Large-scale optimal transport map estimation using projection pursuit

Cheng Meng, Yuan Ke, Jingyi Zhang, Mengrui Zhang,
Wenxuan Zhong, Ping Ma (NeurIPS 2019)

Presenter: Yue Xiang

12.16

- Introduction
- Background
- Projection Pursuit Monge Map Method
- Theoretical Results
- Experiments and Experiments and Summary
- Extensions and Summary

Introduction

- Optimal transport plays essential roles in various machine learning applications, *e.g.*, generative model, color transfer and transfer learning.

Introduction

- Optimal transport plays essential roles in various machine learning applications, *e.g.*, generative model, color transfer and transfer learning.
- The computation of **o**ptimal **t**ransport **m**ap (OTM) is challenging for a large-scale sample with massive sample size and/or high dimension.

Introduction

- Optimal transport plays essential roles in various machine learning applications, *e.g.*, generative model, color transfer and transfer learning.
- The computation of **o**ptimal **t**ransport **m**ap (OTM) is challenging for a large-scale sample with massive sample size and/or high dimension.
  - Estimating the OTM between $\{x_i\}_{i=1}^n \in \mathbb{R}^d$ and $\{y_i\}_{i=1}^n \in \mathbb{R}^d$ requires $\mathcal{O}\left(n^3 \log(n)\right)$.

Existing OTM Estimation Methods

Projection-based method: Random projection method/ sliced method/...

Existing OTM Estimation Methods

Projection-based method: Random projection method/ sliced method/...
Breaking the d-dimensional OTM estimation into a series of 1-d subproblems:

- Denote $\mathbb{S}^{d-1}$ as the d-dimensional unit sphere.

Existing OTM Estimation Methods

Projection-based method: Random projection method/ sliced method/...
Breaking the d-dimensional OTM estimation into a series of 1-d subproblems:

- Denote $\mathbb{S}^{d-1}$ as the d-dimensional unit sphere.
- In each iteration, pick a random direction $\theta \in \mathbb{S}^{d-1}$, and calculate the 1-d OTM between the projected samples $\{\boldsymbol{x}_i^\top \boldsymbol{\theta}\}_{i=1}^n$ and $\{\boldsymbol{y}_i^\top \boldsymbol{\theta}\}_{i=1}^n$ by sorting.

Existing OTM Estimation Methods

Projection-based method: Random projection method/ sliced method/...
Breaking the d-dimensional OTM estimation into a series of 1-d subproblems:

- Denote $\mathbb{S}^{d-1}$ as the d-dimensional unit sphere.
- In each iteration, pick a random direction $\theta \in \mathbb{S}^{d-1}$, and calculate the 1-d OTM between the projected samples $\{\boldsymbol{x}_i^\top \boldsymbol{\theta}\}_{i=1}^n$ and $\{\boldsymbol{y}_i^\top \boldsymbol{\theta}\}_{i=1}^n$ by sorting.
- The "mean map" of the 1-d OTMs serves as the final estimate of target OTM.

Existing OTM Estimation Methods

Projection-based method: Random projection method/ sliced method/...
Breaking the d-dimensional OTM estimation into a series of 1-d subproblems:

- Denote $\mathbb{S}^{d-1}$ as the d-dimensional unit sphere.
- In each iteration, pick a random direction $\theta \in \mathbb{S}^{d-1}$, and calculate the 1-d OTM between the projected samples $\{\boldsymbol{x}_i^\top \boldsymbol{\theta}\}_{i=1}^n$ and $\{\boldsymbol{y}_i^\top \boldsymbol{\theta}\}_{i=1}^n$ by sorting.
- The "mean map" of the 1-d OTMs serves as the final estimate of target OTM.

Computational complexity: $\mathcal{O}(Kn\log(n))$ (K is the number of iterations).

Limitations of Projection-based Methods

- Empirically, the existing projection-based approaches usually require a large number of iterations to convergence/ fail to converge.

Limitations of Projection-based Methods

- Empirically, the existing projection-based approaches usually require a large number of iterations to convergence/ fail to converge.
- We speculate that the slow convergence is because a randomly selected projection direction may not be "informative" enough.

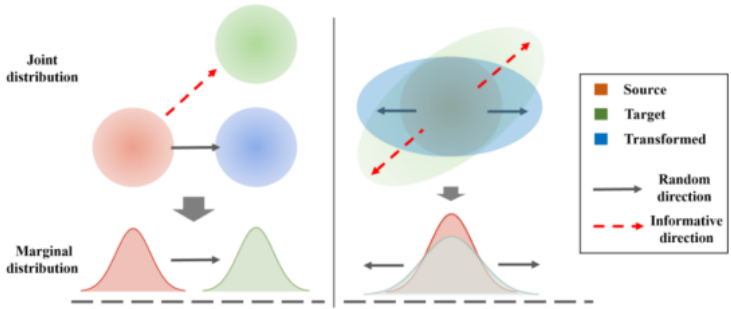


Figure 1: Illustration for the "informative" projection direction

Contributions

- A novel projection-based approach to estimate large-scale OTMs: projection pursuit Monge map (PPMM).
- Use a sufficient dimension reduction technique to estimate the most "informative" projection direction in each iteration.
- Show the convergence of algorithms.

- Introduction
- Background
- Projection Pursuit Monge Map Method
- Theoretical Results
- Experiments
- Extensions and Summary

First let’s go through 3 concepts and try to find their relations...

- Optimal transport
- Projection pursuit
- Sufficient dimension reduction

Optimal Transport Map

- Given 2 continuous random variables $X \in \mathbb{R}^d$, $Y \in \mathbb{R}^d$ with probability distribution p_X and p_Y, respectively.
- OT problem : to find a transport map $\phi : \mathbb{R}^d \to \mathbb{R}^d$ such that $\phi(X)$ and Y have the same distribution.

Optimal Transport Map

- Given 2 continuous random variables $X \in \mathbb{R}^d$, $Y \in \mathbb{R}^d$ with probability distribution p_X and p_Y, respectively.
- OT problem : to find a transport map $\phi : \mathbb{R}^d \to \mathbb{R}^d$ such that $\phi(X)$ and Y have the same distribution.
- Monge formulation : to find the OTM ϕ^* that realizes the infimum

$$\inf_{\phi \in \Phi} \int_{\mathbb{R}^d} \|X - \phi(X)\|^p \, \mathrm{d}p_X.$$

where Φ is the set of all transport maps.

Wasserstein Distance

- p-order Wasserstein distance:

$$W_p\left(p_X, p_Y\right) = \left(\int_{\mathbb{R}^d} \|X - \phi^*(X)\|^p \ \mathrm{d}p_X\right)^{1/p} \tag{1}$$

- Estimate $W_p\left(p_X, p_Y\right)$:
  $\mathbf{X} = \left(\boldsymbol{x}_1, \ldots, \boldsymbol{x}_n\right)^\top \in \mathbb{R}^{n\times d} \sim p_X$, $\mathbf{Y} = \left(\boldsymbol{y}_1, \ldots, \boldsymbol{y}_n\right)^\top \in \mathbb{R}^{n\times d} \sim p_Y$.

$$\widehat{W}_p(\mathbf{X}, \mathbf{Y}) = \left(\frac{1}{n}\sum_{i=1}^{n} \left\|\boldsymbol{x}_i - \widehat{\phi}\left(\boldsymbol{x}_i\right)\right\|^p\right)^{1/p} \tag{2}$$

Wasserstein Distance

- p-order Wasserstein distance:

$$W_p\left(p_X, p_Y\right) = \left(\int_{\mathbb{R}^d} \|X - \phi^*(X)\|^p \ \mathrm{d}p_X\right)^{1/p} \tag{1}$$

- Estimate $W_p\left(p_X, p_Y\right)$:
  $\mathbf{X} = \left(\boldsymbol{x}_1, \ldots, \boldsymbol{x}_n\right)^\top \in \mathbb{R}^{n \times d} \sim p_X$, $\mathbf{Y} = \left(\boldsymbol{y}_1, \ldots, \boldsymbol{y}_n\right)^\top \in \mathbb{R}^{n \times d} \sim p_Y$.

$$\widehat{W}_p(\mathbf{X}, \mathbf{Y}) = \left(\frac{1}{n} \sum_{i=1}^{n} \left\|\boldsymbol{x}_i - \widehat{\phi}\left(\boldsymbol{x}_i\right)\right\|^p\right)^{1/p} \tag{2}$$

- Wasserstein distance between 2 Gaussians have a closed-form solution.

Projection Pursuit

Projection pursuit regression is widely-used for high-dimensional nonparametric regression models:

$$z_i = \sum_{j=1}^{s} f_j\left(\boldsymbol{\beta}_j^\top \boldsymbol{x}_i\right) + \epsilon_i, \quad i = 1, \ldots, n \tag{3}$$

- s: hyper-parameter,
- $\{z_i\}_{i=1}^n \in \mathbb{R}$: univariate response,
- $\{\boldsymbol{x}_i\}_{i=1}^n \in \mathbb{R}^d$: covariates,
- $\{\epsilon_i\}_{i=1}^n$: i.i.d. normal errors.

Projection Pursuit

- Given response $\{\boldsymbol{z}_i\}$ and covariates $\{\boldsymbol{x}_i\}$.
- Aim: to estimate $\{f_j\}_{j=1}^{s} : \mathbb{R} \to \mathbb{R}$ and $\{\boldsymbol{\beta}_j\}_{j=1}^{s} \in \mathbb{R}^d$.

Projection Pursuit

- Given response $\{z_i\}$ and covariates $\{\boldsymbol{x}_i\}$.
- Aim: to estimate $\{f_j\}_{j=1}^s : \mathbb{R} \to \mathbb{R}$ and $\{\boldsymbol{\beta}_j\}_{j=1}^s \in \mathbb{R}^d$.
- Solution: In the k th iteration ($k = 2, \ldots, s$), we find f_k by fitting a 1-d regression model with the current residuals:
  - $R_i^{[k]} = z_i - \sum_{j=1}^{k-1} \widehat{f}_j \left(\widehat{\boldsymbol{\beta}}_j^\top \boldsymbol{x}_i \right), i = 1, \ldots, n.$
  - $(f_k, \boldsymbol{\beta}_k)$ can be estimated by solving the least squares problem

$$\min_{f_k, \boldsymbol{\beta}_k} \sum_{i=1}^n \left[R_i^{[k]} - f_k \left(\boldsymbol{\beta}_k^\top \boldsymbol{x}_i \right) \right]^2.$$

Projection Pursuit for OT

In the k th iteration:

- Seek a new projection direction for the 1-d OTM in the subspace spanned by the residuals of the previously k− 1 directions.
- A direction in the span of used ones can lead to an inefficient 1-d OTM.

Projection Pursuit for OT

In the k th iteration:

- Seek a new projection direction for the 1-d OTM in the subspace spanned by the residuals of the previously k− 1 directions.
- A direction in the span of used ones can lead to an inefficient 1-d OTM.
- Choose the direction that explains the highest proportion of variations in the subspace.
- (Similar to forward variable selection in step-wise regression).

Projection Pursuit for OT

In the k th iteration:

- Seek a new projection direction for the 1-d OTM in the subspace spanned by the residuals of the previously k− 1 directions.
- A direction in the span of used ones can lead to an inefficient 1-d OTM.
- Choose the direction that explains the highest proportion of variations in the subspace.
- (Similar to forward variable selection in step-wise regression).

We estimate this most “informative” direction with sufficient dimension reduction techniques.

Sufficient Dimension Reduction for Regression

- Z: univariate response.
- X: d-dimensional predictor.
- Goal:
  to reduce the dimension of X while preserving its regression relation with Z

Sufficient Dimension Reduction for Regression

- Z: univariate response.
- X: d-dimensional predictor.
- Goal:
  to reduce the dimension of X while preserving its regression relation with Z
  $\Leftrightarrow$ to seek $\boldsymbol{B} \in \mathbb{R}^{d \times q}$ ($q \leq d$) s.t. $Z \perp\!\!\!\perp X \mid \boldsymbol{B}^\top X$.
  (Z depends on X only through $\boldsymbol{B}^T X$.)

Sufficient Dimension Reduction for Regression

- Z: univariate response.
- X: d-dimensional predictor.
- Goal:
  to reduce the dimension of X while preserving its regression relation with Z
  $\Leftrightarrow$ to seek $\boldsymbol{B} \in \mathbb{R}^{d \times q}$ ($q \leq d$) s.t. $Z \perp\!\!\!\perp X \mid \boldsymbol{B}^\top X$.
  (Z depends on X only through $\boldsymbol{B}^T X$.)
- sliced inverse regression (SIR), sliced average variance estimator (SAVE), directional regression (DR), etc.

- Introduction
- Background
- Projection Pursuit Monge Map Method
- Theoretical Results
- Experiments
- Extensions and Summary

Estimate the most “Informative” Projection Direction

- The direction ξ is most ”informative” when the projected samples $X\xi$ and $Y\xi$ have the largest ”discrepancy.”

Estimate the most “Informative” Projection Direction

- The direction ξ is most ”informative” when the projected samples $X\xi$ and $Y\xi$ have the largest ”discrepancy.”
- The metric of the “discrepancy” depends on the choice of the sufficient dimension reduction technique.
  - When using SAVE (sliced average variance estimator) for calculating $\boldsymbol{B}$, the “discrepancy” metric is the difference between $\mathrm{Var}(X\xi)$ and $\mathrm{Var}(Y\xi)$.

Select the most “Informative” Projection Direction with SAVE

Algorithm 1 Select the most “informative” projection direction using SAVE

Input: two standardized matrix $\mathbf{X} \in \mathbb{R}^{n\times d}$ and $\mathbf{Y} \in \mathbb{R}^{n\times d}$

Step 1: calculate $\widehat{\mathbf{\Sigma}} \in \mathbb{R}^{d\times d}$, i.e., the sample variance-covariance matrix of $\binom{\mathbf{X}}{\mathbf{Y}}$

Step 2: calculate the sample variance-covariance matrices of $\mathbf{X}\widehat{\mathbf{\Sigma}}^{-1/2}$ and $\mathbf{Y}\widehat{\mathbf{\Sigma}}^{-1/2}$, denoted as $\widehat{\mathbf{\Sigma}}_1 \in \mathbb{R}^{d\times d}$ and $\widehat{\mathbf{\Sigma}}_2 \in \mathbb{R}^{d\times d}$, respectively

Step 3: calculate the eigenvector $\boldsymbol{\xi} \in \mathbb{R}^d$, which corresponding to the largest eigenvalue of the matrix $((\widehat{\mathbf{\Sigma}}_1 - I_d)^2 + (\widehat{\mathbf{\Sigma}}_2 - I_d)^2)/4$

Output: the final result is given by $\widehat{\mathbf{\Sigma}}^{-1/2}\boldsymbol{\xi}/||\widehat{\mathbf{\Sigma}}^{-1/2}\boldsymbol{\xi}||$, where $||\cdot||$ denotes the Euclidean norm

Theory of SAVE shows that Algorithm 1 can consistently estimate the most “informative” projection direction.

Projection Pursuit Monge Map Algorithm

Algorithm 2 Projection pursuit Monge map (PPMM)

Input: two matrix $\mathbf{X} \in \mathbb{R}^{n \times d}$ and $\mathbf{Y} \in \mathbb{R}^{n \times d}$
$k \leftarrow 0$, $\mathbf{X}^{[0]} \leftarrow \mathbf{X}$
repeat
(a) calculate the projection direction $\boldsymbol{\xi}_k \in \mathbb{R}^d$ between $\mathbf{X}^{[k]}$ and $\mathbf{Y}$ (using Algorithm 1)
(b) find the one-dimensional OTM $\phi^{(k)}$ that matches $\mathbf{X}^{[k]}\boldsymbol{\xi}_k$ to $\mathbf{Y}\boldsymbol{\xi}_k$ (using look-up table)
(c) $\mathbf{X}^{[k+1]} \leftarrow \mathbf{X}^{[k]} + (\phi^{(k)}(\mathbf{X}^{[k]}\boldsymbol{\xi}_k) - \mathbf{X}^{[k]}\boldsymbol{\xi}_k)\boldsymbol{\xi}_k^{\mathsf{T}}$ and $k \leftarrow k + 1$
until converge
The final estimator is given by $\widehat{\phi} : \mathbf{X} \rightarrow \mathbf{X}^{[k]}$

Projection Pursuit Monge Map Algorithm

Algorithm 2 Projection pursuit Monge map (PPMM)

Input: two matrix $\mathbf{X} \in \mathbb{R}^{n \times d}$ and $\mathbf{Y} \in \mathbb{R}^{n \times d}$
$k \leftarrow 0$, $\mathbf{X}^{[0]} \leftarrow \mathbf{X}$
repeat
(a) calculate the projection direction $\boldsymbol{\xi}_k \in \mathbb{R}^d$ between $\mathbf{X}^{[k]}$ and $\mathbf{Y}$ (using Algorithm 1)
(b) find the one-dimensional OTM $\phi^{(k)}$ that matches $\mathbf{X}^{[k]}\boldsymbol{\xi}_k$ to $\mathbf{Y}\boldsymbol{\xi}_k$ (using look-up table)
(c) $\mathbf{X}^{[k+1]} \leftarrow \mathbf{X}^{[k]} + (\phi^{(k)}(\mathbf{X}^{[k]}\boldsymbol{\xi}_k) - \mathbf{X}^{[k]}\boldsymbol{\xi}_k)\boldsymbol{\xi}_k^{\mathsf{T}}$ and $k \leftarrow k+1$
until converge
The final estimator is given by $\widehat{\phi} : \mathbf{X} \to \mathbf{X}^{[k]}$

Assume Alg 2 converges in K iterations:
Memory cost: $\mathcal{O}(Knd^2)$.
Computational cost:

- Step (a): $\mathcal{O}(nd^2)$ (Alg 1); step (b): $\mathcal{O}(n\log(n))$ (sorting).
- Overall: $\mathcal{O}(Knd^2 + Kn\log(n))$.

- Introduction
- Background
- Projection Pursuit Monge Map Method
- Theoretical Results
- Experiments
- Extensions and Summary

Weak Convergence of PPMM Algorithm

Denote ϕ^* as the d-dimensional optimal transport map from p_X to p_Y and $\phi^{(K)}$ as the PPMM estimator after K iterations, i.e. $\phi^{(K)}(\mathbf{X}) = \mathbf{X}^{[K]}$.

Theorem

Let $K \geq Cd$ for some large enough positive constant C, one has

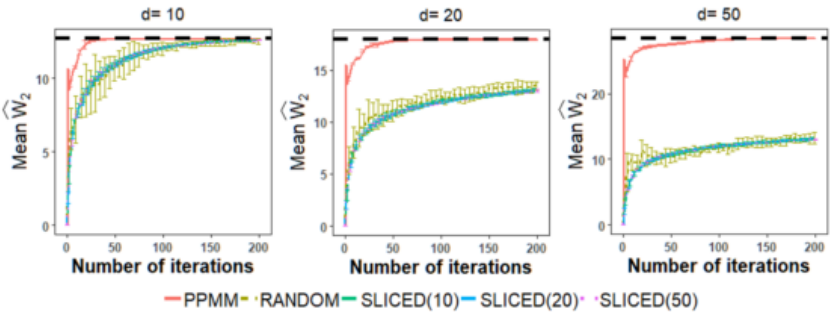
$$\widehat{W}_p\left(\phi^{(K)}(\mathbf{X}), \mathbf{X}\right) \to W_p\left(\phi^*(X), X\right), \quad \textit{and} \quad \phi^{(K)}(\mathbf{X}) \to \phi^*(X) \quad \textit{as} \quad n \to \infty.$$

- Introduction
- Background
- Projection Pursuit Monge Map Method
- Theoretical Results
- Experiments
- Extensions and Summary

Estimation of OTM: $\hat{W}$

- $\mathbf{X} = (\boldsymbol{x}_1, \ldots, \boldsymbol{x}_n)^\top \overset{\text{i.i.d}}{\sim} p_X = \mathcal{N}_d(-\mathbf{2}, \boldsymbol{\Sigma}_X)$, where $\boldsymbol{\Sigma}_X = 0.8^{|i-j|}$ $(i, j = 1, \ldots, d)$.
- $\mathbf{Y} = (\boldsymbol{y}_1, \ldots, \boldsymbol{y}_n)^\top \overset{\text{i.i.d}}{\sim} p_Y = \mathcal{N}_d(\mathbf{2}, \boldsymbol{\Sigma}_Y)$, where $\boldsymbol{\Sigma}_Y = 0.5^{|i-j|}$ $(i, j = 1, \ldots, d)$.
- $n = 10000$, $d = \{10, 20, 50\}$.

We apply PPMM to estimate the OTM between p_X and p_Y from $\{\boldsymbol{x}_i\}_{i=1}^n$ and $\{\boldsymbol{y}_i\}_{i=1}^n$:



Estimation of OTM: Time

CPU Time Per Iteration:

	PPMM	RANDOM	SLICED(10)	SLICED(20)	SLICED(50)
$d = 10$	0.019 (0.008)	0.011 (0.008)	0.111 (0.019)	0.213 (0.024)	0.529 (0.031)
$d = 20$	0.027 (0.011)	0.014 (0.008)	0.125 (0.027)	0.247 (0.033)	0.605 (0.058)
$d = 50$	0.059 (0.036)	0.015 (0.008)	0.171 (0.037)	0.338 (0.049)	0.863 (0.117)

Convergence Time:

	PPMM	RANDOM	SLICED(10)	AUCTIONBF	REVSIM	SHORTSIM
$d = 10$	0.6 (0.1)	4.8 (1.7)	23.0 (2.6)	99.7 (10.4)	40.2 (4.0)	42.5 (3.2)
$d = 20$	2.1 (0.3)	24.4 (3.2)	230.2 (28.4)	109.4 (12.5)	42.6 (5.3)	50.2 (6.6)
$d = 50$	5.5 (0.4)	-	-	125.5 (13.3)	46.5 (5.6)	56.5 (7.1)

Application to generative models

1. Manifold learning: map the training data from the original space $\mathcal{X} \subset \mathbb{R}^d$ to a latent space $\mathcal{Z} \subset \mathbb{R}^{d^*}$ with $d^* \ll d$.

Application to generative models

1. Manifold learning: map the training data from the original space $\mathcal{X} \subset \mathbb{R}^d$ to a latent space $\mathcal{Z} \subset \mathbb{R}^{d^*}$ with $d^* \ll d$.
   The probability distribution of the transformed data in $\mathcal{Z}$ may not be convex, leading to the problem of mode collapse.

Application to generative models

1. Manifold learning: map the training data from the original space $\mathcal{X} \subset \mathbb{R}^d$ to a latent space $\mathcal{Z} \subset \mathbb{R}^{d^*}$ with $d^* \ll d$.
   The probability distribution of the transformed data in $\mathcal{Z}$ may not be convex, leading to the problem of mode collapse.
2. Probability transformation: transporting the distribution in $\mathcal{Z}$ to the uniform distribution $U\left([0,1]^{d^*}\right)$.

Results of Application to Generative Model

We use the “Frechet Inception Distance” (FID) to quantify the similarity between the generated fake sample and the training sample.

- MINST
- Google doodle dataset

	PPMM	RANDOM	SLICED(10)	SLICED(20)	SLICED(50)
MNIST	**0.17** (0.01)	4.62 (0.02)	2.98 (0.01)	3.04 (0.01)	3.12 (0.01)
Doodle (face)	**0.59** (0.09)	8.78 (0.04)	5.69 (0.01)	6.01 (0.01)	5.52 (0.01)
Doodle (cat)	**0.24** (0.03)	8.93 (0.03)	5.99 (0.01)	5.26 (0.01)	5.33 (0.01)
Doodle (bird)	**0.36** (0.03)	7.81 (0.03)	5.44 (0.01)	5.50 (0.01)	4.98 (0.01)

- Introduction
- Background
- Projection Pursuit Monge Map Method
- Theoretical Results
- Experiments
- Extensions and Summary

Extensions and Summary

Extensions:

- Points in source and target samples have non-equal weights.
- Source sample and target sample have different sizes.

Summary...