The Bures-Wasserstein Geometry for Machine Learning

Séminaire Palaisien

December 1st, 2020

Boris Muzellec

Joint work with Marco Cuturi
(Google Brain & CREST, ENSAE)

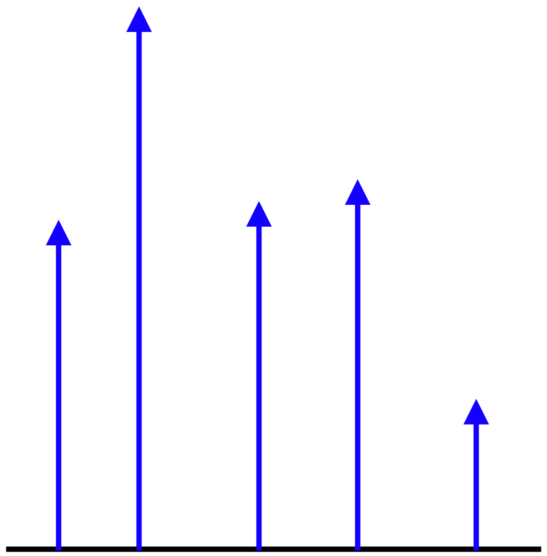
Comparing Distributions

In ML, data sciences, stats… we often need to compare distributions.

Comparing Distributions

In ML, data sciences, stats… we often need to compare distributions.

E.g. we observe data…

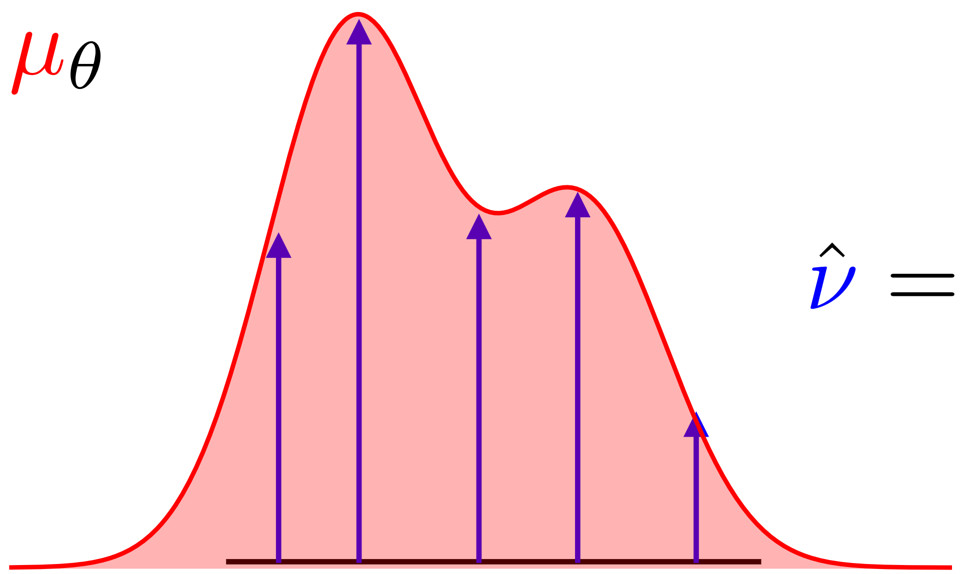


$$\hat{\nu} = \sum_{i=1}^{n} a_i \delta_{x_i}$$

Comparing Distributions

In ML, data sciences, stats… we often need to compare distributions.

E.g. we observe data…



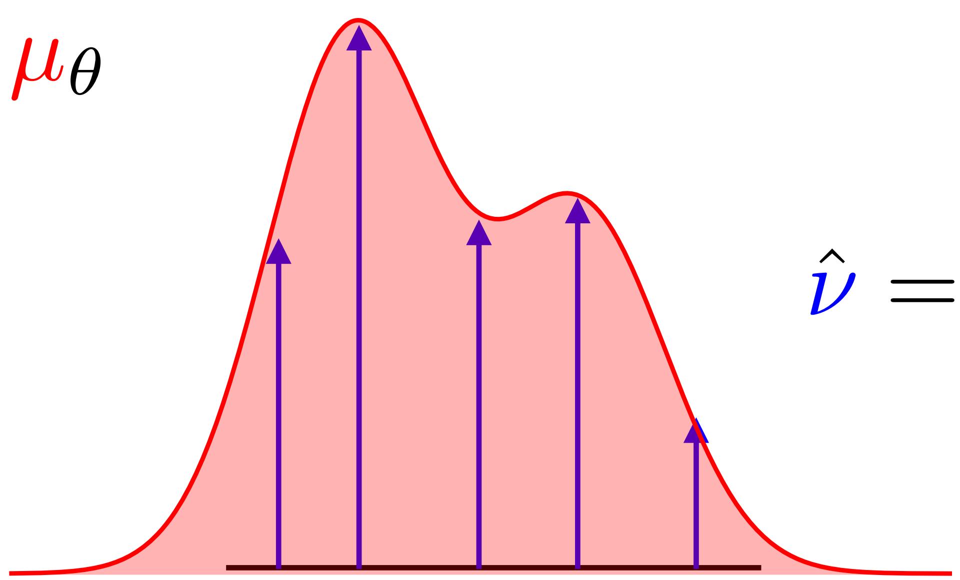
$$\hat{\nu} = \sum_{i=1}^{n} a_i \delta_{x_i}$$

… and we want to fit a model.

Comparing Distributions

In ML, data sciences, stats… we often need to compare distributions.

E.g. we observe data…



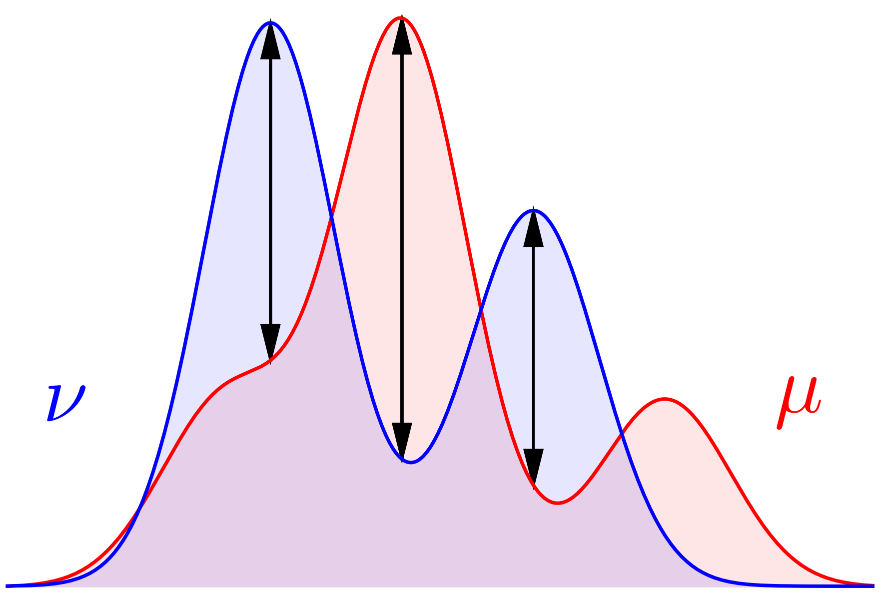
$$\hat{\nu} = \sum_{i=1}^{n} a_i \delta_{x_i}$$

… and we want to fit a model.

How to compare them?

Comparing Distributions

1. "Vertically":



- Look at pointwise differences between densities

$$|{\color{red}p(x)} - {\color{blue}q(x)}| \quad \text{or} \quad \frac{{\color{red}p(x)}}{{\color{blue}q(x)}}$$

- Turn them into a divergence. Examples:

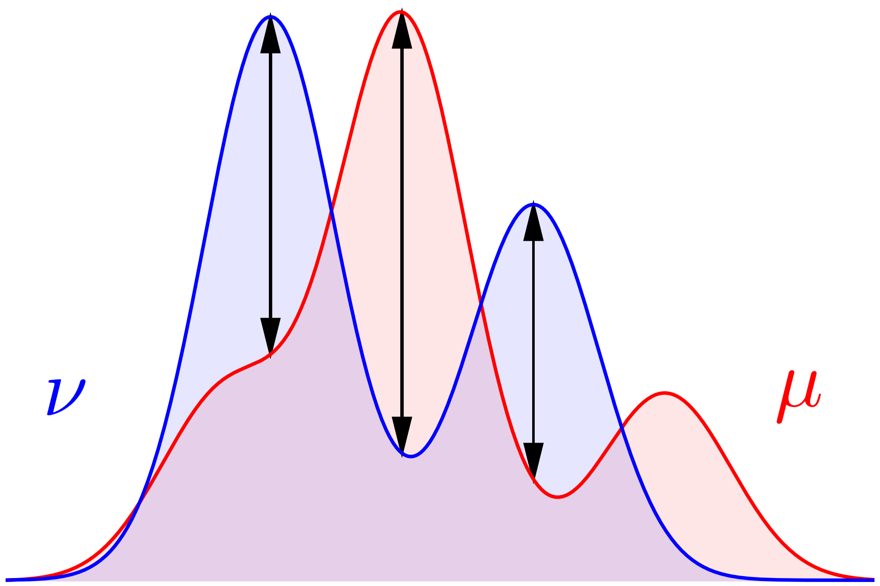
$$\mathrm{TV}({\color{red}\mu}, {\color{blue}\nu}) = \sup_{A \in \mathcal{B}} \left| \int \mathbb{1}_A(x) {\color{red}p(x)} dx - \int \mathbb{1}_A(x) {\color{blue}q(x)} dx \right| \qquad \text{(Total Variation)}$$

$$D_{\mathrm{KL}}({\color{red}\mu}, {\color{blue}\nu}) = \int \log \frac{{\color{red}p(x)}}{{\color{blue}q(x)}} {\color{red}p(x)} dx \qquad \text{(Kullback-Leibler)}$$

$$D_f({\color{red}\mu}, {\color{blue}\nu}) = \int f\left(\frac{{\color{red}p(x)}}{{\color{blue}q(x)}} \right) {\color{blue}q(x)} dx \qquad \text{(f-divergences)}$$

Comparing Distributions

1. "Vertically":



- Look at pointwise differences between densities

$$|p(x) - q(x)| \quad \text{or} \quad \frac{p(x)}{q(x)}$$

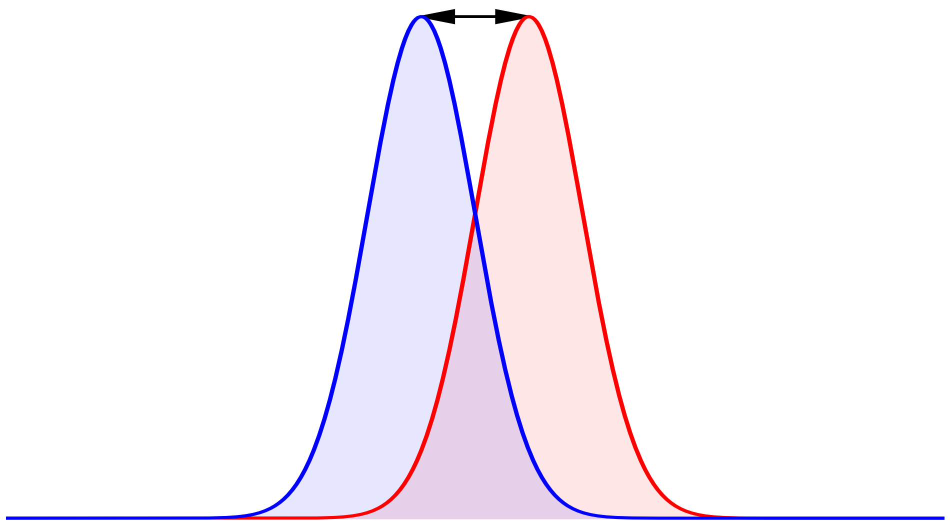
- Turn them into a divergence. Examples:

$$\mathrm{TV}(\mu, \nu) = \sup_{A \in \mathcal{B}} \left| \int \mathbb{1}_A(x) p(x) dx - \int \mathbb{1}_A(x) q(x) dx \right| \qquad \text{(Total Variation)}$$

$$D_{\mathrm{KL}}(\mu, \nu) = \int \log \frac{p(x)}{q(x)} p(x) dx \qquad \text{(Kullback-Leibler)}$$

$$D_f(\mu, \nu) = \int f\left(\frac{p(x)}{q(x)}\right) q(x) dx \qquad \text{(f-divergences)}$$

2. "Horizontally"?

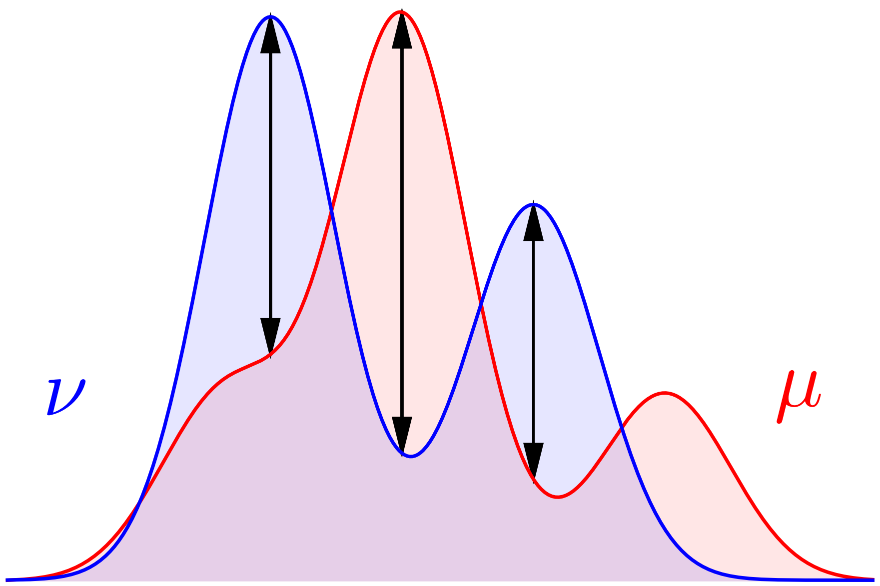


Comparing Distributions

1. "Vertically":

- Look at pointwise differences between densities

$$|p(x) - q(x)| \quad \text{or} \quad \frac{p(x)}{q(x)}$$



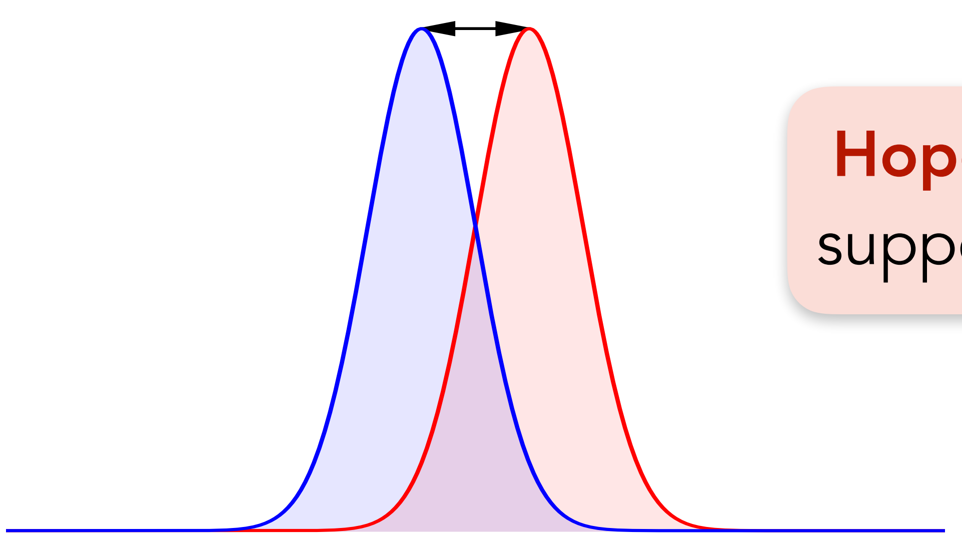
- Turn them into a divergence. Examples:

$$\mathrm{TV}(\mu, \nu) = \sup_{A \in \mathcal{B}} \left| \int \mathbb{1}_A(x) p(x) dx - \int \mathbb{1}_A(x) q(x) dx \right| \qquad \text{(Total Variation)}$$

$$D_{\mathrm{KL}}(\mu, \nu) = \int \log \frac{p(x)}{q(x)} p(x) dx \qquad \text{(Kullback-Leibler)}$$

$$D_f(\mu, \nu) = \int f\left(\frac{p(x)}{q(x)}\right) q(x) dx \qquad \text{(f-divergences)}$$

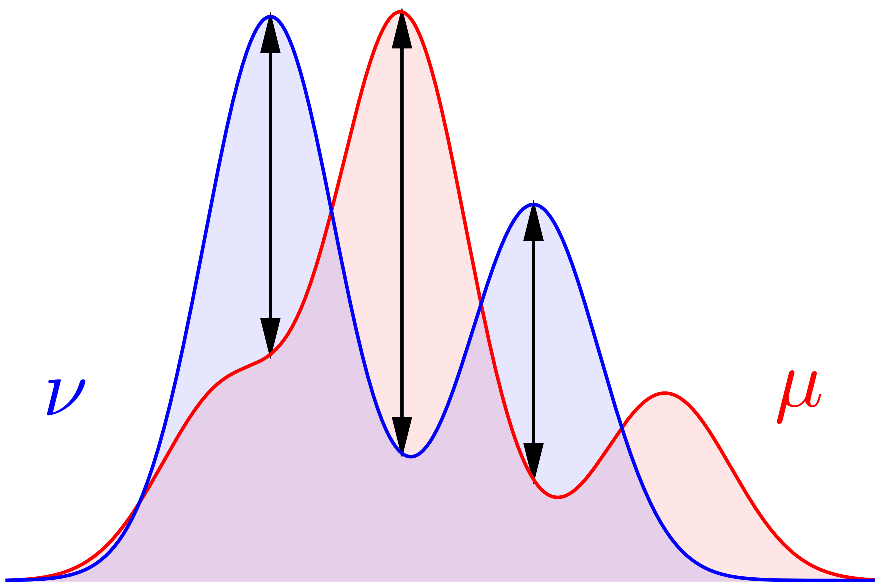
2. "Horizontally"?



Hope: better behaved when the supports of μ and ν are disjoint?

Comparing Distributions

1. "Vertically":



- Look at pointwise differences between densities

$$|p(x) - q(x)| \quad \text{or} \quad \frac{p(x)}{q(x)}$$

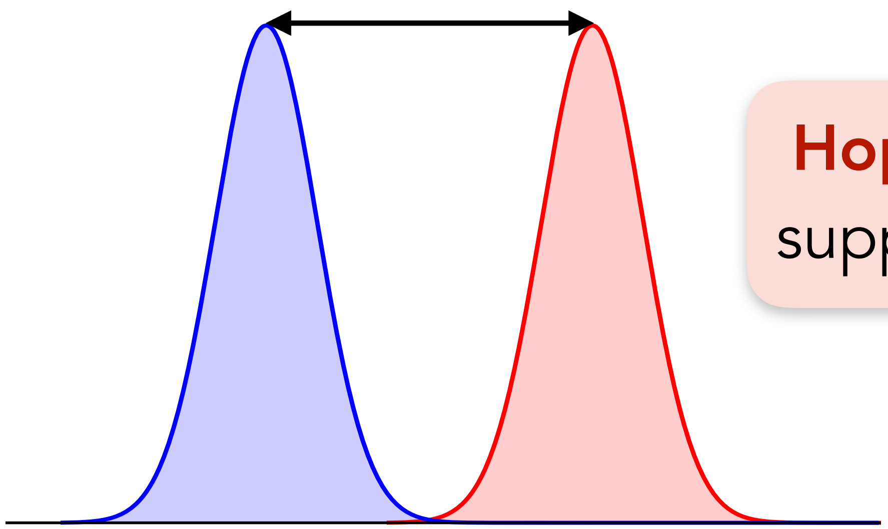
- Turn them into a divergence. Examples:

$$\mathrm{TV}(\mu, \nu) = \sup_{A \in \mathcal{B}} \left| \int \mathbb{1}_A(x) p(x) dx - \int \mathbb{1}_A(x) q(x) dx \right| \qquad \text{(Total Variation)}$$

$$D_{\mathrm{KL}}(\mu, \nu) = \int \log \frac{p(x)}{q(x)} p(x) dx \qquad \text{(Kullback-Leibler)}$$

$$D_f(\mu, \nu) = \int f\left(\frac{p(x)}{q(x)}\right) q(x) dx \qquad \text{(f-divergences)}$$

2. "Horizontally"?



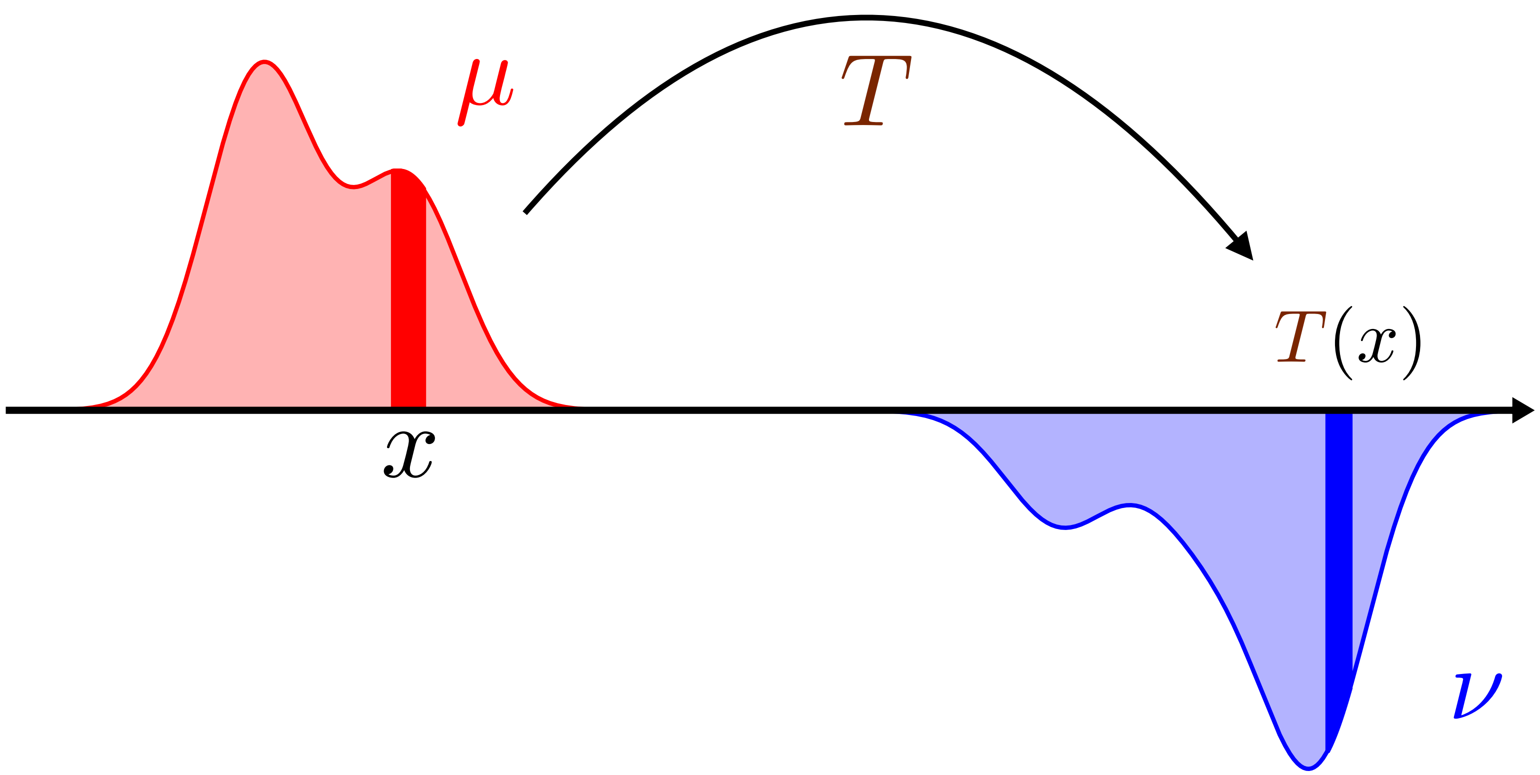
Hope: better behaved when the supports of μ and ν are disjoint?

Monge's Optimal Transport Problem

Def. Wasserstein Distance

How to move earth with minimal effort w.r.t. cost function $c(x,y) = \|x-y\|^2$?

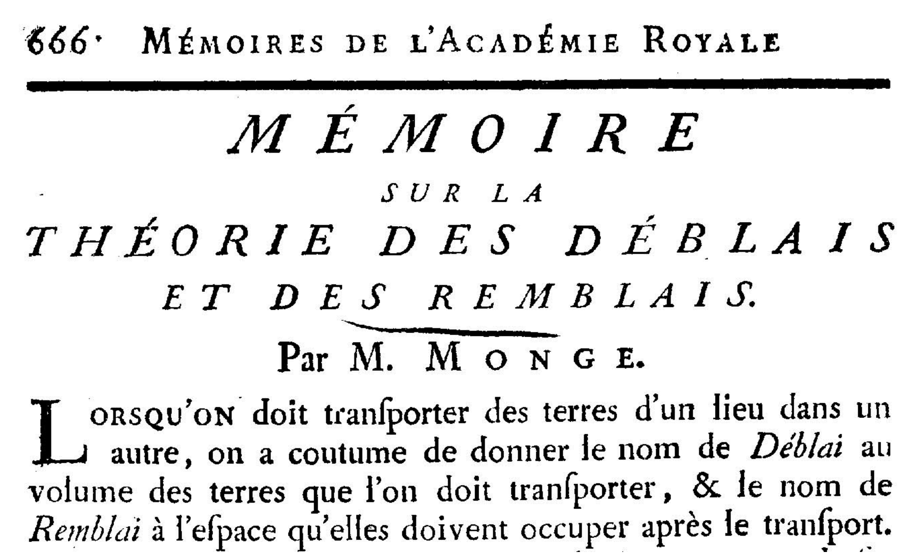
$$W_2^2(\mu,\nu) \stackrel{\text{def}}{=} \inf_{T:\mathbb{R}^d\to\mathbb{R}^d} \int_{\mathbb{R}^d} \|x - T(x)\|^2 \mathrm{d}\mu(x) \quad \text{s.t.} \quad \underbrace{\forall A \subset \mathbb{R}^d, \nu(A) = \mu(T^{-1}(A))}$$



$$X \sim \mu \implies T(X) \sim \nu$$

" T pushes forward μ to ν "

$$T_\sharp \mu = \nu$$



666 MÉMOIRES DE L'ACADÉMIE ROYALE

MÉMOIRE

SUR LA

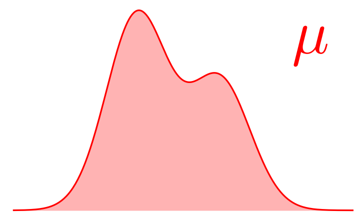
THÉORIE DES DÉBLAIS

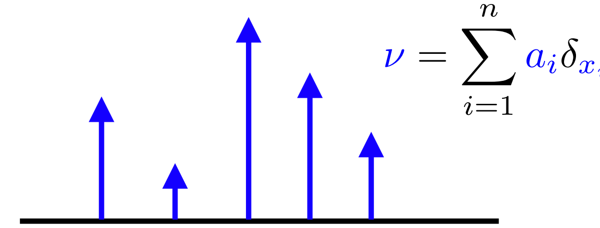
ET DES REMBLAIS.

Par M. MONGE.

Lorsqu'on doit tranſporter des terres d'un lieu dans un autre, on a coutume de donner le nom de *Déblai* au volume des terres que l'on doit tranſporter, & le nom de *Remblai* à l'eſpace qu'elles doivent occuper après le tranſport.

Computing OT in Practice

Continuous 

or discrete 

?

Discrete-discrete: LP in $O(n^3 \log n)$, regularized approaches[3] in $O(n^2)$.

Discrete-continuous: density approx on grid[4], stochastic approx[5].

Continuous-continuous: In general, difficult.

- In low dimension, if $c = \|\cdot\|^2$:
  - Benamou-Brenier's[6] dynamic formulation (variational problem),
  - Equivalent to Monge-Ampère PDE (by Brenier's theorem[7]),
- In high dimension:
  - NN parameterization of potentials[8] or maps[9] (very active in ML),
  - Closed forms:
    - Project to low dimension: Sliced Wasserstein[10][11],
    - $\boxed{\text{Gaussians[12][13][14], Elliptical distributions[15].}}$

[3] Cuturi 2013; [4] Mérigot 2011; [5] Genevay, Cuturi, et al. 2016; [6] Benamou et al. 2000; [7] Brenier 1987; [8] Arjovsky et al. 2017; [9] Makkuva et al. 2020; [10] Rabin et al. 2011; [11] Bonneel et al. 2015; [12] Dowson et al. 1982; [13] Olkin et al. 1982; [14] Takatsu 2011; [15] Gelbrich 1990.

OT with Gaussians

The Bures-Wasserstein Distance

Prop. Wasserstein Distance between Gaussians.

Dowson et al. 1982
Olkin et al. 1982
Givens, 1984

Let $\alpha = \mathcal{N}(\mathbf{a}, \mathbf{A})$ and $\beta = \mathcal{N}(\mathbf{b}, \mathbf{B})$. Then,

$$W_2^2(\alpha, \beta) = \|\mathbf{a} - \mathbf{b}\|^2 + \mathfrak{B}^2(\mathbf{A}, \mathbf{B})$$

Def. Bures Distance for PSD matrices.

Defines a Riemannian metric.

$\forall \mathbf{A}, \mathbf{B} \in S_+^d,$

$$\mathfrak{B}^2(\mathbf{A}, \mathbf{B}) \stackrel{\text{def}}{=} \mathrm{Tr}\mathbf{A} + \mathrm{Tr}\mathbf{B} - 2\mathrm{Tr}(\mathbf{A}^{1/2}\mathbf{B}\mathbf{A}^{1/2})^{1/2}$$

Remarks/examples:

- If $\mathbf{A}$ and $\mathbf{B}$ commute, $\mathfrak{B}(\mathbf{A}, \mathbf{B}) = \|\mathbf{A}^{1/2} - \mathbf{B}^{1/2}\|_F$ (Hellinger distance)
- If $\mathbf{A}, \mathbf{B} \to 0,\ W_2(\alpha, \beta) \to \|\mathbf{a} - \mathbf{b}\| = W_2(\delta_{\mathbf{a}}, \delta_{\mathbf{b}})$
- $\mathfrak{B}(\mathbf{A}, \mathbf{B})$ remains defined even if $\mathrm{rk}\mathbf{A} < d$ (or $\mathrm{rk}\mathbf{B} < d$)

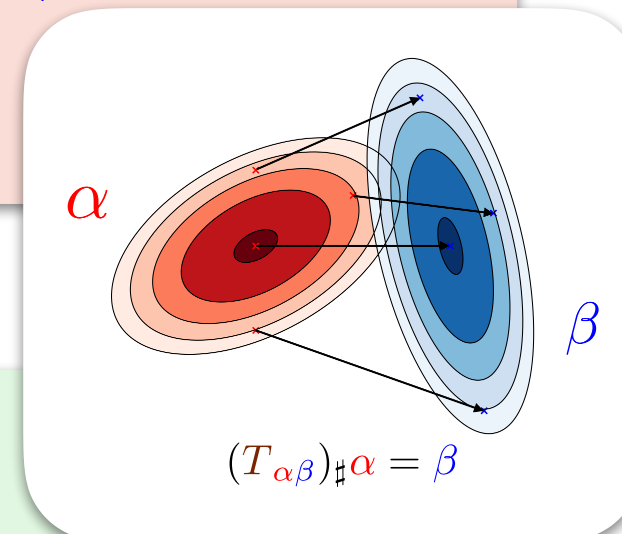
The Bures-Wasserstein Geometry

Prop. Gaussian Monge maps (if not invertible, use pseudo-inverses)

Let $\alpha = \mathcal{N}(\mathbf{a}, \mathbf{A})$ and $\beta = \mathcal{N}(\mathbf{b}, \mathbf{B})$ s.t. $\mathrm{Im}\mathbf{B} \subset \mathrm{Im}\mathbf{A}$. Then

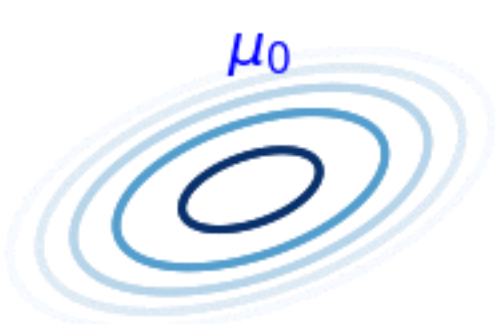
$T_{\alpha\beta} : x \mapsto \mathbf{T}^{\mathbf{AB}}(x - \mathbf{a}) + \mathbf{b}$ is the Monge map from α to β,

with $\mathbf{T}^{\mathbf{AB}} \overset{\text{def}}{=} \mathbf{A}^{-1/2}(\mathbf{A}^{1/2}\mathbf{B}\mathbf{A}^{1/2})\mathbf{A}^{-1/2}$.



Prop. Riemannian geodesics *(Takatsu 2011)*

$$\mathbf{C}_{\mathbf{AB}}(t) = [(1-t)\mathbf{I}_\mathrm{d} + t\mathbf{T}^{\mathbf{AB}}]\mathbf{A}[(1-t)\mathbf{I}_\mathrm{d} + t\mathbf{T}^{\mathbf{AB}}], \quad t \in [0,1]$$



Elliptical Distributions

Gaussian distributions:

$$\mathrm{d}\mathcal{N}(\mathbf{a},\mathbf{A})(x) = g((x-\mathbf{a})^T\mathbf{A}^{-1}(x-\mathbf{a}))\mathrm{d}x, \quad g(x) = |2\pi\mathbf{A}|^{-1/2}\exp(-x/2)$$

Elliptical distributions:

- Everything (Wasserstein distance, Monge maps, etc.) remains valid for any

$$g:\mathbb{R}^d \to \mathbb{R}_+ \text{ s.t. } \int_{\mathbb{R}^d} g(\|x\|^2_{\mathbf{A}^{-1}})\mathrm{d}x = 1$$

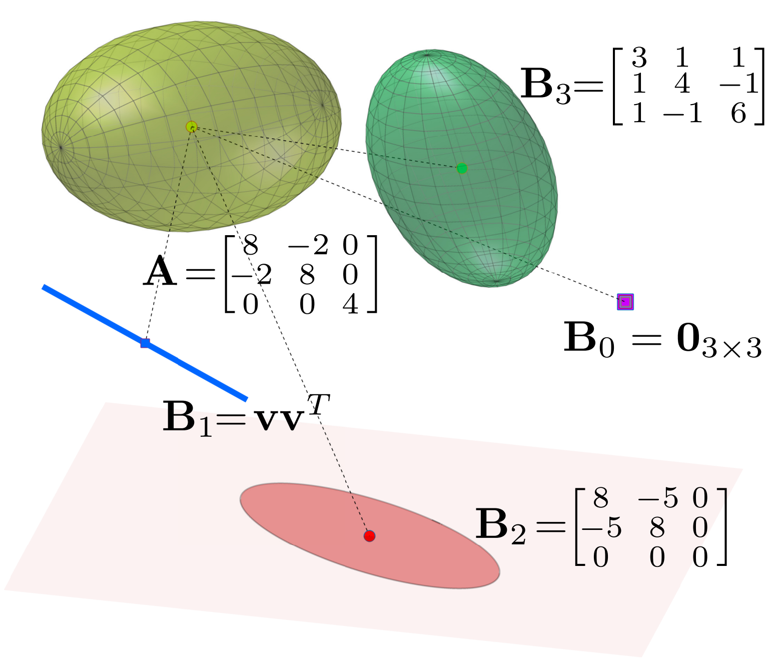
Gelbrich, 1990

- Includes degenerate distributions: replace - $\mathbf{A}^{-1}$ with $\mathbf{A}^\dagger$ (pseudo-inverse)

  - $\mathrm{d}x$ with $\mathrm{d}\lambda_{\mathrm{Im}\mathbf{A}}(x)$

Examples:

- Dirac measures ($\mathbf{A}=0$)
- Uniform measures on ellipsoids ($g(\cdot) \propto \mathbb{1}_{\{\cdot\leq 1\}}$)
- *"Anything with elliptical level sets"*



Bures-Wasserstein Gradient Descent

Gradient-Based Optimization

- Most ML apps. can be cast as minimization problems: $\min_{\theta \in \Theta} \mathbb{E}_{x \sim \mathcal{P}}[l_\theta(x)]$
- Classically solved with (stochastic) gradient descent: $\theta \leftarrow \theta - \eta \nabla_\theta l_\theta(x_i)$

Can we use Bures-Wasserstein as a loss function?

Example: Bures-Wasserstein barycenters

Minimization problem: $\min_{\alpha = \mathcal{N}(\mathbf{a}, \mathbf{A})} \frac{1}{n} \sum_{i=1}^{n} W_2^2(\alpha, \beta_i)$

Gradient update: $\mathbf{A} \leftarrow \mathbf{A} - \eta \nabla_{\mathbf{A}} \frac{1}{n} \sum_{i=1}^{n} W_2^2(\alpha, \beta_i)$

Requirements

1. Compute $\mathfrak{B}^2(\mathbf{A}, \mathbf{B})$
2. Compute $\nabla \mathfrak{B}^2(\mathbf{A}, \mathbf{B})$
3. Do gradient updates

Using chain rule and backprop, we can then generalise to $\min_{\alpha = \mathcal{N}(\mathbf{a}, \mathbf{A})} f(W_2^2(\alpha, \beta))$

Computing and differentiating Bures

$$\mathfrak{B}^2(\mathbf{A},\mathbf{B}) \stackrel{\text{def}}{=} \mathrm{Tr}\mathbf{A} + \mathrm{Tr}\mathbf{B} - 2\mathrm{Tr}(\mathbf{A}^{1/2}\mathbf{B}\mathbf{A}^{1/2})^{1/2}$$

Challenge: How to compute and differentiate $\mathfrak{B}(\mathbf{A},\mathbf{B})$?

Naïve idea: compute and differentiate matrix square roots using SVD.

But:

- SVD is expensive, and we need 2: $\mathbf{A}^{1/2}$ and $(\mathbf{A}^{1/2}\mathbf{B}\mathbf{A}^{1/2})^{1/2}$,
- Automatic differentiation of SVD can be unstable (e.g. with non-distinct singular values or singular matrices).

The Monge Map is All You Need

The Bures distance and its gradient can be computed from $\mathbf{T}^{\mathbf{AB}}$.

$$\mathfrak{B}^2(\mathbf{A},\mathbf{B}) = \mathrm{Tr}\mathbf{A} + \mathrm{Tr}\mathbf{B} - 2\mathrm{Tr}(\mathbf{A}^{1/2}\mathbf{B}\mathbf{A}^{1/2})^{1/2}$$
$$= \mathrm{Tr}\mathbf{A} + \mathrm{Tr}\mathbf{B} - 2\mathrm{Tr}(\mathbf{A}\mathbf{T}^{\mathbf{AB}})$$

Prop. Let $\mathbf{A},\mathbf{B} \in S_+^d$, s.t. $\mathrm{Im}\mathbf{B} \subset \mathrm{Im}\mathbf{A}$. Then,

$$\nabla_{\mathbf{A}}\mathfrak{B}^2(\mathbf{A},\mathbf{B}) = \mathbf{I}_{\mathrm{d}} - \mathbf{T}^{\mathbf{AB}}$$

To compute both $\nabla_{\mathbf{A}}\mathfrak{B}^2(\mathbf{A},\mathbf{B})$ and $\nabla_{\mathbf{B}}\mathfrak{B}^2(\mathbf{A},\mathbf{B})$, we need a method to compute

- $\mathbf{T}^{\mathbf{AB}} = \mathbf{A}^{-1/2}(\mathbf{A}^{1/2}\mathbf{B}\mathbf{A}^{1/2})^{1/2}\mathbf{A}^{-1/2}$, and
- $\mathbf{T}^{\mathbf{BA}} = \mathbf{B}^{-1/2}(\mathbf{B}^{1/2}\mathbf{A}\mathbf{B}^{1/2})^{1/2}\mathbf{B}^{-1/2} = (\mathbf{T}^{\mathbf{AB}})^{-1}$

Newton-Schulz iterations

$$\mathbf{Y}_{k+1} = \frac{1}{2}\mathbf{Y}_k(3\mathbf{I}_\mathrm{d} - \mathbf{Y}_k\mathbf{Z}_k\mathbf{Y}_k), \quad \mathbf{Y}_0 = \mathbf{B}$$
$$\mathbf{Z}_{k+1} = \frac{1}{2}\mathbf{Z}_k(3\mathbf{I}_\mathrm{d} - \mathbf{Z}_k\mathbf{Y}_k\mathbf{Z}_k), \quad \mathbf{Z}_0 = \mathbf{A}$$

Prop. *(Higham, Mackey, Mackey & Tisseur, 2005)*

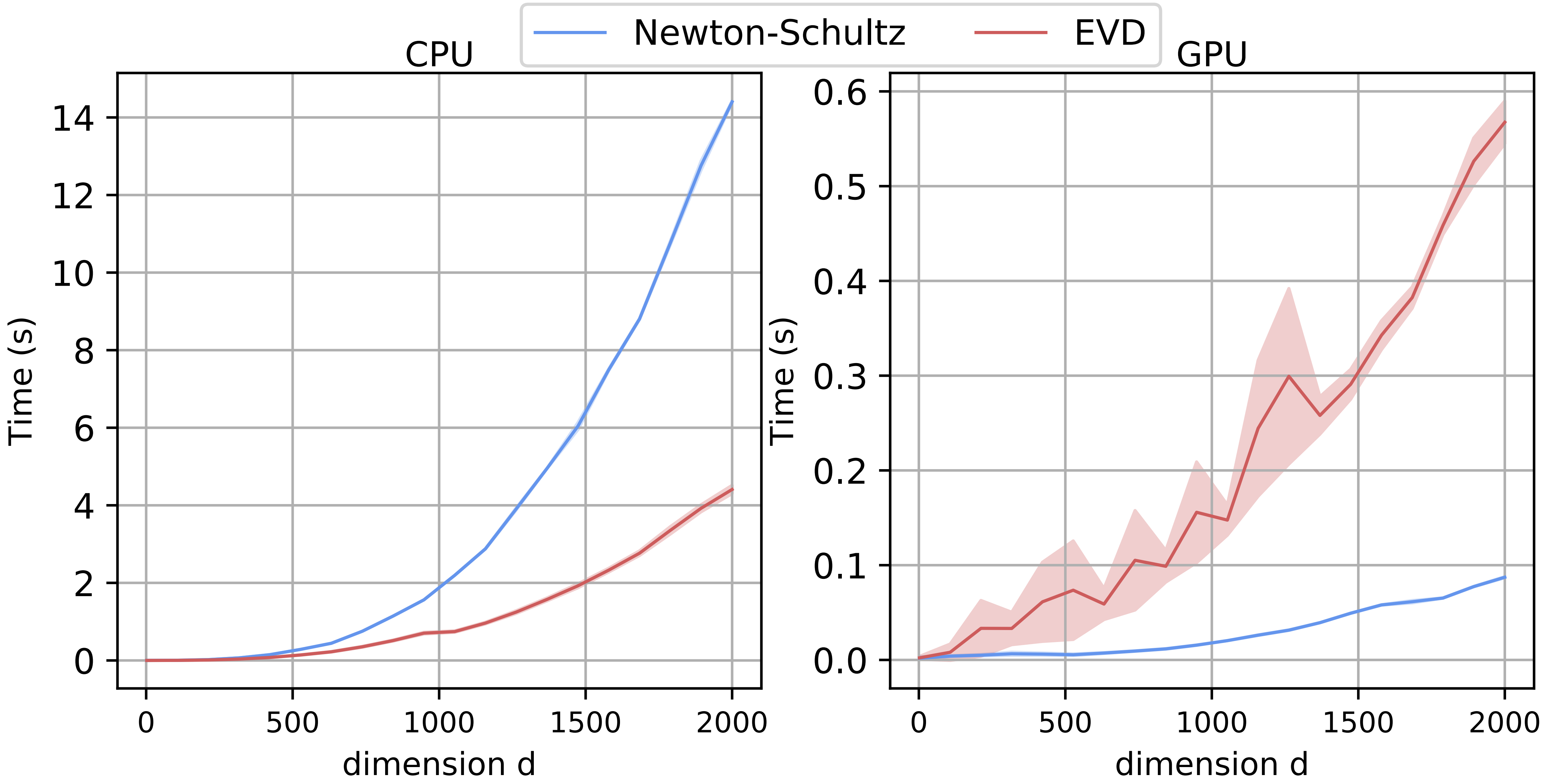
If $\|\mathbf{I}_\mathrm{d} - \left(\begin{smallmatrix} 0 & \mathbf{B} \\ \mathbf{A} & 0 \end{smallmatrix}\right)^2\|_\mathrm{op} < 1$, then $\mathbf{Y}_k \to \mathbf{T}^{\mathbf{AB}}$ and $\mathbf{Z}_k \to \mathbf{T}^{\mathbf{BA}}$ *quadratically**.

*(i.e. $\exists c > 0, \|\mathbf{Y}_{k+1} - \mathbf{T}^{\mathbf{AB}}\|_\mathrm{op} \leq c\|\mathbf{Y}_k - \mathbf{T}^{\mathbf{AB}}\|_\mathrm{op}^2$)

Why bother?

- Easy to parallelise on GPUs (only requires matrix multiplications),
- Yields both $\mathbf{T}^{\mathbf{AB}}$ and $\mathbf{T}^{\mathbf{BA}}$: we get $\nabla_\mathbf{A}\mathfrak{B}^2(\mathbf{A},\mathbf{B})$ and $\nabla_\mathbf{B}\mathfrak{B}^2(\mathbf{A},\mathbf{B})$.

Newton-Schulz vs SVD



Dealing with PSD constraints: avoiding projections

Last issue: $\mathbf{A} - t\nabla_{\mathbf{A}}\mathfrak{B}^2(\mathbf{A}, \mathbf{B})$ is not necessarily PSD.

Option 1: Projected gradient descent. $\Rightarrow$ Requires SVD. ✘

Dealing with PSD constraints: avoiding projections

Last issue: $\mathbf{A} - t\nabla_{\mathbf{A}}\mathfrak{B}^2(\mathbf{A}, \mathbf{B})$ is not necessarily PSD.

Option 1: Projected gradient descent. $\Rightarrow$ Requires SVD. ✗

Option 2: Reparameterize.

- Write $\mathbf{A} = \Pi(\mathbf{L_A}) \stackrel{\text{def}}{=} \mathbf{L_A}\mathbf{L_A}^T$ (necessarily PSD).
- Optimize w.r.t. $\mathbf{L_A}$ and $\mathbf{L_B}$.

Dealing with PSD constraints: avoiding projections

Last issue: $\mathbf{A} - t\nabla_{\mathbf{A}}\mathfrak{B}^2(\mathbf{A},\mathbf{B})$ is not necessarily PSD.

Option 1: Projected gradient descent. → Requires SVD. ✗

Option 2: Reparameterize.

- Write $\mathbf{A} = \Pi(\mathbf{L_A}) \stackrel{\text{def}}{=} \mathbf{L_A}\mathbf{L_A}^T$ (necessarily PSD).
- Optimize w.r.t. $\mathbf{L_A}$ and $\mathbf{L_B}$.

What is the effect on gradient descent?

Dealing with PSD constraints: avoiding projections

Last issue: $\mathbf{A} - t\nabla_{\mathbf{A}}\mathfrak{B}^2(\mathbf{A}, \mathbf{B})$ is not necessarily PSD.

Option 1: Projected gradient descent. $\Rightarrow$ Requires SVD. ✘

Option 2: Reparameterize.

- Write $\mathbf{A} = \Pi(\mathbf{L}_{\mathbf{A}}) \stackrel{\text{def}}{=} \mathbf{L}_{\mathbf{A}}{\mathbf{L}_{\mathbf{A}}}^T$ (necessarily PSD).
- Optimize w.r.t. $\mathbf{L}_{\mathbf{A}}$ and $\mathbf{L}_{\mathbf{B}}$.

What is the effect on gradient descent?

Riemannian geodesics at the cost of Euclidean descent (*BM & Cuturi, 2018*)

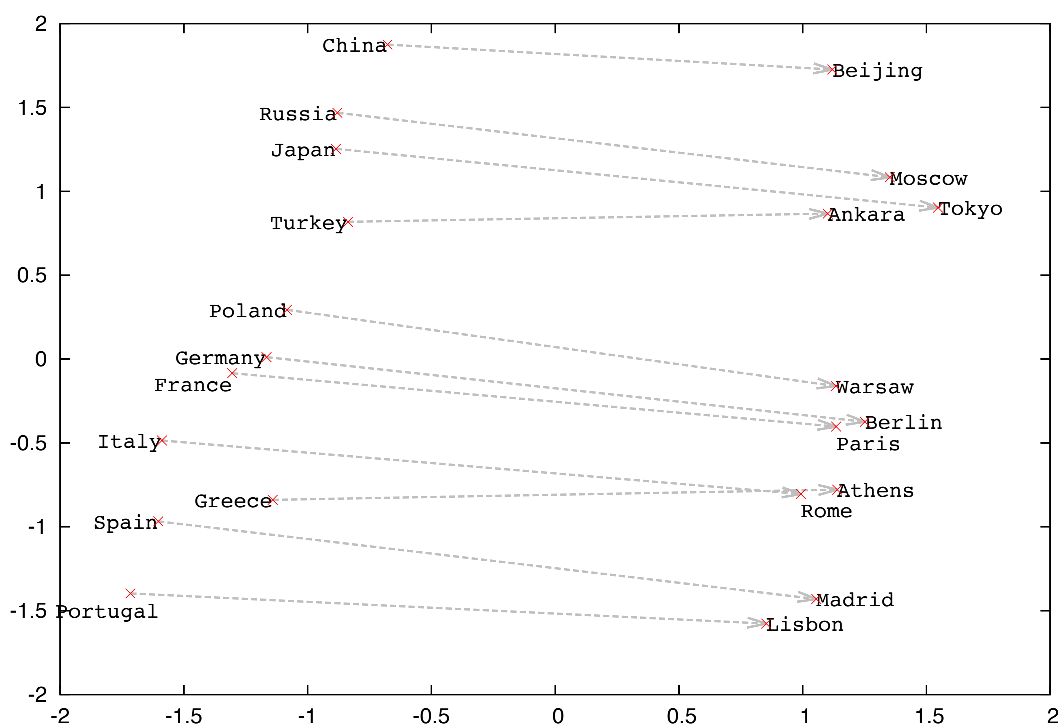
- $\nabla_{\mathbf{L}_{\mathbf{A}}} \frac{1}{2}\mathfrak{B}^2(\mathbf{L}_{\mathbf{A}}{\mathbf{L}_{\mathbf{A}}}^T, \mathbf{B}) = (\mathbf{I}_{\mathrm{d}} - \mathbf{T}^{\mathbf{AB}})\mathbf{L}_{\mathbf{A}}$
- Riem. geodesics: $\mathbf{C}_{\mathbf{AB}}(t) = [(1-t)\mathbf{I}_{\mathrm{d}} + t\mathbf{T}^{\mathbf{AB}}]\mathbf{A}[(1-t)\mathbf{I}_{\mathrm{d}} + t\mathbf{T}^{\mathbf{AB}}], \quad t \in [0,1]$
- "$\Pi(\cdot)$ makes $\mathfrak{B}^2$ flat": $\mathbf{L}_{\mathbf{A}} - t\nabla_{\mathbf{L}_{\mathbf{A}}}\frac{1}{2}\mathfrak{B}^2(\mathbf{A}, \mathbf{B}) \in \Pi^{-1}\{\mathbf{C}_{\mathbf{AB}}(t)\}$

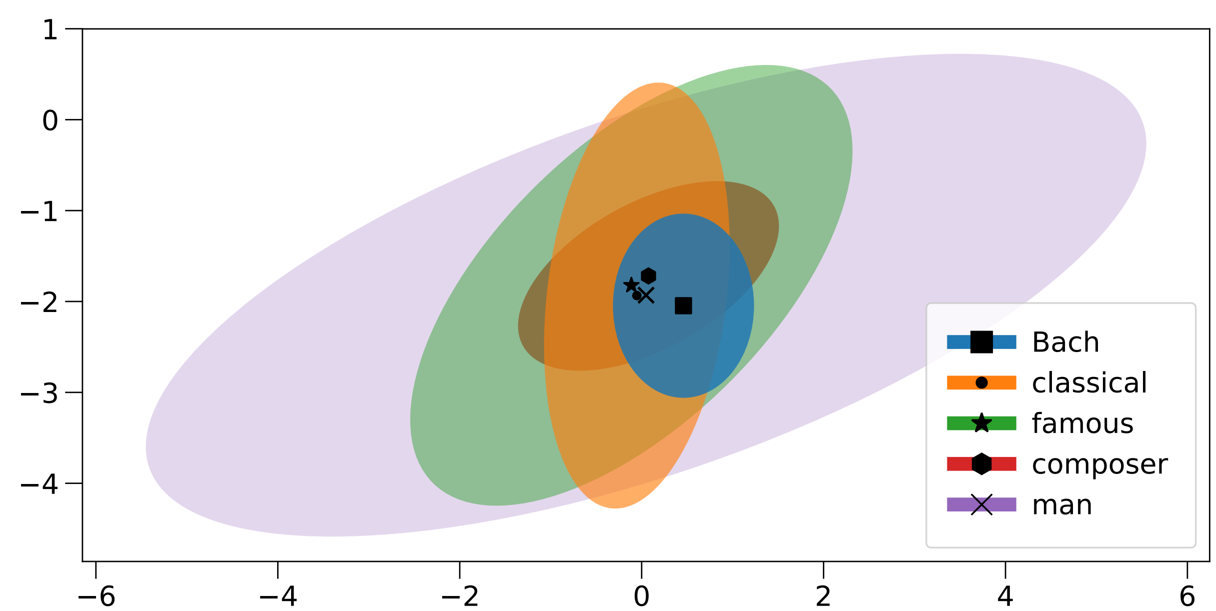
Applications

Application: Learning Representations

Problem: finding representations for objects x in some space $\mathcal{X}$ (e.g. words, graphs, high-dimensional vectors, …)

- **Classic approach:** represent each x as a point $y \in \mathbb{R}^k$.
- ***Elliptical embeddings*** *(BM & Cuturi, 2018)*: represent each x as an elliptical distr. α with parameters $\mathbf{a} \in \mathbb{R}^k$ and $\mathbf{A} \in \mathcal{S}_+^k$, in the Bures-Wassertein geometry.





Allows to encode spread, or uncertainty.

Elliptical Word Embeddings

Training: $\min \sum_{w} \sum_{c \in \mathrm{Pos}(w)} \left[M - [\mu_w : \nu_c]] + \frac{1}{n} \sum_{c' \in \mathrm{Neg}(w)} [\mu_w : \nu_{c'}] \right]_+$

ALL MODELS ARE WRONG BUT SOME ARE USEFUL
ALL MODELS ARE WRONG BUT SOME ARE USEFUL
ALL MODELS ARE WRONG BUT SOME ARE USEFUL

Polarization: $[\alpha : \beta] \stackrel{\text{def}}{=} \langle \mathbf{a}, \mathbf{b} \rangle + \mathrm{Tr}(\mathbf{A}^{1/2}\mathbf{B}\mathbf{A}^{1/2})^{1/2}$

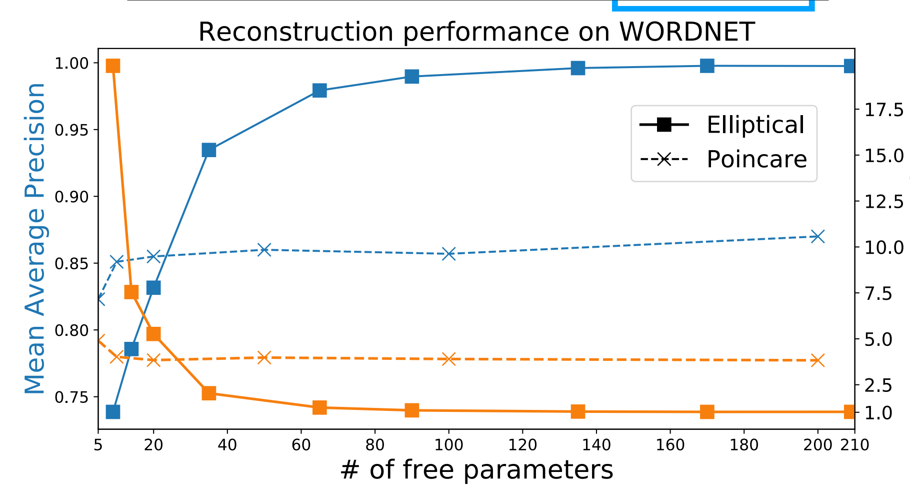
Datasets

`ukWaC` + `WaCkypedia`: 3 billion tokens, 250K unique[16]
`WordNet`: DAG, 80K unique nouns, 740K relationships[17]

Implementation: `cupy` (GPU) + `cython`, on GitHub.

Similarity Benchmark: Spearman Rank Correlation

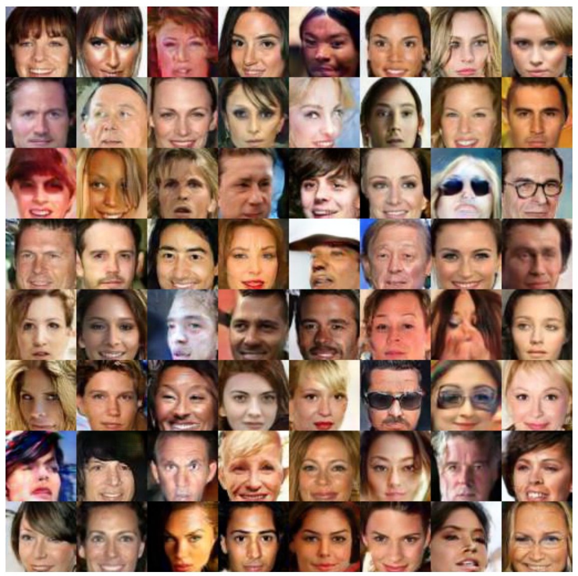
Dataset	W2G/45/C	Ell/12/CM
SimLex	**25.09**	24.09
WordSim	53.45	**66.02**
WordSim-R	61.70	**71.07**
WordSim-S	48.99	**60.58**
MEN	65.16	**65.58**
MC	59.48	**65.95**
RG	**69.77**	65.58
YP	**37.18**	25.14
MT-287	**61.72**	59.53
MT-771	**57.63**	56.78
RW	**40.14**	29.04



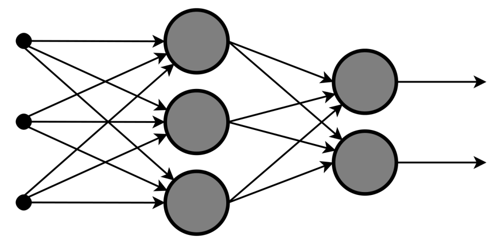
[16] L. Vilnis et al. "Word representations via Gaussian embedding". *ICLR* [2015].
[17] M. Nickel et al. "Poincaré Embeddings for Learning Hierarchical Representations". *NeurIPS*. 2017.

Application: Fréchet Inception Distance (FID, *Heusel et al. 2017*)

A quality score for generative models.



Samples from GAN

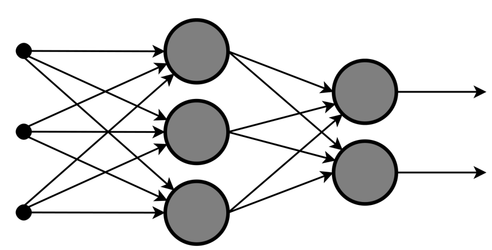


Neural Net (inception)

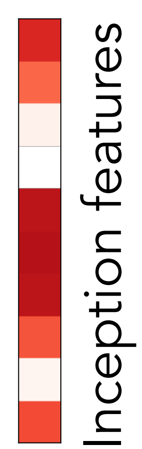
Mean $\hat{\mathbf{m}}$, Cov. $\hat{\Sigma}$

$$\mathrm{FID} = \|\hat{\mathbf{m}} - \mathbf{m}_{\mathrm{true}}\|^2 + \mathfrak{B}^2(\hat{\Sigma}, \Sigma_{\mathrm{true}})$$

Ground Truth Dataset



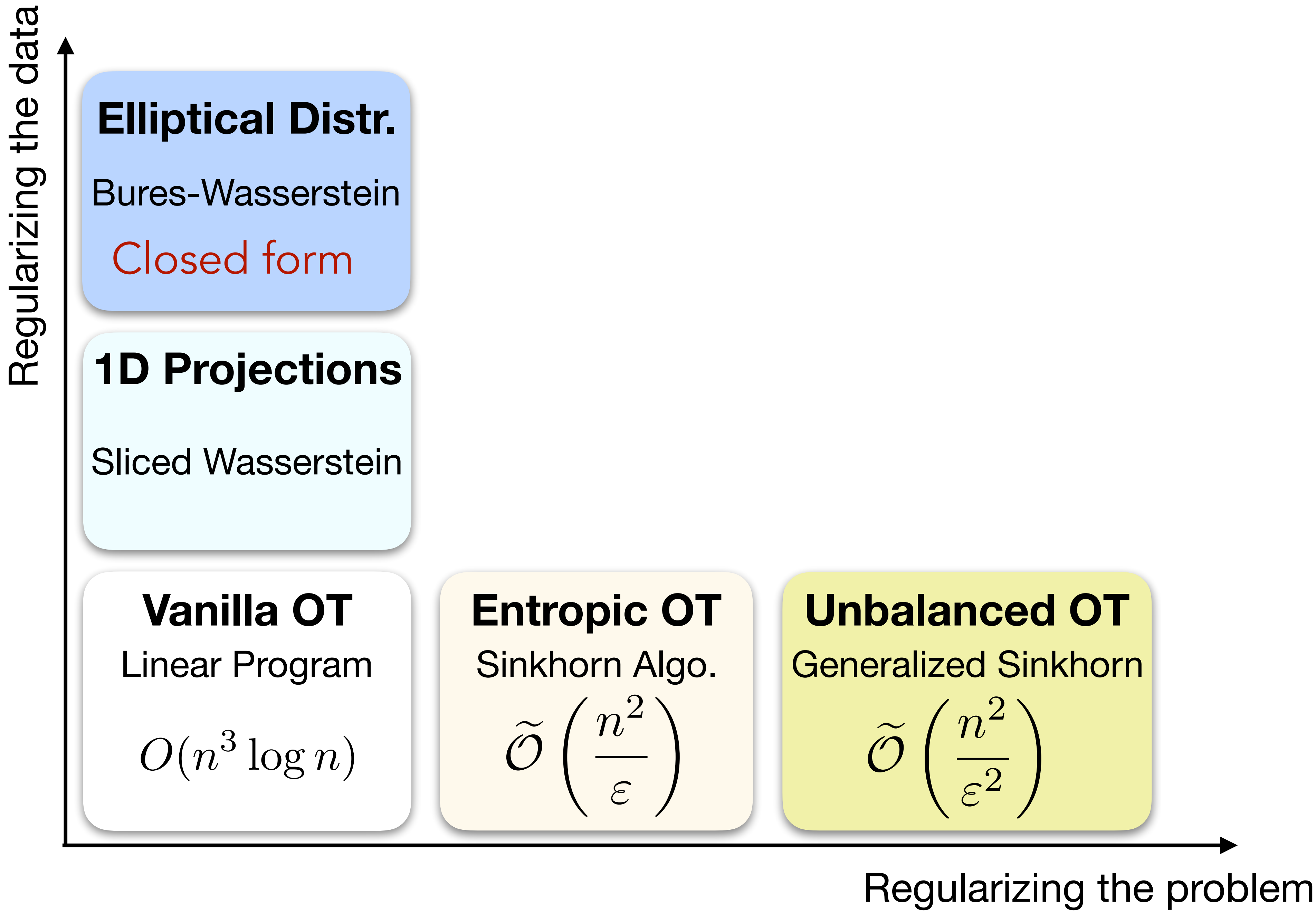
Neural Net (inception)



Mean $\mathbf{m}_{\mathrm{true}}$, Cov. Σ_{true}

Opening: Regularizing OT

Real data comes in a discrete form: $\hat{\mu} = \frac{1}{n}\sum_{i=1}^{n} \delta_{x_i}$. What to do with it?



Opening: Regularizing OT

Real data comes in a discrete form: $\hat{\mu} = \frac{1}{n}\sum_{i=1}^{n} \delta_{x_i}$. What to do with it?

Regularizing the data

Elliptical Distr.

Bures-Wasserstein

Closed form

1D Projections

Sliced Wasserstein

Vanilla OT

Linear Program

$O(n^3 \log n)$

Entropic OT

Sinkhorn Algo.

$\tilde{\mathcal{O}}\left(\frac{n^2}{\varepsilon}\right)$

Unbalanced OT

Generalized Sinkhorn

$\tilde{\mathcal{O}}\left(\frac{n^2}{\varepsilon^2}\right)$

Regularizing the problem

Opening: Regularizing OT

Real data comes in a discrete form: $\hat{\mu} = \frac{1}{n}\sum_{i=1}^{n} \delta_{x_i}$. What to do with it?

Exploration path

Regularizing the data

Elliptical Distr.

Bures-Wasserstein

Closed form

1D Projections

Sliced Wasserstein

Vanilla OT

Linear Program

$O(n^3 \log n)$

Entropic OT

Sinkhorn Algo.

$\widetilde{\mathcal{O}}\left(\frac{n^2}{\varepsilon}\right)$

Unbalanced OT

Generalized Sinkhorn

$\widetilde{\mathcal{O}}\left(\frac{n^2}{\varepsilon^2}\right)$

Regularizing the problem

Janati, BM, Peyré & Cuturi, 2020

NeurIPS 2020: Oral 09/12 at 15:15 (Optimization track)

References I

Arjovsky, M., S. Chintala, & L. Bottou. “Wasserstein Generative Adversarial Networks”. *ICML*. 2017.

Benamou, J.-D. “Numerical resolution of an “unbalanced” mass transport problem”. *ESAIM: Mathematical Modelling and Numerical Analysis - Modélisation Mathématique et Analyse Numérique* 37 (2003), pp. 851–868.

Benamou, J.-D. & Y. Brenier. “A computational fluid mechanics solution to the Monge-Kantorovich mass transfer problem”. *Numerische Mathematik* (2000).

Bhatia, R., T. Jain, & Y. Lim. “On the Bures-Wasserstein distance between positive definite matrices”. *Expositiones Mathematicae* (2018).

Bonneel, N. et al. “Sliced and Radon Wasserstein Barycenters of Measures”. *Journal of Mathematical Imaging and Vision* (2015).

Brenier, Y. “Décomposition polaire et réarrangement monotone des champs de vecteurs”. *CR Acad. Sci. Paris Sér. I Math.* (1987).

References II

Bures, D. "An extension of Kakutani's theorem on infinite product measures to the tensor product of semifinite w*-algebras". *Trans. of the Am. Math. Soc.* (1969).

Cambanis, S., S. Huang, & G. Simons. "On the theory of elliptically contoured distributions". *Journal of Multivariate Analysis* 11.3 (1981), pp. 368–385.

Chizat, L. "Unbalanced optimal transport: Models, numerical methods, applications". PhD thesis. 2017.

Chizat, L. & F. Bach. "On the global convergence of gradient descent for over-parameterized models using optimal transport". *Advances in neural information processing systems*. 2018, pp. 3036–3046.

Cuturi, M. "Sinkhorn distances: Lightspeed computation of OT". *NeurIPS*. 2013.

References III

Dowson, D. & B. Landau. "The Fréchet distance between multivariate normal distributions". *Journal of multivariate analysis* (1982).

Gelbrich, M. "On a formula for the L2 Wasserstein metric between measures on Euclidean and Hilbert spaces". *Mathematische Nachrichten* (1990).

Genevay, A. et al. "Stochastic optimization for large-scale OT". *NeurIPS*. 2016.

Givens, C. R., R. M. Shortt, et al. "A class of Wasserstein metrics for probability distributions.". *The Michigan Mathematical Journal* 31.2 (1984), pp. 231–240.

Heusel, M. et al. "GANs Trained by a Two Time-Scale Update Rule Converge to a Local Nash Equilibrium". *NeurIPS*. 2017.

Higham, N. J. *Functions of Matrices: Theory and Computation*. SIAM, 2008.

References IV

Janati, H. et al. “Entropic Optimal Transport between (Unbalanced) Gaussian Measures has a Closed Form”. *NeurIPS* (2020).

Kantorovich, L. V. “On the translocation of masses”. *Dokl. Akad. Nauk. USSR*. 1942.

Makkuva, A. V. et al. “Optimal transport mapping via input convex neural networks”. *ICML* (2020).

Malagò, L., L. Montrucchio, & G. Pistone. “Wasserstein-Riemannian Geometry of Positive-definite Matrices”. *arXiv preprint arXiv:1801.09269* (2018).

Mérigot, Q. “A multiscale approach to optimal transport”. *Comp. Grap. Forum*. 2011.

Mikolov, T. et al. “Distributed representations of words and phrases and their compositionality”. *NeurIPS*. 2013.

Monge, G. “Mémoire sur la théorie des déblais et des remblais”. *Histoire de l’Académie Royale des Sciences de Paris* (1781).

Muzellec, B. & M. Cuturi. “Generalizing Point Embeddings using the Wasserstein Space of Elliptical Distributions”. *NeurIPS*. 2018.

– .“Subspace detours: Building transport plans that are optimal on subspace projections”. *NeurIPS*. 2019.

Nickel, M. & D. Kiela. “Poincaré Embeddings for Learning Hierarchical Representations”. *NeurIPS*. 2017.

Olkin, I. & F. Pukelsheim. “The distance between two random vectors with given dispersion matrices”. *Linear Algebra and its Applications* (1982).

Peyré, G., M. Cuturi, et al. “Computational Optimal Transport: With Applications to Data Science”. *Foundations and Trends® in Machine Learning* 11.5-6 (2019), pp. 355–607.

Rabin, J. et al. “Wasserstein barycenter and its application to texture mixing”. *SSVM*. 2011.

References VI

- Seguy, V. et al. "Large-Scale Optimal Transport and Mapping Estimation". *ICLR*. 2018.
- Takatsu, A. "Wasserstein geometry of Gaussian measures". *Osaka J. Math.* (2011).
- Vilnis, L. & A. McCallum. "Word representations via Gaussian embedding". *ICLR* (2015).