Optimal Transport for Machine learning

Domain Adaptation and structured data

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

$$
\inf_{T\#\mu_s = \mu_t} \int_{\Omega_s} c(x, T(x)) \mu_s(x) dx
$$

(1)

- Non-convex optimization problem, mapping does not exist in the general case.
The Kantorovich formulation [Kantorovich, 1942] seeks for a probabilistic coupling $\pi \in \mathcal{P}(\Omega_s \times \Omega_t)$ between $\Omega_s$ and $\Omega_t$:

$$\pi_0 = \text{argmin}_\pi \int_{\Omega_s \times \Omega_t} c(x, y) \pi(x, y) \, dx \, dy,$$

s.t. $\pi \in \Pi = \left\{ \pi \geq 0, \int_{\Omega_t} \pi(x, y) \, dy = \mu_s, \int_{\Omega_s} \pi(x, y) \, dx = \mu_t \right\}$

- $\pi$ is a joint probability measure with marginals $\mu_s$ and $\mu_t$.
- Linear Program that always have a solution.
Wasserstein distance

\[ W_p^p(\mu_s, \mu_t) = \min_{\pi \in \Pi} \int_{\Omega_s \times \Omega_t} c(x, y) \pi(x, y) dx dy = E_{(x, y) \sim \pi}[c(x, y)] \]  

where \( c(x, y) = \|x - y\|^p \) 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

Introduction
  Optimal transport
  Optimal transport and machine learning

Optimal transport for domain adaptation
  Supervised learning and Domain adaptation
  Optimal Transport for Domain Adaptation (OTDA)
  Joint distribution OT for domain adaptation (JDOT)

Optimal Transport on structured data
  Gromov-Wasserstein distance for structured data
  Structured data as distributions
  Fused Gromov-Wasserstein distance
  Applications on structured data classification
Optimal transport for domain adaptation
Supervised learning

Traditional supervised learning

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

Empirical risk minimization
We seek for a predictor \( f \) minimizing

\[
\min_f \left\{ \mathbb{E}_{(x,y) \sim \hat{\mathcal{P}}} \ L(y, f(x)) = \sum_j L(y_j, f(x_j)) \right\}
\]  

(4)

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

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 $T$ between the two domains.
- The transport preserves the conditional distributions:
  \[
  P_s(y|x_s) = P_t(y|T(x_s)).
  \]

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 \( \pi_0 \)).
- 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

Dataset

Class 1
Class 2
Samples
Samples
Classifier on

Optimal transport

\[ \Omega_t \]

\[ \Omega_s \]

\[ T_{\gamma_0}(\cdot) \]

\[ \text{Samples } T_{\gamma_0}(x_i^s) \]

\[ \text{Samples } x_i^s \]

\[ \text{Classifier on } x_i^s \]

Classification on transported samples

\[ \Omega_t \]

\[ \text{Samples } T_{\gamma_0}(x_i^t) \]

\[ \text{Samples } x_i^t \]

\[ \text{Classifier on } T_{\gamma_0}(x_i^t) \]

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 \( f : \mathbb{R}^d \rightarrow \mathbb{R} \).
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 $C$ the set of labels.
- Let $P_s(X, Y) \in \mathcal{P}(\Omega \times C)$ and $P_t(X, Y) \in \mathcal{P}(\Omega \times C)$ the source and target joint distribution.
- We have access to an empirical sampling $\hat{P}_s = \frac{1}{N_s} \sum_{i=1}^{N_s} \delta_{x^s_i, y^s_i}$ of the source distribution defined by $X_s = \{x^s_i\}_{i=1}^{N_s}$ and label information $Y_s = \{y^s_i\}_{i=1}^{N_s}$.
- **but** the target domain is defined only by an empirical distribution in the feature space with samples $X_t = \{x^t_i\}_{i=1}^{N_t}$. 
Joint distribution OT (JDOT)

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 $y$

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

and its empirical counterpart

$$
\hat{\mathcal{P}}_t^f = \frac{1}{N_t} \sum_{i=1}^{N_t} \delta_{x_i^t, f(x_i^t)}.
$$

Learning with JDOT

We propose to learn the predictor $f$ that minimize:

$$
\min_{f} \left\{ \mathbb{W}_1(\hat{\mathcal{P}}_s, \hat{\mathcal{P}}_t^f) = \inf_{\pi \in \Pi} \sum_{i,j} \mathcal{D}(x_i^s, y_i^s; x_j^t, f(x_j^t)) \pi_{ij} \right\}
$$

- $\Pi$ is the transport polytope.
- $\mathcal{D}(x_i^s, y_i^s; x_j^t, f(x_j^t)) = \alpha \|x_i^s - x_j^t\|^2 + \mathcal{L}(y_i^s, f(x_j^t))$ 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

\[
\min_{f \in \mathcal{H}, \pi \in \Pi} \sum_{i,j} \pi_{i,j} \left( \alpha d(x_s^i, x_t^j) + \mathcal{L}(y_s^i, f(x_t^j)) \right) + \lambda \Omega(f)
\]  

(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.

\[\pi \text{ update for a fixed } f\]

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

\[f \text{ update for a fixed } \pi\]

\[
\min_{f \in \mathcal{H}} \sum_{i,j} \pi_{i,j} \mathcal{L}(y_s^i, f(x_t^j)) + \lambda \Omega(f)
\]  

(8)

- Weighted loss from all source labels.
- \(\pi\) performs label propagation.
**Least square regression with quadratic regularization**

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

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

- $\hat{y}_j = n_t \sum_j \pi_{i,j} y^s_i$ 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 $\pi$ the optimization problem is equivalent to

$$\min_{f_k \in \mathcal{H}} \sum_{j,k} \hat{P}_{j,k} \mathcal{L}(1, f_k(x^t_j)) + (1 - \hat{P}_{j,k}) \mathcal{L}(-1, f_k(x^t_j)) + \lambda \sum_k \|f_k\|^2$$  \hspace{1cm} (10)

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

  $$Y^s_{i,k} = \begin{cases} 1 & \text{if } y^s_i = k \\ -1 & \text{else} \end{cases}, \quad P^s_{i,k} = \begin{cases} 1 & \text{if } y^s_i = k \\ 0 & \text{else} \end{cases}$$

  with $k \in 1, \cdots, K$ and $K$ being the number of classes.
Loss (9):

\[ L_s(y_i^s, f(g(x_i^s))) + \sum_{i,j} \gamma_{ij} \left( \|g(x_i^s) - g(x_j^t)\|^2 + \lambda_t \mathcal{L}(y_i^s, f(g(x_j^t))) \right) \]

\[ (11) \]

**DeepJDOT [Damodaran et al., 2018]**

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

DeepJDOT [Damodaran et al., 2018]

- Evaluation of DeepJDOT on visual classification tasks.
- Digit adaptation between MNIST, USPS, SVHN, MNIST-M.
- Ablation study: all terms are important.
- TSNE projections of embeddings (MNIST→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.
- Ablation study: all terms are important.
- TSNE projections of embeddings (MNIST → 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.
- Ablation study: all terms are important.
- TSNE projections of embeddings (MNIST→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.
- Ablation study: all terms are important.
- TSNE projections of embeddings (MNIST → 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.
- Ablation study: all terms are important.
- TSNE projections of embeddings (MNIST→MNIST-M).
**Conclusion OTDA**

**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

- 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

- Use 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

• Use both features (labels) and structure (graph).

• Allows for comparison, classification.

• Data science (statistics, means)
Structured data as distributions

\[ \mu = \sum_{i=1}^{n} h_i \delta(x_i, a_i) \]

\[ \mu_A = \sum_{i} h_i \delta_{a_i} \]

\[ \mu_X = \sum_{i} h_i \delta_{x_i} \]

Graph data representation

- 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 structured data

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

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

- Wasserstein good for (empirical) distributions, samples as IID.
- OT can encode structure with OT $L^p$ [Thorpe et al., 2017] by extending the feature space but requires the same ambient space.
Gromov-Wasserstein distance for structured data inspired from Gabriel Peyré

GW for structured data [Memoli, 2011]

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

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

- 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

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

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

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

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

**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 continuous distributions**

- Interpolation between \(\mathcal{W}\) (\(\alpha = 0\)) and \(\mathcal{G}\) (\(\alpha = 1\)) distances.
- Geodesic properties (constant speed, unicity).
\[ \mathcal{FGW}_{p,q,\alpha}(D, D', \mu_s, \mu_t) = \left( \min_{\pi \in \Pi(\mu_s, \mu_t)} \sum_{i,j,k,l} \left( (1-\alpha)M_{i,j}^q + \alpha|D_{i,k} - D_{j,l}'|_q \right)^p \pi_{i,j} \pi_{k,l} \right)^{\frac{1}{p}} \]

**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 \( d = \text{Dim}(\mathcal{X}) + \text{Dim}(\Omega) \):
  
  \[ \mathbb{E}[\mathcal{FGW}(\mu, \mu_n)] = O \left( n^{-\frac{1}{d}} \right) \]
Computing FGW

\[ \pi^* = \arg \min_{\pi \in \Pi(\mu_s, \mu_t)} \text{vec}(\pi)^T Q \text{vec}(\pi) + \text{vec}((1 - \alpha) M)^T \text{vec}(\pi) \]  \hspace{1cm} (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].
\[ \pi^* = \arg \min_{\pi \in \Pi(\mu_s, \mu_t)} \text{vec}(\pi)^TQ\text{vec}(\pi) + \text{vec}(1 - \alpha)M^T\text{vec}(\pi) \] (12)

Algorithm 1  Conditional Gradient (CG) for FGW

1: \( \pi^{(0)} \leftarrow \mu_X\mu_Y^\top \)
2: \textbf{for} \( i = 1, \ldots, \) \textbf{do}
3: \( G \leftarrow \) Gradient from Eq. (12) \textit{w.r.t.} \( \pi^{(i-1)} \)
4: \( \tilde{\pi}^{(i)} \leftarrow \) Solve OT with ground loss \( G' \)
5: \( \tau^{(i)} \leftarrow \) Line-search for loss with \( \tau \in (0, 1) \)
6: \( \pi^{(i)} \leftarrow (1 - \tau^{(i)})\pi^{(i-1)} + \tau^{(i)}\tilde{\pi}^{(i)} \)
7: \textbf{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].
Illustration of FGW distance

(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.
Application of FGW distance

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

Graph classification

- Classification accuracy on classical graph datasets.
- Comparison with state-of-the-art graph kernel approaches and Graph CNN.
- We use $\exp(-\gamma FGW)$ 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

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

<table>
<thead>
<tr>
<th>Without attribute</th>
<th>IMDB-B</th>
<th>IMDB-M</th>
</tr>
</thead>
<tbody>
<tr>
<td>FGW raw sp</td>
<td>63.80+/3.49</td>
<td>48.00+/3.22</td>
</tr>
<tr>
<td>GK k=3</td>
<td>56.00+/3.61</td>
<td>41.13+/4.68</td>
</tr>
<tr>
<td>SP all cv</td>
<td>55.80+/2.93</td>
<td>38.93+/5.12</td>
</tr>
</tbody>
</table>

### Graph classification

- Classification accuracy on classical graph datasets.
- Comparison with state-of-the-art graph kernel approaches and Graph CNN.
- We use \( \exp(-\gamma FGW) \) 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 $p = 1, q = 2$

- Estimate FGW barycenter using Frechet means.
- Barycenter optimization solved via block coordinate descent (on $\pi, D, \{a_i\}_i$).
- 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 $\pi$. 

Euclidean barycenter

\[
\min_x \sum_k \lambda_k \| x - x_k \|^2
\]

FGW barycenter

\[
\min_{D \in \mathbb{R}^{n \times n}, \mu} \sum_i \lambda_i \mathcal{F}_{GW}(C_i, C, \mu_i, \mu)
\]
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

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.
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.
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
Time series averaging

- Comparison with Euclidean, DBA [Petitjean et al., 2011] and Soft-DTW [Cuturi and Blondel, 2017].
- Structure is time position of samples, feature 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.
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 community clustering

\[
\min_{D,\mu} \mathcal{FGW}(D, D_0, \mu, \mu_0)
\]

- Approximate the graph \((D_0, \mu_0)\) with a small number of nodes.
- OT matrix give the clustering affectation.
- Works for single and multiple modes in the clusters.
Graph approximation and comunity clustering

$$\min_{D, \mu} \mathcal{F}_{GW}(D, D_0, \mu, \mu_0)$$

- Approximate the graph \((D_0, \mu_0)\) with a small number of nodes.
- OT matrix give the clustering affectation.
- Works for signle and multiple modes in the clusters.
Conclusion for FGW

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

$$\text{err}_D(f) \overset{\text{def}}{=} \mathbb{E}_{(x,y) \sim \mathcal{P}_t} \mathcal{L}(y, f(x)).$$

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

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

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

$$\Pr_{(x_1, x_2) \sim \Pi(\mu_s, \mu_t)} [|f(x_1) - f(x_2)| > \lambda d(x_1, x_2)] \leq \phi(\lambda).$$
Theorem 1
Let $f$ be any labeling function of $\in \mathcal{H}$. Let
$$\Pi^\ast = \arg\min_{\Pi \in \Pi(\mathcal{P}_s, \mathcal{P}_t^f)} \int_{\Omega \times C} \alpha d(x_s, x_t) + L(y_s, y_t) d\Pi(x_s, y_s; x_t, y_t)$$
and $W_1(\hat{\mathcal{P}}_s, \hat{\mathcal{P}}_t^f)$ the associated 1-Wasserstein distance. Let $f^\ast \in \mathcal{H}$ be a Lipschitz labeling function that verifies the $\phi$-probabilistic transfer Lipschitzness (PTL) assumption w.r.t. $\Pi^\ast$ and that minimizes the joint error $err_S(f^\ast) + err_T(f^\ast)$ w.r.t all PTL functions compatible with $\Pi^\ast$. We assume the input instances are bounded s.t. $|f^\ast(x_1) - f^\ast(x_2)| \leq M$ for all $x_1, x_2$. Let $L$ 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 $\mu_t$, and then for all $\lambda > 0$, with $\alpha = k\lambda$, we have with probability at least $1 - \delta$ that:

$$err_T(f) \leq W_1(\hat{\mathcal{P}}_s, \hat{\mathcal{P}}_t^f) + \sqrt{\frac{2}{c'}} \log\left(\frac{2}{\delta}\right) \left(\frac{1}{\sqrt{N_S}} + \frac{1}{\sqrt{N_T}}\right) + err_S(f^\ast) + err_T(f^\ast) + kM\phi(\lambda).$$

• 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].

**A theory of learning from different domains.**


Ben-David, S., Shalev-Shwartz, S., and Urner, R. (2012).

**Domain adaptation–can quantity compensate for quality?**

In *Proc of ISAIM*.


**Iterative Bregman projections for regularized transportation problems.**

*SISC*.


**Optimal transport for domain adaptation.**

*Pattern Analysis and Machine Intelligence, IEEE Transactions on*. 
**Sinkhorn distances: Lightspeed computation of optimal transportation.**  

**Soft-DTW: a differentiable loss function for time-series.**  
volume 70, pages 894–903, International Convention Centre, Sydney, Australia. PMLR.

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

**Regularized discrete optimal transport.**  
*SIAM Journal on Imaging Sciences*, 7(3).


**Domain adaptation: Learning bounds and algorithms.**  
In *Proc. of COLT*.

**Gromov wasserstein distances and the metric approach to object matching.**  

Monge, G. (1781).  
**Mémoire sur la théorie des déblais et des remblais.**  
De l’Imprimerie Royale.

**Visda: The visual domain adaptation challenge.**  


Urner, R., Shalev-Shwartz, S., and Ben-David, S. (2011).
Access to unlabeled data can speed up prediction time.

Fused gromov-wasserstein distance for structured objects: theoretical foundations and mathematical properties.

Deep hashing network for unsupervised domain adaptation.
In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*.