京都大学
KYOTO UNIVERSITY

Optimal Transport Dictionary Learning and Non-negative Matrix Factorization

Supervisor: Yamamoto Akihiro

Kyoto University Graduate School of Informatics

Antoine Rolet

To Agnès, Jean-Pascal
and Naoko

Acknowledgments

Along the course of my Ph.D. program and the writing of this dissertation, I have had the chance of receiving a lot of help and support, to which I am deeply grateful.

I would like to especially thank my supervisors, Akihiro Yamamoto and Marco Cuturi, who guided me throughout my Ph.D., from the building of my ph.d. project to the submission of this work. I am also very grateful for the warm welcome they extended me at their laboratory, along with everyone working there.

I would also like to acknowledge my all my co-authors: Marco Cuturi, Gabriel Peyré, Hiroshi Sawada, Mathieu Blondel, Viven Seguy, Bharath Bhushan Damodaran and Rémi Flamary, Nicolas Courty. Collaboration with each of them has been elevating both on a personal level and in terms of what they taught me about conducting research.

I am also very grateful to the Ueda Research Laboratory at NTT Communication Science Laboratory in Kyoto, for allowing me to do an internship there and using their data, which led to the publication of our paper on blind source separation.

I would like to thank all the people who helped me write this dissertation, including my supervisor Akihiro Yamamoto, the dissertation reviewers Tatsuya Kawahara and Hisashi Kashima, and my brother Philippe Rolet, for their time, patience and their help towards writing this thesis in its final form.

Lastly I would like to show my deepest gratitude to my recently wedded wife, Naoko Toyoizumi, whose support along the years has been valuable beyond words, and shall not be forgotten.

Abstract

Optimal transport was first formalized by Gaspard Monge in 1781, in order to solve a simple problem: given several piles of dirt and several holes spread on the 2-d plane, how to move the dirt to the holes in a way which minimizes work, where work is defined as the amount of dirt moved multiplied by the length of its travel. Despite the simplicity of its formulation, optimal transport defines a powerful distance, at the cost of a high computational complexity. Recent advances which alleviated this computational burden have led to a rise of its use in machine learning. Indeed, its ability to leverage prior knowledge on data makes it advantageous for many tasks, including image or text classification, dimensionality reduction and musical note transcription to name a few. Additionally, optimal transport can tackle continuous data, and data with different quantizations, allowing it to avoid shortcomings of other distances or divergences in generative models and to be less sensitive to quantization noise.

In this thesis, we study the learning of linear models under an optimal transport cost. More specifically, we address the problems of optimal transport regularized projections on one hand, and of optimal transport dictionary learning on the other. The optimal transport distance lets us incorporate prior knowledge on the type of data, and is less sensitive to *shift* noise—as opposed to the Euclidian distance or other divergences. We show that learning these linear models with an optimal transport loss leads to improvement over classical losses in areas including image processing, natural language processing and sound processing.

We start this monograph by formalizing the optimal transport distance and dictionary learning in Chapter 1, as well as introducing previous works on those matters.

We proceed with the main results of our works in Chapter 2, which addresses our proposed solution to the optimal transport dictionary learning problem. Following previous works on dictionary learning, we solve the problem with alternate optimization on both terms—the dictionary and the weight matrix. We give duality results for both of the sub-problems thus defined. Computing the optimal dictionary and coefficients through a dual problem allow us to get methods which are orders of magnitude faster than primal methods. We show how adding an entropy regularization on the dictionary and weight matrices leads to optimal transport non-negative matrix factorization (NMF), and we discuss the computational implications of optimizing over large datasets. In the experiment section, we compare our method to previous

attempts at optimal transport NMF. We then compare optimal transport NMF to Euclidean and Kullback-Leibler NMF on a topic modeling task. Finally, we showcase how optimal transport can be leveraged to perform cross-domain tasks, bilingual topic modeling for instance.

Building upon the results of Chapter 2, we develop a method for supervised speech blind source separation (BSS) in Chapter 3. Optimal transport allows us to design and leverage a cost between short-time Fourier transform (STFT) spectrogram frequencies, which takes into account how humans perceive sound. We give empirical evidence that using our proposed optimal transport NMF leads to perceptually better results than NMF with other losses, for both isolated voice reconstruction and speech denoising using BSS. Finally, we demonstrate how to use optimal transport for cross-domain sound processing tasks, where frequencies represented in the input spectrograms may be different from one spectrogram to another.

Lastly, we take a step back in Chapter 4 and focus on the coefficient step of the dictionary learning process. This defines what we call optimal transport regularized projections. Noting that pass filters and coefficient shrinkage can be seen as regularized projections under the Euclidean metric, we tackle the task of extending these methods to the optimal transport distance. This however requires us to solve an ℓ_1-norm regularized projection, which cannot be addressed with the duals defined in Chapter 2. We show that in the case of an invertible dictionary, we can extend these dual, which allows us to compute optimal transport coefficient shrinkage. We give experimental evidence that using the optimal transport distance instead of the Euclidean distance for filtering and coefficient shrinkage leads to reduced artifacts and improved denoising results.

Contents

List of Theorems

List of Figures

List of Tables

List of Abbreviations

- ALS: alternating least squares
- BSS: blind source separation
- E: Euclidean
- IS: Itakura-Saito
- KL: Kullback-Leibler
- LP: linear program
- NMF: non-negative matrix factorization
- OT: optimal transport
- PLSI: Probabilistic latent semantic indexing
- SAR: signal-to-artefact ratio
- SDR: signal-to-distortion ratio
- SIR: signal-to-interference ratio
- SNR: signal-to-noise ratio
- STFT: short-time Fourier transform
- SVD: singular value decomposition

Chapter 1

Introduction

This thesis addresses the problem of dictionary learning and non-negative matrix factorization (NMF) with an optimal transport cost. Dictionary learning and NMF are matrix approximation problems, where the approximation is sought in the form of the product of two matrices: $M \simeq D\lambda$. Our focus throughout this work will be on what $\simeq$ represents. More specifically we will answer the questions: how to solve these problem when $\simeq$ means "close with respect to the optimal transport distance", and why would we want to do that.

The optimal transport distance, *a.k.a* the Wasserstein distance or the earth mover's distance, compares two vectors by computing an *optimal* way to fit one into the other. Vectors are thus compared globally, instead of coordinate-wise as is the case with the Euclidean distance. Moreover, we can incorporate prior knowledge on the signal or data being compared by defining what is an *optimal* way to fit a vector into another.

We illustrate this in Figure 1.1, which compares the optimal transport distance with the Euclidean distance when applied to the spectogram of a note played on a musical instrument. Synthetic musical notes are generated by putting weight on a fundamental, and exponentially decreasing weights on its harmonics and sub-harmonics, and finally convoluting with a Gaussian. Figure 1.1 shows examples of such musical notes, and plots the different distances from the note of fundamental 0.95kHz (red line on the left plot) to the note of fundamental 0.95kHz+σ, as functions of σ. With optimal transport, what is measured can be thought of as the cumulative distance between each spike, whereas the Euclidean distance is almost an indicator of whether the spike happen at the same frequency. In other words, the Euclidean distance compares these spectrograms *vertically*, when optimal transport compares them *horizontally*, making it less sensible to noise due to tuning, Doppler effect, coming from a different instrument and so on.

Optimal transport as a loss in optimization problems has been used with applications such as classification [Kusner et al., 2015, Frogner et al., 2015], image generation [Arjovsky et al., 2017, Seguy et al., 2018] and domain adaptation [Redko et al.,

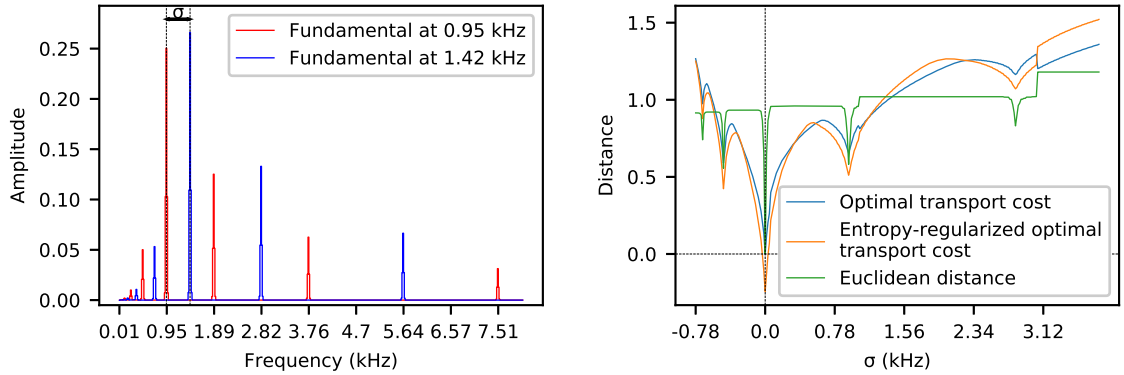


Figure 1.1: Comparison between optimal transport and the Euclidean distance on musical note spectrograms. Left: examples of two musical note spectrograms where the fundamentals' frequencies are separated by σ; right: Euclidean or optimal transport distance from the red spectogram to the blue one, plotted against σ. The red spectogram is fixed and the blue one's fundamental varies according to σ.

Taken from Rolet et al. [2018]

2019]. It has also been used to tackle image processing problems, including Tartavel et al. [2016] for texture synthesis and image reconstruction, Rabin and Papadakis [2015] for foreground extraction or Solomon et al. [2015] for image interpolation.

To the best of our knowledge, the first optimal transport for dictionary learning method was developed by Sandler and Lindenbaum [2009] in the context of NMF. Due to the high computational cost of the problem however, they had to consider a proxy for optimal transport [Shirdhonkar and Jacobs, 2008] which only gives a lax approximation that cannot be controlled to get infinitely close to exact optimal transport. This approximation also only works on special cases of optimal transport, which limits the range of its applications. We expanded their work by proposing to use instead regularized optimal transport in Rolet et al. [2016]. This allows us to get methods which are orders of magnitude faster, and for which the user can control closeness to the exact optimal transport through the regularization strength. It also allows us to perform "cross-domain" dictionary learning, which was not possible using the method of Sandler and Lindenbaum [2009]. We later expanded and refined these method, with an application to Blind Source Separation in Rolet et al. [2018] and optimal transport coefficient shrinkage in Rolet and Seguy [2021]. This thesis is based on these three works.

We start this Chapter by formalizing optimal transport and studying some of its properties, in particular regarding optimization. We then move onto discussing previous works on dictionary learning and NMF, and introducing the main method we use in this work to solve the optimal transport dictionary learning problem.

Notations

We denote matrices in uppercase, vectors in bold lowercase and scalars in lower-case. If M is a matrix, $M^\top$ denotes its transpose, $\boldsymbol{m}_i$ its ith column and $\boldsymbol{m}_{:j}$ its jth row. $\mathbf{1}_n$ denotes the all-ones vector in $\mathbb{R}^n$; when the dimension can be deduced from context we simply write $\mathbf{1}$. For two matrices A and B of the same size, we denote their inner product $\langle A, B\rangle \coloneqq \operatorname{tr}\left(A^\top B\right)$, and their element-wise product (*resp.* quotient) as $A \odot B$ (*resp.* $\frac{A}{B}$). Σ_n is the set of n-dimensional histograms: $\Sigma_n \coloneqq \left\{q \in \mathbb{R}_+^n \mid \langle q, \mathbf{1}\rangle = 1\right\}$. If A is a matrix, A^+ is its Moore-Penrose pseudo-inverse. Exponentials and logarithms are applied element-wise to matrices and vectors.

1.1 Optimal Transport

Optimal transport can be seen as a generalization of optimal assignment between two sets of points. We start by formalizing the definition of optimal transport from this optimal assignment point of view. We then discuss a few properties of optimal transport and equivalent problems or formulations.

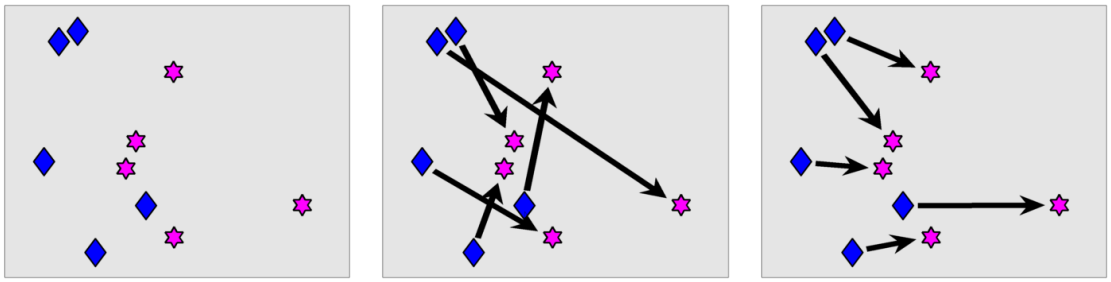
1.1.1 Optimal Assignment

Optimal assignment is the problem, given two sets of points of same size, of finding a coupling (*a.k.a.* assignment) between these sets which minimizes the total distance between points and their counterpart in the coupling.

Formally, we are given finite two sets of n points $\mathcal{A} = \{\boldsymbol{a^i} \in \Omega | 1 \leq i \leq n\}, \mathcal{B} = \{\boldsymbol{b^i} \in \Omega | 1 \leq i \leq n\}$ in a space Ω and a cost of assigning points one to another in the form of a real valued, non-negative bi-function d on Ω. The goal is to find a bijective mapping from $\mathcal{A}$ to $\mathcal{B}$ which solves

$$\min_f \sum_{i=1}^{n} d(\boldsymbol{a^i}, f(\boldsymbol{a^i})). \tag{1.1}$$

Figure 1.2 shows an example of optimal and non-optimal assignment between two clouds of points. If (Ω, d) is a Euclidean space, the cost of an assignment in the Figure is the cumulative length of the arrows. The segments defined by each coupling cannot cross each other in an optimal assignment, otherwise switching two such couplings would reduce the total distance.



(a) Sets $\mathcal{A}$ and $\mathcal{B}$ *(b) Non-optimal assignment* *(c) Optimal assignment*

Figure 1.2: Optimal assignment between two sets of points.

Optimal assignments always exist, since they are the solution of an optimization over a non-empty finite set, however they are not necessarily unique. Indeed in the

complex plane let $\mathcal{A} = \{1, e^{i\pi}\}$ and $\mathcal{B} = \{e^{i\frac{\pi}{2}}, e^{i\frac{3\pi}{2}}\}$. The four points are extremities of a square, grouped along diagonals. There are two assignments, and both have the same cost: $2\sqrt{2}$.

We can rewrite Problem 1.1 as an integer program. Since f is a bijection between two finite sets of same size, we can represent it as a permutation matrix M, that is an $n \times n$ non-negative integer matrix which is all zeros except for exactly one entry per row and column whose value is one. The problem can then be rewritten:

$$\min_{\substack{M \in \mathbb{N}_+^{n \times n} \\ M\mathbf{1}=\mathbf{1} \\ M^\top \mathbf{1}=\mathbf{1}}} \sum_{i,j=1}^{n} M_{i,j} d(\boldsymbol{a^i}, \boldsymbol{b^j}). \tag{1.2}$$

Denoting D the matrix of distances between $\mathcal{A}$ and $\mathcal{B}$, that is $D_{i,j} = d(\boldsymbol{a^i}, \boldsymbol{b^j})$, we can further rewrite the problem as follows:

$$\min_{\substack{M \in \mathbb{N}_+^{n \times n} \\ M\mathbf{1}=\mathbf{1} \\ M^\top \mathbf{1}=\mathbf{1}}} \sum_{i,j=1}^{n} M_{i,j} D_{i,j} = \min_{\substack{M \in \mathbb{N}_+^{n \times n} \\ M\mathbf{1}=\mathbf{1} \\ M^\top \mathbf{1}=\mathbf{1}}} \langle M, D \rangle . \tag{1.3}$$

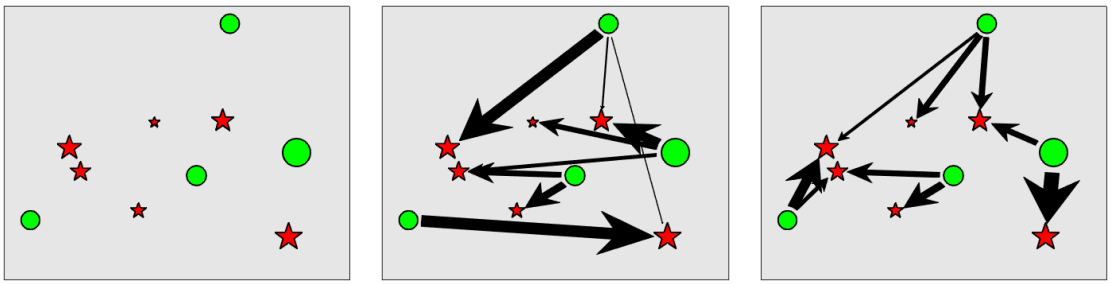
Although there is no known algorithm to solve integer programming in polynomial time in general, optimal assignment can be solved by specific algorithms such as the Hungarian algorithm, which with our notations solves the problem in $O(n^3)$[Tomizawa, 1971].

1.1.2 Exact Optimal Transport

1.1.2.1 Definition

Optimal transport is the generalization of optimal assignment where both sets of points do not have the same cardinal, and where each point has a non-negative weight. The goal is now to assign the weight of points of one set to the weight of points in an optimal way. Figure 1.3 shows example of optimal and non-optimal transport assignments, where the cost of an assignment can be thought of as the sum of the length of the arrows times their width.

We now have two *weighted clouds of points* $\mathcal{A} = \{(x_i \in \mathbb{R}_+, \boldsymbol{a^i} \in \Omega) | 1 \leq i \leq m\}$ and $\mathcal{B} = \{(y_i \in \mathbb{R}_+, \boldsymbol{b^i} \in \Omega) | 1 \leq i \leq n\}$. Since we want to assign the weights in $\mathcal{A}$ to the weight in $\mathcal{B}$, we need the total weight in each cloud to be the same: $\sum_{i=1}^{m} x_i = \sum_{i=1}^{n} y_i$, which can be rewritten more concisely as $\|\boldsymbol{x}\|_1 = \|\boldsymbol{y}\|_1$ since both vectors are in the non-negative orthant. Similarly to Problem 1.3, we will represent an assignment as a matrix T (for transportation matrix). T_{ij} represent the amount of



(a) Weighted sets $\mathcal{A}$ and $\mathcal{B}$. *(b) Non-optimal transport* *(c) Optimal transport*

Figure 1.3: Optimal transport between two weighted sets of point.

weight assigned from $(x_i, \boldsymbol{a^i})$ to $(y_j, \boldsymbol{b^j})$, we thus need to have $T \in \mathbb{R}_+^{n\times m}$. For any j, $\sum_{i=1}^n T_{i,j}$ is the total amount of weight assigned to $(y_j, \boldsymbol{b^j})$, thus $\sum_{i=1}^n T_{i,j} = y_j$. We can rewrite this for all columns as $T\mathbf{1} = \boldsymbol{y}$. Similarly $T^\top \mathbf{1} = \boldsymbol{x}$. We denote $\mathcal{U}(\boldsymbol{x}, \boldsymbol{y})$ the set of transportation matrices from the weight vector $\boldsymbol{x}$ to the weight vector $\boldsymbol{y}$:

$$\mathcal{U}(\boldsymbol{x}, \boldsymbol{y}) \coloneqq \left\{ T \in \mathbb{R}_+^{n\times m} \middle| \begin{matrix} T^\top \mathbf{1} = \boldsymbol{x} \\ T\mathbf{1} = \boldsymbol{y}. \end{matrix} \right\} \tag{1.4}$$

As in the assignment problem, the cost of pairing to points is proportional to their distance, but it is now also proportional to the weight assigned, and we aim at solving

$$\min_{T\in\mathcal{U}(\boldsymbol{x},\boldsymbol{y})} \sum_{i=1}^{n}\sum_{j=1}^{m} T_{i,j} d(\boldsymbol{a^i}, \boldsymbol{b^j}). \tag{1.5}$$

Rewriting this problem in matrix form, we get the following optimal transport problem

$$\min_{T\in\mathcal{U}(\boldsymbol{x},\boldsymbol{y})} \langle T, D \rangle . \tag{1.6}$$

In the early formulation of optimal transport [Monge, 1781], $\mathcal{A}$ represented piles of earth and $\mathcal{B}$ represented holes to be filled, hence the name *earth mover's distance* coined by Rubner et al. [1998]. However the mathematical problem itself is symmetric and the arrows in Figure 1.3 could be directed either way.

Note:

In the case where all weights are integer, the optimal transport problem can be cast into an optimal assignment problem. Indeed we can simply replace each weighted

point by a number of points equal to its weight, at the same position. Furthermore, in the case where all weights are rational numbers, we can again get back to an optimal assignment problem by multiplying all weights by their least common denominator in order to get integer weights. However there is almost no practical use to these equivalences, since in most cases it would greatly increase the number of points in the problem, and algorithms solving the optimal assignment problem are not faster than those solving optimal transport.

In this work, the points of the clouds of points $\mathcal{A}$ and $\mathcal{B}$ are fixed and defined by the application at hand in the form of the matrix D, but their weights $\boldsymbol{x}$ and $\boldsymbol{y}$ may vary. For the remainder of this monograph we thus define the optimal transport between vectors of weights as

$$\mathrm{OT}(\boldsymbol{x}, \boldsymbol{y}) = \begin{cases} +\infty & \text{if } U(\boldsymbol{x}, \boldsymbol{y}) = \emptyset \\ \min\limits_{T \in U(\boldsymbol{x}, \boldsymbol{y})} \langle T, C \rangle & \text{otherwise.} \end{cases} \tag{1.7}$$

Note that $U(\boldsymbol{x}, \boldsymbol{y})$ is a bounded convex polyhedron, so as long as it is not empty, the optimal transport problem has a solution. As a result, if $\|\boldsymbol{x}\|_1 \neq \|\boldsymbol{y}\|_1$, $U(\boldsymbol{x}, \boldsymbol{y})$ is empty and $\mathrm{OT}(\boldsymbol{x}, \boldsymbol{y}) = \infty$. On the other hand, if $\|\boldsymbol{x}\|_1 = \|\boldsymbol{y}\|_1$, $\frac{\boldsymbol{x}\boldsymbol{y}}{\|\boldsymbol{y}\|_1^2} \in U(\boldsymbol{x}, \boldsymbol{y})$ and $\mathrm{OT}(\boldsymbol{x}, \boldsymbol{y})$ is a real-valued number. As was the case for optimal assignment, a solution of the optimal transport problem is not always unique.

1.1.2.2 Optimal Transport as a Minimum Cost Flow

The optimal transport problem can be seen as a minimum cost flow problem on a bipartite graph. A bipartite graph is a graph where vertices are separated into two sets, and there can be an edge between two nodes only if they are not in the same set. Let $V = \{\mathcal{X}_1, \ldots, \mathcal{X}_m, \mathcal{Y}_1, \ldots, \mathcal{Y}_n, \}$ be a set of vertices, and $E = \{(\mathcal{X}_i, \mathcal{Y}_j) | 1 \leq i \leq m,\ 1 \leq j \leq n\}$ be a set of edges. For $1 \leq i \leq m$, $1 \leq j \leq n$, we assign a flow source value of x_i to $\mathcal{X}_i$, a flow sink value of y_j to $\mathcal{Y}_j$ and a cost D_{ij} to the edge $(\mathcal{X}_i, \mathcal{Y}_j)$. Then $\mathcal{G} = (V, E)$ is a bipartite graph, with $n + m$ vertices and nm edges, and the optimal transport between $\boldsymbol{x}$ and $\boldsymbol{y}$ can be found by minimizing the cost of the flow on this graph. Figure 1.4 shows such a graph representing an optimal transport problem.

This representation of optimal transport as a minimum cost flow allows us to solve it with specialized algorithms, for instance Orlin [1993] with $\mathcal{O}(n^3 \log n)$ complexity, assuming $n = m$. The network simplex [Orlin et al., 1993], a specialized version of the simplex, is another method of choice, which has been used in Rubner et al. [1998] and is the algorithm used in the python package pot [Flamary and Courty, 2017].

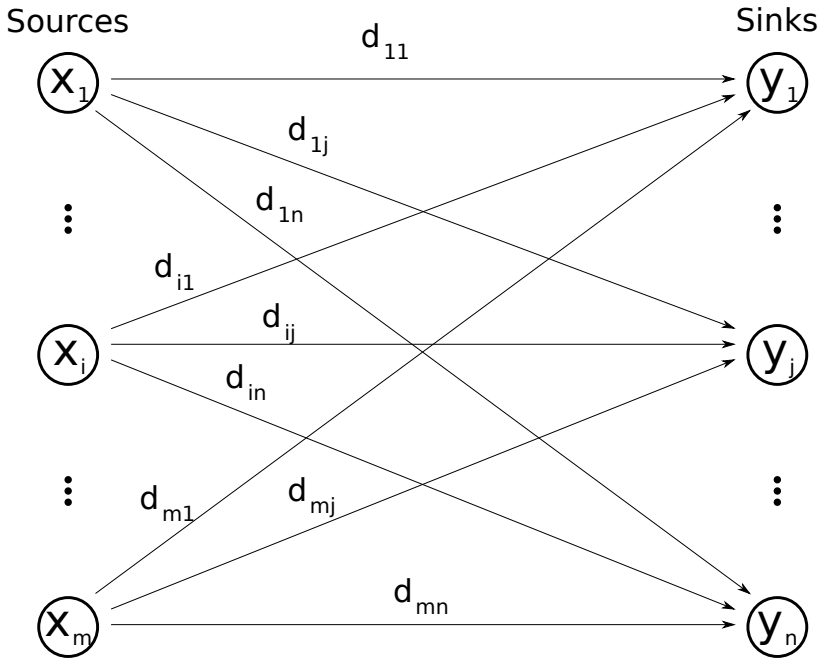


Figure 1.4: Representation of the optimal transport problem as a minimum cost flow on a bipartite graph.

1.1.2.3 Properties

In this section we discuss properties of the optimal transport as a function of its arguments $\boldsymbol{x}, \boldsymbol{y}$. In particular, since we are interested in minimizing objectives which include the optimal transport, we will show that it is convex and see how to compute its sub-gradients.

Dual Problem. Suppose that $U(\boldsymbol{x}, \boldsymbol{y})$ is non-empty. Let us consider the Lagrangian of the optimal transport problem

$$\begin{aligned}\mathcal{L}(T, \boldsymbol{u}, \boldsymbol{v}) &= \langle T, D\rangle - \left\langle T^\top \mathbf{1} - \boldsymbol{x}, \boldsymbol{u}\right\rangle - \langle T\mathbf{1} - \boldsymbol{y}, \boldsymbol{v}\rangle \\ &= \langle T, D\rangle - \left\langle T^\top, \boldsymbol{u}\mathbf{1}^\top\right\rangle + \langle \boldsymbol{x}, \boldsymbol{u}\rangle + \left\langle T, \boldsymbol{v}\mathbf{1}^\top\right\rangle + \langle \boldsymbol{y}, \boldsymbol{v}\rangle \\ &= \left\langle T, D - \mathbf{1}\boldsymbol{u}^\top - \boldsymbol{v}\mathbf{1}^\top\right\rangle + \langle \boldsymbol{x}, \boldsymbol{u}\rangle + \langle \boldsymbol{y}, \boldsymbol{v}\rangle .\end{aligned}$$

The Lagrange dual is then

$$\max_{\boldsymbol{u},\boldsymbol{v}} \min_{T\geq 0} \mathcal{L}(T, \boldsymbol{u}, \boldsymbol{v}) = \max_{\boldsymbol{u},\boldsymbol{v}} \min_{T\geq 0} \left\langle T, D - \mathbf{1}\boldsymbol{u}^\top - \boldsymbol{v}\mathbf{1}^\top\right\rangle + \langle \boldsymbol{x}, \boldsymbol{u}\rangle + \langle \boldsymbol{y}, \boldsymbol{v}\rangle . \tag{1.8}$$

Suppose that for some i, j, $D_{ij} - u_j - v_i < 0$, then $\min_{T\geq 0} \mathcal{L}(T, \boldsymbol{u}, \boldsymbol{v}) = -\infty$, we thus have

$$\max_{\boldsymbol{u},\boldsymbol{v}} \min_{T\geq 0} \mathcal{L}(T,\boldsymbol{u},\boldsymbol{v}) = \max_{\substack{\boldsymbol{u},\boldsymbol{v} \\ D-\mathbf{1}\boldsymbol{u}^\top-\boldsymbol{v}\mathbf{1}^\top\geq 0}} \langle \boldsymbol{x},\boldsymbol{u}\rangle + \langle \boldsymbol{y},\boldsymbol{v}\rangle \,. \tag{1.9}$$

Let $\boldsymbol{u}$ and $\boldsymbol{v}$ be a pair of feasible solutions, *i.e.* such that $D - \mathbf{1}\boldsymbol{u}^\top - \boldsymbol{v}\mathbf{1}^\top \geq 0$, for any real α, $\boldsymbol{u} + \alpha\mathbf{1}$ and $\boldsymbol{v} - \alpha\mathbf{1}$ is also a pair of feasible solutions, for which the objective is $\langle \boldsymbol{x},\boldsymbol{u}\rangle + \langle \boldsymbol{y},\boldsymbol{v}\rangle + \alpha(\langle \boldsymbol{x},\mathbf{1}\rangle - \langle \boldsymbol{y},\mathbf{1}\rangle) = \langle \boldsymbol{x},\boldsymbol{u}\rangle + \langle \boldsymbol{y},\boldsymbol{v}\rangle + \alpha(\|\boldsymbol{x}\|_1 - \|\boldsymbol{y}\|_1)$. As a result if $\|\boldsymbol{x}\|_1 \neq \|\boldsymbol{y}\|_1$, the dual problem is unbounded and its solution is infinite.

Since the optimal transport problem is a linear program with feasible solutions if $U(\boldsymbol{x},\boldsymbol{y})$ is non-empty, strong duality applies and

$$\mathrm{OT}(\boldsymbol{x},\boldsymbol{y}) = \max_{\substack{\boldsymbol{u},\boldsymbol{v} \\ D-\mathbf{1}\boldsymbol{u}^\top-\boldsymbol{v}\mathbf{1}^\top\geq 0}} \langle \boldsymbol{x},\boldsymbol{u}\rangle + \langle \boldsymbol{y},\boldsymbol{v}\rangle \,. \tag{1.10}$$

The maximization problem in Equation 1.10 is the dual of the optimal transport problem. We can verify here that since the optimal transport problem is a linear program, so is its dual. Note that if $(\boldsymbol{u},\boldsymbol{v})$ is a feasible solution of the dual problem, so is $(\boldsymbol{u}+\gamma\mathbf{1},\boldsymbol{v}-\gamma\mathbf{1})$ for all $\gamma \in \mathbb{R}$. The new objective is then $\langle \boldsymbol{x},\boldsymbol{u}\rangle + \langle \boldsymbol{y},\boldsymbol{v}\rangle + \gamma\,\langle \mathbf{1},\boldsymbol{x}-\boldsymbol{y}\rangle$. As a result if $\|\boldsymbol{x}\|_1 \neq \|\boldsymbol{y}\|_1$, $\langle \mathbf{1},\boldsymbol{x}-\boldsymbol{y}\rangle \neq 0$ and the objective is unbounded. This is consistent with our definition in Equation 1.7.

Convexity. With the dual form of the optimal transport problem, we can easily prove that $\mathrm{OT}(\boldsymbol{x},\cdot)$ is convex for all $\boldsymbol{x} \geq 0$.

Let $\boldsymbol{x} \geq 0$, $\boldsymbol{y}^{(1)} \geq 0$, $\boldsymbol{y}^{(2)} \geq 0$ and $0 \leq \alpha \leq 1$, we will show that $\mathrm{OT}(\boldsymbol{x},\alpha\boldsymbol{y}^{(1)} + (1-\alpha)\boldsymbol{y}^{(2)}) \leq \alpha\,\mathrm{OT}(\boldsymbol{x},\boldsymbol{y}^{(1)}) + (1-\alpha)\,\mathrm{OT}(\boldsymbol{x},\boldsymbol{y}^{(1)})$.

Let $\boldsymbol{u}^{(1)}$ (resp. $\boldsymbol{u}^{(2)}$) and $\boldsymbol{v}^{(1)}$ (resp. $\boldsymbol{v}^{(2)}$) be optimal solutions of the dual of the optimal transport problem between $\boldsymbol{x}$ and $\boldsymbol{y}^{(1)}$ (resp. $\boldsymbol{y}^{(2)}$). Since the constraint of the dual problem are linear and do not depend on $\boldsymbol{y}^{(1)}$ or $\boldsymbol{y}^{(2)}$, $\alpha\boldsymbol{u}^{(1)} + (1-\alpha)\boldsymbol{u}^{(2)}$ and $\alpha\boldsymbol{v}^{(1)} + (1-\alpha)\boldsymbol{v}^{(2)}$ are feasible solutions of the dual of the optimal transport problem between $\boldsymbol{x}$ and $\alpha\boldsymbol{y}^{(1)} + (1-\alpha)\boldsymbol{y}^{(2)}$. Thus

$$\begin{aligned}\mathrm{OT}(\boldsymbol{x},\alpha\boldsymbol{y}^{(1)} + (1-\alpha)\boldsymbol{y}^{(2)}) &\leq \left\langle \boldsymbol{x},\alpha\boldsymbol{u}^{(1)} + (1-\alpha)\boldsymbol{u}^{(2)}\right\rangle + \left\langle \boldsymbol{y},\alpha\boldsymbol{v}^{(1)} + (1-\alpha)\boldsymbol{v}^{(2)}\right\rangle \\ &\leq \alpha\,\mathrm{OT}(\boldsymbol{x},\boldsymbol{y}^{(1)}) + (1-\alpha)\,\mathrm{OT}(\boldsymbol{x},\boldsymbol{y}^{(1)})\end{aligned}$$

$\mathrm{OT}(\boldsymbol{x},\cdot)$ is thus convex. Convexity of the optimal transport with respect to the weights $\boldsymbol{y}$ is crucial to this work, since we are interested in minimizing an objective which includes the optimal transport.

Convex Conjugate. The convex conjugate of a function is a useful tool to derive duals for optimization problems. In this work we are interested in the convex conjugate of the optimal transport distance with respect to its second variable, defined as follow:

$$\mathrm{OT}^*(\boldsymbol{x},\boldsymbol{z}) \coloneqq \max_{\boldsymbol{y}} \langle \boldsymbol{z},\boldsymbol{y}\rangle - \mathrm{OT}(\boldsymbol{x},\boldsymbol{y})$$

.

The interest of using duals which involve the convex conjugate of optimal transport is that it is easy to compute, and doesn't require solving an optimization problem:

Theorem 1.1.1 (Convex conjugate)**.** *Let* $\boldsymbol{x} \in \mathbb{R}_+^m$, $\boldsymbol{z} \in \mathbb{R}^n$,

$$\mathrm{OT}^*(\boldsymbol{x}, \boldsymbol{z}) = \sum_{i=0}^{m} x_i \max_j D_{ij} - z_j$$

.

Proof. Let $\boldsymbol{x} \in \mathbb{R}_+^m$, $\boldsymbol{z} \in \mathbb{R}^n$,

$$\begin{aligned}
\mathrm{OT}^*(\boldsymbol{x}, \boldsymbol{z}) &= \max_{\boldsymbol{y}} \langle \boldsymbol{z}, \boldsymbol{y} \rangle - \mathrm{OT}(\boldsymbol{x}, \boldsymbol{y}) \\
&= \max_{\boldsymbol{y}} \langle \boldsymbol{z}, \boldsymbol{y} \rangle - \min_{\substack{T \in \mathbb{R}_+^{n \times m} \\ T\mathbf{1} = \boldsymbol{y} \\ T^\top \mathbf{1} = \boldsymbol{x}}} \langle T, D \rangle \\
&= \max_{\boldsymbol{y}} \langle \boldsymbol{z}, \boldsymbol{y} \rangle + \max_{\substack{T \in \mathbb{R}_+^{n \times m} \\ T\mathbf{1} = \boldsymbol{y} \\ T^\top \mathbf{1} = \boldsymbol{x}}} - \langle T, D \rangle \\
&= \max_{\substack{\boldsymbol{y} \in \mathbb{R}_+^m \\ T \in \mathbb{R}_+^{n \times m} \\ T\mathbf{1} = \boldsymbol{y} \\ T^\top \mathbf{1} = \boldsymbol{x}}} \langle \boldsymbol{z}, \boldsymbol{y} \rangle - \langle T, D \rangle \\
&= \max_{\substack{T \in \mathbb{R}_+^{n \times m} \\ T^\top \mathbf{1} = \boldsymbol{x}}} \langle \boldsymbol{z}, T\mathbf{1} \rangle - \langle T, D \rangle \\
&= \max_{\substack{T \in \mathbb{R}_+^{n \times m} \\ T^\top \mathbf{1} = \boldsymbol{x}}} \left\langle \boldsymbol{z}\mathbf{1}^\top - D, T \right\rangle .
\end{aligned}$$

The last line shows that the convex conjugate of optimal transport corresponds to an optimal transport with cost matrix $D - \boldsymbol{z}\mathbf{1}^\top$, and with one constraint relaxed. As such, for each column i, an optimizer $T^\star$ needs to assign maximum weight, *i.e.* x_i, to $\operatorname{argmax}_j (\boldsymbol{z}\mathbf{1}^\top - D)_{ij} = \operatorname{argmax}_j z_j - D_{ij}$. We thus have

$$\mathrm{OT}^*(\boldsymbol{x}, \boldsymbol{z}) = -\sum_{i=0}^{m} x_i \max_j z_j - D_{ij}$$

.

□

Although the value of $\mathrm{OT}^*(\boldsymbol{x}, \cdot)$ is relatively easy to compute compared to $\mathrm{OT}(\boldsymbol{x}, \cdot)$, neither function is differentiable and we are restricted to sub-gradient methods for solving optimization problems which involve these functions.

Optimal Transport Defines a Distance. So far, even though we have talked about the optimal transport *distance*, we have not described under which conditions optimal transport defines a distance between weighted clouds of points.

When $n = m$ and the cost matrix D is the p-th power ($p \geqslant 1$) of a distance matrix, i.e. $d_{i,j} = \ell(\boldsymbol{y}_i, \boldsymbol{y}_j)^p$ for some $(\boldsymbol{y}_i)$ in a metric space (Ω, ℓ), then $\mathrm{OT}(\cdot, \cdot)^{1/p}$ is a distance on the set of vectors in $\mathbb{R}_+^n$ [Villani, 2003, Theorem 7.3]. This distance is sometimes called the Wasserstein distance of order p.

In this work, the fact that optimal transport is a distance or not is not of practical importance: dictionary learning and NMF are often used with the KL divergence or the Itakura-Saito (IS) divergence. Indeed, neither in our topic modeling experiment of Chapter 2 nor in our BSS experiments of Chapter 3 does the cost matrix satisfy the conditions we just described.

1.1.2.4 Examples

Distance Between Bags-of-features. The optimal transport distance can be used to get a distance on bag-of-features representations of data. A simple example of that is text data. We can represent a text as the set of the words it contains, and give to each word the weight corresponding to its frequency in the text. This representation is lossy, as one usually cannot reconstruct a text given only the frequency of each of its words, but it is a common way to process texts[Berry et al., 1995, Lee and Seung, 2001]. If we can get a distance between words then we can compute an optimal transport between texts. Kusner et al. [2015] used words embeddings [Mikolov et al., 2013] to map words to a Euclidean space and use its distance as the matrix D. Using optimal transport as a distance between texts, which they call the *word mover's distance*, they showed improved k-nearest neighbor classification results over other distances. Figure 1.5 illustrates the idea of mapping words to a Euclidean space and using optimal transport, on two simple sentences.

We use this idea in Chapter 2 to perform topic modeling, and we further propose to use bilingual word embeddings to get a distance between text in different languages, and perform cross-language topic modeling.

Distance Between images. Optimal transport can also be used to compute distances between either grayscale or color images for which we treat each color component independently. In any case, $\boldsymbol{x}$ is a vector representing intensity levels for each pixel of an $n \times m$ image, *i.e.* x_i is the intensity of the pixel located at coordinates $\mathcal{C}(i) \coloneqq (\lfloor i/m \rfloor, i \,\%\, m)$ in the image, where $\lfloor \cdot \rfloor$ is the integer part and $\%$ is the remainder operator. We use as the cost matrix C the matrix of pairwise squared Euclidean distances between the locations of the pixels, that is $c_{ij} = \|\mathcal{C}(i) - \mathcal{C}(j)\|_2^2$. Among other applications, the optimal transport distance between images obtained with this definition of C has been used for computing *optimal transport barycenters* [Cuturi and Doucet, 2014, Solomon et al., 2015] and *Wasserstein principal component analy-*

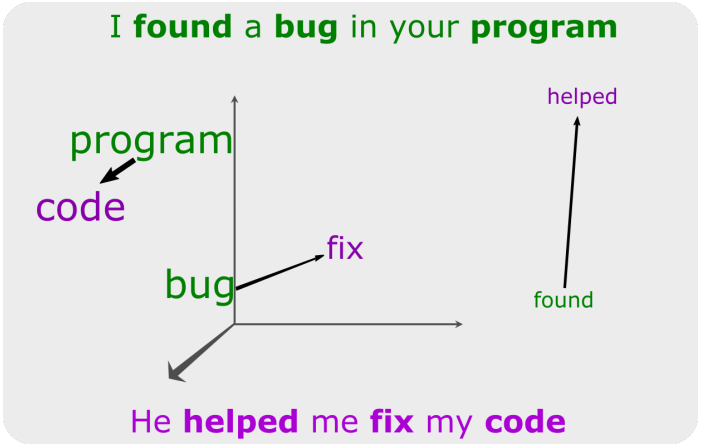


Figure 1.5: Word mover's distance: each word in both sentences are mapped to a point in a Euclidean space, except for stop-words. Optimal transport then allows to compute a distance between both sentences.

sis [Seguy and Cuturi, 2015, Cazelles et al., 2018]. In this work, we use this definition of the optimal transport distance between images in Chapter 4, in order to perform image denoising among others, and in Chapter 2 where we apply our dictionary learning method as a preprocessing step for face recognition.

1.1.3 Entropy Regularized Optimal Transport

We propose to use an entropy regularized version of optimal transport to solve optimization problems involving $\mathrm{OT}(\boldsymbol{x}, \cdot)$. The advantage of using this regularized optimal transport is twofold. First, similarly to Cuturi and Peyré [2016], Rolet et al. [2016], we take advantage of its smooth convex conjugate to derive dual problems that can be solved efficiently. Additionally, it lets us use further accelerations due to the special form of the cost matrix C in the case of optimal transport between images.

1.1.3.1 Definition

The entropy regularized optimal transport was proposed by Cuturi [2013] as a fast approximation of optimal transport. For $\boldsymbol{x}, \boldsymbol{y} \in \mathbb{R}_+^n$, $\gamma > 0$, we define the entropy regularized optimal transport between $\boldsymbol{x}$ and $\boldsymbol{y}$ as:

$$\mathrm{OT}_\gamma(\boldsymbol{x}, \boldsymbol{y}) = \begin{cases} +\infty & \text{if } U(\boldsymbol{x}, \boldsymbol{y}) = \emptyset \\ \min\limits_{T \in U(\boldsymbol{x}, \boldsymbol{y})} \langle T, C \rangle - \gamma E(T) & \text{otherwise,} \end{cases} \tag{1.11}$$

where $E(T) \coloneqq -\langle T, \log(T) \rangle$ is the entropy of T.

In recent years, entropy-regularized optimal transport has gained popularity as a proxy for optimal transport as a loss in optimization problems [Gramfort et al., 2015, Frogner et al., 2015, Seguy et al., 2018] due to both its simplicity and good properties with respect to convex optimization. Indeed, contrary to exact optimal transport, the entropy regularized version is differentiable everywhere, and the simple form of its convex conjugate allows to derive tractable duals for many optimization problems involving OT_γ.

Other regularizations of the optimal transport problem have been proposed [Blondel et al., 2018, Seguy et al., 2018], leading to different properties on the optimal transport plan T. All of the methods presented in this work would also be applicable to the Euclidean norm as a regularizer for example, but we only consider the entropy regularization in this work since we are not actually interested in transport plans, and the entropy regularization benefits from accelerations which make it the only tractable one in the case of transport between images.

1.1.3.2 Properties

Similarly to the non-regularized case, OT_γ is convex with respect to either its variable. One advantage over the exact optimal transport though is that it is differentiable, and so is its convex conjugate.

Dual Problem. The dual problem of regularized optimal transport with a convex regularizer is studied in details in Blondel et al. [2018]. We show here how to derive it for the entropy regularization. Suppose that $U(\boldsymbol{x}, \boldsymbol{y})$ is non-empty. Let us consider the Lagrangian of the regularized optimal transport problem

$$\begin{aligned}\mathcal{L}(T, \boldsymbol{u}, \boldsymbol{v}) &= \langle T, D\rangle - \left\langle T^\top \mathbf{1} - \boldsymbol{x}, \boldsymbol{u}\right\rangle - \langle T\mathbf{1} - \boldsymbol{y}, \boldsymbol{v}\rangle - \gamma E(T) \\ &= \langle T, D\rangle - \left\langle T^\top, \boldsymbol{u}\mathbf{1}^\top\right\rangle + \langle \boldsymbol{x}, \boldsymbol{u}\rangle + \left\langle T, \boldsymbol{v}\mathbf{1}^\top\right\rangle + \langle \boldsymbol{y}, \boldsymbol{v}\rangle + \gamma \langle T, \log T\rangle \\ &= \left\langle T, D - \mathbf{1}\boldsymbol{u}^\top - \boldsymbol{v}\mathbf{1}^\top + \gamma \log T\right\rangle + \langle \boldsymbol{x}, \boldsymbol{u}\rangle + \langle \boldsymbol{y}, \boldsymbol{v}\rangle .\end{aligned}$$

The Lagrange dual is then

$$\max_{\boldsymbol{u},\boldsymbol{v}} \min_{T\geq 0} \mathcal{L}(T, \boldsymbol{u}, \boldsymbol{v}) = \max_{\boldsymbol{u},\boldsymbol{v}} \min_{T\geq 0} \left\langle T, D - \mathbf{1}\boldsymbol{u}^\top - \boldsymbol{v}\mathbf{1}^\top + \gamma \log T\right\rangle + \langle \boldsymbol{x}, \boldsymbol{u}\rangle + \langle \boldsymbol{y}, \boldsymbol{v}\rangle .$$

Let us solve the inner minimization problem. We want to minimize a strictly convex function over an open convex set, thus the optimizer $T^\star$ is unique and the first order conditions tell us:

$$
\begin{aligned}
D - \mathbf{1}\boldsymbol{u}^\top - \boldsymbol{v}\mathbf{1}^\top + \gamma(\log T^\star + \mathbf{1}\mathbf{1}^\top) &= 0 \\
\log T^\star &= \frac{\mathbf{1}\boldsymbol{u}^\top + \boldsymbol{v}\mathbf{1}^\top - D - \gamma\mathbf{1}\mathbf{1}^\top}{\gamma} \\
T^\star &= e^{\frac{\mathbf{1}\boldsymbol{u}^\top + \boldsymbol{v}\mathbf{1}^\top - D - \gamma\mathbf{1}\mathbf{1}^\top}{\gamma}}
\end{aligned}
$$

Plugging the value of $T^\star$ in the dual we get

$$
\begin{aligned}
\max_{\boldsymbol{u},\boldsymbol{v}} \mathcal{L}(T^\star, \boldsymbol{u}, \boldsymbol{v}) &= \max_{\boldsymbol{u},\boldsymbol{v}} \left\langle e^{\frac{\mathbf{1}\boldsymbol{u}^\top + \boldsymbol{v}\mathbf{1}^\top - D - \gamma\mathbf{1}\mathbf{1}^\top}{\gamma}}, D - \mathbf{1}\boldsymbol{u}^\top - \boldsymbol{v}\mathbf{1}^\top + \mathbf{1}\boldsymbol{u}^\top + \boldsymbol{v}\mathbf{1}^\top - D - \gamma\mathbf{1}\mathbf{1}^\top \right\rangle \\
&\quad + \langle \boldsymbol{x}, \boldsymbol{u} \rangle + \langle \boldsymbol{y}, \boldsymbol{v} \rangle \\
&= \max_{\boldsymbol{u},\boldsymbol{v}} \langle \boldsymbol{x}, \boldsymbol{u} \rangle + \langle \boldsymbol{y}, \boldsymbol{v} \rangle - \gamma \left\langle e^{\frac{\mathbf{1}\boldsymbol{u}^\top + \boldsymbol{v}\mathbf{1}^\top - D - \gamma\mathbf{1}\mathbf{1}^\top}{\gamma}}, \mathbf{1}\mathbf{1}^\top \right\rangle
\end{aligned}
$$

Note how the dual of the regularization problem replaced a hard constraint on $\mathbf{1}\boldsymbol{u}^\top + \boldsymbol{v}\mathbf{1}^\top - D$ by a form of regularization on it. Similarly to the non-regularized case, if $\|\boldsymbol{x}\|_1 \neq \|\boldsymbol{y}\|_1$ the solution of the dual problem is $+\infty$. If a finite solution exists however, the maximizer is unique because the objective is strictly concave.

Since the regularized optimal transport problem is a feasible convex optimization problem with linear constraints, strong duality applies and

$$
\mathrm{OT}_\gamma(\boldsymbol{x}, \boldsymbol{y}) = \max_{\boldsymbol{u},\boldsymbol{v}} \langle \boldsymbol{x}, \boldsymbol{u} \rangle + \langle \boldsymbol{y}, \boldsymbol{v} \rangle - \gamma \left\langle e^{\frac{\mathbf{1}\boldsymbol{u}^\top + \boldsymbol{v}\mathbf{1}^\top - D - \gamma\mathbf{1}\mathbf{1}^\top}{\gamma}}, \mathbf{1}\mathbf{1}^\top \right\rangle . \tag{1.12}
$$

Convexity. Convexity is derived in the exact same way as in the non-regularized case. Re-using the same notations we have

$$
\begin{aligned}
\mathrm{OT}_\gamma(\boldsymbol{x}, \alpha\boldsymbol{y}^{(1)} + (1-\alpha)\boldsymbol{y}^{(2)}) &\leq \left\langle \boldsymbol{x}, \alpha\boldsymbol{u}^{(1)} + (1-\alpha)\boldsymbol{u}^{(2)} \right\rangle + \left\langle \boldsymbol{y}, \alpha\boldsymbol{v}^{(1)} + (1-\alpha)\boldsymbol{v}^{(2)} \right\rangle - \\
&\qquad \gamma \left\langle e^{\frac{\mathbf{1}(\alpha\boldsymbol{u}^{(1)} + (1-\alpha)\boldsymbol{u}^{(2)})^\top + (\alpha\boldsymbol{v}^{(1)} + (1-\alpha)\boldsymbol{v}^{(2)})\mathbf{1}^\top - D - \gamma\mathbf{1}\mathbf{1}^\top}{\gamma}}, \mathbf{1}\mathbf{1}^\top \right\rangle \\
&< \left\langle \boldsymbol{x}, \alpha\boldsymbol{u}^{(1)} + (1-\alpha)\boldsymbol{u}^{(2)} \right\rangle + \left\langle \boldsymbol{y}, \alpha\boldsymbol{v}^{(1)} + (1-\alpha)\boldsymbol{v}^{(2)} \right\rangle \\
&\qquad - \gamma\alpha \left\langle e^{\frac{\mathbf{1}\boldsymbol{u}^{(1)\top} + \boldsymbol{v}^{(1)}\mathbf{1}^\top - D - \gamma\mathbf{1}\mathbf{1}^\top}{\gamma}}, \mathbf{1}\mathbf{1}^\top \right\rangle \\
&\qquad - (1-\gamma)\alpha \left\langle e^{\frac{\mathbf{1}\boldsymbol{u}^{(2)\top} + \boldsymbol{v}^{(2)}\mathbf{1}^\top - D - \gamma\mathbf{1}\mathbf{1}^\top}{\gamma}}, \mathbf{1}\mathbf{1}^\top \right\rangle
\end{aligned}
$$

because the exponential is a strictly convex function. Thus $\mathrm{OT}_\gamma(\boldsymbol{x}, \alpha\boldsymbol{y}^{(1)} + (1-\alpha)\boldsymbol{y}^{(2)}) < \alpha\,\mathrm{OT}_\gamma(\boldsymbol{x}, \boldsymbol{y}^{(1)}) + (1-\alpha)\,\mathrm{OT}_\gamma(\boldsymbol{x}, \boldsymbol{y}^{(1)})$ and $\mathrm{OT}_\gamma(\boldsymbol{x}, \cdot)$ is strictly convex.

Convex conjugate. Similarly to the $\gamma = 0$ case, the entropy regularized optimal transport has a simple convex conjugate. Indeed Cuturi and Peyré [2016] showed that it can be expressed in closed form. Furthermore it is differentiable, its gradient is γ-Lipschitz and can also be expressed in closed form:

$$\begin{aligned}\mathrm{OT}^{\star}_{\gamma}(\boldsymbol{x}, \boldsymbol{z}) &= \gamma \left(E(\boldsymbol{x}) + \langle \boldsymbol{x}, \log K\boldsymbol{\alpha} \rangle \right), \\ \nabla_{\boldsymbol{y}} \mathrm{OT}^{\star}_{\gamma}(\boldsymbol{x}, \boldsymbol{z}) &= \boldsymbol{\alpha} \odot \left(K^{\top} \frac{\boldsymbol{x}}{K\boldsymbol{\alpha}} \right),\end{aligned}$$

where $K \coloneqq e^{-D/\gamma}$ and $\boldsymbol{\alpha} \coloneqq e^{\boldsymbol{z}/\gamma}$.

Although we use a definition of optimal transport which differs slightly from Cuturi and Peyré [2016], who consider normalized non-negative vectors, the formulas are the same and proved in the same way.

Proof. Let $\boldsymbol{x} \in \mathbb{R}^m_+$, $\boldsymbol{z} \in \mathbb{R}^n$,

$$\begin{aligned}\mathrm{OT}^{*}_{\gamma}(\boldsymbol{x}, \boldsymbol{z}) &= \max_{\boldsymbol{y}} \langle \boldsymbol{z}, \boldsymbol{y} \rangle - \mathrm{OT}_{\gamma}(\boldsymbol{x}, \boldsymbol{y}) \\ &= -\min_{\boldsymbol{y}} -\langle \boldsymbol{z}, \boldsymbol{y} \rangle + \min_{\substack{T \in \mathbb{R}^{n \times m}_+ \\ T\mathbf{1} = \boldsymbol{y} \\ T^{\top}\mathbf{1} = \boldsymbol{x}}} \langle T, D \rangle + \gamma \langle T, \log T \rangle \\ &= -\min_{\boldsymbol{y}} \min_{\substack{T \in \mathbb{R}^{n \times m}_+ \\ T\mathbf{1} = \boldsymbol{y} \\ T^{\top}\mathbf{1} = \boldsymbol{x}}} -\langle \boldsymbol{z}, \boldsymbol{y} \rangle + \langle T, D \rangle + \gamma \langle T, \log T \rangle \\ &= -\min_{\substack{\boldsymbol{y} \\ T \in \mathbb{R}^{n \times m}_+ \\ T\mathbf{1} = \boldsymbol{y} \\ T^{\top}\mathbf{1} = \boldsymbol{x}}} -\langle \boldsymbol{z}, \boldsymbol{y} \rangle + \langle T, D \rangle + \gamma \langle T, \log T \rangle \\ &= -\min_{\substack{T \in \mathbb{R}^{n \times m}_+ \\ T^{\top}\mathbf{1} = \boldsymbol{x}}} -\langle \boldsymbol{z}, T\mathbf{1} \rangle + \langle T, D \rangle + \gamma \langle T, \log T \rangle \\ &= -\min_{\substack{T \in \mathbb{R}^{n \times m}_+ \\ T^{\top}\mathbf{1} = \boldsymbol{x}}} \left\langle T, D - \boldsymbol{z}\mathbf{1}^{\top} + \gamma \log T \right\rangle \qquad (1.13)\end{aligned}$$

The last line is a (strictly) convex problem with non-empty linear constraint, so strong duality applies and

$$\begin{aligned}\mathrm{OT}^{*}_{\gamma}(\boldsymbol{x}, \boldsymbol{z}) &= -\max_{\boldsymbol{h}} \min_{T \in \mathbb{R}^{n \times m}_+} \left\langle D - \boldsymbol{z}\mathbf{1}^{\top} + \gamma \log T, T \right\rangle - \left\langle \boldsymbol{h}, T^{\top}\mathbf{1} - \boldsymbol{x} \right\rangle \\ &= -\max_{\boldsymbol{h}} \min_{T \in \mathbb{R}^{n \times m}_+} \left\langle D - \boldsymbol{z}\mathbf{1}^{\top} + \gamma \log T, T \right\rangle - \left\langle \boldsymbol{h}\mathbf{1}^{\top}, T^{\top} \right\rangle + \langle \boldsymbol{h}, \boldsymbol{x} \rangle \\ &= -\max_{\boldsymbol{h}} \min_{T \in \mathbb{R}^{n \times m}_+} \left\langle D - \boldsymbol{z}\mathbf{1}^{\top} + \gamma \log T - \mathbf{1}\boldsymbol{h}^{\top}, T \right\rangle + \langle \boldsymbol{h}, \boldsymbol{x} \rangle\end{aligned}$$

The first order condition for the inner minimization problem is:

$$\begin{aligned}
&D - \boldsymbol{z}\mathbf{1}^\top - \mathbf{1}\boldsymbol{h}^\top + \gamma \log T^\star + \gamma \mathbf{1}\mathbf{1}^\top = 0 \\
\Leftrightarrow &\gamma \log T^\star = \mathbf{1}\boldsymbol{h}^\top + \boldsymbol{z}\mathbf{1}^\top - D - \gamma \mathbf{1}\mathbf{1}^\top \\
\Leftrightarrow &T^\star = e^{\frac{\mathbf{1}\boldsymbol{h}^\top + \boldsymbol{z}\mathbf{1}^\top - D - \gamma \mathbf{1}\mathbf{1}^\top}{\gamma}} \\
\Leftrightarrow &T^\star = e^{\frac{\boldsymbol{z}\mathbf{1}^\top}{\gamma}} \odot e^{\frac{-D}{\gamma}} \odot e^{\frac{\mathbf{1}\boldsymbol{h}^\top - \gamma \mathbf{1}\mathbf{1}^\top}{\gamma}} \\
\Leftrightarrow &T^\star = \operatorname{diag}(\boldsymbol{\alpha})\, K \operatorname{diag}\left(e^{\frac{\boldsymbol{h} - \gamma \mathbf{1}}{\gamma}}\right).
\end{aligned} \tag{1.14}$$

Because of strong duality, we know that $T^\star$ is also the optimizer of the primal problem, thus $T^{\star\top}\mathbf{1} = \boldsymbol{x}$, so

$$\begin{aligned}
&\operatorname{diag}\left(e^{\frac{\boldsymbol{h}^\star - \gamma \mathbf{1}}{\gamma}}\right) K^\top \operatorname{diag}(\boldsymbol{\alpha})\, \mathbf{1} = \boldsymbol{x} \\
&\operatorname{diag}\left(e^{\frac{\boldsymbol{h}^\star - \gamma \mathbf{1}}{\gamma}}\right) K^\top \boldsymbol{\alpha} = \boldsymbol{x} \\
&e^{\frac{\boldsymbol{h}^\star - \gamma \mathbf{1}}{\gamma}} \odot (K^\top \boldsymbol{\alpha}) = \boldsymbol{x} \\
&e^{\frac{\boldsymbol{h}^\star - \gamma \mathbf{1}}{\gamma}} = \frac{\boldsymbol{x}}{K^\top \boldsymbol{\alpha}}
\end{aligned}$$

We thus have $T^\star = \operatorname{diag}(\boldsymbol{\alpha})\, K \operatorname{diag}\left(\frac{\boldsymbol{x}}{K^\top \boldsymbol{\alpha}}\right)$. Plugging this value and Equation 1.14 into Equation 1.13, we get

$$\begin{aligned}
\mathrm{OT}_\gamma^*(\boldsymbol{x}, \boldsymbol{z}) &= -\left\langle \operatorname{diag}(\boldsymbol{\alpha})\, K \operatorname{diag}\left(\frac{\boldsymbol{x}}{K^\top \boldsymbol{\alpha}}\right), D - \boldsymbol{z}\mathbf{1}^\top + \gamma(\log \alpha \mathbf{1}^\top + \log K + \log(\frac{\boldsymbol{x}}{K^\top \boldsymbol{\alpha}})^\top \mathbf{1} \right\rangle \\
&= -\left\langle \operatorname{diag}(\boldsymbol{\alpha})\, K \operatorname{diag}\left(\frac{\boldsymbol{x}}{K^\top \boldsymbol{\alpha}}\right), \gamma \log(\frac{\boldsymbol{x}}{K^\top \boldsymbol{\alpha}})^\top \mathbf{1} \right\rangle \\
&= -\left\langle \boldsymbol{\alpha}^\top K \operatorname{diag}\left(\frac{\boldsymbol{x}}{K^\top \boldsymbol{\alpha}}\right), \gamma \log(\frac{\boldsymbol{x}}{K^\top \boldsymbol{\alpha}})^\top \right\rangle \\
&= -\left\langle \boldsymbol{\alpha}^\top K \operatorname{diag}\left(\frac{1}{K^\top \boldsymbol{\alpha}}\right) \operatorname{diag}(\boldsymbol{x}), \gamma \log(\frac{\boldsymbol{x}}{K^\top \boldsymbol{\alpha}})^\top \right\rangle \\
&= -\left\langle \boldsymbol{x}, \gamma \log(\frac{\boldsymbol{x}}{K^\top \boldsymbol{\alpha}}) \right\rangle \\
&= \gamma \left\langle \boldsymbol{x}, \log K^\top \boldsymbol{\alpha} - \log \boldsymbol{x} \right\rangle \\
&= \gamma \left\langle \boldsymbol{x}, \log K^\top \boldsymbol{\alpha} \right\rangle + E(\boldsymbol{x})
\end{aligned}$$

The gradient is obtained by differentiation of this expression. □

Rolet et al. [2016] make use of the simple form of this convex conjugate and its gradient to derive fast algorithms for the optimal transport dictionary learning and NMF problems. Section 4.2.1 showcases the computational gain of using dual methods over primal ones on a simple regression problem.

The bottleneck in computing these formulas is the multiplication with matrix K. Supposing we are working with square images of size m, then $\boldsymbol{x}$ is of size $n = m^2$ and the complexity of multiplying with matrix K is $\mathcal{O}(n^2) = \mathcal{O}(m^4)$. Moreover storing matrix K also has a space complexity of $\mathcal{O}(n^2)$.

Accelerations. In the case where we are computing optimal transport between images, or anytime the cost matrix D is a matrix of pairwise Euclidean distances on a grid , multiplications with matrix K and $K^\top$ are simply Gaussian convolutions of standard deviation $\sigma^2 = \gamma$ [Solomon et al., 2015, ¶5.]. This allows us to compute $\mathrm{OT}_\gamma^\star$ in $\mathcal{O}(n \log n)$ instead of $\mathcal{O}(n^2)$, and to not store the matrix K in memory. Figure 1.6 shows experimental times for multiplicating K with a vector, implementing this operation as either a convolution or an actual matrix multiplication. For images of size lower or equal to 16, the matrix multiplication may be faster. This can be useful for example in dictionary learning or any other task in which images are divided into small patches. In this thesis however, we will only consider full images, accordingly we use the acceleration of Solomon et al. [2015] in all the timing results we report, if applicable.

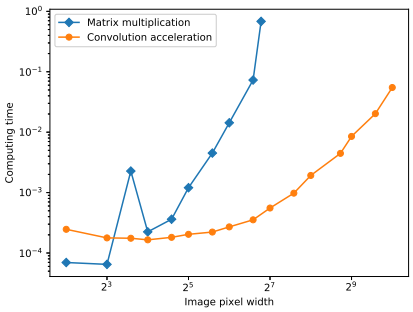


Figure 1.6: Computational time for multiplication with K for a square image with respect to its width (log-scale)

Effect of using the entropy-regularized optimal transport. Since the entropy regularized optimal transport is not a distance, given an input $\boldsymbol{x}$, the point $\boldsymbol{y}$ which minimizes $\mathrm{OT}_\gamma(\boldsymbol{x}, \boldsymbol{y})$ is not $\boldsymbol{x}$:

Lemma 1.1.2 (Closest point)**.** *Let $\boldsymbol{x} \in \mathbb{R}_+^n$,*

$$\operatorname*{argmin}_{\boldsymbol{y} \in \mathbb{R}_+^n} \mathrm{OT}_\gamma(\boldsymbol{x}, \boldsymbol{y}) = \frac{K^\top \boldsymbol{x}}{K\mathbf{1}}$$

Proof. Let $g := \boldsymbol{x} \mapsto 0$, Fenchel duality tells us that

$$\begin{aligned} \min_{\boldsymbol{y}} \mathrm{OT}_\gamma(\boldsymbol{x}, \boldsymbol{y}) + g(\boldsymbol{y}) &= \max_{\boldsymbol{h}=\mathbf{0}} \mathrm{OT}_\gamma^*(\boldsymbol{x}, \boldsymbol{h}) \\ &= \mathrm{OT}_\gamma^*(\boldsymbol{x}, \mathbf{0}) \end{aligned}$$

The primal-dual relationship gives us

$$\boldsymbol{y}^\star = \nabla \operatorname{OT}^*_\gamma(\boldsymbol{x}, \mathbf{0}) = \mathbf{1} \odot \left(K^\top \frac{\boldsymbol{x}}{K\mathbf{1}} \right).$$

□

In the case where $\boldsymbol{x}$ is an image and C is the matrix of squared Euclidean distances between pixel locations, this means that the closest point to any point with respect to the regularized optimal transport is simply a Gaussian blur of standard deviation $\sigma^2 = \gamma$, rescaled to have the same total intensity as the original image. Based on this observation, we set the regularization parameter γ of the entropy-regularized optimal transport to 0.1 in all of our results of Section 4.3, so that the closest point would be a blur of standard deviation 0.1 pixel, which is invisible to the naked eye.

1.2 Dictionary Learning and NMF

1.2.1 Problem Formulation

In this thesis, we are interested in the problem of dictionary learning with an optimal transport cost. Informally, dictionary learning aims at learning a representation of a space as a few elements, which can be linearly combined to approximate data. Consider a collection $X = (\boldsymbol{x}^{(1)}, \dots, \boldsymbol{x}^{(m)})$ of m vectors of dimension n. The goal of dictionary learning is to find k elements $D = (\boldsymbol{d}^{(1)}, \dots, \boldsymbol{d}^{(k)})$ of the same dimension n such that each $\boldsymbol{x}^{(i)}$ can be reconstructed using D, *i.e.* there exists a matrix of mixture weights $\lambda = (\boldsymbol{\lambda}^{(1)}, \dots, \boldsymbol{\lambda}^{(m)})$ such that $X \simeq D\lambda$.

When all elements of X are non-negative, and if it is desirable that all elements of D and λ are non-negative too, this problem becomes that of Non-negative Matrix Factorization (NMF) [Paatero and Tapper, 1994]. Dictionary learning is usually solved by casting it into the following optimization problem:

$$\min_{D \in \mathbb{R}^{n \times k}, \lambda \in \mathbb{R}^{k \times m}} \sum_{i=1}^{m} \ell(\boldsymbol{x}^{(i)}, D\boldsymbol{\lambda}^{(i)}) + R_1(D) + R_2(\lambda), \tag{1.15}$$

where ℓ is a divergence, whose role is to ensure that the data is closely reconstructed by the dictionary D, and R is a regularizer, which enforces desired properties on D and λ, *e.g.* non-negativity or sparsity.

Fast algorithm have been proposed for solving dictionary learning and NMF with the Euclidean distance or the Kullback-Leibler divergence as the data fitting term ℓ[Lee and Seung, 2001]. Sparsity of the coefficients can be obtained by using a ℓ_1 regularization on λ. [Mairal et al., 2009] showed how to tackle this sparse dictionary learning problem in an online setting, that is where λ is available by chunks, and exhibited good performance on many image processing tasks, such as image in-painting and denoising. While outside of the scope of this monograph, optimal transport sparse dictionary learning is an interesting direction for future works. We currently only have fast algorithms for learning sparse coefficients with optimal transport if the dictionary is invertible (detailed in Chapter 4), however for sparse dictionary learning, we would need to be able to learn dictionaries of any size, and especially overfull ones for which our algorithms don't apply.

NMF and dictionary learning methods in general have been used to tackle many other machine learning and signal processing tasks, including (but not limited to) topic modeling [Lee and Seung, 1999, Rolet et al., 2016], matrix completion [Zhang et al., 2006], sound denoising [Schmidt et al., 2007], image denoising [Mairal et al., 2009] and blind source separation[BSS; Sawada et al., 2013, Rolet et al., 2018].

1.2.2 Algorithms

In the simplest form, where ℓ is the squared Euclidean norm and $R_1 = R_2 = 0$, Problem 1.15 can be solved exactly. Indeed, finding the matrix $\hat{X}$ of rank k which is closest to X can be solved with singular value decomposition (SVD) and removing all but the k largest singular values [Eckart-Young theorem; Eckart and Young, 1936]. Since $\hat{X}$ is of rank k, it is easy to then factor it exactly into D and λ.

In the general case however, we do not have a formula for a solution, and usually we do not even have algorithms which output a global optimum, because the problem is not convex. If ℓ is a convex function of its second argument, and R_1 and R_2 are convex functions, then the objective of Problem 1.15 is convex with respect to D and λ separately. This means that when either D or λ is fixed, the objective is convex with respect to the other variable. However the objective is not convex when both variable vary at the same time. Many methods have been proposed to solve the problem in that case and we will now proceed to explore some of them, in particular alternate optimization which is the method we use for optimal transport dictionary learning.

1.2.2.1 Alternate Optimization

Many methods for solving dictionary learning and NMF problems, such as alternating least square (ALS), rely on alternate minimization methods [Paatero and Tapper, 1994, Berry et al., 2007, Kim and Park, 2008], in which we start with an initial value for D (or λ), and then iterate between solving the problem with D fixed and with λ fixed, which can be summarized as follows:

$$\begin{cases} \lambda^{j+1} \in \underset{\lambda}{\operatorname{argmin}} \sum_{i=1}^{m} \ell(\boldsymbol{x}^{(i)}, D^{j}\boldsymbol{\lambda}^{(i)}) + R_2(\lambda) & \text{(1.16a)} \\ D^{i+1} \in \underset{D}{\operatorname{argmin}} \sum_{i=1}^{m} \ell(\boldsymbol{x}^{(i)}, D\boldsymbol{\lambda}^{\boldsymbol{j+1}(i)}) + R_1(D) & \text{(1.16b)} \end{cases}$$

This simple scheme is based on the fact that usually, the problem is convex in D and λ separately:

Theorem 1.2.1 (Convexity)**.** *Suppose that ℓ is a convex function of its second argument, and R_1 and R_2 are convex functions. Then the problems in Equation 1.16a and Equation 1.16b are convex.*

Proof. The proof is the same for both problems, so we only give it for Problem 1.16a. We already know that R_2 is convex. Let $1 \leq i \leq m$, the function $f_i : \ell(\boldsymbol{x}_i, D\boldsymbol{\lambda}_i)$ is convex as the composition of a convex function with a linear map. The objective of the problem in Equation 1.16a is thus convex as a sum. □

Under mild assumptions, we can further guarantee that the sequence of objectives produced by our alternate optimization method converges. First, let us define this sequence $u = (u_i)_{i \in \mathbb{N}}$ as:

$$\text{for } j \in \mathbb{N}, \quad \begin{cases} u_{2j} = \sum_{i=1}^{m} \ell(\boldsymbol{x}^{(i)}, D^j \boldsymbol{\lambda^j}^{(i)}) + R_1(D^j) + R_2(\lambda^j) \\ u_{2j+1} = \sum_{i=1}^{m} \ell(\boldsymbol{x}^{(i)}, D^j \boldsymbol{\lambda^{j+1}}^{(i)}) + R_1(D^j) + R_2(\lambda^{j+1}) \end{cases}$$

The fact that u is decreasing can easily be proved by recursion. If moreover ℓ, R_1 and R_2 are lower-bounded, u converges. In this work, we consider problems where all these conditions are satisfied. In particular as shown in previous sections, OT_γ with $\gamma \geq 0$ is convex of its second argument.

1.2.2.2 Other Methods

Lee and Seung [2001] also use a form of alternate method, with multiplicative updates which reduce the objective but don't solve sub-problems with D or λ fixed. Unfortunately we do not have simple update formulas in the case of optimal transport and we cannot use the same approach. Mairal et al. [2009] solves a sparse dictionary learning problem with an alternate method, but in an online setting, *i.e.* where X is not given in full, but its columns can be sampled and updates are computed on a batch. In a similar way, Cao et al. [2016] solve dictionary learning with a stochastic alternate method, where updates are computed on a subsets of the columns of X selected randomly. These two methods could be used with the optimal transport distance as the reconstruction error term, but are outside of the scope of this work.

1.2.3 Probabilistic Latent Semantic Indexing and NMF

Probabilistic latent semantic indexing (PLSI) is a popular method for topic modeling [Hofmann, 1999]. Given a dataset of text, the goal is to generate *topics*, *i.e.* group words in terms of their meaning in order to analyze the dataset qualitatively. For example, in a dataset of sports news article, we would except to have topics relating to each sports, and maybe topics on tournaments or rules. In PLSA, topics are generated in the form of probabilities on words, where words of high probability should share a similar meaning or concept.

With PLSA, we consider texts as *bags-of-words*: we discard the order of the words in a text, which can be represented as a count-vector of the words it contains. The probabilistic model of PLSA is summarized in Figure 1.7. An event is the generation of a word w in a text t, and we consider these two variables independent knowing the

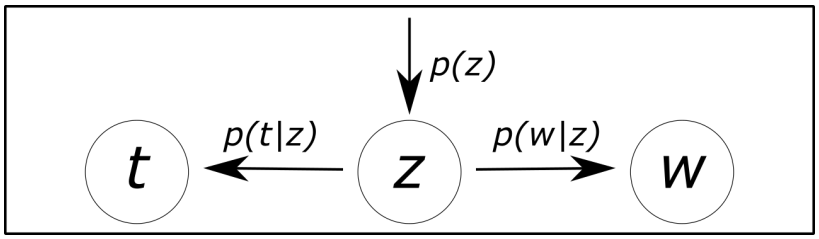


Figure 1.7: Generative model of PLSA.

hidden variable z (the topic). Generation events are independent, and if we assume k topics the probability of assigning a word w_i to a text t_j is

$$p(w_i, t_j) = \sum_{q=1}^{k} p(z_q)p(w_i|z_q)p(t_j|z_q).$$

Given a dataset X, where for $1 \leq i \leq m$ and $1 \leq j \leq n$, x_{ij} is the number of time the word w_i appears in the text t_j. The likelihood of X is

$$\mathcal{L}(x) = \frac{nm!}{\prod_{\substack{i=1\dots m \\ j=1\dots n}} x_{ij}!} \prod_{\substack{i=1\dots m \\ j=1\dots n}} p(w_i, t_j)^{x_{ij}}$$

Hofmann [1999] maximize the likelihood with an expectation-maximization (E-M) approach. We will now show that maximizing $\mathcal{L}(x)$ is equivalent to solving the non-regularized NMF problem with a Kullback-Leibler divergence as the loss. First, note that in order to maximize $\mathcal{L}(x)$, we can drop the constant normalization factor, and also maximize its logarithm. We are thus interested in solving

$$\max \sum_{\substack{i=1\dots m \\ j=1\dots n}} x_{ij} \log p(w_i, t_j) = \max \sum_{\substack{i=1\dots m \\ j=1\dots n}} x_{ij} \log \left(\sum_{q=1}^{k} p(z_q)p(w_i|z_q)p(t_j|z_q) \right)$$

Let $D \in \mathbb{R}_+^{m \times k}$ and $\lambda \in \mathbb{R}_+^{k \times n}$ such that for any i, j and q, $d_ij = p(w_i|z_q)$ and $\lambda_{qj} = p(t_j|z_q)$. Then $(D\lambda)_{ij} = \sum_{q=1}^{k} p(z_q)p(w_i|z_q)p(t_j|z_q)$, and maximizing $\mathcal{L}(x)$ is thus equivalent to

$$\max_{D\lambda} \sum_{\substack{i=1\dots m \\ j=1\dots n}} x_{ij} \log (D\lambda)_{ij}.$$

Up to a constant this is the same as

$$\min_{D\lambda} \sum_{\substack{i=1\dots m \\ j=1\dots n}} x_{ij} \log \frac{1}{(D\lambda)_{ij}} + x_{ij} \log x_{ij} = \min_{D\lambda} \sum_{j=1\dots n} \mathrm{KL}\left(\boldsymbol{x}_j \| D\boldsymbol{\lambda}_j\right).$$

This shows that PLSA is equivalent to solving the non-regularized Kullback-Leibler NMF. We can rephrase this as follows: minimizing the Kullback-Leibler divergence between the empirical distribution X and the modeled distribution $D\lambda$ is equivalent to maximizing the likelihood of the PLSA model.

1.2.4 Other Applications

In addition to topic modeling, NMF and dictionary learning can be used in a variety of tasks and fields, including blind source separation, image processing or recommendation systems.

Blind Source Separation. Given a sound signal which is known to be generated by a mixture of sources (different voices for example), source separation aims at isolating the signal produced by each source. A source separation method is *blind* if it only uses sound data as an input, as opposed to using other information such as the position of microphones relative to the source and to each other for example or the layout of the room in which the sound is produced. NMF has been used to tackle BSS in Schmidt and Olsson [2006] and Sun and Mysore [2013] among others. The idea is to learn a dictionary for each source, concatenate the dictionaries and compute the weights for the signal, and separate the dictionary again to isolate the sources. The learning phase can be done on the signal to separate (unsupervised), or on isolated data (supervised). For more details on NMF for BSS, see Chapter 3, which is dedicated to defining a blind source separation method with optimal transport NMF and gives a more detailed explanation of BSS in general and how to perform it with NMF. We show that using optimal transport leads to good result on BSS both for sound denoising and for separating mixtures of voices.

Image Processing. Dictionary learning and NMF have been applied to many image processing tasks. Sandler and Lindenbaum [2009] for instance use NMF as a pre-processing step for face recognition. They use the coefficient matrix λ as the input for a classification algorithm, which can be thought of as dimensionality reduction. We reproduce their experiment in Chapter 2. Mairal et al. [2009] use sparse dictionary learning, Euclidean dictionary learning with an ℓ_1-norm regularization on λ, for image denoising and in-painting. They cut images into overlapping small sized patches, in the order of 9×9 pixels, and learn an overfull dictionary. They then build a denoised or in-painted image from the reconstructed patches. With this method, the dictionary can be learned on a dataset or directly on the patches of the image to be processed.

Recommendation Systems. Given some appreciation rating of items for a user, recommendation systems aim as suggesting other items to which the user might give a high rating. Such systems might be found in online shops or streaming websites for example: given a customer's purchase or viewing history, such websites are quite interested in suggesting the next item to purchase/view, thus retaining the customer. Early systems based on k-nearest neighbors approaches were soon replaced with ma-

trix factorization approaches [Koren et al., 2009]. In this case, the matrix X we want to factorize is matrix of ratings, for which lines represent items and columns represent users: x_{ij} is the rating given by user j to item i. Some (usually most) of the values x_{ij} are unknown, making the problem a matrix completion problem. Such values are simply removed from the objective, and many matrix factorization algorithms can still be applied, including for example ALS and stochastic gradient descent.

1.3 Contributions

Building on previous works on regularized optimal transport [Cuturi, 2013, Cuturi and Peyré, 2016], we proposed in Rolet et al. [2016] to use entropy-regularized optimal transport as the loss in dictionary learning and NMF problems. We derived dual optimization problems for the sub-problems of dictionary learning with either D or λ fixed. Using these dual problems, we were able to solve optimal transport NMF orders of magnitude faster than their primal counter-parts, and than the previous method of Sandler and Lindenbaum [2009]. The use of regularized optimal transport allows us work with any data for which optimal transport can be computed, which was not the case with the approximation of Shirdhonkar and Jacobs [2008]. This allowed us to define "cross-domain" dictionary learning, for which we showed an application in cross language topic modeling: we summarize a database of text in a chosen language which can be different than the language of the summarized database.

In Rolet et al. [2018], we expanded our method and showed how to apply it to sound processing. We showed that the using optimal transport NMF, we can learn *universal* voice models which can better generalize to unheard voices than when learned with other NMF methods. Based on this observation, we defined an optimal transport Blind Source Separation (BSS) method and showed it outperformed other NMF-based methods when applied to voice denoising. We also showcased how "cross-domain" dictionary learning allows our models to be robust to a change in the data-acquisition process between train and test time.

Finally in Rolet and Seguy [2021], we showed that when D is invertible, our duals can be used to compute sparse optimal coefficients λ. In particular, we can compute sparse decomposition on Fourier or wavelet bases, which allowed us to define *optimal transport wavelet shrinkage*. We showed that using optimal transport to get a sparse representations on these bases lead to reduced artifacts when processing image. We further demonstrated the advantage of optimal transport shrinkage compared to other shrinkage-based methods on an image denoising task, in particular in the presence of non-gaussian noise.

Chapter 2

Optimal Transport Dictionary Learning and Non-negative Matrix Factorization

Cross-Domain Topic Modeling with Optimal Transport NMF

2.1 Chapter Introduction

Consider a collection $X = (\boldsymbol{x}_1, \dots, \boldsymbol{x}_m)$ of m vectors of dimension n. Learning a dictionary for X can be stated informally as the goal of finding k dictionary elements $D = (\boldsymbol{d}_1, \dots, \boldsymbol{d}_k)$ of the same dimension n such that each $\boldsymbol{x}_i$ can be reconstructed using such a dictionary, namely such that there exists a matrix of mixture weights $\lambda = (\boldsymbol{\lambda}_1, \dots, \boldsymbol{\lambda}_m)$ such that $X \simeq D\lambda$.

When all elements of X are non-negative, and if it is desirable that all elements of D and λ are non-negative too, this problem becomes that of Non-negative Matrix Factorization (NMF) [Paatero and Tapper, 1994]. Lee and Seung [2001] proposed two algorithms for NMF, with the aim of solving problems of the form:

$$\min_{D \in \mathbb{R}_+^{n \times k}, \lambda \in \mathbb{R}_+^{k \times m}} \sum_{i=1}^{m} \ell(\boldsymbol{x}_i, D\boldsymbol{\lambda}_i) + R_1(\lambda) + R_2(D),$$

where ℓ is either the Kullback-Leibler divergence or the squared Euclidean distance and R a regularizer. Dictionary learning and NMF have been used for various machine learning and signal processing tasks, such as topic modeling [Hofmann, 1999, Lee and Seung, 1999], matrix completion [Zhang et al., 2006] and sound denoising [Schmidt et al., 2007].

Our goal in this chapter is to generalize these approaches using a regularized optimal transport distance as the data fitting term ℓ. Such distances can leverage additional knowledge on the space of features using a metric between features called the ground metric. Since the seminal work of Rubner et al. [1998], several hundred papers have successfully used EMD in applications. Some recent works have for instance illustrated its relevance for text classification [Kusner et al., 2015], image segmentation [Rabin and Papadakis, 2015] and shape interpolation [Solomon et al., 2015].

We motivate the idea of using an optimal transport fitting error with a toy example described in Figure 2.1. In this example we try to learn dictionaries for histogram representations of i.i.d. samples from mixtures of Gaussians. We consider $n = 100$ distributions $\rho_1, \dots, \rho_n$, each of which is a mixture of three univariate Gaussians of unit variance, with centers picked independently using $\mathcal{N}(-6, 2)$, $\mathcal{N}(0, 2)$ and $\mathcal{N}(6, 2)$ respectively. The relative weights of these Gaussians are picked uniformly on $[0, 1]$ and subsequently normalized to sum to 1 for each distribution. We represent each data sample as a normalized histogram $\boldsymbol{x}_i$ of $n = 100$ bins regularly spaced on the segment $[-12, 12]$. Here the features are points on the quantization grid, and the ground metric is simply the Euclidean distance between these points. Optimal transport NMF recovers components which are centered around -6, 0 and 6 and resemble Gaussian pdfs. Because it is blind to the metric structure of $\mathbb{R}$, KL NMF fail to recover such intuitive components.

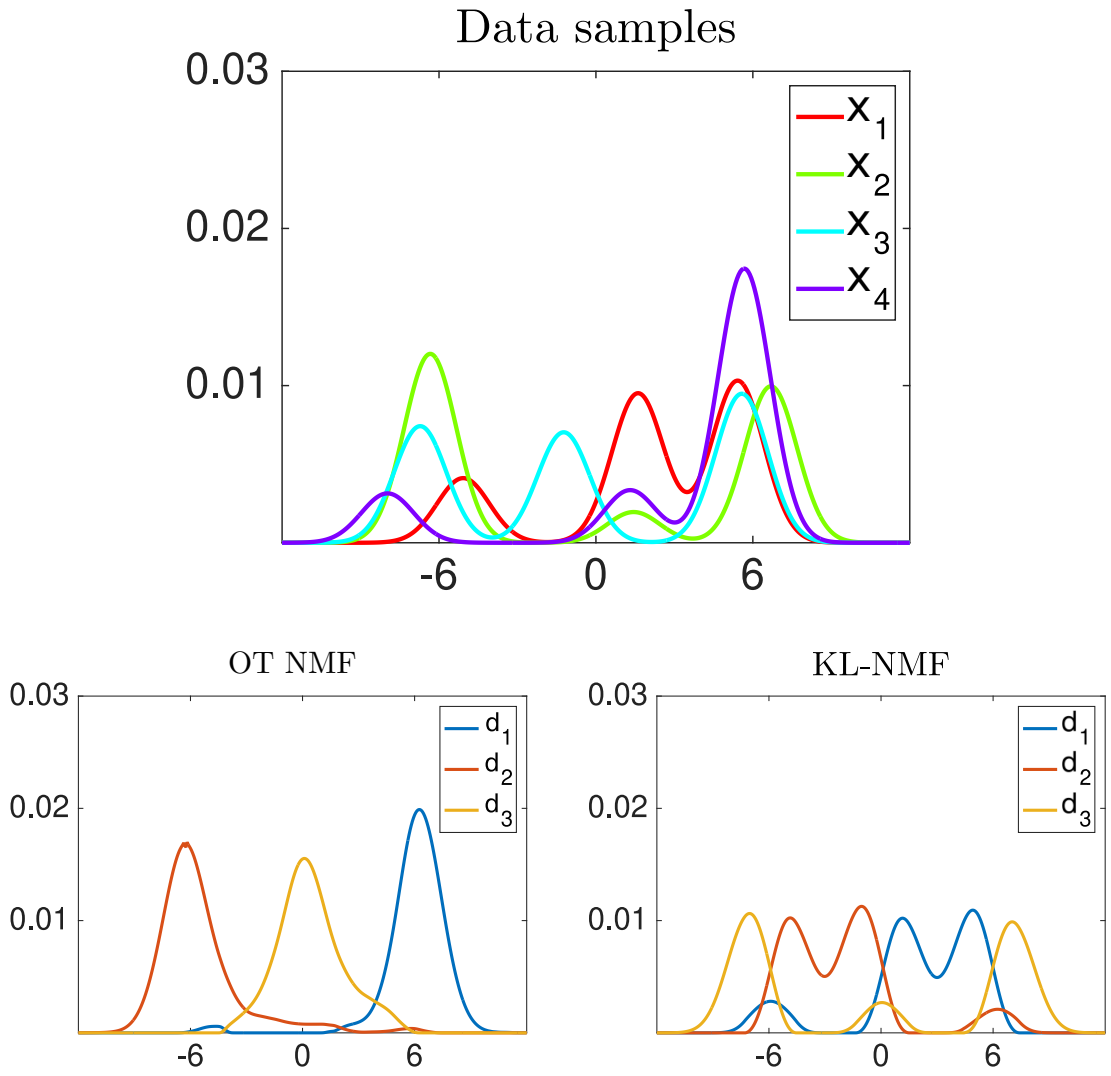


Figure 2.1: Dictionaries learned on mixtures of three randomly shifted Gaussians. Separable distances or divergences do not quantify this noise well because it is not additive in the space of histograms. Top: examples of data histograms. Bottom: dictionary learned with optimal transport (left) and Kullback-Leibler (right) NMF.

Related Work Sandler and Lindenbaum [2009] were the first to consider NMF problems using an optimal transport loss. They noticed that minimizing optimal transport fitting errors requires solving an extremely costly linear program at each iteration of their block-coordinate iteration. Because of this, they settle instead for an approximation of the optimal transport distance proposed by Shirdhonkar and Jacobs [2008]. However this approximation can only be used when the features are in $\mathbb{R}^d$, and its complexity is exponential in d, making it impractical when $d > 3$. Moreover the experimental approximation ratio for $d = 2$ in Shirdhonkar and Jacobs [2008] is rather loose (1.5) even with the best hyper-parameters. Zen et al. [2014] also proposed a semi-supervised method to learn D, λ and a ground metric parameter. Their approach is to alternatively learn the ground metric as proposed previously in [Cuturi and Avis,

2014] and perform NMF by solving two very high dimensional linear programs. They apply their algorithm to histograms of small dimension ($n \leq 16$).

Our Contribution The algorithms we propose to solve dictionary learning and NMF problems with an optimal transport loss scale to problems with far more observations and dimensions than previously considered in the literature [Sandler and Lindenbaum, 2009, Zen et al., 2014]. This is enabled by an entropic regularization of optimal transport [Cuturi, 2013] which results in faster and more stable computations. We give in Section 2.2 a detailed presentation of our algorithms for optimal transport (non-negative) matrix factorization of histogram matrices. In contrast to previously considered approaches, our approach can be applied with any ground metric. As with most dictionary learning problems, our objective is not convex but biconvex in the dictionary D and weights λ and we use a block-coordinate descent approach. We show that each of these sub-problems can be reduced to an optimization problem involving the Legendre-Fenchel conjugate of the objective, building upon recent work in Cuturi and Peyré [2016] that shows that the Legendre-Fenchel conjugate of the entropy regularized optimal transport distance and its gradient can be obtained in closed form. We show in Section 2.3 that these fast algorithms are order of magnitudes faster than those proposed in Sandler and Lindenbaum [2009], whose experiments we replicate. Finally, we show that the features used to describe dictionary elements can be different from those present in the original histograms. We showcase this property to carry out cross-language topic modeling: we learn topics in French using databases of English texts. A Matlab implementation of our methods and scripts to reproduce the experiment in the introduction are available at `http://arolet.github.io/wasserstein-dictionary-learning/`.

2.2 Optimal Transport Dictionary Learning

Let $X \in (\Sigma_n)^m$ be a matrix of m vectors in the n-dimensional simplex. Let k be a number of dictionary elements, fixed in advance. We consider the problem

$$\min_{\lambda \in \mathbb{R}^{k \times m}, D \in \mathbb{R}^{s \times k}} \sum_{i=1}^{m} \mathrm{OT}_\gamma(\boldsymbol{x}_i, D\boldsymbol{\lambda}_i) + R_1(\lambda) + R_2(D) \tag{2.1}$$

Problem (2.1) is convex separately (but not jointly) in D and λ as long as R_1 and R_2 are convex. We propose in what follows to use a block-coordinate descent on D and λ.

Sandler and Lindenbaum [2009] show that when $R = 0$, $\gamma = 0$ and either D or λ is fixed, Equation (2.1) is a linear program of dimensions $m \times n \times s$ with $m \times (n \times s + n + s)$ constraints, each involving 1, n or $t \times k$ variables. Representing these constraints is challenging for common sized datasets, and solving such problems is usually intractable. They proposed to replace the optimal transport distance by an approximation [Shirdhonkar and Jacobs, 2008], for which the gradients are easier to compute. However this approximation can only be used when M is a distance matrix in a Euclidean space of small dimension. We propose instead to use the entropy regularized optimal transport, with $\gamma > 0$. This allows us to consider any cost matrix M, rather than only pairwise distance matrices, and makes the optimization problems smooth and better behaved, in the sense that when D or λ are full rank, the optimizers of each block update is unique. Moreover, we can adjust γ to get closer to zero in order to refine our approximation of standard optimal transport. We propose next in §2.2.3 an entropic regularization on the dictionary D and weights λ to enforce positivity of these coefficients.

Following the popular alternating least square method [ALS, Takane et al., 1977], we consider a two-step procedure: since the objective is convex with respect to either D or λ, we start by initiating D and then alternate between minimizing with respect to λ and minimizing with respect to D. We start by giving simple dual problems for the minimizing steps with D and λ, and follow by outlining our algorithms for dictionary learning and NMF. We then discuss convergence and implementation considerations.

2.2.1 Weights Update

We consider here the case where the dictionary D is fixed, and our goal is to compute mixture weights λ

$$\operatorname*{argmin}_{\lambda \in \mathbb{R}^{k \times m}} \sum_{i=1}^{m} \mathrm{OT}_\gamma(\boldsymbol{x}_i, D\boldsymbol{\lambda}_i) + R_1(\lambda). \tag{2.2}$$

2.2.1.1 Existence and Unicity

We showed in Rolet and Seguy [2021] that Problem 2.2 has simple sufficient conditions for existence and unicity. In short, existence is almost always guaranteed except in degenerate cases, and unicity depends on the properties of D and R_1.

We start by giving simple existence and unicity conditions for the solutions of Problem 2.2. We restrict ourselves to the case where X and λ have only one column, *i.e.* are vectors, for simplicity. However the proofs are easily generalized to the matrix case. Let

$$f : \boldsymbol{\lambda} \mapsto \mathrm{OT}_\gamma(\boldsymbol{x}, D\boldsymbol{\lambda}).$$

We can get simple existence conditions for the solutions of Problem 4.1 based on the domains of f and R, which we call dom_f and dom_R respectively.

Theorem 2.2.1. *If D is full rank and $\mathrm{Im}(D) \cap \mathbb{R}_+^k \neq \{0\}$, then dom_f is compact and non-empty.*

Proof. Suppose that D is full-rank and $\mathrm{Im}(D) \cap \mathbb{R}_+^n \neq \{0\}$. Let $\boldsymbol{a} \in \mathrm{Im}(D) \cap \mathbb{R}_+^n$ such that $\boldsymbol{a} \neq 0$. Let $\boldsymbol{\lambda} \in \mathbb{R}^k$ such that $\boldsymbol{a} = D\boldsymbol{\lambda}$. Let $\boldsymbol{b} = \frac{\|\boldsymbol{x}\|_1}{\|\boldsymbol{a}\|_1}\boldsymbol{\lambda}$, we have $D\boldsymbol{b} = \frac{\|\boldsymbol{x}\|_1}{\|\boldsymbol{a}\|_1}\boldsymbol{a} \geq 0$ and $\|D\boldsymbol{b}\|_1 = \|\boldsymbol{x}\|_1$ so $\boldsymbol{b} \in \mathrm{dom}_f$ and dom_f is not empty.

Let us now prove that dom_f is compact. $\mathrm{dom}_f = \left\{\boldsymbol{\lambda} | D\boldsymbol{\lambda} \geq 0,\, \mathbf{1}^\top D\boldsymbol{\lambda} = \mathbf{1}^\top \boldsymbol{x}\right\}$ is a polyhedron defined as an intersection of an hyperplane and closed half-spaces. It is thus closed and as a subset of $\mathbb{R}^k$, it is compact *iif* it is unbounded, which for a polyhedron is equivalent to not containing any half line.

Let δ be a half-line, we will show that δ is not included in dom_f. There exist some vectors $\boldsymbol{a}, \boldsymbol{b} \in \mathbb{R}^k$ with $\boldsymbol{b} \neq 0$, such that $\delta = \{\boldsymbol{a} + \beta\boldsymbol{b} | \beta \geq 0\}$.

Since D is full rank, $D\boldsymbol{b} \neq 0$. Let $0 < i \leq k$ such that $(D\boldsymbol{b})_i \neq 0$. There are three possible cases:

- $\boldsymbol{a}$ is not in dom_f, then δ is not included in dom_f.
- $\boldsymbol{a} \in \mathrm{dom}_f$ and $(D\boldsymbol{b})_i > 0$:

  Since $\boldsymbol{a} \in \mathrm{dom}_f$, we now that $(D\boldsymbol{a})_i \leq \|\boldsymbol{x}\|_1$. Let $\beta = \frac{\|\boldsymbol{x}\|_1 - (D\boldsymbol{a})_i + 1}{((D\boldsymbol{b})_i)}$, $(D(\boldsymbol{a} + \beta\boldsymbol{b}))_i = \|\boldsymbol{x}\|_1 + 1 > \|\boldsymbol{x}\|_1$, so $\boldsymbol{a} + \beta\boldsymbol{b}$ is not in dom_f and δ is not included in dom_f.
- $\boldsymbol{a} \in \mathrm{dom}_f$ and $(D\boldsymbol{b})_i < 0$:

  Since $\boldsymbol{a} \in \mathrm{dom}_f$, we now that $(D\boldsymbol{a})_i \geq 0$. Let $\beta = \frac{-(D\boldsymbol{a})_i - 1}{((D\boldsymbol{b})_i)}$, $(D(\boldsymbol{a} + \beta\boldsymbol{b}))_i = -1 < 0$, so $\boldsymbol{a} + \beta\boldsymbol{b}$ is not in dom_f and δ is not included in dom_f.

This shows that δ is not included in dom_f, As a closed polyhedron which contains no half-line, dom_f is bounded and thus compact. □

Theorem 2.2.2 (Existence)**.** *Let R be a convex function. If $\mathrm{dom}_R \cap \mathrm{dom}_f$ is not empty and compact, then Problem 4.1 has a solution.*

Proof. The conditions directly imply that Problem 4.1 is a convex problem over a non-empty compact set, so it has a solution. □

In all the applications we considered, dom_R is either $\mathbb{R}^{k\times m}$ or $\mathbb{R}_+^{k\times m}$. A simple condition for the existence of a solution is then $D \geq 0$. In the non-constrained case, a less restrictive condition would be that there exist $\lambda \in \mathbb{R}_+^{k\times m}$ such that $D\lambda \geq 0$ with $D\lambda \neq 0$. In Chapter 4, we solve the problem with an invertible dictionary D, optimal coefficients are thus guaranteed to exist.

Unicity of a solution is follows from strict convexity of either f or R.

Theorem 2.2.3 (Unicity)**.** *Let R_1 be a convex function, $\gamma > 0$. If D is full rank, Problem 4.1 has at most one solution.*

Proof. Suppose that D is full-rank, then it defines an injective linear map. Since $\gamma > 0$, $\mathrm{OT}_\gamma(\boldsymbol{x}, \cdot)$ is strictly convex. f is then strictly convex, and since R_1 is convex the objective of Problem 4.1 is strictly convex. As a result it can have at most one solution.

□

The previous result is only valid for the entropy-regularized optimal transport. We can get unicity of a solution with exact transport by restricting R to strictly convex functions:

Theorem 2.2.4 (Unicity II)**.** *Let R_1 be strictly convex function, $\gamma \geq 0$. Problem 4.1 has at most one solution.*

Proof. $\mathrm{OT}_\gamma(\boldsymbol{x}, \cdot)$ is convex, so f is convex too and the objective of Problem 4.1 is strictly convex. As a result it can have at most one solution.

□

In our applications of Chapter 4, D is invertible and $\gamma > 0$. Theorem 2.2.3 then implies that Problem 2.2 has at most a solution. According to Theorem 2.2.1, dom_f is compact and non-empty.

In the remainder of this work, we assume existence and unicity for the wording of our results. However these results hold whether existence and unicity conditions are actually satisfied or not.

2.2.1.2 Duality

Problem 2.2 has some hidden constraints, indeed if $\boldsymbol{x}_i \neq D\boldsymbol{\lambda}_i$ for any i, the objective is infinite. As such, it could be solved using a proximal method such as FISTA [Beck and Teboulle, 2009], but computing the gradient is equivalent to evaluating each $H_{x_i}(D\lambda_i)$ for $i = 1, \cdots, m$, that is solving m intermediate optimal transport problems. We show in Section 4.2.4 that solving this problem in the dual can be hundreds if not thousands of time faster that a direct primal approach.

We now proceed to prove the main result of this work, which is that Problem 2.2 has a simple dual that can be solved efficiently.

Theorem 2.2.5 (Dual for the Coefficients Step)**.** *The solution $\lambda^\star$ of Problem 2.2 satisfies the primal-dual relationship*

$$D\lambda^\star = \left(\nabla \operatorname{OT}^*_\gamma(\boldsymbol{x}_i, \boldsymbol{h}^\star{}_i)\right)_{i=1}^m \tag{2.3}$$

where $H^\star$ is the solution of the dual problem

$$\min_{H \in \mathbb{R}^{n \times m}} \sum_{i=1}^m \operatorname{OT}^*_\gamma(\boldsymbol{x}_i, \boldsymbol{h}_i) + R_1^*(-D^\top H). \tag{2.4}$$

The advantage of working with Problem 2.4 is that the objective doesn't include OT_γ but replaces it with $\operatorname{OT}^*_\gamma$, whose value and gradient can be computed efficiently using the formulae of Section 1.1.3.

Proof. We rewrite Problem (4.1) by introducing the variable $Q = D\lambda$:

$$\min_{\substack{\lambda \in \mathbb{R}^{k \times m} \\ Q \in \mathbb{R}_+^{n \times m} \\ D\lambda = P}} \sum_{i=1}^m \operatorname{OT}_\gamma(\boldsymbol{x}_i, \boldsymbol{q}_i) + R_1(\lambda).$$

It is a convex problem with linear constraints so strong duality holds, the dual problem being:

$$\begin{aligned} &\operatorname*{argmax}_{H \in \mathbb{R}^{n \times m}} \min_{\substack{\lambda \in \mathbb{R}^{k \times m} \\ Q \in \mathbb{R}_+^{n \times m}}} \sum_{i=1}^m \operatorname{OT}_\gamma(\boldsymbol{x}_i, \boldsymbol{q}_i) - \langle H, Q - D\lambda \rangle + R(\boldsymbol{\lambda}) \\ =&\operatorname*{argmax}_{H \in \mathbb{R}^{n \times m}} \min_{\substack{\lambda \in \mathbb{R}^{k \times m} \\ Q \in \mathbb{R}_+^{n \times m}}} \sum_{i=1}^m \operatorname{OT}_\gamma(\boldsymbol{x}_i, \boldsymbol{q}_i) - \langle \boldsymbol{h}_i, \boldsymbol{q}_i \rangle + \langle H, D\lambda \rangle + R(\boldsymbol{\lambda}) \end{aligned}$$

By definition of OT_γ^*, we get

$$\operatorname*{argmax}_{H\in\mathbb{R}^{n\times m}} \min_{\lambda\in\mathbb{R}^{k\times m}} -\sum_{i=1}^{m} \mathrm{OT}_\gamma^*(\boldsymbol{x}_i, \boldsymbol{h}_i) + \langle H, D\lambda\rangle + R(\lambda) \tag{2.5}$$

$$= \operatorname*{argmin}_{H\in\mathbb{R}^{n\times m}} \max_{\lambda\in\mathbb{R}^{k\times m}} \sum_{i=1}^{m} \mathrm{OT}_\gamma^*(\boldsymbol{x}_i, \boldsymbol{h}_i) + \left\langle -D^\top H, \lambda\right\rangle - R(\lambda) \tag{2.6}$$

$$= \operatorname*{argmin}_{H\in\mathbb{R}^{n\times m}} \sum_{i=1}^{m} \mathrm{OT}_\gamma^*(\boldsymbol{x}_i, \boldsymbol{h}_i) + \max_{\lambda\in\mathbb{R}^{k\times m}} \left\langle -D^\top H, \lambda\right\rangle - R(\lambda) \tag{2.7}$$

Noting that the right side is the convex conjugate of R, we get the dual problem:

$$\operatorname*{argmin}_{H\in\mathbb{R}^{n\times m}} \sum_{i=1}^{m} \mathrm{OT}_\gamma^*(\boldsymbol{x}_i, \boldsymbol{h}_i) + R^*(-D^\top H).$$

Problem (4.3) is simply the Fenchel dual of the original problem, the primal-dual relationship in Equation (2.3) can be recovered from the first order conditions of Problem 2.5 with respect to variable $\boldsymbol{h}$.

□

If R^* is smooth and its gradient can be computed efficiently, we can solve Problem (4.3) with an accelerated gradient method [Nesterov, 1983]. Once the optimizer $H^\star$ of the dual is found, we compute $\lambda^\star$ by solving the primal-dual relationship, *i.e.* Equation (2.3). This equation is simply a system of linear equations, and duality guaranties that it does have a solution.

2.2.2 Dictionary Update

Assuming weights λ are fixed, our goal is now to learn the dictionary matrix D by solving

$$\min_{D\in\mathbb{R}^{s\times k}} \sum_{i=1}^{m} \mathrm{OT}_\gamma(\boldsymbol{x}_i, D\boldsymbol{\lambda}_i) + R_2(D). \tag{2.8}$$

2.2.2.1 Existence and Unicity

As for the coefficient case, existence of a dictionary for which the objective is finite is not guaranteed. Indeed, the optimal transport term is finite only if $D\lambda > 0$ and $\mathbf{1}^\top X = \mathbf{1}^\top D\lambda$. In the limit case where $k = 1$, this reduces to $\lambda \propto \mathbf{1}^\top X$. However, if

$\lambda \geq 0$ and $\mathbf{1}^\top \lambda = \mathbf{1}^\top X$, then any non-negative D with $D^\top \mathbf{1} = \mathbf{1}$ is feasible, and the problem has an infinite number of solutions. In general however, existence conditions tend to be more complicated that for the coefficients case, because the problem is not separable anymore with respect to the columns of X or λ.

For this reason, we always initialize our alternate optimization with $D \geq 0$ and optimize over λ first. This way, at each iteration, the current dictionary is feasible and there is at least one solution.

We can get unicity however, with similar conditions as for the weight update:

Theorem 2.2.6 (Unicity)**.** *Let R_2 be a convex function, $\gamma > 0$. If λ is full rank, Problem 2.8 has at most one solution.*

Theorem 2.2.7 (Unicity II)**.** *Let R_2 be strictly convex function, $\gamma \geq 0$. Problem 2.8 has at most one solution.*

We leave out the proofs, since they are the same as for Theorems 2.2.3 and 2.2.4.

2.2.2.2 Duality

Computing an optimal dictionary for a given weight matrix can be done through a dual problem, in a similar fashion as for the weight learning step:

Theorem 2.2.8 (Dual for the Dictionary Step)**.** *Let $D^\star$ be a solution of Problem 2.8. $D^\star$ satisfies the primal-dual relationship*

$$D^\star \lambda = \left(\nabla \operatorname{OT}_\gamma^* \boldsymbol{x}_i(\boldsymbol{g}^\star{}_i) \right)_{i=1}^m , \tag{2.9}$$

where $H^\star$ is the solution of the dual problem

$$\min_{H \in \mathbb{R}^{s \times m}} \sum_{i=1}^m \operatorname{OT}_\gamma^*(\boldsymbol{x}_i, \boldsymbol{h}_i) + R_2^*(-H \boldsymbol{\lambda}_i^\top). \tag{2.10}$$

Proof. Let us introduce the variable $Q = D\lambda$. Problem (2.8) becomes

$$\min_{D \in \mathbb{R}^{s \times k}, Q \in \mathbb{R}^{s \times m}} \sum_{i=1}^m \operatorname{OT}_\gamma(\boldsymbol{x}_i, \boldsymbol{q}_i) + R_2(D), \text{ s.t. } Q = D\lambda.$$

This is a convex optimization problem with linear constraints, thus strong Lagrange duality holds, with the following dual:

$$\max_{H \in \mathbb{R}^{s \times m}} \min_{D \in \mathbb{R}^{s \times k}, Q \in \mathbb{R}^{s \times m}} \sum_{i=1}^m \operatorname{OT}_\gamma(\boldsymbol{x}_i, \boldsymbol{q}_i) + R_2(D) + \langle H, D\lambda - Q \rangle .$$

We can rewrite this problem to simplify it using OT^*_γ:

$$
\begin{aligned}
&\operatorname*{argmax}_{H\in\mathbb{R}^{s\times m}} \min_{D\in\mathbb{R}^{s\times k}} \langle H, D\lambda\rangle + R_2(D) + \min_{Q\in\mathbb{R}^{s\times m}} \sum_{i=1}^{m} \mathrm{OT}_\gamma(\boldsymbol{x}_i, \boldsymbol{q}_i) - \langle H, Q\rangle \\
=&\operatorname*{argmax}_{H\in\mathbb{R}^{s\times m}} \min_{D\in\mathbb{R}^{s\times k}} \langle H, D\lambda\rangle + R_2(D) + \sum_{i=1}^{m} \min_{\boldsymbol{q}\in\mathbb{R}^{s\times m}} \mathrm{OT}_\gamma(\boldsymbol{x}_i, \boldsymbol{q}) - \langle \boldsymbol{h}_i, \boldsymbol{q}\rangle \\
=&\operatorname*{argmax}_{H\in\mathbb{R}^{s\times m}} \min_{D\in\mathbb{R}^{s\times k}} \langle H, D\lambda\rangle + R_2(D) - \sum_{i=1}^{m} \mathrm{OT}^*_\gamma(\boldsymbol{x}_i, \boldsymbol{h}_i) \qquad (2.11) \\
=&\operatorname*{argmax}_{H\in\mathbb{R}^{s\times m}} \min_{D\in\mathbb{R}^{s\times k}} \left\langle H\lambda^\top, D\right\rangle + R_2(D) - \sum_{i=1}^{m} \mathrm{OT}^*_\gamma(\boldsymbol{x}_i, \boldsymbol{h}_i) \\
=&\operatorname*{argmax}_{H\in\mathbb{R}^{s\times m}} \min_{D\in\mathbb{R}^{s\times k}} -\left\langle -H\lambda^\top, D\right\rangle + R_2(D) - \sum_{i=1}^{m} \mathrm{OT}^*_\gamma(\boldsymbol{x}_i, \boldsymbol{h}_i) \\
=&\operatorname*{argmax}_{H\in\mathbb{R}^{s\times m}} -R_2^*(-H\lambda^\top) - \sum_{i=1}^{m} \mathrm{OT}^*_\gamma(\boldsymbol{x}_i, \boldsymbol{h}_i) \qquad (2.12)
\end{aligned}
$$

Problem 2.12 is equivalent to Problem 2.10. The primal-dual relationship is derived from the first-order conditions of Problem 2.11.

□

Note that here, Problem 2.10 is not separable, this is normal since changing one dictionary element affects all reconstructions.

2.2.3 Algorithms

Both our dictionary learning algorithm and our NMF algorithm rely on alternately solving Problem 2.2 and Problem 2.8. In order to simplify the outline of our algorithm, for any regularizer function f we define:

$$
\begin{aligned}
\mathrm{DL}_{1,f}(D) =& \operatorname*{argmin}_{\lambda\in\mathbb{R}^{k\times m}} \sum_{i=1}^{m} \mathrm{OT}_\gamma(\boldsymbol{x}_i, D\boldsymbol{\lambda}_i) + f(\lambda) \\
\mathrm{DL}_{2,f}(\lambda) =& \operatorname*{argmin}_{D\in\mathbb{R}^{s\times k}} \sum_{i=1}^{m} \mathrm{OT}_\gamma(\boldsymbol{x}_i, D\boldsymbol{\lambda}_i) + f(D).
\end{aligned}
$$

We compute $\mathrm{DL}_{1,f}$ and $\mathrm{DL}_{2,f}$ using FISTA [Beck and Teboulle, 2009].

Dictionary Learning Solving either the weight update or the dictionary update without regularization introduces a constraint in the dual problem. Indeed, let f be the constant 0-valued function of a matrix. For any matrix M, $f^*(M) = \begin{cases} 0 \text{ if } M = 0 \\ \infty \text{ otherwise} \end{cases}$

As a result, when $R_1 = f = 0$, Equation 2.4 becomes

$$\min_{\substack{H \in \mathbb{R}^{s \times m} \\ D^\top \boldsymbol{h}_i = 0}} \sum_{i=1}^{m} \mathrm{OT}_\gamma^*(\boldsymbol{x}_i, \boldsymbol{h}_i). \tag{2.13}$$

Conversely, when $R_2 = f = 0$, Equation 2.10 becomes

$$\min_{\substack{H \in \mathbb{R}^{s \times m} \\ H\lambda^\top = 0}} \sum_{i=1}^{m} \mathrm{OT}_\gamma^*(\boldsymbol{x}_i, \boldsymbol{h}_i). \tag{2.14}$$

Both weight and dictionary updates can be solved with FISTA, where the proximal step is simply a projection on the constraint.

Since the value of the objective of un-regularized dictionary learning depends on D and λ only through their product, multiplying a column of D by a value, and dividing the corresponding row of λ by the same value yields the same objective. As a result, iterates of D and λ could very easily grow infinitely big or small.

In order to those iterate from blowing up, we will require each column of D to have a unit ℓ_1-norm. This is easily done after each dictionary update: let $\boldsymbol{a} = |D|^\top \mathbf{1}$, before going to the next iteration, we perform $D \leftarrow D \operatorname{diag}(\boldsymbol{a})$ and $\lambda \leftarrow \operatorname{diag}\left(\frac{1}{\boldsymbol{a}}\right)\lambda$. As noted earlier, this doesn't affect the objective value. We summarize the overall procedure in Algorithm 1.

Algorithm 1: Optimal Transport Dictionary Learning

Data: Input dataset $X \in \mathbb{R}^{n \times m}$, dictionary size k
Result: Factorization matrices D and λ

begin
  Initialization: set $D \in \mathbb{R}_+^{s \times m}$ randomly, with normalized columns
  while *not converged* **do**
    $\lambda \leftarrow \mathrm{DL}_{1,0}(D)$
    $D \leftarrow \mathrm{DL}_{2,0}(\lambda)$
    $D \leftarrow D \operatorname{diag}\left(|\boldsymbol{d}|^\top \mathbf{1}\right)$
  end
  $\lambda \leftarrow \operatorname{diag}\left(\frac{1}{|\boldsymbol{d}|^\top \mathbf{1}}\right)\lambda$
end

Optimal Transport NMF In order to enforce non-negativity constraints on the variables, we consider the problem

$$\min_{\substack{\lambda\in\mathbb{R}_+^{k\times m}\\ D\in\mathbb{R}_+^{s\times k}\\ D^\top\mathbf{1}=\mathbf{1}}} \sum_{i=1}^{m} H_{x_i}(D\lambda_i) - \rho_1 E(\lambda) - \rho_2 E(D), \tag{2.15}$$

We can solve each sub-problem with either D or λ fixed using Theorem 2.2.5 and Theorem 2.2.8, with

$$R_1(\lambda) = \rho_1 E(\lambda) \tag{2.16}$$

$$R_2(D) = \begin{cases} \rho_2 E(D) & \text{if } D^\top\mathbf{1} = \mathbf{1} \\ \infty & \text{otherwise.} \end{cases} \tag{2.17}$$

The conjugate and gradient of R_1 are:

$$R_1^*(H) = \rho_1 \left\langle \mathbf{1}\mathbf{1}^\top, e^{\frac{H-\rho_1}{\rho_1}} \right\rangle$$
$$\nabla R_1^*(H) = e^{\frac{H-\rho_1}{\rho_1}}.$$

Proof. Let $H \in \mathbb{R}^{k\times m}$,

$$R_1^*(H) = \max_{\lambda\in\mathbb{R}_+^{k\times m}} \langle H, \lambda\rangle - \rho_1 E(\lambda).$$

The right-hand optimization problem is a concave problem, for which the first order conditions read

$$H - \rho_1(\log(\lambda) - \mathbf{1}) = 0$$
$$\Leftrightarrow \lambda = e^{\frac{H-\rho_1}{\rho_1}}.$$

Injecting λ in the objective function yields

$$\begin{aligned} R_1^*(H) &= \left\langle H, e^{\frac{H-\rho_1}{\rho_1}} \right\rangle - \rho_1 \left\langle e^{\frac{H-\rho_1}{\rho_1}}, \log\left(e^{\frac{H-\rho_1}{\rho_1}}\right) \right\rangle \\ &= \left\langle H, e^{\frac{H-\rho_1}{\rho_1}} \right\rangle - \left\langle e^{\frac{H-\rho_1}{\rho_1}}, H - \rho_1 \right\rangle \\ &= \left\langle e^{\frac{H-\rho_1}{\rho_1}}, \mathbf{1}\mathbf{1}^\top \right\rangle. \end{aligned}$$

The gradient is trivially obtained from the expression of R_1^*.

□

The conjugate and gradient of R_2 are:

$$R_2^*(H) = \rho_2 \left\langle \mathbf{1}, \log(e^{\frac{H^\top}{\rho_2}}\mathbf{1}) \right\rangle$$
$$\nabla R_2^*(H) = \frac{e^{\frac{H}{\rho_2}}}{\mathbf{1}\mathbf{1}^\top e^{\frac{H}{\rho_2}}}.$$

Proof. Let $H \in \mathbb{R}^{s\times k}$,

$$R_2^*(H) = \max_{\substack{D\in\mathbb{R}_+^{s\times k}\\ D^\top\mathbf{1}=\mathbf{1}}} \langle H, D\rangle - \rho_2 E(D) \tag{2.18}$$

Let $f : D \mapsto \langle H, D\rangle - \rho_2 E(D)$ be the objective function of the optimization problem in Equation 2.18, and $\mathrm{dom}_f = \left\{D \in \mathbb{R}_+^{s\times k} |\, D^\top\mathbf{1} = \mathbf{1}\right\}$ be the domain of the same optimization problem. Suppose we can find a matrix D in the interior of dom_f, for which $\nabla f(D) \perp \mathrm{dom}_f$, since f is concave, D would maximize f over dom_f. Let us look for such a D.

The condition $\nabla f(D) \perp \mathrm{dom}_f$ means that $\nabla f(D)$ is constant column-wise, *i.e.* $\nabla f(D) = \mathbf{1}\boldsymbol{a}^\top$ for some $\boldsymbol{a} \in \mathbb{R}^k$. We have

$$\begin{aligned}\nabla f(D) =& H - \rho_2 - \rho_2\log(D)\\ =& \mathbf{1}\boldsymbol{a}^\top.\end{aligned}$$

This means that

$$\begin{aligned}\log(D) =& \frac{H - \rho_2 - \mathbf{1}\boldsymbol{a}^\top}{\rho_2}\\ \Leftrightarrow D =& e^{\frac{H-\rho_2-\mathbf{1}\boldsymbol{a}^\top}{\rho_2}}.\end{aligned}$$

We know that $D^\top\mathbf{1} = \mathbf{1}$, so

$$
\begin{aligned}
& e^{\frac{H^\top - \rho_2 - \boldsymbol{a}\mathbf{1}^\top}{\rho_2}}\mathbf{1} = \mathbf{1} \\
\Leftrightarrow \quad & \left(e^{\frac{-\boldsymbol{a}\mathbf{1}^\top}{\rho_2}} \odot e^{\frac{H^\top - \rho_2}{\rho_2}}\right)\mathbf{1} = \mathbf{1} \\
\Leftrightarrow \quad & e^{\frac{-\boldsymbol{a}}{\rho_2}} \odot \left(e^{\frac{H^\top - \rho_2}{\rho_2}}\mathbf{1}\right) = \mathbf{1} \\
\Leftrightarrow \quad & \frac{-\boldsymbol{a}}{\rho_2} + \log\left(e^{\frac{H^\top - \rho_2}{\rho_2}}\mathbf{1}\right) = 0 \\
\Leftrightarrow \quad & \boldsymbol{a} = \rho_2 \log\left(e^{\frac{H^\top - \rho_2}{\rho_2}}\mathbf{1}\right).
\end{aligned}
$$

Substituting the value of $\boldsymbol{a}$ in D, we get $D = \frac{e^{\frac{H-\rho_2}{\rho_2}}}{\mathbf{1}\mathbf{1}^\top e^{\frac{H-\rho_2}{\rho_2}}} = \frac{e^{H/\rho_2}}{\mathbf{1}\mathbf{1}^\top e^{H/\rho_2}} = e^{H/\rho_2}\,\mathrm{diag}\left(\frac{1}{\mathbf{1}^\top e^{H/\rho_2}}\right)$. Note that $D^\top\mathbf{1} = \mathrm{diag}\left(\frac{1}{\mathbf{1}^\top e^{H/\rho_2}}\right) e^{H^\top/\rho_2}\mathbf{1} = \mathbf{1}$, thus $D \in \mathrm{dom}_f$. Moreover $D > 0$, so D is in the interior of dom_f. Since $\nabla f(D) \perp \mathrm{dom}_f$, D is the solution of the optimization problem in Equation 2.18.

We thus have

$$
\begin{aligned}
R_2^*(H) =& f(D) \\
=& \left\langle H, \frac{e^{H/\rho_2}}{\mathbf{1}\mathbf{1}^\top e^{H/\rho_2}} \right\rangle - \rho_2 \left\langle \frac{e^{H/\rho_2}}{\mathbf{1}\mathbf{1}^\top e^{H/\rho_2}}, \log\left(\frac{e^{H/\rho_2}}{\mathbf{1}\mathbf{1}^\top e^{H/\rho_2}}\right) \right\rangle \\
=& \left\langle H, \frac{e^{H/\rho_2}}{\mathbf{1}\mathbf{1}^\top e^{H/\rho_2}} \right\rangle - \rho_2 \left\langle \frac{e^{H/\rho_2}}{\mathbf{1}\mathbf{1}^\top e^{H/\rho_2}}, \log(e^{H/\rho_2}) - \log(\mathbf{1}\mathbf{1}^\top e^{H/\rho_2}) \right\rangle \\
=& \rho_2 \left\langle \frac{e^{H/\rho_2}}{\mathbf{1}\mathbf{1}^\top e^{H/\rho_2}}, \log(\mathbf{1}\mathbf{1}^\top e^{H/\rho_2}) \right\rangle.
\end{aligned}
$$

In order to simplify this expression further, let us write the dot-product as a sum:

$$
\begin{aligned}
R_2^*(H) =& \rho_2 \sum_{i=1}^{s}\sum_{j=1}^{k} \frac{e^{H_{ij}/\rho_2}}{\sum_{p=1}^{k} e^{H_{ip}/\rho_2}} \log(\sum_{p=1}^{k} e^{H_{ip}/\rho_2}) \\
=& \rho_2 \sum_{i=1}^{s} \log(\sum_{p=1}^{k} e^{H_{ip}/\rho_2}) \\
=& \rho_2 \left\langle \mathbf{1}, \log(e^{H^\top/\rho_2}\mathbf{1}) \right\rangle
\end{aligned}
$$

The gradient is obtained through differentiation of the developed expression. For any i, j,

$$
\nabla R_2^*(H)_{ij} = \frac{e^{H_{ij}/\rho_2}}{\log(\sum_{p=1}^{k} e^{H_{ip}/\rho_2})}
$$

□

We can solve each dual sub-problem with fista, which in this case amounts to an accelerated gradient [Nesterov, 1983] since the whole objective is differentiable and there is no constraint.

Algorithm 2: Optimal Transport NMF

Data: Input dataset $X \in \mathbb{R}^{n \times m}$, dictionary size k, ρ_1 and ρ_2
Result: Factorization matrices D and λ

begin
  Initialization: set $D \in \mathbb{R}_+^{s \times m}$ randomly, with normalized columns
  while *not converged* **do**
    $\lambda \leftarrow \mathrm{DL}_{1,R_1}(D)$
    $D \leftarrow \mathrm{DL}_{2,R_2}(\lambda)$
  end
end

2.2.4 Convergence

As pointed by Sandler and Lindenbaum [2009], the alternate optimization process generates a sequence of lower bounded non-increasing values for the objective of Problem (2.1), so the sequence of objectives converges. When, moreover, we use an entropic regularization ($\rho_1, \rho_2 > 0$, §2.2.3), successive updates for D and λ remain in a compact space (see Lemma 2.2.9), and thus satisfy the conditions of [Tropp, 2003, Theorem 3.1], taking into account that the hypothesis made in that theorem that the divergence is definite is not actually used in the proof. Thus every accumulation point of the sequences of iterates of D and λ is a generalized fixed point.

Moreover, if the iterates remain of full rank, then Theorem 3.2 in the same reference applies, and the sequences either converge or have a continuum of accumulation points. Although this full rank hypothesis is not guaranteed to hold, we observe that it holds in practice when the entropic regularization term does not dominate the objective.

Lemma 2.2.9. *All iterates of Algorithm 2 are inside of a compact.*

Proof. All dictionary iterates are in $\left\{D \in \mathbb{R}_+^{s \times k} |\, D^\top \mathbf{1} = \mathbf{1}\right\}$, which is a compact. Because of this restriction on D, iterates of λ must in turn be in $\left\{D \in \mathbb{R}_+^{k \times m} |\, \mathbf{1}^\top \lambda = \mathbf{1}^t opX\right\}$, which is also a compact. □

2.2.5 Implementation

Projection Step for Unconstrained Dictionary Learning We solve Equations (2.4) and (2.10) with projected gradient descent methods. The orthogonal projector of the optimization problem is $\text{proj}_{\text{Ker}(\text{D}^\top)} := G \mapsto G - DD^+G$ in Equation (2.4) and $\text{proj}_{\text{Ker}(\lambda)} := H \mapsto H - H\lambda^+\lambda$ in Equation (2.10). Pre-computing DD^+ (resp. $\lambda^+\lambda$) uses $O(s^2)$ (resp. $O(m^2)$) memory space, and then the projection is performed in complexity $O(s^2 \times m)$ (resp. $O(s \times m^2)$). When either s or m is large, storing such a matrix is too expensive and leads to slowdowns due to memory management. In such a case, we can pre-compute D^+ (resp. λ^+), which takes $O(s \times k)$ (resp. $O(m \times k)$) memory space, and compute $\text{proj}_{\text{Ker}(\text{D}^\top)}(H)$ as $H - D(D^+H)$ (resp. $\text{proj}_{\text{Ker}(\lambda)}(H)$ as $H - (H\lambda^+)\lambda$) in $O(s^2 \times m^2 \times k^2)$ operations.

Parallelization of the Dictionary Update Parallelization on multiple processes is easy for the weights updates because each weight vector λ_i can be computed independently. The dictionary updates however cannot be reduced to completely independent sub-problems. Indeed the regularizer in Equation (2.10) makes a dependence on the columns of D.

We show how to use parallel processes to speed-up the unconstrained dictionary updates. The most computationally expensive part it to solve the optimization problem of Equation (2.10). The objective and gradient of this problem can be computed independently for each column. Then we can gather the gradient on a single process and project it. Since the constraint is linear we can directly project the gradient before computing the step-size of the descent, so that if this computation involves computing the objective (like a backtracking line-search does for example) the projection does not need to be repeated.

We also propose a scheme to partially parallelize the positive dictionary updates. The objective and gradient of the optimization problem in Equation (2.10) when R_2 is the constrained entropy are found by computing $e^{-H\lambda^\top}$, which cannot be computed separately on columns of H. An efficient way to still compute $e^{-H\lambda^\top}$ in parallel is to split H column-wise into $(H^{(1)}, \dots, H^{(p)})$ where p is the number of processes available, and compute $e^{-H^{(i)}\lambda^\top}$ on process i. The managing process computes $e^{-H\lambda^\top}$ as $\prod_{i=1}^{p} e^{-H^{(i)}\lambda^\top}$ (here the product is point-wise) and gives the result to all the other processes so that they can finish computing the gradient. By doing so most of the work is done in parallel and each process only shares a matrix of size $s \times k$ twice per gradient/objective calculation. Since usually $k \ll m$ this allows to use all the available processes while keeping communication overhead low.

2.3 Experiments

2.3.1 Face Recognition

We reproduce here the face recognition experiment of Sandler and Lindenbaum [2009] on the ORL dataset [Samaria and Harter, 1994] with the same preprocessing, classification and evaluation method in order to compare computation time. Each image is down-sampled so that its longer side is 32. We represent images as column vectors that we normalize so that they sum to 1 and store them in matrix X. The cost matrix M is the Euclidean distance between pixels. For evaluation, the dataset is split evenly in two, trained on one set and tested on the other several times, and we take the take best classification performance obtained. Table 2.1 shows the classification accuracy obtained with unconstrained optimal transport Dictionary Learning (Section 2.2.3). The results are comparable to those of Sandler and Lindenbaum [2009].

k	10	20	30	40	50
$\gamma = 1/30$	93%	**95.5**%	**97**%	96.5%	96%
$\gamma = 1/50$	91%	95%	95%	**97**%	94.5%
Sandler09	**94.5**%	90.5%	95%	96.5%	**97**%

Table 2.1: Classification accuracy for the face recognition task on the ORL dataset. Taken from Rolet et al. [2016]

Learning the dictionary and coefficients with a Matlab implementation of our algorithm on an single core of a 2.4Ghz Intel Quad core i7 CPU with $k = 40$ takes on average $20s$ for $\gamma = 1/30$ and $90s$ for $\gamma = 1/50$, while Sandler and Lindenbaum [2009] report up to 20 minutes just for the D step with a comparable CPU. The whole NMF can take up to 10 minutes when we use the entropy positivity barrier with $\rho_1 = \rho_2 = 1/10$.

2.3.2 Topic Modeling

The goal of topic modeling is to extract a few representative histograms of words (*a.k.a.* topics) from large corpora of texts. To tackle this task, Probabilistic Latent Semantic Indexing (PLSI, Hofmann [1999]) learns a non-negative factorization of the form $X = D\Sigma\Lambda$, which models the document generation process: D is the matrix of word probabilities knowing the topic, Σ is the diagonal matrix of topic probabilities and Λ is the matrix of document probability knowing the topic. Ding et al. [2008] shows that PLSI optimizes the same objective as the algorithm in Lee and Seung [1999] for a Kullback-Leibler error term.

We use the same approach as Lee and Seung [1999] to learn topics from a database of texts with NMF. The input data is a *bag-of-words* representation of the documents.

Let $Y = \{y_1, \dots, y_n\}$ be the vocabulary of the database, a text document is represented as vector of word frequencies: X_{ij} is the frequency of the word y_i in the j^{th} text. We get topics D by learning a factorization $D\Lambda$ with NMF. The cost of the factorization is usually its Euclidean distance or Kullback-Leibler divergence to X. In order to use a optimal transport cost instead, we need a meaningful cost for transporting words from one to another.

Recent works [Pennington et al., 2014, Zou et al., 2013], building upon earlier references [Bengio et al., 2003], propose to compute Euclidean embeddings for words such that the Euclidean or cosine distances between the respective image of two words corresponds to some form of semantic discrepancy between these words. As recently shown by Kusner et al. [2015], these embeddings can be used to compare texts using the toolbox of optimal transport: Bag-of-words histograms can be compared with optimal transport distances using the Euclidean metric between the words as the ground metric M. We leverage these results to learn topics from a text database using optimal transport NMF.

2.3.2.1 Datasets

We learned topics on two datasets labeled. Labels are ignored for performing NMF, and are only used for evaluation. For each dataset, let m be the number of documents, n the vocabulary size and c the number of labels. (i) **BBCsport** [Greene and Cunningham, 2006] is a dataset of news articles about sports, labeled according to which sport the article is about, in which we removed stop-words ($n = 12,669$, $m = 737$, $c = 5$). We split the dataset as a 80/20 training / testing set for classification. (ii) **Reuters** is a dataset of news articles labeled according to their area of interest. We used the version described in Cardoso-Cachopo [2007], with the same train-test split for classification, and removed stop-words and words that appeared only once across the corpus ($n = 13,038$, $m = 7,674$, $c = 8$).

2.3.2.2 Monolingual Topic Modeling

We used a pre-trained Glove word embedding [Pennington et al., 2014] to map words to a Euclidean space of dimension 300. Let $\vec{y_1}, \dots, \vec{y_s}$ be the embeddings of the words in the dataset's vocabulary, and $\vec{z_1}, \dots, \vec{z_s}$ the embeddings of the words in the target vocabulary, that is the words that are allowed to appear in the topics. We define the cost matrix of the optimal transport distance as the cosine distance in the embedding: $m_{ij} = 1 - \frac{\langle \vec{y_i}, \vec{z_j} \rangle}{\|\vec{y_i}\| \|\vec{z_j}\|}$. We then find D and λ with optimal transport NMF (OT-NMF, Section 2.2.3).

Figure 2.2 shows a word cloud representation (`wordle.net`) for 4 relevant topics for the dataset BBCsport. Depending on the parameters, the full optimal transport NMF computation takes from 20 minutes to an hour for BBCsport and around 10

Figure 2.2: Word clouds representing 4 *of the* 15 *topics learned on BBCsport in English. Top-left topic: competitions. Top-right: time. Bottom-left: soccer actions. Bottom-right: drugs.*

Taken from Rolet et al. [2016]

hours for Reuters using a Matlab implementation running on a single GPU of an Nvidia Tesla K80 card.

Target Words Selection Since we can choose as target words any word that is defined for the embedding, we need a way to select which to use. We chose to use a list of $3,000$ frequent words in English[1]. Other approaches can be considered such as using the dataset's vocabulary, tokenized or not, or taking the most frequent words for each class in the dataset.

2.3.2.3 Cross-language Topic Modeling

We proposed in Rolet et al. [2016] to leverage optimal transport to perform what we call *cross-domain* learning tasks, which we illustrate here with *bilingual* topic modeling. Our new task will be to get topics in a language that is not the language of the dataset. This means that for each text $\boldsymbol{x}_i$, we compute a reconstruction $D\boldsymbol{\lambda}_i$ which is not in the same language. Because of this, we cannot compute a Euclidean

[1]Available at `https://simple.wiktionary.org/wiki/Wiktionary:BNC_spoken_freq`

distance or KL divergence between $\boldsymbol{x}_i$ and $D\boldsymbol{\lambda}_i$ anymore. However, we are able to do it with optimal transport thanks to bilingual word embeddings.

Our approach is to use the bilingual word embeddings proposed by Lauly et al. [2014] that map words from two different languages to the same Euclidean space. Using these, we are able to compute optimal transport between sentences in two different languages. By setting the vocabulary of the topics as a subset of the words in the target language, we can then learn topics in that language. Figure 2.3 illustrates what we would expect with $k = 1$, which is the optimal transport iso-barycenter problem. We use a pre-trained embeddings of dimension 40 from Lauly et al. [2014] in order to learn topics in French. Note that this method could also learn topics in one language from a bilingual dataset, or in both languages.

As in Section 2.3.2.2, we use the cosine distance in the embedding as the ground metric. Table 2.4 shows word cloud representations for 4 relevant topics for the dataset Reuters. Computation times are similar to those with a target vocabulary in English.

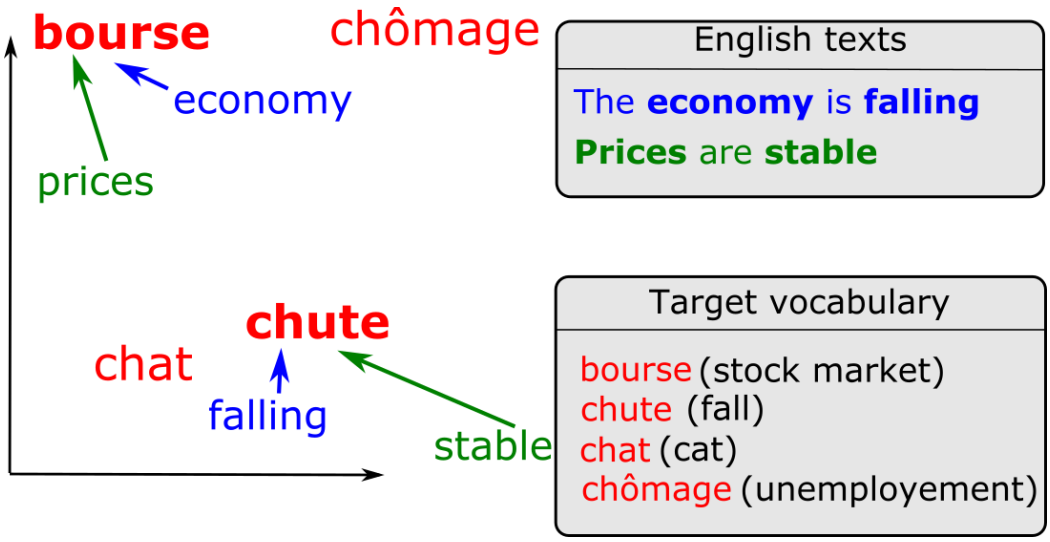


Figure 2.3: The optimal transport iso-barycenter of two English sentences with a target vocabulary in French. Arrows represent the optimal transport plan from a text to the barycenter. The barycenter is supported on the bold red words which are pointed by arrows. The barycenter is not equidistant to the extreme points because the set of possible features is discrete.

Taken from Rolet et al. [2016]

Target words selection We chose as the target dictionary a list of of 6,000 frequent words in French[2].

[2]Available at `http://wortschatz.uni-leipzig.de/Papers/top10000fr.txt`

Figure 2.4: Word clouds representing 4 of the 24 topics learned on Reuters in French. Top-left topic: international trade. Top-right: oil and other resources. Bottom-left: banking. Bottom-right: management and funding.

Taken from Rolet et al. [2016]

Method	KL-NMF	E-NMF	OT-NMF	OT-NMF$_f$
Reuters	6.9%	8.2%	**6.0**%	9.8%
BBCsport	9.4%	12.8%	**5.4**%	20.8%

Table 2.2: Text classification error. OT-NMF$_f$ is the classifier using OT-NMF with a French target vocabulary.

Taken from Rolet et al. [2016]

2.3.2.4 Classification Performance

We compared the classification error obtained on the two datasets with our method to those obtained by using the mixture weights produced by Euclidean NMF (E-NMF) and Kullback-Leibler NMF (KL-NMF). We use a k-NN classifier with a Hellinger distance between the mixture weights. k is selected by 10-fold cross-validation on the training set, using the same partitions for all methods. We set the number of topics to $3c$. Parameters γ, ρ_1 and ρ_2 were set to be as small as we could (small values can make the gradients infinite because of machine precision) without a particular selection procedure. See supplementary materials of Rolet et al. [2016] for a representation of all the topics of every method.

Optimal transport NMF with a target vocabulary in English performs better on this auxiliary task than Euclidean or KL NMF. Although this does not prove that the topics are of better quality, it shows that optimal transport NMF can drastically reduce the vocabulary size without losing discriminative power. As we can see in Figures 2.2, 2.4, the topics themselves are semantically coherent and related to the datasets' content.

English target vocabulary

	a	c	f	r	t
a	21	0	0	0	0
c	0	25	0	0	0
f	0	0	50	**4**	**1**
r	0	0	**3**	26	0
t	0	0	0	0	19

French target vocabulary

	a	c	f	r	t
a	18	0	**1**	0	**2**
c	0	22	**3**	0	**2**
f	**3**	0	44	**5**	**6**
r	0	**3**	**3**	25	**1**
t	0	0	**2**	0	9

Table 2.3: Confusion matrices for BBCsports for k-NN with OT-NMF. Columns represent the ground truth and lines predicted labels. Labels: athletism (a), cricket (c), football (f), rugby (r) and tennis (t).

Taken from Rolet et al. [2016]

The classification error for OT-NMF with a French target vocabulary on BBCsports is rather bad, although the topics are coherent and related to the content of the articles. The confusion matrix (Table 2.3) shows that more than half of the articles about tennis are misclassified. In fact, the other methods produce a topic about tennis, but OT-NMF with a French dictionary does not. Table 2.4 shows the French words closest to some English query words according to the ground metric. While the closest words to football and bank are semantically related to their query word, the closest words to tennis are not. This illustrates how our method relies on the ground metric, given by word embeddings in this case.

football	football supporters championnat sportives sportifs joueurs sportif jeux matches sport
bank	banque banques bei bancaire federal bank emprunts reserve crédit bancaires
tennis	bienfaiteurs murray ex-membre ballet b92 sally sylvia markovic hakim socialo-communiste

Table 2.4: 10 French words closest to some English words according to the ground metric

Taken from Rolet et al. [2016]

2.4 Chapter Conclusion

We show how to efficiently perform dictionary leaning and NMF using optimal transport as the data fitting term, with an optional convex regularizer. Our method can be applied to large datasets in high dimensions and does not require any assumption on the cost matrix. We also show that with this data fitting term, the reconstruction $D\Lambda$ can use different features than the data X. Other than our application to cross-language topic modeling analysis, this can be used for example to reduce the number of target features by quantization for the dictionary while keeping the original features for the dataset.

While we only consider entropy as a barrier for positivity in this work, our approach is valid other regularizers, as long as the gradient of $R_\star$ can be computed efficiently. We believe that extensions to other classes of regularizers is an interesting area for future work.

Chapter 3

Blind Source Separation with Optimal Transport Non-negative Matrix Factorization

Learning Voice Models with Optimal Transport NMF

3.1 Chapter Introduction

Source separation is the task of separating a mixed signal into different components, usually referred to as sources. In the context of sound processing, it can be used to separate speakers whose voices have been recorded simultaneously. Blind source separation (BSS) aims at doing so with only sound data, that is without information such as the time when each source is active or the location of the sources with respect to the recording devices. A common way to address this task is to decompose the signal spectrogram by non-negative matrix factorization [NMF, Lee and Seung, 2001], as proposed for example by Schmidt and Olsson [2006] as well as Sun and Mysore [2013]. Denoting $\tilde{x}_{j,i}$ the (complex) short-time Fourier transform (STFT) coefficient of the input signal at frequency bin j and time frame i, and X its magnitude spectrogram defined as $x_{j,i} = |\tilde{x}_{j,i}|$, the BSS problem can be tackled by solving the NMF problem

$$\min_{D^{(1)}\dots D^{(N)}, W^{(1)}\dots W^{(N)}} \sum_{i=1}^{t} \ell\left(\boldsymbol{x}_i, \sum_{k=1}^{N} D^{(k)} \boldsymbol{w}_i^{(k)}\right) \tag{3.1}$$

where N is the number of sources, t is the number of time windows, $\boldsymbol{x}_i$ is the i^{th} column of X and ℓ is a loss function. Each dictionary matrix $D^{(k)}$ and weight matrix $W^{(k)}$ are related to a single source. In a supervised setting, each source has training data and all the $D^{(k)}$s are learned in advance during a training phase. At test time, given a new signal, separated spectrograms are recovered from the $D^{(k)}$s and $W^{(k)}$s and corresponding signals can be reconstructed with suitable post-processing.

In the present chapter, we propose to use *optimal transport* as a loss between spectrograms to perform supervised speech BSS with NMF. Optimal transport is defined as the minimum cost of moving the mass from one histogram to another. By taking into account a transportation cost between frequencies, this provides a powerful metric to compare STFT spectrograms. One of the main advantage of using optimal transport as a loss is that it can quantify the amplitude of a frequency shift noise, coming for example from quantization or the tuning of a musical instrument. Other metrics such as the Euclidean distance or Kullback-Leibler divergence, which compare spectrograms element-wise, are almost blind to this type of noise (see Figure 3.1). Another advantage over element-wise metrics is that optimal transport enables the use of different quantizations, i.e. frequency supports, at training and test times. Indeed, the frequencies represented on a spectrogram depend on the sampling rate of the signal and the time-windows used for its computation, either of which can change between training and test times. With optimal transport, we do not need to re-quantize the training and testing data so that they share the same frequency support: optimal transport is well-defined between spectrograms with distinct supports as long as we can define a transportation cost between frequencies. Finally, the optimal transport framework enables us to generalize the Wiener filter, a common post-processing for source separation, by using optimal transport plans, so that it can be applied to data quantized on different frequencies.

Using optimal transport as a loss between spectrograms was proposed by Flamary

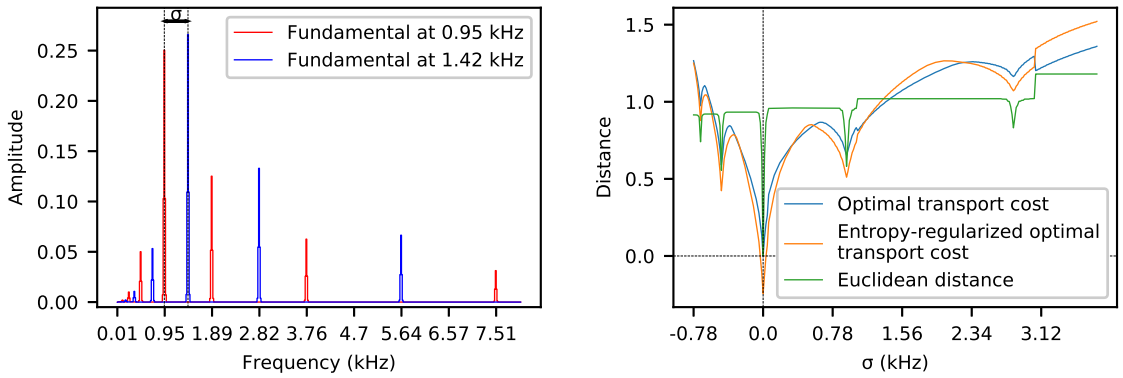


Figure 3.1: Comparison of Euclidean distance and (regularized) optimal transport losses. Synthetic musical notes are generated by putting weight on a fundamental, and exponentially decreasing weights on its harmonics and sub-harmonics, and finally convoluting with a Gaussian. Left: examples of the spectrograms of two such notes. Right: (regularized) optimal transport loss and Euclidean distance from the note of fundamental $0.95kHz$ *(red line on the left plot) to the note of fundamental* $0.95kHz+\sigma$*, as functions of* σ*. The Euclidean distance varies sharply whereas the optimal transport loss captures more smoothly the change in the fundamental. The variations of the optimal transport loss and its regularized version are similar, although the regularized one can become negative.*

Taken from Rolet et al. [2018]

et al. [2016] under the name "optimal spectral transportation". They developed a novel method for unsupervised music transcription which achieves state-of-the-art performance. Their method relies on a cost matrix designed specifically for musical instruments, allowing them to use Diracs as dictionary columns. That is, they *fix* each dictionary column to a vector with a single non-zero entry and learn only the corresponding coefficients. This trivial structure of the dictionary results in efficient coefficient computation. However, this approach cannot be applied as is to speech separation since it relies on the assumption that a musical note can be represented as its fundamental. It also requires designing the cost of moving the fundamental to its harmonics and neighboring frequencies. Because human voices are intrinsically more complex, it is therefore necessary to learn *both* the dictionary and the coefficients, i.e., solve full NMF problems.

In this chapter we start by showing how previous works have been performing blind source separation with non-negative matrix factorization. We then propose a distance between frequencies, which allows us to perform optimal transport non-negative matrix factorization on STFT spectrograms. We apply our NMF framework to isolated voice reconstruction and show that an optimal transport loss yields better results than other classical losses. We show that optimal transport yields comparable results to other losses for BSS, where the sources to separate are voices. Moreover we show that optimal transport achieves better results than other losses for learning a "universal" voice model, *i.e.* a model that can be applied to any voice, regardless of

the speaker. We use this universal voice model to perform speech denoising, which is BSS where one of the source is a voice and the other is noise. Finally, we show how to use our framework for cross-domain BSS, where frequencies represented in the test spectrograms may be different from the ones in the dictionary. This may happen for example when train and test data are recorded with different equipment, or when the STFT is computed with different parameters.

3.2 Signal Separation With NMF

We use a supervised BSS setting similar to the one described in Schmidt and Olsson [2006]. For each source k we have access to training data $X^{(k)}$, on which we learn a dictionary $D^{(k)}$ with NMF

$$\min_{W,D^{(k)}} \sum_{i=1}^{t} \ell(\boldsymbol{x}_i, D^{(k)}\boldsymbol{w}_i) + R_1(W) + R_2(D^{(k)}).$$

Then, given the STFT spectrum of a mixture of sources X, we reconstruct separated spectrograms $X^{(k)} = D^{(k)}W^{(k)}$ for $k = 1, \dots N$ where $W^{(k)}$s are the solutions of

$$\min_{W^{(1)},\dots,W^{(N)}} \sum_{i=1}^{t} \ell(\boldsymbol{x}_i, \sum_{k=1}^{N} D^{(k)}\boldsymbol{w}_i^{(k)}) + \sum_{k=1}^{N} R_1(W^{(k)}).$$

The separated spectrograms $\hat{X}^{(k)}$ are then reconstructed from each $X^{(k)}$ with the process described in Section 3.3.2.

In practice at test time, the dictionaries are concatenated in a single matrix $D = (D^{(k)})_{k=1}^{N}$, and a single matrix of coefficients W is learned, which we decompose as $W = (W^{(k)})_{k=1}^{N}$. This allows us to focus on problems of the form

$$\min_{W,D} \sum_{i=1}^{t} \ell(\boldsymbol{x}_i, D\boldsymbol{w}_i) + R_1(W) + R_2(D).$$

3.2.1 Voice-Voice Separation

We use the method described to separated the voices of two speakers on the same soundtrack. In this case, we have access to training data on each speaker.

3.2.2 Denoising with Universal Models

We can also use BSS to denoise speech data. In this case, we do not have access to training data for speakers in the test set. We only have access to data of other speakers, which we use to learn a "universal" voice model, as in Sun and Mysore [2013]. We also have two sources, the first one being a speaker and the second one a noise source. Here, we are only interested in the reconstruction of the voice, that is $\hat{X}^{(1)}$.

3.3 Method

We now present our approach for optimal transport BSS. First we define a transportation cost between frequencies, which allows us to perform optimal transport NMF on STFT spectrograms. We further show a new method for going from a reconstructed (separated) spectogram to a sound output, using optimal transport with our transportation cost.

3.3.1 Cost Matrix Design

In order to compute optimal transport on spectrogams and perform NMF, we need a cost matrix C, which represents the cost of moving weight from frequencies in the original spectrogram to frequencies in the reconstructed spectrogram. Some general approaches to automatically generate a cost matrix based on the data at hand have been considered in the literature, notably in Cuturi and Avis [2014] and Huang et al. [2016]. These approaches however both require labeled data and cannot be applied to our problem. Instead we have to build a cost matrix based on what we know of the physical properties of the data.

Schmidt and Olsson [2006] use the Mel scale to quantize spectrograms, relying on the fact that the perceptual difference between frequencies is smaller for the high frequency domain than for the low frequency domain. Following the same intuition, we propose to map frequencies to a log-domain and apply a cost function in that domain. Let f_j be the frequency of the j-th bin in an input data spectrogram, where $1 \leq j \leq m$. Let $\hat{f}_{\hat{j}}$ be the frequency of the $\hat{j}$-th bin in a reconstruction spectrogram, where $1 \leq \hat{j} \leq n$. We define the cost matrix $C \in \mathbb{R}^{m \times n}$ as

$$c_{j\hat{j}} = \left| \log(\lambda + f_j) - \log(\lambda + \hat{f}_{\hat{j}}) \right|^p \tag{3.2}$$

with parameters $\lambda \geq 0$ and $p > 0$. Since the Mel scale is a log scale, it is included in this definition for some parameter λ. Some illustrations of our cost matrix for different values of λ are shown in Figure 3.2, with $p = 0.5$. It shows that with our definition, moving weights locally is less costly for high frequencies than low ones, and that this effect can be tuned by selecting λ.

Figure 3.3 shows the effect of p on the learned dictionaries. Using $p = 0.5$ yields a cost that is more spiked, leading to dictionary elements that can have several spikes in the same frequency bands, whereas $p \geq 1$ tends to produce smoother dictionary elements.

Note that with this definition and $p \geq 1$, C is a distance matrix to the power p when the source and target frequencies are the same. If $p = 0.5$, C is the point-wise square-root of a distance matrix and as such is a distance matrix itself. $\mathrm{OT}(.,.)^{1/p}$.

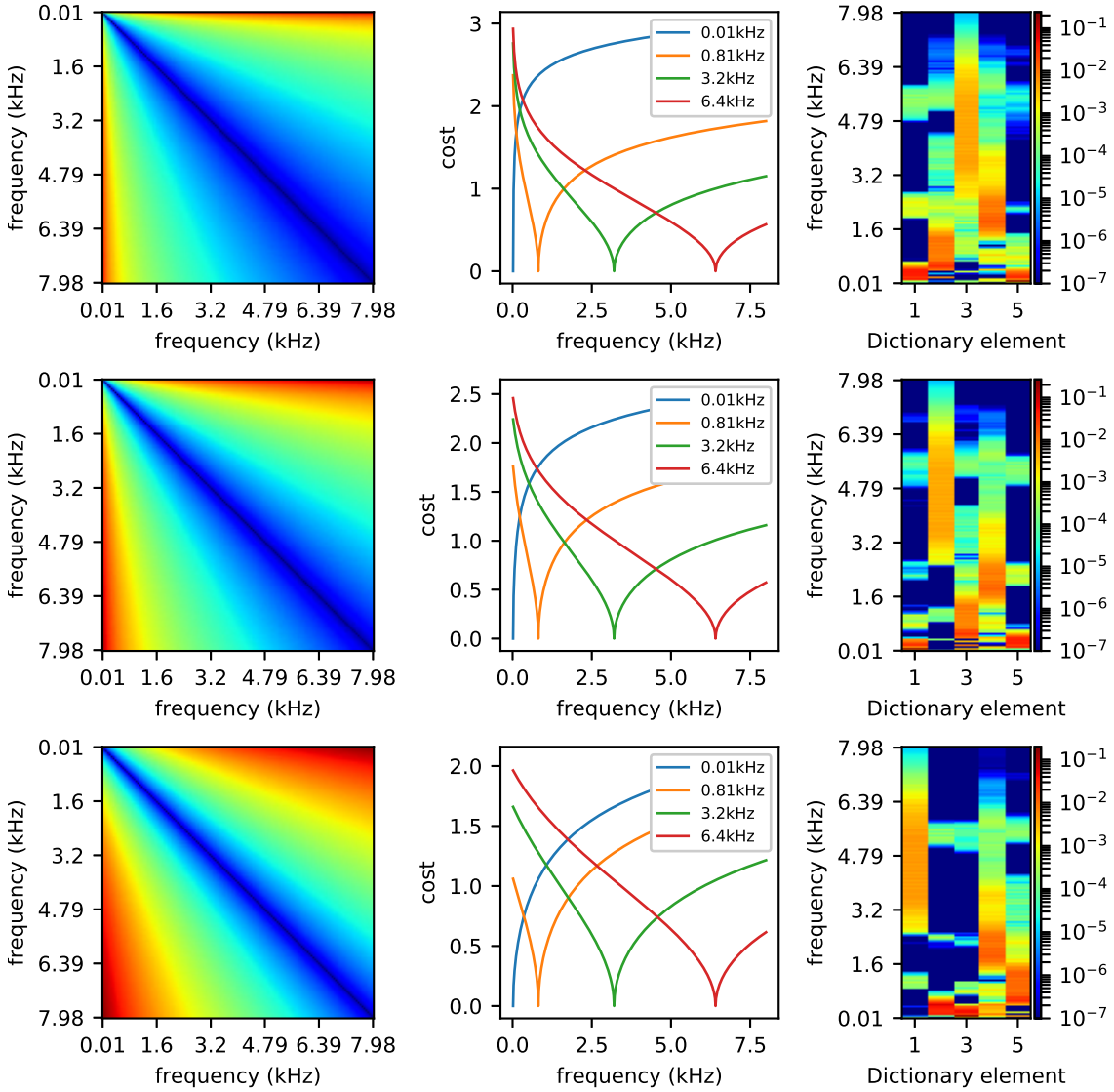


Figure 3.2: λ parameter of the Cost Matrix. Influence of parameter λ of the cost matrix. Left: cost matrix; center: sample lines of the cost matrix; right: dictionary learned on the validation data. Top: $\lambda = 1$; center: $\lambda = 100$; bottom: $\lambda = 1000$.

Taken from Rolet et al. [2018]

Parameters $p = 0.5$ and $\lambda = 100$ yielded better results for Blind Source Separation on the validation set and were accordingly used in all our experiments.

3.3.2 Post-processing

Separated spectra $X^{(1)}$ and $X^{(2)}$ do not give us separated sounds by themselves, they miss the phase information since they were computed on the power part X of the STFT spectrum only. If the separated spectra are in the same domain as X, it is

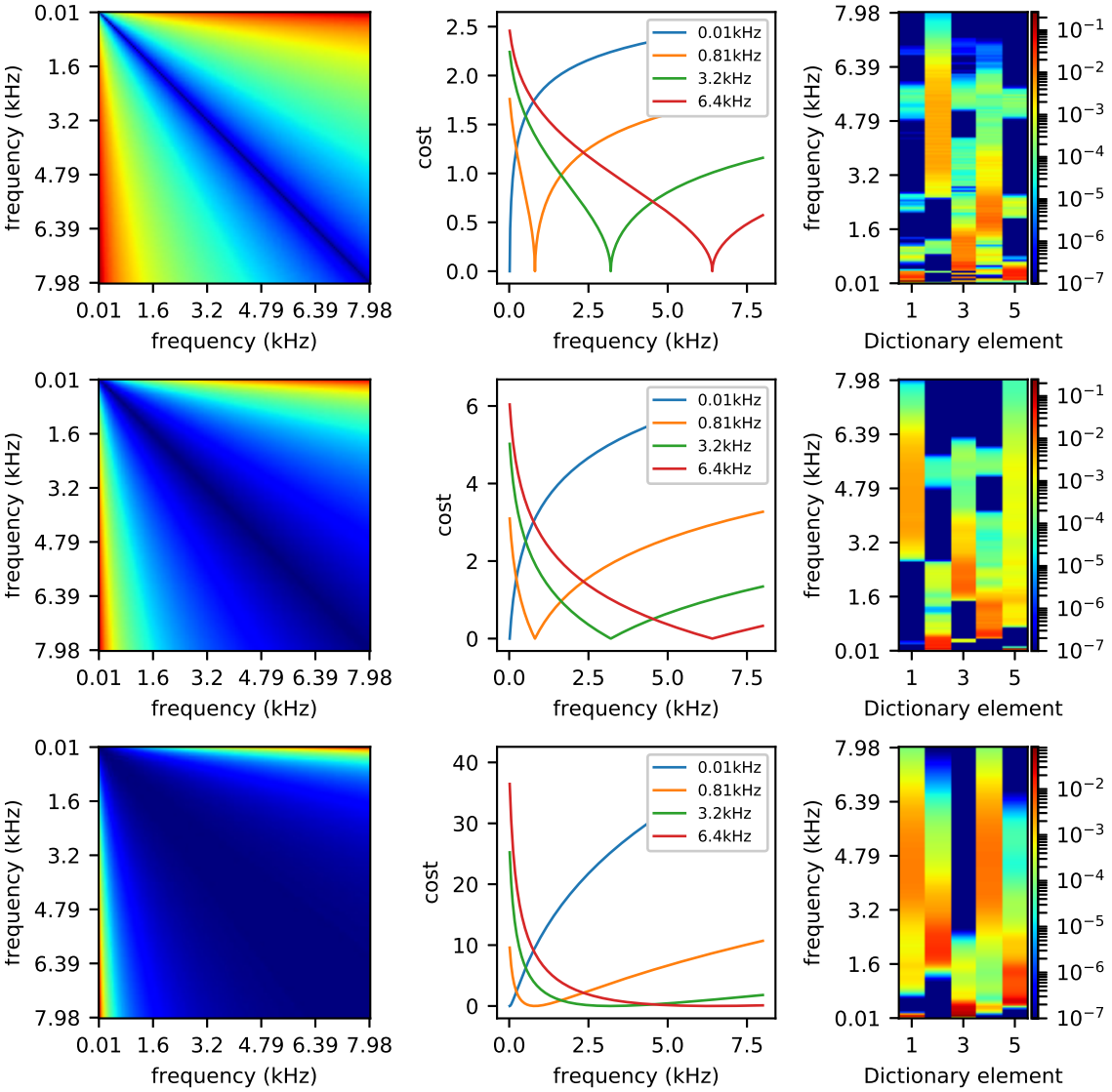


Figure 3.3: Power of the Cost Matrix. Influence of the power p of the cost matrix. Left: cost matrix; center: sample lines of the cost matrix; right: dictionary learned on the validation data. Top: $p = 0.5$; center: $p = 1$; bottom: $p = 2$.

Taken from Rolet et al. [2016]

possible to simply apply to them the phase of the original STFT spectrum, however this produces very audible artifacts, making the reconstructed voices sound "robotic". Instead a standard approach is to apply a *Weiner filter*. We start this section by describing this procedure in order to produce sound output for each separated source, and we follow by explaining how to use optimal transport concept in order to do so even when the reconstructed spectra are in a different domain.

3.3.2.1 Wiener Filter

In the case where the reconstruction is in the same frequency domain as the original signal, the classical way to recover each voice in the time domain is to apply a Wiener filter. Let X be the original Fourier spectrum, $X^{(1)}$ and $X^{(2)}$ the separated spectra such that $X \approx X^{(1)} + X^{(2)}$. The Wiener filter builds $\hat{X}^{(1)} = X \odot \frac{X^{(1)}}{X^{(1)}+X^{(2)}}$ and $\hat{X}^{(2)} = X \odot \frac{X^{(2)}}{X^{(1)}+X^{(2)}}$, before applying the original spectrum's phase and performing the inverse STFT.

3.3.2.2 Generalized Filter

We propose to extend this filtering to the case where $X^{(1)}$ and $X^{(2)}$ are not in the same domain as X. This may happen for example if the test data is recorded using a different sample frequency, or if the STFT is performed with a different time-window than the train data. In such a case, $D^{(1)}$ and $D^{(2)}$ are in the domain of the train data, and to are $X^{(1)}$ and $X^{(2)}$, but X is in a different domain, and its coefficients correspond to different sound frequencies. As such, we cannot use Wiener filtering.

Instead we propose to use the optimal transportation matrices to produce separated signals $\hat{X}^{(1)}$ and $\hat{X}^{(2)}$ in the same domain as X. Let $T_{(i)} \in \underset{\Pi \in U(\boldsymbol{x}_i, \boldsymbol{x}_i^{(1)}+\boldsymbol{x}_i^{(2)})}{\operatorname{argmin}} \langle C, \Pi \rangle$. With Wiener filtering, x_i is decomposed into its components generated by $\boldsymbol{x}_i^{(1)}$ and $\boldsymbol{x}_i^{(2)}$. We use the same idea and separate the transport matrix $T_{(i)}$ into:

$$T_{(i)}^{(1)} = T_{(i)} \operatorname{diag}\left(\frac{\boldsymbol{x}_i^{(1)}}{\boldsymbol{x}_i^{(1)} + \boldsymbol{x}_i^{(2)}}\right)$$
$$T_{(i)}^{(2)} = T_{(i)} \operatorname{diag}\left(\frac{\boldsymbol{x}_i^{(2)}}{\boldsymbol{x}_i^{(1)} + \boldsymbol{x}_i^{(2)}}\right)$$

$T_{(i)}^{(1)}$ (resp. $T_{(i)}^{(1)}$) is a transport matrix between $\frac{\boldsymbol{x}_i^{(1)}}{\boldsymbol{x}_i^{(1)}+\boldsymbol{x}_i^{(2)}}$ (resp. $\frac{\boldsymbol{x}_i^{(2)}}{\boldsymbol{x}_i^{(1)}+\boldsymbol{x}_i^{(2)}}$) and $\hat{\boldsymbol{x}_i}^{(1)}$ (resp. $\hat{\boldsymbol{x}_i}^{(2)}$), where

$$\hat{\boldsymbol{x}_i}^{(1)} = T^{(i)} \frac{\boldsymbol{x}_i^{(1)}}{\boldsymbol{x}_i^{(1)} + \boldsymbol{x}_i^{(2)}}$$
$$\hat{\boldsymbol{x}_i}^{(2)} = T^{(i)} \frac{\boldsymbol{x}_i^{(2)}}{\boldsymbol{x}_i^{(1)} + \boldsymbol{x}_i^{(2)}}$$

Similarly to the classical Wiener filter, we have

$$\begin{aligned}\hat{\boldsymbol{x}}_i{}^{(1)} + \hat{\boldsymbol{x}}_i{}^{(2)} &= T^{(i)} \frac{\boldsymbol{x}_i^{(1)}}{\boldsymbol{x}_i^{(1)} + \boldsymbol{x}_i^{(2)}} + T^{(i)} \frac{\boldsymbol{x}_i^{(2)}}{\boldsymbol{x}_i^{(1)} + \boldsymbol{x}_i^{(2)}} \\ &= T^{(i)} \mathbf{1} \\ &= \boldsymbol{x}_i\end{aligned}$$

Because of this property, the couple $(\hat{\boldsymbol{x}}_i{}^{(1)}, \hat{\boldsymbol{x}}_i{}^{(2)})$ is a fix point of the Wiener Filter. Similarly to the Weiner Filter, we then apply the original phase to $\hat{X}^{(1)}$ and $\hat{X}^{(2)}$ in order to produce separated sounds.

We show in the next section that using this *optimal transport* filter improves results on cross-domain experiments, when compared to the alternative of re-quantizing the dictionaries to fit the domain of the test data.

3.4 Results

In this section we present the main empirical findings of Rolet et al. [2018]. We start by describing the dataset that we used and the pre-processing we applied to it. We then show that the optimal transport loss allows us to have perceptually good reconstructions of single voices, even with few dictionary elements. We show that the optimal transport loss yields comparable results to other classical losses for voice-voice BSS with an NMF model. We also show that our generalized filter yields very similar results to the Wiener filter in the single-domain setting, and can improve upon it in the cross-domain setting. Finally, we show that the optimal transport improves upon these other losses when using a universal voice model for voice denoising.

3.4.1 Dataset and Pre-processing

3.4.1.1 Voice data

We evaluate our method on the English part of the Multi-Lingual Speech Database for Telephonometry 1994 dataset[1]. The data consists of recordings of the voice of four males and four females pronouncing each 24 different English sentences. We split each person's audio file time-wise into 25%-75% train-test data.

3.4.1.2 Noise data

For the speech denoising experiment we consider 4 types of noises: cicadas, drums, subway and sea. For each we gathered one file for training and one file for testing from non-copyrighted sources on the internet[2]. We trimmed the training files so that they are approximately 20 seconds long, and made sure that test files were longer than the voice test sounds. Note that for each noise type the training and testing files were gathered using the same keywords, but can still have quite a bit of variability.

3.4.1.3 Pre-processing

All sound files are re-sampled to $16kHz$ and treated as mono signal. The signals are analyzed by STFT with a Hann window, and a window-size of 1024, leading to 513 frequency bins ranging from 0 to 8kHz. The constant coefficient is removed from the NMF analysis and added for reconstruction in post-processing.

[1] `http://www.ntt-at.com/product/speech2002/`

[2] See availability of data section in Rolet et al. [2018]

3.4.1.4 Parameter selection

Hyper-parameters are selected on validation data consisting if the first male and female voice, which are excluded from the evaluation set. We choose the parameters which yield the best SDR score in the voice-voice BSS experiment for these voices. We also use these voices as the training data for the universal voice model.

3.4.1.5 Initialization

Initialization is performed by setting each value of the dictionary matrix as a random number picked uniformly in $[0, 1]$. It would be possible to set each dictionary column to the optimal transport barycenter (computed for example with Benamou et al. [2015]) of all the time frames of the training data, and adding Gaussian noise (separately for each column). However we did not notice a significant improvement with this initialization, and we only report here the scores with completely random initialization so that the results are comparable to the other methods. When training a model for any loss, we perform the NMF 4 times and keep the model with minimum training loss to reduce the impact of random initialization.

3.4.2 NMF Audio Quality

We first show that using an optimal transport loss for NMF leads to better perceptual reconstruction of voice data. To that end, we evaluated the PEMO-Q score [Huber and Kollmeier, 2006] of isolated test voices.

3.4.2.1 Personal Voice Model

Figure 3.4 shows the mean and standard deviation of the scores for $k \in \{5, 10, 15, 20\}$ with optimal transport (OT), Kullback-Leibler (KL), Itakura-Saito (IS) or Euclidean (E) NMF. In this setting the dictionaries are learned separately on the training data for each voice. These dictionaries are the same as in the following single-domain voice-voice separation experiment. The PEMO-Q score of optimal transport NMF is higher for any value of k, although KL and IS results are still competitive. We found empirically that other scores such as SDR or SNR tend to be better for the Euclidean NMF, even though the reconstructed voices are clearly worse when listening to them (see additional files 1 and 2). Optimal transport can reconstruct clear and intelligible voices with as few as 5 dictionary elements.

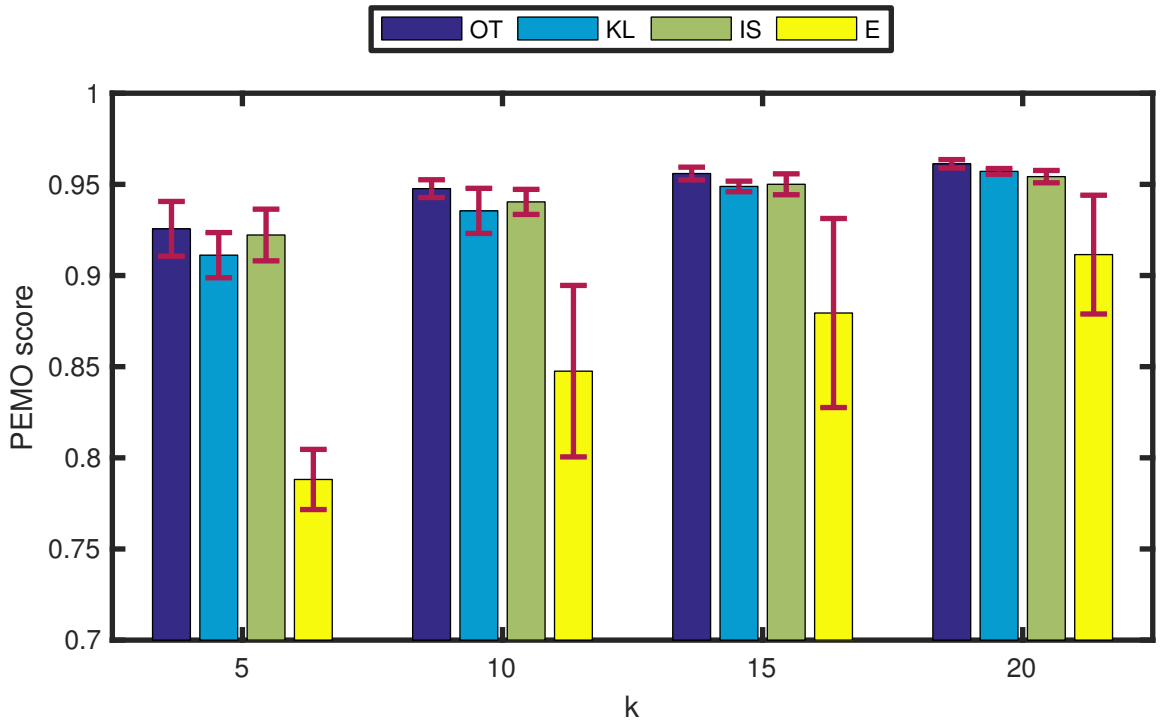


Figure 3.4: Perceptive Quality Score (personal voice model). Average and standard deviation of PEMO scores of reconstructed isolated voices, where the model is learned using separate training data for each voice with optimal transport (dark blue), Kullback-Leibler (light-blue), Itakura-Saito (green) or Euclidean (yellow) NMF.

Taken from Rolet et al. [2018]

3.4.2.2 Universal Voice Model

Figure 3.5 shows the mean and standard deviation of the scores for $k \in \{5, 10, 15, 20\}$ with optimal transport, Kullback-Leibler, Itakura-Saito or Euclidean NMF, in the universal voice model setting. Here only one dictionary is learned for all voices, with the training data of our validation voices. We kept this dictionary for the speech denoising experiment. The PEMO-Q score of optimal transport NMF is significantly higher for any value of k. We believe that because optimal transport compares spectrogram by looking at the optimal flow between their frequencies, the variation of pitch between two speakers become less important that the overall patterns of human voices. Indeed the scores with optimal transport are very similar whether we use a universal or a personal voice model, whereas they drop significantly for the other losses when using a universal model.

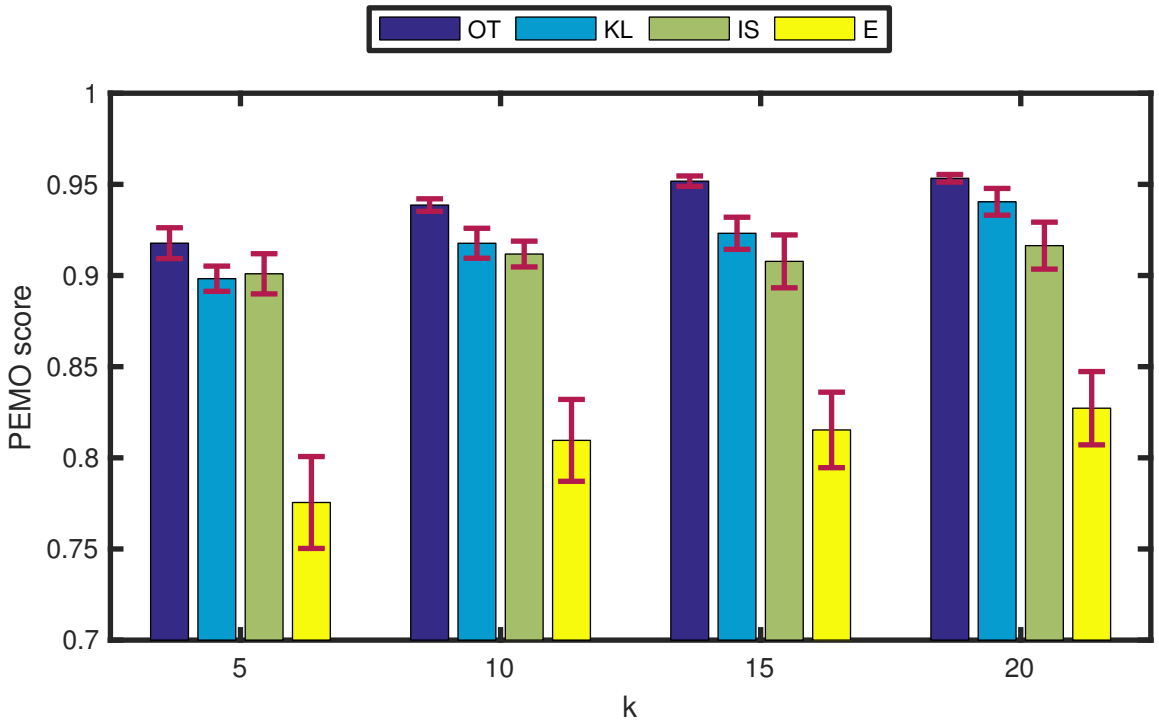


Figure 3.5: Perceptive Quality Score (universal voice model). Average and standard deviation of PEMO scores of reconstructed isolated voices, where the model is learned using the same training data for all voices with optimal transport (dark blue), Kullback-Leibler (light-blue), Itakura-Saito (green) or Euclidean (yellow) NMF.

Taken from Rolet et al. [2018]

3.4.3 Voice-voice Blind Source Separation

We evaluate our Blind Source Separation using the classical signal-to-distortion ratio (SDR) scores evaluated on reconstructed audio files using the Matlab toolbox BSS eval v2.1 [Vincent et al., 2006].

3.4.3.1 Single-Domain Blind Source Separation

We first use NMF to perform BSS in the case of mixtures of two voices, where we have training data for each voice. Here the spectrograms of the training and test data represent the same frequencies: both the training and test data are processed in exactly the same way, so that at train and test time $(f_i)_i = (\hat{f}_i)_i$. We compare using the optimal transport loss for NMF to the Kullback-Leibler divergence, the Itakura-Saito divergence or the Euclidean distance. For baseline methods, we reconstruct the signal using a Wiener filter before applying inverse STFT. For optimal transport-based source separation, we evaluate separation using either the Wiener filter or our generalized filter.

Figure 3.6 shows mean and standard deviation of the SDR, SIR and SAR scores for each method. We can see that although KL NMF achieves a better SDR score, the variability is actually high and the results are comparable for all method.

3.4.3.2 Cross-Domain Blind Source Separation

In this experiment, we artificially generate spectrograms which represent different frequencies for the training and test data by simply changing the STFT window size. For the training data we use a window of size 512, and a window of size 800 for the test data.

Although $(f_i)_i \neq (\hat{f}_i)_i$, we can still compute optimal transport between the spectrograms thanks to our cost matrix, and thus we can use the trained dictionary as is to compute the weight matrix at test time.

In order to compute the weight matrix for the other losses however, we first need to re-quantize the dictionary matrix so that it represents the same frequencies as the test data. We do it by assigning each frequency in the smaller spectrogram to its closest frequency in the larger one. This can be done with the simple linear operation $D \leftarrow AD$ with

$$a_{i,j} = \begin{cases} 1 & \text{if } j = \min \operatorname*{argmin}_k |f_i - \hat{f}_k| \\ 0 & \text{otherwise.} \end{cases}$$

Figure 3.7 shows mean and standard deviation of the SDR, SIR and SAR scores for each method. In the case of the optimal transport loss, we report both the result with the generalized filter, and the Wiener filter applied to $AX^{(k)}$. We can see that the SDR scores have dropped a lot, except with the optimal transport loss combined to our generalized filter. We notice a similar effect on the signal-to-artifact ratio (SAR), meaning that the separation process has created artifacts, which are actually very noticeable when listening the the reconstructed sound, except when using the generalized filter. This is probably due to the fact that the heuristic mapping process cancels a lot of frequencies which were in the test data.

3.4.4 Universal Voice Model for Speech Denoising

3.4.4.1 Setting

We now use NMF to first learn a universal speech model and noise models, and then apply these models for speech denoising. The universal speech model is learned on the concatenated training data of the first male and first female voices of our dataset. For each noise type, we learn a model with NMF on its training data. We then mix test voices with test noise with a pSNR of 0, and use our BSS approach to separate the

voice. All the scores reported are evaluated on the voices only, since reconstruction of the noise is not our goal here.

In this experiment we kept the same parameters for the cost matrix of optimal transport as in the ones selected in the voice-voice BSS experiment. We report the scores for each dictionary size k in $\{5, 10, 15, 20\}$.

3.4.4.2 Results

We can see from Table 3.1 and Table 3.2 that the optimal transport yields significantly better SDR and SIR than other methods for all noises except "sea". This is consistent with our observation that the optimal transport loss allows to good reconstruction with a universal model.

	OT				OT + OT filter				KL				IS				E			
	k				k				k				k				k			
	5	10	15	20	5	10	15	20	5	10	15	20	5	10	15	20	5	10	15	20
cicada	7.7 ± 0.1	8.8 ± 0.1	**8.9** ± 0.1	8.4 ± 0.1	7.3 ± 0.1	8.0 ± 0.1	8.6 ± 0.1	8.1 ± 0.1	7.7 ± 0.1	7.9 ± 0.1	7.9 ± 0.1	7.9 ± 0.1	7.7 ± 0.1	7.9 ± 0.1	7.6 ± 0.1	7.7 ± 0.1	7.9 ± 0.1	8.0 ± 0.1	7.9 ± 0.1	7.5 ± 0.1
drums	1.3 ± 0.6	3.6 ± 0.7	2.7 ± 0.7	2.8 ± 0.5	1.5 ± 0.5	**3.8** ± 0.6	2.7 ± 0.7	2.9 ± 0.6	2.0 ± 0.3	3.3 ± 0.7	2.6 ± 0.4	3.1 ± 0.4	0.5 ± 0.2	0.9 ± 0.1	0.6 ± 0.1	1.0 ± 0.0	1.9 ± 0.6	3.4 ± 0.5	2.0 ± 0.4	2.0 ± 0.3
sea	0.0 ± 0.9	1.5 ± 0.7	3.3 ± 0.5	1.8 ± 1.1	0.0 ± 0.8	1.8 ± 0.6	3.3 ± 0.5	1.9 ± 1.0	1.6 ± 1.3	3.4 ± 1.0	**4.6** ± 0.8	4.3 ± 0.7	1.6 ± 0.8	3.0 ± 0.6	3.7 ± 0.6	3.5 ± 0.5	3.5 ± 1.1	4.1 ± 0.9	4.4 ± 0.9	3.8 ± 0.6
subway	2.0 ± 1.1	**2.8** ± 0.9	1.5 ± 0.9	2.2 ± 1.2	1.8 ± 1.0	2.8 ± 1.0	1.6 ± 0.9	2.3 ± 1.2	1.8 ± 1.3	2.0 ± 1.6	1.9 ± 0.9	1.8 ± 0.9	2.0 ± 0.6	1.4 ± 0.3	1.7 ± 0.3	2.1 ± 0.4	1.5 ± 1.9	1.8 ± 1.2	1.7 ± 1.0	1.7 ± 0.9

Table 3.1: Speech denoising SDR scores

Taken from Rolet et al. [2018]

	OT				OT + OT filter				KL				IS				E			
	k				k				k				k				k			
	5	10	15	20	5	10	15	20	5	10	15	20	5	10	15	20	5	10	15	20
cicada	8.5 ± 0.1	10.0 ± 0.0	**10.2** ± 0.1	9.6 ± 0.1	8.0 ± 0.1	8.8 ± 0.0	9.7 ± 0.1	9.1 ± 0.1	8.5 ± 0.1	8.8 ± 0.1	8.8 ± 0.1	8.9 ± 0.1	8.5 ± 0.1	8.8 ± 0.1	8.5 ± 0.1	8.6 ± 0.1	8.7 ± 0.1	8.8 ± 0.1	8.7 ± 0.1	8.4 ± 0.1
drums	1.9 ± 0.4	5.5 ± 0.7	3.6 ± 0.6	3.8 ± 0.6	2.1 ± 0.3	**5.6** ± 0.6	3.6 ± 0.6	3.9 ± 0.6	2.8 ± 0.2	4.2 ± 0.6	3.2 ± 0.4	3.6 ± 0.4	0.7 ± 0.2	1.1 ± 0.1	0.7 ± 0.1	1.1 ± 0.0	3.1 ± 0.6	4.7 ± 0.5	2.6 ± 0.4	2.5 ± 0.3
sea	1.4 ± 0.9	2.8 ± 0.6	4.6 ± 0.4	3.0 ± 1.0	1.3 ± 0.8	3.0 ± 0.6	4.6 ± 0.4	3.0 ± 1.0	4.7 ± 1.0	6.2 ± 0.8	6.7 ± 0.6	6.2 ± 0.5	4.2 ± 0.3	5.0 ± 0.5	5.7 ± 0.6	5.4 ± 0.4	**10.1** ± 1.0	8.9 ± 0.6	8.1 ± 0.6	5.5 ± 0.4
subway	6.2 ± 1.3	**6.4** ± 0.9	3.1 ± 1.0	4.7 ± 1.2	5.5 ± 1.0	5.9 ± 0.9	3.0 ± 1.0	4.7 ± 1.2	5.2 ± 1.6	4.5 ± 1.7	3.1 ± 0.9	3.1 ± 0.9	4.0 ± 0.8	2.2 ± 0.4	2.1 ± 0.4	2.9 ± 0.5	5.3 ± 2.3	4.8 ± 1.3	4.2 ± 1.0	3.4 ± 0.9

Table 3.2: Speech denoising SIR scores

Taken from Rolet et al. [2018]

	OT				OT + OT filter				KL				IS				E			
	k				k				k				k				k			
	5	10	15	20	5	10	15	20	5	10	15	20	5	10	15	20	5	10	15	20
cicada	16.1 ± 0.3	15.7 ± 0.3	15.2 ± 0.2	15.3 ± 0.2	16.5 ± 0.3	**16.6** ± 0.4	15.6 ± 0.2	15.7 ± 0.2	15.9 ± 0.3	15.7 ± 0.3	15.8 ± 0.3	15.4 ± 0.2	16.4 ± 0.4	15.9 ± 0.3	15.5 ± 0.3	15.3 ± 0.2	16.3 ± 0.4	16.2 ± 0.3	16.2 ± 0.3	15.6 ± 0.3
drums	12.5 ± 1.7	9.0 ± 0.5	11.6 ± 0.6	11.1 ± 0.4	12.9 ± 1.5	9.5 ± 0.5	11.8 ± 0.6	11.4 ± 0.5	11.7 ± 1.6	12.1 ± 0.7	13.4 ± 0.5	14.0 ± 0.4	17.9 ± 2.1	17.4 ± 1.1	**20.7** ± 0.4	19.9 ± 0.4	9.9 ± 0.8	10.5 ± 0.4	12.7 ± 0.4	13.7 ± 0.4
sea	8.1 ± 1.8	9.4 ± 0.9	10.3 ± 0.7	9.8 ± 1.0	8.5 ± 1.8	10.0 ± 0.9	**10.5** ± 0.7	10.2 ± 1.0	5.8 ± 1.4	7.6 ± 1.3	9.5 ± 1.2	9.8 ± 1.0	6.6 ± 1.5	8.5 ± 0.9	9.2 ± 0.7	9.3 ± 1.1	5.1 ± 1.4	6.4 ± 1.1	7.5 ± 1.1	9.9 ± 1.1
subway	5.0 ± 1.3	6.3 ± 1.0	8.6 ± 0.5	7.1 ± 1.0	5.4 ± 1.3	6.8 ± 1.1	8.8 ± 0.5	7.3 ± 1.0	5.9 ± 1.3	7.1 ± 1.2	10.0 ± 0.9	9.5 ± 1.0	7.8 ± 0.9	11.6 ± 1.1	**13.6** ± 0.7	11.9 ± 0.5	5.0 ± 1.4	6.3 ± 1.0	6.8 ± 1.0	8.3 ± 0.8

Table 3.3: Speech denoising SAR scores

Taken from Rolet et al. [2018]

3.4.4.3 Dictionaries

Figures 3.8 and 3.9 show the dictionaries learned for the universal voice model and the cicada noise respectively, with all losses and a dictionary size of 5 and 10. The dictionaries learned with optimal transport tend to be smoother, and maybe with less overlap between dictionary elements. They seem to have high activation on bands, rather than isolated frequencies, and each dictionary element has only a few bands with high activation. The IS loss seems to induce similar effect to a lesser extent, while the KL and even more so the Euclidean loss tend to be spiked, with a lot of spikes for a same dictionary element, and more redundancy between elements.

3.4.4.4 Running times

Our implementation of the method in Python with numpy on 3 CPU cores of $2.93gHz$ takes about 3 minutes to fully learn a dictionary of 5 elements on the cicada training data, which is about 20s long, leading to spectrograms in $\mathbb{R}^{512\times724}$. Test times are around 2 minutes for sound files of around 50s, which is not real-time but close. We used rather tight convergence criteria in these experiments and we believe that these times could be reduced by using better hardware (multi-core, GPUs) and looser convergence criteria. For comparison, computing times for the KL loss, with a similar alternate minimization scheme (with inner optimizations performed with the multiplicative updates of Lee and Seung [2001]) and the same convergence criteria is about 50s for training, and about 20s for testing.

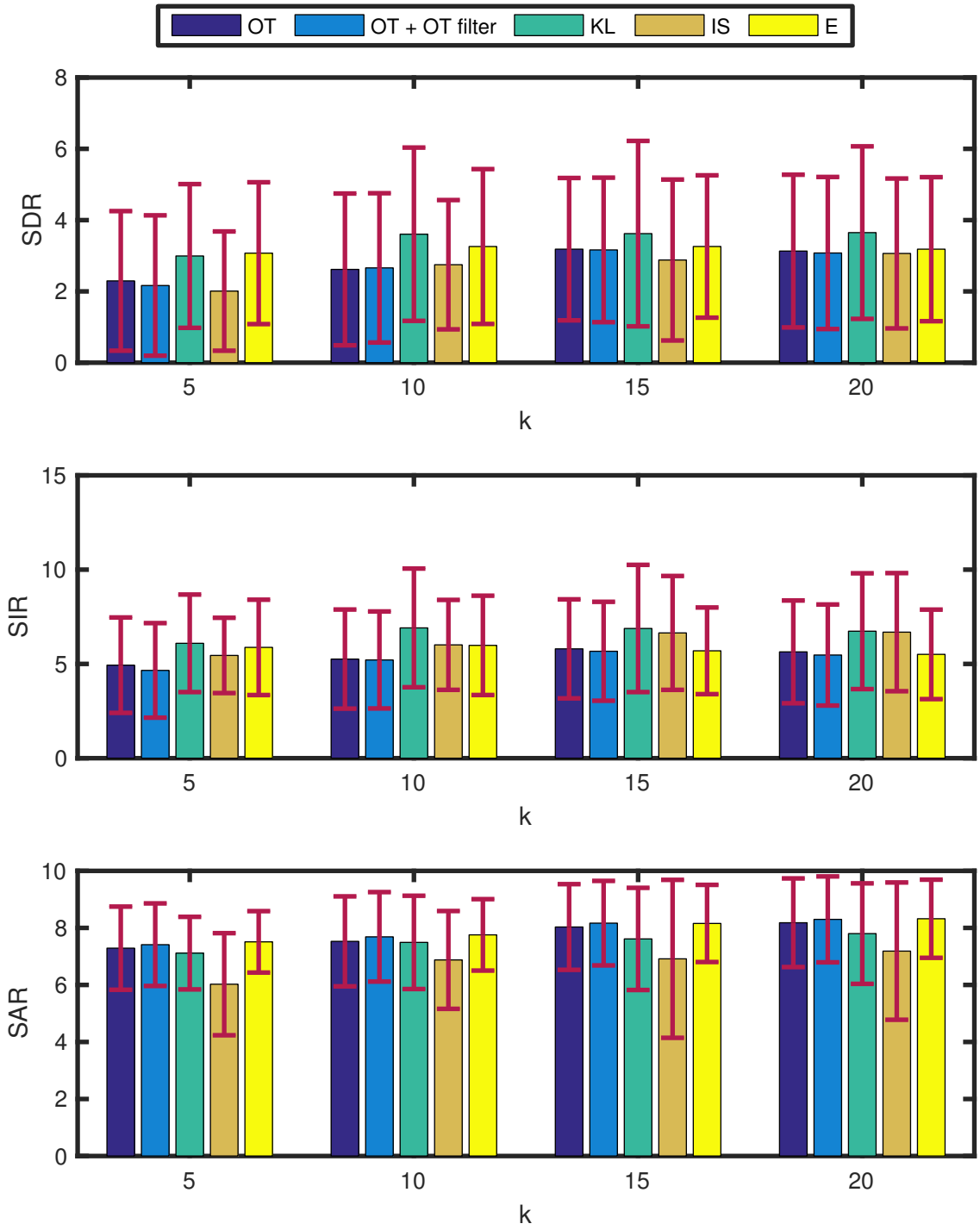


Figure 3.6: Voice-voice Separation Score (single-domain). Average and standard deviation of SDR, SIR and SAR scores for voice BSS, in the single-domain setting where training and testing spectrograms represent the same frequencies. The scores are for NMF with optimal transport (dark blue), optimal transport with our generalized filter (light blue), Kullback-Leibler (green), Itakura-Saito (brown) or Euclidean (yellow) NMF.

Taken from Rolet et al. [2018]

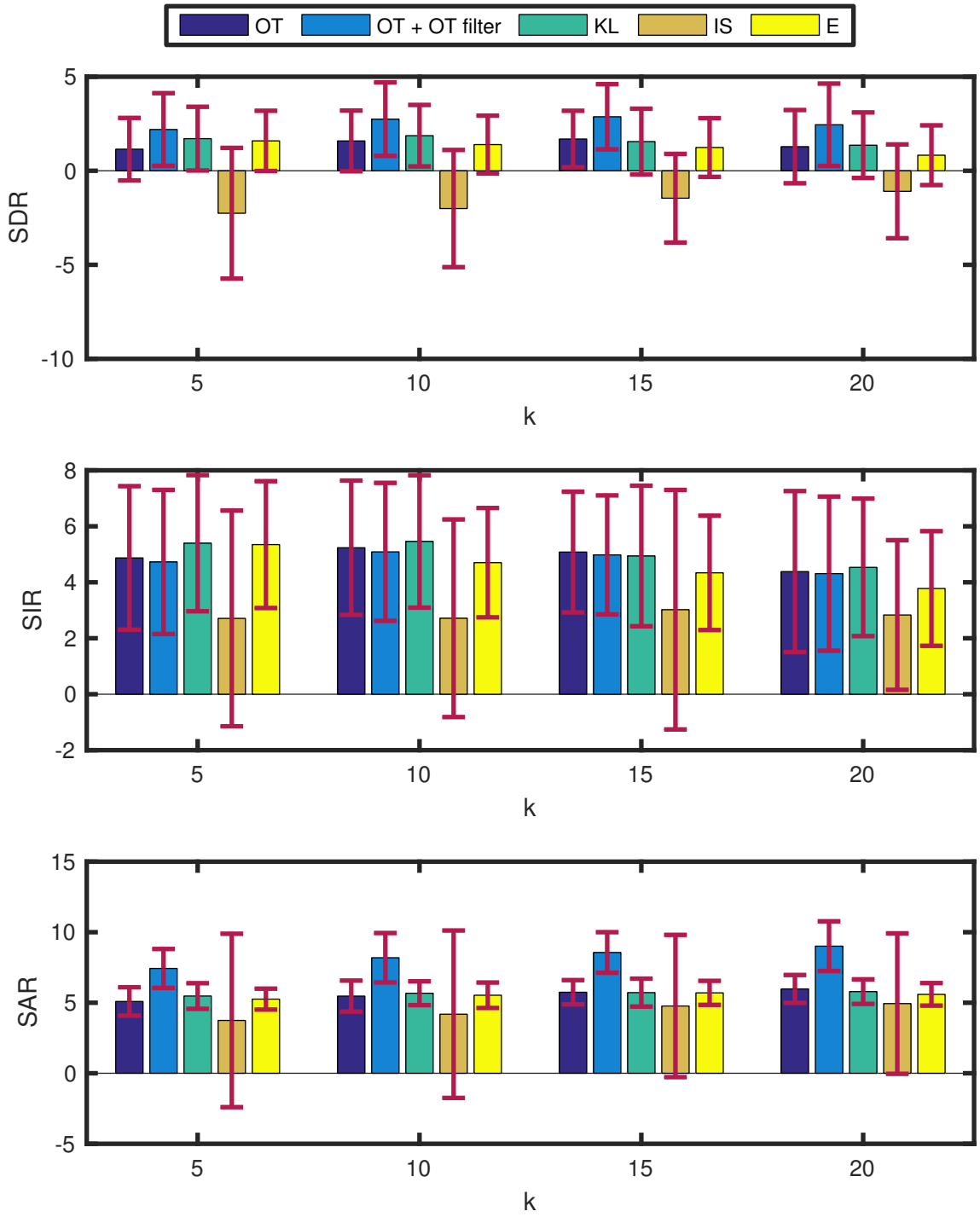


Figure 3.7: Voice-voice Separation Score (cross-domain). Average and standard deviation of SDR, SIR and SAR scores for voice BSS , in the cross-domain setting where training spectrograms have fewer frequencies than testing spectrograms. The scores are for NMF with optimal transport (dark blue), optimal transport with our generalized filter (light blue), Kullback-Leibler (green), Itakura-Saito (brown) or Euclidean (yellow) NMF.

Taken from Rolet et al. [2018]

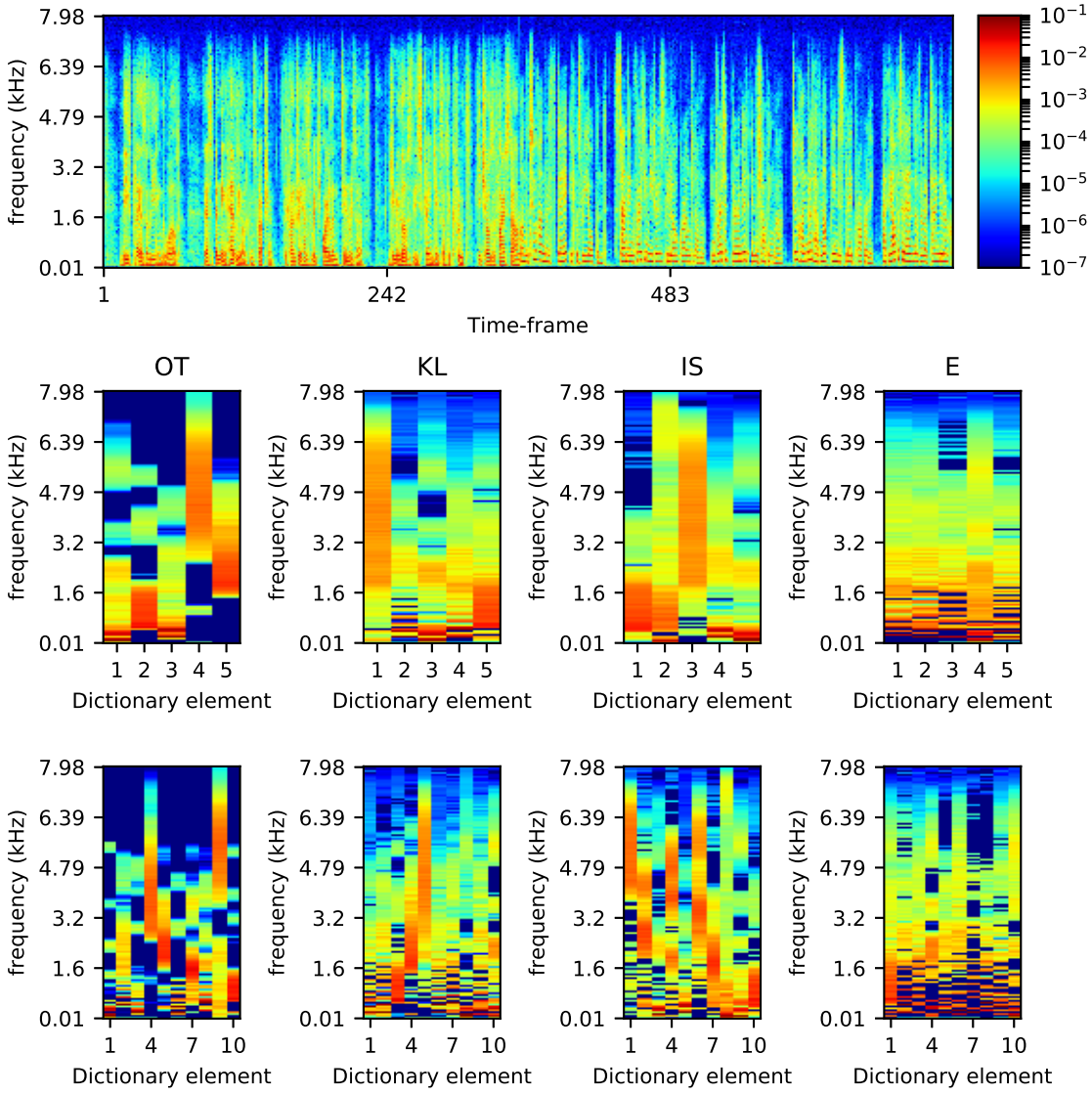


Figure 3.8: Universal Voice Model Dictionaries. Dictionaries learned for the universal model. Top row: spectrogram of the training data. Middle and bottom row: dictionaries learned with respectively 5 *and* 10 *elements, with the optimal transport, Kullback-Leibler, Itakura-Saito and Euclidean loss (from left to right).*

Taken from Rolet et al. [2018]

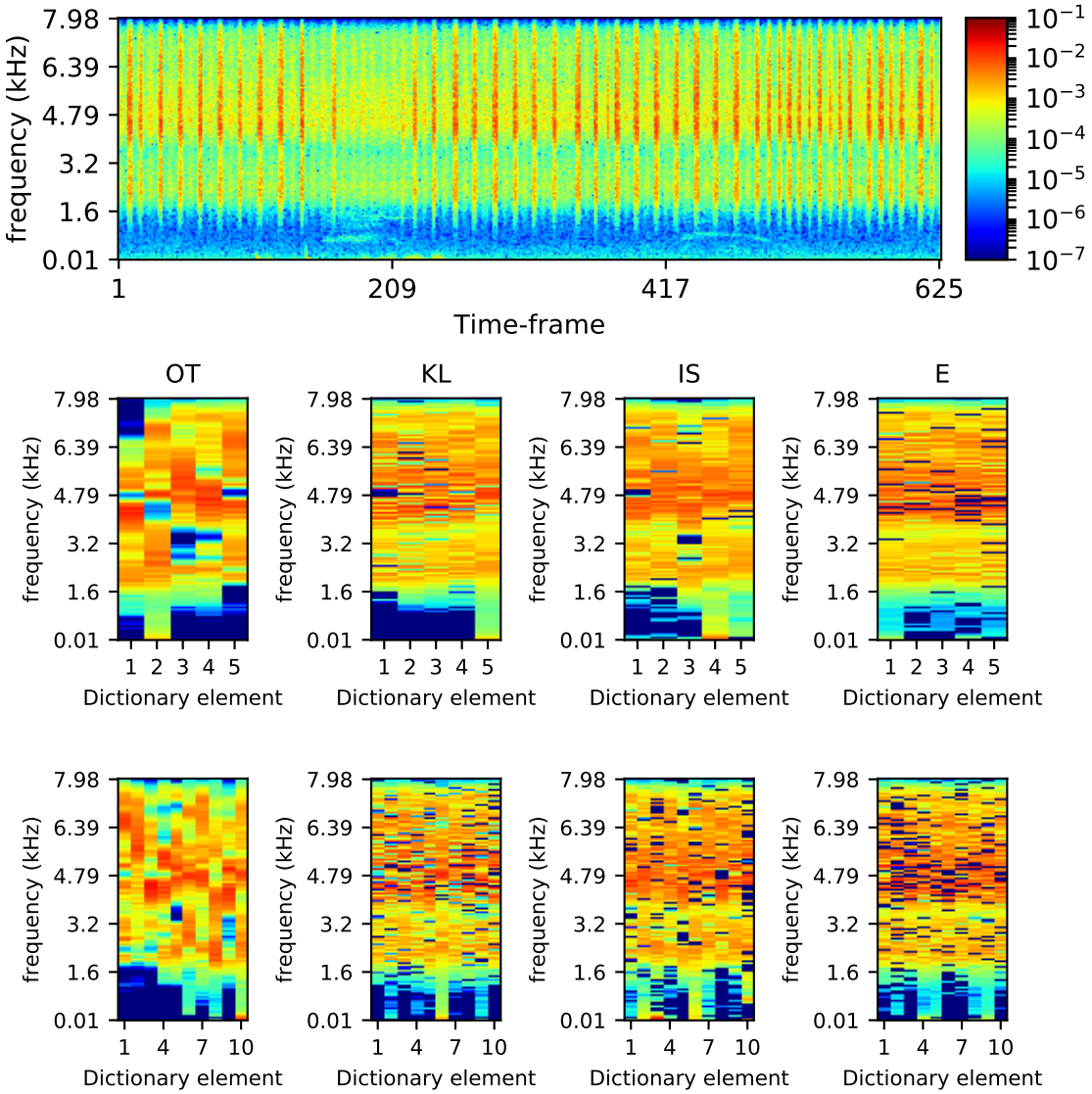


Figure 3.9: Noise Dictionaries. Dictionaries learned for the cicada noise. Top row: spectrogram of the training data. Middle and bottom row: dictionaries learned with respectively 5 *and* 10 *elements, with the optimal transport, Kullback-Leibler, Itakura-Saito and Euclidean loss (from left to right).*

Taken from Rolet et al. [2018]

3.5 Discussion

3.5.1 Regularization of the Transport Plan

In this work we considered entropy-regularized optimal transport as introduced by Cuturi [2013]. This allows us to get an easy-to-solve dual problem since its convex conjugate is smooth and can be computed in closed form. However, any convex regularizer would yield the same duality results, and could be considered as long as its conjugate is computable. For instance, the squared L^2 norm regularization was considered in several recent works [Blondel et al., 2018, Seguy et al., 2018] and was shown to have desirable properties such as better numerical stability or sparsity of the optimal transport plan. Moreover, similarly to entropic regularization, it was shown that the convex conjugate and its gradient can be computed in closed form [Blondel et al., 2018].

3.5.2 Learning Procedure

Following the work of Rolet et al. [2016], we solved the NMF problem with an alternating minimization approach, in which at each iteration a complete optimization is performed on either the dictionary or the coefficients. While this seems to work well in our experiments, it would be interesting to compare with smaller steps approaches like in Lee and Seung [2001]. Unfortunately such updates do not exist to our knowledge: gradient methods in the primal would be prohibitively slow, since they involve solving t large optimal transport problems at each iteration.

3.5.3 Future Work

3.5.3.1 Sparsity

Many works using NMF for sound processing add sparsity-inducing regularization to the NMF loss. This is usually achieved with a $l1$ regularization on the coefficient matrix W[Sun and Mysore, 2013, Li et al., 2004]. We believe such sparsity would also benefit our approach, although $l1$ regularization cannot be applied directly. Indeed we have constraints of the form $\|D\boldsymbol{w}_i\|_1 = \|\boldsymbol{x}_i\|_1$, and since all columns of D are in the simplex, this is equivalent to $\|\boldsymbol{w}_i\|_1 = \|\boldsymbol{x}_i\|_1$, so we already have a hard constraint on the $l1$ norm of W. One solution to this problem is to use an "unbalanced" optimal transport loss[Frogner et al., 2015, Chizat et al., 2018], for which both input do not need to have the same total weight. Unbalanced versions of optimal transport as defined in Chizat et al. [2018] do not have an easy to compute convex conjugate to the best of our knowledge, but Gramfort et al. [2015] casts unbalanced optimal

transport into a regular optimal transport problem, and our approach should work with this loss.

3.5.3.2 Multi-channel Sound Processing

In order to use our framework with multi-channel sound input, the main issue is to have an optimal transport loss between multi-channel spectrograms. A simple way to solve this is to simply treat channels separately and sum the loss on each channel. A more interesting approach in our opinion would be to design a cost matrix which would encode the cost of moving power not only between frequencies, but also between channels.

3.5.3.3 Optimal Transport in Other Models

We believe optimal transport can improve upon other losses between spectrograms in many sound processing tasks, as long as the loss is evaluated between spectrograms. For instance, one can use a speech-denoising auto-encoder as done by Ishii et al. [2013] and use the optimal transport loss with our proposed cost matrix on the reconstructed spectrograms. However the simple linear model of NMF used in our method allows us to have simple and easy to optimize duals. This is not the case with deep neural networks and one would have to resort to more computationally involved primal gradient-based approaches as in Frogner et al. [2015] or Montavon et al. [2016].

3.6 Chapter Conclusion

We showed that using an optimal transport based loss can improve performance of NMF-based models for voice reconstruction and separation tasks. We believe this is a first step towards using optimal transport as a loss for speech processing, possibly using more complicated models such as sparse NMF or deep neural networks. The versatility of optimal transport, which can compare spectrograms on different frequency domains, lets us use dictionaries on sounds that are not recorded or processed in the same way as the training set. This property could also be beneficial to learn common representations (*e.g.* dictionaries) for different datasets.

Chapter 4

Optimal Transport Regularized Projection

Optimal Transport Wavelet Shrinkage and Hard Thresholding for Image Processing

4.1 Chapter Introduction

Coefficient shrinkage has long been a staple method for signal denoising [Donoho, 1995, Kaur et al., 2002]. In its simplest form, it consists in soft-thresholding the coefficients of a signal in the spectral domain (*e.g.* wavelet or Fourier), before going back to the signal domain. Let $\boldsymbol{x}$ be a vector representing a signal, D be the matrix representing a wavelet or Fourier basis, and $\boldsymbol{\lambda}$ be the coefficients of $\boldsymbol{x}$ in the spectral domain ($\boldsymbol{x} = D\boldsymbol{\lambda}$). Coefficient shrinkage of $\boldsymbol{x}$ is $D\theta_\alpha(\boldsymbol{\lambda}) = D\theta_\alpha(D^{-1}\boldsymbol{x})$, for some $\alpha \geq 0$ and with $\theta_\alpha := \boldsymbol{\lambda} \mapsto \text{sign}(\boldsymbol{\lambda})(\boldsymbol{\lambda} - \alpha)_+$. Figure 4.1 explicits wavelet shrinkage as a 3-steps process: i) perform wavelet transform, ii) shrink the coefficients and iii) perform the inverse transform.

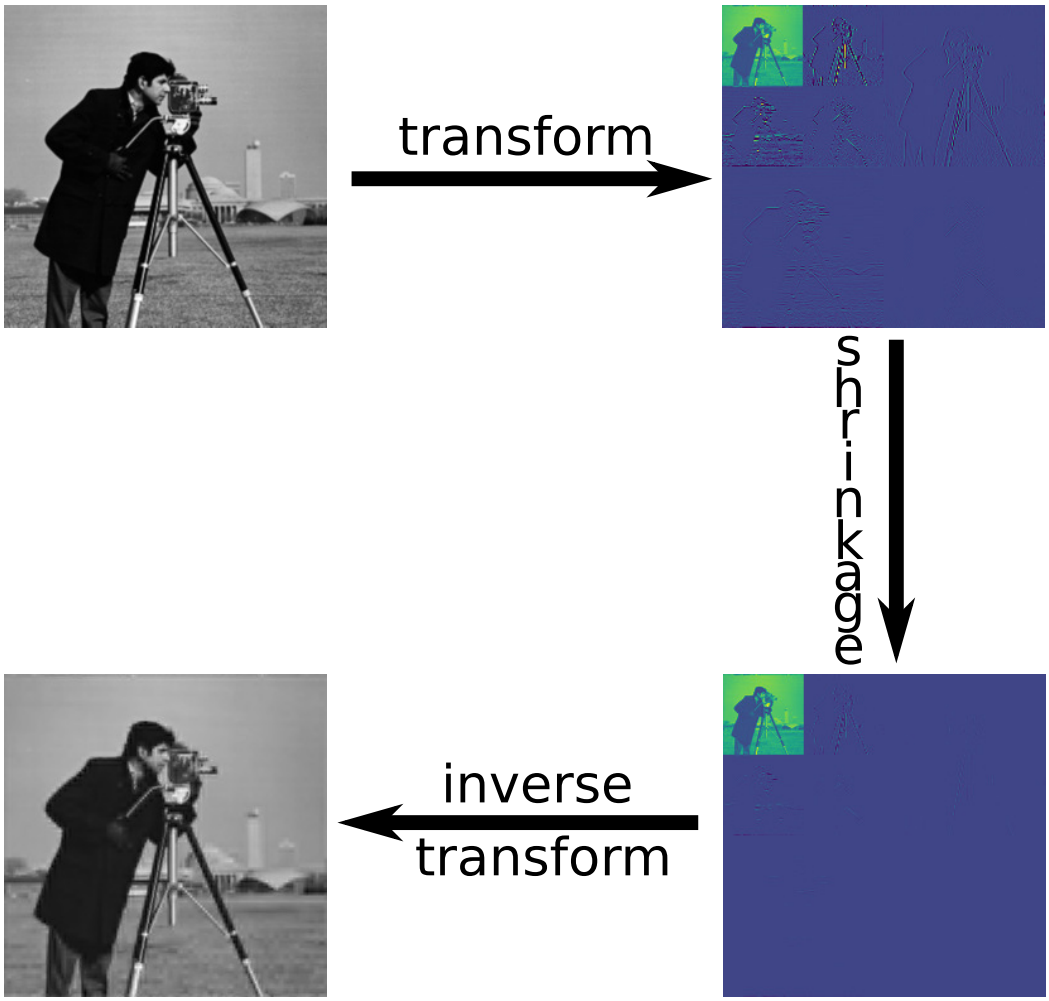


Figure 4.1: Wavelet coefficient shrinkage procedure, with Daubechies wavelets of order 2 at the second decomposition level.

If D is orthonormal, which is the case for orthogonal wavelets and the discrete

cosine transform, coefficient shrinkage amounts to the lasso problem:

$$\theta_\alpha(\boldsymbol{x}) = \operatorname*{argmin}_{\boldsymbol{\lambda}} \|\boldsymbol{x} - D\boldsymbol{\lambda}\|_2^2 + \alpha\|\boldsymbol{\lambda}\|_1.$$

More generally, this problem falls in the scope of regularized least square problems:

$$\min_{\boldsymbol{\lambda}} \|\boldsymbol{x} - D\boldsymbol{\lambda}\|_2^2 + R(\boldsymbol{\lambda}),$$

which can also be thought of as a regularized Euclidean projection, where $\|\boldsymbol{x} - D\boldsymbol{\lambda}\|_2^2$ is a closeness term and R is used to enforce desired properties on $\boldsymbol{\lambda}$. Using an ℓ_1 norm as R leads to coefficient shrinkage, while an indicator function leads to pass-type filtering for example.

Using the Euclidean distance as the signal closeness term leads to artifacts on the reconstructed image $D\boldsymbol{\lambda}$. For example, filtering out high frequency components in the Fourier domain tends to create a "wave" pattern around sharp edges (Figure 4.2). In order to reduce these artifacts we propose to use instead the optimal transport distance, which instead of comparing images pixel-by-pixel, compute the best way to "transport" the intensity of the pixels of an image to fit the other image. This means that images are compared overall, instead of separately for each pixel, yielding less artifact on the reconstructed image, as shown in Figure 4.2c compared to Figure 4.2b.

(a) Original image *(b) Euclidean filter* *(c) Optimal transport filter*

Figure 4.2: Effect of using the Euclidean or optimal transport distance as the closeness term for low pass filtering.

Taken from Rolet and Seguy [2021]

In this chapter, we study regularized projection of an image onto a fixed basis, or dictionary, where the reconstruction error is evaluated using the optimal transport distance. Projection onto a dictionary with respect to the optimal transport distance has been studied for musical note transcription [Flamary et al., 2016], and in the context of dictionary learning and non-negative matrix factorization [Sandler and Lindenbaum, 2009, Rolet et al., 2016, 2018]. However these works did not consider the effect of different regularizers, sparsity-inducing or otherwise, nor did they analyze the qualitative effect of using the optimal transport as the reconstruction error for image processing specifically.

Our contributions

We give simple conditions on D and R for existence and unicity of the optimal transport regularized projection. We derive a method to compute this projection that can be used for any convex regularizer R and dictionary D. We further give fast algorithms for special cases depending on the properties of R and D. This allows us to perform pass-type filtering and sparse decomposition of images onto wavelet or Fourier bases, which was not possible using the previously existing methods of Rolet et al. [2016]. Finally, we show how using the optimal transport distance as the reconstruction error leads to reduced artifacts for same level of sparsity when compared to the Euclidean distance.

This chapter is organized as follows: We start in Section 4.2 by giving computational methods for solving optimal transport regularized projection. Building on these methods, we show in Section 4.3 how to perform optimal transport hard and soft thresholding and pass-type filtering, and compare optimal transport to the Euclidean distance in each case.

4.2 Methods

We now show how to solve regularized optimal transport projection problems. Let us fix $D \in \mathbb{R}_+^{n \times k}$, $\boldsymbol{x} \in \mathbb{R}_+^n$, and let R be a convex function. The regularized optimal transport projection of $\boldsymbol{x}$ onto D is the solution of

$$\min_{\boldsymbol{\lambda} \in \mathbb{R}^k} \mathrm{OT}_\gamma(\boldsymbol{x}, D\boldsymbol{\lambda}) + R(\boldsymbol{\lambda}). \tag{4.1}$$

We proposed in Rolet et al. [2016] fast dual methods for this problem either without a regularizer, or where R is the entropy in order to enforce non-negativity, in the context of NMF. We extended their methods for convex regularizers R with a smooth convex conjugate R^* in Rolet and Seguy [2021], which is summarized in Theorem 2.2.5. We now show how to solve this problem when R^* is not smooth but D is orthonormal or simply invertible, using the methods we proposed in Rolet and Seguy [2021]. These methods work as long as we have access to the proximal operator or R^*, either through a formula or a tractable algorithm. Finally, we propose a general method which only requires a computable proximal operator for R.

4.2.1 Dual Problem

Most methods of this chapter are based on solving the dual problem defined in Theorem 2.2.5. In the regularized projection case, it can be rewritten as:

Theorem 4.2.1. *The solution $\boldsymbol{\lambda}^\star$ of Problem 4.1 satisfies the primal-dual relationship*

$$D\boldsymbol{\lambda}^\star = \nabla \mathrm{OT}_\gamma^*(\boldsymbol{x}, \boldsymbol{h}^\star) \tag{4.2}$$

where $\boldsymbol{h}^\star$ is the solution of the dual problem

$$\min_{\boldsymbol{h} \in \mathbb{R}^n} \mathrm{OT}_\gamma^*(\boldsymbol{x}, \boldsymbol{h}) + R^*(-D^\top \boldsymbol{h}). \tag{4.3}$$

The main point of working with Problem 4.3 instead of the primal problem directly is that it does not require us to compute OT_γ, but only its conjugate. On one hand, it allows us to get much faster algorithms (see Section 4.2.4), and on another it also allows us to solve some problems which we do not know how to solve in the primal, such as ℓ_1 regularized projection, *i.e.* coefficient shrinkage.

If R^* is smooth and its gradient can be computed efficiently, we can solve Problem (4.3) with an accelerated gradient method [Nesterov, 1983], as we did in Chapter 2.

The existence and unicity conditions of Chapter 2 still apply. In particular if D is invertible and $\gamma > 0$, there is a unique solution to Problem 4.1.

4.2.2 Saddle Point Problem

If R^* is not smooth, or if its gradient is expensive to compute, we can still compute the projection by solving a saddle point problem:

Theorem 4.2.2 (Primal-Dual)**.** *Let* $(\boldsymbol{h}^\star, \boldsymbol{\lambda}^\star)$ *be a solution of*

$$\min_{\boldsymbol{h}\in\mathbb{R}^n} \max_{\boldsymbol{\lambda}\in\mathbb{R}^k} \mathrm{OT}_\gamma^*(\boldsymbol{x}, \boldsymbol{h}) + \left\langle -D^\top \boldsymbol{h}, \boldsymbol{\lambda} \right\rangle - R(\boldsymbol{\lambda}). \tag{4.4}$$

Then $\boldsymbol{\lambda}^\star$ *is a solution of Problem 4.1.*

Proof. The proof follows the same path as in Rolet et al. [2016], where R was the non-negative entropy. We rewrite Problem (4.1) as:

$$\min_{\substack{\boldsymbol{\lambda}\in\mathbb{R}^k \\ \boldsymbol{p}\in\mathbb{R}_+^n \\ D\lambda=p}} \mathrm{OT}_\gamma(\boldsymbol{x}, \boldsymbol{p}) + R(\boldsymbol{\lambda}).$$

It is a convex problem with linear constraints so strong duality holds, the problem is then:

$$\max_{\boldsymbol{h}\in\mathbb{R}^n} \min_{\substack{\boldsymbol{\lambda}\in\mathbb{R}^k \\ \boldsymbol{p}\in\mathbb{R}_+^n}} \mathrm{OT}_\gamma(\boldsymbol{x}, \boldsymbol{p}) - \langle \boldsymbol{h}, \boldsymbol{p} - D\boldsymbol{\lambda} \rangle + R(\boldsymbol{\lambda}).$$

By definition of OT_γ^*, we get

$$\begin{aligned} &\max_{\boldsymbol{h}\in\mathbb{R}^n} \min_{\boldsymbol{\lambda}\in\mathbb{R}^k} - \mathrm{OT}_\gamma^*(\boldsymbol{x}, \boldsymbol{h}) + \langle \boldsymbol{h}, D\boldsymbol{\lambda} \rangle + R(\boldsymbol{\lambda}) \\ -&\min_{\boldsymbol{h}\in\mathbb{R}^n} \max_{\boldsymbol{\lambda}\in\mathbb{R}^k} \mathrm{OT}_\gamma^*(\boldsymbol{x}, \boldsymbol{h}) + \left\langle -D^\top \boldsymbol{h}, \boldsymbol{\lambda} \right\rangle - R(\boldsymbol{\lambda}) \end{aligned}$$

□

We propose to solve Problem 4.4 with a primal-dual approach such as Condat [2013] or Lorenz and Pock [2015]. We use the algorithm defined in Theorem 9 of Lorenz and Pock [2015] to make use of preconditioning. Following their notations, we set:

$$\begin{cases} Q = \mathrm{OT}_\gamma^*(\boldsymbol{x}, \cdot), & G = 0, \quad K = -D^\top, \\ F^* = R, & P^* = 0, \quad \alpha_k = 0,\ \forall k. \end{cases}$$

This leads to updates:

$$\begin{cases} \boldsymbol{h}^{k+1} = \boldsymbol{h}^k - \tau(\nabla\, \mathrm{OT}_\gamma^*(\boldsymbol{x}, \boldsymbol{h}^k) - D\boldsymbol{\lambda}^k) \\ \boldsymbol{\xi}^{k+1} = 2\boldsymbol{h}^{k+1} - \boldsymbol{h}^k \\ \boldsymbol{\lambda}^{k+1} = \mathrm{prox}_{\sigma R}(\boldsymbol{\lambda}^k - \sigma D^\top \boldsymbol{\xi}^{k+1}), \end{cases}$$

where prox_f denotes the proximal operator of a function f. Solving the saddle-point problem in that way tends to be slow compared to full dual approaches, as we show in Section 4.2.4. We now focus on special conditions which allow to expand on Theorem 4.2.1.

4.2.3 Special Case: Invertible Dictionary

In the case where R^* is not smooth, we cannot solve Problem (4.3) directly with first order methods. However if D is invertible we can rewrite the problem and solve it with proximal methods.

Theorem 4.2.3. *Let $D \in \mathbb{R}^{n\times n}$ be an invertible matrix. The solution $\boldsymbol{\lambda}^\star$ of Problem 4.1 satisfies*

$$\boldsymbol{\lambda}^\star = D^{-1}\nabla \operatorname{OT}^*_\gamma(\boldsymbol{x}, -D^{\top -1}\boldsymbol{g}^\star) \tag{4.5}$$

where $\boldsymbol{g}^\star$ is the solution of

$$\min_{\substack{\boldsymbol{g}\in\mathbb{R}^n \\ -D^\top \boldsymbol{h}=\boldsymbol{g}}} \operatorname{OT}^*_\gamma(\boldsymbol{x}, -D^{\top -1}\boldsymbol{g}) + R^*(\boldsymbol{g}). \tag{4.6}$$

Proof. Problem 4.6 is obtained by the change of variable $-D^\top \boldsymbol{h} = \boldsymbol{g}$ in Problem 4.3. This same change of variable gives us $D\boldsymbol{\lambda}^\star = \nabla \operatorname{OT}^*_\gamma(\boldsymbol{x}, -D^{\top -1}\boldsymbol{g}^\star)$. □

Assuming that we have access to the proximal operator of R^*, we can solve Problem 4.6 efficiently with a proximal method such as FISTA[Beck and Teboulle, 2009].

4.2.3.1 Orthonormal dictionary

In the case where D is orthonormal, the problem of learning the coefficients can be solved with the invertible special case.

Another solution arises if we rewrite Problem (4.3) as

$$\min_{\boldsymbol{h}\in\mathbb{R}^n} \operatorname{OT}^*_\gamma(\boldsymbol{x}, \boldsymbol{h}) + \Pi(\boldsymbol{h}), \tag{4.7}$$

where $\Pi(\boldsymbol{h}) = R^*(-D^\top \boldsymbol{h})$. We can solve this new problem with FISTA. Indeed, since D is orthonormal, the proximal operator prox_Π of Π can be computed easily. By definition we have

$$\text{prox}_\Pi(\boldsymbol{h}) = \operatorname*{argmin}_{\boldsymbol{y}\in\mathbb{R}^n} \|\boldsymbol{h} - \boldsymbol{y}\|^2 - R^*(-D^\top \boldsymbol{y}).$$

Using the change of variable $\boldsymbol{z} = -D^\top \boldsymbol{y}$, we have

$$\operatorname{prox}_{\Pi}(\boldsymbol{h}) = -D \operatorname*{argmin}_{\boldsymbol{z}\in\mathbb{R}^n} \|\boldsymbol{h} + D\boldsymbol{z}\|^2 - R^*(\boldsymbol{z}).$$

Since D is orthonormal, it follows that

$$\begin{aligned}\|\boldsymbol{h} + D\boldsymbol{z}\|^2 &= \| - D^\top \boldsymbol{h} - D^\top D\boldsymbol{z}\|^2 \\ &= \| - D^\top \boldsymbol{h} - \boldsymbol{z}\|^2.\end{aligned}$$

We can thus compute the proximal operator of Π from that of R^*:

$$\begin{aligned}\operatorname{prox}_{\Pi}(\boldsymbol{h}) =& - D \operatorname*{argmin}_{\boldsymbol{z}\in\mathbb{R}^n} \| - D^\top \boldsymbol{h} - \boldsymbol{z}\|^2 - R^*(\boldsymbol{z}) \\ =& - D \operatorname{prox}_{R^*}(-D^\top \boldsymbol{h}).\end{aligned}$$

The primal-dual relationship becomes

$$\boldsymbol{\lambda}^\star = D^\top \nabla \operatorname{OT}^*_\gamma(\boldsymbol{x}, \boldsymbol{h}^\star). \tag{4.8}$$

We sum up our proposed methods in Table 4.1.

Table 4.1: Algorithms available based on the properties of R and D

Conditions	Method	Gradient	Proximal operator	Primal-dual relationship
R^* differentiable	accelerated gradient[1]	$\nabla \operatorname{OT}^*_\gamma(\boldsymbol{x}, \boldsymbol{h})$ $-D\nabla R^*(-D^\top \boldsymbol{h})$	Not used	$D\boldsymbol{\lambda}^\star = \nabla \operatorname{OT}^*_\gamma(\boldsymbol{x}, \boldsymbol{h}^\star)$
D invertible	FISTA[2]	$-D^{-1}\nabla \operatorname{OT}^*_\gamma(\boldsymbol{x}, -D^{-1\top}\boldsymbol{g})$	$\operatorname{prox}_{R^*}(\boldsymbol{g})$	$\boldsymbol{\lambda}^\star = D^{-1}\nabla \operatorname{OT}^*_\gamma(\boldsymbol{x}, -D^{\top-1}\boldsymbol{g}^\star)$
D orthonormal	FISTA[2]	$\nabla \operatorname{OT}^*_\gamma(\boldsymbol{x}, \boldsymbol{h})$	$-D \operatorname{prox}_{R^*}(-D^\top \boldsymbol{h})$	$\boldsymbol{\lambda}^\star = D^\top \nabla \operatorname{OT}^*_\gamma(\boldsymbol{x}, \boldsymbol{h}^\star)$
None	forward-backward splitting[3]	$\nabla \operatorname{OT}^*_\gamma(\boldsymbol{x}, \boldsymbol{h})$	$\operatorname{prox}_R(\boldsymbol{h})$	$\boldsymbol{\lambda}^\star$ is already available

4.2.4 Time Comparisons

4.2.4.1 Primal VS dual algorithms

In order to compare computation times between a direct primal method and our dual algorithms, we have to select a problem for which similar algorithms can be used,

[1] Nesterov [1983]
[2] Beck and Teboulle [2009]
[3] Lorenz and Pock [2015]

FISTA in this case. In particular we need a problem which can be divided into a smooth part and a part for that has a tractable proximal operator. Let us consider the simple following problem:

$$\min_{\substack{\boldsymbol{\lambda}\in\mathbb{R}^k \\ \mathbf{1}^\top D\boldsymbol{\lambda}=\mathbf{1}^\top \boldsymbol{x} \\ D\boldsymbol{\lambda}\geq 0}} \mathrm{OT}_\gamma(\boldsymbol{x}, D\boldsymbol{\lambda}) + \alpha\|\boldsymbol{\lambda}\|_2^2,$$

where D is the an orthonormal matrix[4]. We can project any $\boldsymbol{\lambda}$ on the constraint $\mathbf{1}^\top D\boldsymbol{\lambda} = \mathbf{1}^\top \boldsymbol{x}$ by projecting $D\boldsymbol{\lambda}$ on the non-negative part of the ℓ_1 sphere of radius $\mathbf{1}^\top \boldsymbol{x}$, and then applying $D^\top$ to the result. The objective is differentiable, we compute the optimal transport part and its gradient with the Sinkhorn algorithm [Cuturi, 2013]. This algorithm's computational bottleneck is also the multiplication with $K = e^{\frac{C}{\gamma}}$, so it benefits from the convolution acceleration defined in Section 1.1.3 as much as our dual methods do.

We also solve the problem with the invertible case of Section 4.2.3, with $R(\boldsymbol{\lambda}) = \alpha\|\boldsymbol{\lambda}\|_2^2$. We then have $R^*(\boldsymbol{h}) = \frac{\alpha}{4}\|\boldsymbol{h}\|_2^2$ and $\mathrm{prox}_{R^*}(\boldsymbol{h}) = \frac{\boldsymbol{h}}{1+\alpha/2}$.

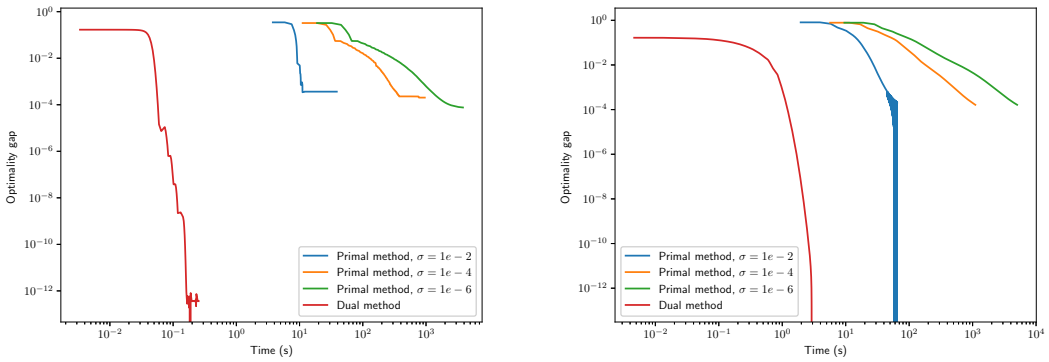


Figure 4.3: Optimality gap with respect to time for a simple l2-regularized projection with a primal approach or our dual approach. Left: FISTA with backtracking line-search. Right: FISTA with a fixed step-size.

Taken from Rolet and Seguy [2021]

Figure 4.3 shows a time comparison of the FISTA algorithm used to solve the primal or dual problem, with either a fixed step-size or a step-size chosen by backtracking line-search. As the figure shows, our dual algorithm is orders of magnitude faster in any of the settings. For both methods, the backtracking line-search heuristic for choosing the step-size leads to faster convergence. However for the primal method, the precision σ to which we solve the regularized transport problem has a direct influence on the quality of the gradient. As a result backtracking line-search is not able to select positive step-size after getting close to the optimal solution when the precision of the Sinkhorn algorithm is too low.

[4]Since D is orthonormal, the problem is actually equivalent to simply $\min_{\boldsymbol{\lambda}} \mathrm{OT}_\gamma(\boldsymbol{x}, \boldsymbol{\lambda}) + \alpha\|\boldsymbol{\lambda}\|_2^2$.

4.2.4.2 Saddle point VS dual algorithms

We now compare computation time for an optimal transport regularized projection problem using a primal-dual approach and a fully dual approach. We perform optimal transport coefficient shrinkage on the DCT coefficients of a 256×256 image using our dual approaches of Section 4.2.3 and the saddle point approach of Section 4.2.2. Although the saddle-point approach has the advantage of being valid for any dictionary D, Figure 4.4 shows that it is orders of magnitude slower to converge than dual approaches.

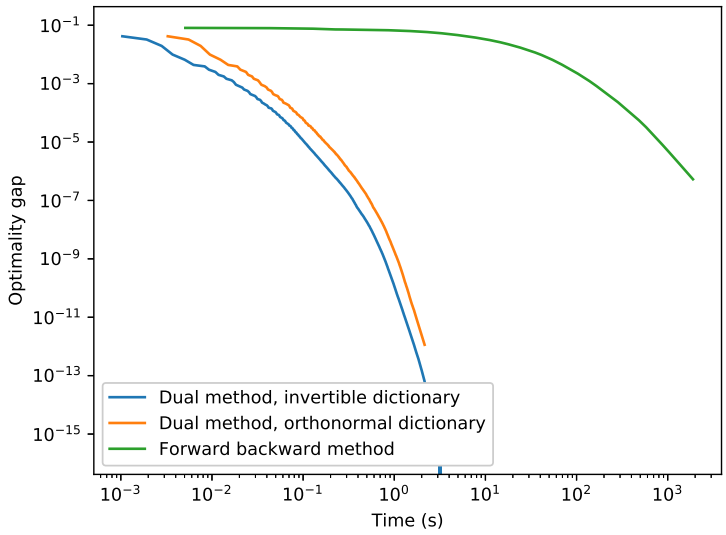


Figure 4.4: Computation time for a same sparse projection problem with a primal-dual method or dual methods.

Taken from Rolet and Seguy [2021]

4.3 Applications

In this section, we show how to use the fast regularized projection methods we derived to perform optimal transport filtering, coefficient shrinkage and hard thresholding. We examine qualitative and quantitative differences of using optimal transport instead of the Euclidean distance on different image processing tasks, namely low-pass filtering, compressing and denoising.

4.3.1 Optimal Transport Filtering

Filtering is the process of setting a subset of the components of $\boldsymbol{\lambda}$ to 0. Let $\mathcal{N}$ be the set of indices of the components that we want to filter out, optimal transport filtering is performed by solving

$$\min_{\substack{\boldsymbol{\lambda}\in\mathbb{R}^k \\ \lambda_i=0\ \forall i\in\mathcal{N}}} \mathrm{OT}_\gamma(\boldsymbol{x}, D\boldsymbol{\lambda}).$$

Although this problem can be solved by removing all non-relevant columns in D and using the non-regularized algorithm in Rolet et al. [2016], in the case of filtering components of DCT or wavelet transforms we can make use of our orthonormal dictionary case of Section 4.2.1 to get a simpler algorithm. Indeed, the filtering can be rewritten as a regularized projection on an orthonormal basis:

$$\min_{\boldsymbol{\lambda}\in\mathbb{R}^k} \mathrm{OT}_\gamma(\boldsymbol{x}, D\boldsymbol{\lambda}) + F_{\mathcal{N}}(\lambda)$$

where

$$F_{\mathcal{N}}(\boldsymbol{\lambda}) = \begin{cases} 0 & \text{if } \forall i \in \mathcal{N} \text{ s.t. } \lambda_i = 0 \\ \infty & \text{otherwise.} \end{cases}$$

The convex conjugate of $F_{\mathcal{N}}$ is

$$F^*_{\mathcal{N}}(\boldsymbol{h}) = \begin{cases} 0 & \text{if } \forall i \in \overline{\mathcal{N}} \text{ s.t. } \lambda_i = 0 \\ \infty & \text{otherwise.} \end{cases}$$

$F^*_{\mathcal{N}}(\boldsymbol{h})$ is not differentiable, however its proximal operator is easy to compute:

$$\mathrm{prox}_{F^*_{\mathcal{N}}}(\boldsymbol{h})_i = \begin{cases} 0 & \text{if } i \in \overline{\mathcal{N}} \\ h_i & \text{otherwise.} \end{cases}$$

which is simply a regular filter on the complementary components to those described by $\mathcal{N}$. Thus we can solve the dual problem:

$$\min_{\boldsymbol{h}\in\mathbb{R}^n} \mathrm{OT}^*_\gamma(\boldsymbol{x}, \boldsymbol{h}) + F^*_{\mathcal{N}}(-D^\top \boldsymbol{h}),$$

and recover $\lambda^\star$ through the primal-dual relationship: $\boldsymbol{\lambda}^\star = D^\top \nabla \operatorname{OT}_\gamma^*(\boldsymbol{x}, \boldsymbol{h}^\star)$.

We use this method to perform low-pass filters on images, and compare the results with regular low-pass filtering, which can be viewed as a regularized projection *w.r.t.* the Euclidean distance with the same regularizer.

Figure 4.5: Low-pass filtering of the DTC coefficients. Left: original image; Center: Euclidean filtering; Right: Optimal Transport filtering. Top: keeping the $1/16^{th}$ *lowest frequencies. Bottom: keeping the* $1/4^{th}$ *lowest frequencies.*

Taken from Rolet and Seguy [2021]

4.3.1.1 Experimental results

Figure 4.5 shows the result of applying a low-pass filter on a 256×256 image, keeping either the $1/16^{th}$ or $1/4^{th}$ coefficients of its discrete cosine transform (DCT) of lowest frequency. We set the regularization parameter γ of the entropy-regularized optimal transport to 0.1, meaning that an optimal transport pass filter of full bandwidth would correspond to a Gaussian blur of standard deviation 0.1 pixel (see Section 1.1.3), which is almost invisible to the naked eye.

Both filtering methods show the wave-like patterns around edges in the image typical of DCT filtering, however these are more pronounced in the case of the classical, "Euclidean" filtering.

4.3.2 Coefficients Shrinkage and Thresholding

Let $\boldsymbol{x} \in \mathbb{R}_+^n$ be a non-negative vector and $D \in \mathbb{R}^{n \times n}$ be an invertible matrix, typically representing a discrete wavelet or Fourier basis. Coefficient shrinkage of $\boldsymbol{x}$ usually refers to soft-thresholding of the coefficients $\boldsymbol{\lambda} = D^{-1}\boldsymbol{x}$ defined as:

$$\mathcal{S}_\alpha(\boldsymbol{\lambda}) = \operatorname{sign}(\boldsymbol{\lambda}) \odot \max\left\{|\boldsymbol{\lambda}| - \alpha, 0\right\}.$$

In the case where D is orthonormal, $\mathcal{S}_\alpha(D^{-1}\boldsymbol{x})$ is also the solution of the l_1 regularized Euclidean projection on D:

$$\mathcal{S}_\alpha(D^{-1}\boldsymbol{x}) = \operatorname*{argmin}_{\boldsymbol{\lambda}} \|\boldsymbol{x} - D\boldsymbol{\lambda}\|_2^2 + \alpha \|\boldsymbol{\lambda}\|_1 .$$

Hard thresholding on the other hand, is defined as

$$\mathcal{H}_\alpha(\boldsymbol{\lambda})_i = \begin{cases} \lambda_i & \text{if } |\lambda_i| > \alpha \\ 0 & \text{otherwise.} \end{cases}$$

Non-zero coefficients are the same for both hard and soft thresholding. If D is orthonormal, $\mathcal{H}_\alpha(D^{-1}\boldsymbol{x})$ is also the solution of the l_0 regularized euclidean projection on D:

$$\operatorname*{argmin}_{\boldsymbol{\lambda}} \|\boldsymbol{x} - D\boldsymbol{\lambda}\|_2^2 + \alpha \|\boldsymbol{\lambda}\|_0 .$$

4.3.2.1 Optimal transport shrinkage

We mirror this definition of shrinkage to define the optimal transport shrinkage of $\boldsymbol{x}$ as

$$\operatorname*{argmin}_{\boldsymbol{\lambda}} \mathrm{OT}_\gamma(\boldsymbol{x}, D\boldsymbol{\lambda}) + \alpha\|\boldsymbol{\lambda}\|_1$$

for some $\alpha > 0$. This problem can be solved efficiently through one of its dual, *i.e.* Problem (4.6) or Problem (4.7) with $R := \boldsymbol{\lambda} \mapsto \alpha\|\boldsymbol{\lambda}\|_1$. The convex conjugate of R^* of R is an indicator of the l_∞ ball of radius α, and its proximal is a projection on that same ball:

$$R^*(\boldsymbol{h}) = \begin{cases} 0 & \text{if } \|\boldsymbol{h}\|_\infty \leq \alpha \\ \infty & \text{otherwise,} \end{cases}$$
$$\operatorname{prox}_R^*(\boldsymbol{h}) = \operatorname{sign}(\boldsymbol{h}) \odot \min(|\boldsymbol{h}|, \alpha).$$

We recover $\boldsymbol{\lambda}^\star$ from the primal-dual relationships defined in Equation (4.5) or Equation (4.8). Because of machine precision, and of the fact that we can never solve the dual exactly, the coefficients we recover are not sparse, but a lot of them are very close to 0. We can however recover the sparsity pattern of $\boldsymbol{\lambda}^\star$ with the first order conditions for Problem (4.4) with respect to $\boldsymbol{\lambda}$. Indeed, these first order conditions are $-D^\top \boldsymbol{h}^\star \in \nabla R(\boldsymbol{\lambda}^\star)$, *i.e.*:

$$-D^\top \boldsymbol{h}^\star \in \left\{ \boldsymbol{a} \in \mathbb{R}^k \middle| \begin{array}{ll} -\alpha \leq a_i \leq \alpha & \text{if } \lambda_i^\star = 0 \\ a_i = \operatorname{sign}(\lambda_i^\star)\alpha & \text{otherwise} \end{array} \right\}.$$

Accordingly, we can set $\lambda_i^\star$ to 0 for all i such that $\left|(D^\top \boldsymbol{h}^\star)_i\right| < \alpha$.

Since $\|\cdot\|_1$ is convex, has full domain and D is full rank, the optimal transport coefficient shrinkage problem has a unique solution according to Theorem 2.2.3 and Theorem 2.2.2.

4.3.2.2 Optimal transport hard thresholding

Since the ℓ_0 norm is not a convex function, we do not have a method to solve the ℓ_0-regularized optimal transport projection. We define hard thresholding of the coefficients by analogy with the Euclidean case, based on the fact that the sparsity pattern for hard and soft thresholding is the same. In other words, hard thresholding corresponds to a pass filters on the non-zero coefficients of the soft-thresholding operator. In terms of optimization problems, this means that if $\boldsymbol{\lambda}^\star$ is the solution of

$$\min_{\boldsymbol{\lambda}} \|\boldsymbol{x} - D\boldsymbol{\lambda}\|_2^2 + \alpha \|\boldsymbol{\lambda}\|_1 ,$$

and denoting $\mathcal{N} = \{i \text{ s.t. } \lambda_i^\star = 0\}$, then

$$\mathcal{H}_\alpha(D^{-1}\boldsymbol{x}) = \operatorname*{argmin}_{\boldsymbol{\lambda} \atop \forall i \in \mathcal{N}, \lambda_i = 0} \|\boldsymbol{x} - D\boldsymbol{\lambda}\|_2 .$$

Similarly, for $\alpha > 0$, we define the optimal transport hard thresholding as the optimal transport pass filter on the non-zero coefficients of

$$\operatorname*{argmin}_{\boldsymbol{\lambda}} \mathrm{OT}_\gamma(\boldsymbol{x}, D\boldsymbol{\lambda}) + \alpha \|\boldsymbol{\lambda}\|_1 .$$

We compute the pass filter using the method defined in Section 4.3.1. Hard thresholding allows us to get better results when the level of noise on the signal is low. We analyze in the remainder of this section the effect of using optimal transport instead

of the usual implicit Euclidean distance when performing hard and soft thresholding for either compression or denoising.

4.3.2.3 Compressing

Hard thresholding can be used to perform compressing, where the goal is to represent an image with as few coefficients as possible, while retaining good image quality.

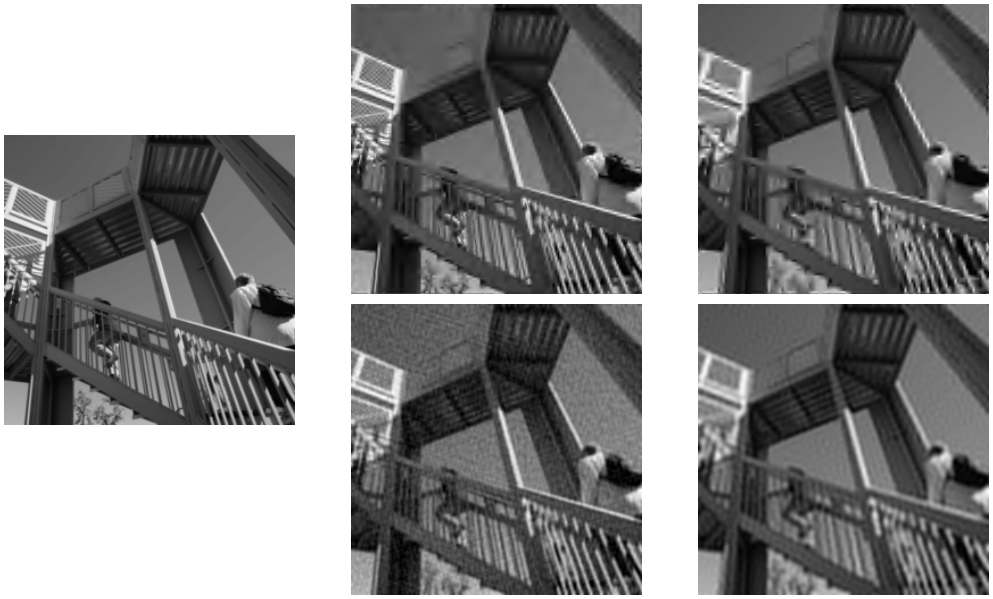
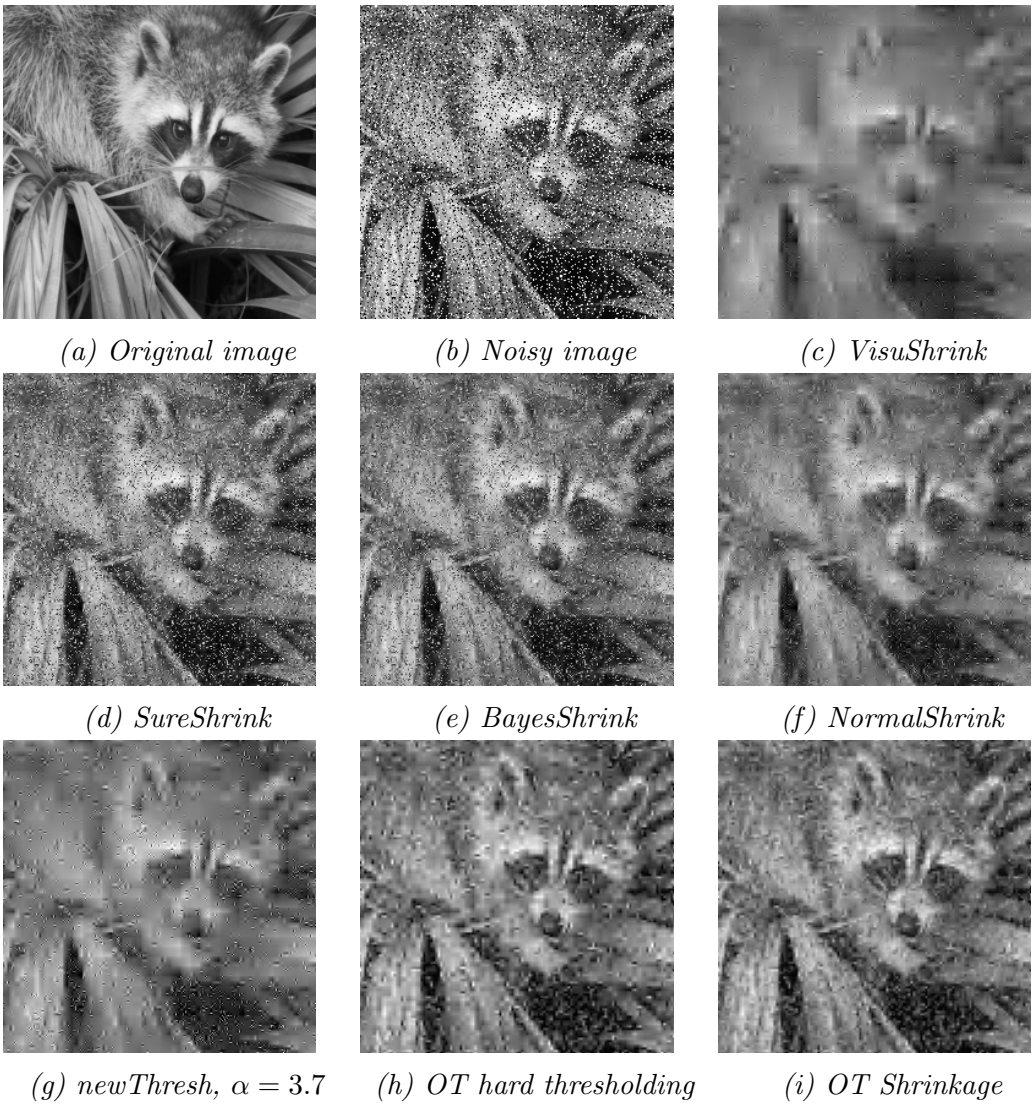


Figure 4.6: Compression with Euclidean or Optimal Transport hard thresholding with biorthogonal spline wavelets of order 2 and dual order 4. Sparsity is set to 95%. *Left: original image; Center: Euclidean hard thresholding; Right: Optimal Transport hard thresholding. Top: biorthogonal spline wavelets decomposition. Bottom: DCT decomposition.*

Taken from Rolet and Seguy [2021]

Figure 4.6 shows the effect of optimal transport and Euclidean hard thresholding on the coefficients of either biorthogonal spline wavelets [Cohen et al., 1992] or DCT decomposition, where 5% of the coefficients are kept. Again we observe higher levels of artifacts with Euclidean thresholding.

For the biorthogonal spline wavelet decomposition, these artifact are especially visible in low contrast areas such as the background. As a result the fence-like structure at the top of the image has almost disappeared with Euclidean thresholding, but is still visible with optimal transport.



(a) Original image *(b) Noisy image* *(c) VisuShrink*

(d) SureShrink *(e) BayesShrink* *(f) NormalShrink*

(g) newThresh, $\alpha = 3.7$ *(h) OT hard thresholding* *(i) OT Shrinkage*

Figure 4.7: Denoising of salt-and-pepper noise of level $\sigma = 10\%$ with Daubechies wavelets of order 2.

Taken from Rolet and Seguy [2021]

4.3.2.4 Denoising

We now examine how optimal transport thresholding compares to other wavelet coefficient shrinkage methods for image denoising. Many of the standard wavelet methods for image denoising perform either a soft or hard thresholding on the coefficients, which makes them inherently Euclidean sparse projection methods. Their main difference is on how to select the threshold. We compare our methods to visuShrink [Donoho and Johnstone, 1994], which selects one global threshold for the image, and *adaptive* methods which select a threshold for each wavelet decomposition level: sureShrink [Donoho and Johnstone, 1995], bayesShrink [Chang et al., 2000] and normalShrink [Kaur et al.,

Noise	σ	normalShrink	sureShrink	bayesShrink	visuShrink	newThresh	OT hard	OT soft
Salt & pepper	5%	0.384	0.353	0.347	0.318	0.328	0.520	**0.545**
	10%	0.333	0.257	0.277	0.191	0.238	0.403	**0.441**
	15%	0.259	0.229	0.278	0.114	0.173	0.319	**0.350**
Gaussian	0.2	0.641	**0.706**	0.704	0.403	0.533	0.666	0.701
	0.3	0.532	0.604	0.600	0.312	0.424	0.581	**0.613**
	0.4	0.459	0.525	0.523	0.256	0.357	0.505	**0.537**

(a) SSIM

Noise	σ	normalShrink	sureShrink	bayesShrink	visuShrink	newThresh	OT hard	OT soft
Salt & pepper	5%	18.826	16.965	16.752	20.140	20.212	22.086	**22.911**
	10%	19.751	16.256	16.996	18.535	19.084	20.631	**21.077**
	15%	19.157	17.310	18.610	17.310	18.069	19.343	**19.781**
Gaussian	0.2	25.123	**25.903**	25.849	22.177	23.775	24.844	25.698
	0.3	23.594	24.163	24.172	20.889	22.288	23.308	**24.190**
	0.4	22.644	23.098	23.097	20.054	21.307	22.328	**23.171**

(b) pSNR

Table 4.2: Denoising scores for different wavelet thresholding methods.
Taken from Rolet and Seguy [2021]

2002]. We also compare our method with Dehda and Melkemi [2017], a thresholding method which uses a smooth thresholding function which can be seen as a trade-off between soft and hard thresholding. We call this method "newThresh" in the experiment.

With optimal transport, adaptive thresholding could be achieved by using either a weighted ℓ_1-norm or a block-sparse regularizer. However we found that this doesn't improve significantly upon simple ℓ_1-norm regularization and we only report the results of "global" thresholding here for simplicity.

For this experiment, we corrupt a 256×256 image with either a Gaussian or a salt-and-pepper noise with several noise levels σ. In the case of the Gaussian noise, σ is the variance and is taken to be 0.2, 0.3 or 0.4 times the mean intensity of the image. For the salt-and-pepper noise, $\sigma \in \{5\%, 10\%, 15\%\}$ is the proportion of pixels that are set to 0, and the same number are set to the maximum intensity (255). We perform coefficient shrinkage on the coefficients of Daubechies wavelets of order 2 [Daubechies, 1992] of the noisy image.

Figure 4.7 shows the images produced by the different thresholding methods for a salt-and-pepper noise. Similarly to the low-pass filtering and compression experiments, optimal transport based thresholding shows less wavelet artifacts. Hard thresholding appears to produce images that are sharper, but also more corrupted.

Table 4.2 reports the pSNR and SSIM [Wang et al., 2004] scores for each method and each noise. Our methods and newThresh each have a free parameter. For newThresh, we report the best score among 15 candidate shape parameters α in a log-scale interval from $1e-4$ to $1e4$. For optimal transport methods, we report

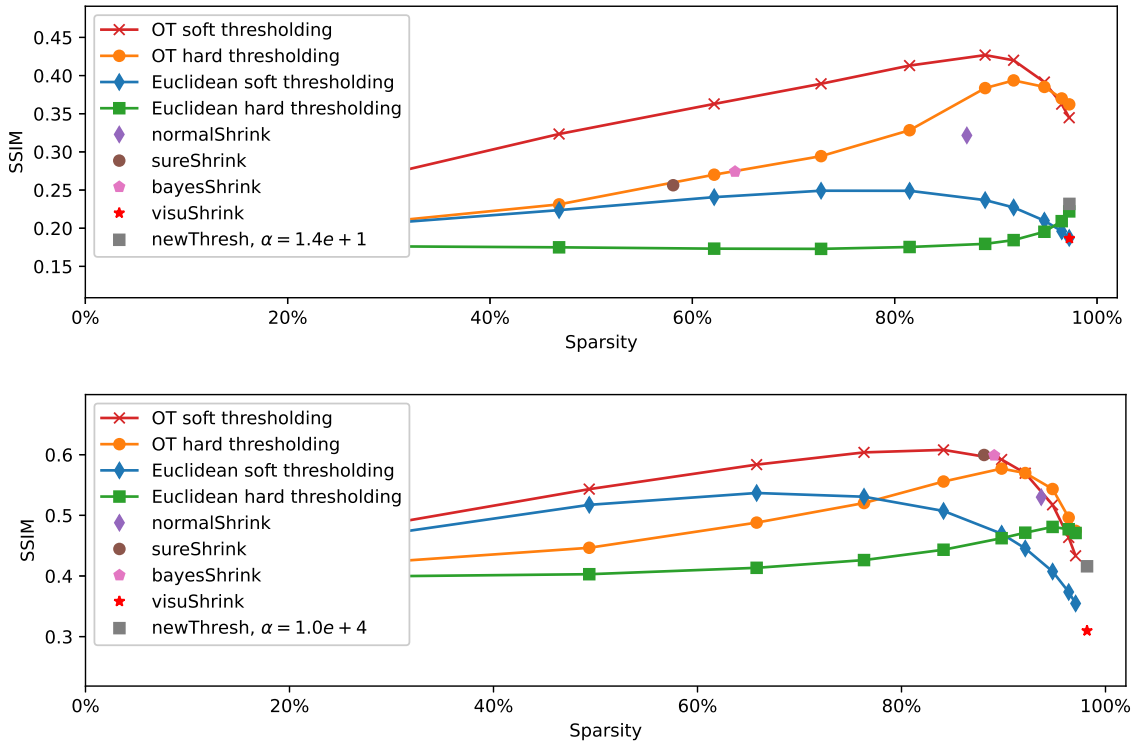


Figure 4.8: Comparison of SSIM values as a function of sparsity. Competitors produce a single image, and are represented as a point. Top: salt-and-pepper noise with $\sigma = 10\%$. Bottom: Gaussian noise with of $\sigma = 0.3$.

the best score among 10 candidate regularization parameters α in the interval from 0.25 to 6. Optimal transport shrinkage improves upon all other methods for both denoising scores, except for a small intensity Gaussian noise, for which sureShrink and baryesShrink perform slightly better. In particular, optimal transport brings significant improvement for a salt-and-pepper noise.

We now investigate how sparsity of the coefficients of the denoised image impacts the denoising score. Figure 4.8 plots the SSIM score with respect to sparsity for all methods. We first observe that if we only compare optimal transport thresholding to its Euclidean counterpart, optimal transport achieves a higher SSIM across all sparsities for both noises. Furthermore, we see that for the salt-and-pepper noise, optimal transport shrinkage achieves better results than other methods, even without selecting the sparsity carefully: all points of the blue curve above 40% sparsity have a better SSIM score than competitors, excluding optimal transport hard thresholding. With a Gaussian noise, which sureShrink and bayesShrink are optimized for, we can see that optimal transport shrinkage is still competitive, and achieves similar results as sureShrink and bayesShrink for the same sparsity.

With optimal transport shrinkage, the denoising output for an image is not defined uniquely, but rather is a function of sparsity (or of the regularization parameter). This is a good thing in a supervised setting, in which a user can modify the regularization parameter α until they are satisfied with the output. However it also means that in an unsupervised, or automated setting, we need a way to select the sparsity level based on the image. Based on Figure 4.8, a simple solution would be to pick any of the sparsities obtained by the outputs of sureShrink, bayesShrink and normalShrink, or their average sparsity.

4.4 Chapter Conclusion

In this chapter we showed how to perform a regularized projection of a signal onto a fixed dictionary with respect to the optimal transport distance. We showed that while the general saddle point method is slow, we can get faster algorithms when either the regularizer's convex conjugate is differentiable or the dictionary is invertible. This last case allows us to perform sparse signal decomposition in various domains, including the discrete Fourier domain or wavelets. In practice, our results show that this optimal transport coefficients shrinkage yields less artifacts than coefficient shrinkage, where the signal is projected with respect to the Euclidean distance. For image denoising, it also outperforms other widely used wavelet based methods such as BayesShrink and SureShrink, especially for images corrupted with non-Gaussian noise.

The remainder of this monograph will show how to adapt and extend our dual methods to perform optimal transport dictionary learning and NMF, and showcase applications in natural language processing and sound processing.

Chapter 5

Conclusion

In this thesis, we have shown how to use optimal transport as the loss for the dictionary learning and non-negative matrix factorization problems. Optimal transport had been used before to address some specific tasks in data processing or pattern recognition, mainly as the metric in metric-based information retrieval algorithms. However the use of optimal transport as a loss for training machine learning models on large datasets was still out of reach due to the high complexity of computing optimal transport and its gradients. Thanks to our dual methods, we were among the firsts to learn models trained with an optimal transport loss, especially on large datasets. We showed the benefit of using optimal transport as opposed to other losses, not only in terms of performance, but also because it allows to learn models which can be applied to data in a different domain.

5.1 Contributions

Building on previous works in optimal transport regularization, we derived a method which allow to solve optimal transport dictionary learning and NMF efficiently in Chapter 2. Mirroring the celebrated alternative least square procedure, we solve optimal transport dictionary learning by alternating an optimal transport regularized projection step and a dictionary update step until convergence. We derived duals for both of these steps which can be solved efficiently by making use of the simple form of the convex conjugate of regularized optimal transport. We showed how optimal transport NMF improves upon its Euclidean and KL counterparts when applied to topic modeling.

We proposed to leverage the versatility of optimal transport to perform what we called *cross-domain* learning, where a model can be learned on, or applied to, data in a different domain. This can be used in topic modeling to learn a single language representation of a dataset in multiple languages for example, and we are not aware

of other methods which could address this task efficiently prior to our work.

After our first results were published, a musical note transcription method based on learning only the coefficients, with a dictionary made of Diracs, was proposed with encouraging results. The method was tractable thanks to the simple form of the dictionary and relied on a ground cost matrix specifically designed for musical instruments to achieve good performance. We decided to further investigate this area by adapting our optimal transport NMF method to sound processing, so that it can be applied to sound data. We showed in Chapter 3 how to design a cost matrix adapted to the processing of sound data, optionally in a cross-domain setting. Applying the principle of the Weiner filter to optimal transport concepts, we showed how to reconstruct sounds from STFT spectrograms using optimal transport plans. A particular interest of optimal transport in source separation tasks is that it seems well adapted to learn *universal* models. That is, dictionaries which can be applied to a whole class of sources (*e.g.* any voice) instead of a specific source. We showed improvements of perceptual scores when compared to Euclidean, KL and IS NMF, both on a voice separation task and on a voice denoising task.

Lastly, we focused in Chapter 4 on one step of the dictionary learning procedure: optimal transport regularized projection. Regularized projection with the Euclidean metric is a common procedure in signal processing, with applications to denoising, or compression in the sparse case. We defined optimal transport regularized projections, and in particular we gave fast methods which can be applied to most regularizers in the case of an invertible dictionary, which we used to solve the optimal transport wavelet shrinkage problem. By applying our methods to clear and noisy image data, we showed that they can improve the performance of their Euclidean counter-part, and that they tend to reduce artifact production.

5.2 Future work

We have shown the interest of using optimal transport as a loss for dictionary learning and NMF, and we are now very interested in seeing it applied to more complicated models. In the Euclidean case for example, adding a sparsity-inducing term has been shown to greatly improve performance on various tasks ranging from image denoising to blind source separation. In order to apply the same idea with optimal transport, the need arises for fast methods to compute sparse coefficients. We were able to derive such methods in the case of an orthonormal or an invertible dictionary, and we believe that similar methods can be developed for the general case. Sparse NMF on another hand is a simpler problem since the ℓ_1-norm in this case is just a linear term, however this term is redundant with the optimal transport constraint that both inputs have the same total weight. We believe that using recent advances in *unbalanced* optimal transport could help, in conjunction with our work, to define fast dual methods for sparse optimal transport NMF.

We were able to apply our NMF method to text data thanks to word embeddings, which leverage neural network architectures to learn feature representations for each word in their training dataset. Such representations are not always available however, and in our sound processing experiments we relied on expert knowledge to define a cost matrix adapted to sound data. More generally, the choice of an optimal cost matrix for a specific type of data remains an open problem. Some works on learning the cost matrix already exist, and we believe that any progress in this area would directly any optimal transport-based method.

Thanks to our dual methods, we showed that optimal transport dictionary and NMF problems can in fact be tractable. Further improvements on the computational side could allow to learn models on bigger or even infinite datasets. For instance, stochastic or batch methods, in which only a subset of the columns of the input matrix are treated at a time, could help both to reduce the complexity of each step of the procedure, but also to avoid getting stuck early in a bad local minimum.

List of Publications

As First Author

- Antoine Rolet, Marco Cuturi, and Gabriel Peyré. Fast dictionary learning with a smoothed wasserstein loss. In *Artificial Intelligence and Statistics*, pages 630–638, 2016

- Antoine Rolet, Vivien Seguy, Mathieu Blondel, and Hiroshi Sawada. Blind source separation with optimal transport non-negative matrix factorization. *EURASIP Journal on Advances in Signal Processing*, 2018(1):53, 2018

- Antoine Rolet and Vivien Seguy. Fast optimal transport regularized projection and application to coefficient shrinkage and filtering. *The Visual Computer*, 2021

Others

- Mathieu Blondel, Vivien Seguy, and Antoine Rolet. Smooth and sparse optimal transport. In *International Conference on Artificial Intelligence and Statistics*, pages 880–889, 2018

- Vivien Seguy, Bharath Bhushan Damodaran, Remi Flamary, Nicolas Courty, Antoine Rolet, and Mathieu Blondel. Large-scale optimal transport and mapping estimation. In *International Conference on Learning Representations (ICLR)*, 2018

Bibliography

Martin Arjovsky, Soumith Chintala, and Léon Bottou. Wasserstein generative adversarial networks. volume 70 of *Proceedings of Machine Learning Research*, pages 214–223. PMLR, 06–11 Aug 2017.

Amir Beck and Marc Teboulle. A fast iterative shrinkage-thresholding algorithm for linear inverse problems. *SIAM Journal on Imaging Sciences*, 2(1):183–202, 2009.

Jean-David Benamou, Guillaume Carlier, Marco Cuturi, Luca Nenna, and Gabriel Peyré. Iterative bregman projections for regularized transportation problems. *SIAM Journal on Scientific Computing*, 37(2):A1111–A1138, 2015.

Y. Bengio, R. Ducharme, P. Vincent, and C. Janvin. A neural probabilistic language model. *The Journal of Machine Learning Research*, 3:1137–1155, 2003.

Michael W Berry, Susan T Dumais, and Gavin W O'Brien. Using linear algebra for intelligent information retrieval. *SIAM review*, 37(4):573–595, 1995.

Michael W Berry, Murray Browne, Amy N Langville, V Paul Pauca, and Robert J Plemmons. Algorithms and applications for approximate nonnegative matrix factorization. *Computational statistics & data analysis*, 52(1):155–173, 2007.

Mathieu Blondel, Vivien Seguy, and Antoine Rolet. Smooth and sparse optimal transport. In *International Conference on Artificial Intelligence and Statistics*, pages 880–889, 2018.

Ben Cao, Qinzhen Xu, Wen Yan, and Luxi Yang. A stochastic alternating minimization approach for large scale low rank matrix factorization. In *2016 8th International Conference on Wireless Communications & Signal Processing (WCSP)*, pages 1–4. IEEE, 2016.

A. Cardoso-Cachopo. Improving Methods for Single-label Text Categorization. PdD Thesis, Instituto Superior Tecnico, Universidade Tecnica de Lisboa, 2007.

Elsa Cazelles, Vivien Seguy, Jérémie Bigot, Marco Cuturi, and Nicolas Papadakis. Geodesic pca versus log-pca of histograms in the wasserstein space. *SIAM Journal on Scientific Computing*, 40(2):B429–B456, 2018.

S Grace Chang, Bin Yu, and Martin Vetterli. Adaptive wavelet thresholding for image denoising and compression. *IEEE transactions on image processing*, 9(9): 1532–1546, 2000.

Lénaïc Chizat, Gabriel Peyré, Bernhard Schmitzer, and François-Xavier Vialard. Scaling algorithms for unbalanced optimal transport problems. *Mathematics of Computation*, 87:2563–2609, 2018.

Albert Cohen, Ingrid Daubechies, and J-C Feauveau. Biorthogonal bases of compactly supported wavelets. *Communications on pure and applied mathematics*, 45(5):485–560, 1992.

Laurent Condat. A primal–dual splitting method for convex optimization involving lipschitzian, proximable and linear composite terms. *Journal of Optimization Theory and Applications*, 158(2):460–479, 2013.

M. Cuturi. Sinkhorn distances: Lightspeed computation of optimal transport. In *Advances in Neural Information Processing Systems*, pages 2292–2300, 2013.

M. Cuturi and D. Avis. Ground metric learning. *The Journal of Machine Learning Research*, 15(1):533–564, 2014.

M. Cuturi and A. Doucet. Fast computation of wasserstein barycenters. In *Proceedings of the 31st International Conference on Machine Learning (ICML-14)*, 2014.

Marco Cuturi and Gabriel Peyré. A smoothed dual approach for variational wasserstein problems. *SIAM Journal on Imaging Sciences*, 9(1):320–343, 2016.

Ingrid Daubechies. *Ten lectures on wavelets*, volume 61. Siam, 1992.

Bachir Dehda and Khaled Melkemi. Image denoising using new wavelet thresholding function. *Journal of Applied Mathematics and Computational Mechanics*, 16(2), 2017.

C. Ding, T. Li, and W. Peng. On the equivalence between non-negative matrix factorization and probabilistic latent semantic indexing. *Computational Statistics & Data Analysis*, 52(8):3913–3927, 2008.

David L Donoho. De-noising by soft-thresholding. *IEEE transactions on information theory*, 41(3):613–627, 1995.

David L Donoho and Iain M Johnstone. Ideal spatial adaptation by wavelet shrinkage. *biometrika*, 81(3):425–455, 1994.

David L Donoho and Iain M Johnstone. Adapting to unknown smoothness via wavelet shrinkage. *Journal of the American Statistical Association*, 90(432):1200–1224, 1995.

Carl Eckart and Gale Young. The approximation of one matrix by another of lower rank. *Psychometrika*, 1(3):211–218, 1936.

R'emi Flamary and Nicolas Courty. Pot python optimal transport library, 2017. URL `https://pythonot.github.io/`.

Rémi Flamary, Cédric Févotte, Nicolas Courty, and Valentin Emiya. Optimal spectral transportation with application to music transcription. In *Advances in Neural Information Processing Systems*, pages 703–711, 2016.

Charlie Frogner, Chiyuan Zhang, Hossein Mobahi, Mauricio Araya, and Tomaso A Poggio. Learning with a wasserstein loss. In *Advances in Neural Information Processing Systems*, pages 2053–2061, 2015.

Alexandre Gramfort, Gabriel Peyré, and Marco Cuturi. Fast optimal transport averaging of neuroimaging data. In *International Conference on Information Processing in Medical Imaging*, pages 261–272. Springer, 2015.

D. Greene and P. Cunningham. Practical solutions to the problem of diagonal dominance in kernel document clustering. In *Proceedings of the 23rd International Conference on Machine learning (ICML-06)*, pages 377–384. ACM Press, 2006.

T. Hofmann. Probabilistic latent semantic indexing. In *Proceedings of the 22nd annual international ACM SIGIR conference on Research and development in information retrieval*, pages 50–57. ACM, 1999.

Gao Huang, Chuan Guo, Matt J Kusner, Yu Sun, Fei Sha, and Kilian Q Weinberger. Supervised word mover's distance. *Advances in neural information processing systems*, 29:4862–4870, 2016.

Rainer Huber and Birger Kollmeier. Pemo-q—a new method for objective audio quality assessment using a model of auditory perception. *IEEE Transactions on audio, speech, and language processing*, 14(6):1902–1911, 2006.

Takaaki Ishii, Hiroki Komiyama, Takahiro Shinozaki, Yasuo Horiuchi, and Shingo Kuroiwa. Reverberant speech recognition based on denoising autoencoder. In *Interspeech*, pages 3512–3516, 2013.

Lakhwinder Kaur, Savita Gupta, and RC Chauhan. Image denoising using wavelet thresholding. In *ICVGIP*, volume 2, pages 16–18, 2002.

Hyunsoo Kim and Haesun Park. Nonnegative matrix factorization based on alternating nonnegativity constrained least squares and active set method. *SIAM journal on matrix analysis and applications*, 30(2):713–730, 2008.

Yehuda Koren, Robert Bell, and Chris Volinsky. Matrix factorization techniques for recommender systems. *Computer*, 42(8):30–37, 2009.

Matt Kusner, Yu Sun, Nicholas Kolkin, and Kilian Weinberger. From word embeddings to document distances. In *International conference on machine learning*, pages 957–966, 2015.

S. Lauly, H. Larochelle, M. Khapra, B. Ravindran, V.C. Raykar, and A. Saha. An autoencoder approach to learning bilingual word representations. In *Advances in Neural Information Processing Systems*, pages 1853–1861, 2014.

Daniel D Lee and H Sebastian Seung. Algorithms for non-negative matrix factorization. In *Advances in neural information processing systems*, pages 556–562, 2001.

D.D. Lee and H.S. Seung. Learning the parts of objects by non-negative matrix factorization. *Nature*, 401(6755):788–791, 1999.

Yuanqing Li, Andrzej Cichocki, and Shun-ichi Amari. Analysis of sparse representation and blind source separation. *Neural computation*, 16(6):1193–1234, 2004.

Dirk A Lorenz and Thomas Pock. An inertial forward-backward algorithm for monotone inclusions. *Journal of Mathematical Imaging and Vision*, 51(2):311–325, 2015.

Julien Mairal, Francis Bach, Jean Ponce, and Guillermo Sapiro. Online dictionary learning for sparse coding. In *Proceedings of the 26th annual international conference on machine learning*, pages 689–696, 2009.

Tomas Mikolov, Kai Chen, Greg Corrado, and Jeffrey Dean. Efficient estimation of word representations in vector space. In *ICLR workshop*, 2013.

Gaspard Monge. Mémoire sur la théorie des déblais et des remblais. *Histoire de l'Académie Royale des Sciences de Paris*, 1781.

Grégoire Montavon, Klaus-Robert Müller, and Marco Cuturi. Wasserstein training of restricted boltzmann machines. In *Advances in Neural Information Processing Systems*, volume 29, pages 3718–3726, 2016.

Yu Nesterov. A method of solving a convex programming problem with convergence rate $o(1/k^2)$. In *Soviet Mathematics Doklady*, volume 27, pages 372—376, 1983.

James B Orlin. A faster strongly polynomial minimum cost flow algorithm. *Operations research*, 41(2):338–350, 1993.

James B Orlin, Serge A Plotkin, and Éva Tardos. Polynomial dual network simplex algorithms. *Mathematical programming*, 60(1-3):255–276, 1993.

P. Paatero and U. Tapper. Positive matrix factorization: A non-negative factor model with optimal utilization of error estimates of data values. *Environmetrics*, 5(2):111–126, 1994.

J. Pennington, R. Socher, and C.D. Manning. Glove: Global vectors for word representation. *Proceedings of the Empiricial Methods in Natural Language Processing (EMNLP 2014)*, 12:1532–1543, 2014.

Julien Rabin and Nicolas Papadakis. Convex color image segmentation with optimal transport distances. In *International Conference on Scale Space and Variational Methods in Computer Vision*, pages 256–269. Springer, 2015.

Ievgen Redko, Nicolas Courty, Rémi Flamary, and Devis Tuia. Optimal transport for multi-source domain adaptation under target shift. In *The 22nd International Conference on Artificial Intelligence and Statistics*, pages 849–858, 2019.

Antoine Rolet and Vivien Seguy. Fast optimal transport regularized projection and application to coefficient shrinkage and filtering. *The Visual Computer*, 2021.

Antoine Rolet, Marco Cuturi, and Gabriel Peyré. Fast dictionary learning with a smoothed wasserstein loss. In *Artificial Intelligence and Statistics*, pages 630–638, 2016.

Antoine Rolet, Vivien Seguy, Mathieu Blondel, and Hiroshi Sawada. Blind source separation with optimal transport non-negative matrix factorization. *EURASIP Journal on Advances in Signal Processing*, 2018(1):53, 2018.

Yossi Rubner, Carlo Tomasi, and Leonidas J Guibas. A metric for distributions with applications to image databases. In *Computer Vision, 1998. Sixth International Conference on*, pages 59–66, 1998.

F.S. Samaria and A.C. Harter. Parameterisation of a stochastic model for human face identification. In *Applications of Computer Vision, 1994., Proceedings of the Second IEEE Workshop on*, pages 138–142. IEEE, 1994.

R. Sandler and M. Lindenbaum. Nonnegative matrix factorization with earth mover's distance metric. In *Computer Vision and Pattern Recognition, 2009. CVPR 2009. IEEE Conference on*, pages 1873–1880. IEEE, 2009.

Hiroshi Sawada, Hirokazu Kameoka, Shoko Araki, and Naonori Ueda. Multichannel extensions of non-negative matrix factorization with complex-valued data. *IEEE Transactions on Audio, Speech, and Language Processing*, 21(5):971–982, 2013.

Mikkel N Schmidt and Rasmus Kongsgaard Olsson. Single-channel speech separation using sparse non-negative matrix factorization. In *Spoken Language Proceesing, ISCA International Conference on (INTERSPEECH)*, 2006.

M.N. Schmidt, J. Larsen, and F.T. Hsiao. Wind noise reduction using non-negative sparse coding. In *Machine Learning for Signal Processing, 2007 IEEE Workshop on*, pages 431–436. IEEE, 2007.

Vivien Seguy and Marco Cuturi. Principal geodesic analysis for probability measures under the optimal transport metric. In *Advances in Neural Information Processing Systems*, pages 3312–3320, 2015.

Vivien Seguy, Bharath Bhushan Damodaran, Remi Flamary, Nicolas Courty, Antoine Rolet, and Mathieu Blondel. Large-scale optimal transport and mapping estimation. In *International Conference on Learning Representations (ICLR)*, 2018.

S. Shirdhonkar and D.W. Jacobs. Approximate earth mover's distance in linear time. In *Computer Vision and Pattern Recognition, 2008. CVPR 2008. IEEE Conference on*, pages 1–8. IEEE, 2008.

Justin Solomon, Fernando de Goes, Gabriel Peyré, Marco Cuturi, Adrian Butscher, Andy Nguyen, Tao Du, and Leonidas Guibas. Convolutional wasserstein distances: Efficient optimal transportation on geometric domains. *ACM Transactions on Graphics*, 34(4), July 2015. ISSN 0730-0301.

Dennis L Sun and Gautham J Mysore. Universal speech models for speaker independent single channel source separation. In *Acoustics, Speech and Signal Processing (ICASSP), 2013 IEEE International Conference on*, pages 141–145. IEEE, 2013.

Yoshio Takane, Forrest W Young, and Jan De Leeuw. Nonmetric individual differences multidimensional scaling: An alternating least squares method with optimal scaling features. *Psychometrika*, 42(1):7–67, 1977.

Guillaume Tartavel, Gabriel Peyré, and Yann Gousseau. Wasserstein loss for image synthesis and restoration. *SIAM Journal on Imaging Sciences*, 9(4):1726–1755, 2016.

Nobuaki Tomizawa. On some techniques useful for solution of transportation network problems. *Networks*, 1(2):173–194, 1971.

J.A. Tropp. An alternating minimization algorithm for non-negative matrix approximation, 2003.

Cédric Villani. *Topics in optimal transportation*. Number 58. American Mathematical Soc., 2003.

Emmanuel Vincent, Rémi Gribonval, and Cédric Févotte. Performance measurement in blind audio source separation. *IEEE transactions on audio, speech, and language processing*, 14(4):1462–1469, 2006.

Zhou Wang, Alan C Bovik, Hamid R Sheikh, and Eero P Simoncelli. Image quality assessment: from error visibility to structural similarity. *IEEE transactions on image processing*, 13(4):600–612, 2004.

G. Zen, E. Ricci, and N. Sebe. Simultaneous ground metric learning and matrix factorization with earth mover's distance. In *Pattern Recognition (ICPR), 2014 22nd International Conference on*, pages 3690–3695. IEEE, 2014.

S. Zhang, W. Wang, J. Ford, and F. Makedon. Learning from incomplete ratings using non-negative matrix factorization. In *SDM*, volume 6, pages 548–552. SIAM, 2006.

W.Y. Zou, R. Socher, D.M. Cer, and C.D. Manning. Bilingual word embeddings for phrase-based machine translation. pages 1393–1398, 2013.