A differentiable approximation for the Linear Sum Assignment Problem with Edition

Luc Brun[(1)], Benoit Gaüzère[(2)], Sébastien Bougleux[(1)], Florian Yger[(3)]

(1) Normandie Univ, ENSICAEN, CNRS, UNICAEN, GREYC, Caen
(2) Normandie Univ, INSA Rouen, Univ. Rouen, Univ. Havre, LITIS, Rouen
(3) PSL Université Paris-Dauphine, LAMSADE, Paris.
France

The LSAP problem

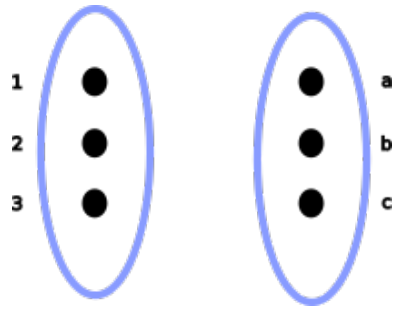
Let us consider:

1. Two sets of n elements, e.g. a set of projects and a set of engineers,
2. a cost function $c(.,.)$ encoding the cost of mapping any elements of the first set onto an element of the second set.

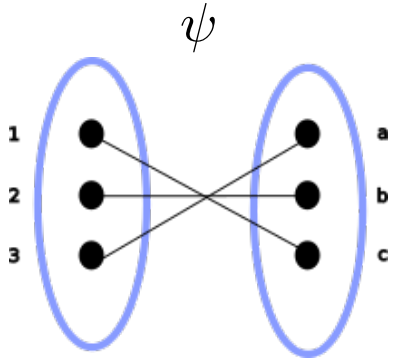
We want to determine a one to one mapping ψ minimizing the sum of costs, i.e. such that:

$$\psi \in \arg\min_{\varphi \in \mathcal{P}_n} \sum_{i=1}^{n} C\left(i, \varphi(i)\right)$$

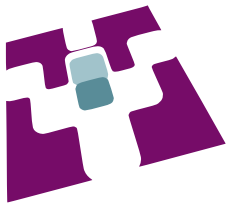
$\mathcal{P}_n$: set of permutations over n elements.



$$C = \begin{array}{c|ccc} & a & b & c \\ \hline 1 & 10 & 15 & 5 \\ 2 & 4 & 3 & 2 \\ 3 & 1 & 5 & 8 \end{array}$$



Solvable using the Hungarian algorithm in $\mathcal{O}(n^3)$ time complexity [Kuhn, 1955].



An alternative formulation

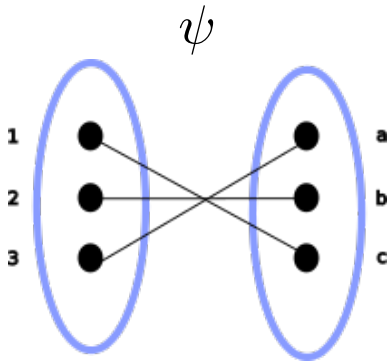
The permutation ψ may be encoded by a permutation matrix X so that:

$$x_{i,j} = \begin{cases} 1 & \text{if } \psi(i) = j \\ 0 & \text{otherwise} \end{cases}$$

We have then to find X^* such that :

$$\begin{aligned} X^* &\in \arg\min_{X \in \mathcal{M}_n} \sum_{i=1}^{n} \sum_{j=1}^{n} c(i,j) x_{i,j} \\ &\in \arg\min_{X \in \{0,1\}^{n \times n}} < C, X > \text{ subject to } X \mathbb{1}_n = \mathbb{1}_n \text{ and } X^T \mathbb{1}_n = \mathbb{1}_n \end{aligned}$$

is minimum.



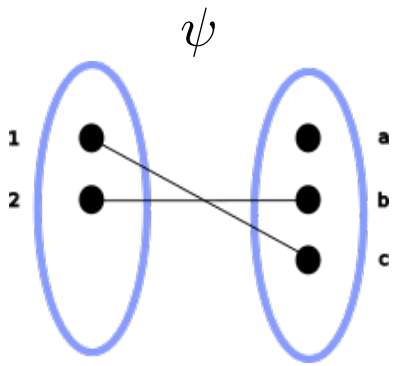
$$X = \begin{array}{c|ccc} & a & b & c \\ \hline 1 & 0 & 0 & 1 \\ 2 & 0 & 1 & 0 \\ 3 & 1 & 0 & 0 \end{array}$$

And if both sets have not the same size ?

Let us denote by n and m the cardinals of both sets, with $n \leq m$. We search for an injective mapping:

$$\begin{aligned} min_{\varphi} \sum_{i=1}^{n} c(i, \varphi(i)) &= min_X \sum_{i=1}^{n} \sum_{j=1}^{m} c(i,j) x_{i,j} \\ &= min_{X \in \{0,1\}^{n \times m}} < C, X > \text{ subject to } \quad X \mathbb{1}_m = \mathbb{1}_n \text{ and } \\ & \qquad\qquad\qquad\qquad\qquad\qquad\qquad\qquad\qquad\qquad\qquad\qquad X^T \mathbb{1}_n \leq \mathbb{1}_m \end{aligned}$$

$$C = \begin{array}{c|ccc} & a & b & c \\ \hline 1 & 10 & 15 & 5 \\ 2 & 4 & 3 & 2 \end{array}$$



$$X = \begin{array}{c|ccc} & a & b & c \\ \hline 1 & 0 & 0 & 1 \\ 2 & 0 & 1 & 0 \end{array}$$

Known as the rectangular Hungarian method. Trivial solution: insert $m - n$ rows filled with 0 in C and apply the Hungarian algorithm.

The cost of unmatched elements

This problem called LSAPE is modeled by an $(n+1) \times (m+1)$ cost matrix C:

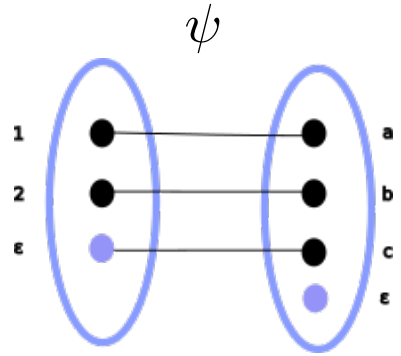
- $\mathbf{C(:, m+1)}$: cost of non matching elements of the first set,
- $\mathbf{C(n+1, :)}$: cost of non matching elements of the second set,

X is a $(n+1) \times (m+1)$ matrix called called an ϵ-assignment matrix:

$$\min_{X \in \{0,1\}^{(n+1)\times(m+1)}} < C, X > \text{ subject to } \begin{cases} Proj_n(X \mathbb{1}_{m+1}) & = & \mathbb{1}_n \text{ and} \\ Proj_m(X^T \mathbb{1}_{n+1}) & = & \mathbb{1}_m \end{cases}$$

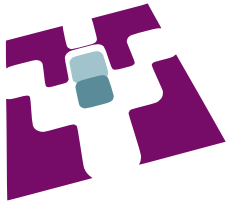
Think of X as a Permutation matrix relaxed on the last row/column.

$$C = \begin{array}{c|cccc} & a & b & c & \epsilon \\ \hline 1 & 10 & 15 & 5 & 10 \\ 2 & 4 & 3 & 2 & 8 \\ \epsilon & 10 & 8 & 3 & 0 \end{array}$$



$$X = \begin{array}{c|cccc} & a & b & c & \epsilon \\ \hline 1 & 1 & 0 & 0 & 0 \\ 2 & 0 & 1 & 0 & 0 \\ \epsilon & 0 & 0 & 1 & 0 \end{array}$$

Optimal solution in $\mathcal{O}(min(n,m)max(n,m)^2)$ [Bougleux et al., 2020]



Applications to Graphs

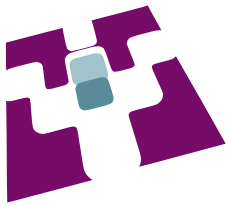
Graph isomorphism: Hungarian method,

Sub Graph isomorphism: Rectangular Hungarian method,

Graph edit distance: LSAPE.

Pb: None of these methods is Deep compliant.

We need a continuous version of LSAPE.



The Sinkhorn Algorithm

Given a $n \times n$ **similarity** matrix S, it exists (under some conditions) two diagonal matrices D_1 and D_2 such that $X = D_1 S D_2$ is doubly stochastic.

$C =$

	a	b	c
1	10	15	5
2	4	3	2
3	1	5	8

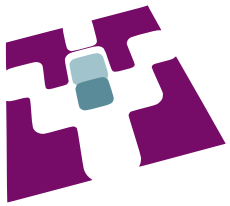
$S =$

	a	b	c
1	10	5	15
2	16	17	18
3	19	15	12

$X =$

	a	b	c
1	.33	.21	.46
2	.29	.40	.31
3	.38	.39	.23

Sinkhorn algorithm may be considered as the transport-version of the Hungarian algorithm (with uniform weights).



A differentiable LSAPE

Under some basic hypothesis and ($S(n+1,:) > 0$ and $S(:,m+1) > 0$), we can define two series of diagonal matrices $D_{1,p}$ and $D_{2,p}$ which converge to a limit such that:

$$X = \lim_{p \to +\infty} D_{1,p} S D_{2,p}$$

is ϵ bi-stochastic ($Proj_n(X \mathbb{1}_{m+1}) = \mathbb{1}_n$ and $Proj_m(X^T \mathbb{1}_{n+1}) = \mathbb{1}_m$).

	a	b	c	ϵ
1	10	15	5	10
2	4	3	2	8
ϵ	10	8	3	0

C

	a	b	c	ϵ
1	10	5	15	0
2	16	17	18	2
ϵ	0	2	7	1

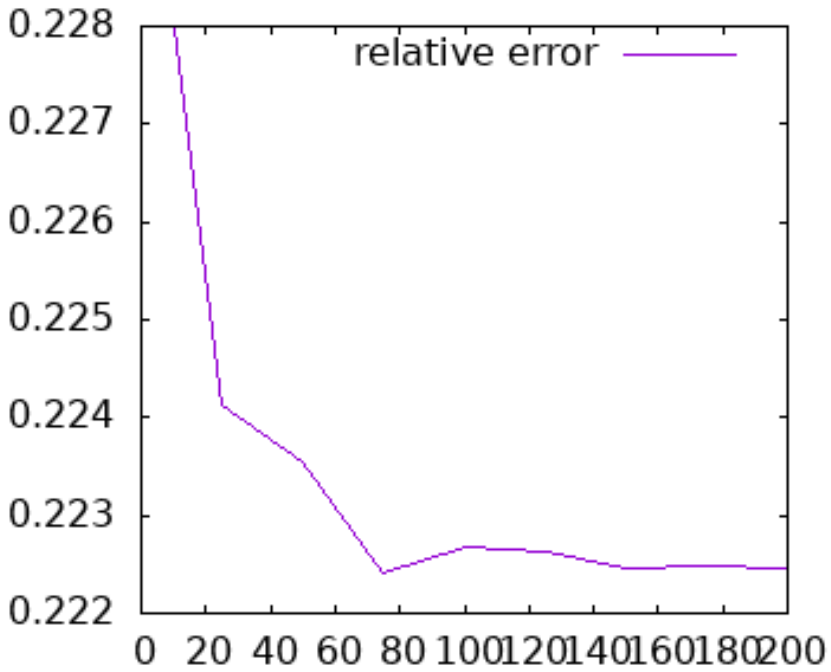
S

	a	b	c	ϵ
1	.58	.20	.22	.0
2	.42	.31	.12	.14
ϵ	0.0	0.5	.66	1

X

Experiments: Comparison Hungarian/Sinkhorn

- For different sizes, we consider 100 random matrices,
- For each matrix we compare the values of $< C, X >$ provided by Sinkhorn with the optimal value provided by the Hungarian algorithm.



- A gap of about 20%.

Experiments: Random matrices

- For LSAPE we have to leverage the cost of ignoring/substituting an element:

$$s_{i,j} = \begin{cases} rand() + 1 & \text{if } i \leq n \wedge j \leq m \\ 0 & \text{if } i = n+1 \wedge j = m+1 \\ hrand() & \text{else} \end{cases}$$

- $h < .5 \Rightarrow s_{i,j} \geq 1 > s_{n+1,j} + s_{i,m+1}$.

$S =$

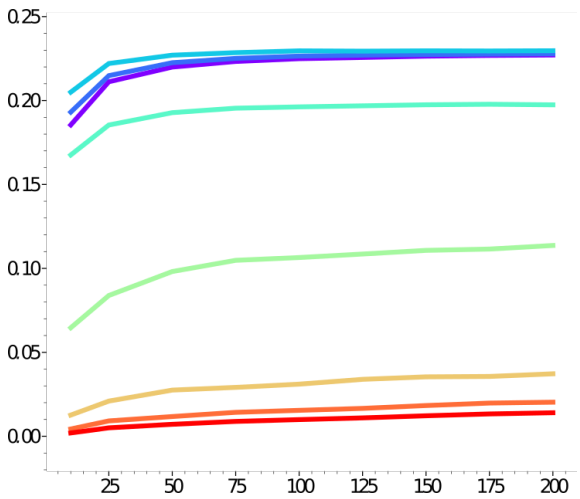
	a	b	c	ϵ
1	1.54	1.29	1.8	.28
2	1.42	1.85	1.09	.24
ϵ	.24	.49	.39	0

h=.5

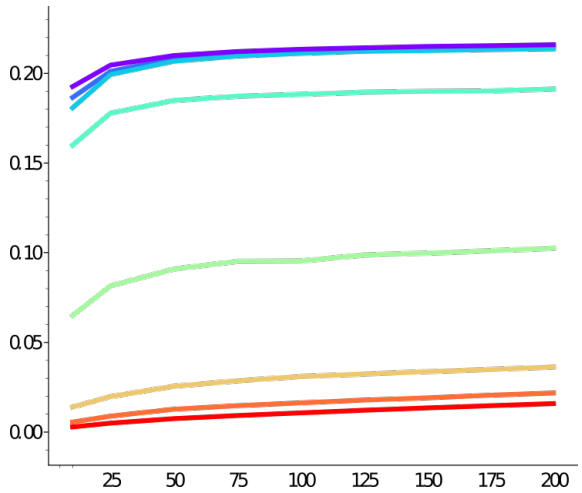
	a	b	c	ϵ
1	1.31	1.9	1.52	1.62
2	1.21	1.19	1.74	1.09
ϵ	.73	1.29	1.88	0

h=2

Experiments (continued)



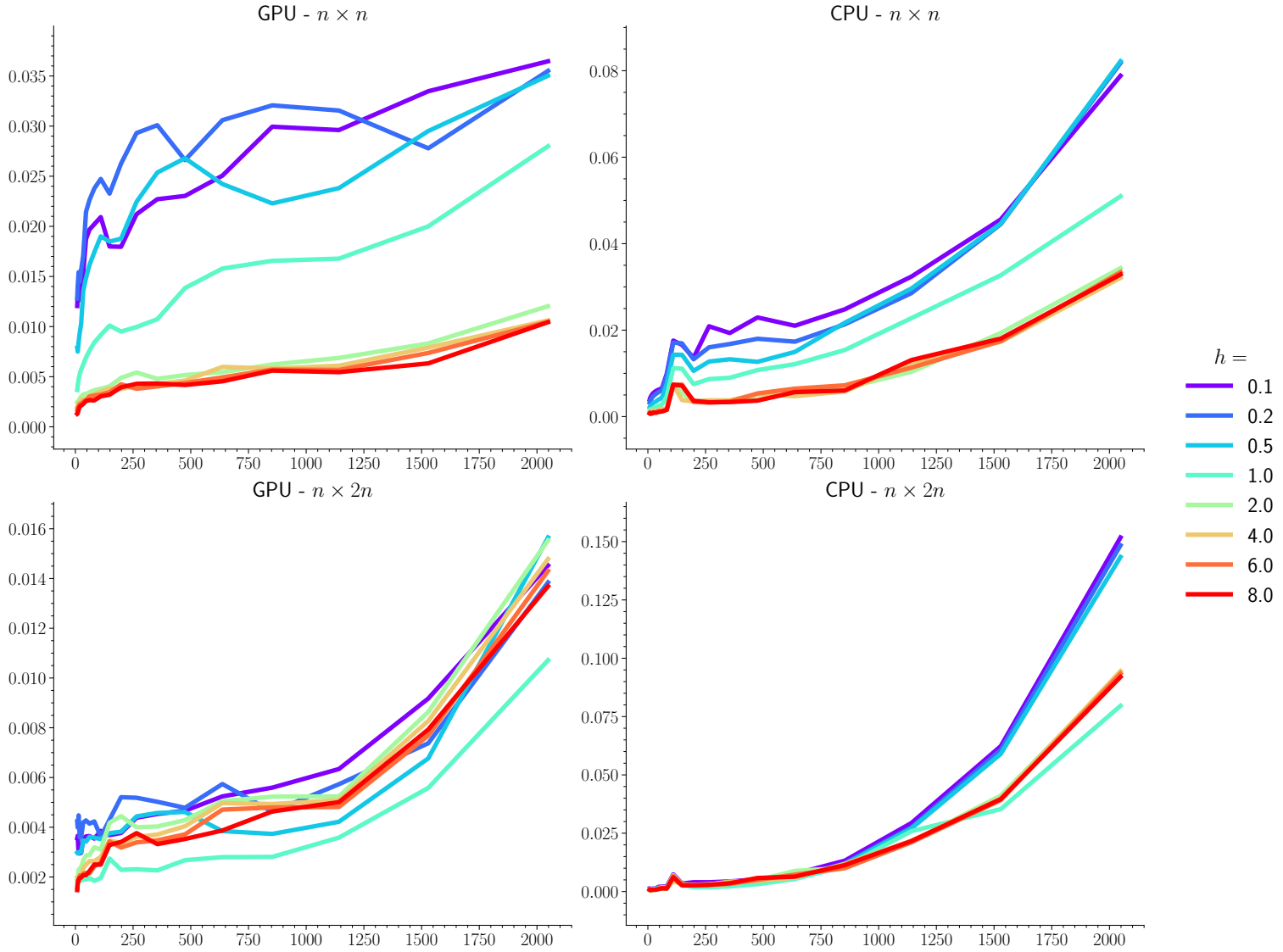
$n \times n$

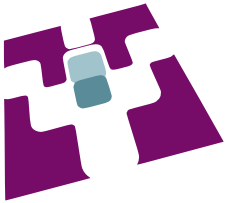


$n \times 2n$

- All values are below or about 20%.
- The error decreases as h increases.

Execution Times

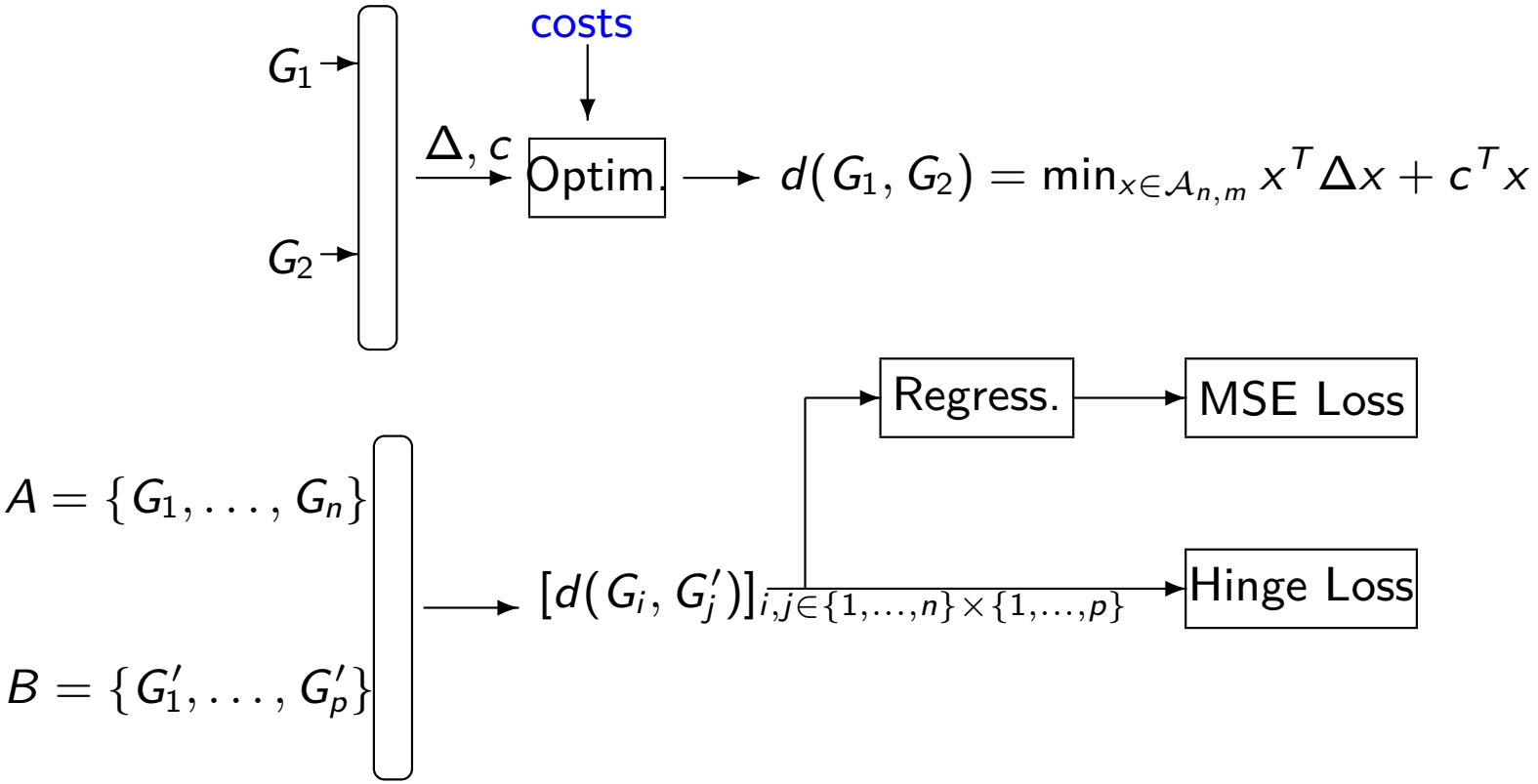


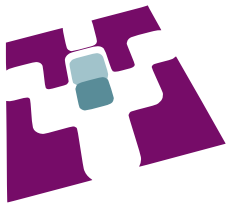


Conclusion

- LSAPE: Allows the computation of mappings with specific costs for ignoring elements.
- Our modified Sinkhorn algorithm allows the insertion of this method into deep learning pipelines.
  - Current Work : Computation of the Graph edit distance.

Application to GED: Metric learning





Bibliography I

Bougleux, S., Gaüzère, B., Blumenthal, D. B., and Brun, L. (2020).

Fast linear sum assignment with error-correction and no cost constraints.

Pattern Recognition Letters, 134:37–45.

Applications of Graph-based Techniques to Pattern Recognition.

Ferradans, S., Papadakis, N., Rabin, J., Peyré, G., and Aujol, J.-F. (2013).

Regularized discrete optimal transport.

In Kuijper, A., Bredies, K., Pock, T., and Bischof, H., editors, *Scale Space and Variational Methods in Computer Vision*, pages 428–439, Berlin, Heidelberg. Springer Berlin Heidelberg.

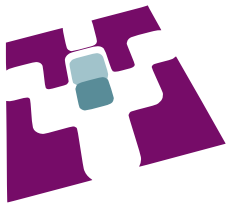
Kuhn, H. W. (1955).

The hungarian method for the assignment problem.

Naval Research Logistics, 2(1-2):83–97.

Séjourné, T., Feydy, J., Vialard, F.-X., Trouvé, A., and Peyré, G. (2019).

Sinkhorn divergences for unbalanced optimal transport.



Some other extensions from transport theory

[Ferradans et al., 2013]:

$$\inf_P < C, P > \text{ subject to } P\mathbb{1}_m = \alpha \text{ and } P^T\mathbb{1}_n \leq \kappa \otimes \beta$$

[Séjourné et al., 2019]:

$$\inf_P < C, P > + D_\varphi(P\mathbb{1}_m, \alpha) + D_\varphi(P^T\mathbb{1}_n, \beta) + \tau D_\varphi(P, \alpha \otimes \beta)$$

LSAPE [Bougleux et al., 2020]:

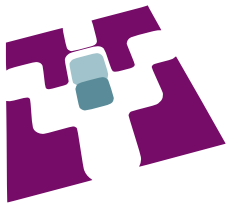
$$\min_P < C, P > \text{ subject to } \begin{cases} Proj_n(P\mathbb{1}_{m+1}) & = & \mathbb{1}_n \text{ and} \\ Proj_m(P^T\mathbb{1}_{n+1}) & = & \mathbb{1}_m \end{cases}$$

Where C is a $(n+1) \times (m+1)$ cost matrix and P is a $(n+1) \times (m+1)$ ϵ-assignment matrix.

From Cost to similarity matrices

- In order to compare our algorithm to LSAPE we have to convert our cost matrices into similarity matrices.
- With Hungarian/Sinkhorn considering $S = c\mathbb{1}_{n+1,m+1} - C$, c large enough provides an equivalent problem.
- For LSAPE we have to consider $S = O - C$ with:

$$O = \left(\begin{array}{c|c} & \vdots \\ 2c\mathbb{1}_{n,m} & c \\ & \vdots \\ \hline \ldots c \ldots & 0 \end{array}\right)$$



Sinkhorn Algorithm

Given a $(n+1) \times (m+1)$ similarity matrix S, let us consider the two series $(x_{1,p}, \ldots, x_{n+1,p})_{p \in \mathbb{N}}$ and $(y_{1,p}, \ldots, y_{m+1,p})_{p \in \mathbb{N}}$ such that $x_{n+1,p} = y_{m+1,p} = 1$ for any p and:

$$\begin{cases} \forall i \in \{1, \ldots, n\} & x_{i,p+1} = \chi_{i,p}^{-1} x_{i,p} \\ \forall j \in \{1, \ldots, m\} & y_{j,p+1} = \gamma_{j,p}^{-1} y_{j,p} \end{cases}$$

with:

$$\begin{cases} \forall i \in \{1, \ldots, n\} & \chi_{i,p} = \sum_{j=1}^{m+1} x_{i,p} a_{i,j} y_{j,p} \\ \forall j \in \{1, \ldots, m\} & \gamma_{j,p} = \sum_{i=1}^{n+1} \chi_{i,p}^{-1} x_{i,p} a_{i,j} y_{j,p} \end{cases}$$

and $\chi_{n+1,p} = \gamma_{m+1,p} = 1$.

The algorithm

function SINKHORN_D1D2($A \in \mathbb{R}^{(n+1)\times(m+1)}$, nb_iter, eps)
    ones_n $\leftarrow \mathbb{1}^{(n+1)}$, ones_m $\leftarrow \mathbb{1}^{(m+1)}$
    y $\leftarrow$ ones_m
    conv $\leftarrow$ `False`, $i \leftarrow 0$
    while $i \leq$ nb_iter and `not` conv **do**
        xp $\leftarrow 1/(A * \text{y})$
        xp[n] $\leftarrow 1$
        yp $\leftarrow 1\ /(A^\top * xp)$
        yp[m] $\leftarrow 1$
        if i≥ 1 **then**
            conv $\leftarrow \|$xp / x - ones_n $\| <$ eps and $\|$ yp / y - ones_m $\| <$ eps
        end if
        x $\leftarrow$ xp
        y $\leftarrow$ yp
        $i \leftarrow i + 1$
    end while
    return diag(x) $*$ A $*$ diag(y)
end function