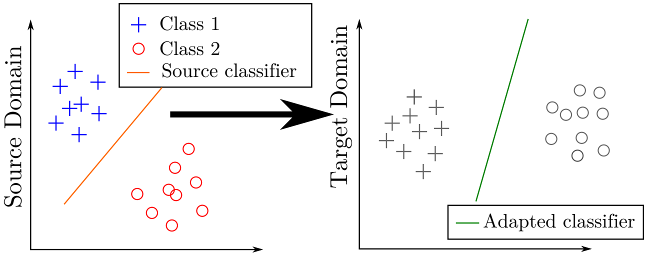
Principle

- Encode image as a distribution in a DNN embedding.
- Transform between images using estimated Monge mapping.
- Linear Monge Mapping (Wasserstein Style Transfer [Mroueh, 2019]).
- Nonlinear Monge Mapping using input Convex Neural Networks [Korotin et al., 2019].
- Allows for transformation between two images but also style interpolation with Wasserstein barycenters.

Domain Adaptation

- Classification problem with data coming from different sources (domains).
- Distributions are different but related.
- Labels only available in the **source domain**, but prediction is conducted in the **target domain**.
- Objective : Train a classifier that performs well in the target domain

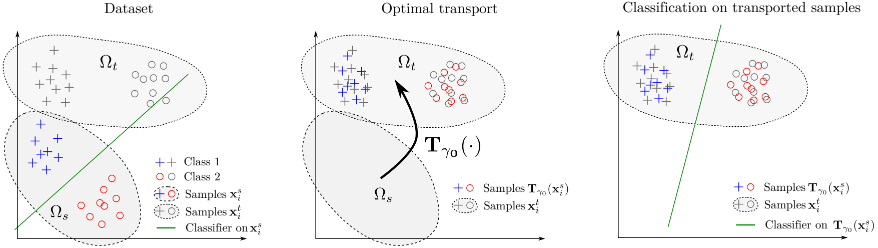
Domain Adaptation problem



Domain Adaptation

- Classification problem with data coming from different sources (domains).
- Distributions are different but related.
- Labels only available in the **source domain**, but prediction is conducted in the **target domain**.
- Objective : Train a classifier that performs well in the target domain

Optimal transport for domain adaptation



Assumptions

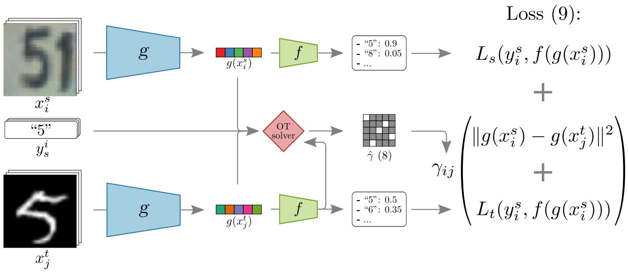
1. There exist an OT mapping m in the feature space between the two domains.
2. The transport preserves the joint distributions:

$$P^s(\mathbf{x}, y) = P^t(m(\mathbf{x}), y).$$

3-step strategy [Courty et al., 2014, Courty et al., 2016]

1. Estimate optimal transport between distributions (use regularization).
2. Transport the training samples on target domain.
3. Learn a classifier on the transported training samples.

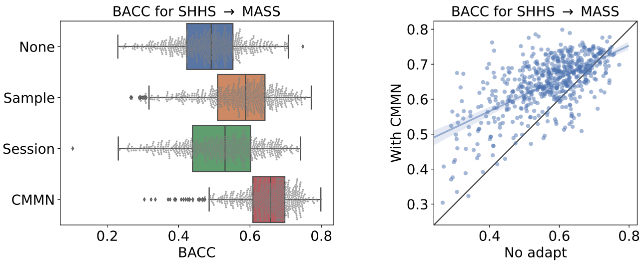
Domain adaptation with optimal transport



Extensions and related works

- JDOT [Courty et al., 2017b] : Joint OT and target predictor estimation.
- [Shen et al., 2018] : Wasserstein Distance Guided Representation Learning.
- DeepJDOT [Damodaran et al., 2018, Fatras et al., 2021] : Deep learning JDOT.
- [Montesuma and Mboula, 2021]: Multi-source DA by mapping to Wass. Bary.
- [Gnassounou et al., 2023]: Convolutional Monge Mapping for Multi-source DA.

Domain adaptation with optimal transport



Extensions and related works

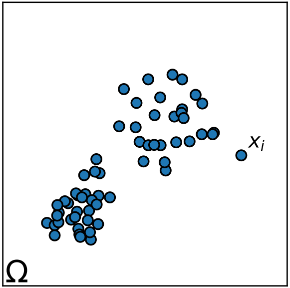
- JDOT [Courty et al., 2017b] : Joint OT and target predictor estimation.
- [Shen et al., 2018] : Wasserstein Distance Guided Representation Learning.
- DeepJDOT [Damodaran et al., 2018, Fatras et al., 2021] : Deep learning JDOT.
- [Montesuma and Mboula, 2021]: Multi-source DA by mapping to Wass. Bary.
- [Gnassounou et al., 2023]: Convolutional Monge Mapping for Multi-source DA.

Outline

Discrete distributions: Empirical vs Histogram

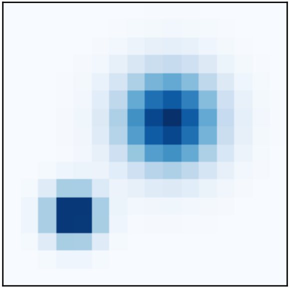
Discrete measure: $\mu = \sum_{i=1}^{n} a_i \delta_{\mathbf{x}_i}, \quad \mathbf{x}_i \in \Omega, \quad \sum_{i=1}^{n} a_i = 1$

Lagrangian (point clouds)

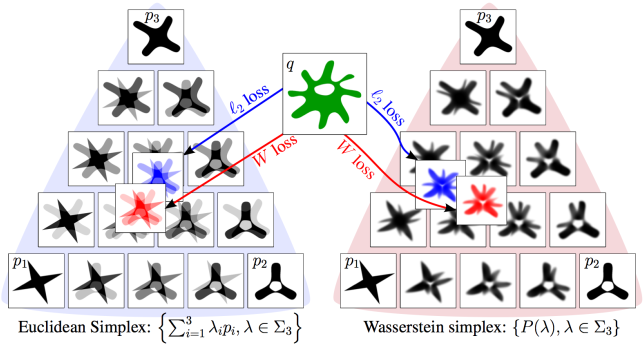


- Constant weight: $a_i = \frac{1}{n}$
- Quotient space: Ω^n, Σ_n

Eulerian (histograms)



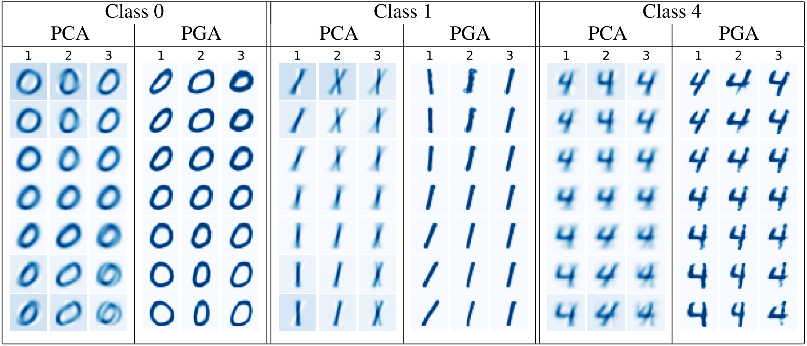
- Fixed positions $\mathbf{x}_i$ e.g. grid
- Convex polytope Σ_n (simplex): $\{(a_i)_i \geq 0; \sum_i a_i = 1\}$



Unsupervised learning on histogram data

- DL with Wasserstein distance [Sandler and Lindenbaum, 2011, Rolet et al., 2016]
- Nonlinear DL with Wasserstein barycenter [Schmitz et al., 2017]
- Geodesic PCA in Wasserstein space [Seguy and Cuturi, 2015, Bigot et al., 2017].
- Approximation using Wasserstein embedding [Courty et al., 2017a].

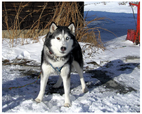
Dictionary Learning and Principal Geodesics Analysis



Unsupervised learning on histogram data

- DL with Wasserstein distance [Sandler and Lindenbaum, 2011, Rolet et al., 2016]
- Nonlinear DL with Wasserstein barycenter [Schmitz et al., 2017]
- Geodesic PCA in Wasserstein space [Seguy and Cuturi, 2015, Bigot et al., 2017].
- Approximation using Wasserstein embedding [Courty et al., 2017a].

Multi-label learning with Wasserstein Loss



Siberian husky

Eskimo dog

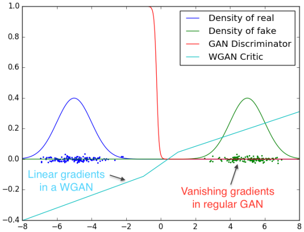
Flickr : street, parade, dragon
Prediction : people, protest, parade

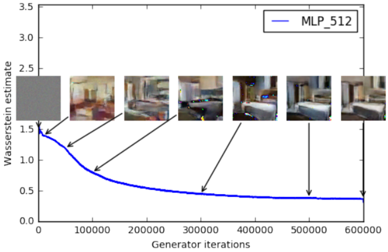
Flickr : water, boat, ref ection, sun-shine
Prediction : water, river, lake, summer;

Learning with a Wasserstein Loss [Frogner et al., 2015]

$$\min_{f} \quad \sum_{k=1}^{N} W_1^1(f(\mathbf{x}_i), \mathbf{l}_i)$$

- Empirical loss minimization with Wasserstein loss.
- Multi-label prediction (labels $\mathbf{l}$ seen as histograms, f output softmax).
- Cost between labels can encode semantic similarity between classes.
- Good performances in image tagging.





Wasserstein GAN [Arjovsky et al., 2017]

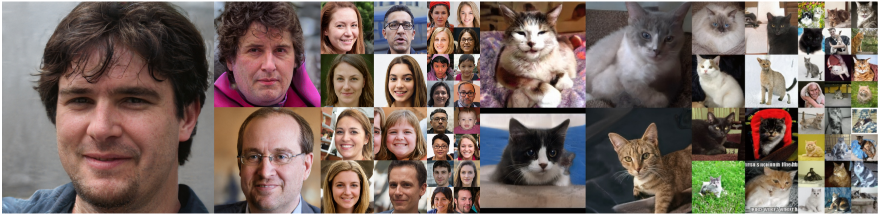
$$\min_{G} \quad W_1^1(G\#\mu_z, \mu_d), \quad \text{s.t. } \mathbf{z} \sim \mathcal{N}(0, \mathbf{I}) \tag{2}$$

- Minimizes the distance between the true μ_d and generated data $G\#\mu_z$.
- Better convergence in practice than classical GANs [Goodfellow et al., 2014].
- Wasserstein in the dual (separable w.r.t. the samples).

$$\min_{G} \sup_{\phi \in \text{Lip}^1} \quad \mathbb{E}_{\mathbf{x} \sim \mu_d}[\phi(\mathbf{x})] - \mathbb{E}_{\mathbf{z} \sim \mu_z}[\phi(G(\mathbf{z}))]$$

- Lipschitzness constrained or penalized [Gulrajani et al., 2017].
- State of the art for image generation with [Karras et al., 2019] (before diffusion).

Wasserstein Generative Adversarial Networks (WGAN)



Wasserstein GAN [Arjovsky et al., 2017]

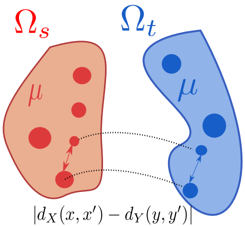
$$\min_G \quad W_1^1(G\#\mu_z, \mu_d), \quad \text{s.t. } \mathbf{z} \sim \mathcal{N}(0, \mathbf{I}) \tag{2}$$

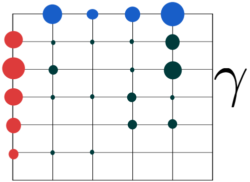
- Minimizes the distance between the true μ_d and generated data $G\#\mu_z$.
- Better convergence in practice than classical GANs [Goodfellow et al., 2014].
- Wasserstein in the dual (separable w.r.t. the samples).

$$\min_G \sup_{\phi \in \text{Lip}^1} \quad \mathbb{E}_{\mathbf{x}\sim\mu_d}[\phi(\mathbf{x})] - \mathbb{E}_{\mathbf{z}\sim\mu_z}[\phi(G(\mathbf{z}))]$$

- Lipschitzness constrained or penalized [Gulrajani et al., 2017].
- State of the art for image generation with [Karras et al., 2019] (before diffusion).

Outline





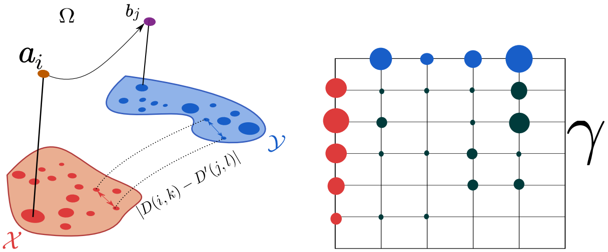
Inspired from Gabriel Peyré

GW for discrete distributions [Memoli, 2011]

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

with $\mu_s = \sum_i a_i \delta_{\mathbf{x}_i^s}$ and $\mu_t = \sum_j b_j \delta_{x_j^t}$ 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.
- Entropy regularized GW proposed in [Peyré et al., 2016].
- Fused GW interpolates between Wass. and GW [Vayer et al., 2018].



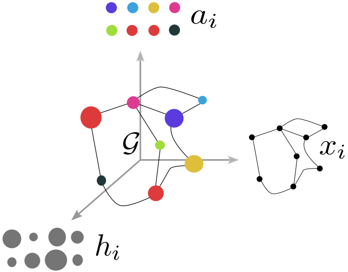
FGW for discrete distributions [Vayer et al., 2018]

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

with $\mu_s = \sum_i a_i \delta_{\mathbf{x}_i^s}$ and $\mu_t = \sum_j b_j \delta_{x_j^t}$ 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.
- Entropy regularized GW proposed in [Peyré et al., 2016].
- Fused GW interpolates between Wass. and GW [Vayer et al., 2018].

Gromov-Wasserstein between graphs



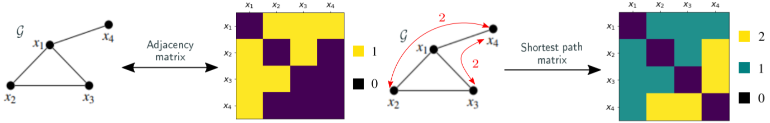
$$\mu = \sum_i h_i \delta_{(x_i, a_i)}$$

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

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

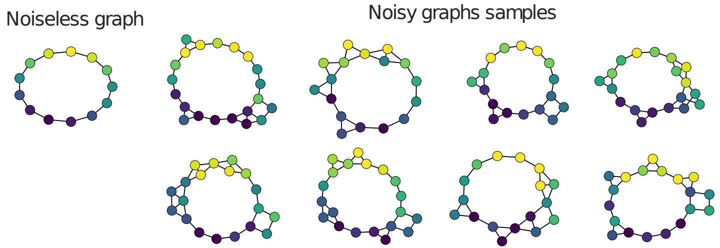
Graph as a distribution $(\boldsymbol{D}, \mathbf{F}, \boldsymbol{h})$

- The positions x_i are implicit and represented as the pairwise matrix $\boldsymbol{D}$.
- Possible choices for $\boldsymbol{D}$: Adjacency matrix, Laplacian, Shortest path, ...

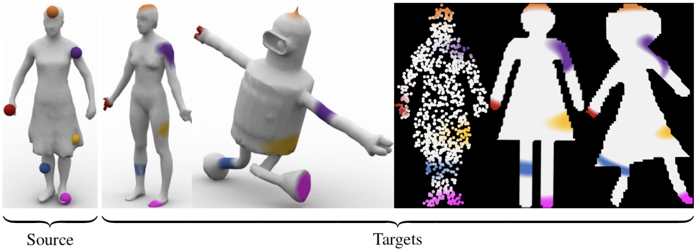


- The node features can be compared between graphs and stored in $\mathbf{F}$.
- h_i are the masses on the nodes of the graphs (uniform by default).

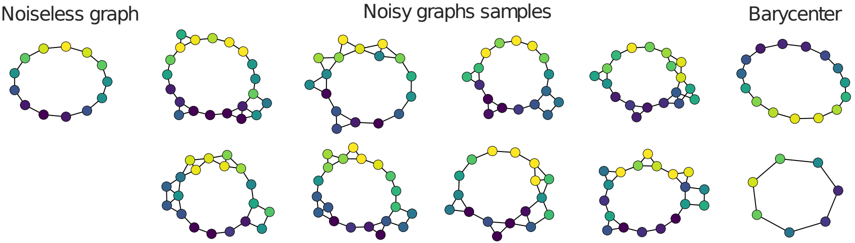
Barycenter/averaging of labeled graphs [Vayer et al., 2018]



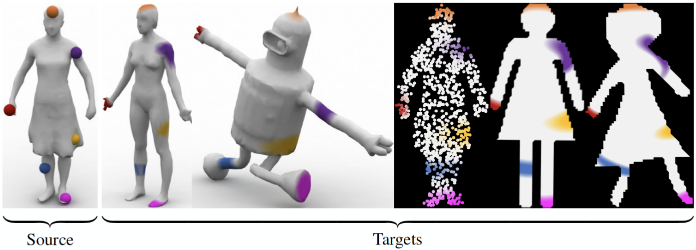
Shape matching between surfaces [Solomon et al., 2016, Thual et al., 2022]



Barycenter/averaging of labeled graphs [Vayer et al., 2018]

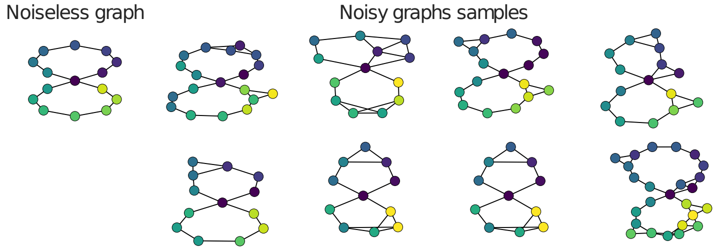


Shape matching between surfaces [Solomon et al., 2016, Thual et al., 2022]

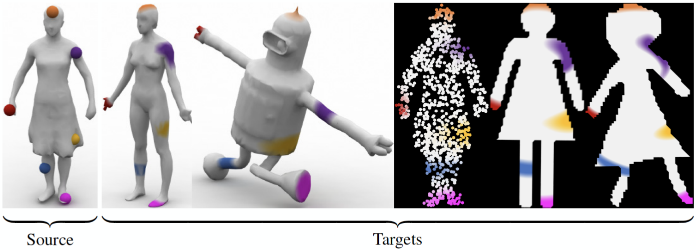


Applications of (F)GW

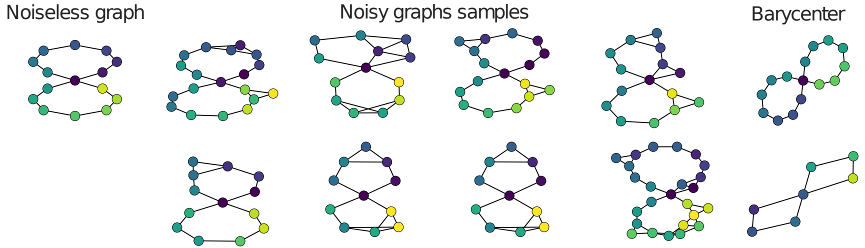
Barycenter/averaging of labeled graphs [Vayer et al., 2018]



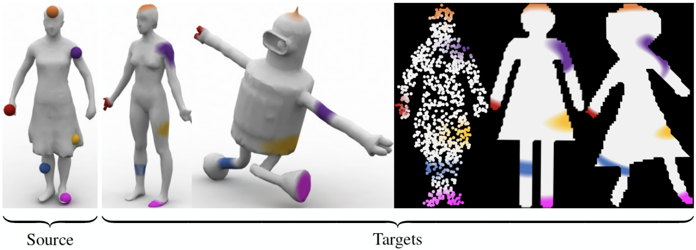
Shape matching between surfaces [Solomon et al., 2016, Thual et al., 2022]



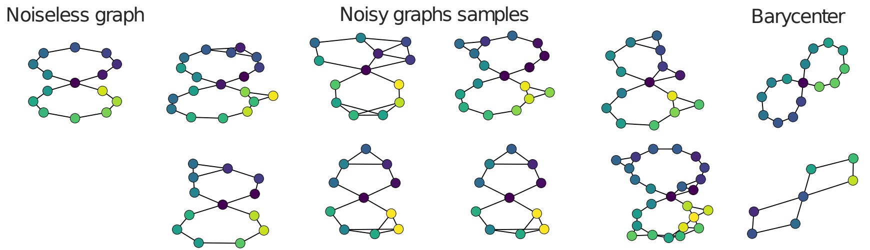
Barycenter/averaging of labeled graphs [Vayer et al., 2018]



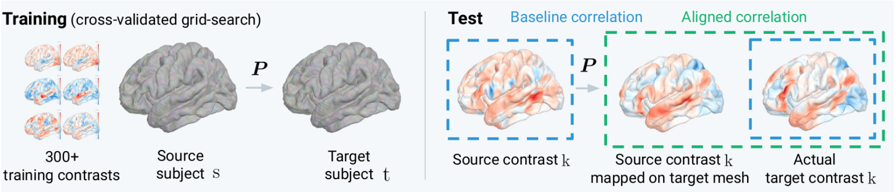
Shape matching between surfaces [Solomon et al., 2016, Thual et al., 2022]



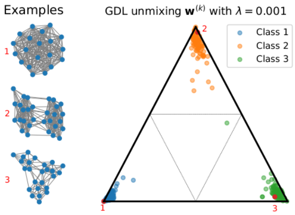
Barycenter/averaging of labeled graphs [Vayer et al., 2018]

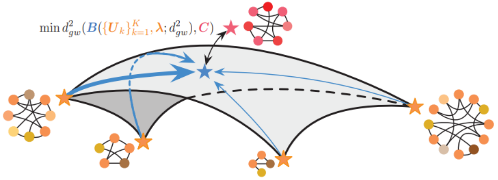


Shape matching between surfaces [Solomon et al., 2016, Thual et al., 2022]



Graph Dictionary Learning





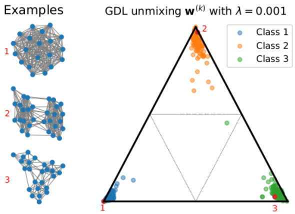
Representation learning for graphs

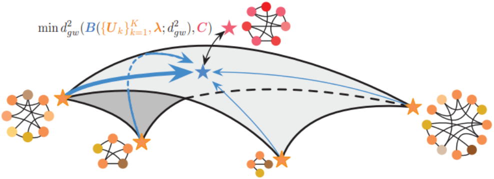
- Learn a dictionary $\{\overline{\mathbf{C}_i}\}_i$ of graph templates to describe a continuous manifold.
- The representation is learned by minimizing the (F)GW distance between the graph reconstruction from the embedding in the dictionary.
- Online Graph Dictionary learning : Linear model [Vincent-Cuaz et al., 2021].

$$\widehat{\mathbf{C}} = \sum_i w_i \overline{\mathbf{C}_i}$$

- GW Factorization : Nonlinear (GW barycenter) model [Xu, 2020].
- Dictionary for structured prediction with GW bary. [Brogat-Motte et al., 2022].

Graph Dictionary Learning





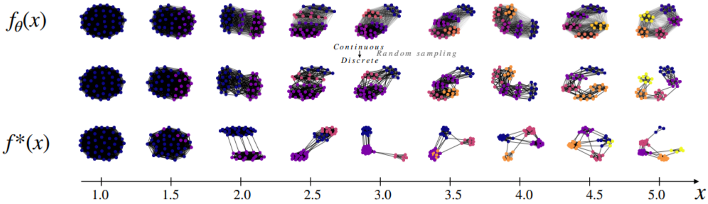
Representation learning for graphs

- Learn a dictionary $\{\overline{\mathbf{C}_i}\}_i$ of graph templates to describe a continuous manifold.
- The representation is learned by minimizing the (F)GW distance between the graph reconstruction from the embedding in the dictionary.
- Online Graph Dictionary learning : Linear model [Vincent-Cuaz et al., 2021].
- GW Factorization : Nonlinear (GW barycenter) model [Xu, 2020].

$$\widehat{\mathbf{C}} = \arg\min_{\mathbf{C}} \sum_i w_i GW(\mathbf{C}, \overline{\mathbf{C}_i})$$

- Dictionary for structured prediction with GW bary. [Brogat-Motte et al., 2022].

Graph Dictionary Learning

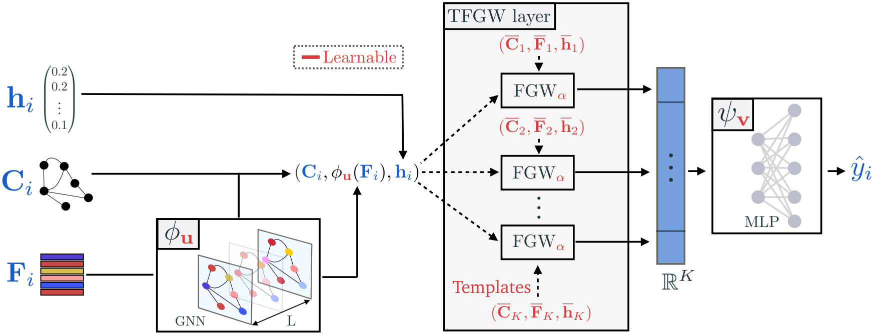


Representation learning for graphs

- Learn a dictionary $\{\overline{\mathbf{C}_i}\}_i$ of graph templates to describe a continuous manifold.
- The representation is learned by minimizing the (F)GW distance between the graph reconstruction from the embedding in the dictionary.
- Online Graph Dictionary learning : Linear model [Vincent-Cuaz et al., 2021].
- GW Factorization : Nonlinear (GW barycenter) model [Xu, 2020].
- Dictionary for structured prediction with GW bary. [Brogat-Motte et al., 2022].

$$f(\mathbf{x}) = \widehat{\mathbf{C}}(\mathbf{x}) = \arg\min_{\mathbf{C}} \sum_i w_i(\mathbf{x}) GW(\mathbf{C}, \overline{\mathbf{C}_i})$$

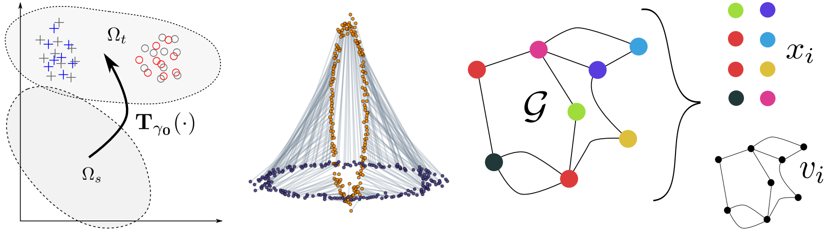
FGW for a pooling layer in GNN



Template based FGW layer (TFGW) [Vincent-Cuaz et al., 2022]

- Principle: represent a graph through its distances to learned templates.
- Learnable parameters are illustrated in red above.
- New end-to-end GNN models for graph-level tasks.
- Sate-of-the-art (still!) on graph classification (1×#1, 3×#2 on paperwithcode).

Outline



Optimal Transport for Machine Learning

- Very dynamic community (NeurIPS OTML workshop every 2 years).
- Distributions are everywhere, and geometry can help.
- OT can be used to map, find correspondances and measure similarity.
- Many extensions: sliced, unbalanced, multi-marginal, ...

What about the next ten years ?

- OT is here to stay, it is a tool that can be adapted/relaxed.
- We need better solvers (faster, more scalable, more robust).

Collaborators

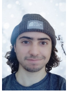
N. Courty A. Rakotomamonjy D. Tuia A. Habrard M. Perrot M. Ducoffe

M. Cuturi K. Lounici A. Férrari 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

+ H. Van Assel, Th. Gnassounou, A. Gramfort

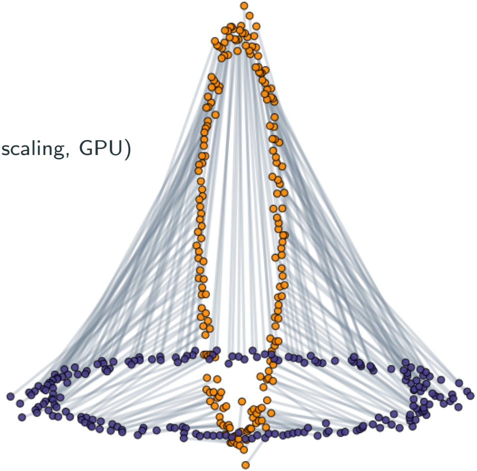
Thank you

Python code available on GitHub:

Python code available on GitHub:

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

- OT LP solver, Sinkhorn (stabilized, $\epsilon-$scaling, GPU)
- Domain adaptation with OT.
- Barycenters, Wasserstein unmixing.
- Wasserstein Discriminant Analysis.



Tutorial on OT for ML:

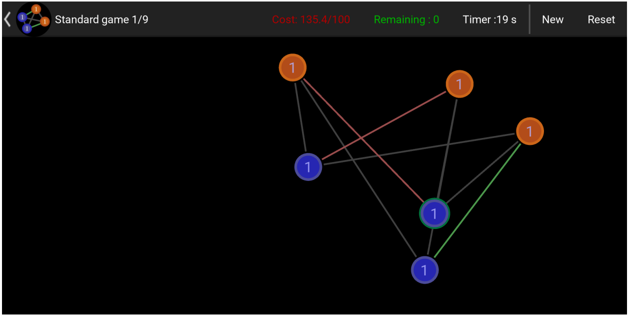
`http://tinyurl.com/otml-isbi`

Papers available on my website:

`https://remi.flamary.com/`

OTGame (OT Puzzle game on android)

OTGame



https://play.google.com/store/apps/details?id=com.flamary.otgame

[Arjovsky et al., 2017] Arjovsky, M., Chintala, S., and Bottou, L. (2017).
Wasserstein generative adversarial networks.
In *Proceedings of the 34th International Conference on Machine Learning*, volume 70, pages 214–223, Sydney, Australia.

[Bigot et al., 2017] Bigot, J., Gouet, R., Klein, T., López, A., et al. (2017).
Geodesic pca in the wasserstein space by convex pca.
In *Annales de l'Institut Henri Poincaré, Probabilités et Statistiques*, volume 53, pages 1–26. Institut Henri Poincaré.

[Brogat-Motte et al., 2022] 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)*.

[Courty et al., 2017a] Courty, N., Flamary, R., and Ducoffe, M. (2017a).
Learning wasserstein embeddings.

References ii

[Courty et al., 2017b] Courty, N., Flamary, R., Habrard, A., and Rakotomamonjy, A. (2017b).
Joint distribution optimal transportation for domain adaptation.
In *Neural Information Processing Systems (NIPS)*.

[Courty et al., 2014] 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)*.

[Courty et al., 2016] Courty, N., Flamary, R., Tuia, D., and Rakotomamonjy, A. (2016).
Optimal transport for domain adaptation.
Pattern Analysis and Machine Intelligence, IEEE Transactions on.

[Cuturi, 2013] Cuturi, M. (2013).
Sinkhorn distances: Lightspeed computation of optimal transport.
In *NIPS*, pages 2292–2300.

References iii

[Damodaran et al., 2018] Damodaran, B. B., Kellenberger, B., Flamary, R., Tuia, D., and Courty, N. (2018).

Deepjdot: Deep joint distribution optimal transport for unsupervised domain adaptation.

[Fatras et al., 2021] Fatras, K., Séjourné, T., Courty, N., and Flamary, R. (2021).

Unbalanced minibatch optimal transport; applications to domain adaptation.

In *International Conference on Machine Learning (ICML)*.

[Ferradans et al., 2014] Ferradans, S., Papadakis, N., Peyré, G., and Aujol, J.-F. (2014).

Regularized discrete optimal transport.

SIAM Journal on Imaging Sciences, 7(3).

[Flamary et al., 2019] Flamary, R., Lounici, K., and Ferrari, A. (2019).

Concentration bounds for linear monge mapping estimation and optimal transport domain adaptation.

arXiv preprint arXiv:1905.10155.

[Frogner et al., 2015] Frogner, C., Zhang, C., Mobahi, H., Araya, M., and Poggio, T. A. (2015).

Learning with a wasserstein loss.

In *Advances in Neural Information Processing Systems*, pages 2053–2061.

[Gnassounou et al., 2023] Gnassounou, T., Flamary, R., and Gramfort, A. (2023).

Convolutional monge mapping normalization for learning on biosignals.

In *Neural Information Processing Systems*.

[Goodfellow et al., 2014] Goodfellow, I., Pouget-Abadie, J., Mirza, M., Xu, B., Warde-Farley, D., Ozair, S., Courville, A., and Bengio, Y. (2014).

Generative adversarial nets.

In *Advances in neural information processing systems*, pages 2672–2680.

[Gulrajani et al., 2017] Gulrajani, I., Ahmed, F., Arjovsky, M., Dumoulin, V., and Courville, A. C. (2017).

Improved training of wasserstein gans.

In *Advances in Neural Information Processing Systems*, pages 5769–5779.

[Kantorovich, 1942] Kantorovich, L. (1942).
On the translocation of masses.
C.R. (Doklady) Acad. Sci. URSS (N.S.), 37:199–201.

[Karras et al., 2019] Karras, T., Laine, S., and Aila, T. (2019).
A style-based generator architecture for generative adversarial networks.
In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 4401–4410.

[Korotin et al., 2019] Korotin, A., Egiazarian, V., Asadulaev, A., Safin, A., and Burnaev, E. (2019).
Wasserstein-2 generative networks.
arXiv preprint arXiv:1909.13082.

[Makkuva et al., 2020] Makkuva, A., Taghvaei, A., Oh, S., and Lee, J. (2020).
Optimal transport mapping via input convex neural networks.
In *International Conference on Machine Learning*, pages 6672–6681. PMLR.

[Memoli, 2011] Memoli, F. (2011).
Gromov wasserstein distances and the metric approach to object matching.
Foundations of Computational Mathematics, pages 1–71.

[Monge, 1781] Monge, G. (1781).
Mémoire sur la théorie des déblais et des remblais.
De l'Imprimerie Royale.

[Montesuma and Mboula, 2021] Montesuma, E. F. and Mboula, F. M. N. (2021).
Wasserstein barycenter for multi-source domain adaptation.
In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 16785–16793.

[Mroueh, 2019] Mroueh, Y. (2019).
Wasserstein style transfer.
arXiv preprint arXiv:1905.12828.

[Peyré et al., 2016] Peyré, G., Cuturi, M., and Solomon, J. (2016).
Gromov-wasserstein averaging of kernel and distance matrices.
In *ICML*, pages 2664–2672.

[Pooladian and Niles-Weed, 2021] Pooladian, A.-A. and Niles-Weed, J. (2021).
Entropic estimation of optimal transport maps.
arXiv preprint arXiv:2109.12004.

[Rolet et al., 2016] Rolet, A., Cuturi, M., and Peyré, G. (2016).
Fast dictionary learning with a smoothed wasserstein loss.
In *Proceedings of the 19th International Conference on Artificial Intelligence and Statistics*, pages 630–638.

[Rubner et al., 2000] Rubner, Y., Tomasi, C., and Guibas, L. J. (2000).
The earth mover's distance as a metric for image retrieval.
International journal of computer vision, 40(2):99–121.

[Sandler and Lindenbaum, 2011] Sandler, R. and Lindenbaum, M. (2011).
Nonnegative matrix factorization with earth mover's distance metric for image analysis.
IEEE Transactions on Pattern Analysis and Machine Intelligence, 33(8):1590–1602.

[Schmitz et al., 2017] Schmitz, M. A., Heitz, M., Bonneel, N., Mboula, F. M. N., Coeurjolly, D., Cuturi, M., Peyré, G., and Starck, J.-L. (2017).
Wasserstein dictionary learning: Optimal transport-based unsupervised non-linear dictionary learning.
arXiv preprint arXiv:1708.01955.

[Seguy et al., 2017] Seguy, V., Bhushan Damodaran, B., Flamary, R., Courty, N., Rolet, A., and Blondel, M. (2017).
Large-scale optimal transport and mapping estimation.

[Seguy and Cuturi, 2015] Seguy, V. and Cuturi, M. (2015).
Principal geodesic analysis for probability measures under the optimal transport metric.
In *Advances in Neural Information Processing Systems*, pages 3312–3320.

[Shen et al., 2018] Shen, J., Qu, Y., Zhang, W., and Yu, Y. (2018).
Wasserstein distance guided representation learning for domain adaptation.
In *AAAI Conference on Artificial Intelligence*.

[Solomon et al., 2016] 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.

[Thual et al., 2022] Thual, A., Tran, H., Zemskova, T., Courty, N., Flamary, R., Dehaene, S., and Thirion, B. (2022).
Aligning individual brains with fused unbalanced gromov-wasserstein.
In *Neural Information Processing Systems (NeurIPS)*.

[Vayer et al., 2018] Vayer, T., Chapel, L., Flamary, R., Tavenard, R., and Courty, N. (2018).
Fused gromov-wasserstein distance for structured objects: theoretical foundations and mathematical properties.

[Vincent-Cuaz et al., 2022] Vincent-Cuaz, C., Flamary, R., Corneli, M., Vayer, T., and Courty, N. (2022).

Template based graph neural network with optimal transport distances.

In *Neural Information Processing Systems (NeurIPS)*.

[Vincent-Cuaz et al., 2021] 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, 2020] Xu, H. (2020).

Gromov-wasserstein factorization models for graph clustering.

In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 34, pages 6478–6485.