

Autoencoder with Dynamic Optimal Transport

Xue Feng

PhD, Department of Mathematics, UC Davis

Women in Optimal Transport
April 17th, 2024

Feng, Xue, and Thomas Strohmer. "Improving Autoencoder Image Interpolation via Dynamic Optimal Transport." arXiv preprint arXiv:2404.08900 (2024).

Dynamic Optimal Transport

- Dynamic optimal transport DIRECTLY finds the geodesic path $\rho(t, s)$ between $\rho_0(s)$ and $\rho_T(s)$ by minimizing the kinetic energy along the path:

Given ρ_0, ρ_T , solve ρ

$$\min_{\rho, \mathbf{v}} \frac{1}{2} \int_0^T \int_{[0,1]^2} \rho(t, s) |\mathbf{v}(t, s)|^2 ds dt,$$

$$\text{s.t. } \partial_t \rho + \nabla \cdot (\rho \mathbf{v}) = 0, ,$$

$$\rho(0, \cdot) = \rho_0, \quad \rho(T, \cdot) = \rho_T,$$

Dynamic Optimal Transport (Benamou 2000)

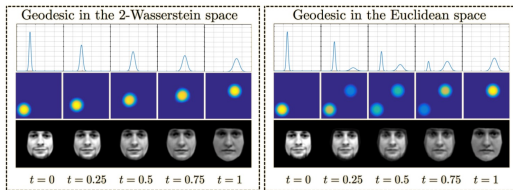


Figure : Interpolation in the optimal transport framework (left) and Euclidean space (right)

- Reformulate it to a convex problem:

$$\begin{aligned} \min_{\rho, \mathbf{m}} \quad & J(\rho, \mathbf{m}) = \frac{1}{2} \int_0^T \int_{[0,1]^2} \frac{|\mathbf{m}(t, s)|^2}{\rho(t, s)} ds dt, \\ \text{s.t.} \quad & \partial_t \rho + \nabla \cdot (\mathbf{m}) = 0, \\ & \rho(0, \cdot) = \rho_0, \quad \rho(T, \cdot) = \rho_T. \end{aligned}$$

- Discretize ρ on a centered grid, \mathbf{m} on a staggered grid, then

$$\begin{aligned} \min_{\rho, \mathbf{m}} \quad & \sum_{t=0,2,\dots,T-1} \mathbf{m}_t^T \text{Diag}(\mathbf{w}_t) \mathbf{m}_t \\ \text{s.t.} \quad & b_t + \nabla \cdot (\mathbf{m}_t) = 0, \quad t = 0, 1, 2, \dots, T-1, \end{aligned}$$

where

$$\mathbf{w}_{t,i,j}^1 = \frac{2}{\rho_{t,i,j} + \rho_{t,i+1,j}}, \quad \mathbf{w}_{t,i,j}^2 = \frac{2}{\rho_{t,i,j} + \rho_{t,i,j+1}}, \quad b_t = \rho_{t+1} - \rho_t.$$

- Solver: Proximal Splitting Methods such as the Douglas–Rachford (DR) algorithm, the alternating direction method of multipliers (ADMM), and a primal-dual (PD) algorithm

Dynamic Optimal Transport Variants

$$\begin{aligned} \min_{\rho, \mathbf{m}} \quad & \frac{1}{2} \int_0^T \int_{[0,1]^2} \frac{|\mathbf{m}(t, s)|^2}{\rho(t, s)} ds dt, \\ \text{s.t.} \quad & \partial_t \rho + \nabla \cdot (\mathbf{m}) = 0, \\ & \mathbf{m}_C = 0, \\ & \rho(0, \cdot) = \rho_0, \quad \rho(T, \cdot) = \rho_T. \end{aligned}$$

when there are obstacles in the environment

$$\begin{aligned} \min_{\rho, \mathbf{m}} \quad & \frac{1}{2} \int_0^T \int_{[0,1]^2} \frac{|\mathbf{m}(t, s)|^2}{\rho(t, s)} + \tau \frac{|s(t, s)|^2}{\rho(t, s)} ds dt \\ \text{s.t.} \quad & \partial_t \rho + \nabla \cdot (\mathbf{m}) = s, \\ & \rho(0, \cdot) = \rho_0, \quad \rho(T, \cdot) = \rho_T \end{aligned}$$

unbalanced OT where $\int_X \rho_0(s) ds \neq \int_X \rho_T(s) ds$

Our algorithm: Solve Dynamic OT with Autoencoder

- key 1: eliminate the momentum m to make the minimization problem unconstrained
- key 2: parameterize the path ρ using autoencoder

Step 1: eliminate \mathbf{m} to make the problem unconstrained

- Fix ρ , we define the path energy over ρ as below:

$$J(\rho) = \min_{\mathbf{m}} \sum_{t=0,1,2,\dots,T-1} \mathbf{m}_t^T \text{Diag}(\mathbf{w}_t) \mathbf{m}_t$$

s.t. $\nabla \cdot (\mathbf{m}_t) = b_t, t = 0, 1, 2, \dots, T - 1$

Note that \mathbf{w} and b are both defined using ρ

- This is a quadratic problem with linear constraint, and its KKT condition is given by

$$\begin{bmatrix} \text{Diag}(\mathbf{w}_t) & \nabla \cdot^T \\ \nabla \cdot & 0 \end{bmatrix} \begin{bmatrix} \mathbf{m}_t \\ \lambda_t \end{bmatrix} = \begin{bmatrix} \mathbf{0} \\ b_t \end{bmatrix}, t = 0, 1, 2, \dots, T - 1, \quad (1)$$

where λ_t is the Lagrange multiplier.

- After solving the KKT condition, we have

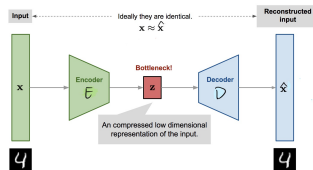
$$J(\rho) = \sum_{t=0}^{T-1} b_t^T \left(\nabla \cdot \text{Diag}(\mathbf{w}_t)^{-1} \nabla \cdot^T \right)^{-1} b_t,$$

Path energy function

Step 2: parameterize the path variable ρ using autoencoder

$$J(\rho) = \sum_{t=0}^{T-1} b_t^T \left(\nabla \cdot \text{Diag}(\mathbf{w}_t)^{-1} \nabla^T \right)^{-1} b_t,$$

Path energy function



Autoencoder

- Motivation: use generator to produce ρ to achieve smooth transitions along $\rho(t)$
- We adopted the decoder of the autoencoder, denoted as \mathcal{D} , as the generator:

$$\rho(t) = \mathcal{D}(tz_0 + (1-t)z_1), \quad 0 \leq t \leq 1,$$

where z_0, z_1 are the latent code of input ρ_0, ρ_1 .

- Denote the autoencoder parameter as θ , then

$$\min_{\rho} J(\rho) \longrightarrow \min_{\theta} J(\rho_{\theta})$$

optimization problem \longrightarrow NN training

Our algorithm for image interpolation

- When we have two data in the training dataset,

Step 1: train an autoencoder whose loss function is

$$\|\hat{\rho}_0 - \rho_0\|^2 + \|\hat{\rho}_1 - \rho_1\|^2 + \alpha J(\mathcal{D}(t\mathcal{E}(\rho_1) + (1-t)\mathcal{E}(\rho_2)))$$

Step 2: generate the interpolation between ρ_0 and ρ_1 using $\mathcal{D}(t\mathcal{E}(\rho_1) + (1-t)\mathcal{E}(\rho_2))$

- When we have multiple data in the training dataset,

Step 1: train an autoencoder whose loss function is

$$\sum_i \|x_i - \hat{x}_i\|^2 + \alpha \sum_{i,j} J(\mathcal{D}_{i \rightarrow j})$$

Step 2: generate the interpolation between ρ_i and ρ_j using $\mathcal{D}(t\mathcal{E}(\rho_i) + (1-t)\mathcal{E}(\rho_j))$

- Comparison with normalizing flow(NF):
 - Recall $\partial_t \rho = -\nabla \cdot (\rho(t)\mathbf{v}(t))$
 - normalizing flow use NN to generate $\mathbf{v}(t)$, and then push ρ_0 to generate $\rho(t)$ using integration; but the target density may not be matched.
 - we generate a path between ρ_0 and ρ_1 directly; \mathbf{v} is baked into the loss function.

Experiment results

- (a) The result of our proposed method. (b) The result of the proximal splitting method

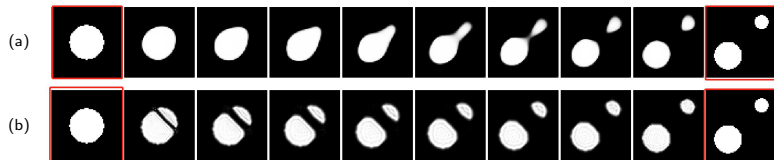
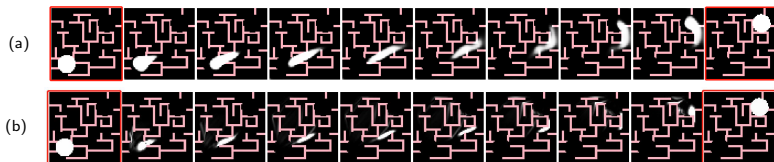
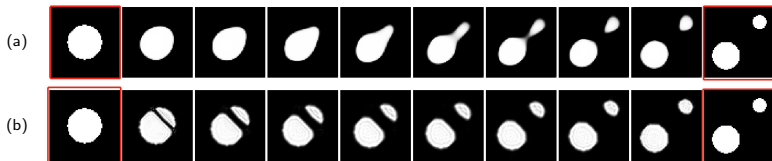


Figure: example (1)

Experiment results

- (a) The result of our proposed method. (b) The result of the proximal splitting method



Experiment results

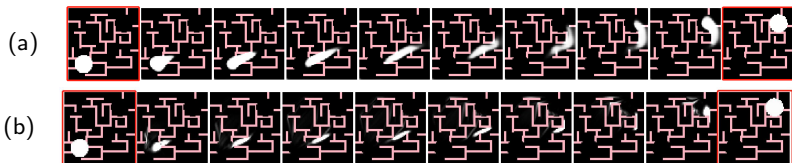
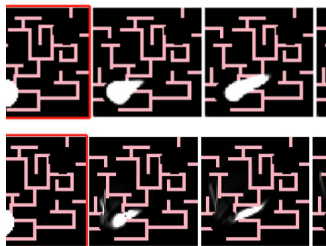


Figure: example when obstacles in the environment (marked pink) are present

- Our algorithm: Solve Dynamic OT with Autoencoder
- Feature of our interpolation results:
 - follow the least energy principle
 - shows a smooth effect visually (a trade off of smoothness and accuracy)
 - works from limited training data to large training data(in my paper)

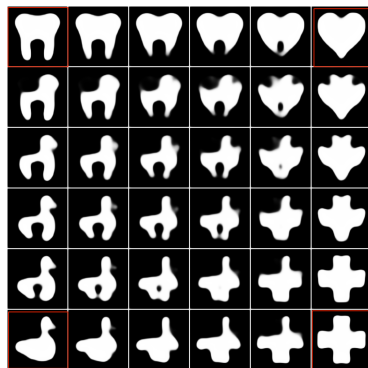
Exploration of the output space of our Trained Autoencoder(on going)

Example

Our method:

- Step 1: train an autoencoder with loss function $\sum_i ||x_i - \hat{x}_i||^2 + \alpha \sum_{i,j} J(\mathcal{D}_{i \rightarrow j})$
- Step 2: generate the barycenter by decoding a convex combination of their corresponding latent codes

$$\mathcal{D}(c_1 z_1 + c_2 z_2 + c_3 z_3 + c_4 z_4), \quad \sum_i c_i = 1.$$



there are only four images (at the corner) in the training dataset.

- The output space is a smooth manifold even with limited training data; $\mathcal{W}(\mathcal{D}(\sum_i \hat{c}_i z_i), \mathcal{D}(\sum_i \tilde{c}_i z_i))$ is small when $\hat{c}, \tilde{c} < \epsilon$
- ongoing work: application on signal recovery

Thanks!

For any further questions, feel free to contact: xffeng@ucdavis.edu