Globally Solving the Gromov-Wasserstein Problem for Point Clouds in Low Dimensional Euclidean Spaces

Lovera Lokeswara
SID. 20798275

Project Background

AUTHORS	MAIN PROBLEM	POSSIBLE APPLICATION	MAIN ALGORITHM

AUTHORS

Martin Ryner

Jan Kronqvist

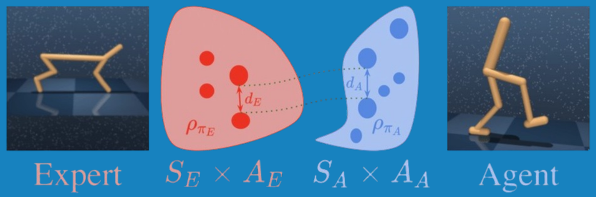
Johan Karlsson

MAIN PROBLEM

Gromov - Wasserstein

Optimal transport problem that finds the assignment between 2 sets while preserving the pairwise distances as much as possible

POSSIBLE APPLICATION



GWIL

Gromov-Wasserstein Imitation Learning

SCOT

Single Cell Alignment with Optimal Transport

MAIN ALGORITHM

<u>Gromov-Wasserstein</u>
Intractable with exponentially increasing computational burden

<u>Quadratic Assignment Problem (QAP)</u>
NP-Hard

<u>Low-Rank QAP</u>
Cutting Plane Algorithm

Gromov-Wasserstein → QAP

Let $x_1, \dots, x_n \in X$ and $y_1, \dots, y_n \in Y$ be 2 sets of points. The problem is to find an assignment π that minimizes this:

$$\min_{\Gamma \in P} \frac{1}{2} \sum_{i,i',j,j'=1}^{n} (d_X(x_i, x_{i'}) - d_Y(y_j, y_{j'}))^2 \Gamma_{i,j} \Gamma_{i',j'}$$

$$= \sum_{i,i',j,j'=1}^{n} (d_X(x_i, x_{i'})^2 - 2 d_X(x_i, x_{i'}) d_Y(y_j, y_{j'}) + d_Y(y_j, y_{j'})^2) \Gamma_{i,j} \Gamma_{i',j'}$$

$$= \langle C_x, C_x \rangle - 2 \langle C_x \Gamma, \Gamma C_y \rangle + \langle C_y, C_y \rangle$$

$$\boldsymbol{\min_{\Gamma \in P} -\langle C_x \Gamma, \Gamma C_y \rangle + \frac{1}{2} (\langle C_x, C_x \rangle + \langle C_y, C_y \rangle)}$$

Where Cx and Cy are the Euclidean distance matrix, so they are positive definite and low-rank matrices

This problem is a QAP of rank **(l_x+2)(l_y+2)**

Reducing the Rank $(lx+2)(ly+2) \rightarrow lxly$

Substitute $\langle C_x\Gamma, \Gamma C_y\rangle$ to the previous equation (**Proposition 1.**)

$$\langle C_x\Gamma, \Gamma C_y\rangle = \langle 2X\Gamma Y^T, 2X\Gamma Y^T\rangle + \langle L, \Gamma\rangle + 2\mathbf{1}^T m_y \mathbf{1}^T m_x,$$

Where $L = 2nm_x m_y{}^T - 4m_x 1^T Y^T Y - 4X^T X 1 m_y{}^T$

Low rank formulation of Gromov-Wasserstein

$$\min_{\Gamma\in P} \quad -\langle 2X\Gamma Y^T, 2X\Gamma Y^T\rangle - \langle L, \Gamma\rangle + c_0$$

Where:

- $L = 2nm_x m_y{}^T - 4m_x 1^T Y^T Y - 4X^T X 1 m_y{}^T$
- $c_0 = (\langle C_x, C_x\rangle + \langle C_y, C_y\rangle - 4\mathbf{1}^T m_y \mathbf{1}^T m_x)/2.$

Relax the Problem

$$\min_{\Gamma \in P} \quad -\langle 2X\Gamma Y^T, 2X\Gamma Y^T \rangle - \langle L, \Gamma \rangle + c_0$$

$$\min_{\Gamma \in \overline{P}} \quad -\langle 2X\Gamma Y^T, 2X\Gamma Y^T \rangle - \langle L, \Gamma \rangle + c_0,$$

Cutting Plane Algorithm: Solve a sequence of relaxed problems that are iteratively strengthened by generating and accumulating valid linear inequality constraints

Proposition 2. *Any optimal solution of the discrete Gromov-Wasserstein problem (1st), is also an optimal solution to the Gromov-Wasserstein problem (2nd). Conversely, problem (2nd) always has an optimal solution in one extreme point, and any optimal extreme point to (2nd) is also an optimal solution to (1st).*

Note: P is the set of all permutation matrices and $\overline{P}$ is the set of all doubly stochastic matrices

Cutting Plane Algorithm

Algorithm Set Up

Objective $(w, W)\text{-space as } \mathcal{F} = \text{Proj}_{W,w}\left(W \in \mathbb{R}^{\ell_x \times \ell_y}, w \in \mathbb{R}, \Gamma \in \overline{P} \;\middle|\; W = 2X\Gamma Y^T,\; w = \langle L, \Gamma\rangle\right)$

Start of Algorithm

1	Create new variables for better analysis	$$\min_{W\in\mathbb{R}^{\ell_x\times\ell_y}, w\in\mathbb{R}, \Gamma\in\overline{P}} \quad -\|W\|_F^2 - w + c_0$$ $$\text{subject to} \quad W = 2X\Gamma Y^T,\; w = \langle L, \Gamma\rangle$$
2	Project out the $\boldsymbol{\Gamma}$ variables, and we define the feasible set polytope $\boldsymbol{\mathcal{F}}$	$$\min_{W\in\mathbb{R}^{\ell_x\times\ell_y}, w\in\mathbb{R}} \quad -\|W\|_F^2 - w + c_0$$ **Subject to** $\langle Z_r, W\rangle + \alpha_r w \le \beta_r, \quad \text{for } r = 1, \dots, N$
3	A bounding box $\boldsymbol{\mathcal{F}}$ to define the set of constraints	$$\min_{\Gamma\in\bar{P}} 2(X\Gamma Y^T)_{i,j} \le W_{i,j} \le \max_{\Gamma\in\bar{P}} 2(X\Gamma Y^T)_{i,j} \quad \text{for } i = 1,\dots,\ell_x;\; j = 1,\dots,\ell_y$$ $$\min_{\Gamma\in\bar{P}}\langle L,\Gamma\rangle \le w \le \max_{\Gamma\in\bar{P}}\langle L,\Gamma\rangle,$$

Algorithm Set Up

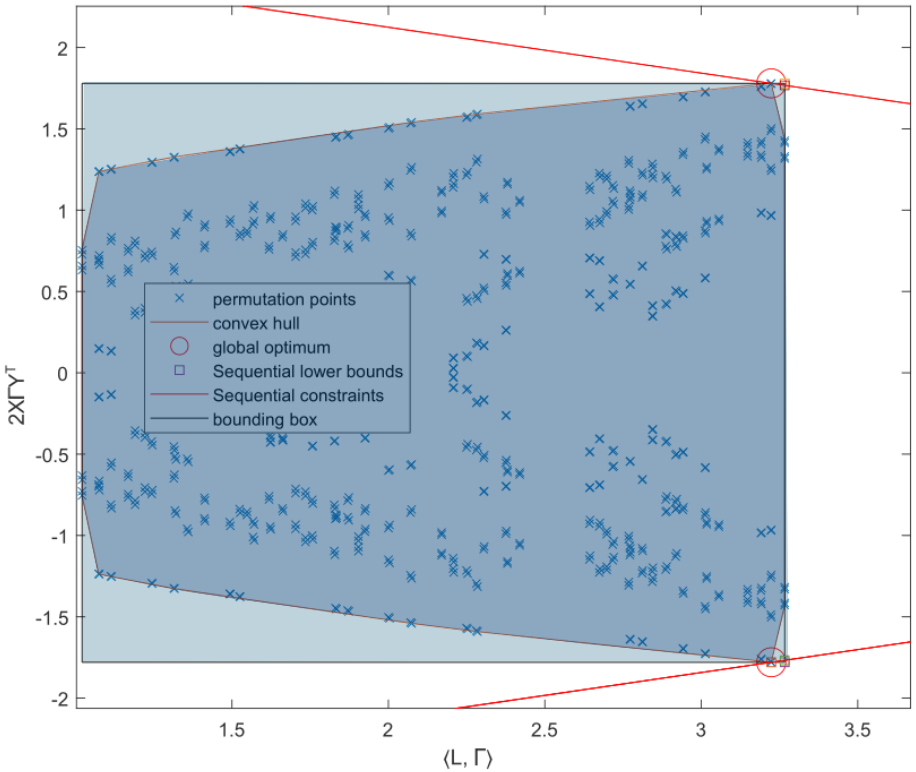
Objective

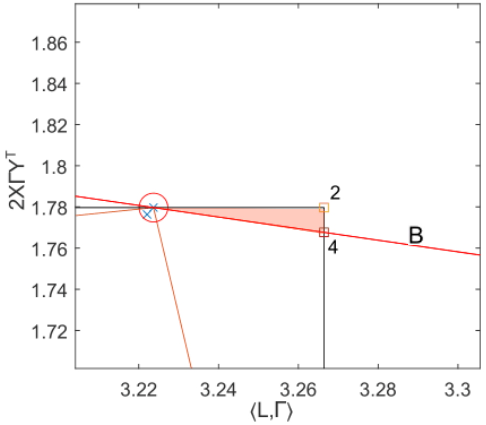
$$(w, W)\text{-space as } \mathcal{F} = \mathrm{Proj}_{W,w}\left(W \in \mathbb{R}^{\ell_x \times \ell_y}, w \in \mathbb{R}, \Gamma \in \overline{P} \;\middle|\; W = 2X\Gamma Y^T,\ w = \langle L, \Gamma\rangle\right)$$

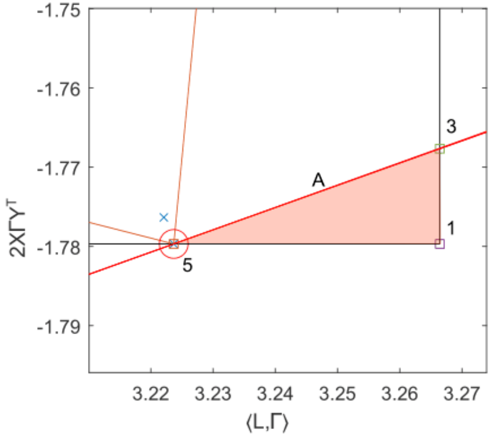
4	Construct new constraint based on the gradient of the objective function (7)	$\nabla_{(w,\mathrm{vec}(W)^T)}(-\|W\|_F^2 - w,) = \left(-1, -2\,\mathrm{vec}(W)^T\right)$
5	New constraint that bound $\mathcal{F}$	$\langle Z_r, W\rangle + \alpha_r w \leq \beta_r, \quad \text{for } r = 1, \dots, N.$ *Let*: $Z_{N+1} = 2W_N$ *and* $\alpha_{N+1} = 1$
6	Update the Upper Bound	$\beta_{N+1} := \max_{W \in \mathbb{R}^{\ell_x \times \ell_y}, w \in \mathbb{R}, \Gamma \in \overline{P}} \langle Z_{N+1}, W\rangle + \alpha_{N+1} w = \max_{\Gamma \in \overline{P}} \langle 4X^T W_N Y + L, \Gamma\rangle$ subject to $W = 2X\Gamma Y^T,\ w = \langle L, \Gamma\rangle$

End of Algorithm

Visualization







Visualization

Pseudocode

Algorithm 1 Gromov-Wasserstein problem

Input $X \in \mathbb{R}^{\ell_x \times n}, Y \in \mathbb{R}^{\ell_y \times n}, \epsilon > 0$ (Define point clouds and give tolerance level)
$L_{\text{bound}} \leftarrow -\infty$, and $U_{\text{bound}} \leftarrow \infty$ (Set lower and upper bounds)
(Z_r, α_r, β_r) for $r = 1, \ldots, N$ from (8), where $N = 2\ell_x \ell_y + 2$ (Set initial constraints)
while $U_{\text{bound}} - L_{\text{bound}} > \epsilon$ **do**
$(w_N, W_N) \leftarrow$ Optimal solution to (7) (Solve (7))
$L_{\text{bound}} \leftarrow -\|W_N\|_F^2 - w_N + c_0$ (Update lower bound)
$\Gamma_N \leftarrow$ Optimal solution to (9) (Solve (9))
$U_{\text{bound}} \leftarrow \min(U_{\text{bound}}, -\|2X\Gamma_N Y^T\|_F^2 - \langle L, \Gamma_N \rangle + c_0)$ (Update upper bound)
$(Z_{N+1}, \alpha_{N+1}, \beta_{N+1}) \leftarrow (2W_N, 1, \langle 4X^T W_N Y + L, \Gamma_N \rangle)$ (Calculate new constraints)
$N \leftarrow N + 1$ (Update iteration number)
end while

Convergence

Proving the convergence of the algorithm

Updating Lower Bound

$$\min_{W \in \mathbb{R}^{\ell_x \times \ell_y}, w \in \mathbb{R}} \quad -\|W\|_F^2 - w + c_0$$

Subject to $\langle Z_r, W \rangle + \alpha_r w \leq \beta_r, \quad \text{for } r = 1, \ldots, N$

LBound $\leftarrow -\|W_N\|_F^2 - w_N + c_0$

Updating Upper Bound

$$\beta_{N+1} := \max_{W \in \mathbb{R}^{\ell_x \times \ell_y}, w \in \mathbb{R}, \Gamma \in \overline{P}} \quad \langle Z_{N+1}, W \rangle + \alpha_{N+1} w \quad = \max_{\Gamma \in \overline{P}} \langle 4X^T W_N Y + L, \Gamma \rangle$$

$$\text{subject to} \quad W = 2X\Gamma Y^T, \; w = \langle L, \Gamma \rangle$$

UBound $\leftarrow \min(UBound, -\|2X\mathbf{\Gamma}_N Y^T\|_F^2 - \langle L, \mathbf{\Gamma}_N \rangle + c_0)$

Theorem

The gap between the upper bound and lower bound in this algorithm will converge to 0 (if the tolerance is ϵ = 0)

Numerical Experiment Result

Algorithm 1 = Proposed Algorithm

1

MILP 1

2

Branch & Bound

3

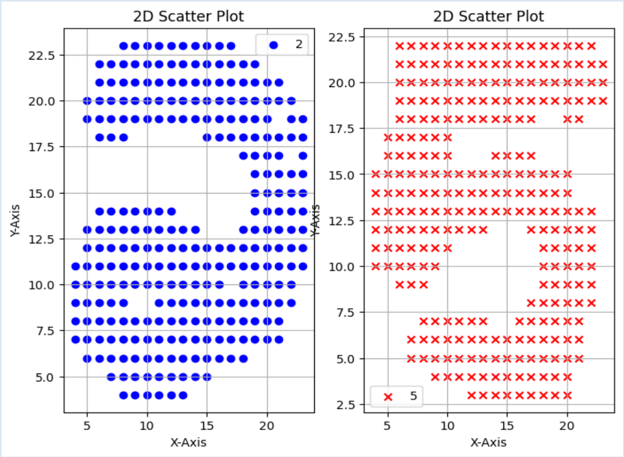
Type	n	ℓ_x, ℓ_y	Rel. error	Algorithm 1 [s] Extreme point / B&B	MILP1 [s]	(6) B&B [s]
$\mathcal{U}$	10	2,2	10^{-8}	0.14 (0.07-0.3) / 21 (6-47)	39 (11-58)	0.15 (0.14-0.16)
$\mathcal{U}$	100	2,2	10^{-8}	0.48 (0.3-0.7) / 86 (52-107)	-	25 (19-39)
$\mathcal{U}$	500	2,2	10^{-8}	11 (9-16) / 408 (269-511)	-	-
$\mathcal{U}$	1000	2,2	10^{-8}	69 (54-85) / 576 (389-1059)	-	-
$\mathcal{U}$	2000	2,2	10^{-8}	460 (313-653) / -	-	-
$\mathcal{U}$	10	2,3	10^{-8}	1.8 (1.2-2.4) / 133 (45-296)	105 (49-147)	2.4(1.8-3.4)
$\mathcal{U}$	100	2,3	10^{-8}	278 (99-813) / -	-	172 (133-221)
$\mathcal{U}$	500	2,3	10^{-8}	9568 / -	-	-
$\mathcal{N}_1$	10	2,3	10^{-8}	0.51 (0.39-0.65) / 708 (233-1184)	146 (66-227)	3 (2.6-4.0)
$\mathcal{N}_1$	100	2,3	10^{-8}	86 (20-275) / -	-	95 (73-116)
$\mathcal{N}_1$	500	2,3	10^{-5}	5310!/ -	-	-
$\mathcal{N}_2$	10	3,3	10^{-2}	1.8 (0.7-3.2) / 142 (73-210)	117 (71-163)	0.2(0.1-0.3)
$\mathcal{N}_2$	100	3,3	10^{-2}	36 (22-55)/ -	-	45(36-65)
$\mathcal{N}_2$	500	3,3	10^{-2}	436 (228-862) / -	-	-
$\mathcal{N}_3$	10	3,3	10^{-2}	1.2 (0.5-2.3) / 22 (11-43)	72 (43-94)	0.2(0.1-0.3)
$\mathcal{N}_3$	100	3,3	10^{-2}	7 (5-8)/ 91 (76-111)	-	10 (9-12)
$\mathcal{N}_3$	500	3,3	10^{-2}	11 (9-16) / 161 (104-226)	-	-
$\mathcal{N}_3$	1000	3,3	10^{-2}	25 (22-29) / 176 (149-224)	-	-
$\mathcal{N}_3$	2000	3,3	10^{-2}	93 (91-100) / 578 (429-691)	-	-

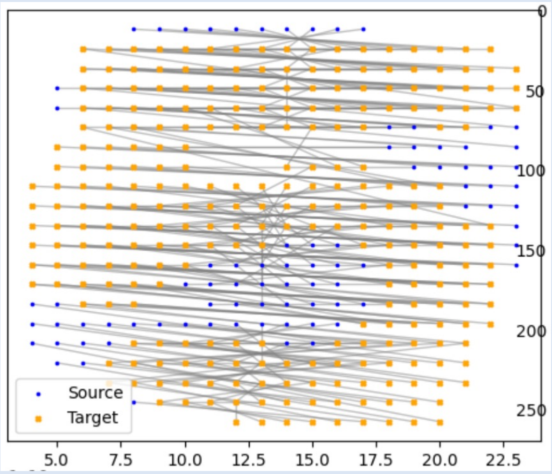
MNIST Dataset

Stage 1: Pre-process the Dataset

Stage 2: Compute the best assignment

Stage 3: Calculate GW Distance and use K-means clustering (k=2)






```
Gromov-Wasserstein distance (X, 5):   1.406569004058838
Gromov-Wasserstein distance (X, 0):   1.0764291286468506
Gromov-Wasserstein distance (X, 8):   1.080357313156128
Gromov-Wasserstein distance (X, 2_2):  0.6204614639282227
Gromov-Wasserstein distance (X, 2_3):  0.7881588339805603
Gromov-Wasserstein distance (X, 3):   1.9450793266296387
Gromov-Wasserstein distance (X, 0_2):  1.4792852401733398
Gromov-Wasserstein distance (X, 3_2):  2.4657485485076904
```

```
Cluster Labels:
Data Point 0: Cluster 1
Data Point 1: Cluster 0
Data Point 2: Cluster 0
Data Point 3: Cluster 0
Data Point 4: Cluster 1
Data Point 5: Cluster 1
Data Point 6: Cluster 0
Data Point 7: Cluster 0
Data Point 8: Cluster 0
```

```
Data Labels:
Data Point 0: Label 2
Data Point 1: Label 5
Data Point 2: Label 0
Data Point 3: Label 8
Data Point 4: Label 2_2
Data Point 5: Label 2_3
Data Point 6: Label 3
Data Point 7: Label 0_2
Data Point 8: Label 3_2
```

Dataset Source: https://www.kaggle.com/datasets/cristiangarcia/pointcloudmnist2d?select=test.csv

Algorithm Modification

Adaptive Epsilon

1. Do not need high precision in early iteration.
2. Faster convergence in early iterations.
3. Later iterations might benefit from stricter convergence criteria.
4. Reduce computational cost while maintaining solution quality.

Scaled EMD

1. Linear programming solvers work better with well-conditioned matrices.
2. Prevents numerical overflow or underflow.
3. Improves the numerical stability.
4. Maintain solution accuracy through proper rescaling and better conditioning of the optimization problem.

Algorithm Modification – Early Stopping and Scaled EMD

Original Algorithm – MNIST Data

- Use a **fixed** epsilon for convergence
- EMD solver can face numerical issues with very large or small values

```
Iter |Bound gap
----------------------
   0|1.992685e+10
   1|2.266376e+09
   2|2.156267e+09
   3|1.930290e+09
   4|1.436316e+09
   5|1.056895e+09
   6|1.045617e+09
   7|8.789524e+08
   8|7.982276e+08
   9|6.165691e+08
  10|6.148932e+08
  11|5.400076e+08
  12|4.844402e+08
  13|4.493938e+08
  14|3.754367e+08
  15|3.406628e+08
  16|3.237967e+08
  17|3.060117e+08
  18|2.887174e+08
  19|2.715465e+08
  20|2.531407e+08
  21|2.505616e+08
  22|2.284552e+08
...
 168|7.629395e-06
 169|1.525879e-05
 170|0.000000e+00
Computation time used:  1.482682228088379
```

Modified Algorithm – MNIST Data

- Use **adaptive** epsilon for convergence
- Prevents numerical overflow or underflow

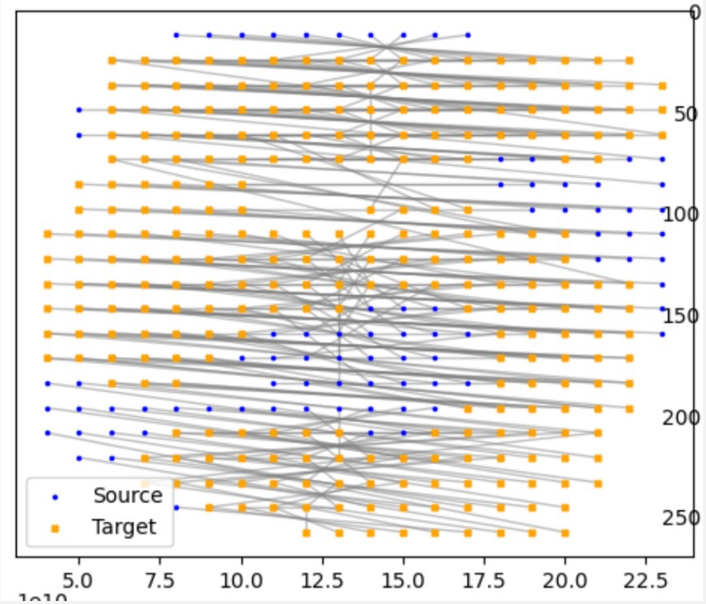
```
Iter |Bound gap
----------------------
   0|1.992685e+10
   1|2.266376e+09
   2|2.156267e+09
   3|1.930290e+09
   4|1.436316e+09
   5|1.056895e+09
   6|1.045617e+09
   7|8.789524e+08
   8|7.982276e+08
   9|6.165691e+08
  10|6.148932e+08
  11|5.400076e+08
  12|4.844402e+08
  13|4.493938e+08
  14|3.754367e+08
  15|3.406628e+08
  16|3.237967e+08
  17|3.060117e+08
  18|2.887174e+08
  19|2.715465e+08
  20|2.531407e+08
  21|2.505616e+08
  22|2.284552e+08
...
 165|6.919998e+00
 166|3.564888e+00
 167|-7.629395e-06
Computation time used:  1.4090585708618164
```

Source: https://www.kaggle.com/datasets/cristiangarcia/pointcloudmnist2d?select=test.csv

Algorithm Modification – Early Stopping and Scaled EMD

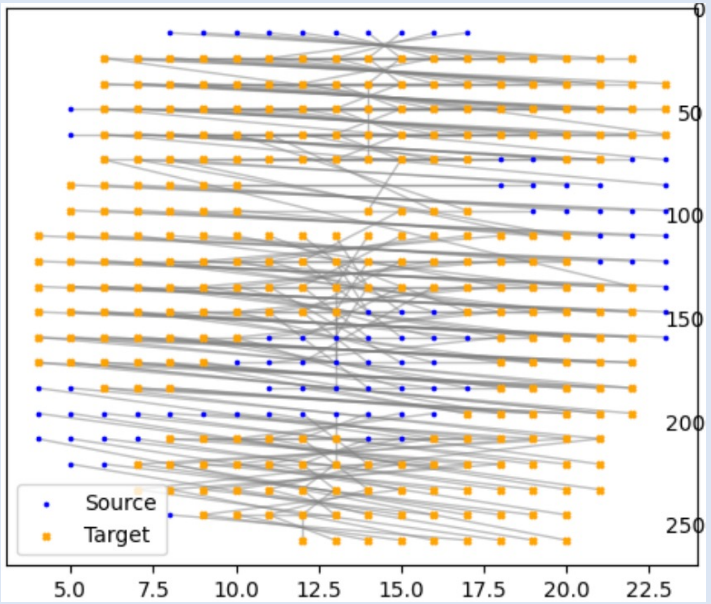
Original Algorithm – MNIST Data

- Use a **fixed** epsilon for convergence
- EMD solver can face numerical issues with very large or small values



Modified Algorithm – MNIST Data

- Use **adaptive** epsilon for convergence
- Prevents numerical overflow or underflow

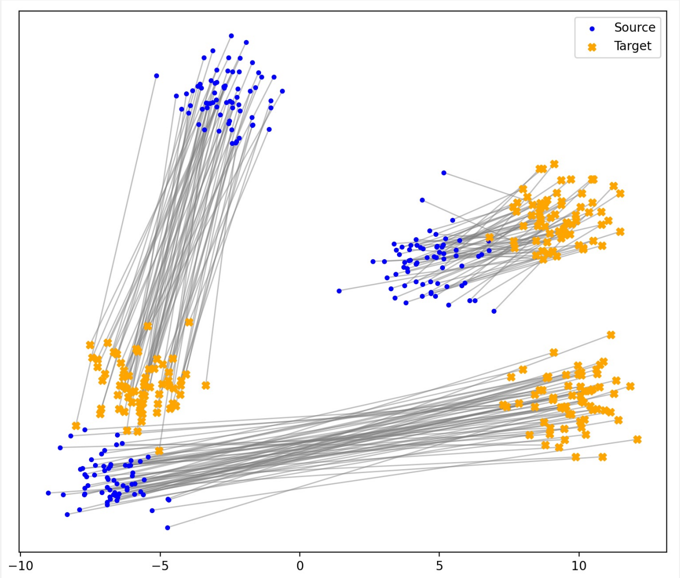


Obtain the same solution with less iteration !!!

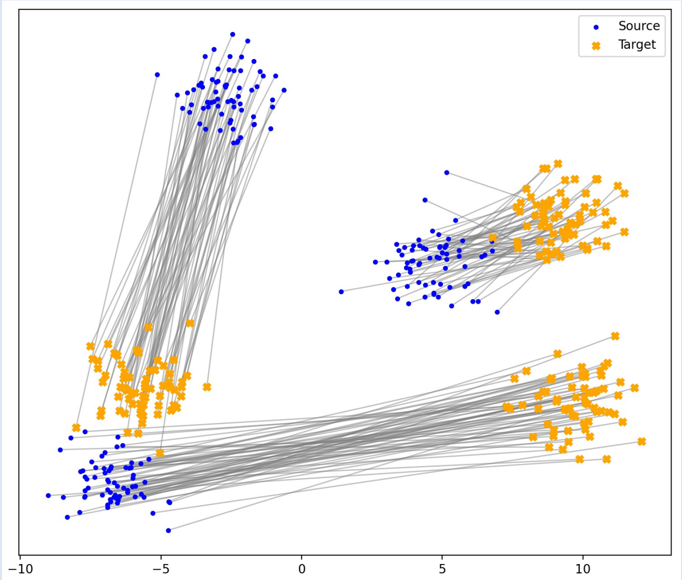
Source: https://www.kaggle.com/datasets/cristiangarcia/pointcloudmnist2d?select=test.csv

Algorithm Modification – Early Stopping and Scaled EMD

Original Algorithm – Artificial Data



Modified Algorithm – Artificial Data



Obtain the same solution with less iteration !!!

```
Computation time used in Global GW (algorithm 1):  0.836761474609375
Computation time used in Modified Global GW (modified algorithm):  0.792569637298584
```

Source: Synthetic Data

Algorithm Modification – Early Stopping and Scaled EMD

Modified Algorithm – Pseudocode

```
def scaled_emd2(a,b,M):
    scale = max(1.0, np.max(np.abs(M)))
    M_scaled = M/scale
    return scale * ot.emd2(a,b,M_scaled)

def adaptive_epsilon(iter_num, base_epsilon=1e-6):
    return base_epsilon * (1 + 0.1 * iter_num)
```

Key takeaway

- The modified algorithm gives us the same global optimal solution

- The modified algorithm needs less iteration.

Thank you!

Code can be found here:
https://github.com/LoveraLokeswara/Gromov_Wasserstein_for_Low_Dimension

Proof of Proposition 1

Proof

$$
\begin{aligned}
\operatorname{tr}(C_x \Gamma C_y \Gamma^T) = \ & \operatorname{tr}((\mathbf{1} m_x^T - 2X^T X + m_x \mathbf{1}^T)\Gamma(\mathbf{1} m_y^T - 2Y^T Y + m_y \mathbf{1}^T)\Gamma^T) \\
&= m_x^T \Gamma(\mathbf{1} m_y^T - 2Y^T Y + m_y \mathbf{1}^T)\Gamma^T \mathbf{1} \\
&\quad - 2\operatorname{tr}(X^T X \Gamma(\mathbf{1} m_y^T - 2Y^T Y + m_y \mathbf{1}^T)\Gamma^T) \\
&\quad + \mathbf{1}^T \Gamma(\mathbf{1} m_y^T - 2Y^T Y + m_y \mathbf{1}^T)\Gamma^T m_x \\
&= m_x^T \mathbf{1} m_y^T \mathbf{1} - 2m_x^T \Gamma Y^T Y \mathbf{1} + n m_x^T \Gamma m_y \\
&\quad - 2m_y^T \Gamma^T X^T X \mathbf{1} + 4\operatorname{tr}(X^T X \Gamma Y^T Y \Gamma^T) - 2\mathbf{1}^T X^T X \Gamma m_y \\
&\quad + n m_y^T \Gamma^T m_x - 2\mathbf{1}^T Y^T Y \Gamma^T m_x + \mathbf{1}^T m_y \mathbf{1}^T m_x \\
&= 4\operatorname{tr}(X^T X \Gamma Y^T Y \Gamma^T) - 4m_x^T \Gamma Y^T Y \mathbf{1} + 2n m_x^T \Gamma m_y \\
&\quad - 4\mathbf{1}^T X^T X \Gamma m_y + 2\mathbf{1}^T m_y \mathbf{1}^T m_x \\
&= \langle 2X\Gamma Y^T, 2X\Gamma Y^T \rangle \\
&\quad + \langle 2n m_x m_y^T - 4m_x \mathbf{1}^T Y^T Y - 4X^T X \mathbf{1} m_y^T, \Gamma \rangle \\
&\quad + 2\mathbf{1}^T m_y \mathbf{1}^T m_x.
\end{aligned}
$$

Proof of Theorem 1

The gap between the upper bound and lower bound in this algorithm will converge to 0 (if the tolerance is $\epsilon = 0$)

Let (w_N, W_N) = optimal solution to (7)

$\mathbf{\Gamma}_N$ = optimal solution to (9) with corresponding points $(\widehat{w}, \widehat{W}) = (\langle L, \mathbf{\Gamma}_N \rangle, 2X\widehat{\mathbf{\Gamma}}_N Y^T)$

$$\epsilon_N = UBound - LBound$$

$$\epsilon_N = \|W_N\|_F^2 - w_N - \left\|\widehat{W}\right\|_F^2 - \widehat{w}$$

The new constraint is $\langle Z_{N+1}, W \rangle + w \leq \beta_{N+1}$

Where $Z_{N+1} = 2W_N$ and $\beta_{N+1} = 2\langle W_N, \hat{W} \rangle + \hat{w}$

$$0 \leq 2\langle W_N, \hat{W} - W \rangle + \hat{w} - w.$$

$$\begin{aligned} \epsilon_N &\leq 2\langle W_N, \hat{W} - W \rangle - w + \|W_N\|_F^2 + w_N - \|\hat{W}\|_F^2 \\ &= w_N - w + 2\langle W_N, W_N - W \rangle - \|\hat{W} - W_N\|_F^2 \\ &\leq w_N - w + 2\langle W_N, W_N - W \rangle \\ &\leq (|w_N - w|^2 + \|W_N - W\|_F^2)^{1/2}(1 + 4\|W_N\|_F^2)^{1/2} \end{aligned}$$

$$\frac{\epsilon}{(1 + 4\max\{\|W\|_F^2 \mid W \in (8)\})^{1/2}} \leq (|w_N - w|^2 + \|W_N - W\|_F^2)^{1/2}$$