Transport and other lagrangian Transforms

Soheil Kolouri and Gustavo K. Rohde

HRL Laboratories, University of Virginia
Outline

- Introduction: intuition, new model for sensor data
- New low-level transforms
  - Linear optimal transport (LOT): discrete, continuous
  - Cumulative Distribution Transform (CDT): 1d signals
  - Radon Cumulative Distribution Transform (R-CDT): 2d signals
- Basic properties
  - Translation, scaling, composition
  - Linear separation
  - Kernel point of view
Introduction
Intuition: Variations in signal intensity and location

faces

Cells, nuclei

brains

galaxies
Linear signal models

- Fourier, Wavelets, frames, etc.: $s \sim \sum_k c_k \phi_k$.
  - $s$: signal
  - $\phi_k$ ‘basis’ elements (functions, vectors)
  - $c_k$ coefficients of representation
- Sparse representation: $\min_c \|c\|_1$, s.t. $s(t) = \sum_k c_k \phi_k(t)$.
- Correspondence between model and data assessed at fixed locations → Eulerian point of view

Today: nonlinear signal model

- $s(t) \sim f'(t)s(f(t))$
- $f(t)$ a transport map/plan
- ‘Carries’ intensity around → Lagrangian point of view
Linear optimal transport (LOT)
**Problem**

Consider the problem of performing pattern recognition (e.g. classification, clustering) on a set of $N$ signals/images using $\text{OT} \rightarrow O(N^2)$ OT problems.

**One possible solution**

Compute transports to a fixed reference, then approximate OT distance by euclidean distance between transport maps.

$\rightarrow O(N)$ OT problems

plus $O(N)$ Euclidean distances

Geometric interpretation

- Let $d\sigma = I_0dx$, $d\mu = I_1dx$, and $d\nu = I_2dx$.
- Define OT maps s.t.: $\det(D\psi_1(x))I_1(\psi_1(x)) = I_0(x)$, and $\det(D\psi_2(x))I_2(\psi_1(x)) = I_0(x)$.
- $P(\mu) = \psi_1 - Id$, and $P(\nu) = \psi_2(x) - Id$ can be viewed as azimuthal projections.

Linear optimal transport (LOT)

- $d^2_W(\mu, \sigma) = \int |\psi_1(x) - x|^2 I_0(x)dx$
- $d^2_{LOT}(\nu, \mu) = \int |P(\mu) - P(\nu)|^2 I_0(x)dx = \|P(\mu) - P(\nu)\|_\sigma^2$
- $d^2_{LOT}(\mu, \sigma) = \|P(\mu)\|_\sigma^2 = d^2_W(\mu, \sigma)$.
Geometric interpretation

- ‘Euclidean’ projection \( P(\mu) = \psi_1 - Id \)
- projection is invertible.
**LOT transform (forward)**

- Step 1: \( \min_{f_1} \int |f_1(x) - x|^2 I_0(x) dx \), s.t. \( \det(Df_1(x)) I_1(f_1(x)) = I_0(x) \)
- LOT transform \( \hat{I} := (f_1 - Id) \sqrt{I_0} \)

**LOT transform (inverse)**

- Inverse LOT: \( I_1(x) = \det(Df_1^{-1}) I_1 \circ f^{-1} \)
- \( f_1^{-1} \): inverse map of \( f_1 \)
‘Discrete’ LOT

- Find particle approximation such that $\mu = \sum_{i=1}^{N} m_i \delta_{x_i}$, and
  $\min_{m_i, x_i} W_2(I_1, \mu)$
- Similar for $\sigma = \sum_{k=1}^{N} q_k \delta_{z_k}$, $\min_{q_k, z_k} W_2(I_1, \mu)$
- Compute discrete (histogram) OT (earth mover’s) distance via linear programming:

$$\min_f \sum_{i} \sum_{k} c(x_i, z_k) f_{i,k}$$

s.t. $\sum_{i} f_{i,k} = q_k$, $\sum_{k} f_{i,k} = m_i$
'Discrete' LOT embedding (transform)

- Compute average position for where each particle will end up
  \[ \bar{x}_k = \frac{1}{q_k} \sum_{i=1}^{N \mu} f_{k,i} x_i \]
- Embedding given by \[ x = \left( \sqrt{q_1 \bar{x}_1} \cdots \sqrt{q_N \sigma \bar{x}_N} \right)^T \]

LOT distance

\[ d_{\alpha \text{LOT}, \sigma} (\mu, \nu)^2 = \sum_{k=1}^{N \sigma} q_k |\bar{x}_k - \bar{y}_k|^2. \]

For dataset of \( N \) images
\( O(N) \) transport problems required.

LOT distance as an approximate OT distance:

- **A**: percentage error between digital image and particle approximation
- **B**: percentage difference between LOT and OT distances
’Continuous’ LOT

**Forward**
Given image $I_1$, and reference $I_0$, define continuous LOT transform by variational minimization (see earlier)

- Step 1: $\min \int_{\Omega} |f(x) - x| I_0^2(x) \, dx \text{ s.t. } \det(Df(x)) I_1(f(x)) = I_0(x)$
- Step 2: LOT defined by $\hat{I}_1 = (f(x) - x) \sqrt{I_0(x)}$

**Inverse**
Inverse transform given by

$$I_1(x) = \det |D f^{-1}(x)| I_0(f^{-1}(x))$$

- $f^{-1}$ and $\det |D f^{-1}(x)|$ approximated numerically (interpolation).

Kolouri et al., Pattern Recognition, 2016.
Matlab code

```
1 I0 = imread('source_face.jpg')
2 I1 = imread('target_face.jpg')
3 [f, g, D, u, v] = LOT(I0, I1); % Forward transform f and g identify the optimal transport map, and u and v identify the optimal transport displacement fields: f = X - u, g = Y - v
4 I1_hat = iLOT(u, v, I0); % iLOT calculates the inverse transform
```
Cumulative Distribution Transform (CDT)
Recall: for 1D distributions, OT match is unique \(\rightarrow\) no optimization required.

Let \(I_0\) and \(I_1\) be two positive definite functions s.t. \(\int I_0 = \int I_1 = 1\).

Compute match:

\[
\int_{-\infty}^{x} I_0(\tau) d\tau = \int_{-\infty}^{f(x)} I_1(\tau) d\tau \rightarrow I_0(x) = f'(x)I_1(f(x))
\]

The integral equation above can be computed using numerical integration.

**Forward Cumulative Distribution Transform (CDT):**

\[
\hat{I}_1(u) = (f(u) - u)\sqrt{I_0(u)}
\]

**Inverse CDT:**

\[
I_1(x) = (f^{-1})'(x)I_0(f^{-1}(x))\sqrt{I_0(x)}
\]

Cumulative distribution transform (CDT)

For a signal database:

\[ I_i = (f_i - \text{Id}) \sqrt{I_0} \]

\[ I_i = (f_i^{-1})' (I_0 \circ f_i^{-1}) \]
Matlab code:

```matlab
1 f = @(x, mu, sigma) (1/sqrt(2*pi*sigma^2)) * exp(-((x-mu).^2)/(2*sigma^2));
2 g = @(x, mu, sigma) 0.5*f(x, mu-3, sigma) + 0.5*f(x, mu+3, sigma);
3 x = -15:0.1:15;
4 p = f(x, mu(i), 1)';
5 q = g(x, mu(i), 1)';
6 template = ones(size(p, 1), 1); % Template is the uniform distribution
7 template = template/sum(template);
8 p_tilde = CDT(template+eps, p+eps); % Calculate CDT Transform
9 q_tilde = CDT(template+eps, q+eps); % Calculate CDT Transform
10 p_hat = iCDT(template, p_tilde); % Inverse CDT Transform
11 g_hat = iCDT(template, q_tilde); % Inverse CDT Transform
```
Introduction
Low level transforms
Properties

Linear optimal transport
Cumulative Distribution Transform
Radon Cumulative Distribution Transform

Radon Cumulative Distribution Transform (R-CDT)
Radon cumulative distribution transform (R-CDT):

Idea: combine radon transform with cumulative distribution transform

Radon transform:

- Forward and inverse:

\[
\mathcal{RI}(t, \theta) = \int_{-\infty}^{\infty} \int_{-\infty}^{\infty} I(x, y) \delta(t - x \cos(\theta) - y \sin(\theta)) dx dy \\
I(x, y) = \int_{0}^{\pi} (\mathcal{RI}(., \theta) \ast w(\cdot)) \circ (x \cos(\theta) + y \sin(\theta)) d\theta
\]

where \( w = \mathcal{F}^{-1}(|\omega|) \).

Radon-CDT transform:

- Forward and inverse:

\[
\left\{
\begin{array}{l}
\hat{\mathcal{RI}}(., \theta) = (f(., \theta) - \text{id}) \sqrt{\mathcal{RI}_0(., \theta)} \\
I = \mathcal{R}^{-1}(\text{det}(Dg)(\mathcal{RI}_0 \circ g))
\end{array}
\right.
\]

where \( g(t, \theta) = [f^{-1}(t, \theta), \theta]^T \), and \( Dg \) is the Jacobian of \( g \).
Example:

\[ \hat{I}(t, \theta) = (f(t, \theta) - t) \sqrt{\hat{I}_0(t, \theta)} \]

Matlab code:

```matlab
1 I 0 = imread ('source_face.jpg')
2 I 1 = imread ('target_face.jpg')
3 [f, df, u, du, RCDT] = RadonCDT(I0, I1); % Forward transform `u' identify the CDT of radon projections (Radon–CDT)
4 I1_hat = iRadonCDT(u, I0); % Inverse Radon CDT
```
Properties
Translation

Let $I$ be a 1D density and $\hat{I}$ its CDT.

Let $I_\tau(x) = I(x - \tau)$:

- The transform is then given by $\hat{I}_\tau(u) = \hat{I}(u) + \tau \sqrt{I_0(u)}$
- For any permissible reference $I_0$.

Scaling

Let $I$ be a 1D density and $\hat{I}$ its CDT.

Let $I_s(x) = sI(sx)$:

- The transform is then given by $\hat{I}(u) = \frac{\hat{I}(u) - x(s-1)\sqrt{I_0(x)}}{s}$
- For any permissible reference $I_0$.

Composition

Let $I_g$ represent a composition of density $I$ with an invertible function $g$, 
$I_g(x) = g'(x)I(g(x))$. The CDT of $I_g$ with respect to the reference density $I_0$ is given by

$$
\hat{I}_g(x) = \left( g^{-1} \left( \frac{\hat{I}(x)}{\sqrt{I_0(x)}} + x \right) - x \right) \sqrt{I_0(x)}
$$

- $g'(x)I(g(x))$ displaces/transports intensities: ‘Lagrangian’ perturbation
- $\left( g^{-1} \left( \frac{\hat{I}(x)}{\sqrt{I_0(x)}} + x \right) - x \right) \sqrt{I_0(x)}$ is an ‘Eulerian’ perturbation

Main point:

- CDT, R-CDT convert ‘Lagrangian’ perturbations to ‘Eulerian’ ones

Example: classifying signals under translation confound

Park et al, ArXiv preprint, 2015
Linearization Properties
Linear separation theorem

Consider signal sets:

\[
\begin{align*}
\text{Let } P & = \{ p_h : h \in \mathbb{C} \} \text{ and } Q = \{ q_h : h \in \mathbb{C} \}, \text{ where} \\
p_h(x) &= h'(x)p_0(h(x)) \text{ and } q_h(x) = h'(x)q_0(h(x))
\end{align*}
\]

Consider class of diffeomorphisms \( \mathbb{C} \) s.t.:

\[
\begin{align*}
i) \quad h & \in \mathbb{C} \iff h^{-1} \in \mathbb{C} \\
ii) \quad h_1, h_2 & \in \mathbb{C} \Rightarrow \alpha h_1 + (1 - \alpha) h_2 \in \mathbb{C}, \quad \forall \alpha \in [0,1] \\
iii) \quad h_1, h_2 & \in \mathbb{C} \Rightarrow h_1(h_2), h_2(h_1) \in \mathbb{C} \\
iv) \quad h'p_0 \circ h & \neq q_0 \quad \forall h_{\theta} \in \mathbb{C}
\end{align*}
\]

CDT linear separation theorem

If i through iv hold, signal classes will be linearly separable in CDT (R-CDT) domain

CDT (R-CDT) linear separation:

Park et al, ArXiv preprint, 2015
Kolouri et al, IEEE TIP, 2016
**Introduction**

**Low level transforms**

**Properties**

**Basic properties**

**Linearization**

**Kernel view**

---

**Cumulative Distribution Transform**

**Signal space**

\( P = \{ p | p = h' (p_0 \circ h), \forall h \in C \} \)

\( Q = \{ q | q = h' (q_0 \circ h), \forall h \in C \} \)

\( C = \{ h | h(x) = x + \tau, \tau \in \mathbb{R} \} \)

**Transform space**

\[ p(x) = p_0(x + \tau) \xrightarrow{\text{CDT}} \tilde{p}(x) = \tilde{p}_0(x) + \tau \sqrt{I_0(x)} \]

\[ q(x) = q_0(x + \tau) \xrightarrow{\text{CDT}} \tilde{q}(x) = \tilde{q}_0(x) + \tau \sqrt{I_0(x)} \]

---

**Projection of the data onto a 3-dimensional discriminant subspace**

**Projection of the transformed data onto a 3-dimensional discriminant subspace**
To run this example in Matlab:

In the code accompanying this tutorial

Test_CDT.m
Interpolation in R-CDT domain:

\[ \alpha \hat{I} + (1 - \alpha) \hat{I}_0, \ \alpha \in \{0, 0.25, 0.5, 0.75, 1\} \]
Linear separation (empirical test):

Kolouri et al, IEEE TIP, 2016
**Linear separation (empirical test):**

<table>
<thead>
<tr>
<th>Nuclei data</th>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Linear SVM</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>Training accuracy</td>
<td>Testing Accuracy</td>
<td></td>
</tr>
<tr>
<td>Image space</td>
<td>100</td>
<td>65.2 ± 6.6</td>
<td></td>
</tr>
<tr>
<td>Radon space</td>
<td>100</td>
<td>62.56 ± 6.7</td>
<td></td>
</tr>
<tr>
<td>Ridgelet space</td>
<td>100</td>
<td>62.92 ± 5.6</td>
<td></td>
</tr>
<tr>
<td>Radon-CDT space</td>
<td>100</td>
<td>75.56 ± 6.21</td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Classification comparison</th>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>PCANet</td>
<td>Radon-CDT</td>
<td></td>
</tr>
<tr>
<td>Face dataset</td>
<td>84.12 ± 11.7</td>
<td>82.62 ± 11.5</td>
<td></td>
</tr>
<tr>
<td>Nuclei dataset</td>
<td>74.16 ± 4.36</td>
<td>75.56 ± 6.21</td>
<td></td>
</tr>
<tr>
<td>Animal face dataset</td>
<td>80.81 ± 5.1</td>
<td>79.42 ± 6.12</td>
<td></td>
</tr>
</tbody>
</table>

Kolouri et al, IEEE TIP, 2016
Kernel point of view
Kernel methods in machine learning:

Positive definite kernels:

\[ \sum_{i=1}^{n} \sum_{j=1}^{n} c_i c_j k(I_i, I_j) \geq 0 \]

→ Mercer theorem/trick

Wasserstein distances as kernels:

- CDT can be safely used as a kernel (Gaussian RBF, polynomial)
- R-CDT can be safely used as a kernel (Gaussian RBF, polynomial)
Recall that (1D):

\[ f'_1(t)I_1(f_1(t)) = f'_2(t)I_2(f_2(t)) = I_0(t) \quad \text{and let} \quad h'(t)I_2(h(t)) = I_1(t) \]

Then \( I_1(f_1(t)) = h'(f_1(t))I_2(h(f_1(t))) \rightarrow f_2(t) = h(f_1(t)) \)

**In 1D CDT isometrically ‘embeds’ \( W_2 \):**

\[
W_2^2(I_2, I_1) = \int (h(t) - t)^2 I_0(t) dt \\
= \int (h(f_1(u)) - f_1(u))^2 \frac{df_1(u)}{du} I_1(f_1(u)) du \\
= \int (f_2(u) - f_1(u))^2 I_0(u) du = \|\hat{I}_2 - \hat{I}_1\|^2_{I_0}
\]
Let $M$ be the set of absolutely continuous positive probability density functions and let $\sigma$ be a template probability measure with corresponding probability density function $I_0 \in M$. Let $\phi_\sigma : M \to \mathcal{V}$ be the CDT. Define a kernel function $k : M \times M \to \mathbb{R}$ to be $k(I_i, I_j) := \langle \phi_\sigma(I_i), \phi_\sigma(I_j) \rangle^d$ for $d \in \{1, 2, \ldots\}$ then $k(\cdot, \cdot)$ is a positive definite kernel.

The same is true for when $\phi$ is the R-CDT (Sliced-Wasserstein kernels):

Kolouri, Zou, Rohde, CVPR 2016
Gaussian Radial Basis Function (RBF) kernel:

Theorem (Jayasumana et al):

Let \((M, d)\) be a metric space and define \(k : M \times M \to \mathbb{R}\) by \(k(I_i, I_j) := \exp(-\gamma d^2(I_i, I_j))\) for all \(I_i, I_j \in M\). Then \(k(., .)\) is a positive definite kernel for all \(\gamma > 0\) if and only if there exists an inner product space \(\mathcal{V}\) and a function \(\psi : M \to \mathcal{V}\) such that \(d(I_i, I_j) = \|\psi(I_i) - \psi(I_j)\|_{\mathcal{V}}\).

Corollary:

- \(k(I_i, I_j) := \exp(-\gamma W_2^2(I_i, I_j)) = \exp(-\gamma \|\hat{I}_i - \hat{I}_j\|^2)\) is a PD kernel.
- Similar result holds for when \(\phi\) is the R-CDT.

Kolouri, Zou, Rohde, CVPR 2016
Clustering with Kernels:

$k$-means clustering evaluation

Within cluster sum of squares

V-measure

Kolouri, Zou, Rohde, CVPR 2016
Classification with Kernels:

SVM classification accuracy

Kolouri, Zou, Rohde, CVPR 2016
Non-linear signal models based on transport

- \( s(t) \sim f'(t)s(f(t)) \) (suitable for signal ‘morphology’ problems)
- \( f(t) \) a transport map/plan
- ‘Carries’ intensity around \( \rightarrow \) Lagrangian point of view

Transforms

- Linear optimal transport (LOT): 2D, 3D, particle, continuous
- Cumulative distribution transform (CDT): 1D
- Radon-cumulative distribution transform (R-CDT): 2D

Properties

- ‘Linearization’
- Kernel point of view