A Sparse Algorithm for Optimal Transport

Bernhard Schmitzer

November 10, 2015
Applications of Numerical Optimal Transport

Versatile Tool

- computer vision & machine learning: histogram comparison [Rubner et al., 2000; Pele and Werman, 2009], ground metric learning [Wang and Guibas, 2012; Cuturi and Avis, 2014]
- imaging: interpolation [Maas et al., 2014], shape matching [Schmitzer and Schnörr, 2015], deformation analysis [Wang et al., 2012], denoising [Lellmann et al., 2014]
- optics [de Castro et al., 2014; Feng et al., 2015; Brix et al., 2015], physics [Frisch et al., 2002; Brenier, 2011], bakery logistics...

X computationally demanding
Solvers & Related Work

Discrete Solvers

- Hungarian method [Kuhn, 1955], Auction algorithm [Bertsekas, 1979], network simplex [Ahuja et al., 1993]

✓ numerically ‘simple & robust’, work on any cost-function

✗ scale poorly on large, dense problems
Solvers & Related Work

Discrete Solvers

- Hungarian method [Kuhn, 1955], Auction algorithm [Bertsekas, 1979], network simplex [Ahuja et al., 1993]
  ✔ numerically ‘simple & robust’, work on any cost-function
  ✗ scale poorly on large, dense problems

Continuous Solvers

- [Brenier, 1991; Haker et al., 2004; Carlier et al., 2010; Benamou et al., 2014], dynamic formulation [Benamou and Brenier, 2000]
  ✔ elegant theory, need only handle transport map
  ✗ restricted to particular ground costs, numerically more challenging
Solvers & Related Work

Discrete Solvers

- Hungarian method [Kuhn, 1955], Auction algorithm [Bertsekas, 1979], network simplex [Ahuja et al., 1993]
- ✔ numerically ‘simple & robust’, work on any cost-function
- ✗ scale poorly on large, dense problems

Continuous Solvers

- [Brenier, 1991; Haker et al., 2004; Carlier et al., 2010; Benamou et al., 2014], dynamic formulation [Benamou and Brenier, 2000]
- ✔ elegant theory, need only handle transport map
- ✗ restricted to particular ground costs, numerically more challenging

Approximations and Tricks

- approximations: cost thresholding [Pele and Werman, 2009], tangent space [Wang et al., 2012], entropy smoothing [Cuturi, 2013]
- multi-scale [Mérigot, 2011; Schmitzer and Schnörr, 2013], sparse iterations [Mérigot and Oudet, 2014; Schmitzer, 2015; Oberman and Ruan, 2015]
Transport Plans / Couplings

\[ \Pi(\mu, \nu) = \{ \pi \in \operatorname{Prob}(X \times Y) : \operatorname{Proj}_X \# \pi = \mu, \operatorname{Proj}_Y \# \pi = \nu \} \]
(Discrete) Optimal Transport in a Nutshell

Transport Plans / Couplings

\[ \Pi(\mu, \nu) = \{ \pi \in \text{Prob}(X \times Y) : \text{Proj}_X \# \pi = \mu, \text{Proj}_Y \# \pi = \nu \} \]

Primal Problem:

\[ \inf_{\pi \in \Pi(\mu, \nu)} \sum_{(x, y) \in X \times Y} c(x, y) \pi(x, y) \]
(Discrete) Optimal Transport in a Nutshell

Transport Plans / Couplings

$$\Pi(\mu, \nu) = \{ \pi \in \text{Prob}(X \times Y): \text{Proj}_X^\sharp \pi = \mu, \text{Proj}_Y^\sharp \pi = \nu \}$$

Primal Problem: $$\inf_{\pi \in \Pi(\mu, \nu)} \sum_{(x, y) \in X \times Y} c(x, y) \pi(x, y)$$

Dual Problem: $$\sup_{(\alpha, \beta) \in \mathbb{R}^X \times Y} \left[ \sum_{x \in X} \alpha(x) \mu(x) + \sum_{y \in Y} \beta(y) \nu(y) \right]$$

subject to $$\alpha(x) + \beta(y) \leq c(x, y)$$ for all $$(x, y) \in X \times Y$$. 

PD Optimality Condition:

$$\pi(x, y) > 0 \Rightarrow \alpha(x) + \beta(y) = c(x, y)$$

Restricted Problem:

$$N \subset X \times Y$$
Transport Plans / Couplings

\[ \Pi(\mu, \nu) = \{ \pi \in \text{Prob}(X \times Y) : \text{Proj}_X \# \pi = \mu, \text{Proj}_Y \# \pi = \nu \} \]

Primal Problem: \[
\inf_{\pi \in \Pi(\mu, \nu)} \sum_{(x, y) \in X \times Y} c(x, y) \pi(x, y)
\]

Dual Problem: \[
\sup_{(\alpha, \beta) \in \mathbb{R}^X \times Y} \left[ \sum_{x \in X} \alpha(x) \mu(x) + \sum_{y \in Y} \beta(y) \nu(y) \right]
\]

subject to \[ \alpha(x) + \beta(y) \leq c(x, y) \text{ for all } (x, y) \in X \times Y. \]
(Discrete) Optimal Transport in a Nutshell

Transport Plans / Couplings

\[ \Pi(\mu, \nu) = \{ \pi \in \text{Prob}(X \times Y): \text{Proj}_X \# \pi = \mu, \text{Proj}_Y \# \pi = \nu \} \]

Primal Problem:

\[
\inf_{\pi \in \Pi(\mu, \nu)} \sum_{(x, y) \in X \times Y} c(x, y) \pi(x, y)
\]

Dual Problem:

\[
\sup_{(\alpha, \beta) \in \mathbb{R}^X \times Y} \left[ \sum_{x \in X} \alpha(x) \mu(x) + \sum_{y \in Y} \beta(y) \nu(y) \right]
\]

subject to \( \alpha(x) + \beta(y) \leq c(x, y) \) for all \((x, y) \in X \times Y \).

PD Optimality Condition: \( \pi(x, y) > 0 \Rightarrow \alpha(x) + \beta(y) = c(x, y) \)
(Discrete) Optimal Transport in a Nutshell

Transport Plans / Couplings

\[ \Pi(\mu, \nu) = \{ \pi \in \text{Prob}(X \times Y): \text{Proj}_X \sharp \pi = \mu, \text{Proj}_Y \sharp \pi = \nu \} \]

Primal Problem:

\[ \inf_{\pi \in \Pi(\mu, \nu)} \sum_{(x, y) \in X \times Y} c(x, y) \pi(x, y) \]

Dual Problem:

\[ \sup_{(\alpha, \beta) \in \mathbb{R}^X \times Y} \left[ \sum_{x \in X} \alpha(x) \mu(x) + \sum_{y \in Y} \beta(y) \nu(y) \right] \]

subject to \( \alpha(x) + \beta(y) \leq c(x, y) \) for all \((x, y) \in X \times Y\).

PD Optimality Condition: \( \pi(x, y) > 0 \Rightarrow \alpha(x) + \beta(y) = c(x, y) \)

Restricted Problem: \( \mathcal{N} \subset X \times Y \)
Multi-Scale Scheme

Intuition & Experience

- only small subset $\mathcal{N} \subset X \times Y$ relevant

How to select $\mathcal{N}$?
Multi-Scale Scheme

Intuition & Experience

- only small subset $\mathcal{N} \subset X \times Y$ relevant

How to select $\mathcal{N}$?

✔ multi-scale scheme
Multi-Scale Scheme

Intuition & Experience

- only small subset $\mathcal{N} \subset X \times Y$ relevant

How to select $\mathcal{N}$?

✅ multi-scale scheme
Multi-Scale Scheme

Intuition & Experience

- only small subset $\mathcal{N} \subset X \times Y$ relevant

How to select $\mathcal{N}$?

✓ multi-scale scheme

How to guarantee global optimality?

- a priori estimates? $\times$ very difficult
- a posteriori: quick verification, ‘smart’ updates of $\mathcal{N}$
Continuous Setting

- $X = Y = \mathbb{R}^n$, $c(x, y) = \|x - y\|^2$
- Optimal coupling induced by map $T : \mathbb{R}^n \to \mathbb{R}^n$: $\pi = (\text{id}, T)_\#\mu$
- $T = \nabla \varphi$ for a convex potential $\varphi : \mathbb{R}^n \to \mathbb{R}$
Polar Factorization & Local Optimality

Continuous Setting

- \( X = Y = \mathbb{R}^n, \ c(x, y) = \|x - y\|^2 \)
- Optimal coupling induced by map \( T : \mathbb{R}^n \rightarrow \mathbb{R}^n: \pi = (\text{id}, T)_\# \mu \)
- \( T = \nabla \varphi \) for a convex potential \( \varphi : \mathbb{R}^n \rightarrow \mathbb{R} \)

Local Optimality \( \Rightarrow \) Global Optimality
Polar Factorization & Local Optimality

Continuous Setting

- $X = Y = \mathbb{R}^n$, $c(x, y) = \|x - y\|^2$
- Optimal coupling induced by map $T : \mathbb{R}^n \to \mathbb{R}^n$: $\pi = (\text{id}, T)_\# \mu$
- $T = \nabla \varphi$ for a convex potential $\varphi : \mathbb{R}^n \to \mathbb{R}$

Local Optimality $\Rightarrow$ Global Optimality

Discrete Equivalents:

- 1D: ‘trivial’: Monge property, 2D: not so much...
Short-Cuts

\[\alpha(x_1) + \beta(y_1) = c(x_1, y_1),\]
\[\alpha(x_1) + \beta(y_2) \leq c(x_1, y_2),\]
\[\beta(y_2) \leq \beta(y_1) + [c(x_1, y_2) - c(x_1, y_1)].\]
Short-Cuts

\[
\alpha(x_1) + \beta(y_1) = c(x_1, y_1),
\beta(y_2) \leq \beta(y_1) + \left[ c(x_1, y_2) - c(x_1, y_1) \right],
\alpha(x_1) + \beta(y_n) \leq c(x_1, y_2) + n - 1 \sum_{i=2}^{n} \left[ c(x_i, y_i+1) - c(x_i, y_i) \right].
\]
Short-Cuts

\[
\begin{align*}
\alpha(x_1) + \beta(y_1) &= c(x_1, y_1), \\
\beta(y_2) &\leq \beta(y_1) + [c(x_1, y_2) - c(x_1, y_1)], \\
\alpha(x_1) + \beta(y_n) &\leq c(x_1, y_2) + n - 1 \sum_{i=2} y_i \\
\end{align*}
\]

\[c(x, y) = \|x - y\|_2: \text{points along straight line are short-cuts}\]
\[ \alpha(x_1) + \beta(y_1) = c(x_1, y_1) \]
\[ \beta(y_2) \leq \beta(y_1) + [c(x_1, y_2) - c(x_1, y_1)] \]
\[ \alpha(x_1) + \beta(y_n) \leq c(x_1, y_2) + n - 1 \sum_{i=2}^{n} [c(x_i, y_i+1) - c(x_i, y_i)] \]
\[ c(x, y) = \|x - y\|_2 \]

Points along straight line are short-cuts.
Short-Cuts

\[
\alpha(x_1) + \beta(y_1) = c(x_1, y_1),
\]

\[
\beta(y_2) \leq \beta(y_1) + [c(x_1, y_2) - c(x_1, y_1)].
\]

\[
\alpha(x_1) + \beta(y_n) \leq c(x_1, y_2) + n - 1 \sum_{i=2}^{n} [c(x_i, y_i+1) - c(x_i, y_i)].
\]

\[
c(x, y) = \|x - y\|_2: \text{points along straight line are short-cuts}
\]
Short-Cuts

\[ \alpha(x_1) + \beta(y_1) = c(x_1, y_1) \]
Short-Cuts

\[ \alpha(x_1) + \beta(y_1) = c(x_1, y_1), \quad \alpha(x_1) + \beta(y_2) \leq c(x_1, y_2) \]
Short-Cuts

\[ \alpha(x_1) + \beta(y_1) = c(x_1, y_1), \quad \alpha(x_1) + \beta(y_2) \leq c(x_1, y_2) \]

\[ \beta(y_2) \leq \beta(y_1) + [c(x_1, y_2) - c(x_1, y_1)] \]
Short-Cuts

- $\alpha(x_1) + \beta(y_1) = c(x_1, y_1)$, $\alpha(x_1) + \beta(y_2) \leq c(x_1, y_2)$
- $\beta(y_2) \leq \beta(y_1) + [c(x_1, y_2) - c(x_1, y_1)]$
- $\alpha(x_1) + \beta(y_n) \leq c(x_1, y_2) + \sum_{i=2}^{n-1} [c(x_i, y_{i+1}) - c(x_i, y_i)]$
Short-Cuts

\[ \alpha(x_1) + \beta(y_1) = c(x_1, y_1), \quad \alpha(x_1) + \beta(y_2) \leq c(x_1, y_2) \]

\[ \beta(y_2) \leq \beta(y_1) + [c(x_1, y_2) - c(x_1, y_1)] \]

\[ \alpha(x_1) + \beta(y_n) \leq c(x_1, y_2) + \sum_{i=2}^{n-1} [c(x_i, y_{i+1}) - c(x_i, y_i)] \leq c(x_1, y_n) \]
Short-Cuts

\begin{align*}
\alpha(x_1) + \beta(y_1) &= c(x_1, y_1), \quad \alpha(x_1) + \beta(y_2) \leq c(x_1, y_2) \\
\beta(y_2) &\leq \beta(y_1) + [c(x_1, y_2) - c(x_1, y_1)] \\
\alpha(x_1) + \beta(y_n) &\leq c(x_1, y_2) + \sum_{i=2}^{n-1} [c(x_i, y_{i+1}) - c(x_i, y_i)] \leq c(x_1, y_n)?
\end{align*}

- continuum, \( c(x, y) = \|x - y\|^2 \): points along straight line are short-cuts
Shielding Condition

Local optimality + shielding neighbourhood $\Rightarrow$ global optimality

$\sum_{x_1, y_n} \phi \geq \sum_{x_2, y_n} \phi$
**Shielding Condition**

![Diagram](image)

\[ c(x_1, y_n) + c(x_2, y_2) \geq c(x_1, y_2) + c(x_2, y_n) \]
Shielding Condition

\[ c(x_1, y_n) + c(x_2, y_2) \geq c(x_1, y_2) + c(x_2, y_n) \]

1. \[ c(x_1, y_n) + c(x_2, y_2) \]
2. \[ c(x_1, y_2) + c(x_2, y_n) \]
Shielding Condition

\[ c(x_1, y_n) + c(x_2, y_2) \geq c(x_1, y_2) + c(x_2, y_n) \]
Shielding Condition

\[ c(x_1, y_n) + c(x_2, y_2) \geq c(x_1, y_2) + c(x_2, y_n) \]
Shielding Condition

\[ c(x_1, y_n) + c(x_2, y_2) \geq \left( c(x_1, y_2) + c(x_2, y_n) \right) \]

1 \hspace{1cm} 2

Local optimality + shielding neighbourhood \Rightarrow global optimality
Shielding Condition

\[
\begin{align*}
\text{Local optimality} + \text{shielding neighbourhood} & \implies \text{global optimality} \\

\text{1. } c(x_1, y_n) + c(x_2, y_2) & \geq c(x_1, y_2) + c(x_2, y_n)
\end{align*}
\]
Shielding Condition

1. \( c(x_1, y_n) + c(x_2, y_n) \geq c(x_1, y_2) + c(x_2, y_n) \)

2. shielding neighbourhood: always find a shielding cell

✓ existence of short-cuts follows
Shielding Condition

\[
\begin{align*}
&c(x_1, y_n) + c(x_2, y_2) &\geq & & c(x_1, y_2) + c(x_2, y_n) \\
\text{shading neighbourhood: always find a shielding cell} \\
\checkmark &\text{existence of short-cuts follows} \\
\checkmark &\text{Local optimality } + \text{ shielding neighbourhood } \Rightarrow \text{ global optimality}
\end{align*}
\]
A Sparse Algorithm

Ingredients

- sparse optimal transport solver $F : \mathcal{N} \leftrightarrow \pi$
- construction of shielding neighbourhood $G : \pi \leftrightarrow \mathcal{N}^\prime$

Iteration

$\pi_{k+1} = F(N_k)$

$N_{k+1} = G(\pi_{k+1})$

until $\pi_k$ is already locally optimal on $N_k$. 

Properties of Algorithm

Calling $F$ is fast when $N_k$ is sparse. Any solver can be used as black box.

When $\pi_1 / N_1$ are good initial guesses: need only few iterations $\Rightarrow$ multi-scale scheme

Design of $G$ must exploit geometric structure of cost-function
A Sparse Algorithm

Ingredients

- sparse optimal transport solver $F : \mathcal{N} \mapsto \pi$
- construction of shielding neighbourhood $G : \pi \mapsto \mathcal{N}$

Iteration

$$\pi_{k+1} = F(\mathcal{N}_k)$$
$$\mathcal{N}_{k+1} = G(\pi_{k+1})$$

until $\pi_k$ is already locally optimal on $\mathcal{N}_k$. 

A Sparse Algorithm

Ingredients

- sparse optimal transport solver $F : \mathcal{N} \mapsto \pi$
- construction of shielding neighbourhood $G : \pi \mapsto \mathcal{N}$

Iteration

$$\pi_{k+1} = F(\mathcal{N}_k) \quad \mathcal{N}_{k+1} = G(\pi_{k+1})$$

until $\pi_k$ is already locally optimal on $\mathcal{N}_k$.

Properties of Algorithm

- Calling $F$ is fast when $\mathcal{N}_k$ is sparse. Any solver can be used as black box.
- When $\pi_1 / \mathcal{N}_1$ are good initial guesses: need only few iterations $\Rightarrow$ multi-scale scheme
- Design of $G$ must exploit geometric structure of cost-function
$G$ for Squared Euclidean Distance

Shielding Condition for $c(x, y) = \|x - y\|^2$

$$\langle y_B - y_s, x_s - x_A \rangle > 0$$
$G$ for Squared Euclidean Distance

Shielding Condition for $c(x, y) = \|x - y\|^2$

$$\langle y_B - y_s, x_s - x_A \rangle > 0$$
Shielding Condition for $c(x, y) = \|x - y\|^2$

$$\langle y_B - y_s, x_s - x_A \rangle > 0$$
G for Squared Euclidean Distance

Shielding Condition for $c(x, y) = \|x - y\|^2$

\[ \langle y_B - y_s, x_s - x_A \rangle > 0 \]
Shielding Condition for $c(x, y) = \|x - y\|^2$

$$\langle y_B - y_s, x_s - x_A \rangle > 0$$

✓ regular grids, ✓ point-clouds with tree structure
G for Squared Euclidean Distance

所谓屏蔽条件为：

$c(x, y) = \|x - y\|^2$

\[\langle y_B - y_s, x_s - x_A \rangle > 0\]

- ✔ regular grids, ✔ point-clouds with tree structure
- ■ mass assignment regular \*, \[|\mathcal{N}| = \mathcal{O}(|X|) \ll \mathcal{O}(|X \times Y|)\]
$G$ for Squared Euclidean Distance

Shielding Condition for $c(x, y) = \|x - y\|^2 + \varepsilon(x, y)$

$$\langle y_B - y_s, x_s - x_A \rangle > 0$$

- regular grids, ✓ point-clouds with tree structure
- mass assignment regular* $\Rightarrow |\mathcal{N}| = O(|X|) \ll O(|X \times Y|)$
- can deal with noise
- more general costs . . .
### Numerical Results: Speed-up

<table>
<thead>
<tr>
<th>Problem size</th>
<th>Solving time/s</th>
<th>Speed-up</th>
</tr>
</thead>
<tbody>
<tr>
<td>0.3</td>
<td></td>
<td></td>
</tr>
<tr>
<td>1</td>
<td></td>
<td></td>
</tr>
<tr>
<td>3</td>
<td></td>
<td></td>
</tr>
<tr>
<td>10</td>
<td></td>
<td></td>
</tr>
<tr>
<td>30</td>
<td></td>
<td></td>
</tr>
<tr>
<td>100</td>
<td></td>
<td></td>
</tr>
<tr>
<td>300</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

#### Sub-solver:
- CPLEX
- LEMON-NS
- LEMON-CS

#### Method:
- dense
- sparse
Numerical Results: Sparsity

![Sparsity graph](image)

- **Sparsity**
  - \( \frac{\sum_k |N_k|}{|X|^2} \)
  - \( \frac{50}{|X|} \)
  - \( \frac{10}{|X|} \)

- **Problem size**
  - 25
  - 36
  - 49
  - 64
  - 81
  - 100

**95% quantile of iteration numbers:** 8
Numerical Results: Sparsity

<table>
<thead>
<tr>
<th>Problem size</th>
<th>X/100</th>
<th>Sparsity</th>
</tr>
</thead>
<tbody>
<tr>
<td>0.001</td>
<td></td>
<td>0.001</td>
</tr>
<tr>
<td>0.002</td>
<td></td>
<td>0.002</td>
</tr>
<tr>
<td>0.005</td>
<td></td>
<td>0.005</td>
</tr>
<tr>
<td>0.01</td>
<td></td>
<td>0.01</td>
</tr>
<tr>
<td>0.02</td>
<td></td>
<td>0.02</td>
</tr>
</tbody>
</table>

\[
\text{Sparsity} = \frac{\sum_k |N_k|}{|X|} \quad \text{and} \quad \frac{50}{|X|}
\]

- 95% quantile of iteration numbers: 8
Numerical Results: Sparsity II

\[ \rho_0 \]

\[ \rho_1 \]

\[ \text{time} \]

\[ \frac{|N_x|}{|Y|} \cdot 10^2 \]

Orientation \( \phi \)

\[ \begin{align*}
\text{Orientation:} & \quad \phi \\
& \quad \pi/2 \\
& \quad \pi \\
& \quad 3\pi/2 \\
& \quad 2\pi
\end{align*} \]
Numerical Results: Sparsity II

\[ \rho_0 \quad \rho_1 \]

\[ \frac{|N_x|}{|Y|} \cdot 10^2 \]

\[ \geq 1 \]

\[ t_{rel}(x) \]

\[ \geq 20 \]

\[ \text{Orientation } \varphi \]

\[ N_x = \{ y \in Y : (x, y) \in \mathcal{N} \} \]

\[ t_{rel} : \text{Barycentric projection of relative transport map} \]
Numerical Results: Noisy Costs

- noise: random ($\eta$) + Lipschitz component ($\lambda$)
- slower with increasing noise (expected), ✓ no immediate breakdown
Shielding Neighbourhoods for More General Costs

Preliminary Results

X more complicated,
✓ point-clouds with tree structure (⇒ multi-scale scheme)
■ intuition: ✓ strictly convex costs, ✓ squared geodesic distance on sphere
■ numerics: • speed, ✓ sparsity
Summary & Outlook

Summary

- verify global optimality locally ⇔ analogy to continuum
- basis for efficient sparsification of dense problems ⇒ combinatorial algorithms become applicable
- speed-up and saves memory

Open Questions

- closer look at other cost functions
- computational complexity
- code!

ArXiv: B. Schmitzer ‘A Sparse Multi-Scale Algorithm for Dense Optimal Transport’ 10/2015


