Optimal Transport for Machine learning

Domain Adaptation and structured data

R. Flamary - Lagrange, OCA, CNRS, Université Côte d'Azur

Meeting ANR MAGA, December 2018, Nancy

Collaborators


N. Courty


A. Rakotomamonjy


D. Tuia


A. Habrard


M. Perrot


V. Seguy


B. B. Damodaran


T. Vayer

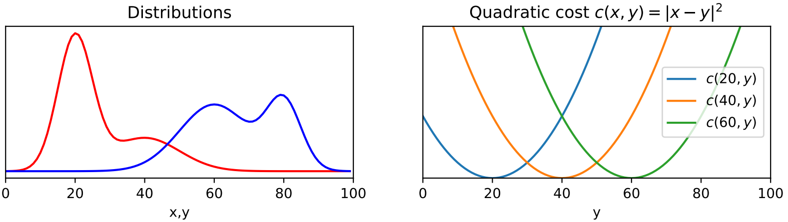

L. Chapel


R. Tavenard

+ ANR OATMIL project members

Introduction

Optimal transport (Monge formulation)

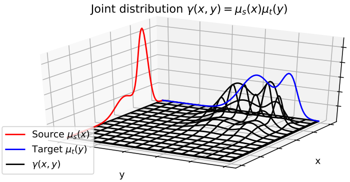


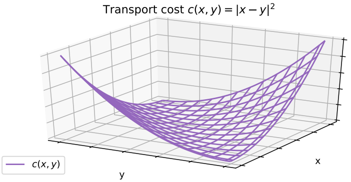
- Probability measures μ_s and μ_t on and a cost function $c : \Omega_s \times \Omega_t \rightarrow \mathbb{R}^+$.
- The Monge formulation [Monge, 1781] aim at finding a mapping $T : \Omega_s \rightarrow \Omega_t$

$$\inf_{T\#\mu_s=\mu_t} \quad \int_{\Omega_s} c(\mathbf{x}, T(\mathbf{x}))\mu_s(\mathbf{x})d\mathbf{x} \tag{1}$$

- Non-convex optimization problem, mapping does not exist in the general case.

Optimal transport (Kantorovich formulation)



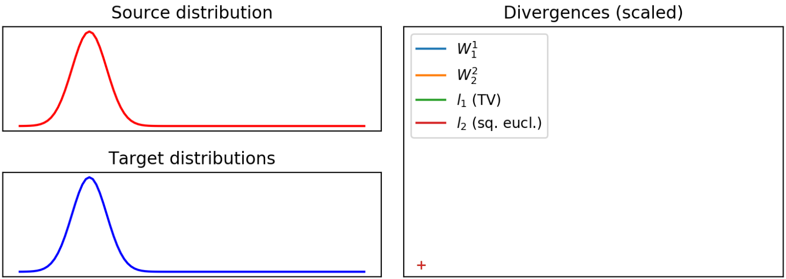


- The Kantorovich formulation [Kantorovich, 1942] seeks for a probabilistic coupling $\boldsymbol{\pi} \in \mathcal{P}(\Omega_s \times \Omega_t)$ between Ω_s and Ω_t:

$$\boldsymbol{\pi}_0 = \underset{\boldsymbol{\pi}}{\operatorname{argmin}} \int_{\Omega_s \times \Omega_t} c(\mathbf{x}, \mathbf{y}) \boldsymbol{\pi}(\mathbf{x}, \mathbf{y}) d\mathbf{x} d\mathbf{y}, \tag{2}$$

$$\text{s.t.} \quad \boldsymbol{\pi} \in \Pi = \left\{ \boldsymbol{\pi} \geq 0, \int_{\Omega_t} \boldsymbol{\pi}(\mathbf{x}, \mathbf{y}) d\mathbf{y} = \mu_s, \int_{\Omega_s} \boldsymbol{\pi}(\mathbf{x}, \mathbf{y}) d\mathbf{x} = \mu_t \right\}$$

- $\boldsymbol{\pi}$ is a joint probability measure with marginals μ_s and μ_t.
- Linear Program that always have a solution.

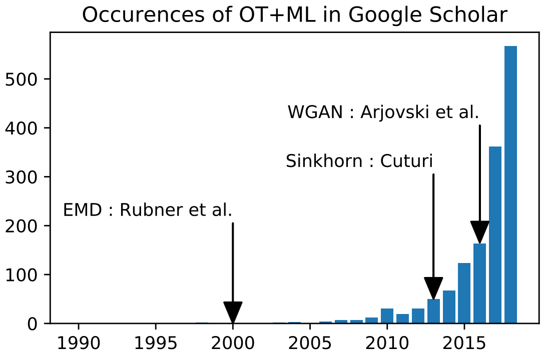


Wasserstein distance

$$W_p^p(\mu_s, \mu_t) = \min_{\boldsymbol{\pi} \in \Pi} \int_{\Omega_s \times \Omega_t} c(\mathbf{x}, \mathbf{y}) \boldsymbol{\pi}(\mathbf{x}, \mathbf{y}) d\mathbf{x} d\mathbf{y} = E_{(\mathbf{x}, \mathbf{y}) \sim \boldsymbol{\pi}}[c(\mathbf{x}, \mathbf{y})] \tag{3}$$

where $c(\mathbf{x}, \mathbf{y}) = \|\mathbf{x} - \mathbf{y}\|^p$ is the ground metric.

- A.K.A. Earth Mover's Distance (W_1^1) [Rubner et al., 2000].
- Do not need the distribution to have overlapping support.
- Subgradients can be computed with the dual variables of the LP.
- Works for continuous and discrete distributions (histograms, empirical).



Short history of OT for ML

- Recently introduced to ML (well known in image processing since 2000s).
- Computationnal OT allow numerous applications (regularization).
- Deep learning boost (numerical optimization and GAN).

Table of content

Optimal transport for domain adaptation

Supervised learning

Amazon

Traditional supervised learning

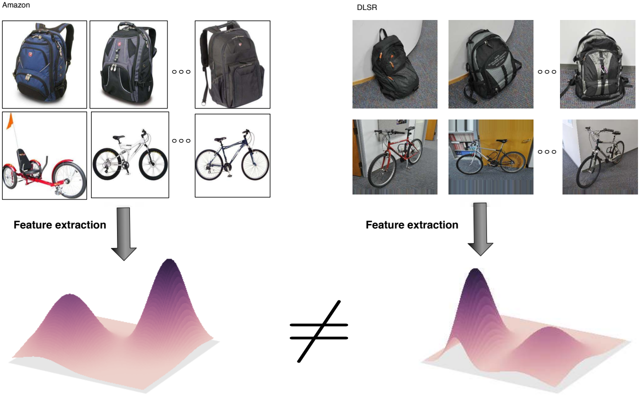
- We want to learn predictor such that $y \approx f(\mathbf{x})$.
- Actual $\mathcal{P}(X, Y)$ unknown.
- We have access to training dataset $(\mathbf{x}_i, y_i)_{i=1,\dots,n}$ $(\widehat{\mathcal{P}}(X, Y))$.
- We choose a loss function $\mathcal{L}(y, f(\mathbf{x}))$ that measure the discrepancy.

Empirical risk minimization
We week for a predictor f minimizing

$$\min_{f} \left\{ \mathop{\mathbb{E}}_{(\mathbf{x},y)\sim\widehat{\mathcal{P}}} \mathcal{L}(y, f(\mathbf{x})) = \sum_{j} \mathcal{L}(y_j, f(\mathbf{x}_j)) \right\} \tag{4}$$

- Well known generalization results for predicting on new data.
- Loss is usually $\mathcal{L}(y, f(\mathbf{x})) = (y - f(\mathbf{x}))^2$ for least square regression and is $\mathcal{L}(y, f(\mathbf{x})) = \max(0, 1 - yf(\mathbf{x}))^2$ for squared Hinge loss SVM.

Domain Adaptation problem

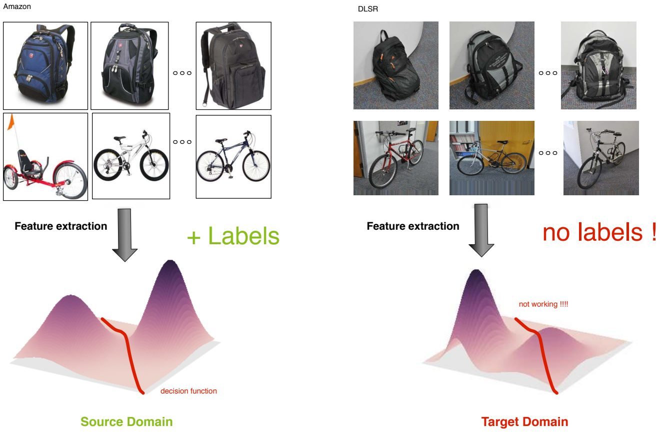


Probability Distribution Functions over the domains

Our context

- Classification problem with data coming from different sources (domains).
- Distributions are different but related.

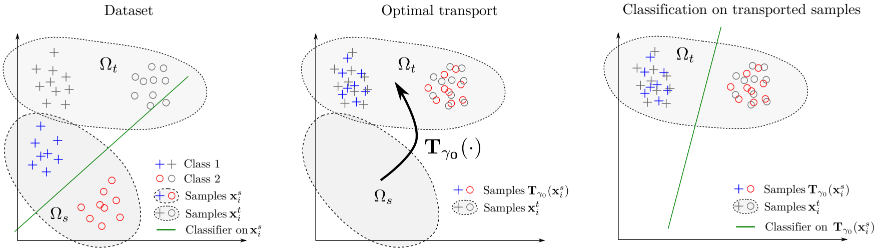
Unsupervised domain adaptation problem



Problems

- Labels only available in the **source domain**, and classification is conducted in the **target domain**.
- Classifier trained on the source domain data performs badly in the target domain

Optimal transport for domain adaptation



Assumptions

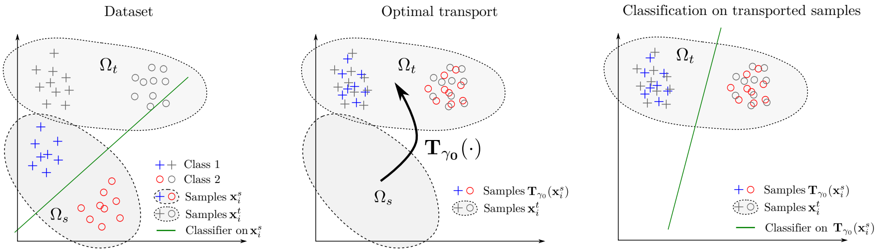
- There exist a transport in the feature space $\mathbf{T}$ between the two domains.
- The transport preserves the conditional distributions:

$$P_s(y|\mathbf{x}_s) = P_t(y|\mathbf{T}(\mathbf{x}_s)).$$

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

1. Estimate optimal transport between distributions.
2. Transport the training samples with barycentric mapping .
3. Learn a classifier on the transported training samples.

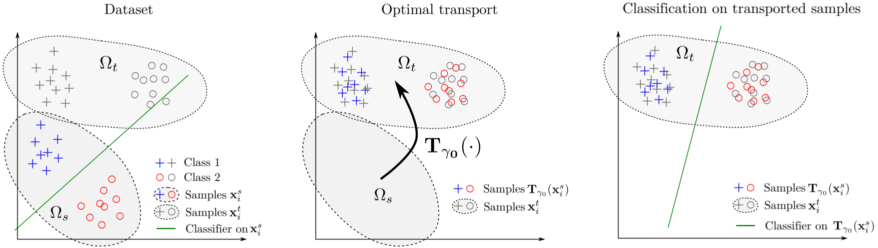
OT for domain adaptation : Step 1



Step 1 : Estimate optimal transport between distributions.

- Choose the ground metric (squared euclidean in our experiments).
- Using regularization allows
  - Large scale and regular OT with entropic regularization [Cuturi, 2013].
  - Class labels in the transport with group lasso [Courty et al., 2016].
- Efficient optimization based on Bregman projections [Benamou et al., 2015] and
  - Majoration minimization for non-convex group lasso.
  - Generalized Conditionnal gradient for general regularization (cvx. lasso, Laplacian).

OT for domain adaptation : Steps 2 & 3



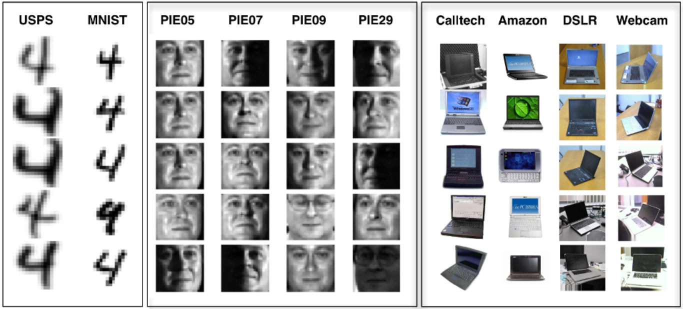
Step 2 : Transport the training samples onto the target distribution.

- The mass of each source sample is spread onto the target samples (line of $\boldsymbol{\pi}_0$).
- Transport using barycentric mapping [Ferradans et al., 2014a].
- The mapping can be estimated for out of sample prediction [Perrot et al., 2016, Seguy et al., 2017].

Step 3 : Learn a classifier on the transported training samples

- Transported sample keep their labels.
- Classic ML problem when samples are well transported.

Visual adaptation datasets



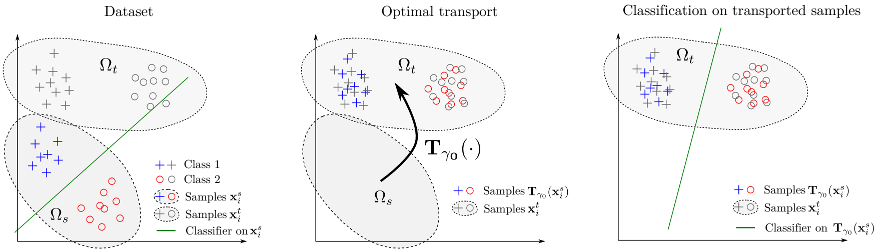
Datasets

- **Digit recognition**, MNIST VS USPS (10 classes, d=256, 2 dom.).
- **Face recognition**, PIE Dataset (68 classes, d=1024, 4 dom.).
- **Object recognition**, Caltech-Office dataset (10 classes, d=800/4096, 4 dom.).

Numerical experiments

- Comparison with state of the art on the 3 datasets.
- OT works very well on digits and object recognition.
- Works well on deep features adaptation and extension to semi-supervised DA.

Optimal transport for domain adaptation



Discussion

- Works very well in practice for large class of transformation [Courty et al., 2016].
- Can use estimated mapping [Perrot et al., 2016, Seguy et al., 2017].

But

- Model transformation only in the feature space.
- Requires the same class proportion between domains [Tuia et al., 2015].
- We estimate a $T : \mathbb{R}^d \rightarrow \mathbb{R}^d$ mapping for training a classifier $f : \mathbb{R}^d \rightarrow \mathbb{R}$.

Joint distribution and classifier estimation

Objectives of JDOT

- Model the transformation of labels (allow change of proportion/value).
- Learn an optimal target predictor with no labels on target samples.
- Approach theoretically justified.

Joint distributions and dataset

- Let $\Omega \in \mathbb{R}^d$ be a feature space of dimension d and $\mathcal{C}$ the set of labels.
- Let $\mathcal{P}_s(X,Y) \in \mathcal{P}(\Omega \times \mathcal{C})$ and $\mathcal{P}_t(X,Y) \in \mathcal{P}(\Omega \times \mathcal{C})$ the source and target joint distribution.
- We have access to an empirical sampling $\hat{\mathcal{P}}_s = \frac{1}{N_s}\sum_{i=1}^{N_s} \delta_{\mathbf{x}_i^s,\mathbf{y}_i^s}$ of the source distribution defined by $\mathbf{X}_s = \{\mathbf{x}_i^s\}_{i=1}^{N_s}$ and label information $\mathbf{Y}_s = \{\mathbf{y}_i^s\}_{i=1}^{N_s}$.
- but the target domain is defined only by an empirical distribution in the feature space with samples $\mathbf{X}_t = \{\mathbf{x}_i^t\}_{i=1}^{N_t}$.

Proxy joint distribution

- Let f be a $\Omega \rightarrow \mathcal{C}$ function from a given class of hypothesis $\mathcal{H}$.
- We define the following joint distribution that use f as a proxy of y

$$\mathcal{P}_t^f = (\mathbf{x}, f(\mathbf{x}))_{\mathbf{x}\sim\mu_t} \tag{5}$$

and its empirical counterpart $\hat{\mathcal{P}_t}^f = \frac{1}{N_t}\sum_{i=1}^{N_t} \delta_{\mathbf{x}_i^t, f(\mathbf{x}_i^t)}$.

Learning with JDOT
We propose to learn the predictor f that minimize :

$$\min_f \quad \left\{ W_1(\hat{\mathcal{P}}_s, \hat{\mathcal{P}_t}^f) = \inf_{\boldsymbol{\pi}\in\Pi} \sum_{ij} \mathcal{D}(\mathbf{x}_i^s, \mathbf{y}_i^s; \mathbf{x}_j^t, f(\mathbf{x}_j^t))\boldsymbol{\pi}_{ij} \right\} \tag{6}$$

- Π is the transport polytope.
- $\mathcal{D}(\mathbf{x}_i^s, \mathbf{y}_i^s; \mathbf{x}_j^t, f(\mathbf{x}_j^t)) = \alpha\|\mathbf{x}_i^s - \mathbf{x}_j^t\|^2 + \mathcal{L}(\mathbf{y}_i^s, f(\mathbf{x}_j^t))$ with $\alpha > 0$.
- We search for the predictor f that better align the joint distributions.
- Generalization bound show that expected risk on target is bounded by 6.

Optimization problem

$$\min_{f\in\mathcal{H},\boldsymbol{\pi}\in\Pi} \quad \sum_{i,j} \boldsymbol{\pi}_{i,j} \left(\alpha d(\mathbf{x}_i^s, \mathbf{x}_j^t) + \mathcal{L}(y_i^s, f(\mathbf{x}_j^t))\right) + \lambda\Omega(f) \tag{7}$$

Optimization procedure

- $\Omega(f)$ is a regularization for the predictor f
- We propose to use block coordinate descent (BCD)/Gauss Seidel.
- Provably converges to a stationary point of the problem.

$\boldsymbol{\pi}$ update for a fixed f

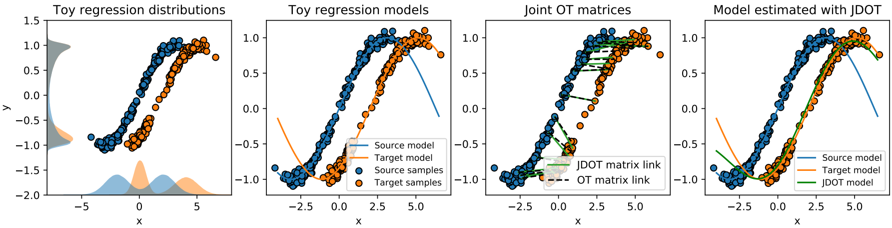
- Classical OT problem.
- Solved by network simplex.
- Regularized OT can be used (add a term to problem (7))

f update for a fixed $\boldsymbol{\pi}$

$$\min_{f\in\mathcal{H}} \quad \sum_{i,j} \boldsymbol{\pi}_{i,j}\mathcal{L}(y_i^s, f(\mathbf{x}_j^t)) + \lambda\Omega(f) \tag{8}$$

- Weighted loss from all source labels.
- $\boldsymbol{\pi}$ performs label propagation.

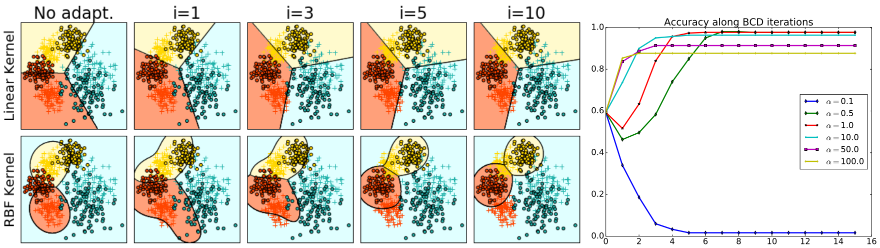
Regression with JDOT



Least square regression with quadratic regularization
For a fixed $\boldsymbol{\pi}$ the optimization problem is equivalent to

$$\min_{f \in \mathcal{H}} \quad \sum_j \frac{1}{n_t} \|\hat{y}_j - f(\mathbf{x}_j^t)\|^2 + \lambda \|f\|^2 \tag{9}$$

- $\hat{y}_j = n_t \sum_j \boldsymbol{\pi}_{i,j} y_i^s$ is a weighted average of the source target values.
- Note that this problem is linear instead of quadratic.
- Can use any solver (linear, kernel ridge, neural network).



Multiclass classification with Hinge loss

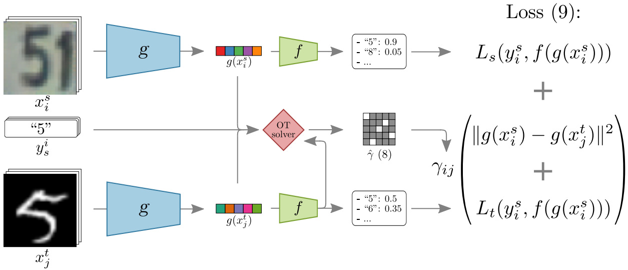
For a fixed $\boldsymbol{\pi}$ the optimization problem is equivalent to

$$\min_{f_k \in \mathcal{H}} \quad \sum_{j,k} \hat{P}_{j,k} \mathcal{L}(1, f_k(\mathbf{x}_j^t)) + (1 - \hat{P}_{j,k}) \mathcal{L}(-1, f_k(\mathbf{x}_j^t)) + \lambda \sum_k \|f_k\|^2 \qquad (10)$$

- $\hat{\mathbf{P}}$ is the class proportion matrix $\hat{\mathbf{P}} = \frac{1}{N_t} \boldsymbol{\pi}^\top \mathbf{P}^s$.
- $\mathbf{P}^s$ and $\mathbf{Y}^s$ are defined from the source data with One-vs-All strategy as

$$Y_{i,k}^s = \begin{cases} 1 & \text{if } y_i^s = k \\ -1 & \text{else} \end{cases}, \quad P_{i,k}^s = \begin{cases} 1 & \text{if } y_i^s = k \\ 0 & \text{else} \end{cases}$$

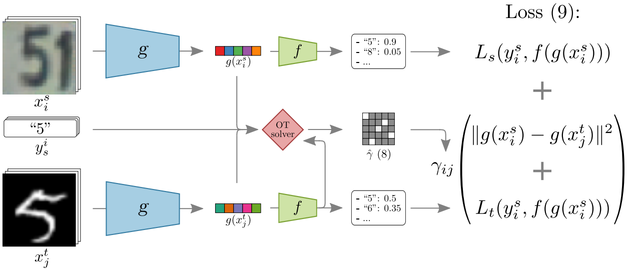
with $k \in 1, \cdots, K$ and K being the number of classes.



$$\min_{\boldsymbol{\pi}\in\Pi,f,g} \quad \frac{1}{n^s}\sum_i L_s\left(y_i^s, f(g(x_i^s))\right) + \sum_{i,j} \boldsymbol{\pi}_{ij}\left(\alpha\|g(x_i^s)-g(x_j^t)\|^2 + \lambda_t \mathcal{L}\left(y_i^s, f(g(x_j^t))\right)\right). \tag{11}$$

DeepJDOT [Damodaran et al., 2018]

- Learn simultaneously the embedding g and the classifier f.
- JDOT performed in the joint embedding/label space.



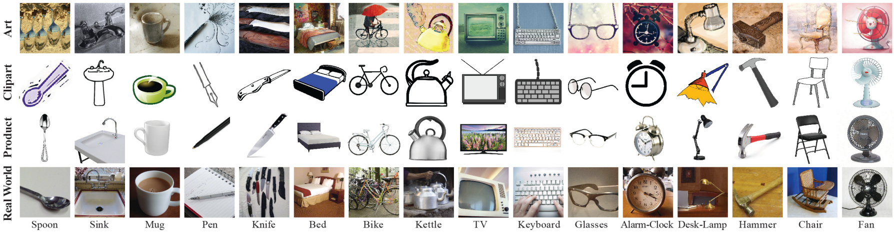
$$\min_{f,g} \mathbb{E}\left[\frac{1}{m}\sum_{i=1}^{m}\mathcal{L}\left(y_i^s, f(g(x_i^s)\right) + \min_{\boldsymbol{\pi}\in\Pi}\sum_{i,j}^{m}\boldsymbol{\pi}_{ij}\left(\alpha\|g(x_i^s) - g(x_j^t)\|^2 + \lambda_t\mathcal{L}\left(y_i^s, f(g(x_j^t))\right)\right)\right] \tag{11}$$

DeepJDOT [Damodaran et al., 2018]

- Learn simultaneously the embedding g and the classifier f.
- JDOT performed in the joint embedding/label space.
- Use minibatch to estimate OT and update g, f at each iterations.
- Scales to large datasets and estimate a representation for both domains.

DeepJDOT [Damodaran et al., 2018]

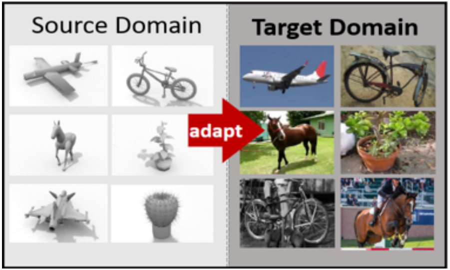
- Evaluation of DeepJDOT on visual classification tasks.
- Digit adaptation between MNIST, USPS, SVHN, MNIST-M.
- Home-office [Venkateswara et al., 2017] and VisDA 2017 [Peng et al., 2017] dataset.
- Ablation study : all terms are impportant.
- TSNE projections of embeddings (MNIST$\rightarrow$MNIST-M).



DeepJDOT [Damodaran et al., 2018]

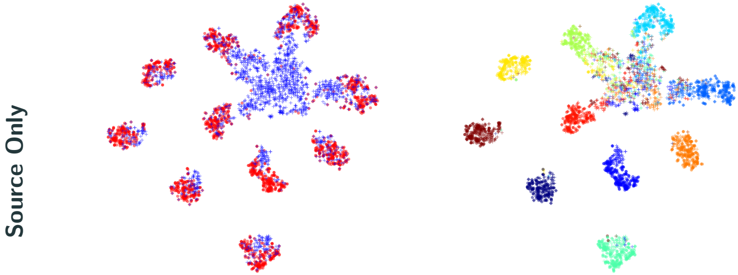
- Evaluation of DeepJDOT on visual classification tasks.
- Digit adaptation between MNIST, USPS, SVHN, MNIST-M.
- Home-office [Venkateswara et al., 2017] and VisDA 2017 [Peng et al., 2017] dataset.
- Ablation study : all terms are impportant.
- TSNE projections of embeddings (MNIST$\rightarrow$MNIST-M).

DeepJDOT in action



DeepJDOT [Damodaran et al., 2018]

- Evaluation of DeepJDOT on visual classification tasks.
- Digit adaptation between MNIST, USPS, SVHN, MNIST-M.
- Home-office [Venkateswara et al., 2017] and VisDA 2017 [Peng et al., 2017] dataset.
- Ablation study : all terms are impportant.
- TSNE projections of embeddings (MNIST$\rightarrow$MNIST-M).

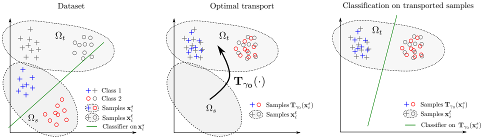


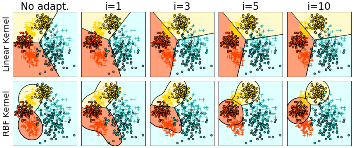
DeepJDOT [Damodaran et al., 2018]

- Evaluation of DeepJDOT on visual classification tasks.
- Digit adaptation between MNIST, USPS, SVHN, MNIST-M.
- Home-office [Venkateswara et al., 2017] and VisDA 2017 [Peng et al., 2017] dataset.
- Ablation study : all terms are impportant.
- TSNE projections of embeddings (MNIST→MNIST-M).

DeepJDOT [Damodaran et al., 2018]

- Evaluation of DeepJDOT on visual classification tasks.
- Digit adaptation between MNIST, USPS, SVHN, MNIST-M.
- Home-office [Venkateswara et al., 2017] and VisDA 2017 [Peng et al., 2017] dataset.
- Ablation study : all terms are impportant.
- TSNE projections of embeddings (MNIST$\rightarrow$MNIST-M).





Optimal transport for DA

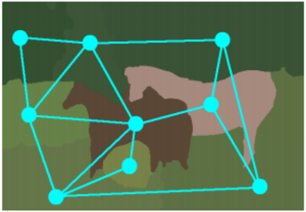
- Model transformation of the features.
- Conditional distribution preserved.
- Mapping between distributions.
- Learn classifier on the transported samples.

Joint distribution OT for DA

- Model transformation of the joint distribution.
- General framework for DA.
- Theoretical justification with generalization bound.

Optimal Transport on structured data

Structured data

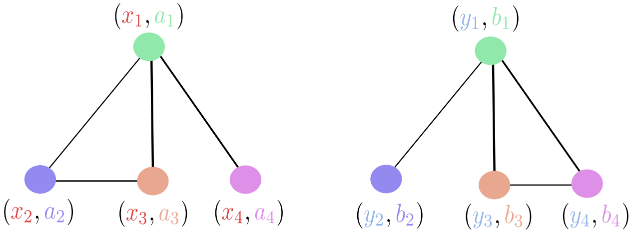


Structured data

- A structure data is viewed as a combination of features informations linked within each other by some structural information.
- Can be seen as a distribution on a joint feature/structure space.
- Example : labeled graph.

Meaningful distances on structured data

- Us both features (labels) and structure (graph).
- Allows for comparison, classification.
- Data science (statistics, means)



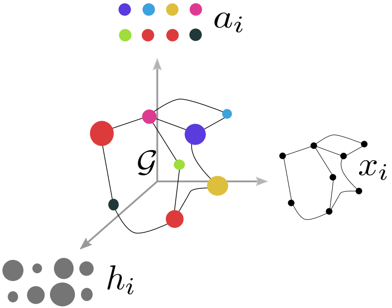
Structured data

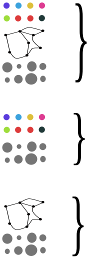
- A structure data is viewed as a combination of features informations linked within each other by some structural information.
- Can be seen as a distribution on a joint feature/structure space.
- Example : labeled graph.

Meaningful distances on structured data

- Us both features (labels) and structure (graph).
- Allows for comparison, classification.
- Data science (statistics, means)

Structured data as distributions





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

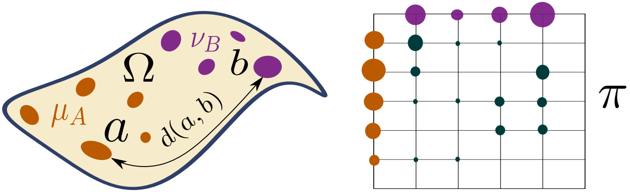
$$\mu_A = \textstyle\sum_i h_i \delta_{a_i}$$

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

Graph data representation

$$\mu = \sum_{i=1}^{n} h_i \delta_{(x_i a_i)}$$

- Nodes are weighted by their mass h_i.
- Features values a_i and b_j can be compared through the common metric
- But no common between the structure points x_i and y_j.

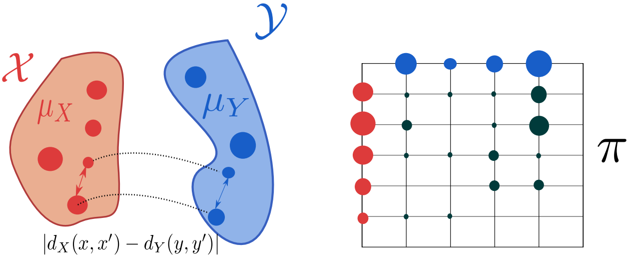


Wasserstein distance for structures data

$$\mathcal{W}_p(\mu_A, \mu_B) = \left(\min_{\pi \in \Pi(\mu_A, \mu_B)} \sum_{i,j} M_{i,j}^p \pi_{i,j} \right)^{\frac{1}{p}}$$

$\mu_A = \sum_i h_i \delta_{a_i}$ and $\mu_B = \sum_j g_j \delta_{b_j}$, $M_{i,j} = \|a_i - b_j\|$

- Wasserstein good for (empirical) distributions, samples as IID.
- OT can encode structure with OT Lp [Thorpe et al., 2017] by extending the feature space but requires the same ambient space.



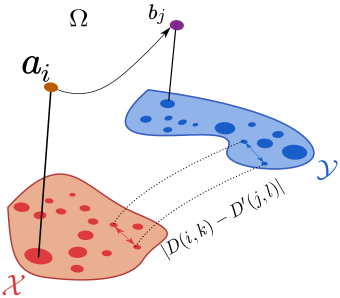
Inspired from Gabriel Peyré

GW for structured data [Memoli, 2011]

$$\mathcal{GW}_p(D, D', \mu_X, \mu_Y) = \left(\min_{\pi \in \Pi(\mu_s, \mu_t)} \sum_{i,j,k,l} |D_{i,k} - D'_{j,l}|^p \pi_{i,j}\, \pi_{k,l} \right)^{\frac{1}{p}}$$

$\mu_X = \sum_i h_i \delta_{x_i}$ and $\mu_Y = \sum_j g_j \delta_{y_j}$ and $D_{i,k} = \|x_i - x_k\|, D'_{j,l} = \|y_j - y_l\|$

- Distance over measures with no common ground space.
- Works well on graphs (using distances between nodes) but do not handle labels.
- Invariant to rotations and translation in either spaces.



Fused Gromov Wasserstein distance

$$\mathcal{FGW}_{p,q,\alpha}(D,D',\mu_s,\mu_t)=\left(\min_{\pi\in\Pi(\mu_s,\mu_t)}\sum_{i,j,k,l}\left((1-\alpha)M_{i,j}^q+\alpha|D_{i,k}-D'_{j,l}|^q\right)^p\pi_{i,j}\,\pi_{k,l}\right)^{\frac{1}{p}}$$

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

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

$$\mathcal{FGW}_{p,q,\alpha}(D, D', \mu_s, \mu_t) = \left(\min_{\pi \in \Pi(\mu_s, \mu_t)} \sum_{i,j,k,l} \left((1-\alpha) M_{i,j}^q + \alpha |D_{i,k} - D'_{j,l}|^q \right)^p \pi_{i,j}\, \pi_{k,l} \right)^{\frac{1}{p}}$$

Metric properties

- $\mathcal{FGW}$ 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.

Other properties for sontinuous distributions

- Interpolation between $\mathcal{W}$ $(\alpha = 0)$ and $\mathcal{GW}$ $(\alpha = 1)$ distances.
- Geodesic properties (constant speed, unicity).

FGW Properties (2)

$$\mathcal{FGW}_{p,q,\alpha}(D, D', \mu_s, \mu_t) = \left(\min_{\pi \in \Pi(\mu_s, \mu_t)} \sum_{i,j,k,l} \left((1-\alpha) M_{i,j}^q + \alpha |D_{i,k} - D'_{j,l}|^q \right)^p \pi_{i,j}\, \pi_{k,l} \right)^{\frac{1}{p}}$$

Bounds and convergence to finite samples

- The following inequalities hold:

$$\mathcal{FGW}(\mu_s, \mu_t) \geq (1-\alpha)\mathcal{W}(\mu_A, \mu_B)^q$$

$$\mathcal{FGW}(\mu_s, \mu_t) \geq \alpha\mathcal{GW}(\mu_X, \mu_Y)^q$$

- Bound when $\mathcal{X} = \mathcal{Y}$:

$$\mathcal{FGW}(\mu_s, \mu_t)^p \leq 2\mathcal{W}(\mu_s, \mu_t)^p$$

- Convergence of finite samples when $\mathcal{X} = \mathcal{Y}$ with $d = Dim(\mathcal{X}) + Dim(\Omega)$:

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

Computing FGW

$$\pi^* = \underset{\pi \in \Pi(\mu_s, \mu_t)}{\arg\min} \quad \mathsf{vec}(\pi)^T Q \mathsf{vec}(\pi) + \mathsf{vec}((1-\alpha)M)^T \mathsf{vec}(\pi) \tag{12}$$

where $Q = -2\alpha D' \otimes D$

Algorithmic resolution ($p = 1$)

- Problem is a non-convex Quadratic Program.
- We use Conditional gradient [Ferradans et al., 2014b] with network simplex solver.
- Convergence to a local minima [Lacoste-Julien, 2016].
- With entropic regularization, projected gradient descent [Peyré et al., 2016].

Computing FGW

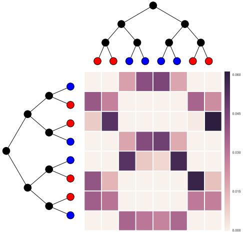
$$\pi^* = \underset{\pi \in \Pi(\mu_s, \mu_t)}{\arg\min} \quad \text{vec}(\pi)^T Q \text{vec}(\pi) + \text{vec}((1-\alpha)M)^T \text{vec}(\pi) \tag{12}$$

Algorithm 1 Conditional Gradient (CG) for FGW

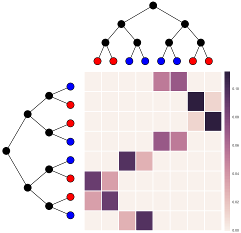
1: $\pi^{(0)} \leftarrow \mu_X \mu_Y^\top$
2: **for** $i = 1, \ldots,$ **do**
3: $\quad G \leftarrow$ Gradient from Eq. (12) *w.r.t.* $\pi^{(i-1)}$
4: $\quad \tilde{\pi}^{(i)} \leftarrow$ Solve OT with ground loss G
5: $\quad \tau^{(i)} \leftarrow$ Line-search for loss with $\tau \in (0,1)$
6: $\quad \pi^{(i)} \leftarrow (1-\tau^{(i)})\pi^{(i-1)} + \tau^{(i)}\tilde{\pi}^{(i)}$
7: **end for**

Algorithmic resolution ($p = 1$)

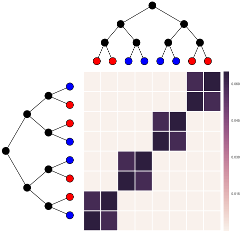
- Problem is a non-convex Quadratic Program.
- We use Conditional gradient [Ferradans et al., 2014b] with network simplex solver.
- Convergence to a local minima [Lacoste-Julien, 2016].
- With entropic regularization, projected gradient descent [Peyré et al., 2016].



(a) $\alpha = 0, \ W = 0$



(b) $0 < \alpha < 1, \ FGW \neq 0$



(c) $\alpha = 1, \ GW = 0$

FGW maps on toy tree

- Uniform weights on the leafs of the tree.
- Structure distance taken as shortest path on the tree.
- Only FGW can encode both features and structures.

Vector attributes	AIDS	BZR	COX2	CUNEIFORM	ENZYMES	PROTEIN	SYNTHETIC
FGW sp	99.44+/-0.47	85.12+/-4.15	77.23+/-4.86	**76.67+/-7.04**	71.00+/-6.76	74.55+/-2.74	**100.00+/-0.00**
FGW sp regul	-	**85.61+/-5.05**	77.66+/-4.17	-	70.17+/-6.81	74.64+/-2.99	-
FGW wsp	99.55+/-0.35	84.88+/-4.34	78.09+/-3.81	-	69.50+/-7.30	75.09+/-2.34	-
FGWDMM sp	-	84.39+/-5.48	76.81+/-4.30	-	61.67+/-7.19	75.00+/-2.59	-
FGWDMM wsp	-	83.17+/-5.05	78.30+/-3.53	-	59.17+/-6.55	75.09+/-3.03	-
HOPPER all cv	99.50+/-0.59	84.15+/-5.26	**79.57+/-3.46**	32.59+/-8.73	45.33+/-4.00	71.96+/-3.22	90.67+/-4.67
PROPA all cv	98.45+/-1.06	79.51+/-5.02	77.66+/-3.95	12.59+/-6.67	**71.67+/-5.63**	61.34+/-4.38	64.67+/-6.70
PSCN k=10	99.80+/-0.24	80.00+/-4.47	71.70+/-3.57	25.19+/-7.73	26.67+/-4.77	67.95+/-11.28	**100.00+/-0.00**
PSCN k=5	**99.85+/-0.23**	82.20+/-4.23	71.91+/-3.40	24.81+/-7.23	27.33+/-4.16	71.79+/-3.39	**100.00+/-0.00**

Graph classification

- Classifiation accuracy on classical graph datasets.
- Comparison with state-of-the-art graph kernel approaches and Graph CNN.
- We use $\exp(-\gamma\mathcal{FGW})$ as a non-positive kernel for an SVM [Loosli et al., 2016] (FGW).
- Train Wassertsein Distance Measure Machine [Rakotomamonjy et al., 2018] (FGWDMM).

Application of FGW distance

Discrete attributes	MUTAG	NCI1	PTC
FGW raw sp	83.26+/-10.30	72.82+/-1.46	55.71+/-6.74
FGW wl h=2 sp	86.42+/-7.81	85.82+/-1.16	63.20+/-7.68
FGW wl h=2 sp regul	84.74+/-8.03	-	63.37+/-6.75
FGW wl h=4 sp	**88.42+/-5.67**	**86.42 +/- 1.63**	**65.31+/-7.90**
FGW wl h=4 sp regul	86.42+/-8.81	-	63.83+/-7.83
GK k=3	82.42+/-8.40	60.78+/-2.48	56.46+/-8.03
PSCN k=10	83.47+/-10.26	70.65+/-2.58	58.34+/-7.71
PSCN k=5	83.05+/-10.80	69.85+/-1.79	55.37+/-8.28
RW all cv	79.47+/-8.17	58.63+/-2.44	55.09+/-7.34
SP all cv	82.95+/-8.19	74.26+/-1.53	-
WL all cv	86.21+/-8.48	85.77+/-1.07	62.86+/-7.23
WL h=2	86.21+/-8.15	81.85+/-2.28	61.60+/-8.14
WL h=4	83.68+/-9.13	85.13+/-1.61	62.17+/-7.80

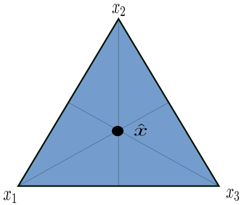
Without attribute	IMDB-B	IMDB-M
FGW raw sp	**63.80+/-3.49**	**48.00+/-3.22**
GK k=3	56.00+/-3.61	41.13+/-4.68
SP all cv	55.80+/-2.93	38.93+/-5.12

Graph classification

- Classifiation accuracy on classical graph datasets.
- Comparison with state-of-the-art graph kernel approaches and Graph CNN.
- We use $\exp(-\gamma\mathcal{FGW})$ as a non-positive kernel for an SVM [Loosli et al., 2016] (FGW).
- Train Wassertsein Distance Measure Machine [Rakotomamonjy et al., 2018] (FGWDMM).

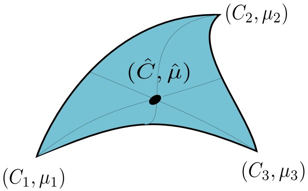
FGW barycenter

Euclidean barycenter



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

FGW barycenter

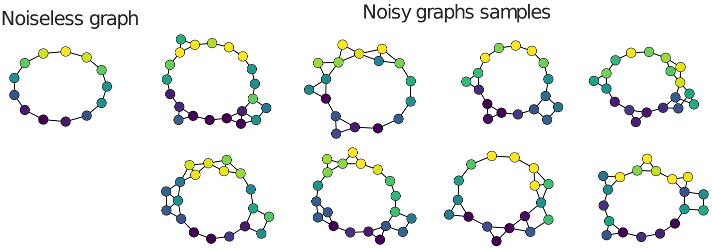


$$\min_{D \in \mathbb{R}^{n \times n}, \mu} \sum_i \lambda_i \mathcal{FGW}(C_i, C, \mu_i, \mu)$$

FGW barycenter $p = 1, q = 2$

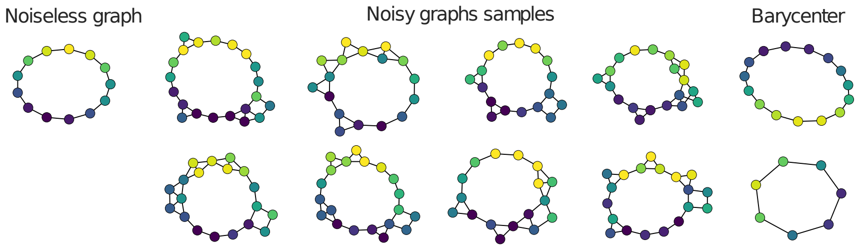
- Estimate FGW barycenter using Frechet means.
- Barycenter optimization solved via block coordinate descent (on $\boldsymbol{\pi}, D, \{a_i\}_i$).
- Can chose to fix the structure (D) or the features $\{a_i\}_i$ in the barycenter.
- $a_{i\,i}$, and D updates are weighted averages using $\boldsymbol{\pi}$.

FGW barycenter on labeled graphs



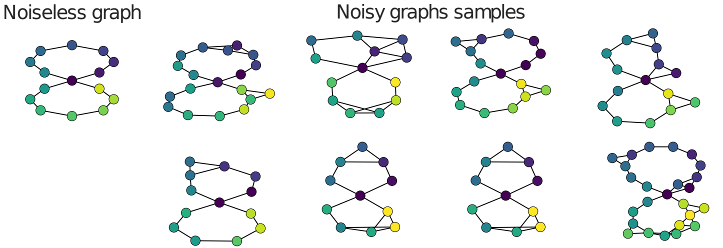
Barycenter of noisy graphs

- 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.



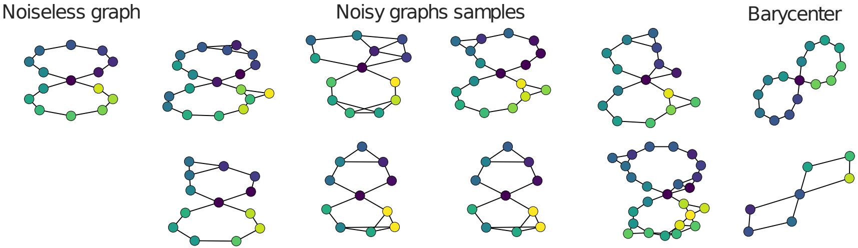
Barycenter of noisy graphs

- 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.



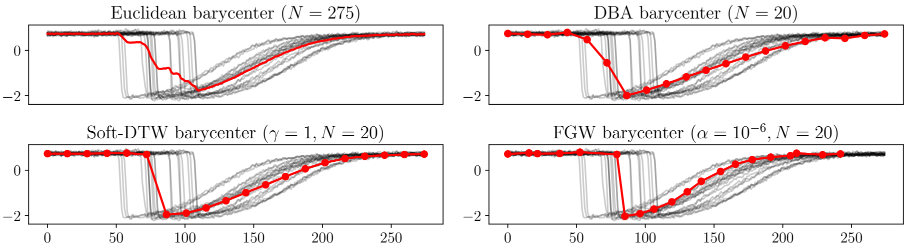
Barycenter of noisy graphs

- 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.



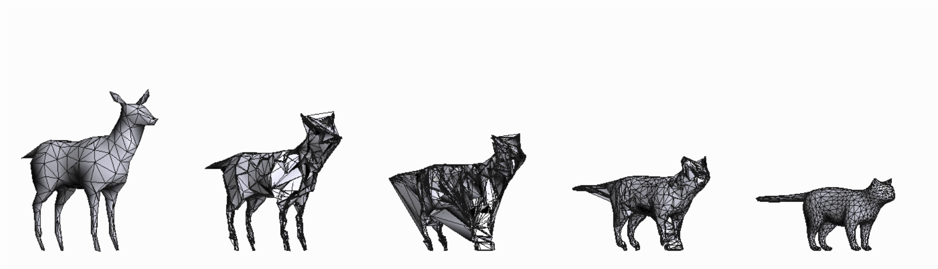
Barycenter of noisy graphs

- 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.



Time series averaging

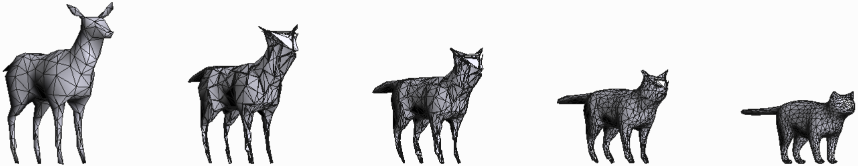
- Comparsion with Euclidean, DBA [Petitjean et al., 2011] and Soft-DTW [Cuturi and Blondel, 2017].
- Structure is time position of samples, fetaure value of the signal.
- Temporal position of nodes recovered with a MDS of D.
- Barycenter have non-regular sampling.



Mesh interpolation

- Two meshes (deer and cat).
- Fix structure from cat, estimate barycenter for the positions of the edges.
- Wasserstien ($\alpha = 0$) do not respect the graph (mesh neighborhood).
- FGW conserve the graph, regularized FGW smoothes the surface.

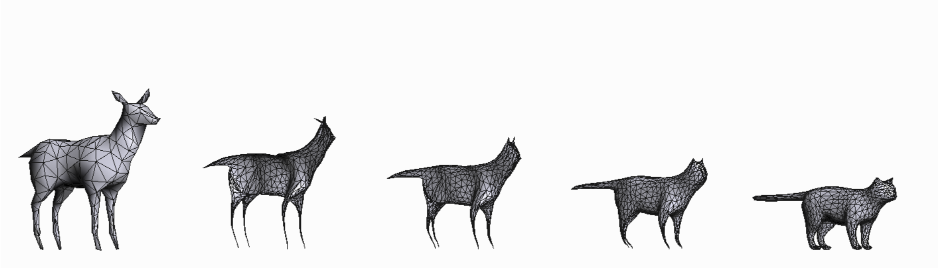
FGW barycenter for mesh interpolation



Mesh interpolation

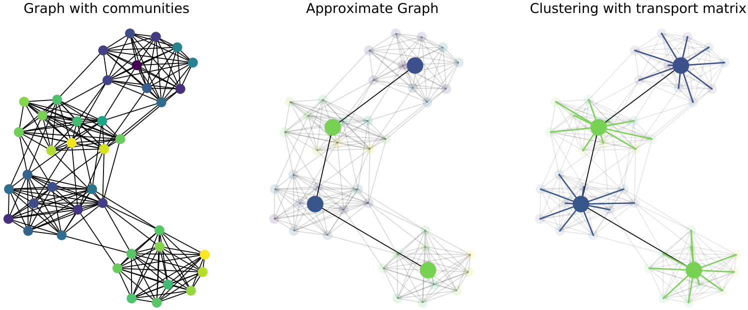
- Two meshes (deer and cat).
- Fix structure from cat, estimate barycenter for the positions of the edges.
- Wasserstien ($\alpha = 0$) do not respect the graph (mesh neighborhood).
- FGW conserve the graph, regularized FGW smoothes the surface.

FGW barycenter for mesh interpolation



Mesh interpolation

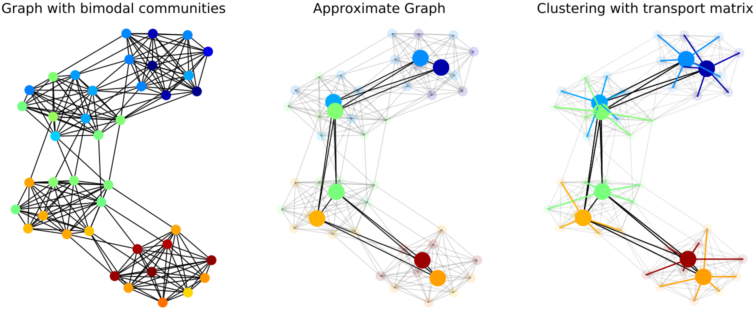
- Two meshes (deer and cat).
- Fix structure from cat, estimate barycenter for the positions of the edges.
- Wasserstien ($\alpha = 0$) do not respect the graph (mesh neighborhood).
- FGW conserve the graph, regularized FGW smoothes the surface.



Graph approximation and comunity clustering

$$\min_{D,\mu} \quad \mathcal{FGW}(D, D_0, \mu, \mu_0)$$

- Approximate the graph (D_0, μ_0) with a small number of nodes.
- OT matrix give the clustering affectation.
- Works for signle and multiple modes in the clusters.

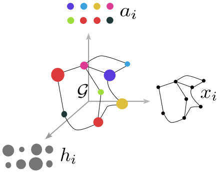


Graph approximation and comunity clustering

$$\min_{D,\mu} \quad \mathcal{FGW}(D, D_0, \mu, \mu_0)$$

- Approximate the graph (D_0, μ_0) with a small number of nodes.
- OT matrix give the clustering affectation.
- Works for signle and multiple modes in the clusters.

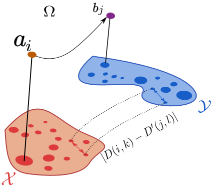
Conclusion for FGW



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

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

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



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

- Model structured data as distributions.
- New versatile method for comparing structured data based on Optimal Transport
- Many desirable distance properties
- New notion of barycenter of structured data such as graphs or time series
- Promising applications for signal over graphs and deep learning for structured data

What next ?

- Devise efficient optimization shemes for large structures.
- Add interpretability to deep neural networks on graph.

Thank you

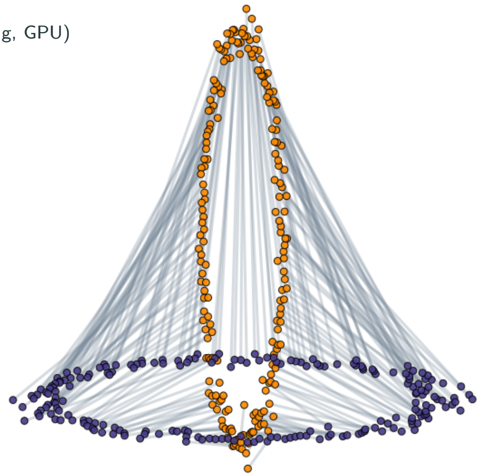
Python code available on GitHub:
`https://github.com/rflamary/POT`

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

Python code for JDOT on GitHub:
`https://github.com/rflamary/JDOT`

Papers available on my website:
`https://remi.flamary.com/`

Post docs available in:
Nice, ~~Rouen~~, ~~Rennes~~ (France)



Generalization bound (1)

Expected loss
The expected loss on a domain D and for a given predictor f is defined as

$$err_D(f) \stackrel{\text{def}}{=} \mathop{\mathbb{E}}_{(\mathbf{x},y)\sim\mathcal{P}_t} \mathcal{L}(y, f(\mathbf{x})).$$

Probabilistic Lipschitzness [Urner et al., 2011, Ben-David et al., 2012]
Let $\phi : \mathbb{R} \to [0,1]$. A labeling function $f : \Omega \to \mathbb{R}$ is ϕ-Lipschitz with respect to a distribution P over Ω if for all $\lambda > 0$

$$Pr_{x\sim P}\left[\exists y : [|f(x) - f(y)| > \lambda d(x,y)]\right] \leq \phi(\lambda).$$

Probabilistic Transfer Lipschitzness
Let μ_s and μ_t be respectively the source and target distributions. Let $\phi : \mathbb{R} \to [0,1]$. A labeling function $f : \Omega \to \mathbb{R}$ and a joint distribution $\Pi(\mu_s, \mu_t)$ over μ_s and μ_t are ϕ-Lipschitz transferable if for all $\lambda > 0$:

$$Pr_{(\mathbf{x}_1,\mathbf{x}_2)\sim\Pi(\mu_s,\mu_t)}\left[|f(\mathbf{x}_1) - f(\mathbf{x}_2)| > \lambda d(\mathbf{x}_1,\mathbf{x}_2)\right] \leq \phi(\lambda).$$

Generalization bound (2)

Theorem 1
Let f be any labeling function of $\in \mathcal{H}$. Let $\Pi^* = \text{argmin}_{\Pi \in \Pi(\mathcal{P}_s, \mathcal{P}_t^f)} \int_{(\Omega \times \mathcal{C})^2} \alpha d(\mathbf{x}_s, \mathbf{x}_t) + \mathcal{L}(y_s, y_t) d\Pi(\mathbf{x}_s, y_s; \mathbf{x}_t, y_t)$ and $W_1(\hat{\mathcal{P}}_s, \hat{\mathcal{P}_t^f})$ the associated 1-Wasserstein distance. Let $f^* \in \mathcal{H}$ be a Lipschitz labeling function that verifies the ϕ-probabilistic transfer Lipschitzness (PTL) assumption w.r.t. Π^* and that minimizes the joint error $err_S(f^*) + err_T(f^*)$ w.r.t all PTL functions compatible with Π^*. We assume the input instances are bounded s.t. $|f^*(\mathbf{x}_1) - f^*(\mathbf{x}_2)| \leq M$ for all $\mathbf{x}_1, \mathbf{x}_2$. Let $\mathcal{L}$ be any symmetric loss function, k-Lipschitz and satisfying the triangle inequality. Consider a sample of N_s labeled source instances drawn from $\mathcal{P}_s$ and N_t unlabeled instances drawn from μ_t, and then for all $\lambda > 0$, with $\alpha = k\lambda$, we have with probability at least $1 - \delta$ that:

$$\begin{aligned} \mathsf{err}_T(f) \;\leq\; & W_1(\hat{\mathcal{P}}_s, \hat{\mathcal{P}_t^f}) + \sqrt{\frac{2}{c'} \log(\frac{2}{\delta})} \left(\frac{1}{\sqrt{N_S}} + \frac{1}{\sqrt{N_T}} \right) \\ & + err_S(f^*) + err_T(f^*) + kM\phi(\lambda). \end{aligned}$$

- First term is JDOT objective function.
- Second term is an empirical sampling bound.
- Last terms are usual in DA [Mansour et al., 2009, Ben-David et al., 2010].

Ben-David, S., Blitzer, J., Crammer, K., Kulesza, A., Pereira, F., and Vaughan, J. W. (2010).
A theory of learning from different domains.
Machine Learning, 79(1-2):151–175.

Ben-David, S., Shalev-Shwartz, S., and Urner, R. (2012).
Domain adaptation–can quantity compensate for quality?
In *Proc of ISAIM*.

Benamou, J.-D., Carlier, G., Cuturi, M., Nenna, L., and Peyré, G. (2015).
Iterative Bregman projections for regularized transportation problems.
SISC.

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

Cuturi, M. (2013).
Sinkhorn distances: Lightspeed computation of optimal transportation.
In *Neural Information Processing Systems (NIPS)*, pages 2292–2300.

Cuturi, M. and Blondel, M. (2017).
Soft-DTW: a differentiable loss function for time-series.
volume 70, pages 894–903, International Convention Centre, Sydney, Australia. PMLR.

Damodaran, B. B., Kellenberger, B., Flamary, R., Tuia, D., and Courty, N. (2018).
Deepjdot: Deep joint distribution optimal transport for unsupervised domain adaptation.

Ferradans, S., Papadakis, N., Peyré, G., and Aujol, J.-F. (2014a).
Regularized discrete optimal transport.
SIAM Journal on Imaging Sciences, 7(3).

Ferradans, S., Papadakis, N., Peyré, G., and Aujol, J.-F. (2014b).
Regularized discrete optimal transport.
SIAM Journal on Imaging Sciences, 7(3):1853–1882.

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

Lacoste-Julien, S. (2016).
Convergence rate of frank-wolfe for non-convex objectives.
arXiv preprint arXiv:1607.00345.

Loosli, G., Canu, S., and Ong, C. S. (2016).
Learning svm in krein spaces.
IEEE transactions on pattern analysis and machine intelligence, 38(6):1204–1216.

Mansour, Y., Mohri, M., and Rostamizadeh, A. (2009).
Domain adaptation: Learning bounds and algorithms.
In *Proc. of COLT*.

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.

Peng, X., Usman, B., Kaushik, N., Hoffman, J., Wang, D., and Saenko, K. (2017).
Visda: The visual domain adaptation challenge.
arXiv preprint arXiv:1710.06924.

Perrot, M., Courty, N., Flamary, R., and Habrard, A. (2016).
Mapping estimation for discrete optimal transport.
In *Neural Information Processing Systems (NIPS)*.

Petitjean, F., Ketterlin, A., and Gançarski, P. (2011).
A global averaging method for dynamic time warping, with applications to clustering.
44(3):678–693.

Peyré, G., Cuturi, M., and Solomon, J. (2016).
Gromov-Wasserstein Averaging of Kernel and Distance Matrices.
In *ICML 2016*, Proc. 33rd International Conference on Machine Learning, New-York, United States.

Rakotomamonjy, A., Traore, A., Berar, M., Flamary, R., and Courty, N. (2018).
Wasserstein Distance Measure Machines.
preprint.

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.

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

Thorpe, M., Park, S., Kolouri, S., Rohde, G. K., and Slepcev, D. (2017).
A transportation lp distance for signal analysis.
Journal of Mathematical Imaging and Vision, 59(2):187–210.

Tuia, D., Flamary, R., Rakotomamonjy, A., and Courty, N. (2015).
Multitemporal classification without new labels: a solution with optimal transport.
In *8th International Workshop on the Analysis of Multitemporal Remote Sensing Images*.

Urner, R., Shalev-Shwartz, S., and Ben-David, S. (2011).
Access to unlabeled data can speed up prediction time.
In *Proceedings of ICML*, pages 641–648.

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

Venkateswara, H., Eusebio, J., Chakraborty, S., and Panchanathan, S. (2017).
Deep hashing network for unsupervised domain adaptation.
In *(IEEE) Conference on Computer Vision and Pattern Recognition (CVPR)*.