Optimal Transport for Machine Learning

Part 2: Machine learning applications

Rémi Flamary

July 4 2022

Hi Paris Summer School, 2022, Paris
Introduction
Three aspects of Machine Learning

Unsupervised learning
- Extract information from unlabeled data
- Find labels (clustering) or subspaces/manifolds.
- Generate realistic data (GAN).

Supervised Learning
- Learning to predict from labeled dataset.
- Regression, Classification.
- Can use unsupervised information (DA, Semi-sup.)

Reinforcement Learning
- Let the machine experiment.
- Learn from its mistakes.
- Framework for learning to play games.
Short history of OT for ML

- Recently introduced to ML (well known in image processing since 2000s).
- Computational OT allow numerous applications (regularization).
- Deep learning boost (numerical optimization and GAN).
Three aspects of optimal transport for ML

Transporting with optimal transport
- Color adaptation in image [Ferradans et al., 2014].
- Style transfer [Mroueh, 2019].
- Domain adaptation [Courty et al., 2016].

Divergence between histograms
- Use the ground metric to encode complex relations between the bins.
- Loss for multilabel classifier [Frogner et al., 2015]
- Adversarial regularization [Fatras et al., 2021].

Divergence between empirical distributions
- Non parametric divergence between non overlapping distributions.
- Generative modeling [Arjovsky et al., 2017].
- Data imputation [Muzellec et al., 2020].
Table of content

Introduction

Mapping with optimal transport
  Optimal transport mapping estimation
  Optimal transport for domain adaptation

Learning from histograms with Optimal Transport
  Unsupervised learning
  Supervised learning

Learning from empirical distributions with Optimal Transport
  Unsupervised learning
  Supervised learning and domain adaptation

Conclusion
Mapping with optimal transport
Mapping estimation

- Barycentric mapping using the OT matrix [Ferradans et al., 2014].
- Linear Monge mapping when data supposed Gaussian [Flamary et al., 2019].
- Smooth mapping estimation
  [Perrot et al., 2016, Seguy et al., 2017, Paty et al., 2020].
- Estimation for $W_2$ using input convex neural networks [Makkuva et al., 2020].
- Can be used to linearize the Wasserstein space [Mérigot et al., 2020]
Transporting the discrete samples

Barycentric mapping [Ferradans et al., 2014]

\[ \hat{T}_{\gamma_0}(x^s_i) = \arg \min_x \sum_j \gamma_0(i, j) c(x, x^t_j). \]  

- The mass of each source sample is spread onto the target samples (line of \( \gamma_0 \)).
- The mapping is the barycenter of the target samples weighted by \( \gamma_0 \).
- Closed form solution for the quadratic loss.
- Limited to the samples in the distribution (no out of sample).
- Trick: learn OT on few samples and apply displacement to the nearest point.
Transporting the discrete samples

Barycentric mapping [Ferradans et al., 2014]

\[
\hat{T}_{\gamma_0}(x^s_i) = \arg \min_x \sum_j \gamma_0(i, j) \| x - x^t_j \|_2^2.
\]

- The mass of each source sample is spread onto the target samples (line of \( \gamma_0 \)).
- The mapping is the barycenter of the target samples weighted by \( \gamma_0 \).
- Closed form solution for the quadratic loss.
- Limited to the samples in the distribution (no out of sample).
- Trick: learn OT on few samples and apply displacement to the nearest point.
Transporting the discrete samples

Distributions

Distributions

Source $\mu_s$

Target $\mu_t$

Classic OT (LP)

Reg. Entropic OT

Barycentric mapping [Ferradans et al., 2014]

$$\hat{T}_{\gamma_0}(x_i^s) = \frac{1}{\sum_j \gamma_0(i, j)} \sum_j \gamma_0(i, j)x_j^t. \quad (1)$$

- The mass of each source sample is spread onto the target samples (line of $\gamma_0$).
- The mapping is the barycenter of the target samples weighted by $\gamma_0$
- Closed form solution for the quadratic loss.
- Limited to the samples in the distribution (no out of sample).
- Trick: learn OT on few samples and apply displacement to the nearest point.
Transporting the discrete samples

Distributions

Source $\mu_s$
Target $\mu_t$

Classic OT (LP)

Reg. Entropic OT

Barycentric mapping [Ferradans et al., 2014]

$$\hat{T}_{\gamma_0}(x^s_i) = \frac{1}{\sum_j \gamma_0(i, j)} \sum_j \gamma_0(i, j)x^t_j.$$  

1. The mass of each source sample is spread onto the target samples (line of $\gamma_0$).
2. The mapping is the barycenter of the target samples weighted by $\gamma_0$.
3. Closed form solution for the quadratic loss.
4. Limited to the samples in the distribution (no out of sample).
5. Trick: learn OT on few samples and apply displacement to the nearest point.
Transporting the discrete samples

Barycentric mapping \cite{Ferradans2014}

\[
\hat{T}_{\gamma_0}(x_i^s) = \frac{1}{\sum_j \gamma_0(i, j)} \sum_j \gamma_0(i, j)x_j^t.
\] (1)

- The mass of each source sample is spread onto the target samples (line of $\gamma_0$).
- The mapping is the barycenter of the target samples weighted by $\gamma_0$.
- Closed form solution for the quadratic loss.
- Limited to the samples in the distribution (no out of sample).
- Trick: learn OT on few samples and apply displacement to the nearest point.
Simultaneous OT matrix and mapping [Perrot et al., 2016]

\[
\min_{T, \gamma \in \mathcal{P}} \langle \gamma, C \rangle_F + \sum_i \| T(x_i^s) - \hat{T}_\gamma(x_i^s) \|^2 + \lambda \| T \|^2
\]

- Estimate jointly the OT matrix and a smooth mapping approximating the barycentric mapping.
- The mapping is a regularization for OT.
- Controlled generalization error (statistical bound).
- Linear and kernel mappings $T$, limited to small scale datasets.
Large scale mapping estimation [Seguy et al., 2017]

- 2-step procedure:
  1. (Stochastic) estimation of regularized $\hat{\gamma}$.
  2. (Stochastic) estimation of $T$ with a neural network.

- OT solved with Stochastic Gradient Ascent in the dual.

- Convergence to the true mapping for small regularization.

- Convergence to the smooth mapping for large $n$
  [Pooladian and Niles-Weed, 2021].
Monge Mapping with input convex neural networks

Principle [Makkuva et al., 2020]

- For the quadratic cost OT between two smooth distribution Brenier theorem states that the Monge mapping is the gradient of a convex function.
- Neural network can be designed to be convex wrt their input (ICNN) [Amos et al., 2017].
- [Makkuva et al., 2020] proposed to estimate directly the Monge as a gradient of an ICNN from the empirical distributions. mapping usin
Seamless copy in images

**Poisson image editing [Pérez et al., 2003]**

- Use the color gradient from the source image.
- Use color border conditions on the target image.
- Solve Poisson equation to reconstruct the new image.

Example and webcam demo: [https://github.com/ncourty/PoissonGradient](https://github.com/ncourty/PoissonGradient)
Seamless copy in images

Poisson image editing [Pérez et al., 2003]
- Use the color gradient from the source image.
- Use color border conditions on the target image.
- Solve Poisson equation to reconstruct the new image.

Seamless copy with gradient adaptation [Perrot et al., 2016]
- Transport the gradient from the source to target color gradient distribution.
- Solve the Poisson equation with the mapped source gradients.
- Better respect of the color dynamic and limits false colors.
Seamless copy in images

Poisson image editing [Pérez et al., 2003]
- Use the color gradient from the source image.
- Use color border conditions on the target image.
- Solve Poisson equation to reconstruct the new image.

Seamless copy with gradient adaptation [Perrot et al., 2016]
- Transport the gradient from the source to target color gradient distribution.
- Solve the Poisson equation with the mapped source gradients.
- Better respect of the color dynamic and limits false colors.
Seamless copy in images

Poisson image editing [Pérez et al., 2003]
- Use the color gradient from the source image.
- Use color border conditions on the target image.
- Solve Poisson equation to reconstruct the new image.

Seamless copy with gradient adaptation [Perrot et al., 2016]
- Transport the gradient from the source to target color gradient distribution.
- Solve the Poisson equation with the mapped source gradients.
- Better respect of the color dynamic and limits false colors.

Example and webcam demo: https://github.com/ncourty/PoissonGradient
Monge 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 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.
**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 $\gamma_0$).
- Transport using barycentric mapping [Ferradans et al., 2014].
- 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

<table>
<thead>
<tr>
<th>USPS</th>
<th>MNIST</th>
<th>PIE05</th>
<th>PIE07</th>
<th>PIE09</th>
<th>PIE29</th>
<th>Calltech</th>
<th>Amazon</th>
<th>DSLR</th>
<th>Webcam</th>
</tr>
</thead>
<tbody>
<tr>
<td><img src="image1.png" alt="Image" /></td>
<td><img src="image2.png" alt="Image" /></td>
<td><img src="image3.png" alt="Image" /></td>
<td><img src="image4.png" alt="Image" /></td>
<td><img src="image5.png" alt="Image" /></td>
<td><img src="image6.png" alt="Image" /></td>
<td><img src="image7.png" alt="Image" /></td>
<td><img src="image8.png" alt="Image" /></td>
<td><img src="image9.png" alt="Image" /></td>
<td><img src="image10.png" alt="Image" /></td>
</tr>
</tbody>
</table>

#### 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

- State of the art performances on the 3 datasets.
- Works well on deep features adaptation and extension to semi-supervised DA.
Multi-subject P300 classification [Gayraud et al., 2017]

- Objective: reduce calibration for BCI users.
- P300 signal is different across subjects so adapting models is hard.
- Perform XDAWN [Rivet et al., 2009] as pre-processing.
- Use OTDA to adapt each subject in the dataset to a new subject.
- Train independent classifier on transported data and perform aggregation.
Multi-subject P300 classification [Gayraud et al., 2017]

- Objective: reduce calibration for BCI users.
- P300 signal is different across subjects so adapting models is hard.
- Perform XDAWN [Rivet et al., 2009] as pre-processing.
- Use OTDA to adapt each subject in the dataset to a new subject.
- Train independent classifier on transported data and perform aggregation.
EEG sleep stage classification [Chambon et al., 2018]

- Use pre-trained neural network.
- Adapt with OTDA on the penultimate layer.
- OTDA best DA approach to adapt between EEG recordings.

Prostate cancer classification [Gautheron et al., 2017]

- Adaptation of MRI voxel features from 1.5T to 3T.
- Achieve good performance across subjects and modality with no target labels.
Heterogeneous Domain Adaptation with GW

- OT for DA initially proposed by [Courty et al., 2016].
- Use the OT matrix to transfer labels or samples between datasets.
- GW find correspondences across spaces but very noisy.
- Semi-supervised strategy allows very good performances.
- Alternative: Co-optimal transport that find correspondances between the variables and samples simultaneously [Redko et al., 2020].

Semi-supervised Heterogeneous Domain Adaptation [Yan et al., 2018]
Introduction

Mapping with optimal transport
  - Optimal transport mapping estimation
  - Optimal transport for domain adaptation

Learning from histograms with Optimal Transport
  - Unsupervised learning
  - Supervised learning

Learning from empirical distributions with Optimal Transport
  - Unsupervised learning
  - Supervised learning and domain adaptation

Conclusion
Data as histograms

- Fixed bin positions $x_i$ e.g. grid, simplex $\Delta = \{(\mu_i)_i \geq 0; \sum_i \mu_i = 1\}$
- A lot of datasets come under the form of histograms.
- Images are photo counts (black and white), text as word counts.
- Natural divergence is Kullback–Leibler.
- Not all data can be seen as histograms (positivity+constant mass)!
Dictionary learning on histograms

\[
\min_{D,H} \sum_i W_C(v_i, D h_i)
\]

- NMF: columns of \( D \) and \( H \) are on the simplex.
- Metric \( C \) can encode spatial relations between the bins of the histograms.
- Ground metric learning [Zen et al., 2014].
- Fast DL with regularized OT [Rolet et al., 2016].
Dictionary learning on histograms

\[ \min_{D,H} \sum_i W_C(v_i, Dh_i) \]

- NMF: columns of $D$ and $H$ are on the simplex.
- Metric $C$ can encode spatial relations between the bins of the histograms.
- Ground metric learning [Zen et al., 2014].
- Fast DL with regularized OT [Rolet et al., 2016].
Optimal Spectral Transportation (OST)

OT linear spectral unmixing of musical data [Flamary et al., 2016]

\[
\min_{h \in \Delta} \ W_C(v, Dh)
\]  \hspace{1cm} (2)

- Objective: robustness to harmonic magnitude and small frequency shift
- Encode harmonic structure in the cost matrix (harmonic robustness).
- Can use simple dictionary (diracs on fundamental frequency).
- Very fast solver for sparse and entropic regularization.

Demo: https://github.com/rflamary/OST
Geodesic PCA in the Wasserstein space [Bigot et al., 2017]

- Generalization of Principal Component Analysis to the Wasserstein manifold.
- Regularized OT [Seguy and Cuturi, 2015].
- Approximation using Wasserstein embedding [Courty et al., 2017a].
Learning with a Wasserstein Loss [Frogner et al., 2015]

$$\min_f \sum_{k=1}^{N} W_1(f(x_i), l_i)$$

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

Principle [Fatras et al., 2021]

\[ R_C(f, x) = \max_{\|v\| \leq \epsilon} W_C(f(x + v), f(x)) \]

- Use (virtual) adversarial examples to promote a better generalization of DNN (close samples should have close predictions) [Miyato et al., 2018].
- The ground metric \( C \) in regularization \( R_C(f, x) \) encodes pairwise class relations and will promote smooth/complex between them.
- State of the art performance for learning with label noise when using semantic relations between the classes for \( C \) (word2vec).
Outline

Introduction

Mapping with optimal transport
  Optimal transport mapping estimation
  Optimal transport for domain adaptation

Learning from histograms with Optimal Transport
  Unsupervised learning
  Supervised learning

Learning from empirical distributions with Optimal Transport
  Unsupervised learning
  Supervised learning and domain adaptation

Conclusion
Empirical distributions A.K.A datasets

\[ \mu = \sum_{i=1}^{n} a_i \delta_{x_i}, \quad x_i \in \Omega, \quad \sum_{i=1}^{n} a_i = 1 \]

Empirical distribution

- Two realizations never overlap.
- Training base of all machine learning approaches.
- How to measure discrepancy?
- Maximum Mean Discrepancy ($\ell_2$ after convolution).
- Wasserstein distance.
**OT for modeling cell development**

Principle [Schiebinger et al., 2019]

- Developmental trajectories of cells from stem cells to more specialized.
- Cell populations are samples at different times with scRNA-seq.
- Optimal transport can be used to find mapping/correspondances between across population measurements.
- Unbalanced OT is used to model cellular growth and death rates.
Principle [Schiebinger et al., 2019]

- Developmental trajectories of cells from stem cells to more specialized.
- Cell populations are samples at different times with scRNA-seq.
- Optimal transport can be used to find mapping/correspondences between across population measurements.
- Unbalanced OT is used to model cellular growth and death rates.
Generative Adversarial Networks (GAN) [Goodfellow et al., 2014]

\[
\min_G \max_D E_{x \sim \mu_d}[\log D(x)] + E_{z \sim \mathcal{N}(0,1)}[\log(1 - D(G(z)))]
\]

- Learn a generative model \( G \) that outputs realistic samples from data \( \mu_d \).
- Learn a classifier \( D \) to discriminate between the generated and true samples.
- Make those models compete (Nash equilibrium [Zhao et al., 2016]).
Generative Adversarial Networks (GAN) [Goodfellow et al., 2014]

\[
\min_G \max_D \mathbb{E}_{x \sim \mu_d} \left[ \log D(x) \right] + \mathbb{E}_{z \sim \mathcal{N}(0, I)} \left[ \log (1 - D(G(z))) \right]
\]

- Learn a generative model \( G \) that outputs realistic samples from data \( \mu_d \).
- Learn a classifier \( D \) to discriminate between the generated and true samples.
- Make those models compete (Nash equilibrium [Zhao et al., 2016]).
- Generator space has semantic meaning [Radford et al., 2015].
- But extremely hard to train (vanishing gradients).
Wasserstein Generative Adversarial Networks (WGAN)

Wasserstein GAN [Arjovsky et al., 2017]

\[
\min_G \ W_1^1(G \# \mu_z, \mu_d),
\]

- Minimizes the Wasserstein distance between the data \( \mu_d \) and the generated data \( G \# \mu_z \) where \( \mu_z = \mathcal{N}(0, \mathbf{I}) \).
- No vanishing gradients! Better convergence in practice.
- Wasserstein in the dual (separable w.r.t. the samples).

\[
\min_G \ \sup_{\phi \in \text{Lip}^1} E_{x \sim \mu_d}[\phi(x)] - E_{z \sim \mu_z}[\phi(G(z))]
\]

- \( \phi \) is a neural network that acts as an *actor critic*
Neural network belonging to Lip\(^1\)?

- Not really! [Arjovsky et al., 2017] proposes to do weight clipping that forces an upper bound on the Lipschitz constant.
- It is actually the supremum over K-Lipschitz functions that is approximated by a neural network

\[
\max_{f \in \text{NN class}} L_{WGAN}(f, G) \leq \sup_{\|\phi\|_L \leq K} L_{WGAN}(\phi, G) = K \cdot W_1^1(G(z), \mu_d)
\]

- Actually not equivalent to solve the optimal transport, but gradients are aligned.

**Improved WGAN [Gulrajani et al., 2017]**

\[
\min_G \sup_{f \in \text{NN class}} \mathbb{E}_{x \sim \mu_d}[f(x)] - \mathbb{E}_{z \sim \mu_z}[f(G(z))] + \lambda \mathbb{E}_{x \sim \mu_d}[(\|\nabla f(x)\| - 1)^2]
\]

Relaxation of the constraint (for \(W_1\) the gradient of the potential is 1 almost everywhere).
Wasserstein GAN loss on Biomedical images

Reconstructing low dose CT images [Yang et al., 2018]

\[
\min_G W_1^1(G \# \mu_l, \mu_f) + \lambda_1 E_{x \sim \mu_l} [\|VGG(x_l) - VGG(G(x_l))\|^2],
\]

(4)

- Use Wasserstein to make reconstruction of quarter dose CT images ($\mu_l$) similar to high dose (resolution) CT images ($\mu_f$).
- Perceptual loss based on VGG [Simonyan and Zisserman, 2014] embedding to keep image information.
Reconstructing low dose CT images [Yang et al., 2018]

\[
\min_G W_1^1(G \# \mu_l, \mu_f) + \lambda_1 E_{x \sim \mu_l} [\|VGG(x_l) - VGG(G(x_l))\|^2],
\]

- Use Wasserstein to make reconstruction of quarter dose CT images ($\mu_l$) similar to high dose (resolution) CT images ($\mu_f$).
- Perceptual loss based on VGG [Simonyan and Zisserman, 2014] embedding to keep image information.
Reconstructing low dose CT images [Yang et al., 2018]

\[
\min_G W_1^1(G\#\mu_l, \mu_f) + \lambda_1 E_{x \sim \mu_l} [\|VGG(x_l) - VGG(G(x_l))\|^2],
\]

(4)

- Use Wasserstein to make reconstruction of quarter dose CT images (\(\mu_l\)) similar to high dose (resolution) CT images (\(\mu_f\)).
- Perceptual loss based on VGG [Simonyan and Zisserman, 2014] embedding to keep image information.
Wasserstein Discriminant Analysis (WDA)

\[
\max_{P \in S} \frac{\sum_{c,c'>c} W_\lambda(PX^c, PX^{c'})}{\sum_c W_\lambda(PX^c, PX^c)} \tag{5}
\]

- \(X^c\) are samples from class \(c\).
- \(P\) is an orthogonal projection;

- Converges to Fisher Discriminant when \(\lambda \to \infty\).
- Non parametric method that allows nonlinear discrimination.
- Problem solved with gradient ascent in the Stiefel manifold \(S\).
- Gradient computed using automatic differentiation of Sinkhorn algorithm.
Wasserstein Discriminant Analysis (WDA)

\[
\max_{P \in S} \frac{\sum_{c,c'} W_\lambda(PX^c, PX^{c'})}{\sum_c W_\lambda(PX^c, PX^c)} \quad (5)
\]

- \(X^c\) are samples from class \(c\).
- \(P\) is an orthogonal projection;

- Converges to Fisher Discriminant when \(\lambda \to \infty\).
- Non parametric method that allows nonlinear discrimination.
- Problem solved with gradient ascent in the Stiefel manifold \(S\).
- Gradient computed using automatic differentiation of Sinkhorn algorithm.
Wasserstein Discriminant Analysis (WDA)

\[
\max_{P \in S} \frac{\sum_{c, c' > c} W_\lambda(PX^c, PX^{c'})}{\sum_c W_\lambda(PX^c, PX^c)}
\]  \hspace{1cm} (5)

\begin{itemize}
  \item \(X^c\) are samples from class \(c\).
  \item \(P\) is an orthogonal projection;
  \item Converges to Fisher Discriminant when \(\lambda \to \infty\).
  \item Non parametric method that allows nonlinear discrimination.
  \item Problem solved with gradient ascent in the Stiefel manifold \(S\).
  \item Gradient computed using automatic differentiation of Sinkhorn algorithm.
\end{itemize}
Data imputation with Optimal Transport

Missing Data imputation [Muzellec et al., 2020]

\[
\min_{X^{imp}} \mathbb{E}[SD(\mu_m(\hat{X}), \mu_m(X))] 
\]

- \(X \odot M\) is the partially observed data with binary mask \(M\).
- \(\hat{X} = X \odot M + (1 - M) \odot X^{imp}\) is the data imputed by \(X^{imp}\)
- \(\mu_m(X)\) is a minibatch of \(X\), expectation is taken w.r.t. the minibatches.
- Out of sample imputation with model [Muzellec et al., 2020, Algo 2 & 3]
- Optimizing minibatch Wasserstein is a classical approach [Fatras et al., 2020].
Domain adaptation with Wasserstein distance

Domain adaptation for deep learning [Shen et al., 2018]

- Modern DA aim at aligning source and target in the deep representation:
  DANN [Ganin et al., 2016], MMD [Tzeng et al., 2014], CORAL [Sun and Saenko, 2016].

- Wasserstein distance (WGAN loss [Arjovsky et al., 2017]) used as objective for the adaptation [Shen et al., 2018].
Joint Distribution Optimal Transport for DA

Learning with JDOT [Courty et al., 2017b]

\[
\min_f \left\{ W_1(\hat{P}_s, \hat{P}_t^f) = \inf_{\gamma \in \Pi} \sum_{i,j} D(x_i^s, y_i^s; x_j^t, f(x_j^t)) \gamma_{ij} \right\}
\]

- \(\hat{P}_t^f = \frac{1}{N_t} \sum_{i=1}^{N_t} \delta_{x_i^t, f(x_i^t)}\) is the proxy joint feature/label distribution.
- \(D(x_i^s, y_i^s; x_j^t, f(x_j^t)) = \alpha \|x_i^s - x_j^t\|^2 + \mathcal{L}(y_i^s, f(x_j^t))\) with \(\alpha > 0\).
- We search for the predictor \(f\) that better align the joint distributions.
- OT matrix does the label propagation (no mapping).
- JDOT can be seen as minimizing a generalization bound.
**JDOT for large scale deep learning**

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 [Fatras et al., 2020].
- Scales to large datasets and estimate a representation for both domains.
- TSNE projections of embeddings (MNIST → MNIST-M).

\[
\text{Loss (9):} \quad L_s(y_i^s, f(g(x_i^s))) + \\
\quad \left( \|g(x_i^s) - g(x_j^t)\|^2 \right) + \\
\quad L_t(y_i^s, f(g(x_i^s)))
\]
"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 [Fatras et al., 2020].
- Scales to large datasets and estimate a representation for both domains.
- TSNE projections of embeddings (MNIST $\rightarrow$ MNIST-M).
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 [Fatras et al., 2020].
- Scales to large datasets and estimate a representation for both domains.
- TSNE projections of embeddings (MNIST $\rightarrow$ MNIST-M).
Outline

Introduction

Mapping with optimal transport
- Optimal transport mapping estimation
- Optimal transport for domain adaptation

Learning from histograms with Optimal Transport
- Unsupervised learning
- Supervised learning

Learning from empirical distributions with Optimal Transport
- Unsupervised learning
- Supervised learning and domain adaptation

Conclusion
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.
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.
- Gromov Wasserstein.
- Solvers for Numpy/Pytorch/Jax/tensorflow/Cupy

Link for the practical session:
https://github.com/PythonOT/OTML_course_2022

**Input convex neural networks.**


**Wasserstein gan.**


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

*SISC.*


**Geodesic pca in the wasserstein space by convex pca.**

Domain adaptation with optimal transport improves eeg sleep stage classifiers.
In *2018 International Workshop on Pattern Recognition in Neuroimaging (PRNI)*, pages 1–4. IEEE.

Learning wasserstein embeddings.

Joint distribution optimal transportation for domain adaptation.
In *Neural Information Processing Systems (NIPS)*.

Optimal transport for domain adaptation.
*Pattern Analysis and Machine Intelligence, IEEE Transactions on.*

**Sinkhorn distances: Lightspeed computation of optimal transportation.**


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


**Wasserstein adversarial regularization for learning with label noise.**

*Pattern Analysis and Machine Intelligence, IEEE Transactions on.*


**Learning with minibatch wasserstein : asymptotic and gradient properties.**

In *International Conference on Artificial Intelligence and Statistics (AISTAT).*
  
  Regularized discrete optimal transport.
  
  SIAM Journal on Imaging Sciences, 7(3).

  
  Optimal spectral transportation with application to music transcription.
  

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

  
  Learning with a wasserstein loss.
  

**Domain-adversarial training of neural networks.**


**Domain adaptation using optimal transport: application to prostate cancer mapping.**


**Optimal transport applied to transfer learning for p300 detection.**


**Generative adversarial nets.**


**Improved training of wasserstein gans.**


**Wasserstein-2 generative networks.**


**Optimal transport mapping via input convex neural networks.**


**Quantitative stability of optimal transport maps and linearization of the 2-wasserstein space.**

In *International Conference on Artificial Intelligence and Statistics*, pages 3186–3196. PMLR.

**Virtual adversarial training: a regularization method for supervised and semi-supervised learning.**


**Wasserstein style transfer.**


**Missing data imputation using optimal transport.**


**Regularity as regularization: Smooth and strongly convex brenier potentials in optimal transport.**

In *International Conference on Artificial Intelligence and Statistics*, pages 1222–1232. PMLR.

Poisson image editing.

*ACM Trans. on Graphics*, 22(3).


Mapping estimation for discrete optimal transport.

In *Neural Information Processing Systems (NIPS)*.


Entropic estimation of optimal transport maps.


Unsupervised representation learning with deep convolutional generative adversarial networks.


Co-optimal transport.
In *Neural Information Processing Systems (NeurIPS)*.


xdawn algorithm to enhance evoked potentials: application to brain–computer interface.


Fast dictionary learning with a smoothed wasserstein loss.
In *Proceedings of the 19th International Conference on Artificial Intelligence and Statistics*, pages 630–638.


Nonnegative matrix factorization with earth mover’s distance metric for image analysis.

**Optimal-transport analysis of single-cell gene expression identifies developmental trajectories in reprogramming.**


**Large-scale optimal transport and mapping estimation.**


**Principal geodesic analysis for probability measures under the optimal transport metric.**


**Wasserstein distance guided representation learning for domain adaptation.**

In *AAAI Conference on Artificial Intelligence*. 

**Very deep convolutional networks for large-scale image recognition.**


**Deep domain confusion: Maximizing for domain invariance.**


**Semi-supervised optimal transport for heterogeneous domain adaptation.**

In *IJCAI*, pages 2969–2975.

**Low-dose ct image denoising using a generative adversarial network with wasserstein distance and perceptual loss.**


**Simultaneous ground metric learning and matrix factorization with earth mover’s distance.**

In *Pattern Recognition (ICPR), 2014 22nd International Conference on*, pages 3690–3695.


**Energy-based generative adversarial network.**