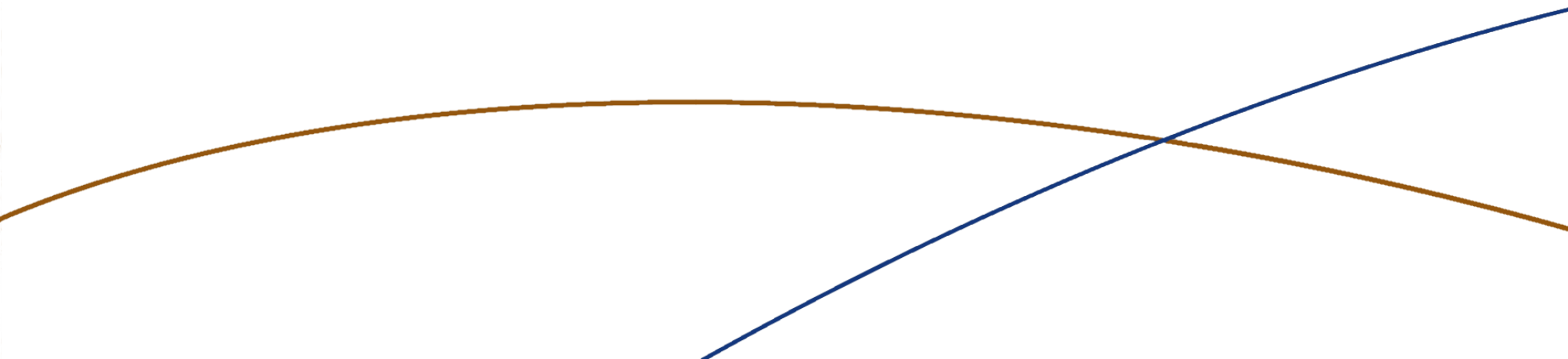
Optimal mass transport
a brief introduction

Filip Elvander

Dept. Information and Communications Engineering

Aalto University

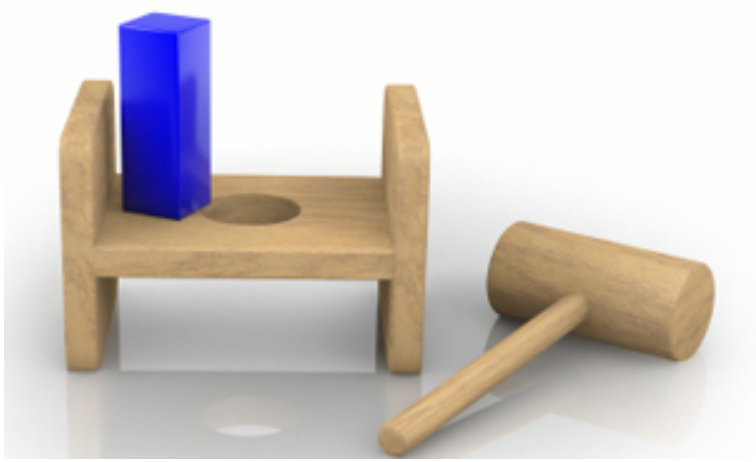


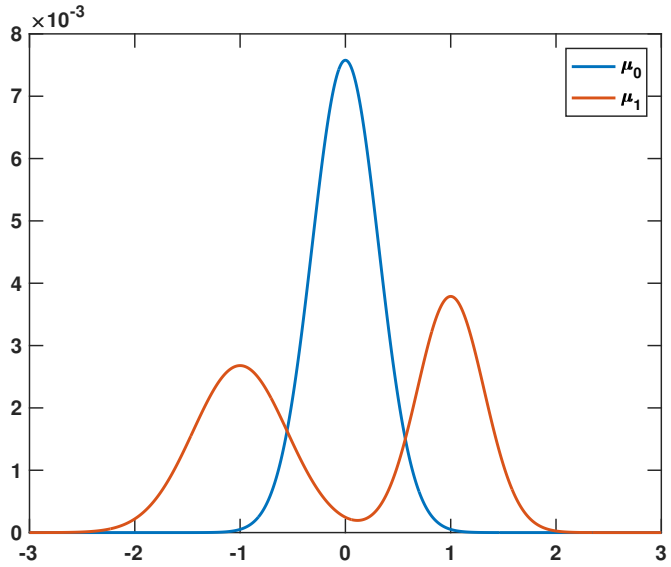
In this talk

- Brief introduction to optimal mass transport (OMT)
  - Motivation and historical background
  - Mathematical description
- Applications in signal processing
  - Solving inverse problems
  - Source localization and tracking
  - Interpolation for interference cancellation
- Computational tools for handling large(ish)-scale problems
- Alternative names for OMT: Earth mover's distance, Wasserstein loss (common in machine learning).

Why optimal mass transport (OMT)?

- All scientific frameworks need a way to measure distances
  - How well does my model fit my data?
  - How do I fit a model to empirical measurements in the first place?
- How should we compare mass distributions?





Where do mass distributions show up?

- Statistics: non-parametric inference.
- Signal processing: power spectra over frequency and/or space.
- Machine learning: distribution of data and network output.
- Optimal and stochastic control: ensemble steering
- ...

Standard distances

Consider two densities Φ_0 and Φ_1 over $\mathbb{R}$. Two common ways of comparing p and q are

$$L_p: \quad \left(\int_{\mathbb{R}} |\Phi_0(x) - \Phi_1(x)|^p \, dx \right)^{1/p}, \; p \in [1, \infty)$$

$$\text{Kullback-Leibler divergence}: \quad \int_{\mathbb{R}} \Phi_0(x) \log \left(\frac{\Phi_0(x)}{\Phi_1(x)} \right) dx.$$

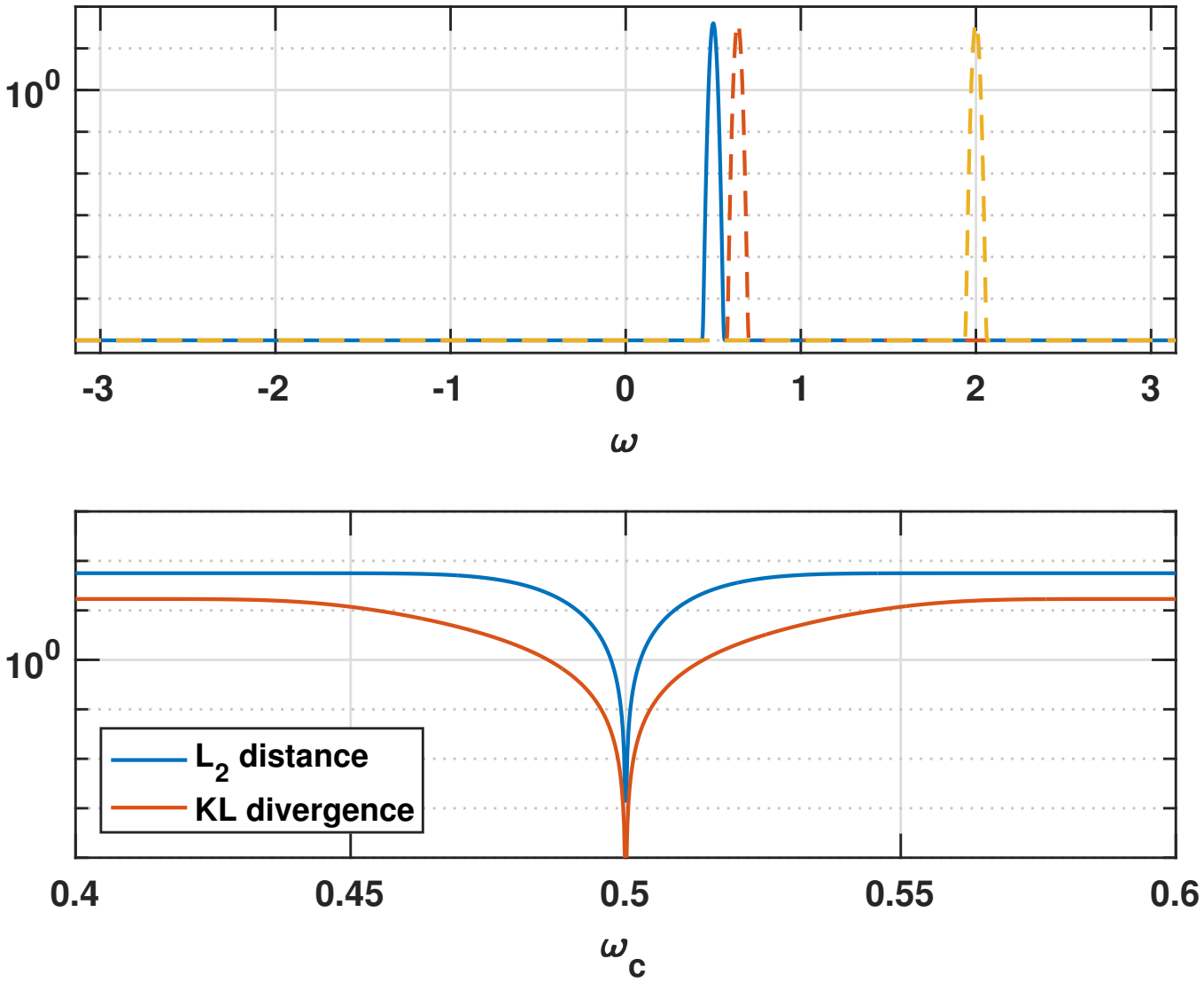
If discrete measures (histograms):

$$\ell_p: \quad \|\boldsymbol{\varphi}_0 - \boldsymbol{\varphi}_1\|_p^{1/p} = \left(\sum_{k=1}^{n} |[\boldsymbol{\varphi}_0]_k - [\boldsymbol{\varphi}_1]_k|^p \right)^{1/p}$$

$$\text{KL}: \quad \sum_{k=1}^{n} [\boldsymbol{\varphi}_0]_k \log \left(\frac{[\boldsymbol{\varphi}_0]_k}{[\boldsymbol{\varphi}_1]_k} \right)$$

- What happens with the distances as the supports of Φ_0 and Φ_1 become disjoint?
- How about higher dimensions?

Shifting concentrated distributions



Problems

Modeling:

- Is it meaningful to compare distribution point-by-point?
- What are the implications for computing ”average” distributions, clustering, and so on?

Algorithms and estimation:

- If the distance saturates, its gradient vanishes. How do you make progress towards finding minima of your cost function?
- Implications for classical statistical estimation as well as for machine learning

Optimal mass transport

Historical background

- Gaspard Monge (1746-1818), French mathematician
- High posts in society post-revolution, and close friend of Napoleon.
- How can building material most efficiently be distributed to construction sites?
- Military applications: supply lines. Research kept secret during the Napoleonic wars.

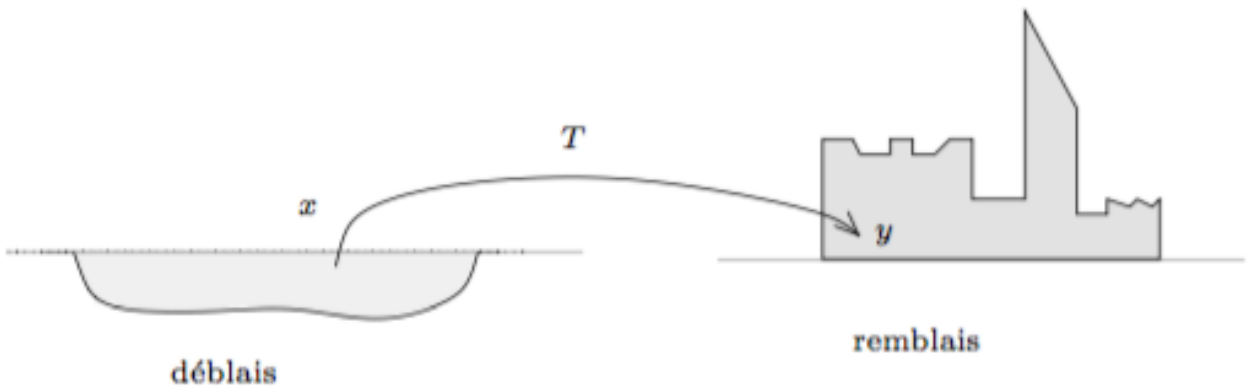


Fig. 3.1. Monge's problem of déblais and remblais

Image from C. Villani, *Optimal Transport - Old and New* (Springer, 2009)

Optimal mass transport: ingredients

- Two spaces, $\mathcal{X}$, $\mathcal{Y}$, and two distributions $\mu_X \in \mathcal{M}_+(\mathcal{X})$ and $\mu_Y \in \mathcal{M}_+(\mathcal{Y})$.
- What is the cheapest way of rearranging μ_X to μ_Y?
- The effort of rearrangement is measured in terms of a "ground cost", $c : \mathcal{X} \times \mathcal{Y} \to \mathbb{R}_+$.
- Example: $\mathcal{X} = \mathcal{Y} = \mathbb{R}^3$ and $c(x,y) = \|x-y\|_2^2$.

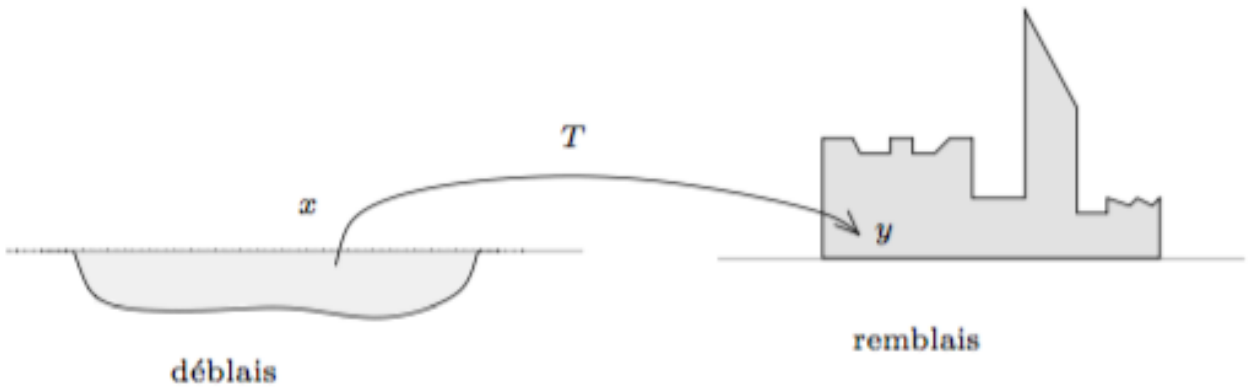


Fig. 3.1. Monge's problem of déblais and remblais

Image from C. Villani, *Optimal Transport - Old and New* (Springer, 2009)

Optimal mass transport: Monge

Define the distance between μ_X and μ_Y as

$$S(\mu_X, \mu_Y) \triangleq \min_{T:\mathcal{X}\to\mathcal{Y}} \int_{\mathcal{X}} c(x, T(x)) d\mu_X(x) \text{ , s.t. } T_{\#}\mu_X = \mu_Y .$$

- $x \in \mathcal{X}$ is mapped to $T(x) \in \mathcal{Y}$ with ground cost $c(x, T(x))$.
- The amount of mass at x is $d\mu_X(x)$.
- T rearranges μ_X into μ_Y ($T_{\#}\mu_X = \mu_Y$).
- Drawbacks: (1) difficult nonlinear problem, (2) Cannot split masses

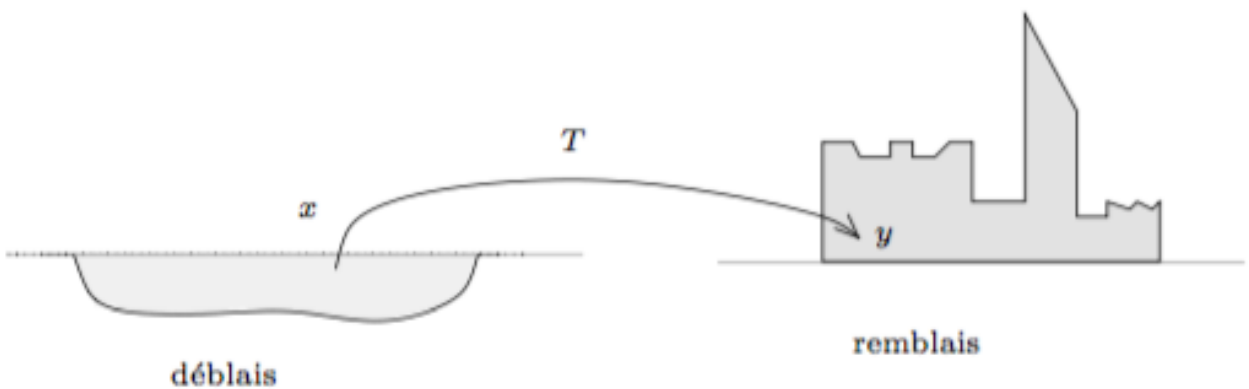
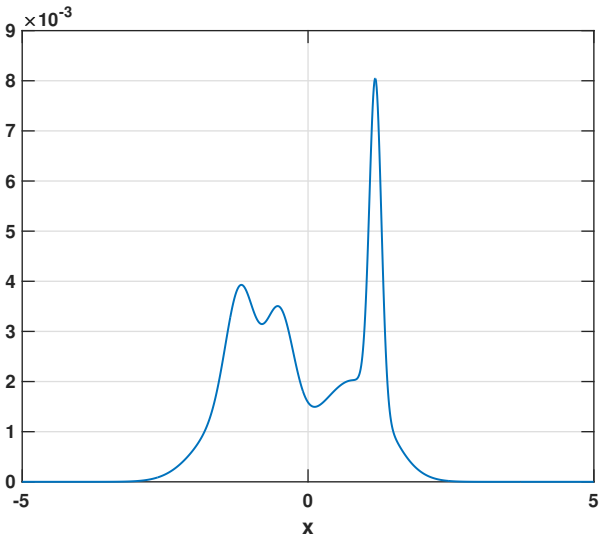


Fig. 3.1. Monge's problem of déblais and remblais

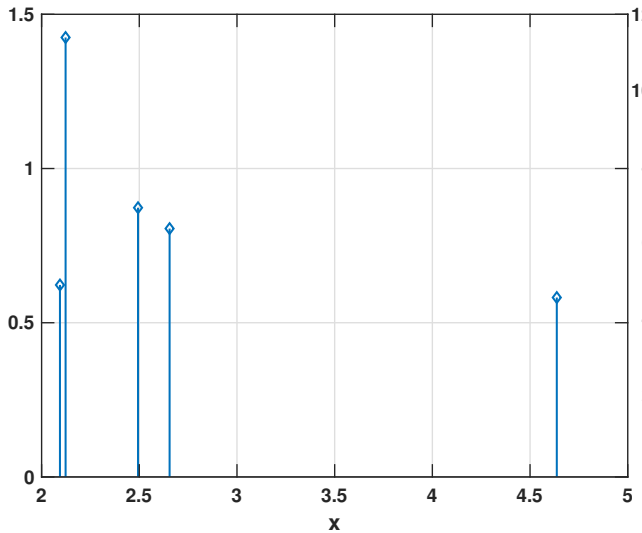
Image from C. Villani, *Optimal Transport - Old and New* (Springer, 2009)

Different types of distributions

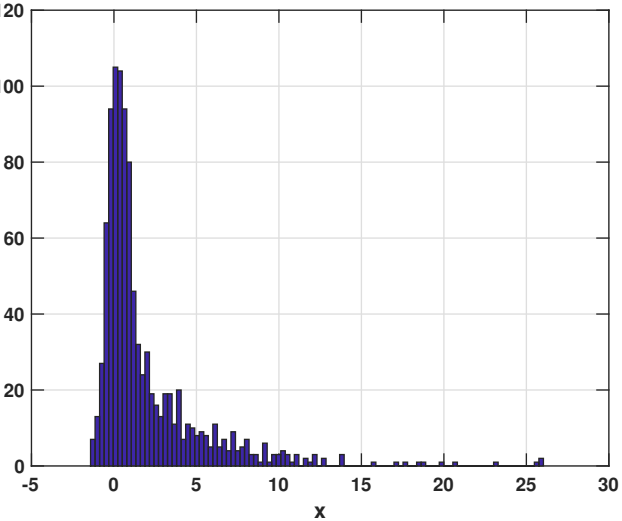
- Discrete distributions: what happens if the cardinalities do not match (different numbers of ”points”)?
- How to transport between densities and discrete measures?



Density.



Discrete.



Empirical (histogram).

Relaxing the problem (Kantorovich)

Next breakthrough: 1940s by Soviet mathematician Leonid Kantorovich.

- One of the founding fathers of linear programming.
- Nobel Memorial Prize in Economics for contributions in operations research.
- Findings unknown for a long time in the West due to Cold War secrecy.

Key idea:

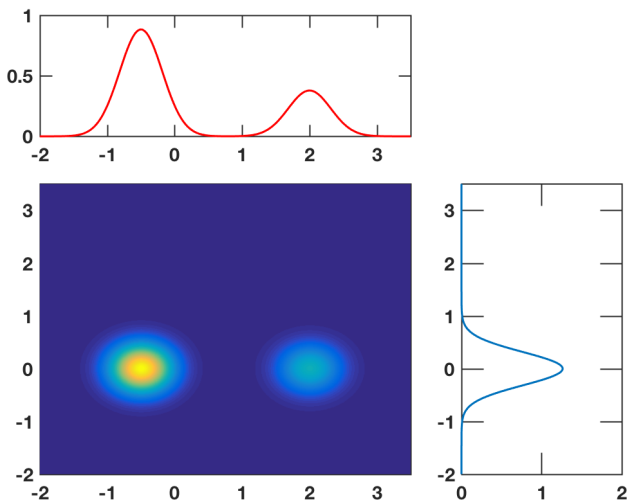
- If $X \sim d\mu_X$, and $Y \sim d\mu_Y$ are random variables, then $Y = T(X)$. The relationship is deterministic.
- What if we just require X and Y to have a joint distribution M instead? That is, the relationship between X and Y is stochastic.

Kantorovich formulation

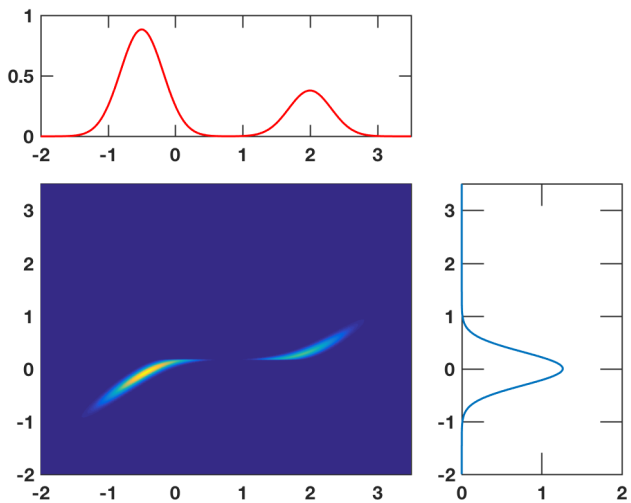
Instead of a **mapping** $T : \mathcal{X} \to \mathcal{Y}$, look for a **coupling** $M \in \mathcal{M}_+(\mathcal{X} \times \mathcal{Y})$:

$$\underset{M}{\text{minimize}} \int c(x,y)dM(x,y) \text{ , s.t. } \int dM(\cdot,y) = d\mu_X \text{ , } \int dM(x,\cdot) = d\mu_Y .$$

- Equivalent to Monge problem in many interesting cases.
- Allows for mass splitting: discrete measures not a problem.
- Convex problem!



$M = \mu_X \otimes \mu_Y$.



Minimizing M for $c(x,y) = (x-y)^2$.

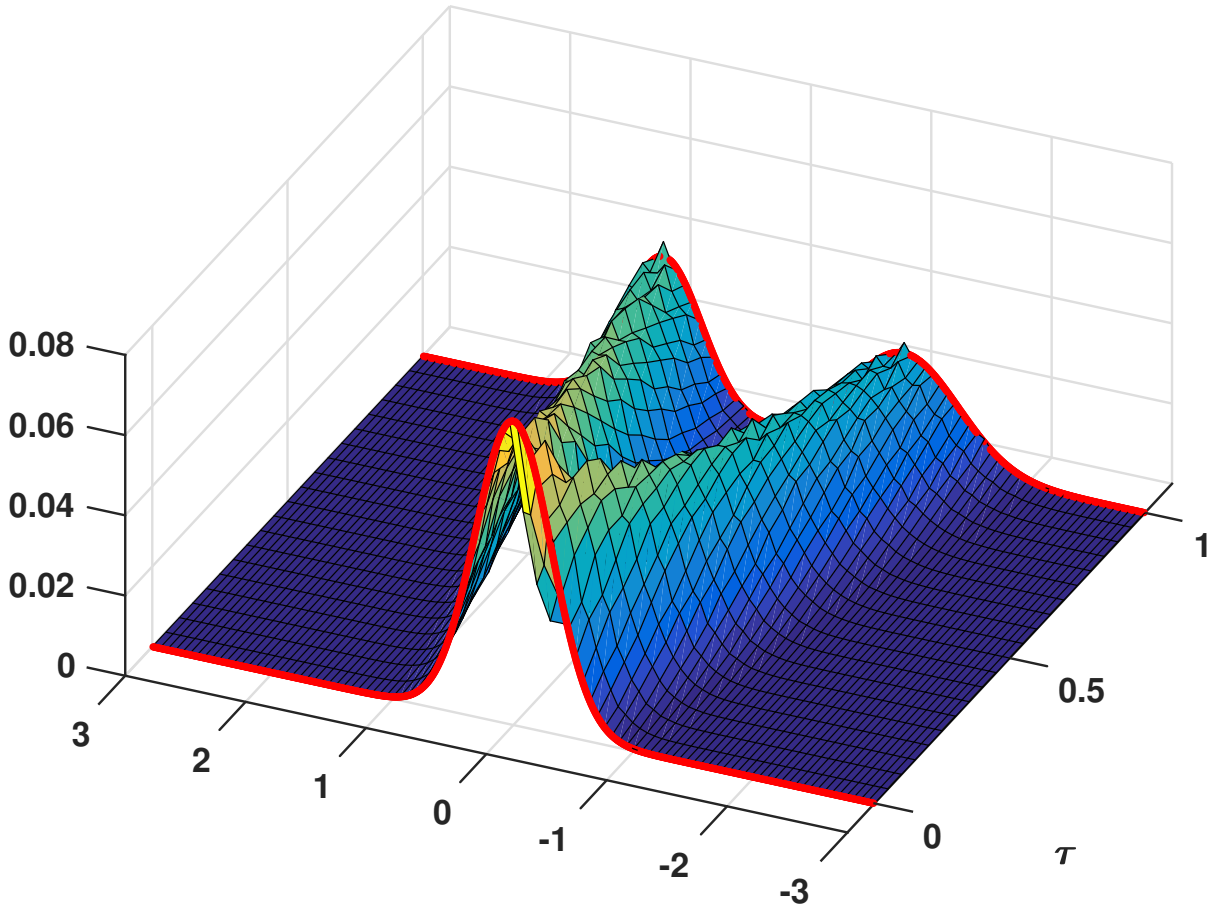
What does the OMT solution mean?

- $dM(x,y)$ is the amount of mass moved from x to y.
- Let's say that $c(x,y)$ is the cost of taking the optimal path from x to y.
- For an optimal M, the whole distribution moves in an optimal way.

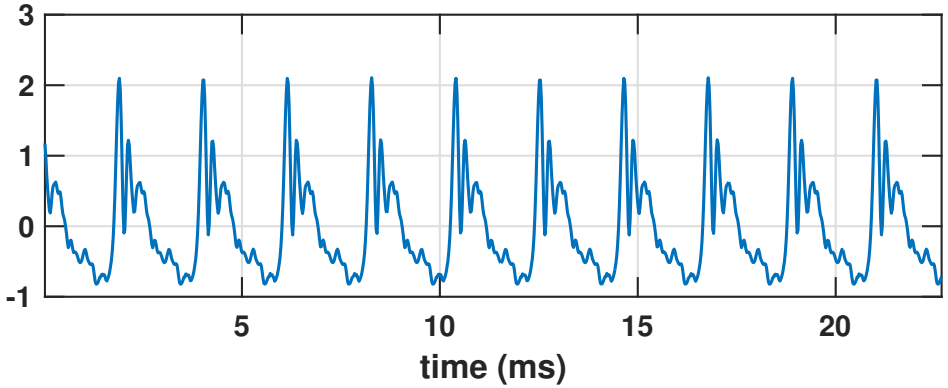
For example, $c(x,y) = \|x-y\|_2^2$ implies movement along lines $(1-t)x+ty$, for $t \in [0,1]$.

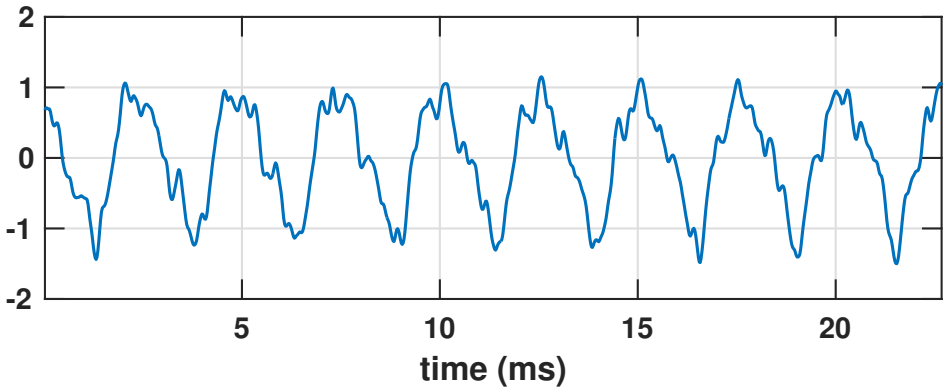
- In general, the key feature of OMT is that the space of distributions $\mathcal{M}_+(\mathcal{X})$ inherits properties of $\mathcal{X}$ via the "ground cost" c.
- Think of comparing a sequence of images pixel by pixel (L_2) or in terms of movement of objects (OMT).
- This means that we can interpolate between distributions or compute generalized averages ("barycenters"), depending on our application.

Interpolation of distributions

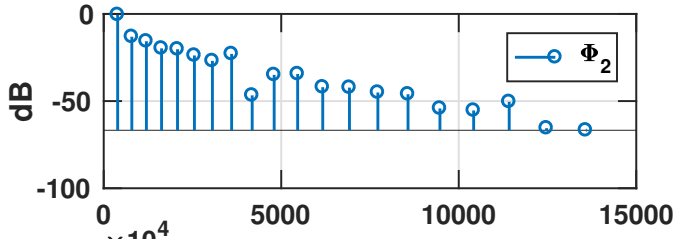


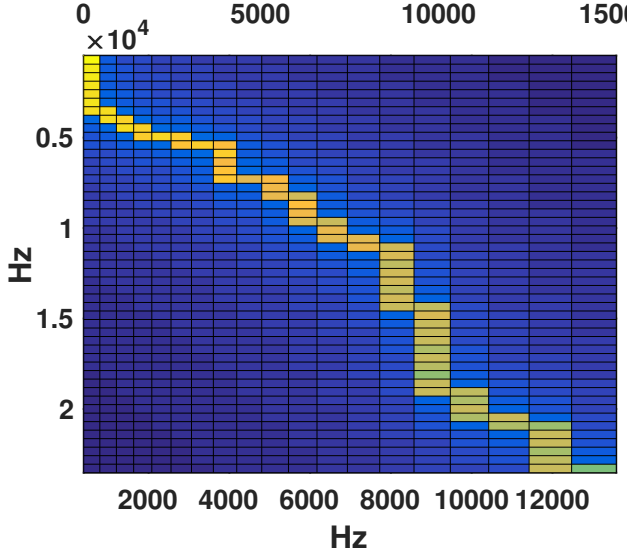
Sound interpolation

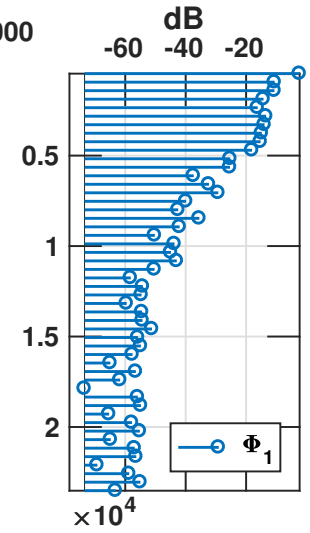




First Second

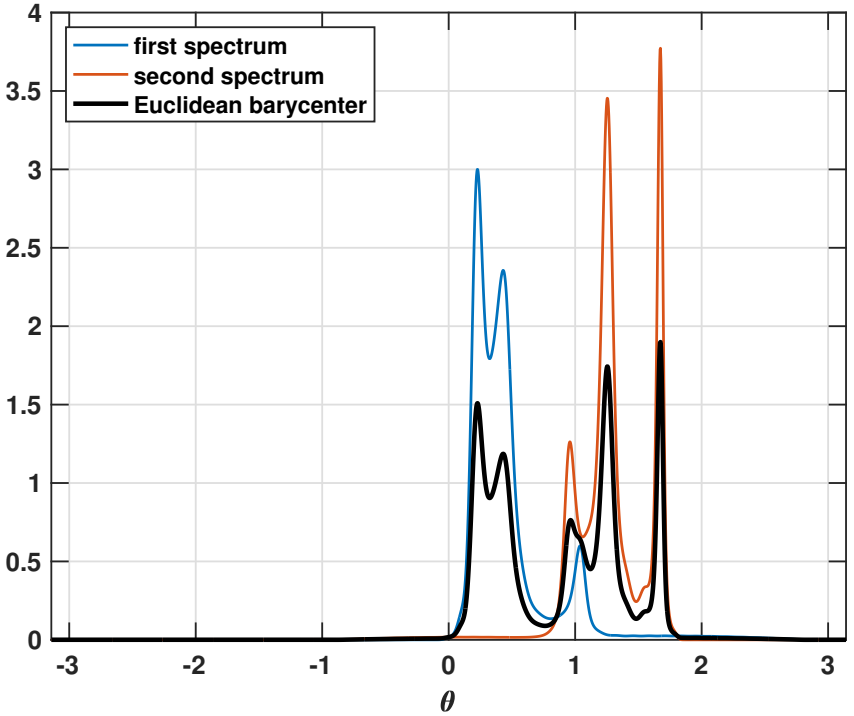




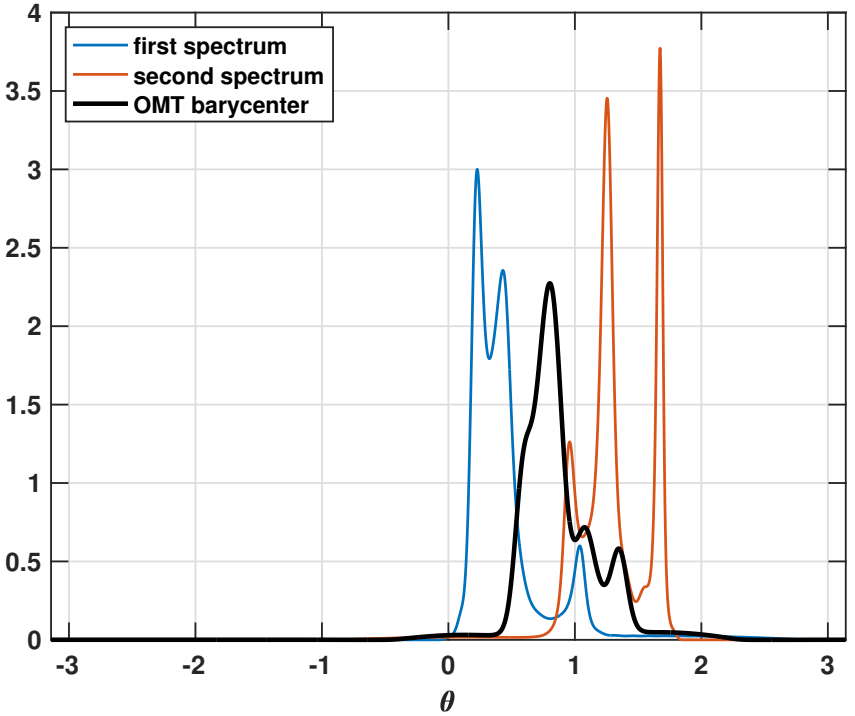


Play

Generalized averaging (barycenter)



Euclidean.



OMT barycenter.

Inverse problems

We can also use OMT when we do not have access to the distributions μ_X. Let Γ be a linear operator, and let $r_X = \Gamma(\mu_X)$.

$$\underset{M}{\text{minimize}} \int c(x,y)dM(x,y) \text{ , s.t. } \Gamma\left(\int dM(\cdot,y)\right) = r_x \text{ , } \Gamma\left(\int dM(x,\cdot)\right) = r_Y$$

- Induces a distance and a geometry for objects r_X and r_Y.
- Allows us to compare them using the ground cost c, as well as interpolate them.
- Example: $\Gamma(\mu_X)_k = \int e^{i\omega k} d\mu_X(\omega)$, i.e., r_X is a covariance sequence.
  - We can compare covariance sequences in terms of their frequency content.
  - Opens up the possibility for estimation, tracking, generalized averaging, and more.
  - Array processing, image registration, and more.

DoA estimation

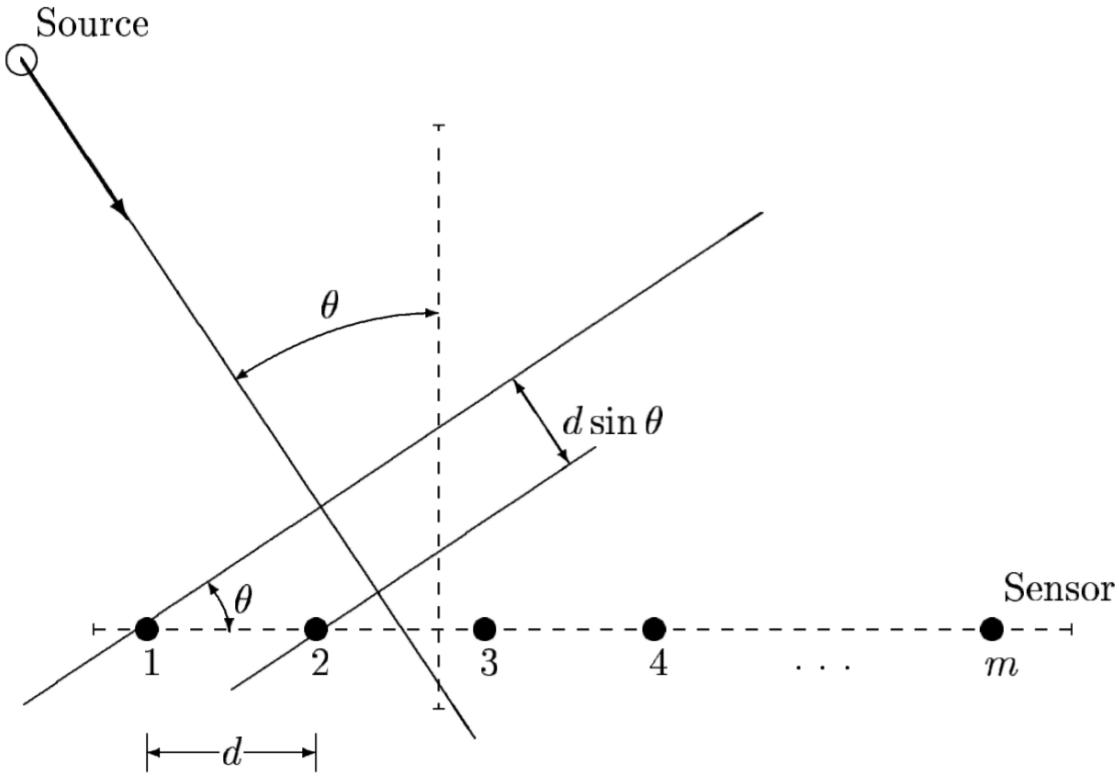
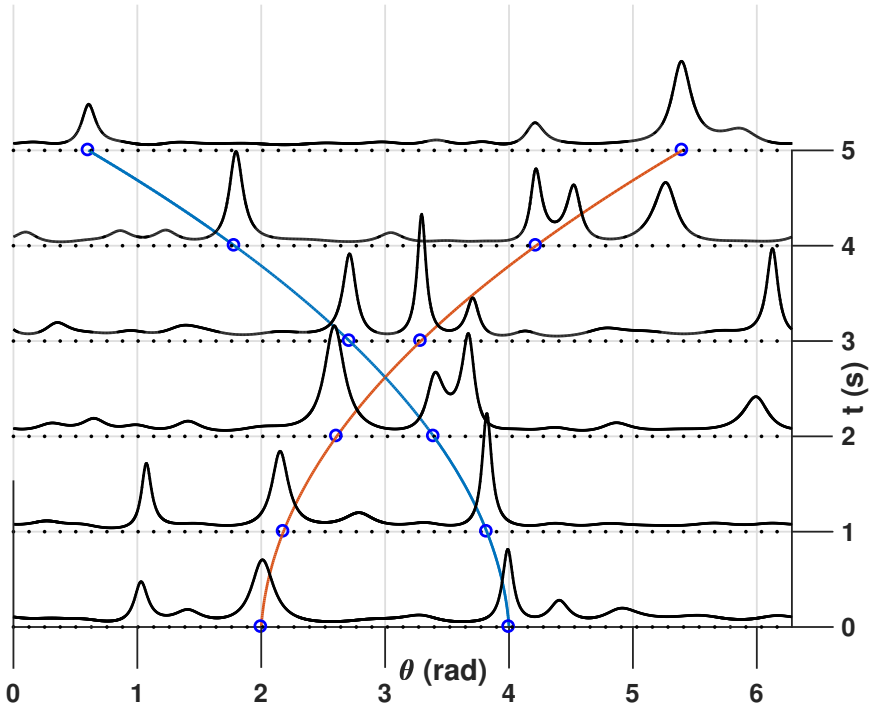


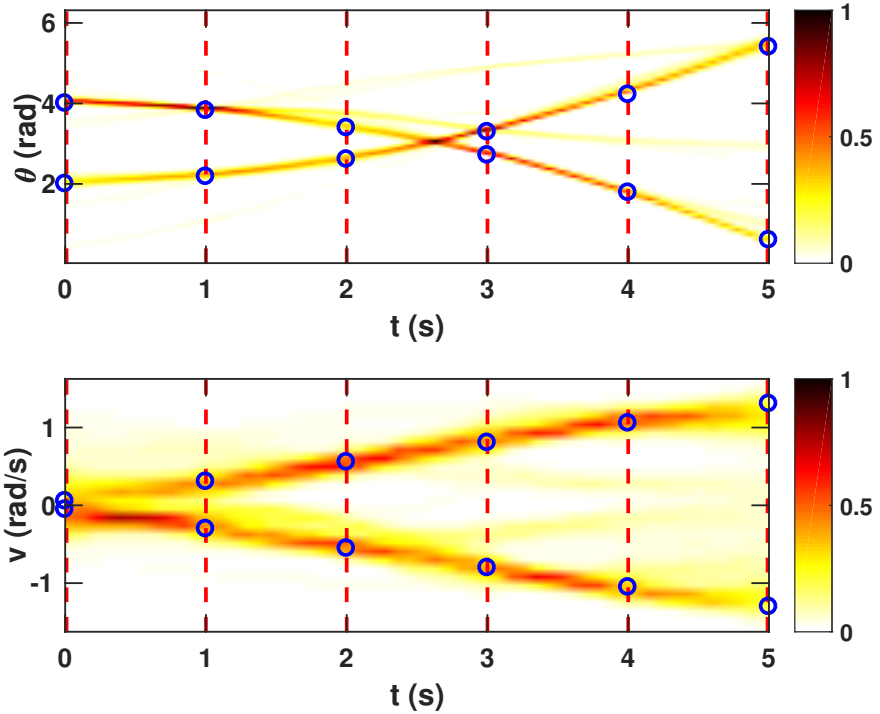
Figure 6.5. The uniform linear array scenario.

Image from Stoica and Moses, *Spectral analysis of signals* (Prentice Hall, 2005).

Tracking over time

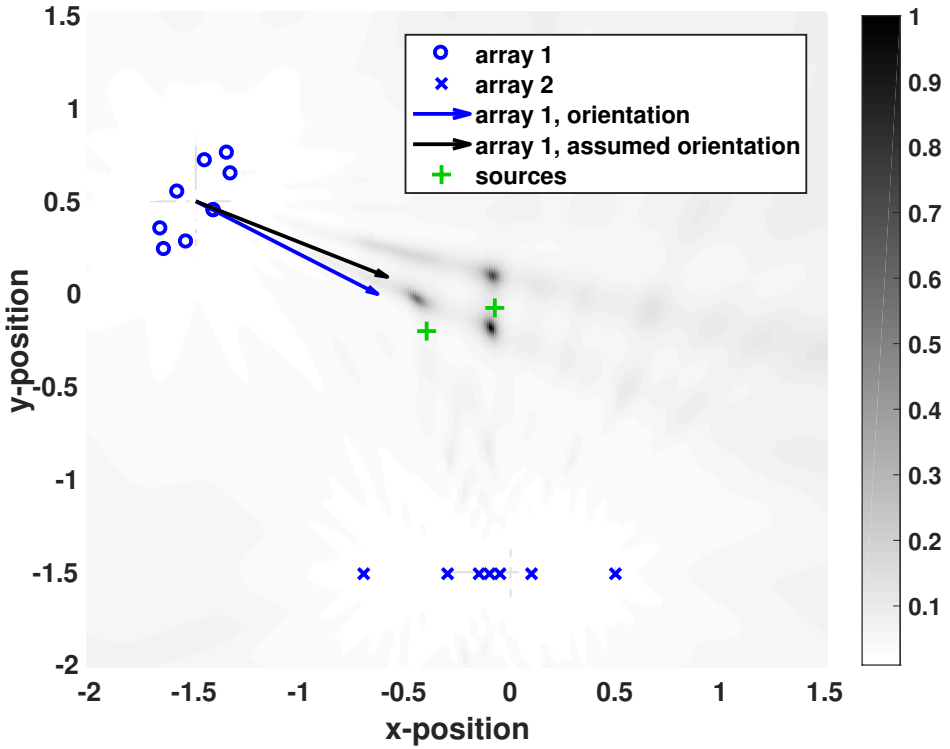


Point-wise MVDR.

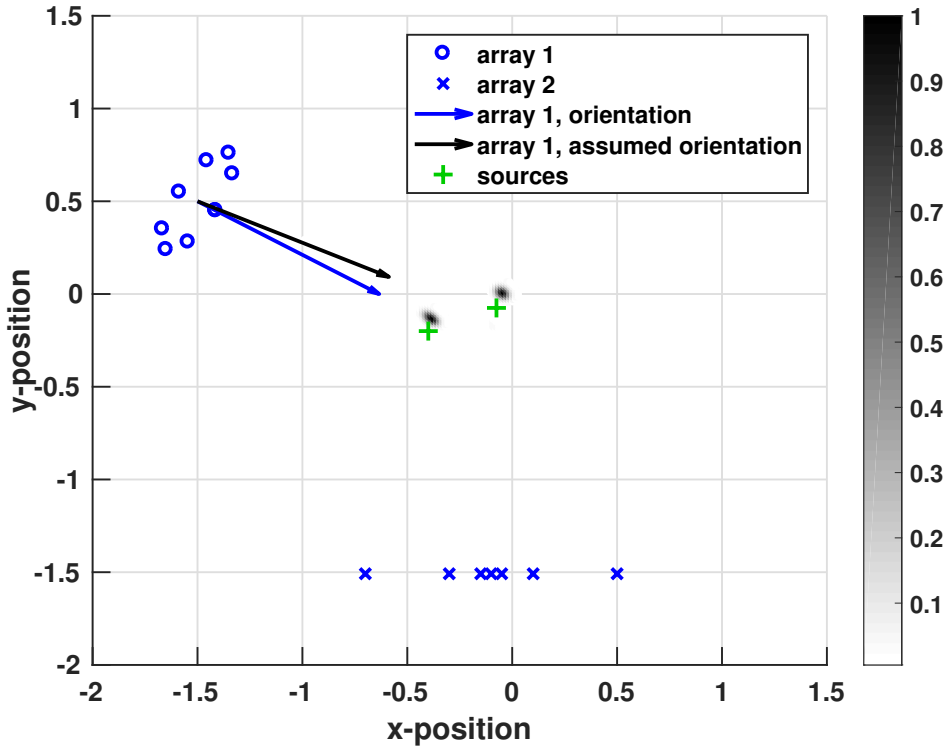


OMT estimate.

Fusing inconsistent data



MUSIC estimate



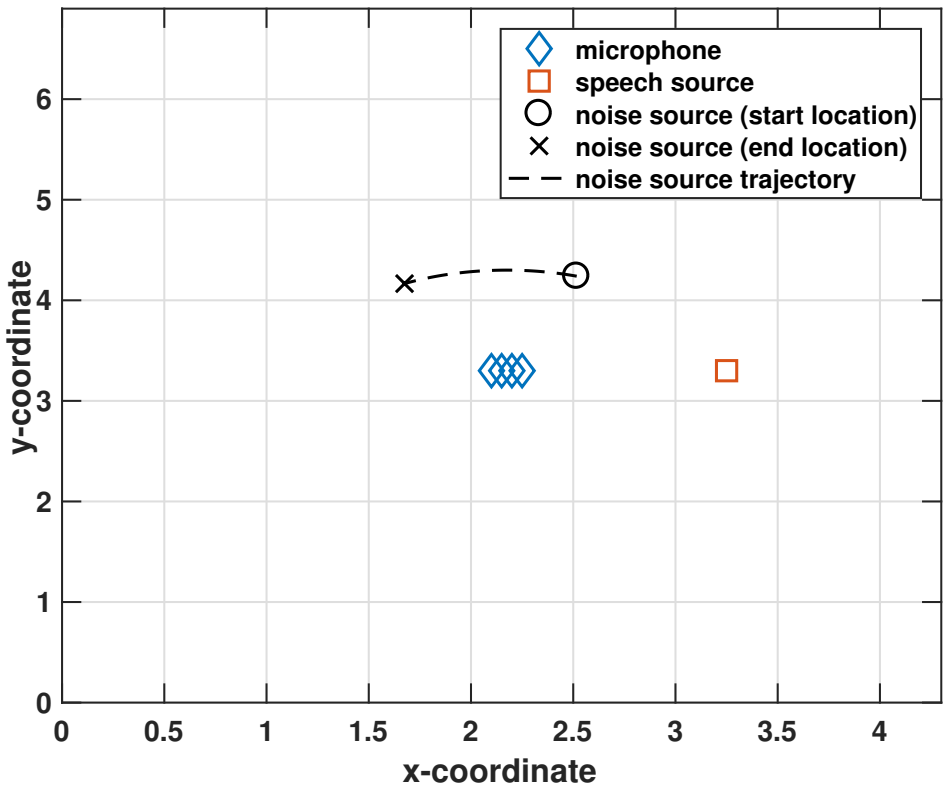
OMT barycenter

Noise reduction

- $\boldsymbol{y}(t) = \mathbf{h}(t)s(t) + \mathbf{n}(t)$,
- $\mathbf{R}(t) = \mathbb{E}\left(\mathbf{n}(t)\mathbf{n}(t)^H\right)$,
- Noise reduction using MVDR filter:

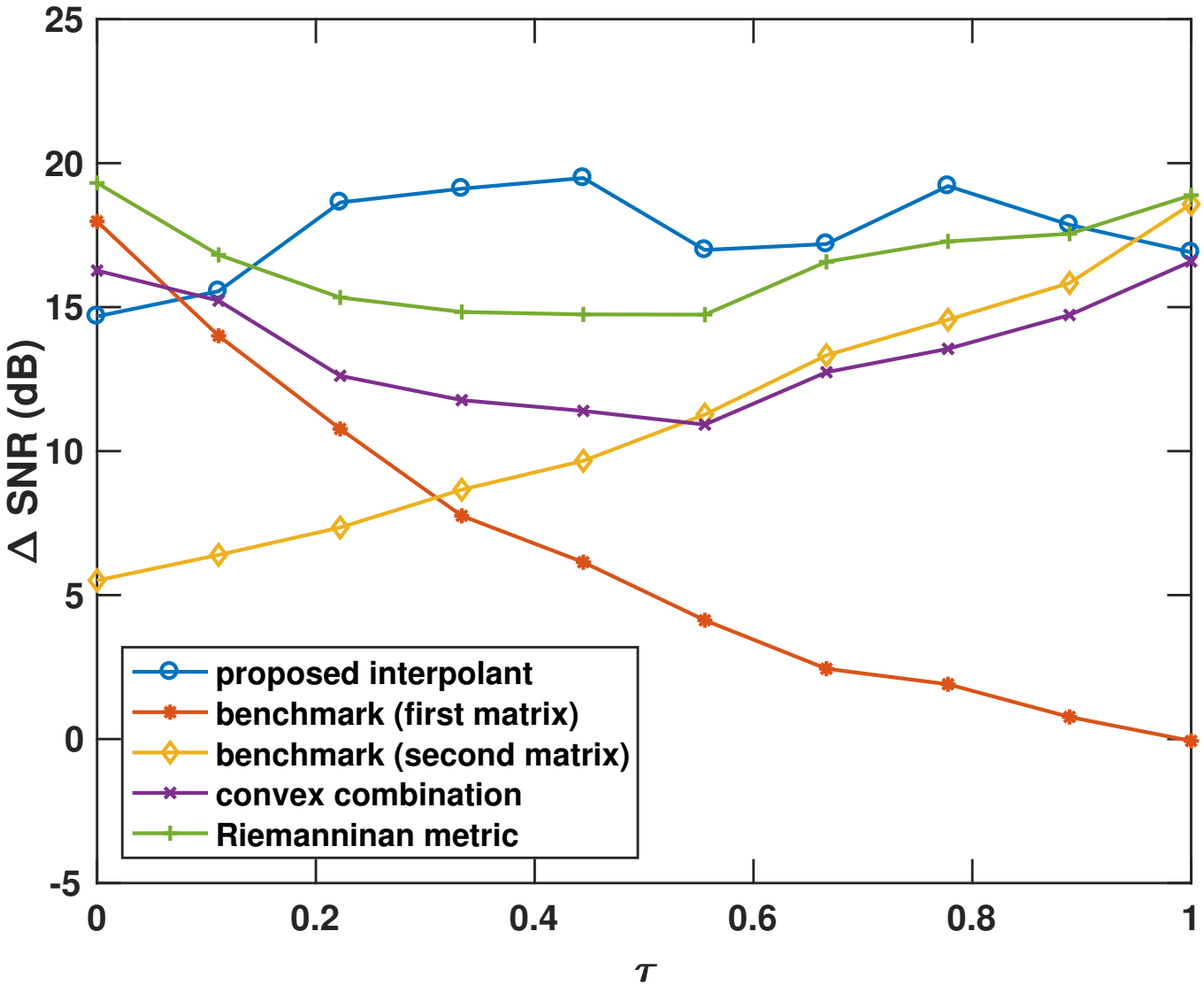
$$\mathbf{w}(t) = \arg\min_{\mathbf{w}} \mathbf{w}^H \hat{\mathbf{R}}(t)\mathbf{w}$$
$$\text{s.t. } \mathbf{w}^H \mathbf{h}(t) = 1$$

- Noise-only signal only at t_0 and t_1, and $\hat{\mathbf{R}}(t_0) \neq \hat{\mathbf{R}}(t_1)$.
- Interpolate $\hat{\mathbf{R}}(t)$ for $t \in (t_0, t_1)$ and compute corresponding filters.



Noise reduction

Original | No interpolation | Convex | OMT



Discretization

Ideally, we would like to solve

$$\underset{M}{\text{minimize}} \int c(x,y)dM(x,y) \text{ , s.t. } \int dM(\cdot,y) = d\mu_X \text{ , } \int dM(x,\cdot) = d\mu_Y .$$

However, in practice, we will have to discretize the problem:

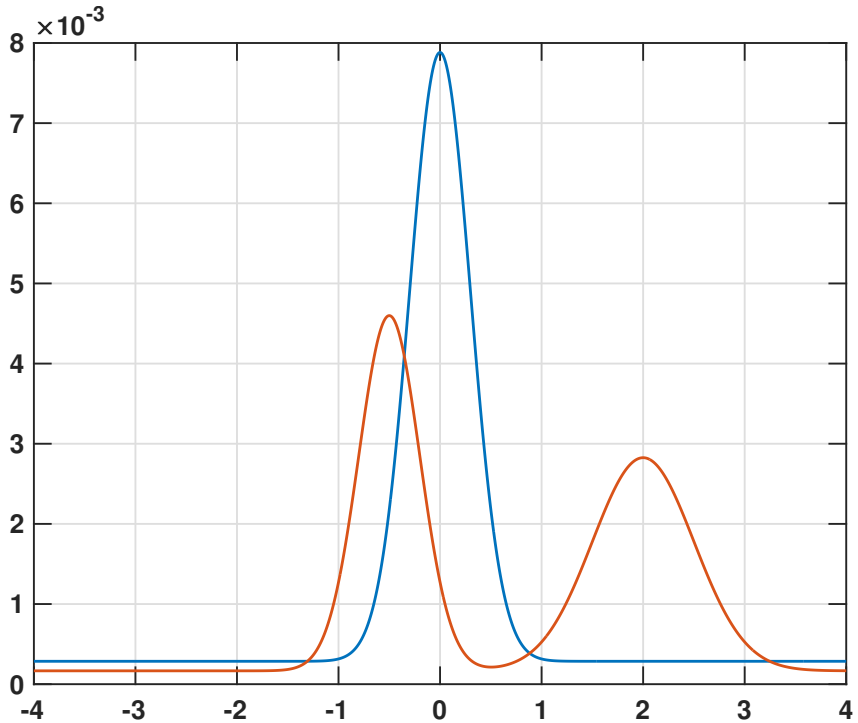
$$\mu_X \leftrightarrow \boldsymbol{\mu}_X \in \mathbb{R}_+^n \text{ , } \mu_Y \leftrightarrow \boldsymbol{\mu}_y \in \mathbb{R}_+^n$$

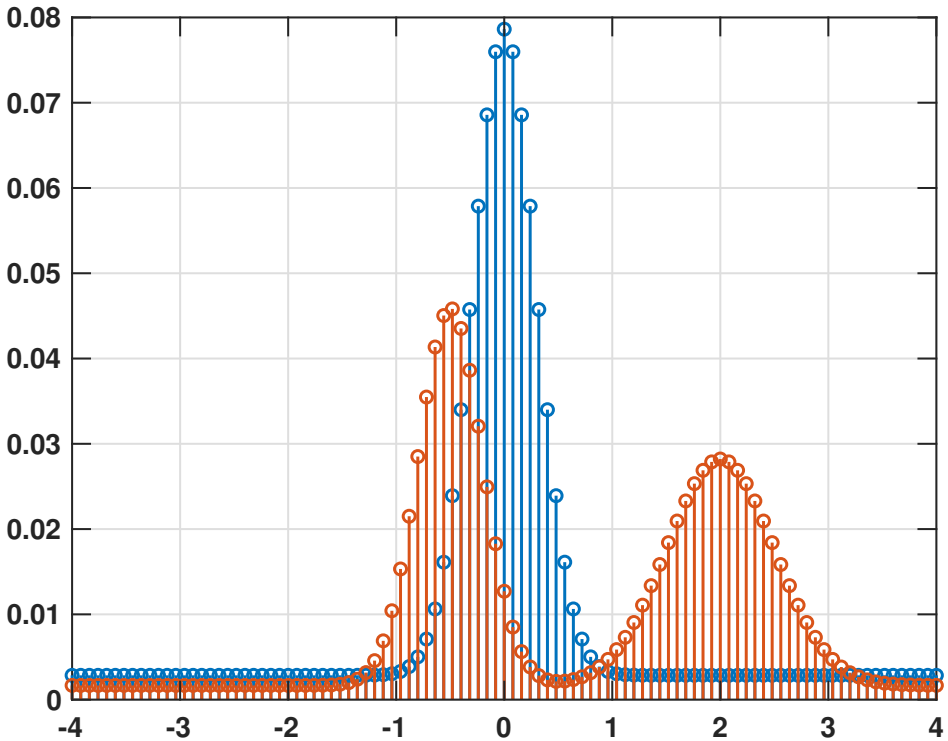
and solve

$$\underset{\mathbf{M}\in\mathbb{R}_+^{n\times n}}{\text{minimize}} \quad \text{trace}\left(\mathbf{C}^T\mathbf{M}\right) \text{ , s.t. } \mathbf{M1} = \boldsymbol{\mu}_X \text{ , } \mathbf{M}^T\mathbf{1} = \boldsymbol{\mu}_Y .$$

- $\text{trace}\left(\mathbf{C}^T\mathbf{M}\right) = \sum_{k,\ell} \mathbf{C}_{k,\ell}\mathbf{M}_{k,\ell}$
- $\mathbf{C}_{k,\ell} = c(x_k, y_\ell)$, cost matrix
- $\mathbf{1}$ vector of ones (length n).
- Standard linear program: can be very slow to solve, especially as dimension increases.

Discretization





Entropy regularization

In order get a better problem, consider the discrete entropy $D(\mathbf{M})$ defined as

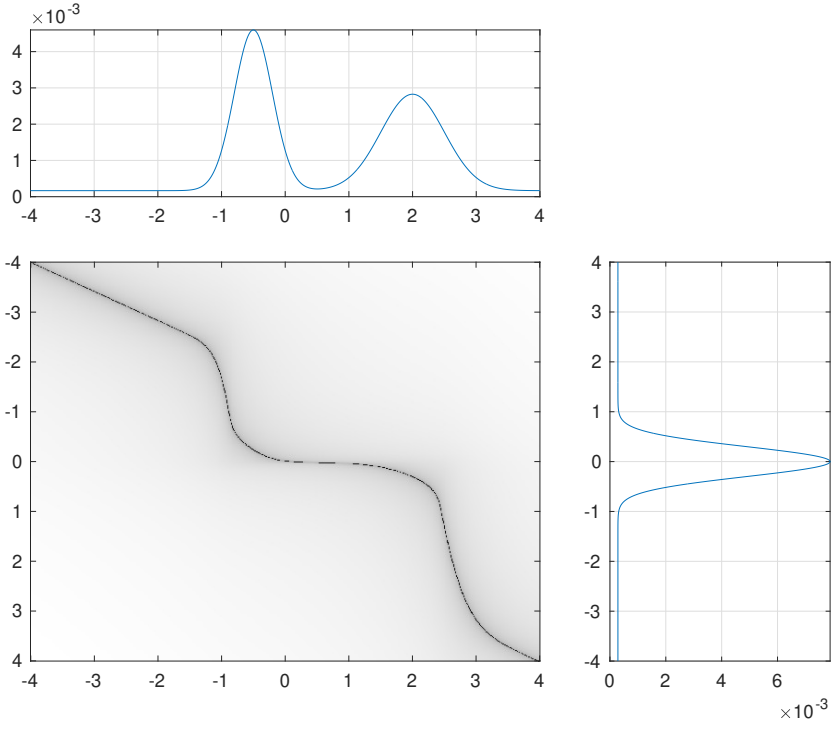
$$D(\mathbf{M}) = -\sum_{k,\ell} \mathbf{M}_{k,\ell} \left(\log(\mathbf{M}_{k,\ell}) - 1 \right).$$

and the augmented problem (with $\varepsilon > 0$)

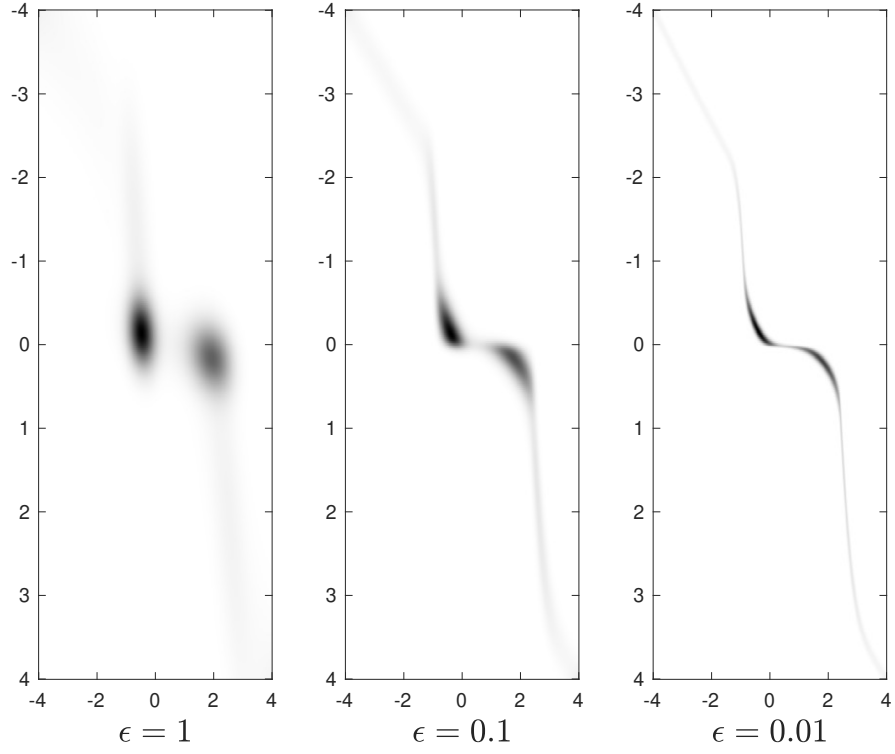
$$\underset{\mathbf{M} \in \mathbb{R}_+^{n \times n}}{\text{minimize}} \quad \text{trace}\left(\mathbf{C}^T \mathbf{M} \right) - \varepsilon D(\mathbf{M}) \text{ , s.t. } \mathbf{M}\mathbf{1} = \boldsymbol{\mu}_X \text{ , } \mathbf{M}^T \mathbf{1} = \boldsymbol{\mu}_Y.$$

- $D(\mathbf{M})$ is 1-strongly concave, yielding an $\varepsilon-$strongly convex problem. The optimal $\mathbf{M}$ is unique.
- As $\varepsilon \to 0$, the solution converges to the optimal $\mathbf{M}$ of the original problem with maximum entropy.
- As $\varepsilon \to \infty$, $\mathbf{M} \to \boldsymbol{\mu}_X \boldsymbol{\mu}_Y^T$ (independent coupling).
- In addition to strong convexity, the penalty $-D(\mathbf{M})$ also leads to a simple solution algorithm.

The effect of entropy



LP solution



Entropy regularization.

Matrix scaling

In fact, it is easy to show (just derivatives) that the unique solution $\mathbf{M}$ can be written as

$$\mathbf{M}_{k,\ell} = \mathbf{u}_k \mathbf{K}_{k,\ell} \mathbf{v}_\ell \iff \mathbf{M} = \operatorname{diag}(\mathbf{u}) \mathbf{K} \operatorname{diag}(\mathbf{v})$$

where $\mathbf{K} = \exp(-\mathbf{C}/\varepsilon)$ (elementwise), and $\mathbf{u}, \mathbf{v} \in \mathbb{R}_+^n$. Finding the optimal $\mathbf{M}$ is thus equivalent to solving the equations

$$\operatorname{diag}(\mathbf{u}) \mathbf{K} \operatorname{diag}(\mathbf{v}) \mathbf{1} = \boldsymbol{\mu}_X \ , \ \operatorname{diag}(\mathbf{v}) \mathbf{K}^T \operatorname{diag}(\mathbf{u}) \mathbf{1} = \boldsymbol{\mu}_Y$$

which can be simplified to

$$\mathbf{u} \odot \mathbf{K}\mathbf{v} = \boldsymbol{\mu}_X \ , \ \mathbf{v} \odot \mathbf{K}^T \mathbf{u} = \boldsymbol{\mu}_Y$$

- This is sometimes called the matrix scaling problem.
- Note: even though $\mathbf{M} = \operatorname{diag}(\mathbf{u}) \mathbf{K} \operatorname{diag}(\mathbf{v})$ is unique, there are many solutions $(\mathbf{u}, \mathbf{v})$ as $(a\mathbf{u}, \frac{1}{a}\mathbf{v})$ also is a solution for any $a > 0$.

Sinkhorn's algorithm

It turns out that we can get a solution by solving the individual equations in an alternating fashion. This is called Sinkhorn's algorithm.

Pick an initial $\mathbf{v}^{(0)}$ (e.g, $\mathbf{v}^{(0)} = \mathbf{1}$). For $j = 1, 2, ...,$ do

1. solve $\mathbf{u} \odot \mathbf{K}\mathbf{v}^{(j-1)} = \boldsymbol{\mu}_X \Rightarrow \mathbf{u}^{(j)} = \boldsymbol{\mu}_X ./ \mathbf{K}\mathbf{v}^{(j-1)}$,
2. solve $\mathbf{v} \odot \mathbf{K}^T\mathbf{u}^{(j)} = \boldsymbol{\mu}_Y \Rightarrow \mathbf{v}^{(j)} = \boldsymbol{\mu}_Y ./ \mathbf{K}^T\mathbf{u}^{(j)}$.

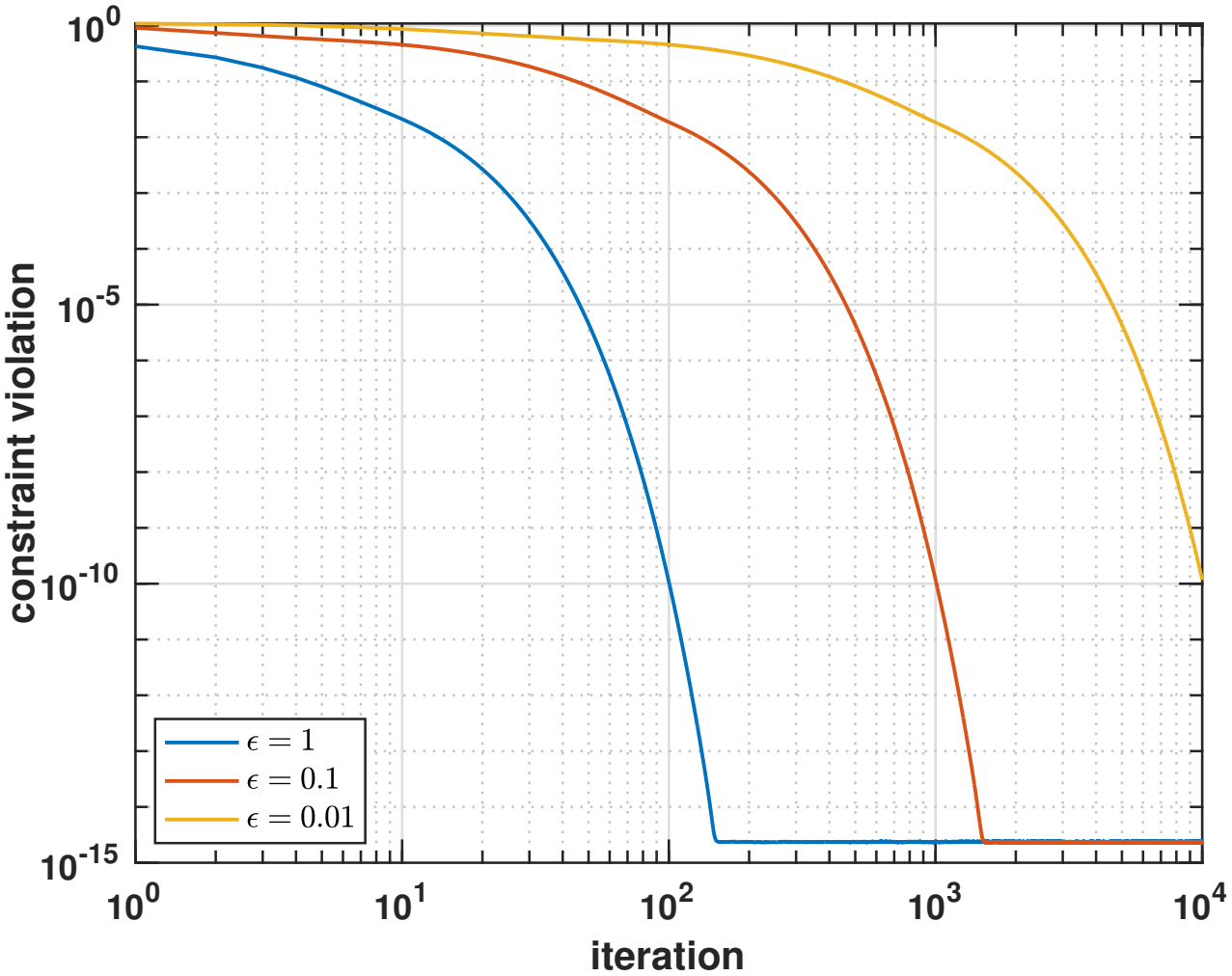
These iterations converge (in a specific sense) as $j \to \infty$, yielding the optimal $\mathbf{M} = \mathrm{diag}(\mathbf{u})\,\mathbf{K}\,\mathrm{diag}(\mathbf{v})$.

- The cost per iteration is $\mathcal{O}(n^2)$ due to the matrix-vector multiplication.
- Can be performed faster in many cases: decoupling between dimensions for d-dimensional problems, exploit convolution properties.

Caveats

- Convergence speed is related to ε: the smaller ε is, the slower the convergence.
- Small ε can cause numerical instability: typically the elements of $\mathbf{u}$ and $\mathbf{v}$ can become really big/small leading to over/underflow.
- For example, although $\mathbf{K}$ may be a convolution kernel it might not be possible to compute $\mathbf{Ku}$ and $\mathbf{Kv}$ using FFTs due to numerical instability.

Sinkhorn convergence



Constraint violation: $\left\| \mathbf{u}^{(j)} \odot \mathbf{K}\mathbf{v}^{(j)} - \boldsymbol{\mu}_x \right\|_1$.

Extensions to inverse problem

Sinkhorn's algorithm is connected to the dual of the entropy-regularized transport problem:

$$\underset{\mathbf{f},\mathbf{g}\in\mathbb{R}^n}{\text{maximize}}\; L(\mathbf{f},\mathbf{g}) = \mathbf{f}^T\boldsymbol{\mu}_X + \mathbf{g}^T\boldsymbol{\mu}_Y - \varepsilon e^{\mathbf{f}^T/\varepsilon}\mathbf{K}e^{\mathbf{g}/\varepsilon},$$

where $\mathbf{f}$ and $\mathbf{g}$ are the Lagrangian multipliers of the constraints. Note the correspondence

$$\mathbf{u} = e^{\mathbf{f}/\varepsilon}\;,\; \mathbf{v} = e^{\mathbf{g}/\varepsilon}.$$

Sinkhorn's algorithm then in fact is equivalent to block-coordinate ascent of the dual problem:

1. minimize $L(\mathbf{f},\mathbf{g}^{(j-1)})$, yielding $\mathbf{f}^{(j)}$,
2. minimize $L(\mathbf{f}^{(j)},\mathbf{g})$, yielding $\mathbf{g}^{(j)}$.

The corresponding inverse problem can be solved in the same way: maximize the dual of

$$\underset{\mathbf{M}\in\mathbb{R}_+^{n\times n}}{\text{minimize}} \quad \text{trace}\left(\mathbf{C}^T\mathbf{M}\right) - \varepsilon D(\mathbf{M})\;,\; \text{s.t. } \boldsymbol{\Gamma}\mathbf{M}\mathbf{1} = \mathbf{r}_X\;,\; \boldsymbol{\Gamma}\mathbf{M}^T\mathbf{1} = \mathbf{r}_Y.$$

Summary

- OMT is a powerful and flexible tool in modeling,
- Not limited to ”conventional” mass distributions: one can consider matrix-valued (and complex) measures. Also possible to formulate clustering problems.
- Very rich theory, and a lot of ongoing research activity in efficient computational tools.
- A lot of potential applications!

Some starting points:

- G. Peyré and M. Cuturi (2019), *Computational Optimal Transport* with companion site *optimaltransport.github.io*
- Toolbox: Python Optimal Transport *pythonot.github.io*,
- Inverse problems: Elvander et. al. (2020), *Multi-marginal optimal transport using partial information with applications in robust localization and sensor fusion* with accompanying code.