Wasserstein Dictionary Learning

Collaborators: Matthieu Heitz, Nicolas Bonneel, Fred Ngolè, David Coeurjolly, Marco Cuturi, Gabriel Peyré, Jean-Luc Starck
morgan.schmitz@cea.fr

arXiv:1708.01955

26/01/2018

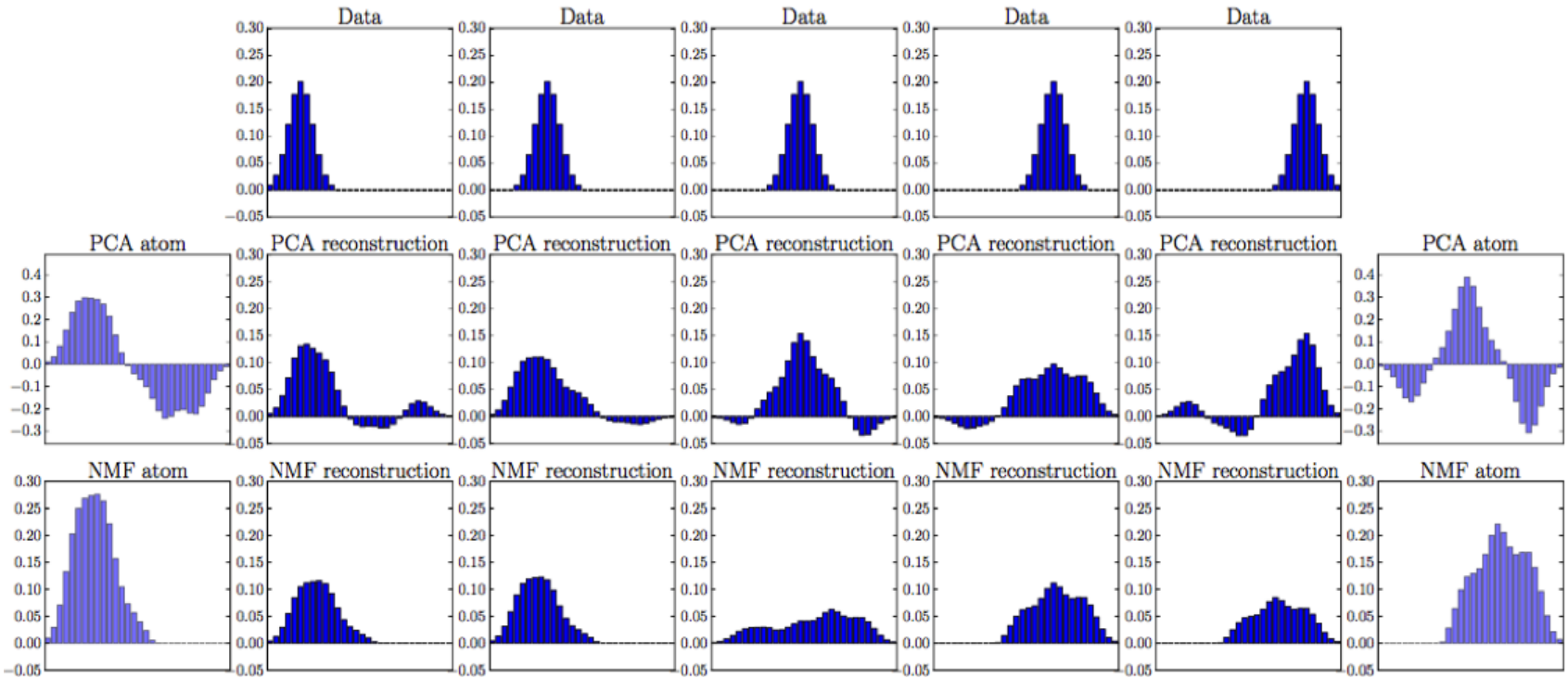
Dictionary Learning

Usual linear case

- Data $X \in \mathbb{R}^{N \times M}$, dictionary $D \in \mathbb{R}^{N \times S}$, codes $\Lambda \in \mathbb{R}^{S \times M}$
- Learn D and Λ such that $X \approx D\Lambda$
- Throw in whichever requirements suits you:
  - Compact representation? Take $S < N$ (dimensionality reduction)
  - Sparsity? Add l_0 or l_1 constraint
  - Positivity? Add a constraint! (NMF)
- Ultimately, data is reconstructed through *linear* combinations of dictionary atoms

Dictionary Learning

Toy example



Outline

Optimal Transport
Overview

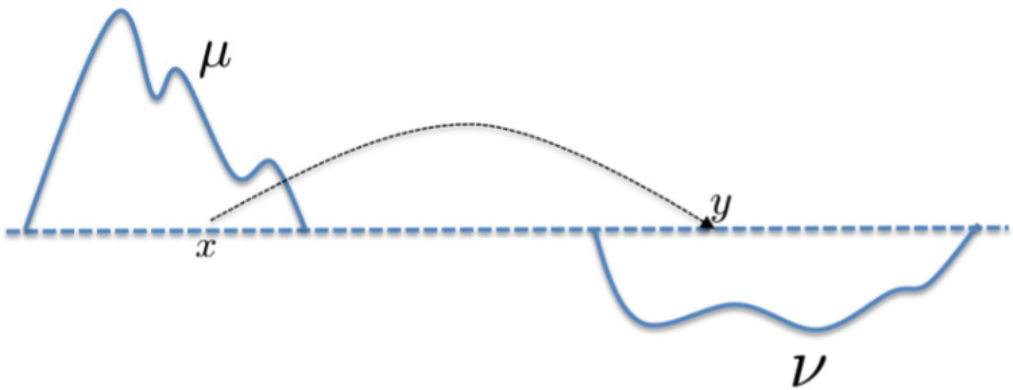


Figure: Graphical representation of the mass transportation problem: find the optimal way of moving a heap of sand μ into a hole ν knowing the cost of moving grains of sand to and from any position

Wasserstein Distances

Discrete setting

- Knowledge of the cost of moving: cost matrix $C \in \mathbb{R}^{N \times N}$
- Optimal Transport distance defined as:

$$W(\mu, \nu) = \min_{T \in \Pi(\mu,\nu)} \langle T, C \rangle$$

- Where $\Pi(\mu, \nu) := \left\{ T \in \mathbb{R}_+^{N \times N}, T\mathbb{1}_N = \mu, T^\top \mathbb{1}_N = \nu \right\}$ is the set of acceptable *transport plans*
- Recent numerical schemes allow for fast computation of (approximate) Wasserstein distances[1]

[1] Cuturi, 2013

Barycenters
Euclidean case

- Inputs $D = (d_1, \ldots, d_S)$ and weights $\lambda = (\lambda_1, \ldots, \lambda_S)$
- Euclidean barycenter:

$$B(D, \lambda) := \underset{u \in \mathbb{R}^N}{\operatorname{argmin}} \sum_{s=1}^{S} \lambda_s \|u - d_s\|_2^2 = \sum_{s=1}^{S} \lambda_s d_s$$

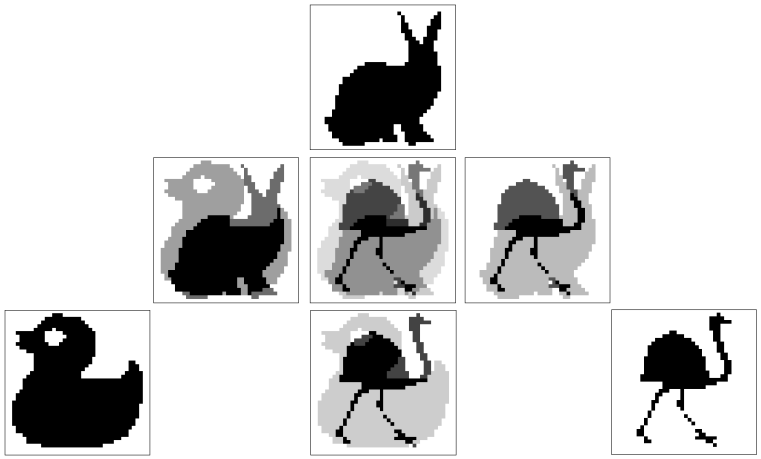


Figure: Euclidean simplex

Barycenters
Wasserstein case

- Similarly, define Wasserstein barycenters as:

$$P(D, \lambda) = \underset{u}{\operatorname{argmin}} \sum_{s=1}^{S} \lambda_s W_\gamma(u, d_s) = \ldots ?$$

- Iterative scheme to compute approximate Wasserstein barycenter: $P^{(L)}(D, \lambda) \approx P(D, \lambda)$ after L iterations

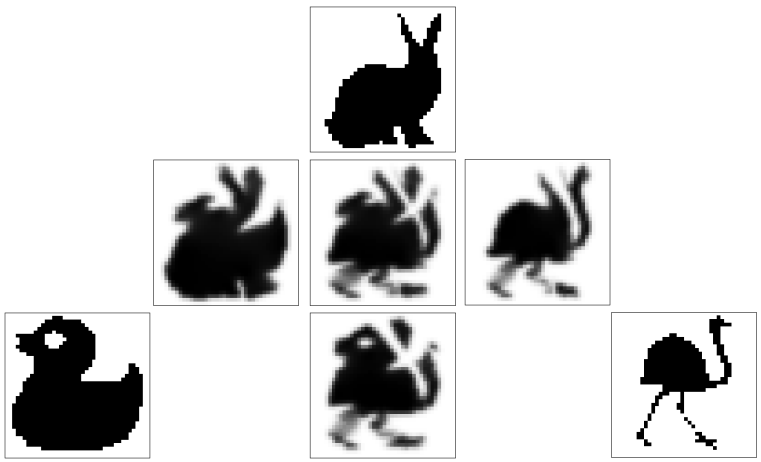
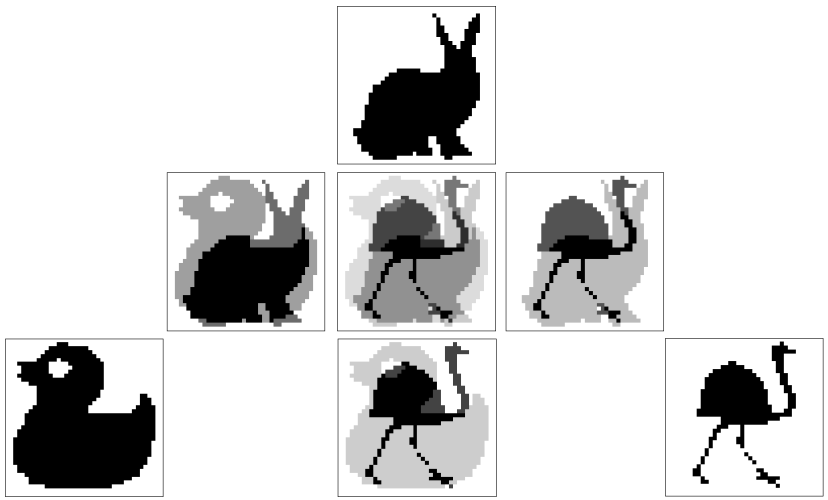


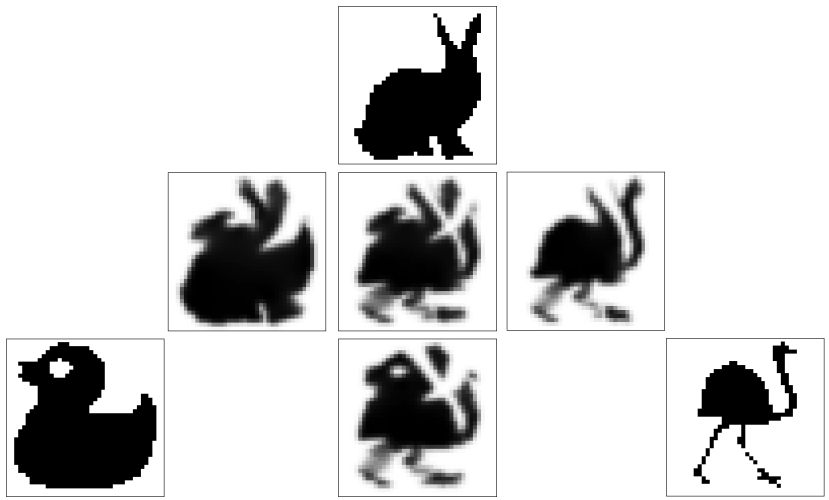
Figure: Wasserstein simplex

Outline

Barycenters
Simplices comparison



(a) Euclidean simplex



(b) Wasserstein simplex

Wasserstein Dictionary Learning

- Idea: instead of $X \approx D\Lambda$, how about $X \approx P(D, \Lambda)$?
- Optimization problem:

$$\underset{D,\Lambda}{\operatorname{argmin}}\, \mathcal{E}(D, \Lambda) := \sum_{i=1}^{M} L(x_i, P(D, \lambda_i))$$

- Gradient descent in both dictionary and weights: need $\partial_D P, \partial_\lambda P$

WDL & deep learning
Automatic differentiation

- Automatic differentiation: differentiate *algorithm* instead of analytical operator
- WDL: gradients are obtained by automatic differentiation of iterative scheme to compute Wasserstein barycenters
- Deep learning: gradients are obtained by backprop!

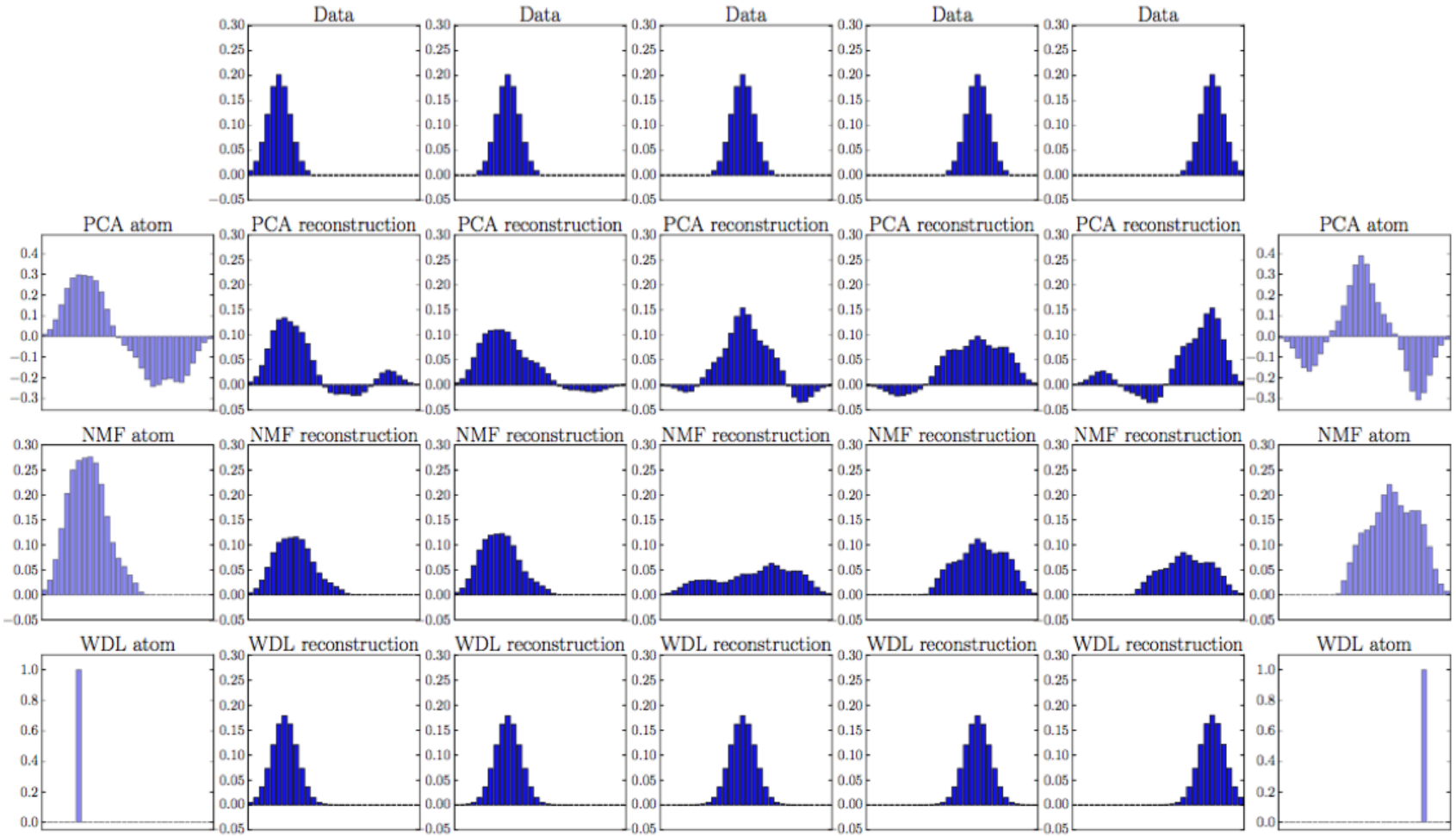
WDL & deep learning

WDL & autoencoders

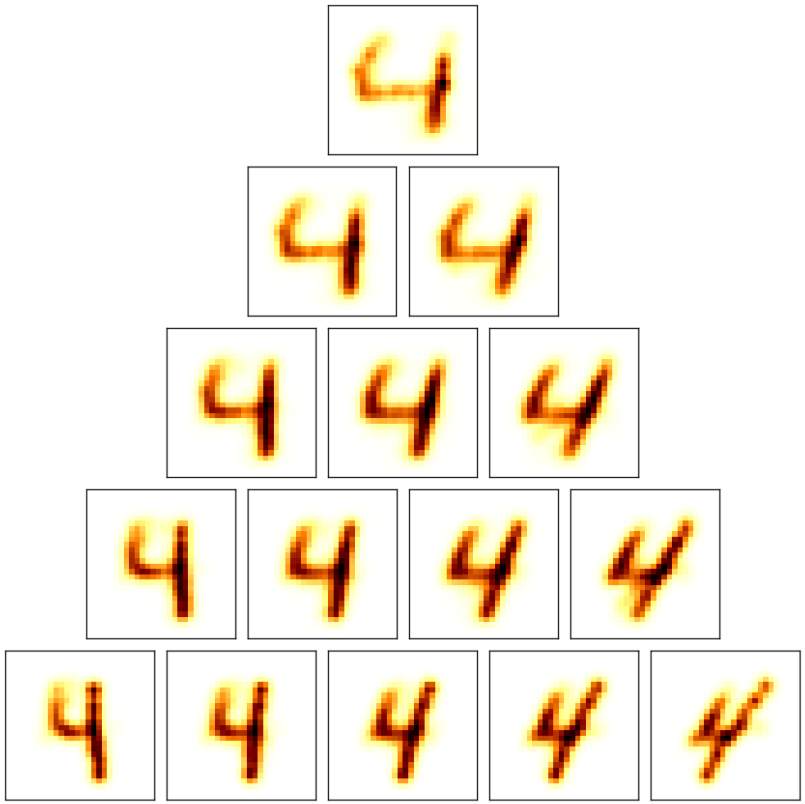
- Autoencoders: learn both encoder and decoder nets
- Learning a Wasserstein dictionary similar to learning one very specific such pair:
  - Encode new datapoint through its barycentric weights in the atoms' simplex;
  - The Wasserstein barycenter operator $P^{(L)}(D, .)$ *is* the decoder.

Outline

Toy examples
Translated gaussians



Toy examples
Handwritten digits



- WDL recovers Principal Wasserstein Geodesics[2]!

[2] Seguy & Cuturi 2015

Chromatic variation of the PSF

Motivation

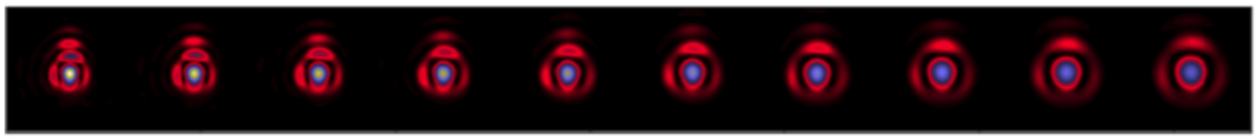


Figure: Simulated Euclid-like PSF variation for different wavelengths (400 to 900nm)

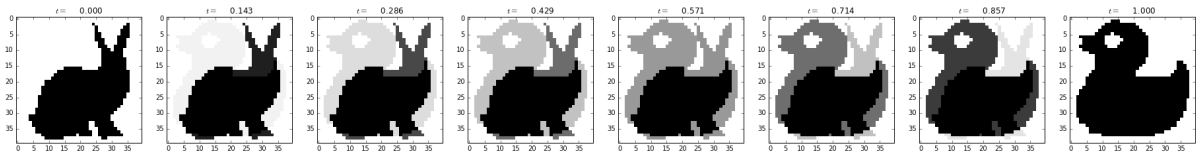


Figure: Euclidean barycenters

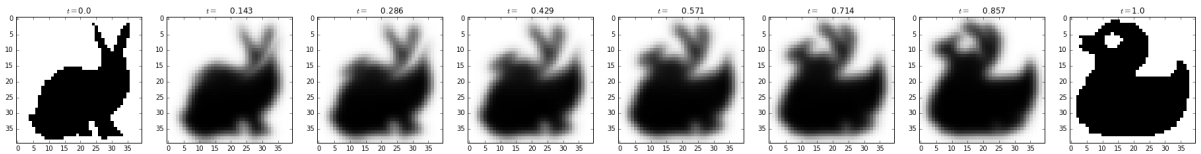


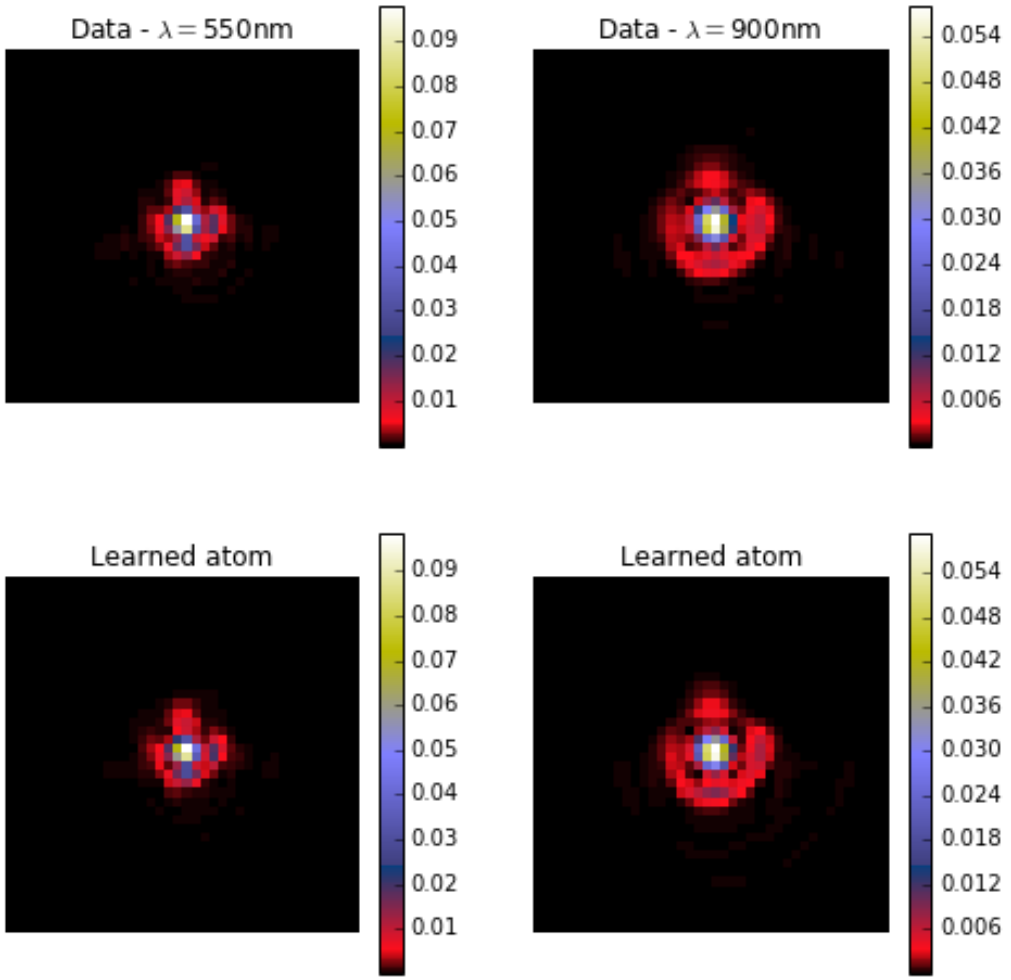
Figure: Wasserstein displacement interpolation

Chromatic variation of the PSF
Setting

- Training data: set of simulated Euclid-like PSFs at different wavelengths
- Learn 2 atoms on a set of monochromatic PSFs, initialize as constant images
- Initialize weights as projected wavelength

Chromatic variation of the PSF

Atoms learned



Chromatic variation of the PSF

Reconstruction

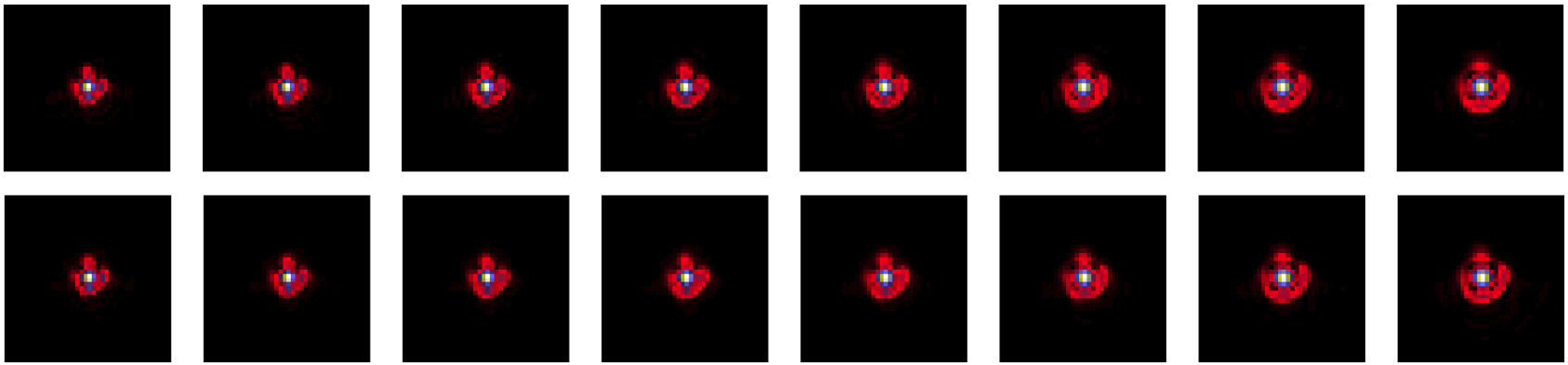


Figure: Original PSFs (top row) and their reconstruction with our method (bottom row)

Outline

Toward chromatic decomposition of star images

Real-life setting

- Stars give measurement of the PSF integrated with their own SED S_*
- Our objective is to obtain the integrated PSF for a given galaxy's SED $S_\bullet \neq S_*$
- Need to decompose PSF into 'monochromatic' components:

$$Y = \sum_{i=1}^{n} S_*(\lambda_i) H_i$$

  - Y: observed star image;
  - $S_*(\lambda_1), \ldots, S_*(\lambda_n)$: discretized star SED;
  - $H_1, \ldots, H_n$: monochromatic PSFs.

Toward chromatic decomposition of star images
λRCA

- Use Optimal Transport to break degeneracy
- Reconstruct star image as weighted sum of intermediary steps (Wasserstein barycenters) in the transportation of extreme-wavelengths PSFs
- Use whole field of stars to learn basis set of transport plans

Toward chromatic decomposition of star images

λRCA (preliminary results)

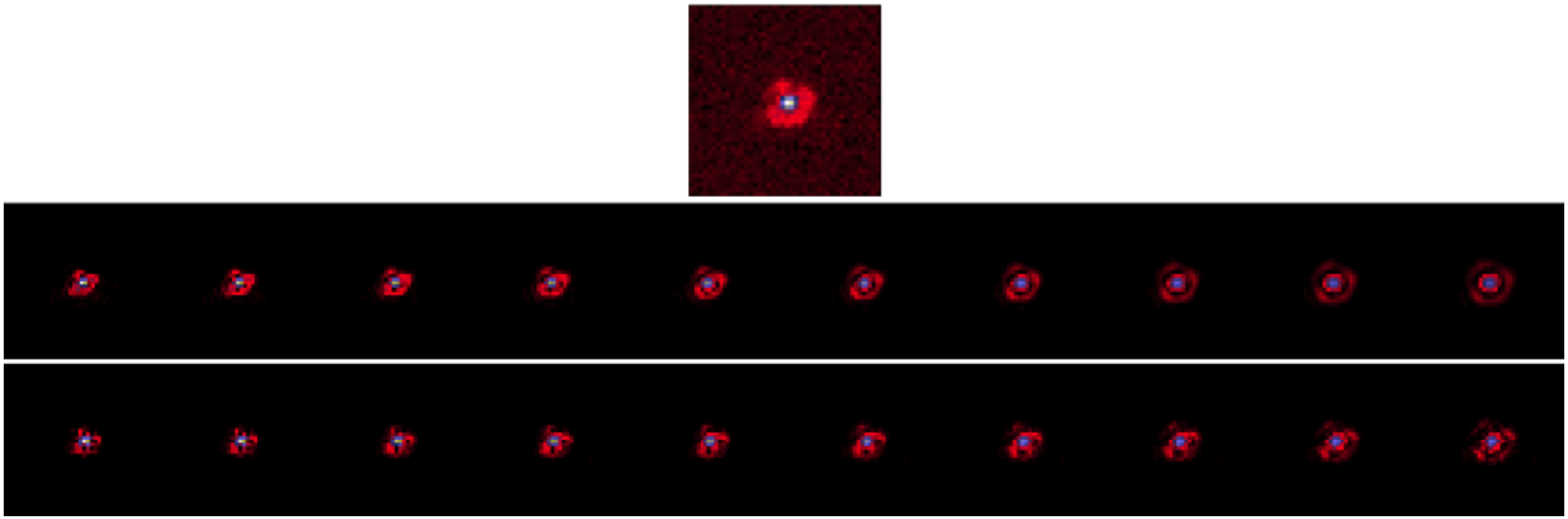


Figure: From top to bottom: observed polychromatic PSF, true monochromatic PSFs, reconstructed monochromatic PSFs

Summary

- Wasserstein Dictionary Learning: a new, Optimal Transport-based representation learning method
- Acknowledgements and funding:

CENTRE NATIONAL D'ÉTUDES SPATIALES

DE LA RECHERCHE À L'INDUSTRIE

EDALE

morgan.schmitz@cea.fr
http://www.cosmostat.org/people/mschmitz/

Appendix

Optimal Transport - WDL

Graphical representation

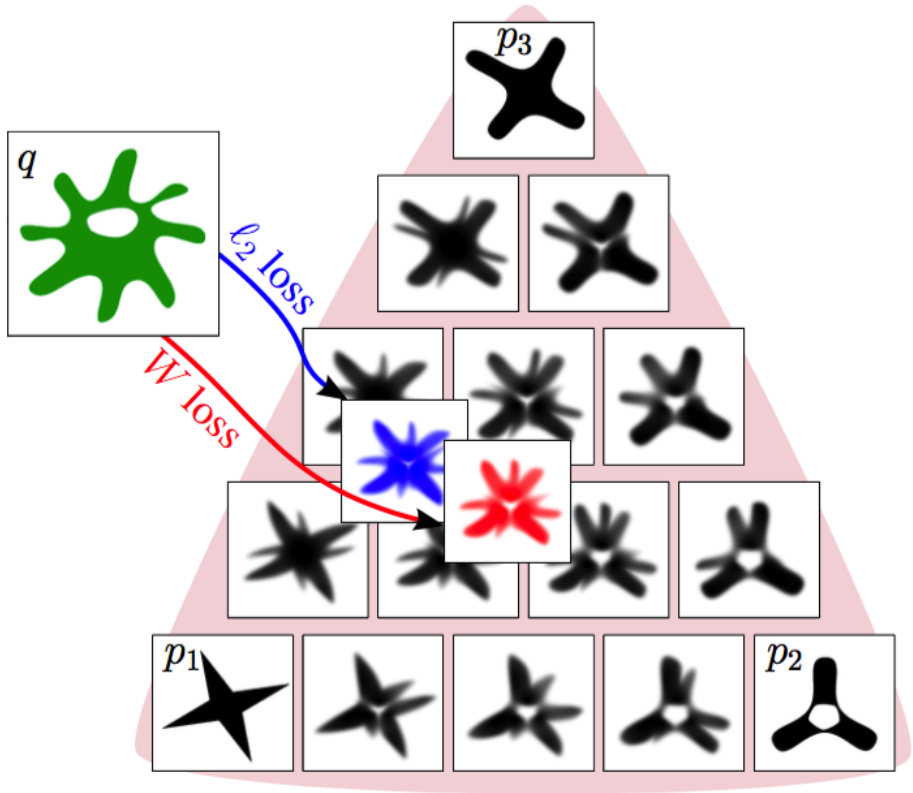


Figure: Graphical representation of the WDL approach. Source: Bonneel et al, 2016

Wasserstein Distances

Continuous setting

- The optimal cost achieved in the mass transportation problem defines an Optimal Transport distance between any two measures μ, ν on some space Ω:

$$W(\mu,\nu) := \min\left\{\int_{\Omega\times\Omega} c(x,y)d\pi(x,y),\ \pi\in\Pi(\mu,\nu)\right\}$$

- Where $\Pi(\mu,\nu)$ is the set of bivariate measures with marginals μ, ν:
  - $\int_\Omega \pi(x,y)dx = \nu(y)$
  - $\int_\Omega \pi(x,y)dy = \mu(x)$

Wasserstein Distances

Discrete setting

- Discrete case: if $|\Omega| = N$, c reduces to $C \in \mathbb{R}^{N \times N}$ and:

$$W(\mu, \nu) = \min_{T \in \Pi(\mu,\nu)} \langle T, C \rangle$$

- Where $\Pi(\mu, \nu) := \left\{ T \in \mathbb{R}_+^{N \times N}, T\mathbb{1}_N = \mu, T^\top \mathbb{1}_N = \nu \right\}$ is the set of acceptable *transport plans*
- Recent numerical schemes allow for fast computation of (approximate) Wasserstein distances[3]

[3] Cuturi, 2013

Appendix

Numerical Optimal Transport
Entropic penalty

- Adding an entropy penalty term:
  $W_\gamma(\mu,\nu) := \min_{T\in\Pi(\mu,\nu)}\langle T, C\rangle + \gamma H(T)$
- Leads to the Sinkhorn algorithm being useable for linear convergence to W_γ:
  - $b^{(0)} := \mathbb{1}_N$
  - $a^{(l)} := \frac{\nu}{K^\top b^{(l-1)}}$
  - $b^{(l)} := \frac{\mu}{K a^{(l)}}$
  - $T^{(L)} := \Delta\left(b^{(L)}\right) K \Delta\left(a^{(L)}\right)$
- Where $K = \exp(-C/\gamma)$

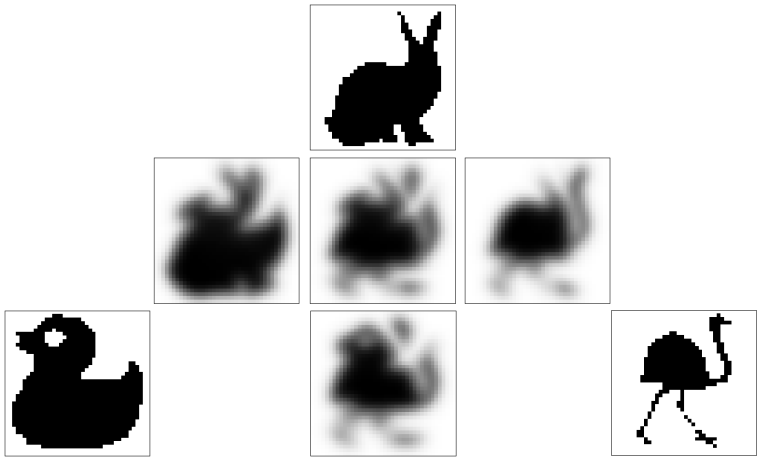
Numerical Optimal Transport

Entropic penalty & barycenter computation

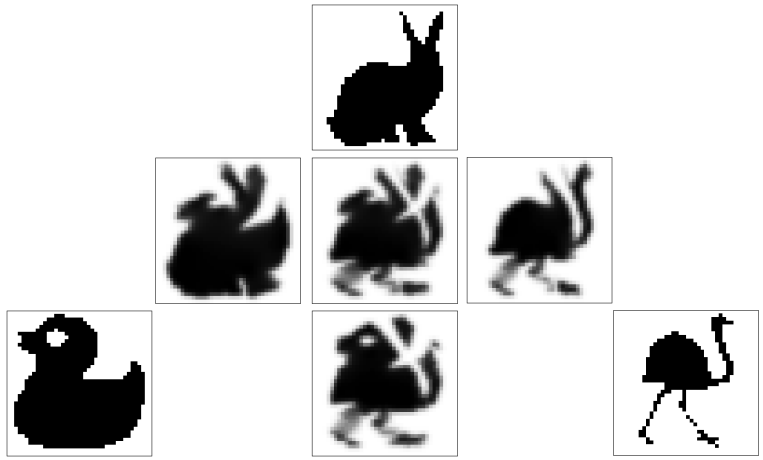
- Generalized Sinkhorn algorithm for barycenter computation:
  - $P^{(l)}(D, \lambda) = \prod_{s=1}^{S} \left(K^\top a_s^{(l)} \right)^{\lambda_s}$
  - $a_s^{(l)} = \frac{d_s}{K b_s^{(l-1)}}$
  - $b_s^{(l)} = \frac{P^{(l)}(D,\lambda)}{K^\top a_s^{(l)}}$
- Where $K = \exp(-C/\gamma)$

Wasserstein simplices

Effect of γ



(a) $\gamma = 8$

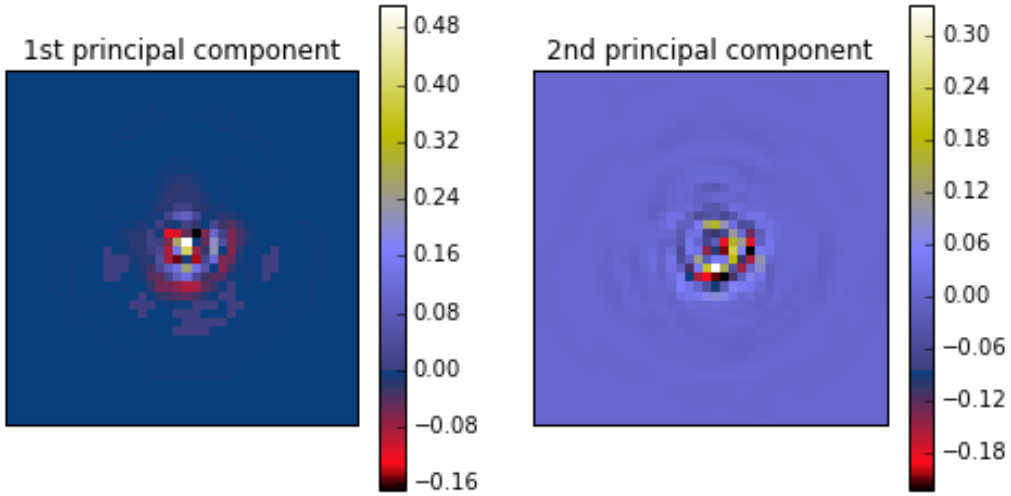


(b) $\gamma = 1$

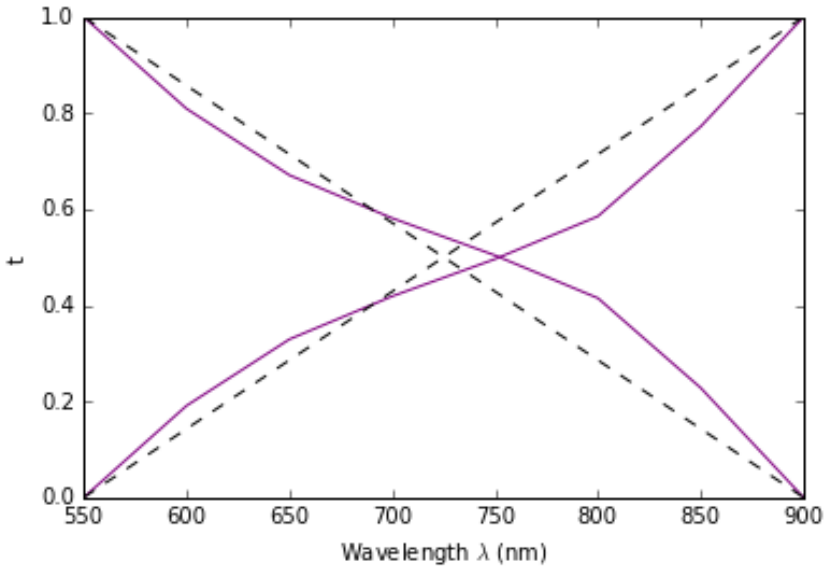
Figure: Wasserstein simplices for different γ values

Appendix

Chromatic variation of the PSF
PCA components



Chromatic variation of the PSF
Weights



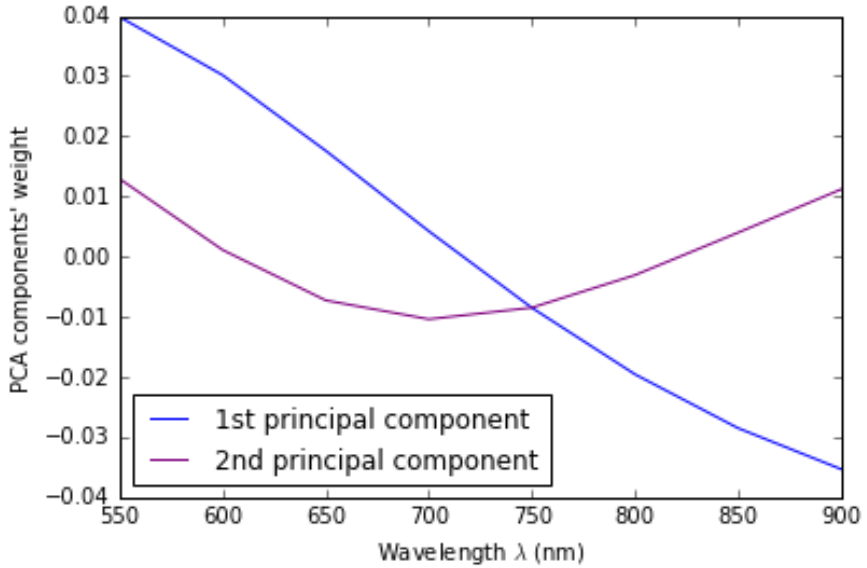
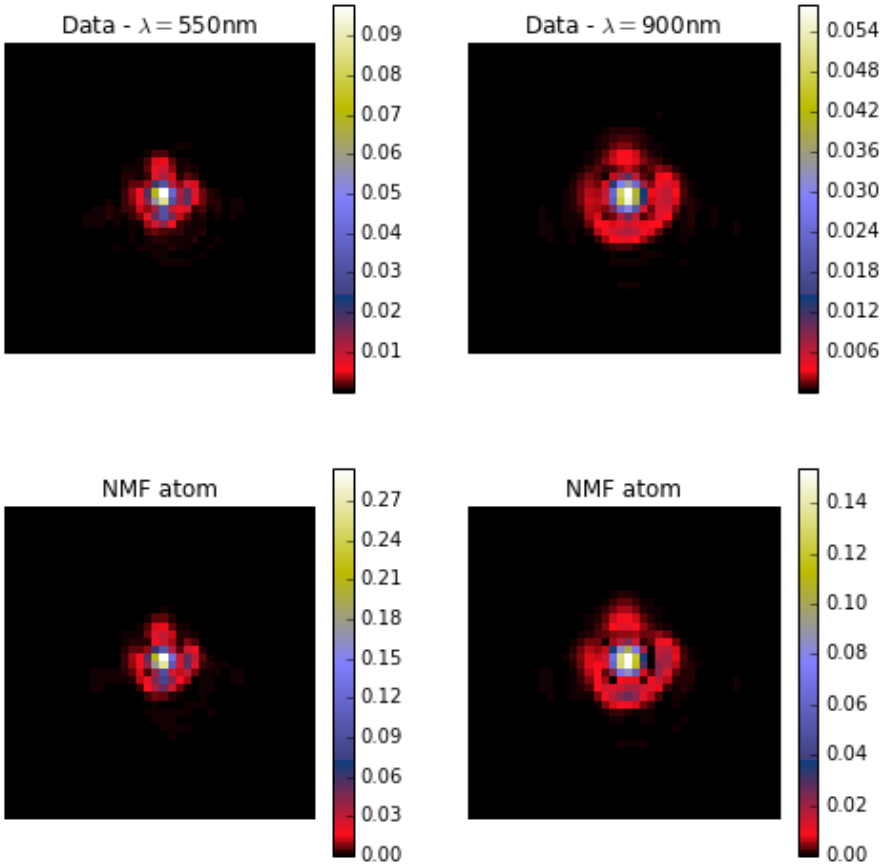


Figure: Left-hand side: weights reached by our method at convergence. Right-hand side: PCA-learned codes corresponding to the first two principal components

Chromatic variation of the PSF
NMF atoms

Back to PSF Application



Cardiac sequence

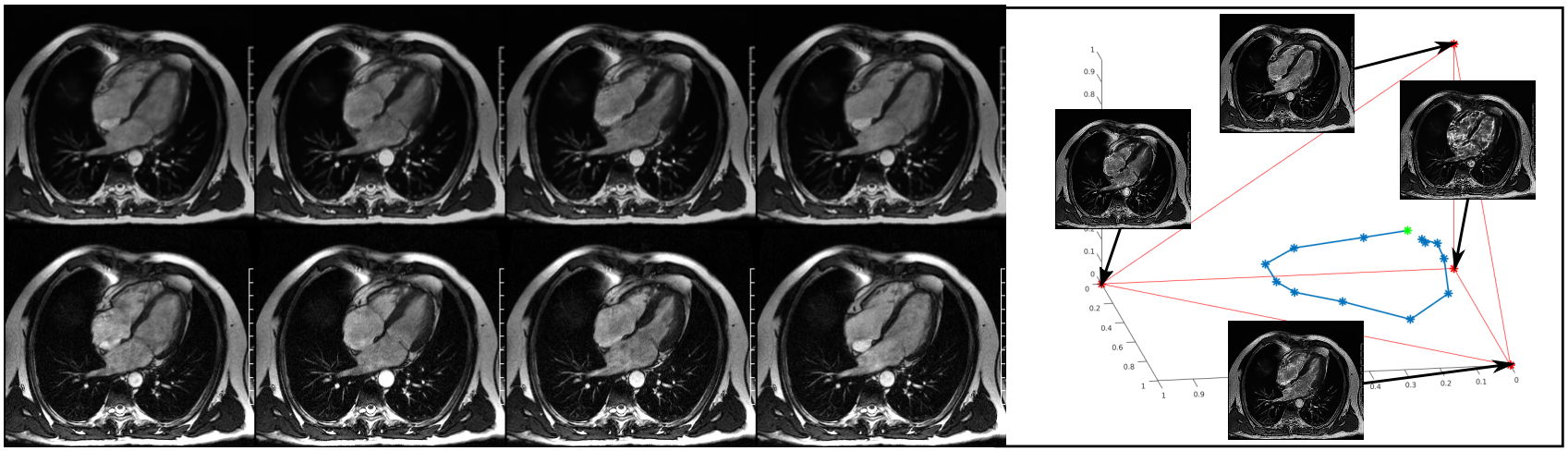


Figure: Left: True frames (bottom row) and reconstructions after application of our method. Right: Projection of the weights in the 4 atom basis.

Face editing
MUG Dataset

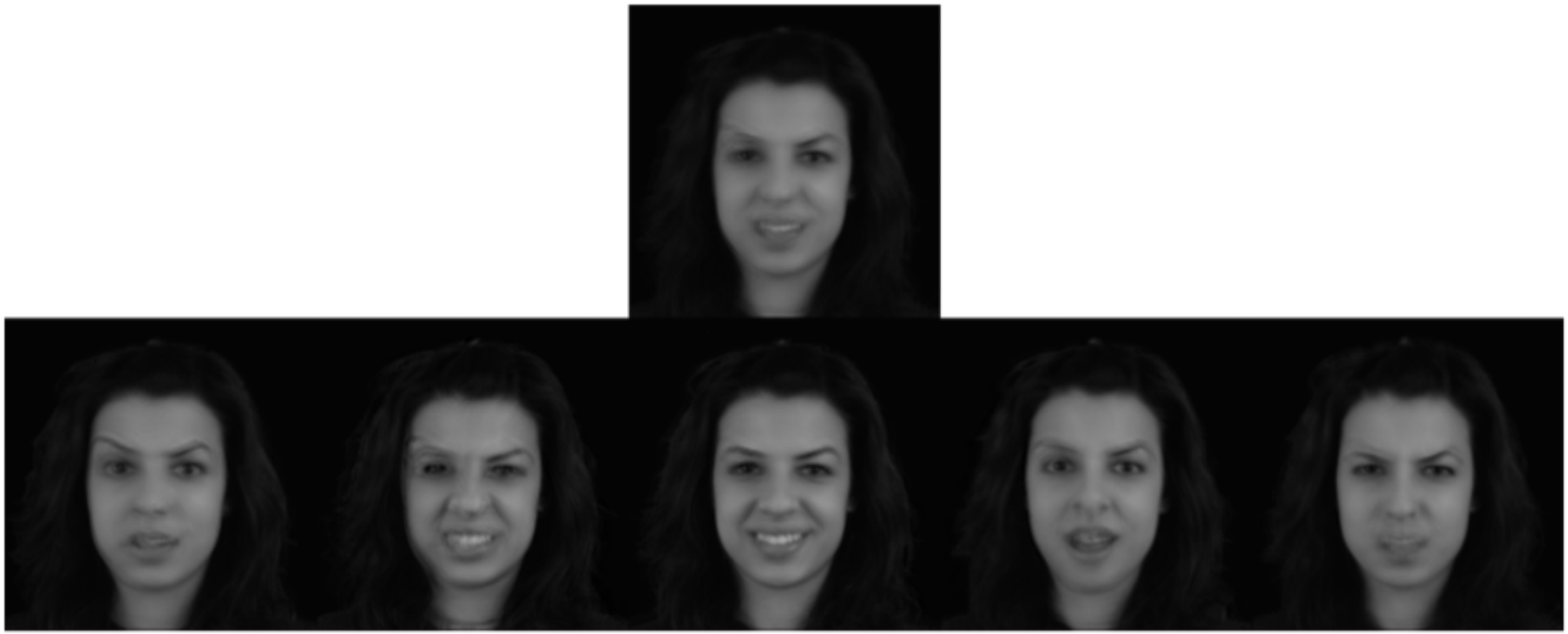


Figure: Top row: isobarycenter of 5 learnt atoms. Bottom row: extrapolation giving 30% weight importance to each of the 5.

Appendix

Wasserstein Dictionary Learning

Automatic differentiation

- Differentiating true P operator impossible and/or comes at prohibitive cost
- $P^{(L)}$ is what we are going to use in practice, and is just the result of iteratively applying simple mathematical operations:

$$P^{(L)} = f(f(\ldots(f(\mathbb{1}_N, D, \lambda)), D, \lambda), D, \lambda)$$

- Automatic differentiation[4]: differentiate *algorithm* by applying chain rule
- Can be done by specialized libraries[5] or ▸ 'by hand'

[4] Griewank & Walther, 2008

[5] http://deeplearning.net/software/theano/

Handmade automatic differentiation
Definitions

$$P^{(l)}(D,\lambda) = \Psi(b^{(l-1)}(D,\lambda), D, \lambda) \quad (1)$$

$$b^{(l)}(D,\lambda) = \Phi(b^{(l-1)}(D,\lambda), D, \lambda) \quad (2)$$

where:

$$\Psi(b, D, \lambda) := \prod_s \left(K^\top \frac{d_s}{K b_s} \right)^{\lambda_s}$$

$$\Phi(b, D, \lambda) := \left[\left(\frac{\Psi(b, D, \lambda)}{K^\top \frac{d_1}{K b_1}} \right)^\top, \ldots, \left(\frac{\Psi(b, D, \lambda)}{K^\top \frac{d_S}{K b_S}} \right)^\top \right]^\top$$

Handmade automatic differentiation
Notations

- Introduce:

$$\xi_y^{(l)} := \left[\partial_y \xi(b^{(l)}, D, \lambda)\right]^\top \qquad B_y^{(l)} := \left[\partial_y b^{(l)}(D, \lambda)\right]^\top$$

  where ξ can be Ψ or Φ, y can be D or λ

- And denote:

$$v^{(L-1)} := \Psi_b^{(L-1)}\left(\nabla L(P^{(L)}(D, \lambda), x)\right) \tag{3}$$

$$\forall l < L-1, v^{(l-1)} := \Phi_b^{(l-1)}\left(v^{(l)}\right) \tag{4}$$

Handmade automatic differentiation
Derivations

Then by total differentiation and the chain rule, differentiating (1) yields:

$$\left[\partial_D P^{(l)}(D,\lambda)\right]^\top = \Psi_D^{(l-1)} + B_D^{(l-1)}\Psi_b^{(l-1)} \tag{5}$$

And, differentiating (2):

$$B_D^{(l)} = \Phi_D^{(l-1)} + B_D^{(l-1)}\Phi_b^{(l-1)} \tag{6}$$

Handmade automatic differentiation
Derivations

We then have, by definitions (3)-(4) and by (5) and (6):

$$\begin{aligned}\nabla_D \mathcal{E}_L(D,\lambda) &= \Psi_D^{(L-1)}\left(\nabla\mathcal{L}(P^{(L)}(D,\lambda),x)\right) + B_D^{(L-1)} v^{(L-1)} \\ &= \Psi_D^{(L-1)}\left(\nabla\mathcal{L}(P^{(L)}(D,\lambda),x)\right) + \Phi_D^{(L-2)}\left(v^{(L-1)}\right) + \\ &\qquad B_D^{(L-2)}\left(v^{(L-2)}\right) \\ &= \dots \\ \nabla_D \mathcal{E}_L(D,\lambda) &= \Psi_D^{(L-1)}\left(\nabla\mathcal{L}(P^{(L)}(D,\lambda),x)\right) + \sum_{l=0}^{L-2} \Phi_D^{(l)}\left(v^{(l+1)}\right)\end{aligned}$$

where the sum starts at 0 because $B_D^{(0)} = 0$ since we initialized $b^{(0)}$ as a constant vector

Appendix

Going further: unbalanced WDL
Overview

- Recall discrete Wasserstein distance definition:

$$W(p,q) := \min_{T \in \Pi(\mu,\nu)} \langle T, C \rangle$$

$$\Pi(\mu,\nu) := \left\{ T \in \mathbb{R}_+^{N \times N}, T\mathbb{1}_N = \mu, T^\top \mathbb{1}_N = \nu \right\}$$

- Identical to:

$$W(p,q) := \min_{T \in \mathbb{R}_+^{N \times N}} \langle T, C \rangle + \iota_{\{\mu\}}(T\mathbb{1}_N) + \iota_{\{\nu\}}(T^\top \mathbb{1}_N)$$

- Unbalanced Optimal Transport: relaxation by replacing ι with other similarity criterion, *e.g.*:

$$W_{\mathrm{KL}}(p,q) := \min_{T \in \mathbb{R}_+^{N \times N}} \langle T, C \rangle + \mathrm{KL}\left(\mu | T\mathbb{1}_N\right) + \mathrm{KL}\left(\nu | T^\top \mathbb{1}_N\right)$$

Going further: unbalanced WDL
Application

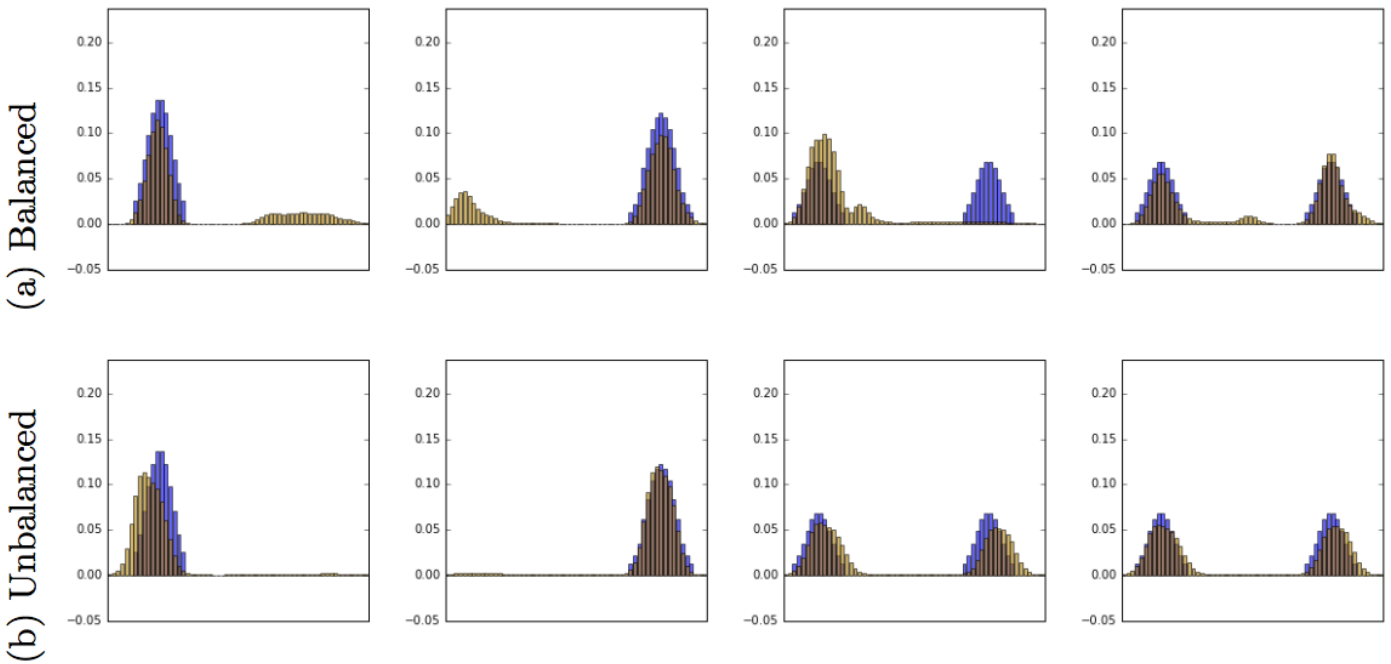


Figure: Example datapoints from set of bimodal Gaussians (in blue) and reconstructions from our method (in yellow) in both the balanced and the unbalanced cases