Optimal transport with Laplacian regularization
Applications to domain adaptation and shape matching

R. Flamary, N. Courty, A. Rakotomamonjy, D. Tuia

NIPS 2014 Workshop on Optimal Transport & Machine Learning

December 13, 2014
Objective

Approach

▶ Investigate the use of optimal transport to transport the samples from one distribution to another.
▶ Promote graph regularization on the transported samples.

Applications:

**Domain adaptation**
Transport samples to the new domain then train classifier.

**Shape matching**
Align meshes in computer graphics using OT.
Optimal transport for discrete distribution

Dataset and discrete distributions

\[ \mu_s = \sum_{i=1}^{n_s} p_i^s \delta_{x_i^s}, \quad \mu_t = \sum_{i=1}^{n_t} p_i^t \delta_{x_i^t} \]  

1

- \( \delta_{x_i} \) is the Dirac at location \( x_i \in \mathbb{R}^d \).
- \( p_i^s \) and \( p_i^t \) are probability masses.
- \( \sum_{i=1}^{n_s} p_i^s = \sum_{i=1}^{n_t} p_i^t = 1 \)
- In this work \( p_i^s = \frac{1}{n_s} \) and \( p_i^t = \frac{1}{n_t} \).
- Samples stored in matrices

\[ X_s = [x_1^s, \ldots, x_{n_s}^s]^\top \in \mathbb{R}^{n_s \times d} \]
\[ X_t = [x_1^t, \ldots, x_{n_t}^t]^\top \in \mathbb{R}^{n_t \times d} \]
Regularized optimal transport

Optimization problem

\[
\gamma_0 = \arg \min_{\gamma \in \mathcal{P}} \langle \gamma, C \rangle_F + \lambda \Omega(\gamma)
\]

where \( C \) is a transportation cost matrix, \( \Omega(\cdot) \) is a regularization term and

\[
\mathcal{P} = \left\{ \gamma \in (\mathbb{R}^+)^{n_s \times n_t} \mid \gamma 1_{n_t} = \mu_s, \gamma^T 1_{n_s} = \mu_t \right\}
\]

For classical OT, the regularization term is \( \Omega(\cdot) = 0 \).
Regularized optimal transport

Optimization problem

\[ \gamma_0 = \arg \min_{\gamma \in \mathcal{P}} \langle \gamma, C \rangle_F + \lambda \Omega(\gamma) \]  \hspace{1cm} (2)

where \( C \) is a transportation cost matrix, \( \Omega(\cdot) \) is a regularization term and

\[ \mathcal{P} = \left\{ \gamma \in (\mathbb{R}^+)^{n_s \times n_t} \mid \gamma \mathbf{1}_{n_t} = \mu_s, \gamma^T \mathbf{1}_{n_s} = \mu_t \right\} \]

For classical OT, the regularization term is \( \Omega(\cdot) = 0 \).
Choice of the regularization

**OT\textsubscript{sinkhorn} : Entropy based regularization (Cuturi [1])**

- Information theory based regularization:
  \[
  \Omega(\gamma) = \sum_{i,j} \gamma_{i,j} \log(\gamma_{i,j})
  \]

- Shrinkage effect on the transported samples for large regularization.
- Efficient solver but sometimes numerical problems.

**LOT : Laplacian regularization (Ferradans et al. [2])**

- Encode graph based knowledge in the optimization problem.
- Regularization focus on the displacement of the samples during interpolation.
- Can be expressed as a Linear Program (LP) or as a Quadratic Program (QP).
- Use interior point Frank-Wolfe algorithm to solve the QP.
Transporting the discrete samples (1)

Interpolation $t=0.5$ for LP
Interpolation $t=0.5$ for Sinkhorn

Interpolation between discrete distributions

- $\gamma_0$ defines the distribution of the mass of each source sample onto the target samples.
- Symmetric interpolation between source ($t = 0$) and target ($t = 1$):

$$\mu(t) = \sum_{i=1, j=1}^{n_s, n_t} \gamma_{i,j} \delta(1-t)x_i^s + tx_j^t$$

The number of dirac in the intermediate $0 < t < 1$ interpolation is the number of nonzero coefficients in $\gamma$. 
Transporting the discrete samples (2)

Interpolation samples between distributions

▶ The original mass of each source sample is spread onto the target samples as defined by $\gamma_0$.
▶ Position of the transported samples:

$$\hat{X}_s = \text{diag}(\gamma_0 1_{n_t})^{-1} \gamma_0 X_t \quad \text{and} \quad \hat{X}_t = \text{diag}(\gamma_0^\top 1_{n_s})^{-1} \gamma_0^\top X_s. \quad (4)$$

▶ Transformed sample at the center of mass of its transported distribution.
▶ For uniform distributions $p_i^s = \frac{1}{n_s}$ we have $\hat{X}_s = n_s \gamma_0 X_t$. 

Interpolation s $\rightarrow$ t for LP

Interpolation s $\rightarrow$ t for Sinkhorn
Two flavors of Laplacian regularization

Similarity matrix $S^s$ defines a graph of similarity between source samples.

Regularizing the sample displacement [2]

- Similar samples should have similar displacement.
- Rigid displacements in clusters for large regularization.
- LOT$_{disp}$.

Regularizing the sample position

- Similar samples should be transported to similar positions.
- Shrinkage to the center of mass of clusters for large regularization.
- **Our contribution**.
- LOT$_{pos}$. 
Graph regularization for the sample position

- We want similar samples to have similar positions after transport:

\[
\Omega_{pos}(\gamma) = \frac{1}{N^2} \sum_{i,j} S_{i,j}^s \| \hat{x}_i^s - \hat{x}_j^s \|^2
\]

- With uniform distributions the transported sample \( \hat{x}_i^s \) is linear w.r.t. \( \gamma \).
- Regularization term is quadratic w.r.t. \( \gamma \).
Laplacian regularization for sample position (2)

Reformulation of the regularization

- The regularization term can be expressed in matrix form as:

\[
\Omega_{pos}(\gamma) = \frac{1}{N_s^2} \sum_{i,j} S_{i,j}^s \| \hat{x}_i^s - \hat{x}_j^s \|_2^2 = \text{Tr}(X_t^\top \gamma^\top L_s \gamma X_t)
\]

where \( L_s = \text{diag}(S_s 1) - S_s \) is the Laplacian of the graph \( S_s \).

- Gradient of the trace is easy to obtain during the optimization.

Symmetric regularization

- Regularization \( \Omega_{pos}(\cdot) \) promotes only similarity in the transported source samples.

- We can also use a symmetric regularization of the form:

\[
\Omega_{pos}(\gamma) = \frac{1 - \alpha}{N_s^2} \sum_{i,j} S_{i,j}^s \| \hat{x}_i^s - \hat{x}_j^s \|_2^2 + \frac{\alpha}{N_t^2} \sum_{i,j} S_{i,j}^t \| \hat{x}_i^t - \hat{x}_j^t \|_2^2
\]

where \( 0 \leq \alpha \leq 1 \) is a term that weighs the importance of the source/target regularizations.
Example of LOT$^{pos}$

Sim. graph with $S_{i,j}^s > 0$

Small $\lambda$

Large $\lambda$

Discussion

- Similarity graph $S^s$ obtained by a Gaussian kernel with threshold (left).
- The transported samples are illustrated in white for a small regularization (center) and a large regularization (right).
- Clusters and local structures are promoted.
- Per-cluster shrinkage for large regularization.
Laplacian regularization for sample displacement

Graph regularization for the sample displacement

- \( \hat{x}_i^s - x_i^s \) is the displacement of source sample \( x_i^s \) during transport.
- We want similar samples to have similar displacements:

\[
\Omega(\gamma) = \frac{1}{N_s^2} \sum_{i,j} S_{i,j}^s \| (\hat{x}_i^s - x_i^s) - (\hat{x}_j^s - x_j^s) \|^2
\]

- Quadratic regularization term.
Reformulation of the regularization term

- The regularization term can be expressed in matrix form as:

\[
\Omega_{\text{disp}}(\gamma) = \text{Tr}(\mathbf{X}_t^\top \gamma^\top \mathbf{L}_s \gamma \mathbf{X}_t) + \left\langle \gamma, -\mathbf{N}_s (\mathbf{L}_s + \mathbf{L}_s^\top) \mathbf{X}_s \mathbf{X}_t^\top \right\rangle_F + c_s
\]

with \(c_s = \text{Tr}(\mathbf{X}_s^\top \mathbf{L}_s \mathbf{X}_s)\) a constant w.r.t. \(\gamma\).

- Similarly to \(\Omega_{\text{pos}}(\cdot)\), one can use a symmetric regularization for the displacement.
Example of LOT$_{disp}$

Sim. graph with $S^{s}_{i,j} > 0$
Small $\lambda$
Large $\lambda$

Discussion

- Similarity graph $S^s$ obtained by a Gaussian kernel with threshold (left).
- The transported samples displacements are illustrated in red for a small regularization (center) and a large regularization (right).
- Clusters and local structures are promoted.
- Per-cluster rigid translation for large regularization.
Optimization with Frank-Wolfe Algorithm

- Resulting optimization problem for both Laplacian regularization is a quadratic Program (QP).
- [2] proposed to use a Frank-Wolfe Algorithm to solve the problem.

Algorithm for symmetric regularization of the sample position

0. Initialize $k = 0$ and $\gamma^0 \in \mathcal{P}$.
1. Compute the solution of the linear problem $\gamma^* = \arg \min_{\gamma \in \mathcal{P}} \langle \gamma, C_k \rangle_F$ with

$$C_k = C + (1 - \alpha)(L + L^\top)\gamma X_t X_t^\top + \alpha X_s X_s^\top \gamma (\tilde{L} + \tilde{L}^\top)$$

2. Find the optimal step $0 \leq \alpha^k \leq 1$ with descent direction $\Delta \gamma = \gamma^* - \gamma^k$ such that

$$\alpha^k = -\frac{1}{2} \frac{\langle \Delta \gamma, C \rangle_F + \lambda_s \text{Tr}(X_t^\top \Delta \gamma^\top (L_s + L_s^\top)\gamma^k X_t) + \lambda_t \text{Tr}(X_s^\top \Delta \gamma (L_t + L_t^\top)\gamma^k X_s)}{\lambda_s \text{Tr}(X_t^\top \Delta \gamma^\top L_s \Delta \gamma X_t) + \lambda_t \text{Tr}(X_s^\top \Delta \gamma L_t \Delta \gamma^\top X_s)}$$

3. $\gamma^{k+1} \leftarrow \gamma^k + \alpha^k \Delta \gamma$, set $k \leftarrow k + 1$ and go to step 1.
Numerical experiments

Domain adaptation (DA)

- Classification problem (blue VS red).
- Classical simulated *two-moons* problem.
- Increasing adaptation difficulty.
- Comparison with state of the art DA.

Non-rigid shape matching

- Register 3D shapes.
- Use the FAUST dataset[3].
- Preliminary matching results.
- Mean average error for transported vertices.
Domain adaptation problem

<table>
<thead>
<tr>
<th></th>
<th>10°</th>
<th>20°</th>
<th>30°</th>
<th>40°</th>
<th>50°</th>
<th>70°</th>
<th>90°</th>
</tr>
</thead>
<tbody>
<tr>
<td>SVM (no adapt)</td>
<td>0.000</td>
<td>0.104</td>
<td>0.24</td>
<td>0.312</td>
<td>0.4</td>
<td>0.764</td>
<td>0.828</td>
</tr>
<tr>
<td>DASVM [4]</td>
<td>0.000</td>
<td>0.000</td>
<td>0.259</td>
<td>0.284</td>
<td>0.334</td>
<td>0.747</td>
<td>0.82</td>
</tr>
<tr>
<td>PBDA [5]</td>
<td>0.000</td>
<td>0.094</td>
<td>0.103</td>
<td>0.225</td>
<td>0.412</td>
<td>0.626</td>
<td>0.687</td>
</tr>
<tr>
<td>OT LP</td>
<td>0.000</td>
<td>0.000</td>
<td>0.031</td>
<td>0.102</td>
<td>0.166</td>
<td>0.292</td>
<td>0.441</td>
</tr>
<tr>
<td>OT (_{\text{sinkhorn}})</td>
<td>0.000</td>
<td>0.000</td>
<td>0.000</td>
<td>0.000</td>
<td>0.013</td>
<td>0.202</td>
<td>0.386</td>
</tr>
<tr>
<td>LOT(_{\text{pos}})</td>
<td>0.000</td>
<td>0.000</td>
<td>0.000</td>
<td>0.000</td>
<td>0.000</td>
<td>0.022</td>
<td>0.152</td>
</tr>
<tr>
<td>LOT(_{\text{disp}})</td>
<td>0.000</td>
<td>0.000</td>
<td>0.000</td>
<td>0.000</td>
<td>0.000</td>
<td>0.067</td>
<td>0.384</td>
</tr>
</tbody>
</table>

Discussion

- Non-linear classification problem handled by a SVM with Gaussian kernel with parameters set by k-fold validation.
- Regularization parameters set empirically.
- Mean error rate over 10 samplings reported in the table.
- Good performances of Optimal transport for domain adaptation.
- Better performance of LOT\(_{\text{pos}}\) for this problem.
Domain adaptation problem (2)

- Domain adaptation problem for different rotations.
- Decision function for LP transport (no regularization).
- Decision function for entropy based regularization \( \text{OT}_{\text{sinkhorn}} \).
- Decision function for position based laplacian regularization \( \text{LOT}_{\text{pos}} \).
Domain adaptation problem (2)

- Domain adaptation problem for different rotations.
- **Decision function for LP transport (no regularization).**
- Decision function for entropy based regularization ($\text{OT}_{\text{sinkhorn}}$).
- Decision function for position based laplacian regularization ($\text{LOT}_{\text{pos}}$).
Domain adaptation problem (2)

- Domain adaptation problem for different rotations.
- Decision function for LP transport (no regularization).
- Decision function for entropy based regularization ($\text{OT}_{\text{sinkhorn}}$).
- Decision function for position based laplacian regularization ($\text{LOT}_{\text{pos}}$).
Domain adaptation problem (2)

- Domain adaptation problem for different rotations.
- Decision function for LP transport (no regularization).
- Decision function for entropy based regularization ($\text{OT}_{\text{sinkhorn}}$).
- **Decision function for position based laplacian regularization ($\text{LOT}_{\text{pos}}$).**
Shape matching problem

- We want to match different shapes from the FAUST dataset.
- Ground truth available (exact matching between watertight meshes).
- Performance measure: distance of the transported vertices to the true target vertices (mean and max).
- 3D plot of the vertices assignments.
Shape matching problem (2)

- Mean error in cm (max errors in cm).
- Laplace regularization slightly better than classic OT.
- $\text{LOT}_{pos}$ and $\text{LOT}_{disp}$ have similar performances in average.
- Encouraging preliminary results.
Mean error in cm (max errors in cm).

Laplace regularization slightly better than classic OT.

LOT$_{pos}$ and LOT$_{disp}$ have similar performances in average.

Encouraging preliminary results.
Shape matching problem (3)

Two matching examples

- Lines show vertex displacement.
- Vertices are sorted: perfect assignment means a diagonal matrix.
- Left: #1 ⇒ #2, Error = 7.5 (37.2)
- Right: #3 ⇒ #6, Error = 11.0 (46.9).
Conclusion

Optimal transport with Laplacian regularization

- When data has a graph structure, regularize to keep it during the transport.
- Two flavors of Laplacian regularization (depends on the problem).
- Use Frank-Wolfe to solve the problem (efficient LP solvers, early stopping).
- Encouraging results on two applications.

Next steps

- More numerical experiments on real life datasets.
- Use label during graph computation for domain adaptation.
- Large scale optimization procedure.
- Find the regularization parameters automatically.
References

M. Cuturi.
Sinkhorn distances: Lightspeed computation of optimal transportation.

S. Ferradans, N. Papadakis, G. Peyré, and J-F. Aujol.
Regularized discrete optimal transport.

Federica Bogo, Javier Romero, Matthew Loper, and Michael J. Black.
FAUST: Dataset and evaluation for 3D mesh registration.
In *Proceedings IEEE Conf. on Computer Vision and Pattern Recognition (CVPR)*,
Piscataway, NJ, USA, June 2014. IEEE.

L. Bruzzone and M. Marconcini.
Domain adaptation problems: A dasvm classification technique and a circular
validation strategy.
*Pattern Analysis and Machine Intelligence, IEEE Transactions on*, 32(5):770–787,
May 2010.

A PAC-Bayesian Approach for Domain Adaptation with Specialization to Linear
Classifiers.
In *ICML*, pages 738–746, Atlanta, USA, 2013.