Wasserstein Wormhole (ICML 2024 proceeding)

Like Ma*
SDS lab of Gaoling School of Artificial Intelligence
Renmin University of China

June 6th, 2024

Contents

Preliminaries

- Discrete optimal transport Given 2 discrete distributions, say $P_x = \Sigma_{i=1}^{n} \mu_i \delta_{x_i}$ and $P_y = \Sigma_{j=1}^{m} \nu_j \delta_{x_j}$ (where δ is the Dirac measure), the optimal transport problem is to find the plan matrix in the feasible set $U(\mu, \nu) := \{P \in \mathbb{R}_+^{n,m} : \Sigma_{j=1}^{m} P_{i,j} = \nu_i, \Sigma_{i=1}^{n} P_{i,j} = \mu_j\}$ minimizing the total cost:

$$\min_{P \in U(\mu,\nu)} \Sigma_{i,j=1}^{n,m} P_{i,j} C_{i,j}. \tag{1}$$

- When the cost function $C_{i,j} = c(x_i, y_j) = |x_i - y_j|^p$, we call such OT distance the Wassestein-p distance. In most applications, we usually compute the entropic Wasserstein distance instead the primal one to compute much faster and a little bit less accurate by adding a regularizer $\epsilon H(P)$, where $H(P) = -\Sigma_{i,j} P_{i,j} log(P_{i,j} - 1)$.

Motivation

Since computation of OT distances is laborious, since even the Sinkhorn algorithm and its variants (namely computing entropic OT instead) have time complexity $O(n^2)$, but in practice, problems can be complex with large scale such as aspects:

- computational biology (Nitzan et al., 2019; Schiebinger et al., 2019; Bunne et al., 2023),
- numerical geometry (Su et al., 2015),
- numerical chemistry (Wu et al., 2023),
- image processing (Feydy et al., 2017).

Also, interpretability has not been studied in the explicit computation of Wasserstein barycenters (Cuturi, 2014) or interpolate between point clouds (Chewi et al., 2021).
The difficulties are tackled by the Wormhole method in this paper, which is analogous to multidimensional scaling (Torgerson, 1952), and both methods try to find an embedding that preserves distances between samples.

the Fundamental idea

Computing the distance in Euclidean space is much easier than computing the Wasserstein distance, and the Wasserstein spaces are even infinitely dimensional, so we try to find a set of points $\{x_1, ..., x_n\}$ in Euclidean space which can represent points clouds (X_i, X_j) in the Wasserstein spaces, keeping $|x_i, x_j| = W_{p,\epsilon}(X_i, X_j)$ in the computation of regularized Wasserstein distance. With the Wasserstein diatence matrix $D_{i,j} = W_{p,\epsilon}(X_i, X_j)$, it is natural to consider the notion of Euclidean distance matrix below:

定义

An N-dimensional (square) distance matrix D is Euclidean if there exists a set of points $X = x_1, ..., x_N$ in $x_i \in R^d$ such that for all i, j: $|x_i, x_j| = W_{p,\epsilon}(X_i, X_j) = D_{i,j}$.

the Fundamental idea

定理

From (Gower, 1985), a matrix D is an Euclidean Distance Matrix if and only if 1. all elements are non-negative, 2. all diagonal elements are 0 3. the criterion matrix F=−JDJ is positive-semidefinite (PSD) where $J = I_N - \frac{1_N 1_N^T}{N}$ *is the centering matrix and* 1_N *is the N-dim 1 vector.*

The definition of a metric function requires the distance of each point to itself to be 0, but due to the added regularization term, the $W_{p,\epsilon}$ of 2 same distributions is strictly negative, which can introduce biases when used to optimize generative models. Thus, we use Sinkhorn divergence defined below:

$$S_{p,\epsilon}(X_i, X_j) = W_{p,\epsilon}(X_i, X_j) - \frac{W_{p,\epsilon}(X_i, X_i) + W_{p,\epsilon}(X_j, X_j)}{2}. \tag{2}$$

Due to the large cohort size, we let Wormhole be optimized by mini-batches to minimize the discrepancy between the embedding pairwise distances and the pairwise Wasserstein distances of the batch point clouds.

Illustration of the idea

Why Wasserstein distance matrix is non-Euclidean? Consider the set of distributions supported on points $x_1 = (0,0), x_2 = (0,1), x_3 = (1,0), x_4 = (1,1)$, where the probability of each points is either $\{0, 0.25, 0.5, 0.75, 1\}$ with sum 1, then there are $N = C_7^4 = 35$ (think on a grid 4x4) possible point clouds. Let D be the OT distance matrix (with the coordinates), the double centered Wasserstein distance matrix (−JDJ) must be positive semi-definite to be Euclidean, but D is not.

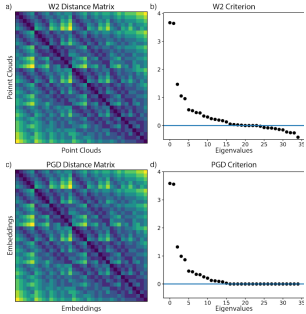


Figure 1: The approximated matrix (35x35) is, by PGD (discussed later)

Warmhole

the dataset in our setting is a cohort of N point clouds $\Omega = \{X_1, ..., X_N\}$, where each point cloud X_i contains n_i points in $\mathbb{R}^d$. To find T(X) that encodes X_i into embeddings α_i, whose Euclidean distances approximate W^p distances $D_{i,j}$. By computing the pairwise Wasserstein distance matrix for small mini-batches, we can train T(X) so that encoding distances directly match Wasserstein, or minimizing $\Sigma_{i,j}(\|\alpha_i - \alpha_j\|^2 - D_{i,j})^2$ (difficulty: choosing $\{\alpha_1, ...\alpha_N\}$, non-convex). Since point clouds contain varying numbers of samples, and Wasserstein distance is invariant with respect to the order of samples within point clouds ($W_p(X, PY) = W(X, Y)$, for P a permutation matrix), $T(x)$ has 2 requirements:

1. T(X) suits any point cloud sizes.
2. be permutation invariant: T(X) =T(PX)

Transformers fit both, so it is to encode. The proof sketch is to use the fact that P is unitary, and softmax function is pointwise, so for Attention A:

$$A(PX) = softmax(\frac{PXW_Q W_K^T X^T P^T}{\sqrt{d}})PXW_V = P \cdot softmax(\frac{XW_Q W_K^T X^T P^T P}{\sqrt{d}}) = PA(X) \quad (3)$$

the Warmhole encoder structure

A well-trained encoder can give us relatively accurate (compared with other accelerating methods) with shorter time, since the computation of Eulidean distance (between α_i, α_j as above) requires only linear time (O(d)).

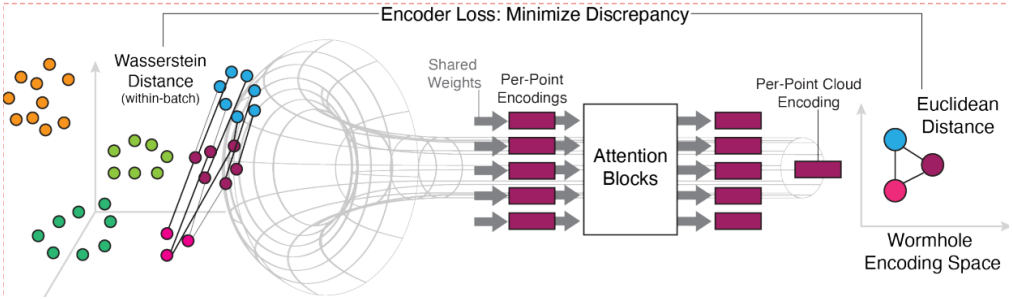


Figure 2: Three clouds are mapped to 3 points in $\mathbb{R}^d$ in corresponding colours, isometrically.

The Warmhole decoder (not shown) is a second transformer maps points in Euclidean space back to clouds, and it will to be discussed next page.

Warmhole: the training

At each training step, a mini-batch of clouds is randomly sampled (from the entire training set Ω). We train encoder and decoder by stochastic gradient descent method so that pairwise distance between mini-batch embeddings matching their Sinkhorn divergence $S_{p,\epsilon}$ and decodings reconstructing input point clouds. The matrix distance is measured in the Frobenius-2 norm (L_{enc}) as above.

Input: point clouds $\Omega = \{X_i\}_{i=1}^{N}$, initialized encoder T and decoder G networks, batch size B, learning rate ε_t

repeat

- Sample B point clouds $\{x_i\}_{i=1}^{B}$ from Ω
- Calculate encodings $\alpha_i = T(x_i)$
- Compute pairwise OT in batch $D_{i,j} = S_\varepsilon(x_i, x_j)$
- Evaluate stress $\mathcal{L}_{enc} = \sum_{i,j}(\|\alpha_i - \alpha_j\|_2^2 - D_{i,j})^2$
- Produce decodings $\hat{x}_i = G(\alpha_i)$
- Evaluate decoding error $\mathcal{L}_{dec} = \sum_i S_\varepsilon(x_i, \hat{x}_i)$
- Update encoder $T \leftarrow T - \varepsilon_t \nabla(\mathcal{L}_{enc} + \mathcal{L}_{dec})$
- Update decoder $G \leftarrow G - \varepsilon_t \nabla(\mathcal{L}_{dec})$

until Convergence

The decoder explains why the model is more accurate, and allows more general operations in the OT spaces, such as barycenter estimation and OT interpolation.

PGD to check the solution

Recall the characteristic theorem of matrix being Euclidean, the matrix D' satisfies the 3 conditions (C1) Positive semi-definite $F = -JD'J$. (C2) Hollow: diagonal entries all 0. (C3) All entries non-Negative. Set of matrices satisfying each condition is convex, thus so is their intersection (thus the paper has a typo). The projected gradient descent method thus gives an accurate unique global optimizer, since the problem below is simplified to a convex problem (satisfying C1 C2 C3).

$$L = \min_{\hat{D}} \Sigma_{i,j} (\hat{D}_{i,j} - D_{i,j})^2 \tag{4}$$

Input: Distance Matrix D, Initial Solution D'_0, Learning Rate ε_t

repeat

$D^*_{t+1} = D'_t - \varepsilon_t \nabla(\|D'_t - D\|^2_F)$ {GD Step}

$D'_{t+1} = \text{proj}_{C_1 \cup C_2 \cup C_3}(D^*_{t+1})$ {Dykstra}

until Convergence

RETURN $X = \text{cMDS}(D')$ {Embed D' with MDS}

Also, the projection to $C_2 \bigcap C_3$ is simple (by letting the diagonal be 0, and all other entries compared with 0, with the larger one left).

bounds of the PGD

Only for small cohorts, the theoretical assurances (the 2 theorems below) of PGD provide a useful benchmark for Wormhole.

定理

For a given distance matrix D and the eigen-decomposition $\{\lambda_i, v_i\}_{i=1}^{N}$ *of the matrix –JDJ, the optimal has a lower bound:*

$$L^* \geq L = \Sigma_{i:\lambda_i<0}\lambda_i^2 \tag{5}$$

定理

The optimal has an upper bound with ($\delta g_i = \frac{1}{2}\Sigma_{i:\lambda_i<0}\lambda_i \cdot v_{i,j}^2$)*:*

$$L^* \leq U = \Sigma_{i,j}(\delta g_i + \delta g_j)^2 + \Sigma_{i:\lambda_i<0}\lambda_i^2, \tag{6}$$

where $v_{i,j}$ *is the j-th element of i-th eigenvector of -JDJ.*

proof sketch of the bounds

For D' satisfying the 3 conditions, with P the square matrix NxN of $\frac{1}{N}$, we decompose $D - D' = A + B$, where $A = JDJ - JD'J$
$B = PD'J + JD'P + PD'P - PDJ - JDP - PDP$, first, by direct computation : $\|D - D'\|_F^2 = \|A\|_F^2 + \|B\|_F^2$, then a minimizer minimizing both A and B parts satisfies 1 constraint out of 3 only, which forces the lower bound estimation.
By enlarging the B part with correct A part,

$$\Sigma_{i,j}(\delta g_i + \delta g_j)^2, \tag{7}$$

which concludes the second.
In the proof on the first theorem, we can find for matrix K:

$$proj_{C_1}(K) = PK + (PK)^T + PKP - \Sigma_{i:\lambda_i<0}\lambda_i \cdot v_i v_i^T. \tag{8}$$

By putting this together with the computation of the projection on $C_2 \bigcap C_3$, we are enabled to find the Dykstra step in Algorithm 2 easily.

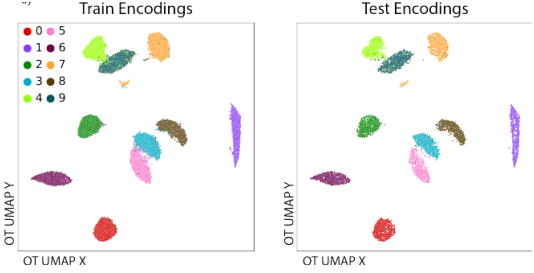
Experiments

Classical MDS produces far from optimal errors, this is why we just use it to embed back as a distribution at the last step.

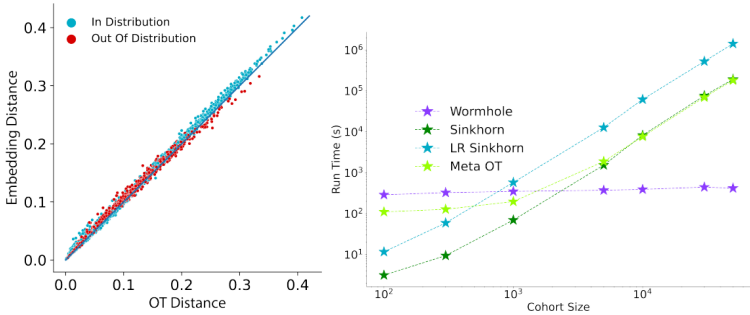
Dataset	Lower	Upper	PGD	Wormhole	cMDS
Simplex (35)	0.765	6.369	1.117	1.420	7.134
Gaussian (128)	0.129	2.552	0.168	0.401	2.681
MNIST (256)	0.042	0.616	0.058	0.100	0.657

Figure 3: The lower and upper bounds of the loss

The encodings of traning and test sets of MNIST (2-dim) are nearly identical with UMAP visualization.



Experiments: computation correlation and the speed



The method is also robust in the out-of-distribution (OOD for abbreviation) case (in an MNIST dataset for example), whose computation speed (Warmhole) is also faster than all other acceleraters in the entire MNIST dataset.
The closer correlation to 1, the better the approximtion is. Since the correlations are between actual and approximated Wasserstein distances (divenrgences).

Experiments (Accuracy included)

Finally, let us consider the model classifier's accuracy (label accuracy). The table below is a summary (including accurary) of each dataset used in this study and results from every benchmarked algorithm, where cloud size denotes the median number of points in each point cloud for every dataset, and OOM stands for out of memory.

	Dataset Parameters			OT Correlation				Label Accuracy			
Name	Cohort Size	Cloud Size	Dimension	Wormhole	DiffusionEMD	DWE	Fréchet	Wormhole	DiffusionEMD	DWE	Fréchet
MNIST	70,000	105	2	**0.98**	0.845	0.92	0.86	**0.98**	0.84	0.97	0.49
Fashion-MNIST	70,000	356	2	**0.99**	0.97	0.98	0.99	**0.87**	0.77	**0.87**	0.73
ModelNet40	12,311	2048	3	**0.99**	0.85	0.97	0.95	**0.86**	0.69	0.78	0.61
ShapeNet	15,011	2048	3	**0.99**	0.81	0.97	0.97	**0.98**	0.94	0.97	0.92
Rotated ShapeNet (GW)	15,011	512	3	**0.98**	NA	NA	0.68	**0.82**	NA	NA	0.16
MERFISH Cell Niches	256,058	11	254	**0.97**	OOM	OOM	OOM	**0.98**	OOM	OOM	OOM
SeqFISH Cell Niches	57,407	29	351	**0.98**	OOM	OOM	OOM	**0.97**	OOM	OOM	OOM
scRNA-seq Atlas	2,185	69	2500	**0.98**	OOM	OOM	OOM	**0.96**	OOM	OOM	OOM

From the set we can see that at both accuracy an correaltion, the model is always the best among the algorithms. Further, its lower complexity also leads to lower memory required (never OOM).

Conclusion

The 7 authors initiated a novel method (Wasserstein Wormhole) to scale computations of Wasserstein distances to large cohorts, which also shows the potential to accelerate $GW_{2,\epsilon}$ computation (the GW row, though without any derivation), since the theoretical foundation is approximating an Euclidean matrix. The main advantages compared with other methods are:

- higher accuracy
- compute faster
- higher correlation

thanks for listening!