Regularized Optimal Transport

Marco Cuturi

book with Gabriel Peyré

https://optimaltransport.github.io/

Optimal Transport

\[ \mu = \sum_{i=1}^{n} a_i \delta_{x_i} \]

\[ \nu = \sum_{j=1}^{m} b_j \delta_{y_j} \]
$$\mu = \sum_{i=1}^{n} a_i \delta_{x_i}$$

$$\nu = \sum_{j=1}^{m} b_j \delta_{y_j}$$
Optimal Transport

\[ \mu = \sum_{i=1}^{n} a_i \delta_{x_i} \]

\[ \nu = \sum_{j=1}^{m} b_j \delta_{y_j} \]
Optimal Transport

\[
\mu = \sum_{i=1}^{n} a_i \delta_{x_i}
\]

\[

\nu = \sum_{j=1}^{m} b_j \delta_{y_j}
\]
Kantorovich Problem
Kantorovich Problem
Kantorovich Problem à la française
Optimal Transport

\[ \mu = \sum_{i=1}^{n} a_i \delta_{x_i} \]

\[ \nu = \sum_{j=1}^{m} b_j \delta_{y_j} \]

\((\Omega, D)\)
Consider $\mu = \sum_{i=1}^{n} a_i \delta_{x_i}$ and $\nu = \sum_{j=1}^{m} b_j \delta_{y_j}$.

$M_{XY} \overset{\text{def}}{=} [D(x_i, y_j)^p]_{ij}$

$U(a, b) \overset{\text{def}}{=} \{ P \in \mathbb{R}_+^{n \times m} | P1_m = a, P^T1_n = b \}$

**Def. Optimal Transport Problem**

$$W_p^p(\mu, \nu) = \min_{P \in U(a, b)} \langle P, M_{XY} \rangle$$
Solving the OT Problem

$M_{XY}$

$U(a, b)$
Solving the OT Problem

\[ M_{XY} \]

\[ U(a, b) \]

\[ P^* \]
Early application: Earth Mover’s
Early application: Earth Mover’s
Early application: Earth Mover's

\[ \text{dist}(I_1, I_2) = W_1(\mu, \nu) \]
Word Mover’s Distance

document 1
Obama speaks to the media in Illinois

document 2
The President greets the press in Chicago

μ ‘Obama’ ‘President’ ‘Chicago’ ‘Illinois’
ν ‘greets’ ‘speaks’ ‘media’ ‘press’

word2vec embedding
Word Mover’s Distance

\[ \text{dist}(D_1, D_2) = W_2(\mu, \nu) \]

[\text{Kusner’15}]
Variational OT Problems in ML

Up to 2010: OT solvers used mostly for retrieval in databases of histograms

\[ W_p(\mu, \nu) = ? \]
\[ W_p(\mu, \nu) \leq \ldots ? \]

The field has now transitioned to OT as a loss or fidelity term

\[ \arg\min_{\mu \in \mathcal{P}(\Omega)} F(W_p(\mu, \nu_1), W_p(\mu, \nu_2), \ldots, \mu) = ? \]
\[ \nabla_{\mu} W_p(\mu, \nu_1) = ? \]

Recent spike in interest for \cite{Ambrosio05}
“Wasserstein + Data” Problems

• Quantization: \( k \)-means problem [Lloyd’82]
  \[
  \min_{\mu \in \mathcal{P}(\mathbb{R}^d)} \ W_2^2(\mu, \nu_{\text{data}})
  \]
  \[
  |\text{supp } \mu| = k
  \]

• [McCann’95] Interpolant
  \[
  \min_{\mu \in \mathcal{P}(\Omega)} (1 - t)W_2^2(\mu, \nu_1) + tW_2^2(\mu, \nu_2)
  \]

• [JKO’98] PDE’s as “gradient” flows in \((\mathcal{P}(\Omega), W)\).
  \[
  \mu_{t+1} = \arg\min_{\mu \in \mathcal{P}(\Omega)} J(\mu) + \lambda_t W_p^p(\mu, \mu_t)
  \]
Example: Barycenters
Example: Barycenters

$$\min_{\mu \in \mathcal{P}(\Omega)} \sum_{i=1}^{N} \lambda_i W_p^p(\mu, \nu_i)$$
Example: Barycenters

$\lambda \in \Sigma_3$

Wasserstein mean

$L_2$ mean
Example: Barycenters

\[ \lambda \in \Sigma_3 \]

Wasserstein mean

\[ L_2 \text{ mean} \]
Ex: Barycenters for shapes

Graphics: simple testing ground for relevance of Wasserstein geometry
Ex: Barycenters for shapes

Graphics: simple testing ground for relevance of Wasserstein geometry
Ex: Barycenters for shapes
Example: Learning with a $W$ Loss

Dataset $\{(x_i, y_i)\}$, $x_i \in \mathbb{R}^p$, $y_i \in \mathbb{R}_+^n$

Goal is to find $f_\theta : \text{Images} \mapsto \text{Labels}$
Example: Learning with a $W$ Loss

$$\min_{\theta \in \Theta} \sum_{i=1}^{N} \mathcal{L}(f_{\theta}(x_i), y_i)$$
Example: Learning with a $W$ Loss

$$
\min_{\theta \in \Theta} \sum_{i=1}^{N} \mathcal{L}(f_\theta(x_i), y_i)
$$
Example: Learning with an $W$ Loss

$$\min_{\theta \in \Theta} \sum_{i=1}^{N} \mathcal{L}(f_{\theta}(x_i), y_i)$$

Use for $\mathcal{L}$ a Wasserstein type loss.

[Frogner’15]
Example: Generative Models

We collect data

\[ \nu_{\text{data}} = \frac{1}{N} \sum_{i=1}^{N} \delta x_i \]
Example: Generative Models

We collect data

$$\nu_{\text{data}} = \frac{1}{N} \sum_{i=1}^{N} \delta_{\mathbf{x}_i}$$

We fit a parametric family of densities

$$\{p_{\theta}, \theta \in \Theta\}$$

e.g. $$\theta = (m, \Sigma); p_{\theta} = \mathcal{N}(m, \Sigma)$$
Statistics 0.1: Density Fitting

\( \nu_{\text{data}} \)

\( \rho_{\theta_{\text{done}}} \)

aim for a “good” fit
Maximum Likelihood Estimation

ON AN ABSOLUTE CRITERION FOR FITTING FREQUENCY CURVES.


1. If we set ourselves the problem, in its frequent occurrence, of finding the arbitrary function of known form, which best suit a set of observations, we are met at the outset by an apparently fundamental difficulty which appears to invalidate any results we may attempt to obtain.

\[
\max_{\theta \in \Theta} \frac{1}{N} \sum_{i=1}^{N} \log p_{\theta}(x_i)
\]
Maximum Likelihood Estimation

\[ \max_{\theta \in \Theta} \frac{1}{N} \sum_{i=1}^{N} \log p_{\theta}(x_i) \]

\[ \log 0 = -\infty \]

\[ p_{\theta}(x_i) \text{ must be } > 0 \]
Maximum Likelihood Estimation

Equivalent to a KL projection in the space of probability measures

$$\min_{\theta \in \Theta} \text{KL}(\nu_{\text{data}} \parallel p_{\theta})$$
Maximum Likelihood Estimation

Equivalent to a KL projection in the space of probability measures

\[ \min_{\theta \in \Theta} \text{KL}(\nu_{\text{data}} \| p_{\theta}) \]
In higher dimensional spaces...

Ambient space: hypercube $d = 30.000$
Generative Models
Generative Models

\[ \mu \]

**latent space**

**data space**

\( \nu_{\text{data}} \)
Generative Models

\[ f_{\theta} \] : latent space \rightarrow data space

latent space

data space

\( \mu \)

\( \nu \)_{data}
Generative Models

\[ \mu \]

\[ \mathbf{z} \]

\[ f_\theta : \text{latent space} \rightarrow \text{data space} \]

\[ \mathbf{z} = \begin{bmatrix} .32 \\ .8 \\ .34 \\ \vdots \\ .01 \end{bmatrix} \]
Generative Models

\[ \mathbf{z} = \begin{bmatrix} 0.32 \\ 0.8 \\ 0.34 \\ \vdots \\ 0.01 \end{bmatrix} \]

\[ f_\theta : \text{latent space} \rightarrow \text{data space} \]

\[ f_\theta(\mathbf{z}) \]

\[ \nu_{\text{data}} \]

[Image showing the generative process from latent space to data space with a specific example vector \( \mathbf{z} \).]
Generative Models

\[
f_\theta : \text{latent space} \rightarrow \text{data space}
\]

**Latent space**

\[
z = \begin{bmatrix} .32 \\ .8 \\ .34 \\ \vdots \\ .01 \end{bmatrix}
\]

\[
f_\theta(z)
\]

**Data space**
Generative Models

$f_\theta : \text{latent space} \rightarrow \text{data space}$
Generative Models

Goal: find $\theta$ such that $f_{\theta, \mu}$ fits $\nu_{data}$
Generative Models

Goal: find $\theta$ such that $f_{\theta \# \mu}$ fits $\nu_{\text{data}}$
Generative Models

$f_\theta : \text{latent space} \rightarrow \text{data space}$

Latent space

Data space

$
\max_{\theta \in \Theta} \frac{1}{N} \sum_{i=1}^{N} \log p_\theta(x_i) = \min_{\theta \in \Theta} \text{KL}(\nu_{\text{data}} \parallel p_\theta)
$

MLE
Generative Models

\( f_\theta : \text{latent space} \rightarrow \text{data space} \)

\[
\max_{\theta \in \Theta} \frac{1}{N} \sum_{i=1}^{N} \log f_\theta(\mu(x_i)) = \min_{\theta \in \Theta} \text{KL}(\nu_{\text{data}} || f_\theta(\mu))
\]
Generative Models

$f_\theta : \text{latent space } \rightarrow \text{data space}$

Need a more flexible discrepancy function to compare $\nu_{\text{data}}$ and $f_{\theta\#\mu}$
Workarounds?

- Formulation as adversarial problem [GPM...’14]

$$\min_{\theta \in \Theta} \max_{\text{classifiers}} \text{Accuracy}_g ( (f_{\theta # \mu}, +1), (\nu_{\text{data}}, -1) )$$
• Formulation as adversarial problem [GPM...’14]

$$\min_{\theta \in \Theta} \max_{\text{classifiers } g} \text{Accuracy}_g \left( (f_{\theta \#\mu}, +1), (\nu_{\text{data}}, -1) \right)$$
Workarounds?

• Formulation as adversarial problem [GPM...’14]

$$\min_{\theta \in \Theta} \max_{\text{classifiers } g} \text{Accuracy}_g ((f_{\theta \# \mu}, +1), (\nu_{\text{data}}, -1))$$
• Formulation as adversarial problem \[\text{[GPM...’14]}\]

\[
\min_{\theta \in \Theta} \max_{\text{classifiers } g} \text{Accuracy}_g ((f_{\theta \# \mu}, +1), (\nu_{\text{data}}, -1))
\]
• Formulation as adversarial problem \[ \text{GPM...’14} \]

\[
\min_{\theta \in \Theta} \max_{\text{classifiers } g} \text{Accuracy}_g ((f_{\theta \# \mu}, +1), (\nu_{\text{data}}, -1))
\]
• Formulation as adversarial problem \[ \text{[GPM...’14]} \]

\[
\min_{\theta \in \Theta} \max_{\text{classifiers } g} \text{Accuracy}_g \left( (f_{\theta}; \mu, +1), (\nu_{\text{data}}, -1) \right)
\]
Workarounds?

- Formulation as adversarial problem [GPM...’14]

\[
\min_{\theta \in \Theta} \max_{\text{classifiers}} \text{Accuracy}_g ((f_{\theta \# \mu}, +1), (\nu_{\text{data}}, -1))
\]
Another idea?

- Use a **metric** $\Delta$ for probability measures, that can handle measures with non-overlapping supports:

$$\min_{\theta \in \Theta} \Delta(\nu_{\text{data}}, p_{\theta}), \quad \text{not} \quad \min_{\theta \in \Theta} \text{KL}(\nu_{\text{data}} \| p_{\theta})$$

- The original GAN paper can be interpreted in that light using the Jensen-Shannon divergence.
Minimum $\Delta$ Estimation

MINIMUM CHI-SQUARE, NOT MAXIMUM LIKELIHOOD!

BY JOSEPH BERKSON
Mayo Clinic, Rochester, Minnesota

COMPUTATIONAL STATISTICS & DATA ANALYSIS

Minimum Hellinger distance estimation for Poisson mixtures

Dimitris Karlis, Evdokia Xekalaki*
Department of Statistics, Athens University of Economics and Business, 76 Portision Str., 104 34 Athens, Greece

Available online at www.sciencedirect.com

On minimum Kantorovich distance estimators
Federico Bassetti$^a$, Antonella Bodini$^b$, Eugenio Regazzini$^{a,*}$
we are only interested in the case where the generative models from i.i.d. data with unknown distribution play the role of the discriminator. In addition to the discriminator network, both trained to outwit the generator and the discriminator. In particular, two adversarial nets are used: generative adversarial nets (GAN) and generative moment matching network (GMMN). The GMMN is a generative model that aims to match the moments of the data distribution with the moments of the model-generated distribution. We consider training a deep neural network to generate samples from an unknown distribution. We frame learning as an optimization minimizing a two-sample test based on kernel maximum mean discrepancy (MMD). This work builds on a proposal due to Goodfellow et al. [1]. Their recent example is the Neural Autoregressive Density Estimation (NADE) [10] and deep Boltzmann networks [3]. A successful approach to learning deep generative neural networks: a framework introduced by Goodfellow et al. [1]. Their instantiation is due to MacKay [7], although recent example is the Neural Autoregressive Density Estimation (NADE) [10] and deep Boltzmann networks [3]. A successful approach to learning deep generative neural networks: a framework introduced by Goodfellow et al. [1].

We are particularly interested in the case where the generative models from i.i.d. data with unknown distribution play the role of the discriminator. In addition to the discriminator network, both trained to outwit the generator and the discriminator. In particular, two adversarial nets are used: generative adversarial nets (GAN) and generative moment matching network (GMMN). The GMMN is a generative model that aims to match the moments of the data distribution with the moments of the model-generated distribution. We consider training a deep neural network to generate samples from an unknown distribution. We frame learning as an optimization minimizing a two-sample test based on kernel maximum mean discrepancy (MMD). This work builds on a proposal due to Goodfellow et al. [1]. Their recent example is the Neural Autoregressive Density Estimation (NADE) [10] and deep Boltzmann networks [3]. A successful approach to learning deep generative neural networks: a framework introduced by Goodfellow et al. [1]. Their instantiation is due to MacKay [7], although
Inference in generative models using the Wasserstein distance

Training generative neural networks via Maximum Mean Discrepancy optimization

MMD GAN: Towards Deeper Understanding of Moment Matching Network

Wasserstein Training of Restricted Boltzmann Machines

Generative Model Estimation

Chun-Liang Li1,∗ Wei-Cheng Chang1,∗ Yu Cheng2, Yiming Yang1, Barnabás Póczos1
1 Carnegie Mellon University, 2 IBM Research
{chunliang,wchang2,yiming,bapoczos}@cmu.edu
chengyu@us.ibm.com

Generative Moment Matching Networks

Inference in generative models using the Wasserstein distance

Wasserstein GAN

Gintare Karolina Dziugaite
University of Cambridge

Daniel M. Roy
University of Toronto

Zoubin Ghahramani
University of Cambridge

Grégoire Montavon
Technische Universität Berlin
gregoire.montavon@tu-berlin.de

Klaus-Robert Müller∗
Technische Universität Berlin
klaus-robert.mueller@tu-berlin.de

Marco Cuturi
CREST, ENSAE, Université Paris-Saclay
marco.cuturi@ensae.fr

1 Courant Institute of Mathematical Sciences
2 Facebook AI Research
Generative Model Estimation

MMD GAN: Towards Deeper Understanding of Moment Matching Network

Inference in generative models using the Wasserstein distance

Learning Generative Models with Sinkhorn Divergences

Wasserstein GAN

Improving GANs Using Optimal Transport

Aude Genevay
CEREMADE,
Université Paris-Dauphine

Gabriel Peyré
CNRS and DMA,
École Normale Supérieure

Marco Cuturi
ENSAE CREST
Université Paris-Saclay

Tim Salimans†
OpenAI
tim@openai.com

Han Zhang†
Rutgers University
han.zhang@cs.rutgers.edu

Alec Radford
OpenAI
alec@openai.com

Dimitris Metaxas
Rutgers University
dnm@cs.rutgers.edu

Grégoire Montavon
Technische Universität Berlin
gregoire.montavon@tu-berlin.de

Klaus-Robert Müller†
Technische Universität Berlin
klaus-robert.mueller@tu-berlin.de

Marco Cuturi
CREST, ENSAE, Université Paris-Saclay
marco.cuturi@ensae.fr

Inference in generative models using the Wasserstein distance

Esben Bernton, Mathieu Gerber, Pierre E. Jacob, Christian P. Robert

Martin Arjovsky†, Soumith Chintala‡, and Léon Bottou†‡

†Courant Institute of Mathematical Sciences
‡Facebook AI Research
Optimal Transport in ML

OT is establishing itself as a generic toolbox to handle probability measures in ML tasks.
OT Computations

Discrete - Discrete

Discrete - Continuous

Continuous - Continuous
OT Computations

Discrete - Discrete

Network flow solvers
Auction algorithm
(Entropic) regularization

Discrete - Continuous

low dim.

Continuous - Continuous

[Merigot’11][Kitagawa’16][Levy’15]

Stochastic Optimization

[Genevay’16]

PDE

[Benamou’98]
Solving the OT Problem

\[ M_{XY} \]

\[ U(a, b) \]

\[ P^* \]

\[ \text{min cost flow solver used in practice.} \]

\[ O(n^3 \log(n)) \]
Solving the OT Problem

\[ M_{XY} \]

\[ U(a, b) \]

\[ P^* \]

\[ \min \text{ cost flow solver} \]
\[ O(n^3 \log(n)) \]

Solution \( P^* \) unstable and not always unique.
Solving the OT Problem

\[ M_{XY} \]

\[ U(\mathbf{a}, \mathbf{b}) \]

\[ \{P^*\} \]

\[ \min \text{ cost flow solver used in practice.} \]

\[ O(n^3 \log(n)) \]

Solution \( P^* \) unstable and not always unique.
Solving the OT Problem

\[ M_{XY} \]

\[ U(\alpha, \beta) \]

\[ \{P^*\} \]

*min cost flow solver used in practice.*

\[ O(n^3 \log(n)) \]

Solution \( P^* \) unstable and not always unique.
Solving the OT Problem

\[ M_{XY} \]

\[ U(a, b) \]

\[ P^* \]

**min cost flow solver used in practice.**

\[ O(n^3 \log(n)) \]

**Solution \( P^* \) unstable and not always unique.**
Solving the OT Problem

$M_{XY}$

$U(a, b)$

$P^*$

$W_p^p(\mu, \nu)$ not differentiable.

Solution $P^*$ unstable and not always unique.

$\min$ cost flow solver used in practice.

$O(n^3 \log(n))$
Discrete OT Problem
Discrete OT Problem

An implementation of the Earth Movers Distance.
Based on the solution for the Transportation problem as described in
"Introduction to Mathematical Programming" by F. S. Hillier and

Copyright © 1988 Yossi Rubner
Computer Science Department, Stanford University
E-mail: rubner@cs.stanford.edu URL: http://visi.cs.stanford.edu/~rubner

#include <stdio.h>
#include <stdlib.h>
#include <math.h>

#define "emd.h"
#define DEBUG_LEVEL 0

/*
 * DEBUG_LEVEL:
 * 0 = NO MESSAGES
 * 1 = PRINT THE NUMBER OF ITERATIONS AND THE FINAL RESULT
 * 2 = PRINT THE RESULT AFTER EVERY ITERATION
 * 3 = PRINT ALSO THE FLOW AFTER EVERY ITERATION
 * 4 = PRINT A LOT OF INFORMATION (PROBABLY USEFUL ONLY FOR THE AUTHOR)
 */

#define MAX_SIG_SIZE1 (MAX_SIG_SIZE+1) /* FOR THE POSSIBLE DUMMY FEATURE */

/* NEW TYPES DEFINITION */

/* node1_t IS USED FOR SINGLE-LINKED LISTS */
typedef struct node1_t
{
    int i1;
    double val1;
    struct node1_t *Next;
} node1_t;

/* node1_t IS USED FOR DOUBLE-LINKED LISTS */
typedef struct node2_t
{
    int i1, j1;
    double val1;
    struct node2_t *NextC;
    /* NEXT COLUMN */
    struct node2_t *NextH;
    /* NEXT HEW */
} node2_t;

/* GLOBAL VARIABLE DECLARATION */
static int _1, _2;
/* SIGNATURES SIZES */
static float _1[C[MAX_SIG_SIZE1][MAX_SIG_SIZE1]]; /* THE COST MATRIX */
static node2_t _x[C[MAX_SIG_SIZE1][MAX_SIG_SIZE1]]; /* THE BASIC VARIABLES VECTOR */
Discrete OT Problem

38

M
X
Y
U(a, b)

P
O(n^3 \log(n))

network flow solver used in practice.
Solution: Regularization

Wishlist:
- Faster & scalable
- More stable
- Robust
- Differentiable
Def. Regularized Wasserstein, $\gamma \geq 0$

$$W_\gamma(\mu, \nu) \overset{\text{def}}{=} \min_{P \in \mathcal{U}(a,b)} \langle P, M_{XY} \rangle - \gamma E(P)$$

$$E(P) \overset{\text{def}}{=} -\sum_{i,j=1}^{nm} P_{ij} (\log P_{ij} - 1)$$

Note: Unique optimal solution because of strong concavity of entropy
Def. Regularized Wasserstein, $\gamma \geq 0$

$$W_\gamma(\mu, \nu) \overset{\text{def}}{=} \min_{P \in \mathcal{U}(a,b)} \left\langle P, M_{XY} \right\rangle - \gamma E(P)$$

Note: **Unique** optimal solution because of strong concavity of entropy
Entropic Regularization [Wilson’62]

**Def.** Regularized Wasserstein, $\gamma \geq 0$

$$W_\gamma(\mu, \nu) \overset{\text{def}}{=} \min_{P \in \mathcal{U}(a,b)} \langle P, M_{XY} \rangle - \gamma E(P)$$

Note: Unique optimal solution because of strong concavity of entropy
Entropic Regularization [Wilson’62]

Def. Regularized Wasserstein, \( \gamma \geq 0 \)

\[
W_\gamma(\mu, \nu) \overset{\text{def}}{=} \min_{P \in U(a, b)} \langle P, M_{XY} \rangle - \gamma E(P)
\]

“static” problem associated with Schrödinger problem
Prop. If $P_\gamma \overset{\text{def}}{=} \text{argmin} \left\langle P, M_{XY} \right\rangle - \gamma E(P)$

$P \in U(a,b)$

then $\exists! u \in \mathbb{R}^n_+, v \in \mathbb{R}^m_+$, such that

$$P_\gamma = \text{diag}(u) K \text{diag}(v), \quad K \overset{\text{def}}{=} e^{-M_{XY}/\gamma}$$
Prop. If \( P_{\gamma} \overset{\text{def}}{=} \arg\min_P \langle P, M_{XY} \rangle - \gamma E(P) \)

\[ P \in U(a, b) \]

then \( \exists! u \in \mathbb{R}_+^n, v \in \mathbb{R}_+^m \), such that

\[
P_{\gamma} = \text{diag}(u) K \text{diag}(v), \quad K \overset{\text{def}}{=} e^{-M_{XY}/\gamma}
\]

\[
L(P, \alpha, \beta) = \sum_{ij} P_{ij} M_{ij} + \gamma P_{ij} (\log P_{ij} - 1) + \alpha^T (P 1 - a) + \beta^T (P^T 1 - b)
\]

\[
\frac{\partial L}{\partial P_{ij}} = M_{ij} + \gamma \log P_{ij} + \alpha_i + \beta_j
\]

\[
\text{if } \frac{\partial L}{\partial P_{ij}} = 0 \quad \Rightarrow P_{ij} = e^{\frac{\alpha_i}{\gamma} - \frac{M_{ij}}{\gamma} + \frac{\beta_j}{\gamma}} = u_i \ K_{ij} \ v_j
\]
Prop. If $P_{\gamma} \overset{\text{def}}{=} \arg \min P \in U(a, b) \langle P, M_{XY} \rangle - \gamma E(P)$

then $\exists! u \in \mathbb{R}_+^n, v \in \mathbb{R}_+^m$, such that

$$P_{\gamma} = \text{diag}(u)K\text{diag}(v), \quad K \overset{\text{def}}{=} e^{-M_{XY}/\gamma}$$

$$P_{\gamma} \in U(a, b) \iff \begin{cases} 
\text{diag}(u)K\text{diag}(v)1_m = a \\
\text{diag}(v)K^T\text{diag}(u)1_n = b 
\end{cases}$$
Prop. If \( P_\gamma \overset{\text{def}}{=} \arg\min_{P \in U(a, b)} \left\langle P, M_{XY} \right\rangle - \gamma E(P) \) then \( \exists! \mathbf{u} \in \mathbb{R}^n_+, \mathbf{v} \in \mathbb{R}^m_+ \), such that

\[
P_\gamma = \text{diag}(\mathbf{u}) K \text{diag}(\mathbf{v}), \quad K \overset{\text{def}}{=} e^{-\frac{M_{XY}}{\gamma}}
\]
Prop. If \( P_\gamma \overset{\text{def}}{=} \arg\min_{P \in \mathcal{U}(a,b)} \langle P, M_{XY} \rangle - \gamma E(P) \) then \( \exists! u \in \mathbb{R}_+^n, v \in \mathbb{R}_+^m \), such that

\[
P_\gamma = \operatorname{diag}(u) K \operatorname{diag}(v), \quad K \overset{\text{def}}{=} e^{-M_{XY}/\gamma}
\]

\[
P_\gamma \in \mathcal{U}(a,b) \iff \begin{cases} \operatorname{diag}(u) K \operatorname{diag}(v) 1_m & = a \\ \operatorname{diag}(v) K^T \operatorname{diag}(u) 1_n & = b \end{cases}
\]
Prop. If $P_\gamma \overset{\text{def}}{=} \arg\min_{P \in U(a,b)} \langle P, M_{XY} \rangle - \gamma E(P)$

then $\exists! \mathbf{u} \in \mathbb{R}_+^n, \mathbf{v} \in \mathbb{R}_+^m$, such that

$$P_\gamma = \text{diag}(\mathbf{u}) K \text{diag}(\mathbf{v}), \quad K \overset{\text{def}}{=} e^{-M_{XY}/\gamma}$$

$$P_\gamma \in U(a,b) \iff \begin{cases} \text{diag}(\mathbf{u}) K \mathbf{v} = a \\ \text{diag}(\mathbf{v}) K^T \mathbf{u} = b \end{cases}$$
Prop. If $P_\gamma \overset{\text{def}}{=} \arg\min_{P \in U(a,b)} \langle P, M_{XY} \rangle - \gamma E(P)$ then $\exists! u \in \mathbb{R}_+^n, v \in \mathbb{R}_+^m$, such that

$$P_\gamma = \text{diag}(u)K\text{diag}(v), \quad K \overset{\text{def}}{=} e^{-M_{XY}/\gamma}$$
Prop. If \( P_\gamma \overset{\text{def}}{=} \arg \min_{P \in U(a,b)} \langle P, M_{XY} \rangle - \gamma E(P) \)

then \( \exists! \mathbf{u} \in \mathbb{R}_+^n, \mathbf{v} \in \mathbb{R}_+^m \) such that

\[
P_\gamma = \text{diag}(\mathbf{u}) K \text{diag}(\mathbf{v}), \quad K \overset{\text{def}}{=} e^{-M_{XY} / \gamma}
\]

\[
P_\gamma \in U(a,b) \iff \left\{
\begin{array}{l}
\mathbf{u} = a / K \mathbf{v} \\
\mathbf{v} = b / K^T \mathbf{u}
\end{array}
\right.
\]
Sinkhorn’s Algorithm : Repeat

1. \( u = a/Kv \)
2. \( v = b/K^Tu \)
Sinkhorn’s Algorithm : Repeat

1. \( u = a/Kv \)
2. \( v = b/K^T u \)

- [Sinkhorn’64] proved convergence for the first time.
- [Lorenz’89] linear convergence, see [Altschuler’17]
- \( O(nm) \) complexity, GPGPU parallel [Cuturi’13].
- \( O(n \log n) \) on gridded spaces using convolutions. [Solomon’15]
Sinkhorn in between $W$ and $\text{MMD}$

\[ \mu = \sum_{i=1}^{n} a_i \delta_{x_i} \quad \nu = \sum_{j=1}^{m} b_j \delta_{y_j} \]

\[ W^p(\mu, \nu) = \langle P^*, M_{XY} \rangle \]
Sinkhorn in between $W$ and $MMD$

\[ \mu = \sum_{i=1}^{n} a_i \delta_{x_i} \quad \nu = \sum_{j=1}^{m} b_j \delta_{y_j} \]

\[ W_\gamma(\mu, \nu) = \langle P_\gamma, M_{XY} \rangle \]

\[ W^p(\mu, \nu) = \langle P^*, M_{XY} \rangle \]
Sinkhorn in between $W$ and $MMD$

\[
\mu = \sum_{i=1}^{n} a_i \delta_{x_i} \quad \nu = \sum_{j=1}^{m} b_j \delta_{y_j}
\]

\[
\mathcal{E}(\mu, \nu) = \langle ab^T, M_{XY} \rangle
\]

\[
W_\gamma(\mu, \nu) = \langle P_\gamma, M_{XY} \rangle
\]

\[
W^p(\mu, \nu) = \langle P^*, M_{XY} \rangle
\]
Sinkhorn in between $W$ and MMD

\[ \mu = \sum_{i=1}^{n} a_i \delta_{x_i} \quad \nu = \sum_{j=1}^{m} b_j \delta_{y_j} \]

\[ \mathcal{E}(\mu, \nu) = \langle ab^T, MXY \rangle \]

\[ W_\gamma(\mu, \nu) = \langle P_\gamma, MXY \rangle \]

\[ W^p(\mu, \nu) = \langle P^*, MXY \rangle \]
Sinkhorn in between $W$ and $\text{MMD}$

$$\mathcal{E}(\mu, \nu) = \langle ab^T, M_{XY} \rangle$$

$$\text{MMD}(\mu, \nu) = \mathcal{E}(\mu, \nu) - \frac{1}{2}(\mathcal{E}(\mu, \mu) + \mathcal{E}(\nu, \nu))$$

$$W_\gamma(\mu, \nu) = \langle P_\gamma, M_{XY} \rangle$$

$$\bar{W}_\gamma(\mu, \nu) = W_\gamma(\mu, \nu) - \frac{1}{2}(W_\gamma(\mu, \mu) + W_\gamma(\nu, \nu))$$

$$W^p(\mu, \nu) = \langle P^*, M_{XY} \rangle$$
Sinkhorn in between $W$ and $MMD$

\[
\text{MMD}(\mu, \nu) = \mathcal{E}(\mu, \nu) - \frac{1}{2} (\mathcal{E}(\mu, \mu) + \mathcal{E}(\nu, \nu))
\]

$\gamma \to \infty$

\[
\tilde{W}_\gamma(\mu, \nu) = W_\gamma(\mu, \nu) - \frac{1}{2} (W_\gamma(\mu, \mu) + W_\gamma(\nu, \nu))
\]

$\gamma \to 0$

\[
W^p(\mu, \nu) = \langle P^*, M_{XY} \rangle
\]
How to compare them?

i.i.d samples \( x_1, \ldots, x_n \sim \mu, y_1, \ldots, y_m \sim \nu, \)

\[
\hat{\mu}_n \overset{\text{def}}{=} \frac{1}{n} \sum_i \delta_{x_i}, \quad \hat{\nu}_m \overset{\text{def}}{=} \frac{1}{m} \sum_j \delta_{y_j}
\]

**Computational properties**

Effort to compute/approximate \( \Delta(\hat{\mu}_n, \hat{\nu}_m) \)?

**Statistical properties**

\[
|\Delta(\mu, \nu) - \Delta(\hat{\mu}_n, \hat{\nu}_n)| \leq f(n)?
\]
Sinkhorn in between $W$ and $MMD$

\[
MMD(\mu, \nu) = \mathcal{E}(\mu, \nu) - \frac{1}{2} (\mathcal{E}(\mu, \mu) + \mathcal{E}(\nu, \nu))
\]

\[
(n + m)^2
\]

\[
O(1/\sqrt{n})
\]

\[
W^p(\mu, \nu) = \langle P^*, M_{XY} \rangle
\]

\[
O((n + m)nm \log(n + m))
\]

\[
O(1/n^{1/d})
\]
Sinkhorn in between $W$ and $MMD$

$\text{MMD}(\mu, \nu) = \mathcal{E}(\mu, \nu) - \frac{1}{2}(\mathcal{E}(\mu, \mu) + \mathcal{E}(\nu, \nu))$

$(n + m)^2$ \hspace{5cm} $O(1/\sqrt{n})$

$\bar{W}_\gamma(\mu, \nu) = W_\gamma(\mu, \nu) - \frac{1}{2}(W_\gamma(\mu, \mu) + W_\gamma(\nu, \nu))$

$O((n + m)^2)$ \hspace{2cm} $O\left(\frac{1}{\gamma^{d/2} \sqrt{n}}\right)$ \hspace{2cm} [GCBCP’18]

$W^p(\mu, \nu) = \langle P^*, M_{XY} \rangle$

$O((n + m)nm \log(n + m))$ \hspace{2cm} $O(1/n^{1/d})$ \hspace{2cm} [FSVATP’18]
Differentiability of $W$

$$W((a, X), (b, Y))$$

$$\mu = \sum_{i=1}^{n} a_i \delta_{x_i}$$

$$\nu = \sum_{j=1}^{m} b_j \delta_{y_j}$$
Differentiability of $W$

\[ W((a + \Delta a, X), (b, Y)) = W((a, X), (b, Y)) + ?? \]

\[
\mu = \sum_{i=1}^{n} a_i \delta_{x_i}
\]

\[
\nu = \sum_{j=1}^{m} b_j \delta_{y_j}
\]
Differentiability of $W$

$W((a + \Delta a, X), (b, Y)) = W((a, X), (b, Y)) + ?$

$\mu = \sum_{i=1}^{n} a_i \delta_{x_i}$

$\nu = \sum_{j=1}^{m} b_j \delta_{y_j}$

$a \leftarrow a + \Delta a$
Sinkhorn $\rightarrow$ Differentiability

\[ W((a, X + \Delta X), (b, Y)) = W((a, X), (b, Y)) + \Delta \]

\[ \mu = \sum_{i=1}^{n} a_i \delta_{x_i} \]

\[ \nu = \sum_{j=1}^{m} b_j \delta_{y_j} \]

\((\Omega, D)\)
Sinkhorn $\rightarrow$ Differentiability

\[ W((a, X + \Delta X), (b, Y)) = W((a, X), (b, Y)) + \text{??} \]

\[ \mu = \sum_{i=1}^{n} a_i \delta_{x_i} \]

\( (\Omega, D) \)

\[ X \leftarrow X + \Delta X \]

\[ \nu = \sum_{j=1}^{m} b_j \delta_{y_j} \]
Sinkhorn: A Programmer View

Def. For $L \geq 1$, define

$$W_L(\mu, \nu) \overset{\text{def}}{=} \langle P_L, M_{XY} \rangle,$$

where $P_L \overset{\text{def}}{=} \text{diag}(u_L) K \text{diag}(v_L)$,

$v_0 = 1_m; l \geq 0, u_l \overset{\text{def}}{=} a/K v_l, v_{l+1} \overset{\text{def}}{=} b/K^T u_l$.

Prop. $\frac{\partial W_L}{\partial X}$, $\frac{\partial W_L}{\partial a}$ can be computed recursively, in $O(L)$ kernel $K \times$ vector products.
Def. For $L \geq 1$, define

$$W_L(\mu, \nu) \overset{\text{def}}{=} \langle P_L, M_{XY} \rangle,$$
Def. For $L \geq 1$, define

$$W_L(\mu, \nu) \overset{\text{def}}{=} \langle P_L, M_{XY} \rangle,$$

Prop. $\frac{\partial W_L}{\partial X}, \frac{\partial W_L}{\partial a}$ can be computed recursively, in $O(L)$ kernel $K \times$ vector products.

[Hashimoto’16] [Bonneel’16][Shalit’16]
Primal Descent on Regularized $W$

\[
\min_{a \in \Sigma_{h \times h}} \sum_{i=1}^{N} \lambda_i W_\gamma(a, b_i)
\]

[Cuturi’14]
Primal Descent on Regularized $W$

$$\min_{a \in \Sigma_{h \times h}} \sum_{i=1}^{N} \lambda_i W_\gamma(a, b_i)$$

[Cuturi’14]
Primal Descent on Regularized $W$

$$\min_{a \in \Sigma_{h \times h}} \sum_{i=1}^{N} \lambda_i W_\gamma(a, b_i)$$

[Cuturi’14]
We describe first the two direct competitors of our method: the Wasserstein distances. The Wasserstein distance is defined as the minimization of the expected cost of transporting one distribution to another. However, this definition can be surprisingly ill-posed, as we see next.

Indeed, it is also known that the 2-Wasserstein mean of two univariate (continuous) Gaussian densities of variance 1 can be surprisingly ill-posed as we see next.

The Wasserstein mean is defined as the barycenter of two probability distributions. In the continuous case, the Wasserstein mean is the solution of a particular linear program. However, this linear program is ill-posed, and its solution may be extremely unstable.

Regularization techniques are used to stabilize the solution. For example, the Sinkhorn algorithm is a popular regularization technique. The Sinkhorn algorithm solves a related problem, where the cost matrix is non-negative and the marginal constraints are relaxed. The solution of this problem is close to the Wasserstein mean, but it is not exactly the same.

That plot is obtained by using smoothed spline interpolations of the means. That objective can then be evaluated by running the Sinkhorn fixed-point algorithm for a chosen tolerance. That precision can be measured by the deviation of the subroutine but noisy gradients and their gradient is equal to zero. That precision can be measured by the difference of the subroutine but noisy gradients and their gradient is equal to zero.

The objective value for that smoothed approximation in Figure 1.

The objective value for that smoothed approximation in Figure 2.

The bottom-right plot displays the solution of Equation (11), which is the exact optimal solution of Equation (12).

Given optimal couplings $\gamma^* \in \Gamma(p,q)$, the 2-Wasserstein barycenter is equal to the marginal common to all those couplings.

We compare them with our smooth dual approach to compute the Wasserstein barycenter of 12 variables and 300 constraints. We observe that the discretized barycenter can be better observed in the top-right (stair) plot, which gives us a better understanding of the barycenter.

Naturally, one would expect the barycenter of two univariate (continuous) Gaussian densities of variance 1 to be close, in some sense, to the discretized histograms laid out on the 100 grid, as displayed in the bottom-left plot.

The solution of Equation (12) of the two densities $\mathcal{N}(2, 1)$ and $\mathcal{N}(-2, 1/4)$, respectively denoted $p$ and $q$, is the exact optimal solution.

The objective value for that smoothed approximation in Figure 2.

Indeed, it is also known that the 2-Wasserstein mean of two univariate (continuous) Gaussian densities of variance 1 can be surprisingly ill-posed, as we see next.

The objective value for that smoothed approximation in Figure 2.

Indeed, it is also known that the 2-Wasserstein mean of two univariate (continuous) Gaussian densities of variance 1 can be surprisingly ill-posed, as we see next.
Experiments

In many applications a beneficial effect of smoothing the resulting solution, which may have a barycenter may be extremely unstable. Regularization of the argmin of a linear program, the true Wasserstein distance illustrates however that, because it is defined as the Euclidean norm between the row and column marginals zero, as shown in \((\ref{eq:wasserstein})\). This numerical experiment does not contradict the fact that the discretized barycenter can be surprisingly ill-posed as we see next.

Given optimal couplings \(X\) and \(\gamma\) where the discrete evaluations of these densities are displayed in the bottom-left plot, is the exact optimal solution of Equation \((\ref{eq:wasserstein})\). This fact is illustrated in the top-left plot of Figure 1.

Indeed, it is also known that the \(2\)-Wasserstein mean of a Gaussian has, indeed, a smaller objective value for that smoothed approximation obtained with the simplex has, indeed, a smaller objective value.

\[ W^2_2(p, q) = \sum x \gamma(x, y) \left( \frac{|x|}{\gamma(x, y)} \right)^2, \]

\[ W^2_1(p, q) = \sum x \gamma(x, y) \left( \sqrt{|x|} \right)^2, \]

\[ W^2_N(p, q) = \sum x \gamma(x, y) \left( \frac{|x|^2}{\gamma(x, y)} \right)^2. \]

That \(WBP\) reduces to a linear program of that precision can be measured by the difference in the old and new values of the objective function.

That objective can be evaluated by running the Sinkhorn fixed-point algorithm must be chosen. That precision can be measured by the difference in the old and new values of the objective function.

\[ \left( \frac{1}{K} \sum_{k=1}^{K} \log \left( \gamma(x, y) \right) \right)_{\text{new}} - \left( \frac{1}{K} \sum_{k=1}^{K} \log \left( \gamma(x, y) \right) \right)_{\text{old}} \]

The objective value for that smoothed approximation that has been re-scaled to have a median value of 1).
On Regularizing or Not

Figure 6: Barycenters in Wasserstein space over $\mathbb{R}^n$ for different values of $n$. The same barycenter is shown for different grid sizes and different numbers of marginal distributions. As reported in [Schmitzer’16], the discretization artifacts are more pronounced for larger grid sizes and when the number of marginal distributions is increased. Therefore, only the deviation from this constraint is of interest. Therefore, only the deviation from this constraint is of interest. Therefore, only the deviation from this constraint is of interest.

[Schmitzer’16]
to each kernel separately and a multi-scale coarse-to-fine approach can be used, as outlined in

before, we introduce scaling factors

Note that relative to the dualization (prox and proxdiv steps of

The terms

independently and one gets a standard Sinkhorn update for every

Left

Figure 6: Barycenters in Wasserstein space over

Center

Left


multi-marginal

\[ \epsilon = 0.1 \, h^2 \]

\[ \epsilon = 2 \, h^2 \]

multi-marginal

[Schmitzer’16]
Dictionary Learning

\[
\min_{A \in (\Sigma_n)^K, \Lambda \in (\Sigma_K)^N} \sum_{i=1}^{N} W \left( b_i, \sum_{k=1}^{K} \Lambda_{i k} a_k \right)
\]

Data samples

[Sandler’11] [Zen’14] [Rolet’16]
Dictionary Learning

\[
\min_{A \in (\Sigma_n)^K, \Lambda \in (\Sigma_K)^N} \sum_{i=1}^N W \left( b_i, \sum_{k=1}^K \Lambda_k^i a_k \right)
\]

Wasserstein NMF

KL NMF

[Sandler’11] [Zen’14] [Rolet’16]
OT Dictionary Learning

- [Hoffman’98] proposed to learn dictionaries (topics) for text, seen as histograms-of-words.

\[ \Omega = \{ \text{words} \}, \quad |\Omega| \approx 13,000 \]

- Vector embeddings for words [Mikolov’13] [Pennington’14] defines geometry:

\[ D(\text{public, car}) = \|x_{\text{public}} - x_{\text{car}}\|^2 \]

- Data: 7,034 Reuters, 737 BBC sports news articles
Topic Models

[Rolet’16]
Inverse Wasserstein Problems

• consider Barycenter operator:

\[ b(\lambda) \overset{\text{def}}{=} \arg\min_a \sum_{i=1}^{N} \lambda_i W_\gamma(a, b_i) \]

• address now Wasserstein inverse problems:

Given \(a\), find \(\arg\min_{\lambda} \mathcal{E}(\lambda) \overset{\text{def}}{=} \text{Loss}(a, b(\lambda))\)

\( \lambda \in \Sigma_N \)
Wasserstein Inverse Problems

Euclidean Simplex: \( \left\{ \sum_{i=1}^{3} \lambda_i p_i, \lambda \in \Sigma_3 \right\} \)

Wasserstein simplex: \( \{ P(\lambda), \lambda \in \Sigma_3 \} \)
Barycenters = Fixed Points

Prop. [BCCNP’15] Consider $\mathbf{B} \in \Sigma_d^N$ and let $\mathbf{U}_0 = 1_{d \times N}$, and then for $l \geq 0$:

$$b^l \overset{\text{def}}{=} \exp \left( \log \left( K^T \mathbf{U}_l \right) \lambda \right);$$

$$\begin{cases} V_{l+1} \overset{\text{def}}{=} \frac{b^l 1_N^T}{K^T \mathbf{U}_l}, \\ U_{l+1} \overset{\text{def}}{=} \frac{\mathbf{B}}{K V_{l+1}}. \end{cases}$$
Using Truncated Barycenters

• instead of using the exact barycenter

\[ \arg\min_{\lambda \in \Sigma_N} \mathcal{E}(\lambda) \overset{\text{def}}{=} \text{Loss}(a, b(\lambda)) \]

• use instead the L-iterate barycenter

\[ \arg\min_{\lambda \in \Sigma_N} \mathcal{E}^{(L)}(\lambda) \overset{\text{def}}{=} \text{Loss}(a, b^{(L)}(\lambda)) \]

• Differentiate using the chain rule.

\[ \nabla \mathcal{E}^{(L)}(\lambda) = \left[ \partial b^{(L)} \right]^T (g), \quad g \overset{\text{def}}{=} \nabla \text{Loss}(a, \cdot)|_{b^{(L)}(\lambda)} \]
Gradient / Barycenter Computation

function SINKHORN-DIFFERENTIATE((p_s)_{s=1}^S, q, \lambda)

\forall s, b^{(0)}_s \leftarrow 1
(w, r) \leftarrow (0^S, 0^{S \times N})

for \ell = 1, 2, \ldots, L // Sinkhorn loop
\forall s, \varphi^{(\ell)}_s \leftarrow K^\top \frac{p_s}{Kb^{(\ell-1)}_s}

p \leftarrow \prod_s \left( \varphi^{(\ell)}_s \right)^{\lambda_s}
\forall s, b^{(\ell)}_s \leftarrow \frac{p}{\varphi^{(\ell)}_s}

g \leftarrow \nabla \mathcal{L}(p, q) \odot p

for \ell = L, L - 1, \ldots, 1 // Reverse loop
\forall s, w_s \leftarrow w_s + \langle \log \varphi^{(\ell)}_s, g \rangle
\forall s, r_s \leftarrow -K^\top \left( K \left( \frac{\lambda_s g - r_s}{\varphi^{(\ell)}_s} \right) \odot \frac{p_s}{(Kb^{(\ell-1)}_s)_{2}} \right) \odot b^{(\ell-1)}_s

g \leftarrow \sum_s r_s

return P^{(L)}(\lambda) \leftarrow p, \nabla \mathcal{E}_L(\lambda) \leftarrow w
Application: Volume Reconstruction

Shape database $(p_1, \ldots, p_5)$

Input shape $q$

Projection $P(\lambda)$

Iso-surface

[Bonneel’16]
Application: Color Grading
Application: Color Grading

\[ \lambda_0 = 0.03 \]
\[ \lambda_1 = 0.12 \]
\[ \lambda_2 = 0.40 \]
\[ \lambda_3 = 0.43 \]
Application: Color Grading
Application: Color Grading

Wasserstein Barycentric Coordinates: Histogram Regression using Optimal Transport, SIGGRAPH'16 [BPC'16]
The Euclidean barycentric coordinates consist of 8 non-zero values,
while the Wasserstein barycentric coordinates have 9.
Although the Euclidean barycenter looks sharper, close inspection
reveals overlapping boundaries and edges (see insets) while our
projections share 7 of these coefficients, with comparable weights.
These two support, with 9 and 8 non-zero weights respectively. These two
loss. The coefficients selected by these two procedures have a sparse
histogram projections, using the Euclidean simplex (middle) and
while the Wasserstein simplex of the
Application: Brain Mapping

Figure 11: The test MRI is projected on both simplexes using a
projection

Figure 12: This tool is illustrated with applications to color manipulation,
Future work.
The Euclidean barycentric coordinates consist in 8 non-zero values, although the Euclidean barycenter looks sharper, close inspection reveals overlapping boundaries and edges (see insets) while our projections share 7 of these coefficients, with comparable weights.

This paper introduces the concept of Wasserstein barycentric coordinates, which are used to approximate reflectance and shape inference.

Future work.

Figure 8: Using the image search engine Flickr, we use the top 10 results for the query 'autumn' (here, with Commercial use allowed and Interesting) and use them to color grade a summer image. (First row) For different loss functions, we show the non-zero barycentric coordinates. We illustrate this tool with applications to color manipulation, histogram projections, using the Euclidean simplex (middle) and Wasserstein simplex (right), both computed using an algorithm.

Figure 11. The test MRI is projected on both simplexes using a projection algorithm.

We observed that our Wasserstein barycentric coordinates are often very sparse. This sparsity might be attributed to the fact that the input histogram will unlikely be faithfully approximated by Wasserstein barycenters. For applications such as shape inference, the coefficient selection is a challenging problem due to the sparsity of the coordinates. Our proposed method can be carried out using coarser gradient approximations, without impacting convergence.

Memory requirements are limited by its memory requirements, and remains slow for databases exceeding more than 10-20 dense histograms. Memory requirements increase linearly with the number of iterations exceeding more than 10-20 dense histograms. Memory requirements increase linearly with the number of iterations.

When the histogram database is far from the input histogram, the number of iterations required to converge is significantly higher.

Performance.

When using a single target histogram, this can lead to large color distortion. The coefficients selected by these two procedures have a sparse structure, which can be exploited for efficient computation. Our method is competitive with existing techniques that use a single target histogram, as illustrated in the figure.

The method requires less memory than existing techniques, while achieving comparable performance. The method is also faster, as shown in the table.

Table 1: Comparison of memory requirements and computation time for different algorithms. Our method requires less memory and is faster than existing techniques.


color matching of Pitie et al. [2007] to transfer colors from the most contributing photographs (numbered 0, 2, 4, 6 and 8). As existing techniques use a single target histogram, this can lead to large color distortion. Our method is competitive with existing techniques, as shown in the table.

Table 2: Comparison of memory requirements and computation time for different algorithms. Our method requires less memory and is faster than existing techniques.
end-to-end \( W \) Dictionary Learning

\[
\min_{A \in (\Sigma_n)^K} \lambda \in (\Sigma_K)^N \sum_{i=1}^{N} \mathcal{L} (b_i, a(\lambda_i))
\]

[Schmitz’18]
end-to-end \( W \) Dictionary Learning

\[
\min_{A \in (\Sigma_n)^K, \Lambda \in (\Sigma_K)^N} \sum_{i=1}^{N} \mathcal{L}(b_i, a(\lambda_i))
\]

[Schmitz’18]
Minimum Kantorovich Estimators

\[
\min_{\theta \in \Theta} W(\nu_{\text{data}}, f_{\theta} \# \mu)
\]

[Bassetti’06] 1st reference discussing this approach.

Challenge: \( \nabla_{\theta} W(\nu_{\text{data}}, f_{\theta} \# \mu) \)?

[Montavon’16] use regularized OT in a finite setting.

[Arjovsky’17] (WGAN) uses a NN to approximate dual solutions and recover gradient w.r.t. parameter

[Bernton’17] (Wasserstein ABC)

[Genevay’17, Salimans’17] (Sinkhorn approach)
Proposal: Autodiff OT using Sinkhorn

Approximate $W$ loss by the transport cost $\bar{W}_L$ after $L$ Sinkhorn iterations.
Example: MNIST, Learning $f_{\theta}$
Example: MNIST, Learning $f_\theta$

Latent space $[0, 1]^2$
Example: Generation of Images

Driven by applications in computer graphics, neuroscience, and AI, there has been a surge of interest in generative models. The most prominent and successful one is Generative Adversarial Networks (GANs) proposed by Goodfellow et al. [2014].

[Salimans’18]
Example: Generation of Images

[Salimans’18] arxiv.org/pdf/1710.05488
Concluding Remarks

• **Regularized OT** is much faster than OT when handled in full generality.

• **Regularized OT** can interpolate between $W$ and the MMD / Energy distance (MMD) metrics.

• The solution of *regularized OT* is “*auto-differentiable*”.

• Many open problems remain in ML that can be addressed with OT.