Numerical methods for OT

Rodrigue Lelotte

Université Paris 1 Panthéon–Sorbonne

Optimal transport (OT) pops up a little bit everywhere in contemporary mathematics, from pure to very much applied fields.

(Un)fortunately, solving OT numerically is daunting endeavour. This has become a pretty much hot topic in the recent years with a very active & quite big community of researchers.

There are a lot of different types of OT problems (*e.g. multi-marginal*, *dynamical*, *martingale*, *weak*, *semi-discrete*, *moment-constrained* etc.). Here, we will focus on the standard *discrete OT*, and present some standard numerical methods.

The interested reader may roam on the following references:

- G. Peyré and M. Cuturi, Computational Optimal Transport, 2019.
- Q. Mérigot and B. Thibert, Optimal transport: discretization and algorithms, 2020.

Optimal transport: *the gist of it*

Given $\Pi(\mu, \nu)$ the set of **transport plans** for fixed **marginals** $\mu \in \mathcal{P}(X)$ and $\nu \in \mathcal{P}(Y)$, a **transport cost** $c : X \times Y \to \mathbb{R}_+$, the OT reads in full generality

$$\boxed{OT(\mu, \nu) \quad \underset{def}{=} \quad \inf_{\pi \in \Pi(\mu,\nu)} \int_{X \times Y} c(x, y) d\pi(x, y)}$$

This is an (a priori) *infinite-dimensional* problem. Except in very few special cases, optimal plan(s) are unknown. If we want to solve *numerically* the OT, it is required that we *discretise* the problem in some way so as to make it fit on a computer — *i.e.* making it *finite-dimensional*.

Discrete OT

Here, we will discretise in space the marginal spaces $X := \{x_1, \dots, x_n\}$ and $Y := \{y_1, \dots, y_m\}$. The marginals are now (normalised) **vectors** $\mu = (\mu_i)_{i=1,\dots,n} \in \mathbb{R}^n$ and $\nu = (\nu_j)_{j=1,\dots,m} \in \mathbb{R}^m$. The transport plans are now **matrices** $\pi \in \mathbb{R}_+^{n \times m}$ so that $\pi_{ij} \geq 0$ is the amount of mass send from x_i to y_j. The marginal constraints read $\sum_{j=1}^m \pi_{ij} = \mu_i$ and $\sum_{i=1}^n \pi_{ij} = \nu_j$. Finally, the cost of transportation becomes a **cost matrix** $C \in \mathbb{R}^{n \times m}$ where $C_{ij} := c(x_i, y_j)$. Then, the discrete OT reads

$$\boxed{\mathrm{OT}(\mu, \nu) \quad \underset{def}{=} \quad \min_{\pi \in \Pi} \langle C, \pi \rangle := \sum_{i,j} C_{ij} \pi_{ij}}$$

Remark 1. *This corresponds to the general OT setting with atomic measures* $\mu := \sum_{i=1}^n \mu_i \delta_{x_i}$ *and* $\nu := \sum_{j=1}^m \nu_j \delta_{y_j}$.

Π is a convex polytope

The constraint set $\Pi \subset \mathbb{R}_+^{n \times m}$ is a convex polytope. It is defined by $n + m$ linear equations but one is redundant, so there is effectively $n + m - 1$ equations. Indeed, assume the marginal constraints are verified for all $i = 1, \ldots, n$ and for $j = 1, \ldots, m - 1$. Then it is verified for $j = m$, since

$$\sum_{i=1}^{m} \pi_{im} = \sum_{i=1}^{m} \left(\mu_i - \sum_{j=1}^{m-1} \pi_{ij} \right) = \underbrace{\sum_{i=1}^{m} \mu_i}_{=1} - \sum_{j=1}^{m-1} \underbrace{\sum_{i=1}^{m} \pi_{ij}}_{=\nu_j} = \nu_m$$

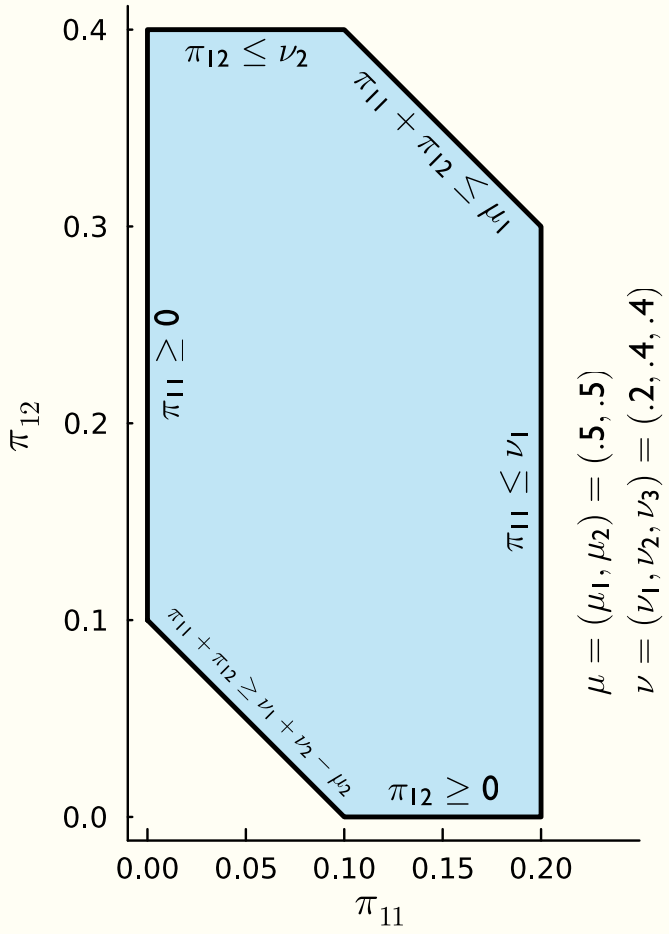
This means there is effectively $nm - (n + m - 1)$ *free* variables.

Example: If $n = m = 2$, then there is one free variable, say π_{11}. Indeed $\pi_{12} = \mu_1 - \pi_{11}$, $\mu_{12} = \nu_1 - \pi_{11}$ and $\pi_{22} = \nu_2 - \pi_{12} = \nu_2 - \mu_1 + \pi_{11}$. Now, it must be that $\pi_{ij} \geq 0$ for all i, j so that the constraint set is parametrised the segment $[\mu_1 - \nu_2, \min(\mu_1, \nu_2)] \subset \mathbb{R}$.

Example: Consider that $n = 2$ and $m = 3$, so that there is only two free variables this time, say π_{11} and π_{12}. Indeed,

$$\begin{cases} \pi_{13} = \mu_1 - \pi_{11} - \pi_{12} \\ \pi_{21} = \nu_1 - \pi_{11} \\ \pi_{22} = \nu_2 - \pi_{12} \\ \pi_{23} = \mu_2 - \pi_{21} - \pi_{22} \end{cases}$$

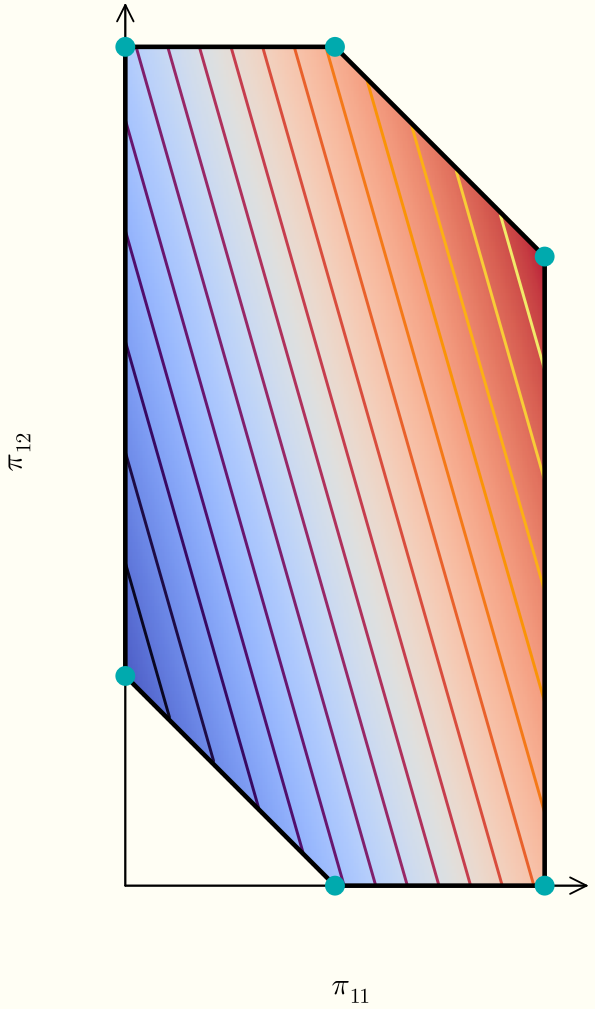
The feasible polytope is then determined using the inequalities $\pi_{ij} \geq 0$ for all i, j.



Vertices of Π

It is a well-known fact that *the minimum of a linear function on a convex set is attained on the* (*relative*) *boundary of the feasible set*. In fact, more precisely on an extreme point or *vertex*, where we recall that:

Definition 1. *A* **vertex** z *of a convex set* C *is a point such that it cannot be written as a (proper) convex combination of two other points in* C*, to wit if* $z = (1-t)x + ty$ *for* $x, y \in C$ *and* $t \in (0,1)$ *then* $x = y = z$*.*



Vertices of Π

Vertices of Π have an interesting structure that has been the subject of extensive research.

Example When $n = m$ and $\mu_i = \nu_i = n^{-1}$ for all $i = 1, \dots, n$ (*i.e.* uniform marginals on X and Y) then Π is called the **Birkhoff polytope**. Its vertices exactly correspond to *permutations matrices*. In particular $|\mathbf{ext}(\Pi)| = n!$.

One (*extremely*) nice numerical feature of vertices of Π is that they are **sparse** in the sense that they contain only a few amount of non-zero entries. Indeed, if $\pi \in \mathbf{ext}(\Pi)$ then $|\mathbf{supp}(\pi)| \leq m+n-1$ where $\mathbf{supp}(\pi) := \{(i,j) : \pi_{ij} > 0\}$. This means that storing these matrices has $\mathcal{O}(n+m)$ memory footprint whereas a **dense** plan — for instance $\pi = \mu\nu^\intercal$ — requires $\mathcal{O}(nm)$

Vertices of Π *in the infinite-dimensional setting*

Same story holds in the infinite-dimensional OT problem, *i.e.* $\exists$ an optimal plan which is in $\mathbf{ext}(\Pi)$. The vertices of Π are also typically "sparse", in the sense that their (measure-theoretic) supports are "small" in some sense.

It is not hard to prove that Monge-type plans $\pi = (I, T)^\sharp \mu$ where $T^\sharp \mu = \mu$ are elements $\mathbf{ext}(\Pi)$. Their supports are small in the sense that are contained in graphs, so $\dim_H(\mathbf{supp}(\pi)) = d \leq \dim_H(\mathbb{R}^d \times \mathbb{R}^d)$. The converse is **not** true: *there* (may) *exists extreme points which are not of a Monge-type*.

The supports of $\pi \in \mathbf{ext}(\Pi)$ have non-trivial structures , see *e.g.* [*Extremal doubly stochastic measures and optimal transportation*, Ahmad et al. 2018]

Network simplex algorithm

Discrete OT is a *Linear Programming* (LP) problem — *i.e. linear objective + convex polytope constraint set*. The standard method to solve LPs is the **Simplex algorithm** [Dantzig, 1947]. In full generality this is a *family* of methods relying on moving on the edges of the feasible polytope from vertices to vertices until a minimiser is found. *How* to move from one vertex to another is somehow problem-dependent.

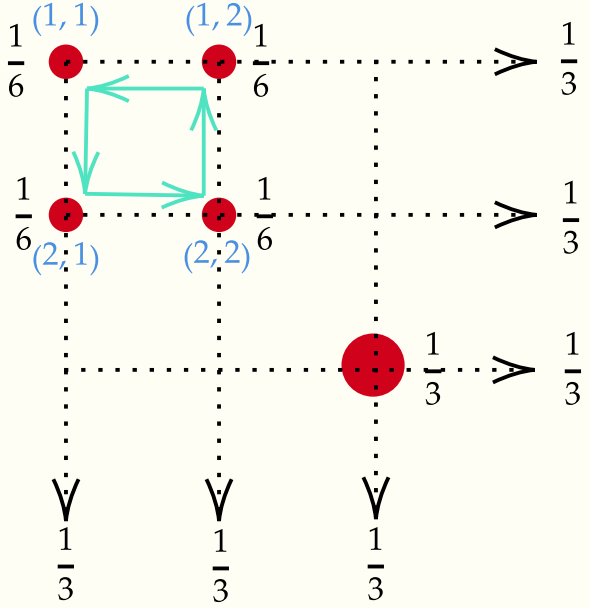
We will now present the **Network Simplex (NS) algorithm** tailored for the discrete OT problem. `It is usually presented using a graph-theoretical langage, but I will phrase things slightly differently and swerve away from graphs`

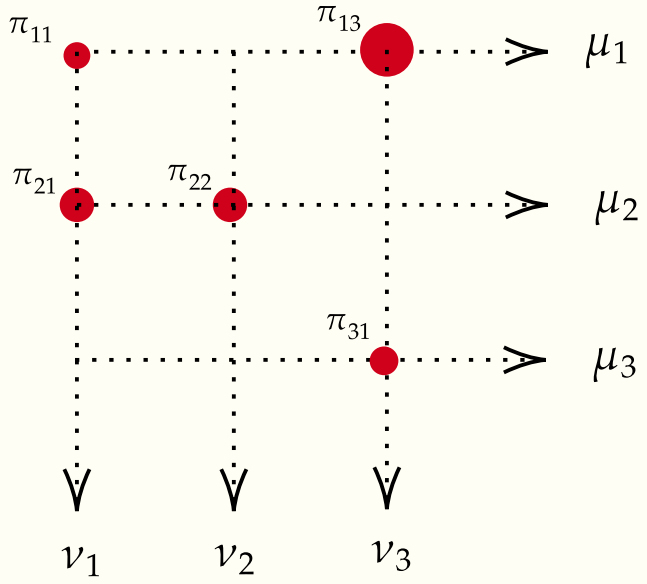
First, we need to explore a little bit more the structure of elements of **ext**(Π).

A property of **ext**(Π)

Lemma 2. *Let* $\pi \in \mathbf{ext}(\Pi)$. *Then, the set* $\mathbf{supp}(\pi)$ *is acyclic in the sense that there does* **not** *exist a chain* $\{(i_s, j_s)\} \subset \mathbf{supp}(\pi)$ *with* $s = 0, \ldots, N$ *– for some (even)* N *– such that* $i_{2s} = i_{2s+1}$ *and* $j_{2s+1} = i_{2(s+1)}$ *for all* $s = 0, \ldots, \frac{N}{2} - 1$ *and* $i_0 = i_N$.

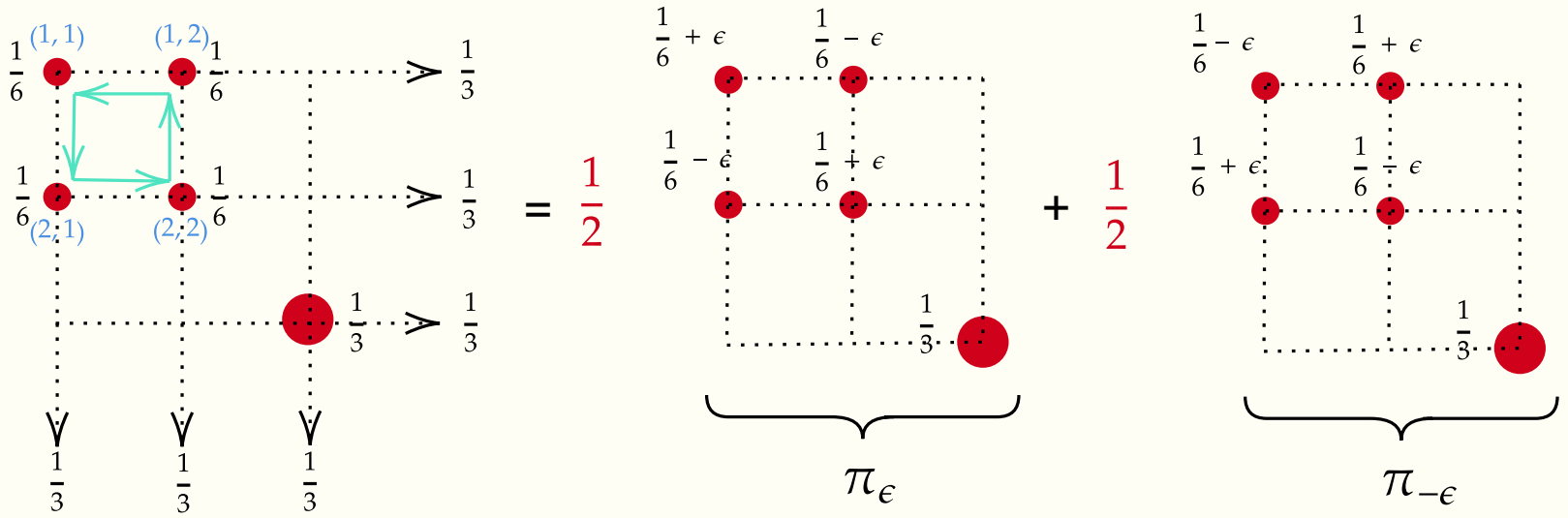
In words: *you cannot close a shape by alternating moves along x-axis and y−axis.* On the left: *not* acylic; on the right: acyclic.





Proof of Lemma 2 (*ex figura patet*)

Proof. Assume there is a cycle. Given $\varepsilon > 0$, follow the chain and add ε to π_{i_s,y_s} and remove ε to next point $\pi_{i_{s+1},y_{s+1}}$ and so on. Let π_ε the resulting matrix. It verifies the marginal constraint because this is a cycle. Repeat this operation with $-\varepsilon$ and let $\pi_{-\varepsilon}$ be the resulting matrix. If $|\varepsilon| < \min_{s=0,\dots,N} |\pi_{i_s,j_s}|$, then π_ε and $\pi_{-\varepsilon}$ remain with nonnegative entries. Then $\pi = \frac{1}{2}(\pi_\varepsilon + \pi_{-\varepsilon})$, a contradiction. □



(Non)degenerate vertices

Using the preceding lemma, we can show that:

Lemma 3. *If* $\pi \in \mathbf{ext}(\Pi)$*, then* $|\mathbf{supp}(\pi)| \leq m + n - 1$*. If* $|\mathbf{supp}(\pi)| > m + n - 1$*, then* π *must have a cycle — and is therefore not a vertex.*

(Omitted proof). This is combinatorics. □

Definition 4. *A vertex whose support has fewer that* $n + m - 1$ *nonzero elements is called* **degenerate**.

❗ While a vertex has necessarily no cycle in its support, the converse statement is **not** true. Nevertheless, we do have:

Lemma 5. *Let* $\pi \in \Pi$ *is such that* $\mathbf{supp}(\pi)$ *does not contain a cycle* **and** *such that* $|\mathbf{supp}(\pi)| = m + n - 1$*. Then* $\pi \in \mathbf{ext}(\Pi)$.

(Discrete) Kantorovich duality: *the gist of it*

The NS algorithm is actually a *primal-dual method*, in the sense that it uses the dual as a *critic* in order to verify whether or not the current iterate is optimal.

In the discrete OT setting, the **Kantorovich duality** reads

$$\max_{(f,g)\in\mathbb{R}^n\times\mathbb{R}^m} \left\{ \langle f, \mu\rangle + \langle g, \nu\rangle : f_i + g_j \leq C_{ij} \text{ for all } i,j \right\}$$

- to wit *Kantorovich potentials* $f, g : \mathbb{R}^d \to \mathbb{R}$ are replaced by **vectors**. The **KKT conditions** reads: $\pi \in \Pi$ *(resp. an admissible pair* (f,g)*) is an optimal for the primal (resp. dual) if and only if* $f_i + g_j = C_{ij}$ *as soon as* $\pi_{ij} > 0$.

Complementary pair to $\pi \in \Pi$

Definition 6. *Given $\pi \in \Pi$, we will say that a pair (f, g) is complementary to π if $f_i + g_j = C_{ij}$ for all $(i, j) \in \mathbf{supp}(\pi)$.*

Beware that (f, g) need *not* be an admissible pair for the Kantorovich dual (that's the all point of it) *i.e.* there may exists $(i, j) \in (\mathbf{supp}(\pi))^c$ such that $f_i + g_j > C_{ij}$.

Remark 2. *A complementary pair need not always exist: consider $n = m = 2$ and $S = [\![2]\!] \times [\![2]\!]$ and assume that the cost matrix is zero everywhere except $C_{22} = 1$. Any complementary pair (f, g) to $\pi \in \Pi$ such that $\mathbf{supp}(\pi) = S$ (i.e. full support) must solve:*

$$\begin{cases} f_1 + g_1 = 0 \\ f_1 + g_2 = 0 \\ f_2 + g_1 = 0 \\ f_2 + g_2 = 1 \end{cases}$$

and from the two 1^{rst} equations, $f_1 = -g_1$ and $f_1 = -g_2$. From the 3^{rd}, $f_2 = f_1$. But then last equation entails $0 = f_2 - f_1 = 1$, a blatant contradiction !

Complementary pair to $\pi \in \Pi$

Nevertheless, complementary pairs exists from extreme points of Π:

Lemma 7. *If $\pi \in \mathbf{ext}(\Pi)$ then there exists a complementary pair (f, g) to π. Moreover, this pair is unique up to additive constants if π is nondegenerate.*

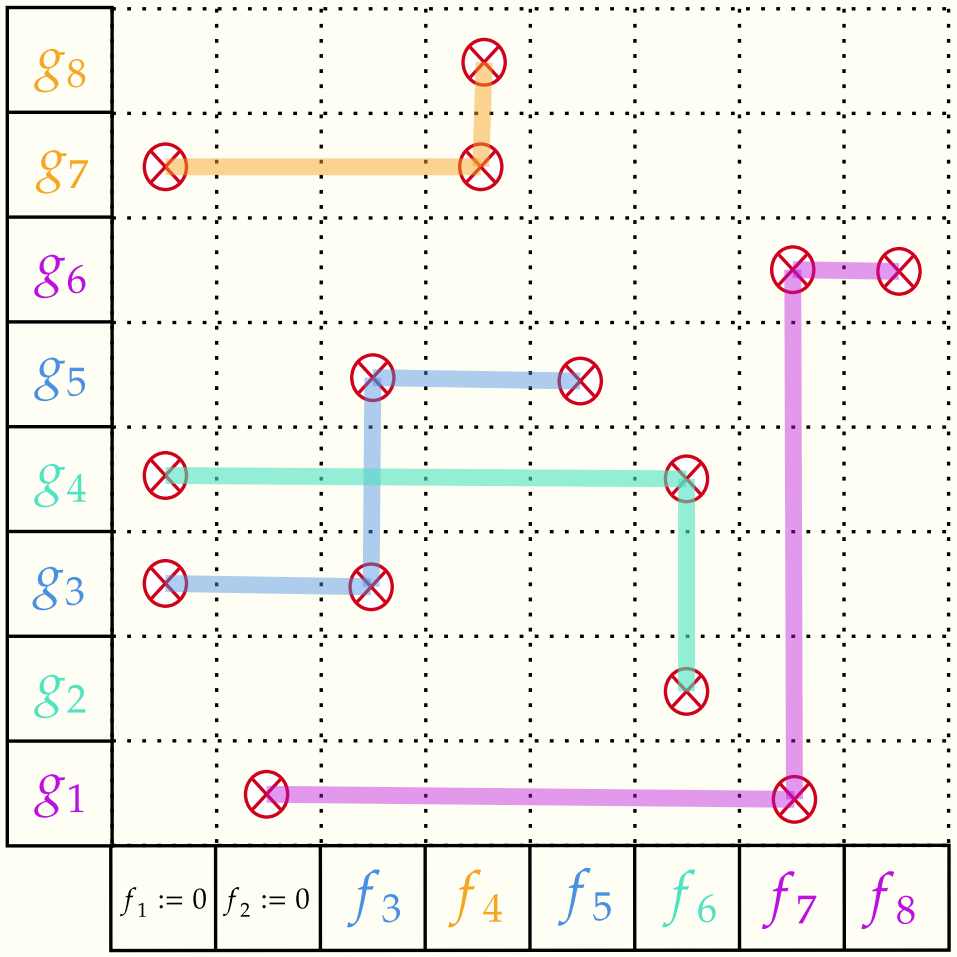
Remark 3. *Solving $f_i + g_j = C_{ij}$ for all $(i, j) \in S := \mathbf{supp}(\pi)$ is equivalent to solving a linear system of $|S| \leq n + m - 1$ equations with $n + m - 1$ variables[1]. If $|\mathbf{supp}(\pi)| < n + m - 1$ — that is π is degenerate — the system is underdetermined so there is an infinite # of solutions (if there exists one).*

---

[1] Recall (again) that there is a *redundant* variable in the dual variables, hence the *minus* 1

Proof of Lemma 7 (by heuristical construction)

We start with $f_1 := 0$. Then, for all j such that $(1, j) \in \mathbf{supp}(\pi)$ we let $g_j := C_{1j} - f_1 = C_{1j}$. Then, for each of these j's, repeat the procedure but along the other dimension. That is, for a specific j_0, consider all i such that $(i, j_0) \in \mathbf{supp}(\pi)$ and set $f_i := C_{ij_0} - g_{j_0}$. *Two paths never cross because S has no cycle* ! Thus this procedure terminates / is consistent. Repeat on each "connected components" until f and g are fully determined.



Network simplex algorithm

The *Network Simplex algorithm* reads as follows:

(A1) Start with a feasible $\pi^0 \in \mathbf{ext}(\Pi)$ such that $|\mathbf{supp}(\pi^0)| = n + m - 1$, that is π^0 is a **nondegenerate** vertex.

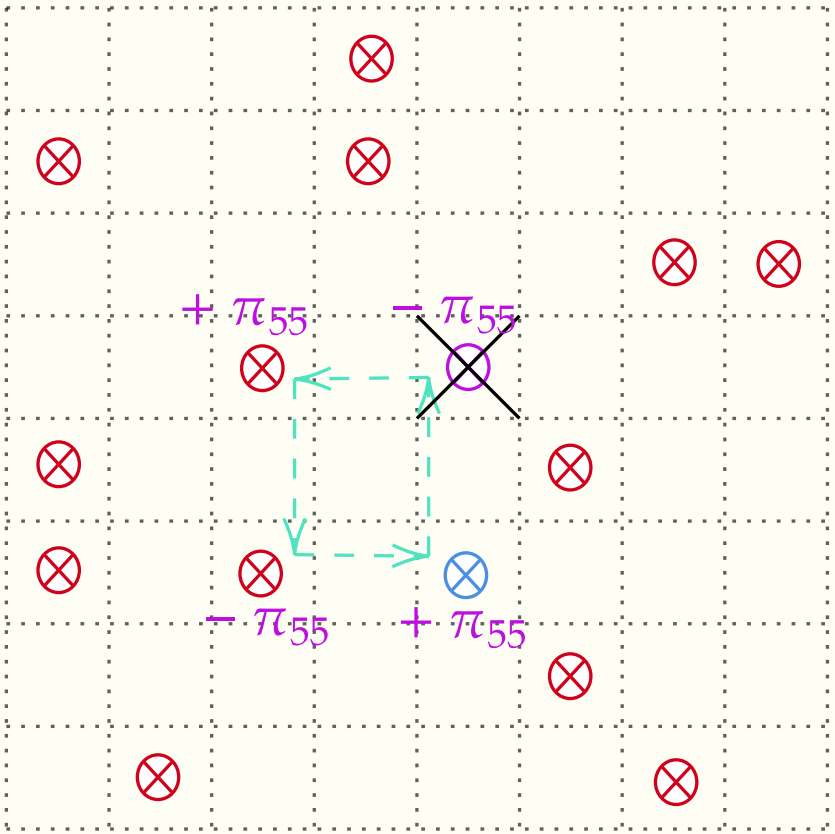
(A2) Build the unique complementary pair (f, g) for π^0 (up to additive constant), using for instance previous construction.

(A3) Search for $(i, j) \in [\![n]\!] \times [\![m]\!]$ such that $f_i + g_j > C_{ij}$. If none is found, then π^0 is optimal in virtue of KKT conditions. Otherwise, let (i, j) be such a pair and move on to **(A4)**.

(A4) Then, $S := \mathbf{supp}(\pi^0) \cup \{(i, j)\}$ has necessarily a cycle since $|S| = n + m > n + m - 1$ (cf. Lemma 3)Then, we move to a new vertex $\pi^1 \in \mathbf{ext}(\Pi)$ constructed as follows (*see next slide*).

How is **(A4)** conducted ?

We will create a new plan π^1 by creating mass at $\{(i,j)\}$ and removing it on the other points of the cycle so as to keep the marginal constraint *and* destroying the induced cycle (so that $\pi^1 \in \mathbf{ext}(\Pi)$ still according to Lemma 5). Simply take $\pi^1_{ij} = \min \pi^0_{kl}$ where the minimum runs over the "odd" points of the cycle starting from (i,j).



Example: Here, $(i,j) = (5,3)$ and the "odd" points of the induced cycle are $(5,5)$ and $(3,3)$. Assume for instance $\pi_{55} < \pi_{33}$. Then, set $\pi^1_{53} := \pi_{55}$ and remove / add π_{55} following the cycle.

Et voilà !

We then go back at **(A2)** – provided that π^1 is **nondegenerate**, *cf. infra* – and repeat until an optimal transport plan is reached. *Et voilà !*

One can actually prove that each iteration increases strictly the objective, see [Peyré & Cuturi, 2019] – which evidently implies convergence in a finite # of step (**Why ?**).

💻 This algorithm can be implemented efficiently using graph-theoretic tools. [Orlin, 1997] proved that it has polynomial complexity, and [Tarjan, 1997] gave an improved complexity bound of $\mathcal{O}((n+m)nm\log(n+m))$. This can accomodate n, m of the order of ~ 1000, but no more – at least for generic costs (*i.e. dense* cost matrices).

Degeneracy

In the preceding example, it may happen that $\pi_{33} = \pi_{55}$. In this case, we do not destroy one but *two* points. This means that $|\mathbf{supp}(\pi^1)| = n + m - 2$ — *i.e.* π^1 is now a **degenerate** vertex.

If we find a point (i, j) that violates the dual constraint, it may happen that adding (i, j) to $\mathbf{supp}(\pi^1)$ does *not* induce a cycle. **It is then unclear how to conduct (A4) !**

In fact, if $|\mathbf{supp}(\pi^1)| < n + m - 1$, **(A2)** is ambiguous since (f, g) is not unique, *cf.* Remark 3. The idea is then to solve for a complementary pair by adding the constraint $f_i + g_j = C_{ij}$ – thus removing the underdeterminacy. At the next step the primal solution remains unchanged. This is done until we add an edge that creates a cycle (which happens, *cf.* Lemma 3).

Dual ascent methods

The NS algorithm used mostly the primal formulation of discrete OT as a starting point — although under the hood it appeals to the dual as a critic.There are also important / historical methods that rather use the *dual formulation* of the discrete OT as a starting point. We refer to [Peyré & Cuturi, 2019] for a nice introduction on some of these methods, such as the (ε-)**auction algorithm** of [Bertsekas, 1981] (and [Bertsekas & Eckstein, 1988]). These methods as **ascent methods** in the sense that they propose directions in which the dual increases.

From a computational point of view, these methods are sometimes more advantagous than NS since they can exploit **parallelised computing architecture** — although they remain rather *inadequate* in very high dimension.

Unconstrained Kantorovich dual

We will briefly mention now from a much theoretical point of view why the dual is a *bad* problem optimisation-wise.

Let us start by recalling that the Kantorovich dual can be expressed as an unconstrained problem using the discrete analogue to the $c-$**transform**. Given $f \in \mathbb{R}^n$, define its *c-transform* $f^c \in \mathbb{R}^m$ as $f_j^c := \min_{i=1,\dots,n}\{C_{ij} - f_i\}$

For any admissible (f, g) we have $g_j \leq C_{ij} - f_i$ hence $g_j \leq f_j^c$, henceforth $D(f, g) \leq D(f, f^c) =: D(f)$. Therefore, the dual can be made unconstrained:

$$OT(\mu, \nu) = \max_{f \in \mathbb{R}^n} D(f) = \sum_{i=1}^{n} f_i \mu_i + \sum_{j=1}^{m} f_j^c \nu_j$$

Concavity of c-transform(s)

Let $f^1, f^2 \in \mathbb{R}^n$ and $f := (1-t)f_1 + tf_2$ for $t \in [0,1]$ be a convex combination of f_1 and f_2. Then

$$f_j^c \stackrel{def}{=} \min_{i=1,\ldots,n} \{C_{ij} - f\} = \min_{i=1,\ldots,n} \left\{(1-t)[C_{ij} - f_i^1] + t[C_{ij} - f_i^2]\right\}$$

Now, using the elementary facts that $\min\{f+g\} \geq \min f + \min g$ and that $\min \lambda f = \lambda \min f$ for any $\lambda \geq 0$, we have

$$\begin{aligned} f_j^c &\geq (1-t) \min_{i=1,\ldots,n} \left\{C_{ij} - f_i^1\right\} + t \min_{i=1,\ldots,n} \left\{C_{ij} - f_i^2\right\} \\ &\geq (1-t)(f^1)_j^c + t(f^2)_j^c \end{aligned}$$

This entails that $\mathbb{R}^n \ni f \mapsto f_j^c \in \mathbb{R}$ is a concave function for all $j = 1, \ldots, m$.

Concavity of Kantorovich dual

Therefore $D : \mathbb{R}^n \to \mathbb{R}$ is itself a **concave** as the sum of the linear function $f \mapsto \langle f, \mu \rangle$ and the function $f \mapsto \langle f^c, \nu \rangle$ which is also concave as a (nonnegative) linear combinaison of concave functions.

Nevertheless, it is *not* strictly concave — not only do we have $(f + \alpha)^c = f^c - \alpha$ hence $D(f + \alpha) = D(f)$ for all $\alpha \in \mathbb{R}$, but we can also find other (less trivial) directions where it is linear (**Exercise !**)

Non-smoothness of Kantorovich dual

We have then rephrased the dual as a maximisation problem of a concave function. Therefore, we may want to appeal to a **gradient ascent method**.

Unfortunately, the dual is **not** smooth. Indeed, the c-transform is **not** a differentiable function. Take for instance $n = 2$ and assume that $c_{1j} = 0$ for all $j = 1, \dots, m$, so that $c_j(f) = \min\{-f_1, -f_2\} = -\max\{f_1, f_2\}$ which has an obvious kink on the diagonal $\Delta := \{f_1 = f_2\} \subset \mathbb{R}^2$.

› The Kantorovich dual is therefore a **bad** problem optimisation-wise, not only because it lacks strict (or strong) concavity, but also because it is not smooth.

[⋆] Going *super* !

Although differentiability is not accessible, we may look at the superdifferential D, where we recall that:

Definition 8. *The superdifferential of a concave function* $f : \mathbb{R}^d \to \mathbb{R}$ *at* x *is the set* $\partial f(x) := \{u \in \mathbb{R}^d : f(y) \leq f(x) + u^\intercal(y - x)\}$.

The idea is then to do a gradient ascent where the gradient is replaced by a *super*gradient, that is $f^{k+1} = f^k + \eta_k g^k$ where $g^k \in \partial D(f^k)$

[⋆] Going *super* !

Exercise 1. *Show that* $\partial c_j(f) := \mathbf{conv}(\{-e_i : i \in I_j(f)\})$ *where* $I_j(f) := arg\,min_{i=1,\dots,n}\{C_{ij} - f_i\}$ – and where **conv** denotes the convex hull.

Exercise 2. *Using the preceding exercise, show that* $g \in \partial D(f)$ *if and only if* $g_i = \mu_i - \sum_{j=1}^{m} \nu_j \lambda_{ij}$ *where the* λ_{ij}*'s are nonnegative and are such that* $\lambda_{ij} = 0$ *if* $i \notin I_j(f)$ *and* $\sum_{i \in I_j(f)} \lambda_{ij} = 1$ *for all* $j = 1, \dots, m$.

Exercise 3. *Show that at optimality, namely when* $0 \in \partial D(f)$, *then we recover the KKT conditions. That is, build a* $\pi \in \Pi$ *such that* $f_i + f_j^c = C_{ij}$ *as soon as* $\pi_{ij} > 0$. **Hint:** *define* $\pi_{ij} = \nu_j \lambda_{ij}$ *for well-chosen* λ_{ij}*'s.*

[⋆] Going *super* !

The **Bertsekas auction algorithm** for $n = m$ with uniform marginals $\mu_i = \nu_i = n^{-1}$ is actually a special kind of supergradient ascent method which consists in choosing i^* such that $e_{i^*} \in \partial D(f^{(\ell)})$ and do an *exact* line-search $t \in \arg\max_s D(f^{(\ell)} + s e_{i^*})$ and let $f^{(\ell+1)} := f^{(\ell)} + t e_{i^*}$. Note that we modify the Kant. potential *one coordinate at a time*.

Exercise 4. *Show that if f is not optimal, then there exists i such that $e_i \in \partial D(f)$ — when $n = m$ and marginals are uniform!*

Remark 4. *In a nonsmooth & merely concave setting, coordinate ascent algorithms may typically get stuck in non-optimal configurations, a phenomenon called jamming. That's why $\varepsilon-$auction is usually prefered.*

Entropic OT: *turning on the heat*

Previous methods are inadequate in (very) high dimension, *i.e.* $n, m \gg 1$. **Entropic OT** fundamentally changed the game and has sparked a *major revival* (birth ?) of numerical OT starting from the seminal works of [Cuturi, 2013] & [Benamou et al., 2015].

Bits of history. We can trace back entropic OT — although without any mention to "OT" *stricto sensu* — at least to the work of physicist E. Schrödinger in 1932. Less well-known is that it also hidden in the *classical Density Functional Theory* (DFT) literature, which is related to computational chemistry. For instance, the "Kantorovich duality" for entropic OT is hidden in [Chayes, Chayes & Lieb, 1984]. Entropic OT shares tenacious link with *statistical physics* — where entropy accounts for the temperature.

Entropy

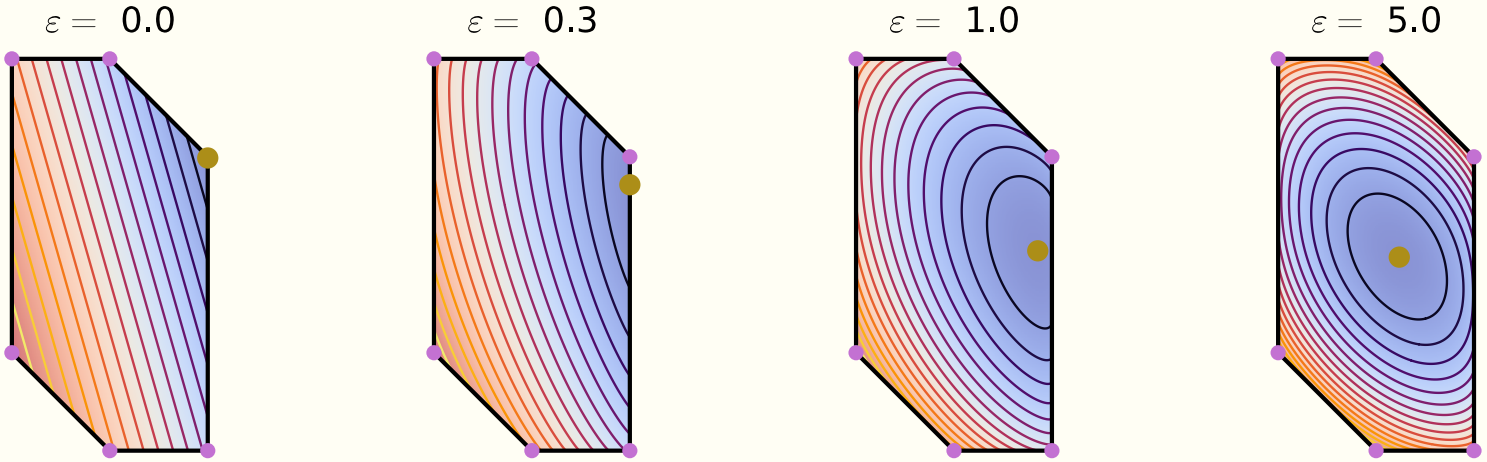
Let $H(\pi) := -\sum_{i,j} \pi_{ij}(\log \pi_{ij} - 1)$ be the **entropy** of π. The function $H : \mathbb{R}^{n \times m} \to \mathbb{R}$ is strictly concave. Indeed, letting $\phi(x) := -x \ln x + 1$, we have $\phi''(x) = -\frac{1}{x}$ so that $H(\pi) = \sum_{i,j} \phi(\pi_{ij})$. In fact, since $\phi''(x) \leq -1$ for $x \leq 1$, H is strongly convex on Π since $\pi_{ij} \leq 1$.

The idea of entropic OT is to use the entropy H as regularisation parameter weighted by a morally small parameter $\varepsilon > 0$:

$$\boxed{\mathrm{OT}_\varepsilon(\mu, \nu) \quad \underset{def}{=} \quad \min_{\pi \in \Pi} \Big\{ \langle C, \pi \rangle - \varepsilon H(\pi) \Big\}}$$

Key properties of entropic OT

This makes the problem *strictly* convex, hence there exists an **unique** solution $\pi_\varepsilon \in \Pi$ — contrary to original OT which may be degenerated. The entropy pushes the original LP solution away from the boundary of Π inside. The solution π_ε progressively moves towards the solution of OT as $\varepsilon \to 0$ and to an "entropic center" as $\varepsilon \to \infty$ (*cf. infra*).



Key properties of entropic OT

Proposition 9. *The unique solution π_ε of OT_ε converges to an optimal solution of OT with maximal entropy within the set of all optimal solutions. On the other hand, one has $\pi_\varepsilon \to \mu \otimes \nu := \mu\nu^\intercal$ when $\varepsilon \to \infty$.*

Proof. Let us prove the limit $\varepsilon \to 0$. By compactness of Π, extract $\varepsilon' \to 0$ s.t. $\pi_\varepsilon \to \pi$ where $\pi \in \Pi$. Let η be any optimal plan for OT, so that $0 \leq \langle C, \pi_{\varepsilon'} \rangle - \langle C, \eta \rangle \leq \varepsilon'(H(\pi_{\varepsilon'}) - H(\eta))$. Letting $\varepsilon' \to 0$ — and by continuity of H — the righthand side vanishes, so $\langle C, \pi \rangle \leq \langle C, \eta \rangle$ henceforth π is optimal for OT. Now, dividing by ε' and letting $\varepsilon' \to 0$ shows that $H(\eta) \leq H(\pi)$, thus π must maximise the entropy among minimisers. □

Continuous entropic OT

In the continuous setting, one needs to consider the **relative entropy** with respect to the product measure $p := \mu \otimes \nu$, that is

$$S(\pi|p) := \int_{\mathbb{R}^d} \frac{d\pi}{dp} \ln(\frac{d\pi}{dp}) dp$$

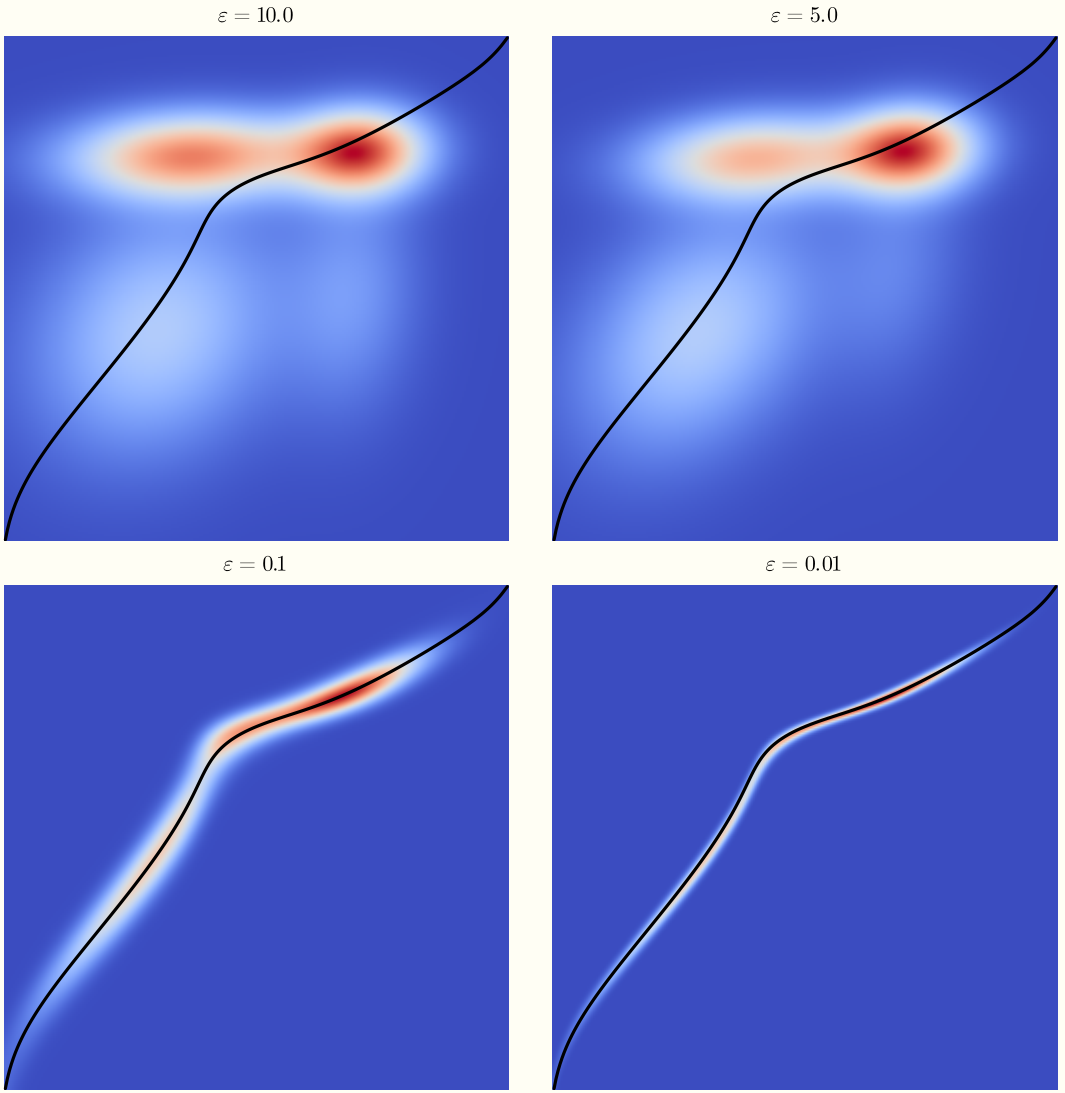
– and let $S(\pi|p) = \infty$ if π is not absolutely continuous with respect to p. Then the EOT in full generality reads

$$\boxed{OT_\varepsilon(\mu,\nu) \quad \underset{def}{=} \quad \min_{\pi \in \Pi} \left\{ \int_{X\times Y} c(x,y) d\pi(x,y) + \varepsilon S(\pi) \right\}}$$

Again, there exists an unique minimiser π_ε by strict convexity. An important remark is that $\pi_\varepsilon \ll p$ necessarily, meaning that its support is typically $X \times Y$. Adding entropy therefore forces the plan to have full support — in particular, it is not of a Monge-type.

Example: the limit $\varepsilon \to 0$.

Here μ and ν are mixtures of one-dimensional Gaussians. The (exact) optimal transport map T is known (*i.e.* the monotone one). We then discretise the marginal spaces using $n = m = 250$ points. The plans π_ε are computed at decreasing $\varepsilon \downarrow 0$. We remark that π_ε has full support and concentrates near the graph T (resp. as $\mu \otimes \nu$) as $\varepsilon \to 0$ (resp. $\varepsilon \gg 1$).



Optimality conditions

Let us introduce the Lagrangian associated to EOT:

$$\mathcal{L}(\pi, f, g) = \langle C, \pi \rangle + \varepsilon H(\pi) - \langle f, \pi \mathbb{1}_m - \mu \rangle - \langle g, \pi^T \mathbb{1}_n - \mu \rangle$$

– where $\mathbb{1}_k$ denotes the vector with k zeros, so that marginal constraints read $\pi \mathbb{1}_m = \mu$ and $\pi^\intercal \mathbb{1}_n = \nu$. This is a smooth & stricty convex function in π, so let us solve for the first-order optimality condition. A trivial computation shows

$$\partial_{\pi_{ij}} \mathcal{L} = C_{ij} + \varepsilon \ln \pi_{ij} - f_i - g_j.$$

Solving $\partial_{\pi_{ij}} \mathcal{L} = 0$ leads to $\pi_{ij} = e^{f_i/\varepsilon} K^\varepsilon_{ij} e^{g_j/\varepsilon}$ where we define $K^\varepsilon_{ij} := e^{-C_{ij}/\varepsilon}$. The resulting matrix $K^\varepsilon := [K^\varepsilon_{ij}]$ is sometimes called the *Gibbs kernel*.

Optimality conditions & duality

From what precedes, we obtain:

Theorem 10. *The unique solution π_ε to OT_ε has the form $[\pi_\varepsilon]_{ij} = u_i K^\varepsilon_{ij} v_j$ for two (unknown) vectors $u \in \mathbb{R}^n$ and $v \in \mathbb{R}^m$.*

Proof. *Cf. supra* with $u_i = e^{f_i/\varepsilon}$ and $v_j = e^{g_j/\varepsilon}$. □

The vectors u, v (or equivalently f and g) will be determined by solving $\sup_{f,g} \inf_\pi \mathcal{L}(\pi, f, g)$. Plugging the optimal π obtained from a pair (f, g) *supra*, we obtain `[under strong duality]`:

$$OT_\varepsilon(\mu,\nu) = \sup_{f,g} \left\{ \langle f, \mu\rangle + \langle g, \nu\rangle - \varepsilon \sum_{i,j} \exp\left(\frac{f_i + g_j - C_{ij}}{\varepsilon}\right) \right\}$$

and this is the **entropic Kantorovich duality**.

Entropic Kantorovich duality

The entropic dual is now a **smooth** & **strictly** concave problem — modulo adding / removing constants. If we solve for the first-order optimality conditions, we find:

$$\partial_{f_i} D_\varepsilon(f, g) = \mu_i - \sum_{j=1}^{m} \underbrace{\exp\left(\frac{f_i + g_j - C_{ij}}{\varepsilon}\right)}_{:=\pi_{ij}} = 0$$

– and likewise for $\partial_{g_j} D_\varepsilon(f, g) = 0$. Therefore $\pi := [\pi_{ij}] \in \Pi$. This leads to the converse statement of Theorem 10:

Theorem 11. *If $\pi = \mathbf{diag}(u) K^\varepsilon \mathbf{diag}(v)$ verifies the marginal constraints, then it is the unique optimal plan for OT_ε.*

We used the notation $\mathbf{diag}(u)$ to denote the diagonal matrix whose diagonal coefficients are those of the vector $u \in \mathbb{R}^n$.

Sinkhorn's algorithm

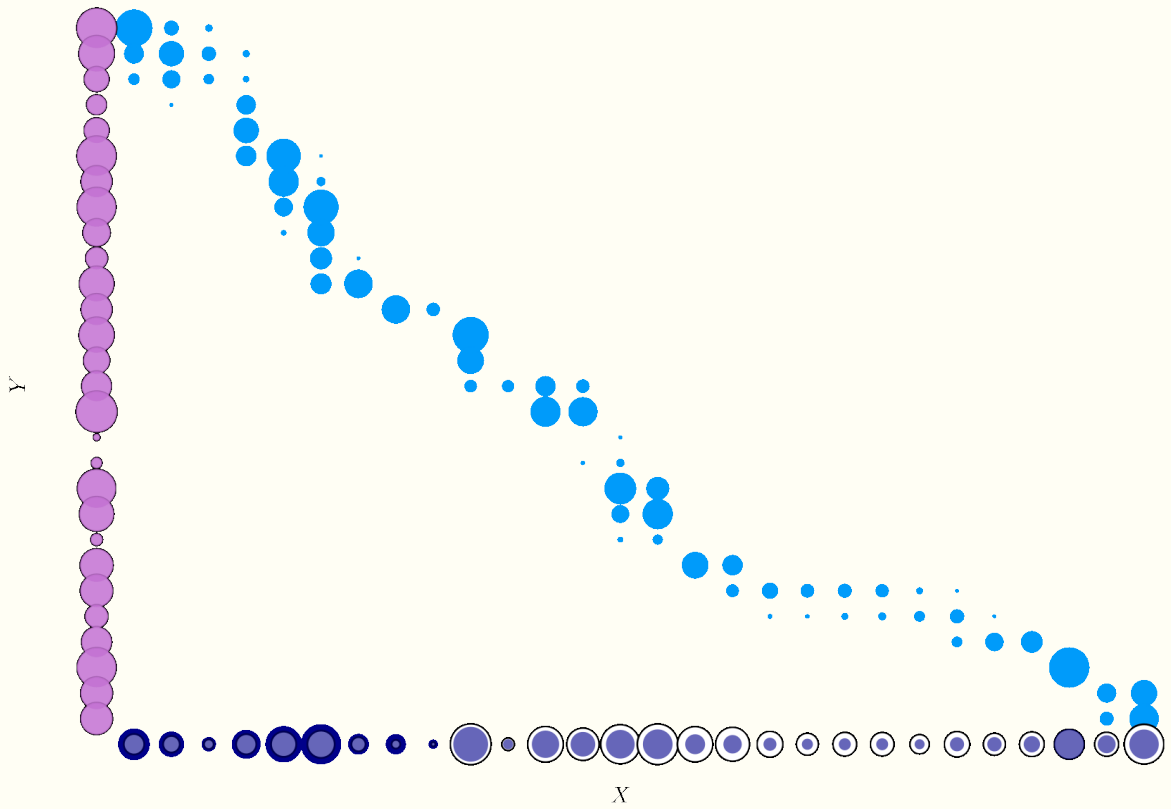
From Theorem 11, solving the EOT is equivalent to a *matrix scaling problem*: we want to find $u \in \mathbb{R}^d$ and $v \in \mathbb{R}^m$ such that $\pi \mathbb{1}_m = \mu$ and $\pi^\intercal \mathbb{1}_n = \nu$ where $\pi := \mathbf{diag}(u) K^\varepsilon \mathbf{diag}(v)$. This is then equivalent to $u \odot Kv = \mu$ and $v \odot K^\intercal u = \nu$ — where $\odot$ corresponds to entrywise multiplication of vectors.

An natural way to handle these equations is to solve them iteratively, first solving for u and then solving for v. This two updates define the **Sinkhorn's algorithm**:

$$u^{(\ell+1)} := \frac{\mu}{Kv^{(\ell)}}, \qquad \text{then} \quad v^{(\ell+1)} = \frac{\nu}{K^\intercal u^{(\ell+1)}}$$

– where division is understood entrywise. **These iterations converge**, see *e.g.* [Peyré & Cuturi, 2019]

Example: Visualisation of Sinkhorn's updates



An animated `.gif` that shows the update of $\pi^{(\ell)} := u^{(\ell)} K^{\varepsilon} v^{(\ell-1)}$ along iterations. One marginal constraint is always exactly verified (*That's expected !*). Although $\pi_{ij}^{(\ell)} > 0$ (and note that $\pi_{ij}^{\varepsilon} > 0$ for all i, j from Theorem 10), we used some threshold procedure for visualisation.

Implementation & complexity

While NS & other combinatorial methods are rather complicated to implement, Sinkhorn's algorithm is deceptively simple:

```
function sinkhorn(mu, nu, C; eps=0.1, niter=200)
  K = exp.(-C / eps)
  u, v = ones(length(mu)), ones(length(nu))
  for k = 1:niter
    u .= mu ./ (K * v)
    v .= nu ./ (K' * u)
  end
  return Diagonal(u) * K * Diagonal(v)
end
```

One can show that to reach τ-approximation of the exact (unregularised) OT solution, the worst-case complexity of Sinkhorn's algorithm is $\mathcal{O}(n^2 \ln n \tau^{-3})$. In practice, it benefits from highly efficient subroutines — vectorisations, GPU & parallelisation...

A first observation

If f is fixed, we can choose g that minimises $D_\varepsilon(f, \cdot)$. By strict concavity, we solve the first-order condition:

$$\partial_{g_j} D_\varepsilon(f, g) = \nu_j - \sum_{i=1}^{n} \exp\left(\frac{f_i + g_j - C_{ij}}{\varepsilon}\right) = 0.$$

This implies that $\exp\left(\frac{g_j}{\varepsilon}\right) \sum_{i=1}^{n} \exp\left(-\frac{C_{ij} - f_i}{\varepsilon}\right) = \nu_j$. Letting $u \in \mathbb{R}^n$ and $v \in \mathbb{R}^m$ so that $u_i := \exp(f_i/\varepsilon)$ and $v_j := \exp(g_j/\varepsilon)$, this reads $v_j = \frac{\nu_j}{\sum_{i=1}^{n} K_{ij}^{\varepsilon} u_i}$ which is the same as $v = \frac{\nu}{K^\intercal u}$. Sinkhorn's algorithm is therefore equivalent to an (*exact*) **coordinate gradient ascent** on the dual:

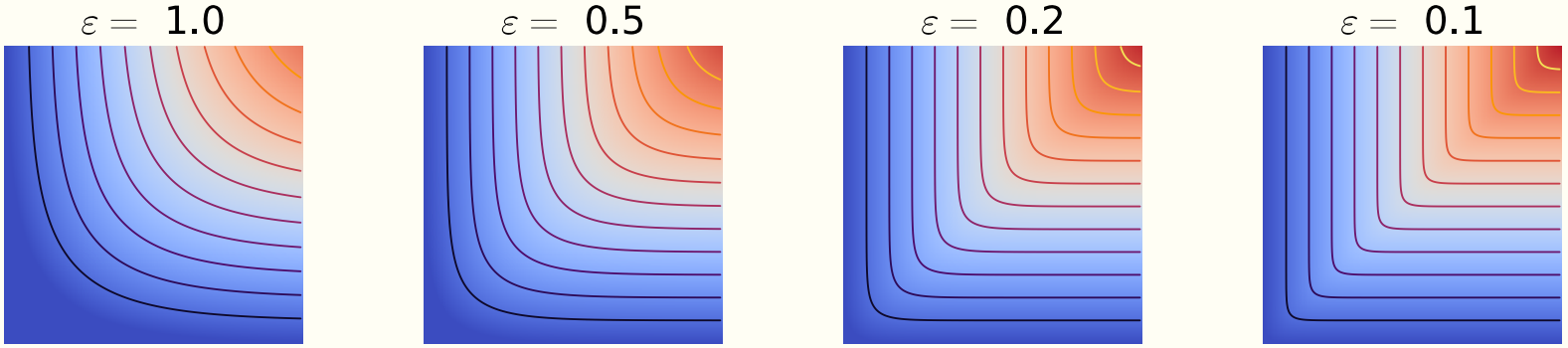
$$f^{(\ell+1)} = \operatorname*{arg\,min}_{f \in \mathbb{R}^d} D_\varepsilon(f, g^{(\ell)}), \qquad g^{(\ell+1)} = \operatorname*{arg\,min}_{g \in \mathbb{R}^m} D_\varepsilon(f^{(\ell+1)}, g)$$

A second observation

We may assume without loss of generality that $\mu_i, \nu_j > 0$ (**Why ?**) for all $i, j > 0$, so that we obtain

$$g_j = \varepsilon \ln(\nu_j) - \varepsilon \ln \sum_{i=1}^{n} \exp\left(-\frac{C_{ij} - f_i}{\varepsilon}\right) = \varepsilon \ln(\nu_j) + \min_{\varepsilon}\{C_{(\cdot)j} - f\}$$

where we introduce the *softmin operator* $\min_\varepsilon : \mathbb{R}^d \to \mathbb{R}$ defined as $\min_\varepsilon z := -\varepsilon \ln \sum_{i=1}^{d} e^{-z_i/\varepsilon}$ for all $z \in \mathbb{R}^d$. The softmin operator is a smooth approximation of the min operator, *i.e.* $\min_\varepsilon z \to \min z$ as $\varepsilon \to 0$.



A second observation (cont') : *the* (c, ε)*-transform*

Let us introduce the $(c, \varepsilon)-$**transform** $f^{c,\varepsilon}$ of $f \in \mathbb{R}^n$ so that $f_j^{c,\varepsilon} = \min_\varepsilon\{C_{(\cdot)j} - f\}$. [*In the limit* $\varepsilon \to 0$, *we recover the* $c-$*transform !*] Then, the dual rewrites with a *single* variable f:

$$OT_\varepsilon(\mu, \nu) = \max_{f \in \mathbb{R}^n} \left\{ \langle f, \mu \rangle + \langle f^{c,\varepsilon}, \nu \rangle + c(\varepsilon, \nu) \right\}$$

where $c(\varepsilon, \nu)$ is a (computable & unimportant) constant depending only on ε and ν. Now, contrary to the c-transform, (c, ε)-transform is **smooth**, so that we can appeal to gradient ascent methods on the entropic dual — *which is essentially Sinkhorn's, then, according to what precedes !*

Checkpoint

There is (at least) two way to think about entropic OT:

Primal. *Entropic OT is adding an entropic penalisation term so that the (primal) problem becomes strictly convex with (hence an unique) minimiser π_ε. Solving for the KKT conditions gives an explicit form for π_ε which, from a numerical point of view, can be solved efficiently using matrix scaling techniques.*

Dual. *Entropic OT consists in substituting to the (non-smooth) c-transform its smoothed version, namely the $(c, \varepsilon)-$transform, in the Kantorovich dual. This makes the dual smooth and $\varepsilon-$strongly concave, thus amenable to standard gradient ascent techniques.*

The regime $\varepsilon \to 0$

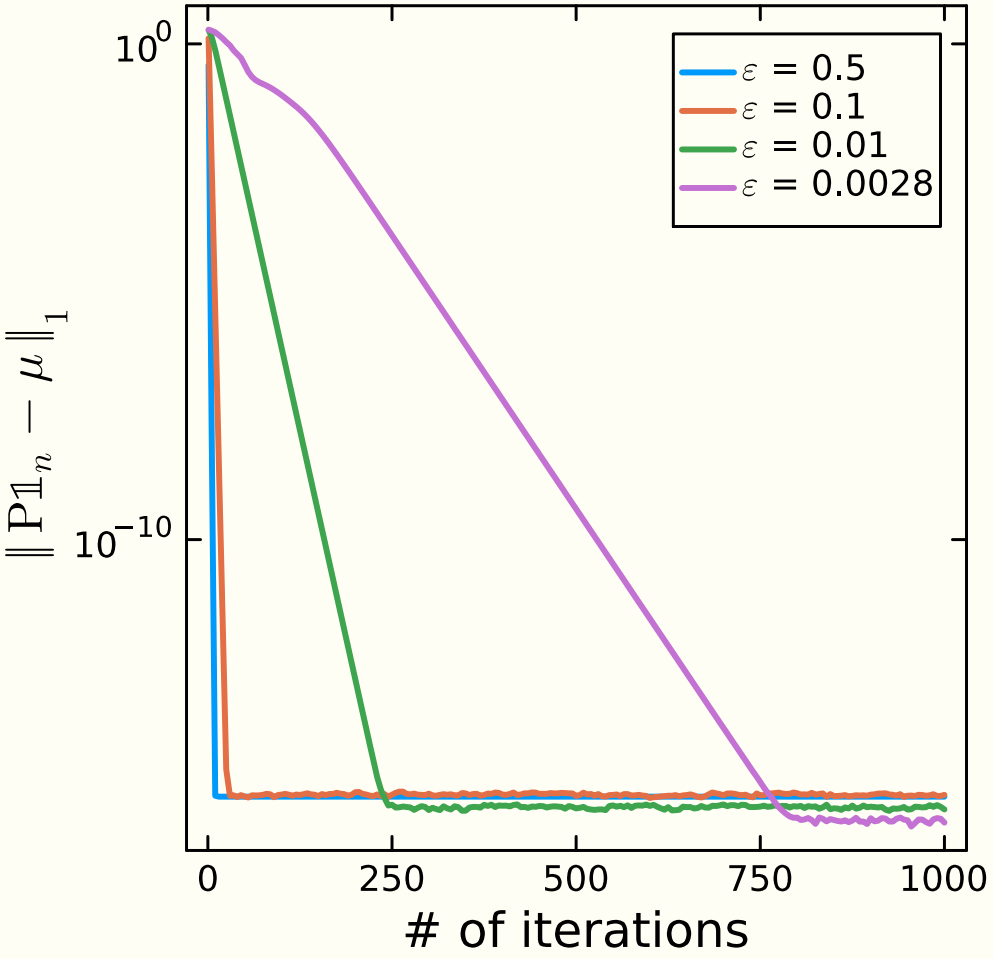
Thanks to Proposition 9, to approximate a solution to (unregularised) OT, we would like to solve EOT for $0 < \varepsilon \ll 1$ *as small as possible*. But there are two limitations to letting $\varepsilon \to 0$:

Theoretical one. As $\varepsilon \to 0$, the problems become *theoretically* harder to solve. There are many ways to see that: one is that the entropic dual, which is smooth & ε-strongly concave, looses both its concavity property and its smoothness at $\varepsilon = 0$.

Numerical one. As $\varepsilon \to 0$, coefficients of the Gibbs kernel $K_{ij} = \exp(-C_{ij}/\varepsilon)$ become crazily small and are eventually set to 0 once they reach the floating-point accuracy of your machine. This is called *underflow* in computer science.

How to overcome these limitations ?

Theoretical ones. *You can't.* That's built-in in the theory, and **there is no magic**: *As* $\varepsilon \to 0$, *the complexity worsen.* There are ways to make things better by using all sorts of clever procedures, but no general tool tailored to all situations. This is also related to so-called *critical slowing down*: *Sampling a Gibbs measure* $\pi(x) \propto \exp(-E(x)/\varepsilon)$ *becomes harder (eventually impossible) as* $\varepsilon \to 0$.



How to overcome these limitations ?

Numerical one. Do not work with the exponential directly, but move to the *log-domains*. That is: *Instead of working with* u *and* v*, work with* $\ln u$ *and* $\ln v$ – where ln is understood entrywise. In particular, we never have to build the Gibbs kernel K.

We recall that at optimality $u = \exp(f/\varepsilon)$ and $v = \exp(g/\varepsilon)$ where f and g are entropic Kantorovich potentials — *i.e.* solutions to the entropic Kantorovich dual, *cf.* Slide 38. We also recall that *Sinkhorn is equivalent to coordinate gradient ascent* of the entropic dual:

$$\left\{\begin{array}{lcl} g_j^{(\ell+1)} & = & \varepsilon \ln(\nu_j) + \min_\varepsilon\{C_{(\cdot)j} - f^{(\ell)}\} \\ f_i^{(\ell+1)} & = & \varepsilon \ln(\mu_i) + \min_\varepsilon\{C_{i(\cdot)} - g^{(\ell+1)}\} \end{array}\right.$$

The *log-sum-exp* trick

For all $z \in \mathbb{R}^d$ and all $\underline{z} \in \mathbb{R}$, the softmin operator rewrites

$$\min_\varepsilon z = \underline{z} - \varepsilon \ln \sum_{i=1}^{d} e^{-(z_i - \underline{z})/\varepsilon}.$$

The idea of **log-sum-exp trick** is to take $\underline{z} := \min z$. This stabilises numerically the computation of $\min_\varepsilon z$ because the shifted coefficients $z_i - \underline{z}$ are (hopefully much) smaller, preventing underflowing.

Beware this has nevertheless a computationnal cost, since one needs to compute at each iteration $\min_i \{C_{ij} - f^{(\ell)}\}$ for all $i = 1, \ldots, n$ — and then $\min_j \{C_{ij} - g^{(\ell+1)}\}$ for all $j = 1, \ldots, m$ at the next iteration. For generic costs matrices, this costs $\mathcal{O}(nm)$ operations. *There's no free lunch* ;-)

Conclusion and beyond

Solving discrete OT is *not* simple endeavour. To solve *exactly* this problem, we can appeal to clever combinatorial algorithms such as *Network Simplex* algorithm or other methods tailored for LPs such as *Interior-point methods*. These methods nevertheless do not scale for large problems, *i.e.* $n, m \gg 1$, and can be hard and cumbersome to implement in practice.

For large-scale problems, we then typically resort to entropic methods, which can be efficiently implemented on GPUs. Of course, this comes at the price of not solving *exactly* the OT problem but only an approximation of it. And letting $\varepsilon \to 0$ is not "free" at all.

Conclusion and beyond

Numerics for OT is a **very active field of research**. I am personally interested in the *multimarginal problem*: instead of having only two marginals μ, ν, you have N of them $\mu_1, \dots, \mu_N$ — where N is a big as you want. This problem appears notably in quantum chemistry, where N is the *number of electrons*.

If you discretise the multimarginal OT problem, you find yourselves with *tensors* with N dimensions, so that the number of variables **grows exponentially as** n^N where n is the $\#$ of discretisation points. *So* **you cannot even store** *your tensor cost on your computer if* $n, N \gg 1$ *!* So Sinkhorn's algorithm is a dead-end ! Moreover, in quantum chemistry, we typically want to reach very accurate computations. Bottom line, *this requires some thinking* ;-)