Study of the aggregation procedure : patch fusion and generalized Wasserstein barycenters

Alexandre Saint-Dizier

UNIVERSITÉ DE PARIS

Laboratoire MAP5

THÈSE DE DOCTORAT DE MATHÉMATIQUES APPLIQUÉES
ED 386

Study of the aggregation procedure : patch fusion and generalized Wasserstein barycenters

---

Présentée par:
Alexandre SAINT-DIZIER

Dirigée par:
Mme. Julie DELON
M. Charles BOUVEYRON

Présentée et soutenue publiquement le 17 décembre 2020
devant un jury composé de :

Nicolas COURTY,	*Professeur des Universités*,	Université Bretagne Sud,	**rapporteur**
Nicolas PAPADAKIS,	*Chargé de recherche*,	Université de Bordeaux,	**rapporteur**
Erwan LE PENNEC,	*Professeur des Uninversités*,	Ecole Polytechnique,	**président du jury**
Agnès DESOLNEUX,	*Directeure de recherche*,	ENS Paris Saclay,	**examinatrice**
Arthur LECLAIRE,	*Maître de conférence*,	Université de Bordeaux,	**examinateur**
Julie DELON,	*Professeure des Universités*,	Université de Paris,	**directrice de thèse**
Charles BOUVEYRON,	*Professeur des Universités*,	Université Côte d'Azur,	**directeur de thèse**

Université de Paris

LABORATOIRE MAP5

Abstract

Title: Study of the aggregation procedure : patch fusion and generalized Wasserstein barycenters

Abstract: This thesis is focused on patch-based methods, a particular type of image processing algorithm. These methods include a step called aggregation, which consists in reconstructing an image from a set of overlapping patches and statistical models on these patches. The aggregation step is formalized here as a fusion operation on distributions living on different subspaces but not disjoint. We introduce first a new fusion method based on probabilistic considerations, directly applied to the aggregation problem. It turns out that this operation can also be formulated in a more general setup, like a generalization of a barycenter problem between distributions. This lead us to study this new problem from an optimal transport theory perspective.

Keywords: Image restoration, Denoising, Optimal Transport, Wasserstein barycenter, Baysesian modeling, Patch aggregation.

Titre: Etude du problème d'agrégation : fusion de patchs et barycentres de Wasserstein généralisés

Résumé: Cette thèse porte sur une classe particulière d'algorithmes de traitement d'images : les méthodes par patchs. Ces méthodes nécessitent une étape appelée agrégation, qui consiste à reformer une image à partir d'un ensemble de patchs, et de modèles statistiques sur ces mêmes patchs. L'étape d'agrégation est formalisée ici comme une opération de fusion de distributions vivant sur des espaces différents mais non-disjoints. On propose d'abord une méthode de fusion basée sur des considérations probabilistes, directement applicable au problème d'agrégation. Il se trouve que cette opération peut aussi se formuler dans un contexte plus général comme une généralisation d'un problème de barycentre entre distributions, ce qui amène à l'étudier dans un deuxième temps du point de vue du transport optimal.

Mots clefs : Restauration d'image, Débruitage, Transport optimal, Barycenter de Wasserstein, Modèles Bayesiens, Aggrégation de patchs.

Remerciements

Quand j'étais petit, après avoir passé la phase du fantasme de vouloir devenir astronaute (quoique j'eus toujours une préférence marquée pour les astronaumes, que je trouvais bien plus méritant) ou champion de sport, j'ai fini par m'accorder sur un projet qui me semblait bien plus raisonnable, accessible et surtout partageable. Car, quand on est un enfant, il faut bien savoir affirmer avec conviction ce que l'on souhaite faire quand on sera devenu une autre (grande ?) personne sous peine de passer pour un idiot. Ainsi, quand les grandes personnes me posaient leur fameuse question du projet professionnel, je répondais avec un peu de nonchalance, ayant conscience du réalisme décévant de mon ambition, que je voulais devenir *chercheur*. Pour que la question ne semble pas totalement superficielle, beaucoup me demandait, feignant plus ou moins d'insatisfaction, de préciser : chercheur en quoi ? Mon ton se faisait alors subitement moins affirmatif, mais la réponse finissait par se faire entendre : Chercheur en mathématique, évidemment, en quoi d'autre ?... Car, à l'époque, cela allait de soi, le chercheur cherche la vérité, et il la trouve dans les mathématiques.

Il se trouve, comme le lecteur en a surement déjà été averti, qu'après mes études de mathématiques, je me suis trouvé en position de faire une thèse, et de devenir, au yeux de tous, un chercheur véritable. Faut-il voir là la conséquence déterministe de mes aspirations de jeunesse ou l'implacable précision de l'intuition juvénile ? Toujours est-il que cette interrogation ne m'a pas empêché de me jeter tête baissé dans ce projet fou et pourtant si évident dans mon esprit. Le sentiment d'invulnérabilité que m'avait prodigué ma réussite en prépa et l'euphorie qui en découlait commençaient alors à s'affaiblir et après trois riches et fastes années d'Ecole, j'en devenais un peu nostalgique. Je désirais retrouver cette toute-puissance passée et l'assurance qui l'accompagnait. Ce doctorat semblait être l'occasion rêvée de me consacrer à nouveau à l'apprentissage intensif. J'allais de nouveau avoir quelques années, stables et libres, pour apprendre. Quoi ? Tout. Je voulais tout savoir, tout connaître, enfin au moins apprendre suffisament pour devenir un chercheur. Et je me suis donc mis, de tout mon coeur, à chercher à apprendre...

Et voilà, après trois ans de recherches intensives, qu'ai-je trouvé ? Difficile à dire... Mais, s'il y a bien une chose que la thèse m'a apprise, quelque chose que la vie nous apprend tout aussi bien d'ailleurs, c'est que quand on cherche, on trouve. Toujours. On ne trouve pas toujours ce que l'on cherche, certes. Mais le monde ne serait-il pas terriblement fade si c'était le cas ? Je n'aurais en tout cas pas trouvé un millième de ce que j'ai découvert. Car comment aurais-je pu m'attendre à tant de révélations, comment aurais-je pu anticiper tant de prises de conscience, comment aurais-ju pu envisager le moindre des rebondissements de mes explorations si je n'avais que rencontré ce que j'attendais ? J'en suis aujourd'hui convaincu, le chercheur est toujours récompensé, à la seule condition qu'il réussisse à reconnaître sa récompense. Même si ce manuscrit présente une partie des résultats de mes recherches et si le titre de docteur représente une certaine récompense, ce que j'ai trouvé et appris en dépasse les étroites marges et la grandiose symbolique. Mes trouvailles, je les garde au fond de mon coeur, précieusement, et je ne remercierai jamais assez le ciel de m'avoir tant gâté. Rien que d'y penser, j'en ai le vertige, et j'ai bien peur d'être incapable d'éprouver une gratitude à la mesure de ce don faramineux. Et quand bien même le pourrais-je, comment fait-on pour remercier le ciel ? A cela, chacun sa technique, et même si je compte bien continuer de développer les miennes dans la suite de ma carrière de *chercheur*, je vais pour l'instant me contenter de remercier ses principaux agents impliqués dans la bonne conduite de cette thèse.

Mes premiers et plus chaleureux remerciemments vont tout naturellement à mon incroyable directrice de thèse, j'ai nommé Julie. Sache que j'ai immédiatement su en te voyant, comme un coup de foudre doctoreux, qu'il fallait que je fasse ma thèse avec toi. Tu as d'abord montré quelques réserves à mes avances, mais l'évidence de mon destin a rapidement su te convaincre. Et l'expérience a su me prouver à quel point mon intuition était bonne. Merci Julie. Merci infiniment. Tu as été une merveilleuse directrice de thèse, la meilleure que je n'ai jamais eu pour tout te dire. Merci pour tout, merci pour ce que tu es. Quelqu'un m'avait annoncé, avant que je commence, que tu étais (je cite) "une belle personne", le tout prononcé avec cette dose de solennité et de conviction qui en dit long. Et bien, je confirme, le dire est long. Je remercie ensuite Charles, mon deuxième directeur de thèse, avec qui j'eus des rapports plus lointain (surtout après sa fuite du laboratoire), mais qui fut le co-directeur dont j'avais parfaitement besoin. Merci Charles pour ta bienveillance et ton professionalisme. Et merci au jury de ma thèse , aux rapporteurs et aux quelques collègues qui se sont donnés la peine de se pencher sur mon modeste travail.

Sans autre transition que cette phrase, je remercie tout mes collègues et en particuliers mes compères doctorants, éphémères eux-aussi de leur état, qui m'ont plus ou moins supporté durant ces trois longues années, et avec qui ce fut toujours un plaisir de profiter de la splendeur de la terrasse (et de son rebord, enfin je ne vais pas m'épancher sur le sujet...). Une petite pensée particulière pour les merveilleux anges du labo qui m'ont béni de leur présence tout au long de ma thèse et à qui je souhaite un prochain et heureux passage dans l'au-dela. Merci à ceux qui m'ont cotoyé de près : Alessandro (cretino !), Juliana (<3), Cambyse (ma bible !), Pierre (le prend pas mal de ne pas avoir de private), Vincent (Oh Putain ! C'est marée basse...), Claire (tu vas bien finir par t'énerver), Anton (vieux frère), Pierre-Louis (ouais mais toi t'es un rugbyman), Alasdair (j'espère que cette parenthèse n'est pas trop longue), Léo (bof), Pierre le jeune (je te fous ta raclée à Mario Kart), Arthur (Puisse-tu un jour voir la lumière), Ousmane (ton grand coeur cache un grand homme), Remy (et ton levain alors !). Merci à ceux que j'ai moins connu, soit par éloignement de bureau, de génération ou d'âme : Warith, Fabien, Alkéos, Valentin, Vivien, Allan, Andrea, Antoine, Antoine (un autre), Mélina, Anne-Sophie, Yen, Noura, Remy, Zoé, Florian, Kevin et Paul. Je vous porte et porterai tous à jamais dans mon coeur. Merci également à Georges, Marc, Camille, Lionel et Antoine, les rares permanents avec qui le contact me fut agréable. Je remercie également le MAP5, ce petit oasis parisien au milieu de ce désert du Sahaclay, qui a bien voulu m'accueillir, et l'université de Paris Descartes la Sorbonne France Monde Univers des Saint-Pères (rayer les mentions inutiles), pour ses locaux inoubliables, ayant la particularité unique et étonnante d'être à la fois affreux et génialissimes.

Je remercie mes proches, qu'ils le soient depuis longtemps ou depuis peu, pour la chaleur de leur proximité. Merci à ma sublime famille (pour ne pas dire belle), je vous aime, et je ne prends que petit à petit conscience de la chance que j'ai eu et j'ai de pouvoir grandir à vos côtés. Merci papa, Patrick, pour tant de choses qu'il serait vain d'essayer de les lister. Merci maman, Sandrine, pour tout ce que tu as voulu m'apporter. Merci à mon frérot, Charles, avec qui l'histoire ne fait que commencer. Merci à ma soeurette adorée, Marie-Claire, que je ne pourrai jamais cesser d'aimer. Merci Juliette pour ta complicité et ta compagnie dont je ne me lasse pas. Merci Pauline pour nos (trop rares) moments passés ensemble et ton énergie si unique (et m'avoir amené à Gennetines :P). Merci Arthur pour tout ce que tu m'as transmis et continuera de me transmettre. Merci Stephane et Marie d'être deux magnifiques personnes en plus d'être un merveilleux couple (et accessoirement de m'avoir sauvé la vie). Merci Henry pour ton soutien inconditionnel. Merci également à Michel,

Liliane, Thierry, Céline, Thibault, Robin, Mathilde, votre rôle dans ma vie n'est pas moins vital. Et je remercie tout mes amis, d'un temps ou de toujours, avec qui j'ai appris et vécu tant de chose. Merci Jean-Noël d'être depuis toujours un insatiable compagnon de vie et de découverte. Merci pour ta présence, ta confiance depuis tant d'années. Merci Zéphyr et Julien, je suis heureux de vous avoir rencontré et d'avoir partagé de si belles choses avec vous. J'espère que cela continuera. Merci Merci Martin (x2) et Virgile pour vos belles leçons de vie que vous continuez malgré vous à me donner. Merci Djerby simplement d'être, j'aurais aimé que l'on puisse plus se voir. Merci Charles pour notre relation naissante et distante. Merci à Jeremy, Clémence, Elodie et toute la clique de mazagran pour leur belle énergie et toutes ces soirées et moments partagés. Merci Gab pour ton investissement dans mes projets cinématographiques, j'en reviens toujours pas que tu m'aies aidé à ce point. Merci Matthieu, Maxence, Florence, Mathilde et le destin pour votre participation dans mes projets, j'espère un jour pouvoir vous rendre la pareil... Merci Pierre, Damien et F-H pour nos coinches semestrielles. Merci Augustin de m'avoir fait découvrir le PSB, grand club s'il en est, avec qui j'ai partagé une belle et émouvante dernière saison de basket, dédicace également à Benjamin pour nos soirées (footing) pizza film. Vielen dank Vera, du hast noch keine Idee wie wichtig und reissig dein einfluss auf meinem Leben war. Et un grand merci également à mon sublime studio, mon bébé, qui m'a abrité avec passion durant ces quelques années et continuera pour quelques temps encore, et qui m'a permis de découvrir la magie du 243. Merci à Mélinée, Elie, Ted, Fred, Angélique, Hugues, Claire, Anaelle et les autres, sans qui le charme mythique de cette copro ne serait encore qu'une légende. Merci également au petit Jacques pour leur divin fromage et leur bonne humeur inépuisable.

Enfin, le plus important et le plus superflu, je remercie toutes les personnes dont j'ai croisé le chemin et qui m'ont tant apporté et donné : Gisèle, Maïla, Antonio, Aaron, Clémence, Constance, Claire, Solène, Jeanne, Chloé, Gaëlle, Marion, Marie, Hugo, Thomas, Theresa, Mélissa, Raf, Kirsten, Eric, Caroline, Maurice, Yves, Radjiv, Anne, Alex, Hugo, Victor, Pierre, Théo, Eve, Fataneh, Antoine, Tatiana et ceux que j'oublie. Merci à toute les personnes qui m'ont prises en stop, avec qui j'ai partagé une danse, un sourire ou plus, et sans qui la vie serait bien morne. Merci Frieda, pour ton amour qui fut si inédit pour moi. Un remerciement spécial à Anca, sans qui mes recherches n'auraient jamais pu décoller, et qui fut l'axiome fondamental de mes nouvelles théories de vie, et un spécial thanks à Billie avec qui je n'ai pas fini d'apprendre et de chercher le mystère de tout mon coeur.

Merci.

Contents

Notations

Here are the conventions and notations used in this thesis :

- For two real numbers a and b, $[\![a, b]\!]$ refers to the set of all integers i such that $a \leq i \leq b$.
- The Identity function on any space E : $\begin{cases} E & \to E \\ x & \to x \end{cases}$ is denoted by I_E and, if there is no ambiguity by I. In a context of square matrices, I_n means the $n \times n$ identity matrix.
- $\mathbb{1}_n$ is the vector of ones of size n.
- In a probabilistic context, Random variables are denoted with upper-case letters (X for instance), while their values are denoted with lower-case letters (x). If ν is a probability distribution, then $X \sim \nu$ means that ν is the density of X.
- If E is a space, then $\mathcal{P}(E)$ is the set probability distributions on this space.
- If X is a continuous (resp. discrete) random variable, we denote by $p(X = x)$ the value of the probability density function (resp. the probability) of X at x.
- $X \sim \mathcal{N}(\mu, \Sigma)$ means that X is a Gaussian random variable, with expectation μ and covriance Σ. With a slight abuse of notation, we shall use the same notation for the Gaussian density function of expectation μ and covariance Σ with respect to the Lebesgue measure γ on $\mathbb{R}^d$, namely

$$\forall x \in \mathbb{R}^d, \mathcal{N}(x|\mu, \Sigma) = \frac{1}{(2\pi)^{d/2}\sqrt{det\Sigma}} \exp\left(-\frac{1}{2}(x-\mu)^T\Sigma^{-1}(x-\mu)\right).$$

- Matrices are denoted with uppercase letters, and vectors by lowercase letters. The coefficients of a matrix M (resp. a vector) of size $n \times m$ (resp. of size n) are denoted by $M_{i,j}$ (resp. a_i) for $i \in [\![1, n]\!]$ and $j \in [\![1, m]\!]$. The trace of a matrix M is denoted by $Tr(M)$, its determinant by $\det(M)$, its adjoint by M^T, and its Frobenius norm by $||M||_{Frob} = \sqrt{Tr(M^T M)}$.
- The Kronecker's product between two matrices $M \in \mathbb{R}^{n \times m}$ and $M' \in \mathbb{R}^{n' \times m'}$ is denoted by $M \otimes M' \in \mathbb{R}^{nn' \times mm'}$ and is defined as:

$$M \otimes M' = \begin{pmatrix} M_{1,1}M' & M_{1,2}M' & \cdots & M_{1,m}M' \\ M_{2,1}M' & M_{2,2}M' & \cdots & M_{2,m}M' \\ \cdots & \cdots & \cdots & \cdots \\ M_{n,1}M' & M_{n,2}M' & \cdots & M_{n,m}M' \end{pmatrix}$$

- For $f : X \to \mathbb{R}$ and $g : Y \to \mathbb{R}$ two functions, $f \oplus g : X \times Y \to \mathbb{R}$ is a function defined by $\forall (x, y) \in X \times Y, (f \oplus g)((x, y)) = f(x) + g(y)$. Similarly, for two vectors $a \in \mathbb{R}^n$ and $b \in \mathbb{R}^m$, $a \oplus b$ is the matrix of size $n \times m$ such that $\forall (i, j) \in [\![1, n]\!] \times [\![1, m]\!], (a \oplus b)_{i,j} = a_i + b_j$.

- Let E be a finite set and consider a map $\phi : E \to \mathbb{R}$, and F a subset of E. We denote by $\phi_{|F}$ the restriction of the map ϕ to F. If ν is a probability distribution on $\mathbb{R}^E$, we define $\nu_{|F}$ the marginal distribution of ν on $\mathbb{R}^F$. If $F = \emptyset$, by convention, we define $\nu_{|F} = 1$.
- In the context of image processing, u will refers to a natural image, $\widehat{u}$ to a noisy version of u, and $\widetilde{u}$ to a restored (denoised) version of u.

Introduction

Humanity took over 1.000 billions pictures in 2017, which represent over 30.000 pictures each seconds. Each of these pictures were taken by a digital camera, and most of them from a smartphone. With the recent explosion of this device, digital images became extremely accessible and are widely used in many aspects of life. Smartphone cameras enable to make photographs very easily, thanks to their small size and their automation. These aspects make it an incredible tool as well as a big technical challenge: how to obtain high-quality pictures from a so small and light device?

A digital camera is roughly speaking a lens and a grid of sensors. Each sensor can count the number of photons it receives during a short amount of time (*the shutter time*). The smaller the sensor, the less light is receives and therefore the fewer photons it counts, which means a low luminosity. To increase it, we can increase the shutter time, but this is not always possible and leads to blurry images if the object or the photographer is moving. The other solution is to numerically increase this number by multiplying it by a constant: it is called the *ISO*. In theory, this enables to obtain images as bright as we want even with very small sensors. But in practice, the ISO also multiplies the errors of measurement (it also exists for analogue cameras). Imagine a very reliable sensor, that counts an additional photon from time to time. If this sensor, receives 1000 photons during the shutter time, it may count 1001, which makes an imperceptible difference. Now, let us imagine the same sensor, but 10 times smaller, if would receive instead $^{1000}/_{10^2} = 10$ photons and would count 11, which makes it 1100 instead of 1000 on the same scale of luminosity. This error is an example of noise, and more generally of image degradation. The description above is obviously a huge simplification of the actual functioning of a digital camera, and many other issues and phenomenons have to be dealt with to obtain a photograph. Between the RAW image (the genuine information measured by the sensors) and the actual image saved in the gallery, lots of algorithmic treatments have to be made to transform this information into a deliverable image (this is besides why the previsualization of the smart phone camera is in "real time" on the screen while we have to wait a few seconds after taking a picture). Among them, some must correct the degradations and improve the imperfections of the RAW signal: it is the goal of image restoration.

There are many types of image degradations: noise, missing data, deformation, compression, blur, etc... Even the noise, while being the most basic example of image degradation, can be of many different forms (shot noise, white noise, anisotropic noise, film grain, etc…). Images can also be taken in lots of context and by lots of devices: MRI, satellite, camera, smartphone camera, drawing, etc.... Image restoration is the answer to the following inverse problem: given a degraded image (and a model of the degradation), recover the underlying "original" image. It is a wide and rich topic, as there is a big variety of degradations and original images: the restoration heavily depends on the type of image. The restoration of a MRI result is very different from of satellite images or from a portrait. Besides, in most practical case, there is no real "original image", it has to be created by the image restoration algorithm according to some criteria (help the diagnostics, look nicer, help the algorithms

of automatic surveillance, etc...).

In this thesis, we focus on photographs that could have been taken by any personal imaging device, that we shall call *natural images*. We shall also only consider the noise degradation, and in particular white Gaussian noise. Yet, the ideas developed here can be extended to wider type of problems. As we shall see, this assumption is quite reasonable for practical applications on natural images, and still includes the main challenges behind image restoration: How to mathematically capture and reconstruct the essence of natural images ?

The first part exposes a first work linking Bayesian model and an image processing operation called *aggregation*. This work introduces a new framework which can be naturally extended to a more general formulation on distributions. The second part is concerned by this problem from an optimal transport perspective.

Part I: A unified view on patch aggregation

In part I, we present a study on patch aggregation, based on Saint-Dizier, Delon, and Bouveyron, 2020. After presenting the conceptual and mathematical background of patch based methods for image denoising in Chapter 1, we show the limits of the actual patch aggregation schemes and propose a new one in Chapter 2.

Patch-based methods are efficient in image restoration and image processing in general. A patch is a small part of the image, generally square and connected, i.e. a set of $n \times n$ adjacent pixels. Patch-based methods rely mainly on the same framework divided in 3 steps:

- **Patch extraction:** It consists in transforming the image into a set of patches. Usually, a given size s is chosen, and all the overlapping patches of size $s \times s$ are extracted. Some work have been made on more sublte patch extractions, like the adaptative patch size methods presented in Deledalle, Duval, and Salmon, 2012, and some recent developments in texture synthesis suggest that careful patch extraction can greatly improve the computational time for a minor loss of performance (see Launay and Leclaire, 2019).

- **Patch editing:** It consists in processing the set of patches obtained from the patch extraction instead of directly the image u. It enables to use powerful data processing tools that would not scale to the size of the image. The patches are usually used to infer a model which serves to restore the patches, with a MLE or a MAP estimator.

- **Patch aggregation:** After the patch editing step, the patches a priori no longer agree, which means that they do not have the same value on their overlap. The reconstruction of the image is therefore not straightforward.

The concept of patch and patch-based methods are illustrated in Figure 1. These methods implicitly assume the *self-similarity principle*, which states that the image is redundant and that small parts of the image (the patches) are repeated with only slight variations. The success of patch-based methods on natural images have shown the relevance of this principle, however it would not be valid on different type of images, like MRI or images in Fourier space.

Patch editing step have suscited a lot of attention in the past few years and have benefited from the recent development of statistics, data science and machine learning. Lots of different models and inference methods have been proposed for this step. Among them, we have some main categories: dictionary based methods

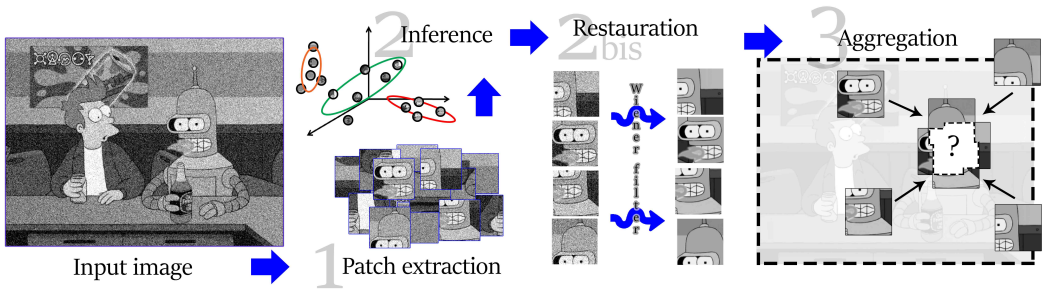


FIGURE 1: Illustrations of the different step of patch-based denoising.

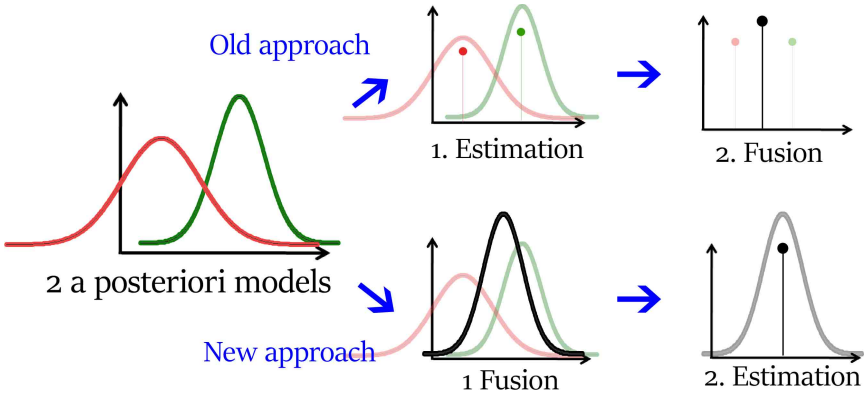


FIGURE 2: Illustration of the new proposed paradigm for the patch aggregation. The aggregation is no longer a fusion of estimations of patch model, but instead a single estimation of patch models fusion.

(Aharon, Elad, and Bruckstein, 2006 for instance), sparse reresentation based methods (Yu and Sapiro, 2011, Danielyan, Katkovnik, and Egiazarian, 2012), non-local category (Buades, Coll, and Morel, 2005, Chatterjee and Milanfar, 2011) and Bayesian inference (Zoran and Weiss, 2011). Lots of them rely on Gaussian models (Lebrun, Buades, and Morel, 2013) and Gaussian mixture models (Houdard, Bouveyron, and Delon, 2017, Zoran and Weiss, 2011).

We first show that, with the current state of the art of the patch editing step, the main limitation of patch based methods comes from the aggregation step. Indeed, most methods estimate a restored version of each patch (using their model) and only aggregate them into the image, using the so-called *uniform aggregation* (simply averaging the patches together) or some basic variations. This approach prevents from using efficiently the information of the model to reconstruct the image, and assumes (mistakenly) that the restored versions of the patches are independent.

To tackle this limitation, we propose a new paradigm in patch aggregation as presented in Figure 2. This implies extending the definition of patches to what we called the *patch models*. A patch model P is defined by

$$P = (\Omega, \nu),$$

where $\Omega \subset \mathbb{R}^2$ is the *domain of the patch model* (its location on the image) and $\nu \in \mathcal{P}(\mathbb{R}^{|\Omega|})$ is *the distribution of the patch model*.

This formalization enables more flexibility when handling patches, and permits to define the notions of the agreement and fusion of patch model. We define the fusion of two patch models (with bounded densities), $P_1 = (\Omega_1, f_1 dx)$ and $P_2 = (\Omega_2, f_2 dx)$, by $P_1 \odot P_2 = (\Omega, f dx)$ with $\Omega = \Omega_1 \cup \Omega_2$ and

$$\forall x \in \mathbb{R}^\Omega, \quad f(x) = \frac{f_1(x_{|\Omega_1}) f_2(x_{|\Omega_2})}{\int_{z \in \mathbb{R}^\Omega} f_1(z_{|\Omega_1}) f_2(z_{|\Omega_2}) dz}.$$

This operation is symmetric, associative and transitive, and thus enables to define a patch model aggregation as a fusion of all patch models. This new approach of patch aggregation turns out to embed all the previous aggregation schemes, and generalizes the notion of EPLL of Zoran and Weiss, 2011. This operation also has a close form solution for Gaussian distributions, which makes it directly applicable for Bayesian patch-based methods.

This idea of merging patch models to create a bigger patch model "containing" them seems promising, but the proposed fusion suffers from limitations. The problem can be considered with more generality as such: how to merge (or interpolate) different distributions living in different space but having some overlapping components. This question is more deeply related to distribution theory and led us to consider it using optimal transport, as a tool to handle distributions. This problem is the core of part II.

Part II: Generalized Wasserstein Barycenter

In this part, we present a natural extension to the problem raised in part I for patch models. Chapter 3 is a survey of a classical optimal transport theory and some of its variant like Wasserstein barycenter, multi-marginal optimal transport and the entropic regularization. Chapter 4 introduces a new problem raised by the study of Chapter 2 adapted to optimal transport theory. It turns out that it can be casted as a generalization of the Wasserstein barycenter problem.

Optimal transport theory is centered around the Kantorovich (or Monge-Kantorovich) problem:

$$\mathcal{L}_c(\mu, \nu) = \inf_{\gamma \in \Pi(\mu,\nu)} \int_{X \times Y} c d\gamma.$$

μ and ν are the original and target distribution. $\mathcal{L}_c(\mu, \nu)$ is called the *transport cost* and enables to define a (non-trivial) distance between distributions. Chapter 3 is devoted to introduce this problem and the main tools and results that will be useful throughout this thesis, with some insight on extensions and deeper studies. It will introduce in particular the multi-marginal optimal transport problem, as a generalization of the Kantorovich problem, that will turn out to be of special interest to our original motivation.

We will consider the following problem: Given $(\nu_1, \dots, \nu_K)$ some probability distributions, $c_1, \dots, c_K$ some cost functions and $P_1, \dots, P_K$ linear applications, with $P_i : \mathbb{R} \to \mathbb{R}^{d_i}$, find

$$\inf_{\nu \in \mathcal{P}(\mathbb{R}^d)} \sum_{k=1}^{K} \mathcal{L}_{c_i}(\nu_i, P_i \# \nu). \tag{GWB}$$

A solution of Equation GWB is called a *generalized Wasserstein barycenter*. $P_i \# \nu$ is called the *push-forward* of ν by P_i, and corresponds to the distribution naturally induce on $\mathbb{R}^{d_i}$ by ν through P_i. In the case where P_is are the canonical projections

$\mathbb{R}^d \to \mathbb{R}^{d_i}$, $P_i\#\nu$s are the marginals of ν according to the variables involved in the P_i and we find the setup corresponding to the patch aggregation (or fusion) presented in Chapter 2. The aggregation of a set of patch model can be seen as a generalized Wasserstein barycenter.

Chapter 4 presents a study and a characterization of generalized Wasserstein barycenter in the case where the cost functions c_i are all square of Euclidian norms. This problem is deeply related to the thoroughtful study of 2-Wasserstein barycenter problem by Agueh and Carlier, 2011, and the multi-marginal approach for tomographic reconstruction of Carlier, Oberman, and Oudet, 2015. However, those two problems are particular cases of (GWB) which requires more subtleties, especially in the definition of the dual.

Still, we provide some similar results, and show that (GWB) is linked to a multi-marginal problem

$$\inf_{\gamma \in \Pi(\nu_1,\dots,\nu_K)} \int_{X_1 \times \cdots \times X_K} c d\gamma,$$

with c carefully chosen. This enables to solve the problem through generalized entropic regularization and Sinkhorn iterations. The problem also turns out to be Gaussian friendly. We show some results applied to geometrical reconstruction and to the fusion of patch models.

Part I

A unified view on patch aggregation

Chapter 1

Patch-based methods and image denoising

Introduction

This chapter presents the mathematical and conceptual background to the ideas presented in this first part. Most of the ideas presented here will be useful to develop the work presented in Chapter 2, and to understand the motivation behind it. This involves the basics of the EM-algorithm and some of its applications, especially to the Gaussian Mixture Models (GMM) and some extensions in Section 1.1. Section 1.2 presents the state-of-the-art for patch-based image denoising and some of the main concepts behind the most popular algorithms. I included besides some leads which naturally arise from this presentation that I have unsuccessfully explored and eventually let down. Yet, they still have their relevance, as they complete my thoughts, and they may be the basis for some future deeper explorations.

1.1 EM-algorithm and mixture models

1.1.1 Gaussian Mixture Models (GMM)

A mixture model is a way to model data with a distribution obtained by combining several other distributions.

Denoting $(\phi_k)_{k\in[1..K]}$ a set of K probability distributions on $\mathbb{R}^n$ and $\pi \in [0,1]^K$ such that $\sum_k \pi_k = 1$, we can define the mixture model

$$\forall x \in \mathbb{R}^n, p(x) = \sum_{k=1}^{K} \pi_k \phi_k(x),$$

where K is called the number of components, $(\phi_k)_{k\in[1..K]}$ are the components and π is the vector of mixture coefficients. The components can be any distribution chosen arbitrarily, even other mixtures. The distribution p can be seen as a blend of the ϕ_k, according to the proportion π_k, which explains the terminology "mixture". This is a very powerful way to combine simple models in order to obtain a more complex and generic one.

A simple way to sample from the mixture distribution p is to choose first one of the component ϕ_k, with probability π_k, and then sample from it, which leads naturally to a classical and very powerful way to model the mixture model. It consists in considering a latent variable Z, unobserved, which refers to which component of the mixture the corresponding observed variable X should be sampled. The couple (X, Z) follows the graphical model presented in figure 1.1. From a data set perspective, Z is then an indicator telling from which component X is coming. This way of

FIGURE 1.1: Graphical model of a mixture model (Image from Bishop, 2006)

seeing a mixture model is very useful in regards to its applications, for instance for clustering purposes, which consists in finding the "right" z for every observed data x.

In this regard, Z is naturally modeled using a multinomial distribution $Z = (Z_k)_{k\in[\![1,K]\!]}$ with

$$p(Z_k = 1) = \pi_k.$$

Z could be as well modeled by a discrete uniform distribution on $[\![1, K]\!]$ for instance, but the multinomial modelling is more convenient, since we can easily write the distribution of Z in a close form :

$$p(z) = \prod_{k=1}^{K} \pi_k^{z_k}.$$

Then the distribution of X conditionally to Z is naturally defined by $p(x|z_k = 1) = \phi_k(x)$, which gives the final formula:

$$p(x, z) = \prod_{k=1}^{K} (\pi_k \phi_k(x))^{z_k}.$$

We can check that, indeed, we have $p(x) = \sum_z p(z) \times p(x|z) = \sum_{k=1}^{K} \pi_k \phi_k$. We can also use the Bayes formula to have the posterior distribution

$$p(z_k = 1|x) = \frac{\pi_k \phi_k(x)}{\sum_j \pi_j \phi_j(x)}. \tag{1.1}$$

All these considerations enable to use the powerful tools of the graphical models theory to work efficiently with mixture models. One of their biggest strength is that they can in most case approximate any kind of distribution, which makes them very useful in data processing. However, most of the mixture models lead to complex inference, even with simple and well-understood basic distributions such as Gaussian distributions.

A mixture model with only Gaussian components is called a Gaussian Mixture Model (GMM). This mixture model is very widely used because of its flexibility, its low complexity and, as we shall see, its compatibility with the EM-algorithm. The components of a GMM can there be written as

$$\phi_k = \mathcal{N}(.|\mu_k, \Sigma_k),$$

where (μ_k, Σ_k) corresponds to the parameter of the k-th Gaussian. The mixture models have then additional parameters, that we can regroup into $\theta = (\theta_k)_{k\in[1..K]} = (\pi, \mu, \Sigma)$, which yields our GMM distribution:

$$p(.|\theta) = \sum_{k=1}^{K} \pi_k \mathcal{N}(.|\mu_k, \Sigma_k).$$

Thanks to their flexibility, GMMs can be used to approximate distributions. The inference consists in estimating a vectorial parameter θ. A very natural and popular way to infer θ is to use the maximum likelihood principle. We need for that to express the total log-likelihood $\mathcal{L}(x, \theta)$ for an arbitrary set of samples $x = (x_n)_{n\in[\![1,N]\!]}$ and our parameter θ of the GMM:

$$\mathcal{L}(x, \theta) = \sum_{n=1}^{N} \log(\sum_{k=1}^{K} \pi_k \mathcal{N}(x_n|\mu_k, \Sigma_k)). \tag{1.2}$$

Even if this expression seems easier to work with than the regular likelihood, maximizing it doesn't lead to a convex problem and is not tractable in practice. To tackle this issue, some iterative method must be used to try to obtain some local maximum, like a gradient descent (see Gepperth and Pfülb, 2019 for instance) or the EM-algorithm (see McLachlan and Krishnan, 2007) which will be described in the next section.

Before going any further in how to optimize the (log-)likelihood, it is interesting to think about what we really wish to obtain. The maximum likelihood is in general a good estimator, since we can prove its consistency and normality under certain standard conditions. But in our case, the log-likelihood function is not bounded. For instance, if one of the component is exactly located at a point and if the others are chosen randomly, then the likelihood of the configuration can tend arbitrarily fast to infinity as the variance of the component on the point tends to zero.

Proposition 1. *We have*

$$\sup_{\theta} \mathcal{L}(x, \theta) = +\infty.$$

Proof. We choose $\theta_\sigma = (\pi_k, \mu_k, \Sigma_k)_{k\in[\![1,K]\!]}$ such that $\forall k \in [\![1, K]\!], \pi_k = \frac{1}{K}$, $(\mu_1, \Sigma_1) = (x_1, \sigma^2 \mathrm{I}$ and $\forall k \geq 2, (\mu_k, \Sigma_k) = (0, \mathrm{I})$.

Then, we have

$$\begin{aligned}
\mathcal{L}(x, \theta) &= \sum_n \log\left(\frac{1}{K}\sum_k \mathcal{N}(x_n|\mu_k, \Sigma_k)\right) \\
&\geq \sum_{n=2}^{N} \log\left(\frac{1}{K}\sum_{k=2}^{K} \mathcal{N}(x_n|\mu_k, \Sigma_k)\right) + \log\left(\frac{1}{K}\sum_k \mathcal{N}(x_1|\mu_k, \Sigma_k)\right) \\
&\geq A + \log\left(\frac{1}{K}\mathcal{N}(x_1|x_1, \sigma^2\mathrm{I})\right)
\end{aligned}$$

with $A = \sum_{n=2}^{N} \log\left(\frac{1}{K}\sum_{k=2}^{K} \mathcal{N}(x_n|\mu_k, \Sigma_k)\right)$, which does not depend on σ.

We have besides $\mathcal{N}(x_1|x_1, \sigma^2\mathrm{I}) = \frac{1}{(2\pi)^{d/2}\sigma}$, so we have

$$\mathcal{L}(x, \theta_\sigma) \to_{\sigma\to 0} +\infty.$$

□

This shows that we do not wish to achieve a global maximum (which will always be $+\infty$), but rather find the “best” local maximum, which should be obtained among the set of "good" θ, which corresponds to mixture of non-degenerated Gaussian distributions.

1.1.2 EM-algorithm

The EM algorithm is a powerful way to find local maxima of the log-likelihood function following the graphical model of figure 1.1. There are several ways to understand the EM algorithm. One of the most intuitive, as presented for instance in Bishop (2006) (see for more details), is to interpret it with the Kullback-Leibler divergence, or KL-divergence.

Recall of information theory

We recall here some basics on information theory. The quantity of information given by an observation x of X is defined by

$$I(x) = -\log p(x).$$

The idea behind this definition is to consider that the rarest an event occurs, the most information it contains when it happens. We have then $I(x) = 0$ if the event x is "obvious" ($p(x) = 1$), and $I(x) = +\infty$ if the event x is "impossible" ($p(x) = 0$). Indeed, observing something impossible breaks the model and therefore gives infinite information. The choice of the log function is somewhat arbitrary, mainly due to commodity.

Then, we define the entropy of the random variable by

$$H[X] = -\int_x p(x) \log p(x) = \mathbb{E}[I].$$

The entropy is the average quantity of information one could expect to have with one observation of X. The entropy is always positive for discrete random variable (the integration becomes a sum of positive quantities), but can take negative values for variables with density.

In our application, we consider an unknown probability distribution p, and we want to approach it with another distribution q supposed to be known (a GMM for instance). A way to measure the distance between those two distributions from a probabilistic point of view is to consider the amount of information lost by getting them mixed up. It is then unavoidable to lose information by thinking that $X \sim q$ instead of $X \sim p$. In order to quantify this loss, we define the KL-divergence as:

$$KL(p||q) = -\int_x p(x) \log(\frac{q(x)}{p(x)})dx = \mathbb{E}_p\left[I_q\right] - \mathbb{E}_p\left[I_p\right].$$

This is the difference between the average information given by a random variable following q and by an independent one following p (which is optimal since it is the real distribution). The KL-divergence has some nice properties: it is a pseudo distance (non symmetric), but non-negative and vanishes if and only if $p = q$. For discrete variables, the KL-divergence is defined identically, with a sum instead of an integration.

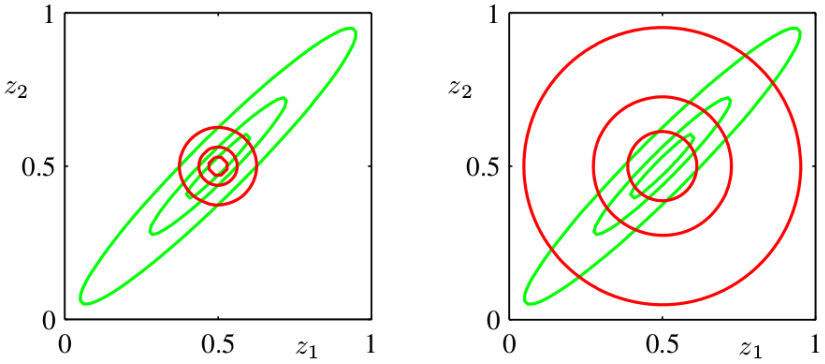


FIGURE 1.2: Illustration of the Kullback-Leibler divergence behavior. The green lines represent p, a Gaussian distribution we want to approximate with q, a circular Gaussian distribution (red lines). On the left, $KL(q||p)$ is minimized, so the red distribution cannot afford to take large values outside the green lines. On the right, $KL(p||q)$ is minimized, so the red lines must include the green ones. (Image from Bishop (2006))

Minimizing the KL-divergence corresponds to minimizing the loss of information induced by our approximation q, which is a decent goal in our problem. Hence, it is interesting to think about what will make this quantity small.

- If p is small in a region where q is large, then the log term will take large values, but the contribution of this region will still be small due to the linear term in p.
- If q is small in a region where p is large, then the log term will take large values, but won't be reduced by the linear term.

This behaviour is highlighted on Figure 1.2. Hence, minimizing $KL(p||q)$ with respect to q is trying to approximate p where it is large, and minimizing the inverse is trying to avoid p where it is low.

Motivation of the EM algorithm

Suppose that X and Z are two random variables, following the graphical model presented in Figure 1.1. We introduce a new density distribution q on the latent variable Z. Recalling Equation (1.2), we can now observe the following decomposition:

$$\log p(X|\theta) = L(q,\theta) + KL(q||p(\cdot|X)) \tag{1.3}$$

with $L(q,\theta) = \sum_Z q(Z) \log(\frac{p(X,Z|\theta)}{q(Z)})$ and $KL(q||p) = -\sum_Z q(Z) \log(\frac{p(Z|X,\theta)}{q(Z)})$.

Since the KL is non-negative, we have a lower-bound $L(q,\theta)$ on the log-likelihood, which is more likely to be convex. Instead of maximizing directly the likelihood function, we can maximize its lower bound, and then refresh it from the new position and so on. The EM-algorithm is the succession of these two steps, which are called the E-step and the M-step.

- **E-step** : Maximize the lower bound, which is equivalent to minimize $KL(q||p(\cdot|X))$ with respect to q. If this is tractable, this simply corresponds to set q to be $p(\cdot|X)$ (See Section 1.1.2).
- **M-step** : Maximize $L(q,\theta) = \sum_Z q(Z) \log(\frac{p(X,Z|\theta)}{q(Z)})$ with respect to θ. This is equivalent to maximize $\sum_Z q(Z) \log(p(X,Z|\theta)) = \mathbb{E}_q[\log(p(X,Z|\theta)]$.

Algorithm

Algorithm 1 EM-Algorithm

Input: I.i.d. observations $x = (x_1, \dots, x_n)$ of the random variable X
Output: Parameter θ
1: Initialize the parameter θ
2: **while** not converged **do**
3: Set $q = p(.|x) = \arg\max_q L(q, \theta)$
4: Set $\theta = \arg\max_\theta \mathbb{E}_q[\log(p(x, Z|\theta)]$
5: **end while**

The EM-algorithm is presented on Algorithm 1. It consists in an alternate maximization of a function of two variables. But, since we have from (1.3) that $\log p(X|\theta) \geq L(q, \theta)$ for any q and θ, the likelihood only increases through the steps of the algorithm. This ensures that the algorithm converges to a maximum, which will likely be local if the initialization is not ill-conditioned, i.e. if the likelihood does not diverge next to the initialization. This fact shows that the EM algorithm is very initialization dependent, and how to correctly initialize it depends on the problem and is most of the time still an open issue.

EM in practice

As for all the iterative algorithms, the initialization is crucial in order to use properly the EM algorithms, also because we already know that there are no global maximum, so we have to be close to the local maximum that we are looking for before the beginning of the algorithm. When used for clustering purposes, the most popular way to initialize it is to use a K-means algorithm, which can also be seen as a version of the EM-algorithm (Bishop (2006)). However, this is mostly chosen for convenience: there is no proof of its efficiency and it does not necessarily give the best results in practice. Several techniques have been proposed to help finding the best local maximum like Small EM, CEM, SEM. For instance, some work have been made in Biernacki, Celeux, and Govaert (2003) to optimize the EM procedure specifically for the Gaussian mixture problem. Another interesting idea introduced by Ueda and Nakano (1998) is to smooth the likelihood function in order to remove the saddle points between the local maxima which prevent the algorithm from being stuck on bad local maxima and helps it reaching the highest ones. These methods are discussed in details in He et al., 2004 for clustering purposes. As said, initialization is still an open problem and have no general good solution. One must look up for each problem which solution suits the most.

1.1.3 Application of EM to Gaussian mixture inference

The EM-algorithm is very general and can be applied to any type of mixture. However, in the case of GMMs, the iterations have a close form solution, which makes it particularly useful to infer it.

As seen in Algorithm 1, the E-step for Gaussian mixtures consists in calculating the posterior distribution $p(.|X)$, and the M-step consists in maximizing the expected value of the total log-likelihood according to this distribution and with

respect to the model parameters. Recalling Equation (1.1), we have

$$p(z_k = 1|x) = \frac{\pi_k \mathcal{N}(x|\mu_k, \Sigma_k)}{\sum_j \pi_j \mathcal{N}(x|\mu_j, \Sigma_j)}.$$

Considering $(x_n)_{n\in[1..N]}$ a i.i.d sample of the mixture, we write $\tau_{n,k} = p(z_{n,k} = 1|x_n)$. The E-step consists finally simply in calculating the $\tau_{n,k}$.

For the M-step, we have to maximize

$$\mathbb{E}_{z|x}[\log \mathcal{L}(x, z|\theta)] = \sum_{n=1}^{N}\sum_{k=1}^{K} \mathbb{E}[z_{n,k}] \times \log(\pi_k \mathcal{N}(x_n|\mu_k, \Sigma_k)) = \sum_{n=1}^{N}\sum_{k=1}^{K} \tau_{n,k} \times \log(\pi_k \mathcal{N}(x_n|\mu_k, \Sigma_k))$$

with the constraint $\sum_k \pi_k = 1$ and $\pi \geq 0$. We can rewrite this expression

$$\mathbb{E}_{z|x}[\log \mathcal{L}(x, z|\theta)] = \sum_k N_k \pi_k - \sum_k \frac{1}{2} N_k \log\det(\Sigma_k) - \sum_k \frac{1}{2} N_k Tr(S_k \Sigma_k^{-1}) + cst$$

with $N_k = \sum_n \tau_{n,k}$ and $S_k = \sum_n \tau_{n,k}(x_n - \mu_k)(x_n - \mu_k)^T$. The maximization has a close form solution which can be found using a Lagrange multiplier, which gives the expressions for the M-step :

$$\mu_k = \frac{\sum_k \tau_{n,k} x_k}{N_k}, \ \pi_k = \frac{N_k}{N}, \ \Sigma_k = \frac{S_k}{N_k}. \tag{1.4}$$

1.1.4 High Dimensional Data Clustering (HDDC)

Even if the EM-algorithm is very useful and efficient to infer GMM, it has other limitations than the initialization. Performing the E-step and the M-step require to compute and invert the covariance matrices, which can become problematic as the dimension of the problem grows. Besides, applied to clustering in high-dimensional space can create numerical issues as the data become sparse (matrices are no longer full rank for instance). For these reasons, Bouveyron, Girard, and Schmid, 2007 introduced the High-Dimensional Data Clustering (HDDC). The idea is to add more control onto the covariances of the componentes of our Gaussian mixture model. Instead of considering the covariance matrix, we consider directly its diagonalization. For the sake of clarity, we will reason here on one cluster, but what follows can be applied on each cluster independently.

Let fix $k \in [\![1, K]\!]$. We consider the diagonalization of the covariance matrix $\Sigma_k = Q_k^t \Delta_k Q_k$ with Δ_k a diagonal matrix. We assume that the signal (or cluster) that we are trying to model lives in a subspace of dimension d_k, i.e. that the covariance matrix has d_k informative eigenvalues and that the $p - d_k$ remaining ones come from uniform noise, and therefore are identically equal to $b_k \in \mathbb{R}$. Hence, Δ_k has the form

$$\Delta_k = \begin{pmatrix} a_{k,1} & & & & (0) \\ & \cdots & & & \\ & & a_{k,d} & & \\ & & & b_k & \\ (0) & & & & \cdots \\ & & & & & b_k \end{pmatrix}. \tag{1.5}$$

This assumption is not restrictive, since any covariance can be written in this form for $d_k = p - 1$, but it greatly helps focusing on the dimension reduction. The

regularization can be toughed with some similar additional assumptions, that the different classes share the noise, or even impose that the first eigenvalues take the same value. All these models have been studied in Bouveyron, Girard, and Schmid (2007), and we will only consider the most general model called $[a_{kj}b_kQ_kd_k]$ (to respect the notations of the paper), because it's the most adapted to our application.

This hypothesis doesn't constraint the space of considered distributions, since for $d_k = p - 1$, we have the general Gaussian mixture distributions. Infering this model will naturally correspond to the standard GMM maximum likelihood since the other cases are just some more compact ways to write very particular cases of the GMM. But, as we wish to model the presence of noise and have a model as sparse as possible, we would like to have d_k as low as possible without altering too much the accuracy of the model.

Let us fix d_k for now. We can now compute this particular M-step:

$$\theta = \arg\max_{\theta\in\Theta} \mathbb{E}_q[\log(p(X, Z|\theta)]$$

where Θ is the restricted set of GMM parameters with covariance that can be diagonalized like in (1.5).

Using the notations of Section 1.1.3, having $\lambda_{k,1}, ..., \lambda_{k,p}$ the eigenvalues of the empirical covariance S_k, the parameters that maximize the complete log-likelihood (see Equation 1.4) are the same, except for the covariance, whose parameters are:

$$a_{k,j} = \lambda_{k,j} \text{ for } j \in [\![1, d_k]\!] \text{ and } b_k = \frac{\sum_{i=d_k+1}^{p} \lambda_i}{p - d_k}.$$

The result is very intuitive, since the eigenvalues corresponding to the signal are untouched, and the one corresponding the noise are averaged. Q_k correspond also in a natural way to the eigenvectors of Σ_k. It is important to note that in this parametrization, only the d_k first eigenvectors are useful to store. The $p - d_k$ other directions can be chosen arbitrary as an orthonormal basis of the remaining space. Furthermore, we can write b_k as follows

$$b_k = \frac{Tr(\Sigma_k) - \sum_{i=1}^{d_k} \lambda_i}{p - d_k}$$

which enables to estimate b_k precisely, since the eigenvectors corresponding to the largest eigenvalues are well estimated by the algorithms.

We considered here the case of a single cluster (and therefore of a single Gaussian distribution). More generally, we work with several clusters, each of them having its own intrinsic dimension d_k and potentially its own noise b_k. All the formulas above can be applied independently to each cluster. It is also possible to add more limitations to the GMM, for instance by forcing that they share the same dimension or the same eigenvectors. This leads to slight modifications in the formula, see Bouveyron, Girard, and Schmid, 2007 for more details.

Thanks to the introduction of the intrinsic dimensions $d_1, \ldots, d_K$ of the clusters, the HDDC extension of the EM algorithm applied to GMM has lot of computational avantages which become really relevant when working with high-dimensional spaces. The parameters are then simpler and faster to compute and store. If the d_k are known a priori as the data of the problem, HDDC is preferable to regular GMM.

If the intrinsic dimensions $d_1, \ldots, d_K$ are fixed before starting the EM algorithm, HDDC has the same convergence properties than the latter. However, in practice, we want to infer the intrinsic dimensions simultaneously and choose the $d_1, \ldots, d_K$ according to some criterion. In this case, the M-step becomes in general slightly

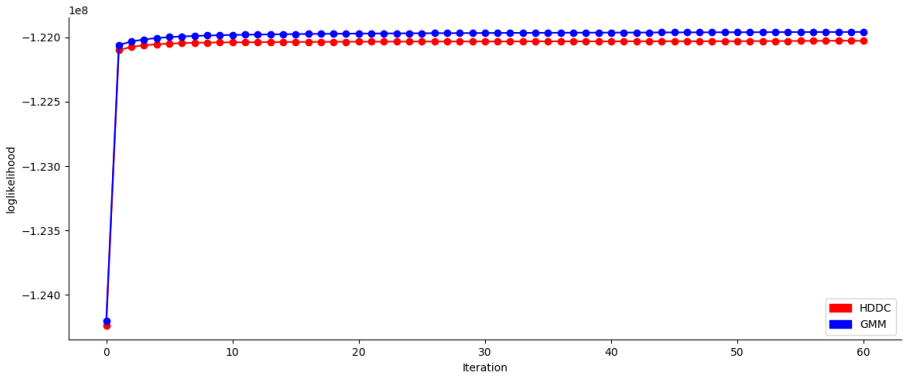


FIGURE 1.3: Evolution of the likelihood during the learning of the GMM and the HDDC model

suboptimal and the algorithm no longer has theoretical convergence guaranties. But in practice, the algorithm still converges to a local maxima (see figure 1.3).

The real issue of this technique is to be able to estimate the dimensions $d_1, \dots, d_K$, which are almost never inputs of the problem but often an important feature to estimate. As we said, if we relax this parameter, the maximum likelihood is obtained for $d_1 = \cdots = d_K = p - 1$, which corresponds to a classical GMM. Therefore, the problem of estimating *the right* d_k in this context is ill-posed. We would like the d_k to be as small as possible, but without loosing too much information, i.e. with a good estimation of the noise.

We could compute the likelihood with respect to the d_k to see how it behaves. This can be seen with this proposition :

Proposition 2. *The maximum likelihood of the M-step of the HDDC model is obtained with* $d_1 = \cdots = d_K = p - 1$.

Proof. The result is pretty obvious, since the HDDC model is equivalent to the GMM, so they have the same maximum likelihood. We can yet compute how the likelihood behave with the d_k, even if this dependence is very complex. We consider for simplicity only cluster, as this reasoning can be applied independently to all of them.

Let $f(d) = \max \mathcal{L}_d(x, \theta)$. We know that

$$\arg\max_{a,b,Q} \mathcal{L}_d(X, \theta) = ((\lambda_i)_{i\in[1..d]}, \frac{\sum_{i=d+1}^{p} \lambda_i}{p-d}, \text{ eigenvectors from } \Sigma)$$

Besides, $\arg\max_{a,b,Q,d} \mathcal{L}(X, \theta) = \arg\max_d f(d)$ because d is discrete. And we have

$$f(d) = cst - \frac{1}{2}[\sum_{i=1}^{d} \log \lambda_i + (p-d)\log(\frac{1}{p-d}\sum_{i=d+1}^{p} \lambda_i)] - \frac{1}{2}Tr(\Sigma_k^{-1} S_k),$$

where

$$Tr(\Sigma_k^{-1} S_k) = Tr(Q_k^T \Delta_k Q_k Q_k^T diag((\lambda_i)_{i\in[1..p]}) Q) = \sum_{i=1}^{d} \frac{\lambda_i}{\lambda_i} + \sum_{i=d+1}^{p} \frac{\lambda_i}{\frac{1}{p-d}\sum_{j=d+1}^{p} \lambda_j} = p.$$

So $\arg\max_{a,b,Q,d} \mathcal{L}(X,\theta) = \arg\max_d \sum_{i=1}^{d} \log\lambda_i + (p-d)\log(\frac{1}{p-d}\sum_{i=d+1}^{p}\lambda_i)$. Finally, Jensen's inequality tells us that

$$\mathcal{L}_p(X,\theta) - \mathcal{L}_d(X,\theta) = (p-d)(\log(\frac{1}{p-d}\sum_{i=d+1}^{p}\lambda_i) - \frac{1}{p-d}\sum_{i=d+1}^{p}\log\lambda_i) \geq 0$$

so $\arg\max_{a,b,Q,d} \mathcal{L}(X,\theta) = p-1$. □

The proof of Proposition 2 gives the expression of the likelihood with respect to d:

$$\mathcal{L}_d(X,\theta) = (p-d)(\frac{1}{p-d}\sum_{i=d+1}^{p}\log\lambda_i - \log(\frac{1}{p-d}\sum_{i=d+1}^{p}\lambda_i)) + \text{cst}.$$

The estimation of the dimensions $d_1,\dots,d_K$ is really a central problem in the model, and is very tedious in practice because of the ill-posedness. Furthermore, due to the exponential number of possible choices p^K (with K clusters) for the dimensions of the clusters, we cannot compute all possibilities and then use a certain criterion.

Estimating the d_k is very closely related to the estimation of the noise(s). It can be reasonable to assess that the noise is the same for every cluster. Hence, we could replace the estimation of $d_1,\dots,d_K$ by the estimation of a single noise b, which is likely to be known from a prior assumption. This method has lots of advantages ; it is much more intuitive, it can be known or guessed independently from the algorithm, and can be inferred with a Bayesian criterion as well. This is the idea behind HDMI (Houdard, Bouveyron, and Delon, 2017), presented in Section 1.2.4.

1.1.5 Maximum likelihood denoising

As we shall see later on, modeling the noise is very powerful for denoising purposes, more than trying to reduce it by averaging independent data. It enables then to use estimators like the Maximum Likelihood Estimator (MLE) or the expectation, depending on the goal and the context.

In the case of Gaussian noise, which will be our case of interest, those two estimators are the same and have a close form formula that we will recall here. As we shall see, we even have access to the posterior distribution.

Let X, Y be two Gaussian random variables, with the given probability distribution, $p(x) = \mathcal{N}(x|\mu,\Lambda^{-1})$ and $p(y|x) = \mathcal{N}(y|Ax+b,L^{-1})$.

Then we have

$$\begin{aligned} p(y) &= \mathcal{N}(y|A\mu+b, L^{-1}+A\Lambda^{-1}A^T) \\ p(x|y) &= \mathcal{N}(x|\Sigma(A^TL(y-b)+\Lambda\mu),\Sigma), \end{aligned} \tag{1.6}$$

with $\Sigma = (\Lambda + A^TLA)^{-1}$.

In our case, we have $Y = X+\epsilon$, with $Y \sim \sum_k \pi_k\mathcal{N}(.|\mu_k,\Sigma_k)$ and $\epsilon \sim \mathcal{N}(.|\sigma^2\mathrm{I})$, so

$$X \sim \sum_k \pi_k\mathcal{N}(.|\mu_k,\Sigma_k-\sigma^2\mathrm{I}).$$

So we have $p(x|y) = \sum_z p(x|y,z)p(z|y) = \sum_k p(x|y,z)\tau_k(y)$ using the same notation as in Section 1.1.3. So, using (1.6) for each cluster, with $A=\mathrm{I}$, $L^{-1}=\sigma^2\mathrm{I}$ and $b=0$, we obtain

$$p(x|y) = \sum_k \mathcal{N}(x|\mu_k + \mathbf{\Sigma}_k\frac{y-\mu_k}{\sigma^2}, \mathbf{\Sigma}_k)\tau_k(y) \tag{1.7}$$

with $\mathbf{\Sigma}_k = ((\Sigma_k - \sigma^2 \mathrm{I})^{-1} + \frac{1}{\sigma^2}\mathrm{I})^{-1}$.

Having an observation y of Y, we have then easily access to the expected value of X :

$$\mathbb{E}[X|y] = \sum_k \left(\mu_k + \mathbf{\Sigma}_k \frac{y - \mu_k}{\sigma^2}\right) \tau_k(y), \tag{1.8}$$

and to the maximum a posteriori estimator

$$\mu_{k_m} + \widetilde{\Sigma}_{k_m} \frac{y - \mu_{k_m}}{\sigma^2} \tag{1.9}$$

with $k_m = \arg\max_k \frac{\tau_{k_m}(y)}{\sqrt{\det \widetilde{\Sigma}_{k_m}}}$, which gives in both case an interesting candidate for the denoised value.

1.1.6 Hyperparameters estimation

As we saw in Section 1.1.4, having some hyperparameters (i.e. parameters outside the model) like the intrinsic dimension d in GMM or the number of classes K in a clustering problem helps a lot when they are known thanks to some prior knowledge. However, it is rarely the case, and there is rarely a "unique" good solution and like the other parameters, they must be estimated to reach a decent solution.

Hyperparameters estimation is really crucial and is a complicated issue, since it is in general ill-posed. For example, to choose the number of classes in a clustering context, the "perfect" solution (in term of pure clustering objective) will always be to take one class for each point. This solution is obviously not what we want to achieve. The real goal in hyperparameter selection is to find the "optimal" trade-off between accuracy and complexity, and therefore to find a criteria which leads to such a trade-off. The difficulty becomes then to be able to compare quantitatively accuracy and complexity, which is problematic since they are already hard to measure independently and there is a priori no common scale between the values to which they can be associated.

Still, some methods have been developed to estimate the hyperparameters, trying to take advantages of an underlying Bayesian framework. The accuracy is then measured by the posterior probability of the observation and the complexity is an (increasing) function of the number of parameters. This idea have led to the so-called penalization criterions, which are the most popular hyperparameters estimation tools. Some other heuristics exist, like the popular slope heuristic, presented in detail in Baudry, Maugis, and Michel, 2012 and can offer some nice alternative.

On general penalization criteria

The idea of penalization criteria is to add a penalization term to the likelihood of the model to be able to compare different model with respect to a certain point of view. Obviously, as the number of parameters grows, the likelihood is supposed to grow as well since the model will be able to better fit the data. This leads to two main drawbacks, first, an over-fitting of the training data, and secondly that we may overparameterize the model. This is typically the case in clustering, where the correct number of clusters is the "lowest possible". Otherwise, the best clustering would be to consider each point as a single cluster.

This lead to the idea of penalizing the likelihood with a quantity which depends on the numbers of parameters $\mathcal{K}$. The criterion is then defined as $crit(\mathcal{M}, \mathcal{K}) =$

$\log(\theta^{ML}) - pen(\mathcal{K})$. We then select the best $\mathcal{K}$ as

$$\mathcal{K} = \underset{\mathcal{K} \in [\mathcal{K}_{min}..\mathcal{K}_{max}]}{\arg\max} \; crit(\mathcal{K})$$

.

This idea is pretty similar to adding a prior to a hyper parameter, like an exponential law and try to compute its maximum likelihood. But the Bayesian framework is not very convenient in this context, because we rarely have non-heuristical prior on the hyperparameters and mainly this "hyperprior" would introduce another hyperparameter. Besides, the penalization term can only depend on the value of one of the hyper parameters, we cannot choose between models differing by more than one hyper parameter.

Several penalization criteria have between studied in the literature. General results have been proved by Nishii (1988). He considered penalization of the form $pen(K) = c_n \times K$, with K the complexity of the model, and n the number of input data. He proved that if the true distribution does not belong to the considered model, then :

- If $c_n = o(n)$, then the criterion cannot underestimate the complexity of the model
- If $\frac{c_n}{\log\log n} \to \infty$, then the criterion is strongly consistent, which means that the chosen model will converge towards the simplest model among those who minimize the KL-divergence to the true distribution.

More details can be found in Baudry (2009). These kinds of results can give hope that these criterions will work, but there are still not very precise and strong.

Even if these criterions have been largely studied, most of the work remains experimental, and to my knowledge, no complete review have been made on the subject. The rest of the section exposes the main facts on the considered criterion, with some references for further precision.

AIC

Criterion The Aikake Information Criterion (AIC) was one of the first selection model criterion, introduced first by Akaike (1973). The penalization is simply the dimension of the model. Thus, the criterion is

$$AIC = \log p(x|\theta^{ML}) - \mathcal{K}$$

This correspond to $c_n = 1$ in the Nishii framework, which doesn't guaranty the consistence of the criterion, and Nishii (1988) showed that AIC can be inconsistent.

Motivation The idea behind it is to consider the empirical likelihood $\frac{1}{N}\log p(x|\theta^{ML})$ as an estimator of $\mathbb{E}[\log p(.|\theta^{ML})] = -KL(p||p(.|\theta^{ML})) + cst$, which is asymptotically correct thanks to the law of large numbers. Hence, maximizing the empirical likelihood enables to minimize the KL-divergence between the true distribution and the estimated one. However, the same data are used to compute θ^{ML} and to estimate $\mathbb{E}[\log(p(.|\theta^{ML})]$, a bias is therefore induced. It is shown in Akaike (1973) that this bias could be asymptotically estimated by the dimension of the parameter space, which makes AIC an estimator of $\mathbb{E}[\log p(.|\theta^{ML})]$.

Properties The AIC tries to find the closest distribution to the real one in the sense of the KL divergence, which is already intuitively the kind of behavior one could have expected of a criterion. Furthermore, it has some other very important properties, studied by Yang (2005). The AIC criterion is not consistent in general, but is asymptotically optimal under the average square error loss. It means that under some conditions, if the true distribution does not belong to the models, then the model selected by AIC will be the "best" model in terms of average square error loss.

BIC

Criterion The Bayesian Information Criterion was introduced by Schwarz (1978). The penalization is stronger than the AIC criterion :

$$BIC = \log p(x|\theta^{ML}) - \frac{1}{2}\mathcal{K}\log n$$

with n the number of points. The BIC tries to select the model with the highest posterior model probability.

Motivation The idea of the BIC is to estimate the marginalized likelihood of the model : $\int_\theta p(X,\theta|\mathcal{M})$ with the Laplace approximation. Let x_0 be a mode of the likelihood, then we have $\log \mathcal{L}(x,\theta) = \log \mathcal{L}(x,\theta^{ML}) - \frac{1}{2}(\theta-\theta^{ML})^T(-\nabla^2 \log \mathcal{L})_{|\theta=\theta^{ML}}(\theta-\theta^{ML}) + o\left((\theta-\theta^{ML})^2\right)$. We can therefore approximate the distribution near its mode with a Gaussian distribution. The Laplace approximation consists in generalizing this approximation to the whole space, giving an approximation for the integral :

$$\log \int_\theta p(X,\theta|\mathcal{M}) \approx \log p(x|\theta^{ML}) + \log \int_\theta e^{-\frac{1}{2}(\theta-\theta^{ML})^T(-\nabla^2 \log \mathcal{L})_{|\theta=\theta^{ML}}(\theta-\theta^{ML})}$$

This can be very efficient in practice if the distribution is concentrated around its mode, but completely false in general, especially if the distribution has several modes. It has been shown by Schwarz, 1978 that this integral could be approximated by $-\frac{1}{2}\mathcal{K}\log n$.

Properties The main result on BIC is its consistency. If the true distribution belongs to the model, then the probability of selecting the right model by BIC will converge to 1 as $n \to \infty$. See Yang (2005) for further details.

ICL

Criterion The ICL was introduced by Biernacki, Celeux, and Govaert (2000) in order to mimic the derivation of the BIC in a clustering purpose. We have

$$ICL = \log p(x|\theta^{ML}) - \frac{1}{2}\mathcal{K}\log n - ENT(x,\theta)$$

with $ENT(x,\theta) = -\sum_n \sum_k \tau_{i,k}(\theta^{ML}) \log \tau_{i,k}(\theta^{ML})$ is the entropy. The idea behind it is that, in a clustering framework or to avoid overfitting, we don't really want to be close to the distribution, but rather aim at finding the right number of cluster. More details can be found in Baudry (2009).

Motivation The ICL criterion is specific to mixture model, and its value, like for the BIC, comes from the Laplace approximation, but applied to the classification likelihood, in which Z is supposed to be known (in contrast with BIC where its expected value is considered):

$$\log \mathcal{L}_c(x, z|\theta) = \sum_n \sum_k z_{n,k} \log(\pi_k \mathcal{N}(X|\mu_k, \Sigma_k)) = \log \mathcal{L}(x|\theta) + \sum_n \sum_k z_{n,k} log(\tau_k(x, \theta)),$$

which gives in average $ICL = BIC - ENT(x, \theta)$.

Properties This criterion was introduced to be applied on a classification problem, with the idea that some of the components we wish to obtain could not precisely fit the model, and could need several classes for only one component. This explains why BIC can tend to overestimate the number of components, because it tries too much to fit to the distribution. On the other hand, ICL penalizes similar classes, and therefore seems more adapted to clustering purpose. See Baudry (2009) for more details.

Discussion

Figure 1.4 presents a comparison of the different penalization criterions on the noise estimation problem of an image and the number of cluster estimation using an algorithm based on HDDC. The context of this experiment will be better explained in Section 1.2.4. For now, we can just consider the noise σ and the number of clusters as hyperparameters. As we see, the penalization criterions are quite efficient to estimate the noise, on which they almost all agree. It is interesting to note that the noise, even if considered as a parameter, corresponds to a real quantitative value, in opposition to the number of classes, which is way more subjective and has a more subtle impact on the efficiency of the model.

Hyperparameters estimation is a very appealing problem. As shown in Figure 1.4, it can be an efficient tool to tune hyperparameters in an automatic or semi-automatic fashion. However, their main drawback is that their computation implies doing a dichotomy on the value of the hyperparameter, which involves inferring a new model each time and is not feasible in practice when they are a bit elaborated. In the application that we shall make of the presented models, their use would not be relevant, that is why we won't use them in the rest of this thesis.

1.2 Image denoising

In the field of image processing, lots of challenges have arised in the last decades, with the explosion of digital images. Among them, one of most basic and fundamental is the problem of image restoration. Unlike analogical images (argentic cameras), after the capture of an image by a digital device (camera, scanner, IRM, ...), the picture is rarely ready to be used, and has to be post-processed by algorithms that *improve* its quality, based on what is known and expected of the result. This is the main difficulty of image restoration: the algorithms have to *guess* what can improve the image, relying on their objective and/or every prior knowledge on the nature of the degradation. As a matter of fact, most problems of image restoration are ill-posed. Among them, the denoising problem has a particular place : it is one of the simplest since the signal is only degraded by addition of noise and still remains fundamental.

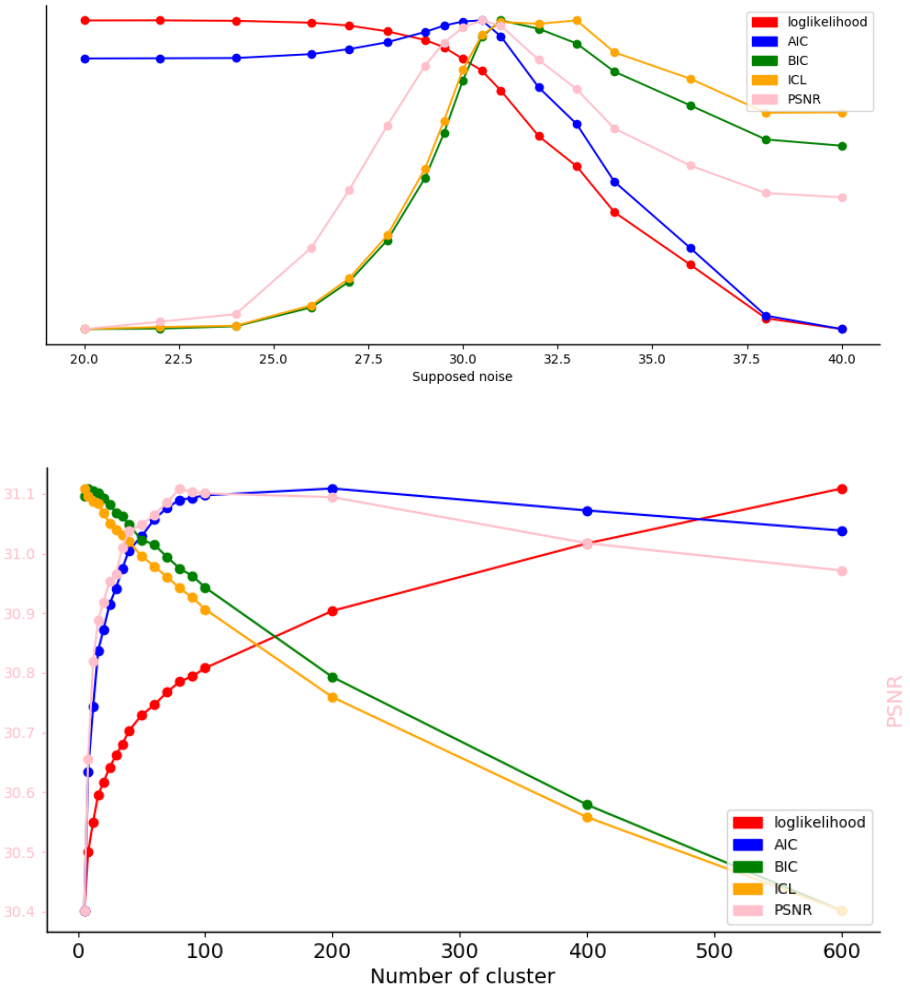


FIGURE 1.4: Comparison of all the presented criterions for the noise estimation (related to the instrisic dimension) and the number of classes for the inference of an HDDC model on the patches of an image with artificial white Gaussian noise of standard deviation $\sigma = 30$. See Section 1.2.4 for further details.

In the following, we will only consider this problem on natural images, obtained from camera, but everything that we will discuss could be extended to wider applications.

1.2.1 The denoising problem

The denoising problem is posed as followed: given an image with noise, be able to estimate it and recover the image without the noise. The assumption behind this problem is that it exists a genuine noise-free image u, and that we observe the image

$$\widehat{u} = u + \epsilon,$$

where ϵ is the noise that we can model depending on the context of the problem.

This problem is completely ill-posed, since theoretically, all images are possible, even those with noise. A photograph of a *noisy* image of a scene becomes a natural image once it is captured. There are no way that one could hope to be able to solve the problem without any further assumption, and the way of solving it has to be related to the purpose behind it. For instance, in the field of medical imaging, the denoised image must help the diagnostic, which gives already some contraints and ideas to work with. In the case of photography, we want to obtain a nice-looking image that our eyes could have observed, corresponding to what we call the *visual quality*. Even if it is the ultimate criterion, it is highly subjective and does not help formalizing the problem.

Yet, in most cases, what we really want to achieve is to have nice looking images. The most popular way to measure and compare the efficiency of a denoising algorithm is the PSNR (Peak Signal to Noise Ratio) defined by

$$\text{psnr}(\widetilde{u}, u) = 10 \times \log_{10}\left(\frac{d^2}{MSE(\widetilde{u}, u)}\right)$$

with MSE the mean square error: $MSE(u_1, u_2) = \frac{1}{|u_1|}\sum_{pixel\ p}(u_1(p) - u_2(p))^2$ and d the dynamic of the signal (255 in the standard cases). It is equivalent to the MSE (or the $RMSE$ as sometimes used (for Root Mean Square Error)) but enables to measure with decibels in a increasing scale: the higher the PSNR, the better the result.

In order to test an algorithm, one has to artificially add noise to a test image, considered to be noise-free, and see *how far* the result of the algorithm is to the ground truth, thanks to the PSNR. This measurement is not really justified and far to be optimal, but it remains really simple. Some works have tried to improve the criterium, trying to better mathematically catpure the notion of visual quality. It is the case of the structural similarity criterion (SSIM), introduced by Wang et al., 2004. This criterium became quite popular but remains a second choice after the PSNR due to its complexity and the lack of clarity of its added value. In particular, Hore and Ziou, 2010 compared it to the PSNR, and showed that they are somewhat related.

The key challenge in image denoising is to understand and express mathematically what makes an image look *natural*. The first observation we can make is that natural images are somewhat piecewise smooth, as they are most of the time a combination of objects. As a first approximation, we can suppose that they are composed of constant zones delimited by the contours. A first idea is therefore to impose regularization conditions on the noisy image while trying to preserve as much as possible the contour, and then hope that is will lead back to the natural image. This have led to the so-called varational methods, which were first very popular in the 90's, see for

instance Rudin, Osher, and Fatemi, 1992. However, this assumption is very limited, the smoothness of a natural image is more subtle than piecewise constantness. This idea doesn't take into account many particularities of images, like the textures for instance, which are very crucial in natural images.

A second idea for denoising algorithms is the so-called "self-similarity" principle, which states that an image is made of the complicated combination of a limited set of small and simple patterns. This idea have led to patch based methods that we shall develop in Section 1.2.3.

A third type of methods comes from the recent developpement of neural networks, which became very efficient in image denoising. They can be seen as a very powerful and automatic way to make use of the two previous ideas. Some neural networks are trained on patches, like Zhang et al., 2017 for instance. The real strength of neural network is that they can handle different level of noise, as presented in Wang and Morel, 2014, Zhang, Zuo, and Zhang, 2018 and Islam et al., 2018. Their drawbacks are the following : they need a huge training data set and they do not help better understanding and improving the concept behind the image denoising problem. Some work like Soltanayev and Chun, 2018 have been made to tackle the issue of the size of the data set.

1.2.2 Denoising generalities

We refer to the review of Lebrun et al., 2012 for more details on the idea developped in this section.

About the noise

The presence of noise is unavoidable in photography. In a numerical camera, pixels of the image are captured by small sensors that are never perfectly similar and accurate. This always adds some noise to the picture. Besides, the quantum nature of light itself adds noise in any situation. For standard setting on a personal camera, this phenomenom is not quite visible, but as soon as the luminosity is low and/or the shutter time is fast, images become very noisy. We can even observe noise with our own eyes in the dark (even if our brain is used to ignore it).

Hence, the noise in cameras has most of time the same origin and can pretty acurately be modeled at each pixel using a i.i.d. Poisson distribution. For large enough values of the expected value (i.e. not excessively low luminosity), this can accurately be approximated by a Gaussian distribution : $\mathcal{N}(u(i), u(i))$. However, in this approximation, the variance still depends on the signal. In order to get the white Gaussian noise, way more convenient to work with, one can apply a Variance Stabilizing Transformation (VST) to the signal, process the image, and then apply an inverse VST to get the final result. More details on this can be found in Makitalo and Foi, 2010. Hence, it is relevant, for practical applications, to assume that the noise is a white Gaussian noise.

The noise model mainly depends of the context of the scene and the quality of the camera. It is therefore pretty accurate to assume that it is the same for the whole image. Hence, the only remaining uncertainity to model properly the noise is the variance, which will differ on every input. There are several ways to estimate the variance of the noise from a single image that we will not discuss here, but we can therefore assume that the noise variance is known. More details can be found in Colom, Buades, and Morel, 2014.

Eventually, most images nowadays are color images. Since the captors of the different color channel are very similar, we can still suppose that the noise on each channels is added in a similar fashion. A first idea to process a color image is to denoise each channel independently. However, this does not work very well, since it creates color artifacts to which the eye is very sensible. Fortunately, the RGB data is not the only way to store an image, and it appears that this idea works fine with other representations, like YUV, which can be obtained from an RGB image by a multiplication by the following matrix:

$$YUV = \begin{pmatrix} 0.30 & 0.59 & 0.11 \\ -0.15 & -0.29 & 0.44 \\ 0.61 & -0.51 & -0.10 \end{pmatrix} \text{ or } Y_0U_0V_0 = \begin{pmatrix} \frac{1}{3} & \frac{1}{3} & \frac{1}{3} \\ \frac{1}{2} & 0 & -\frac{1}{2} \\ \frac{1}{4} & -\frac{1}{2} & \frac{1}{4} \end{pmatrix}.$$

Therefore, an algorithm on black and white images can directly be applied to color images.

In the rest of the thesis, we will therefore consider only black and white images with a white Gaussian noise whose variance is known.

Main denoising principles

As we said in Section 1.2.1, the main challenge of denoising problems is to find a way to capture the subtle regularity of natural images without destroying the textures. Most of the denoising algorithms rely on these 3 main ideas :

- The *self-similarity* principle, which states that a natural image is a combination of similar lower-scale patterns. This idea has led to the so-called patch-based methods, discussed in detail in Section 1.2.3.
- Sparsity of patches in a fixed basis, which has led to transform thresholding methods (based of the thresholding of the signal in a sparse representation). This is the same assumption as the one behind the famous JPEG compression, that, in a proper basis, the image can be sparsly represented. However, these methods often create ringing phenomena.
- Sparsity on a learned dictionary. Instead of assuming that the image is sparse in an universal basis, the idea is to learn a dictionary which will represent sparsely the image. These methods lead to optimization problem with sparsity constraints, for example finding $\widetilde{\alpha} = \arg\min_\alpha ||\alpha||_0$ such that $||\widehat{u} - D\alpha||^2 \leq \lambda$ where D is the dictionary, and λ a data-fidelity parameter.

Denoising tools

In addition to the main ideas and mathematics behind a denoising algorithm, there are other "meta" ideas that can be plugged to any method in order to slightly improve the results. Those are useful when one tries to reach the highest possible performance, but can prevent from developing new ideas and from identifying the flaws of a method. We will present some of the most popular ones here, but will not use them in the rest of this thesis.

- *Aggregation of estimates* : if we have m "independent" different estimate, we can try to average them to decrease the variance. For example, we can denoise an image by averaging the output of all the algorithms presented in Section 1.2.4.

- *Iteration/oracle filters* : The idea is to first denoise roughly the input image using any algorithm, and then use this result as an oracle to help the main algorithm to perform its task (for clustering for instance). For instance, in NL-Bayes, this idea helps greatly to detect similar patches, and then to average more efficiently the noisy ones (see Section 1.2.4).

- *Twicing* : After having a first result with a denoising algorithm, one can compute the residual noise with a simple subtraction to the original input. There are often lots of information left in the estimation, and another algorithm can be applied to this residual to better estimate the noise.

- *Multiscale algorithm* : One of the difficulty faced by most of denoising algorithms is the low-frequency noise which is harder to differentiate from the *true* image than the high-frequency one. A way to solve this issue is to denoise the image at different resolutions. Starting at the smaller one (where it is way easier since the noise is highly reduced by the scaling), we can then use each result as a basis for the next resolution until the one of the input image.

- *Constant zone enforcement* : The eye is very sensitive to small variations in a constant region, therefore it can be useful, inside an algorithm, to detect these constant zones (if the variance of a patch is below a certain threshold for instance) and apply a different treatment to these patches.

1.2.3 Patch-based methods

Patch based methods were first introduced in the end of the 20^{th} century, have led to a new paradigm in image processing, and were applied to various image processing problem such as inpainting (Wexler, Shechtman, and Irani, 2007, Newson et al., 2014, Criminisi, Pérez, and Toyama, 2004), image synthesis (Efros and Leung, 1999, Kwatra et al., 2005), denoising (Buades, Coll, and Morel (2005)) or editing (Barnes et al., 2009, Frigo et al., 2016), improving the state of the art. The non local methods have been largely studied and improved, with some variants (Kervrann and Boulanger, 2006), adaptation to other type of noise (Deledalle, Tupin, and Denis, 2010) and extensions to more complex inverse problems (Peyré, Bougleux, and Cohen, 2008). Concerning the denoising, most of the state of the art methods rely nowadays on probabilistic models, such as NL-Bayes (Lebrun, Buades, and Morel (2013)), or in the works of Yang (2005), Zoran and Weiss (2011), and Wang and Morel (2013). These works have led the state of the art until recently where they were overcome by neural networks. However, patch-based methods are still interesting, as they don't need data set to train and can perform image-based denoising, they are better understood theoretically.

A patch, as illustrated Figure 1.5 is a small piece of image (in practice, between 3×3 and 25×25). In our context, it should be of the size where the self-similarity principle applies. We focus here on the case of images, which are 2D-signals, but all the concepts presented hereafter are much more general and can be defined for signals with any number of dimensions. As presented in Figure 1.6, there are 3 main steps in patch-based signal processing methods :

1. Patch extraction : divide the image into patches

2. Patch editing : perform learning, computation and restoration of the patches from step 1.

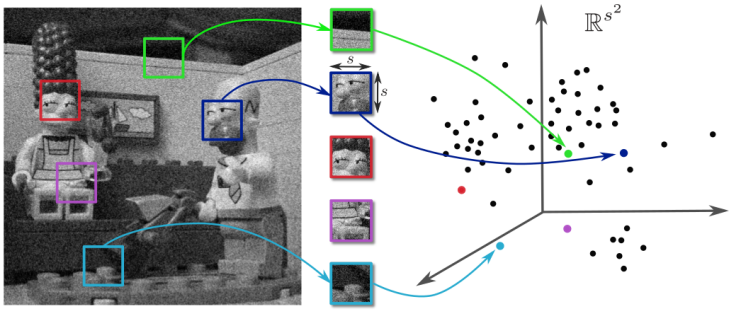


FIGURE 1.5: Illustration of the patch transformation of an image (image from Houdard, Bouveyron, and Delon, 2017)

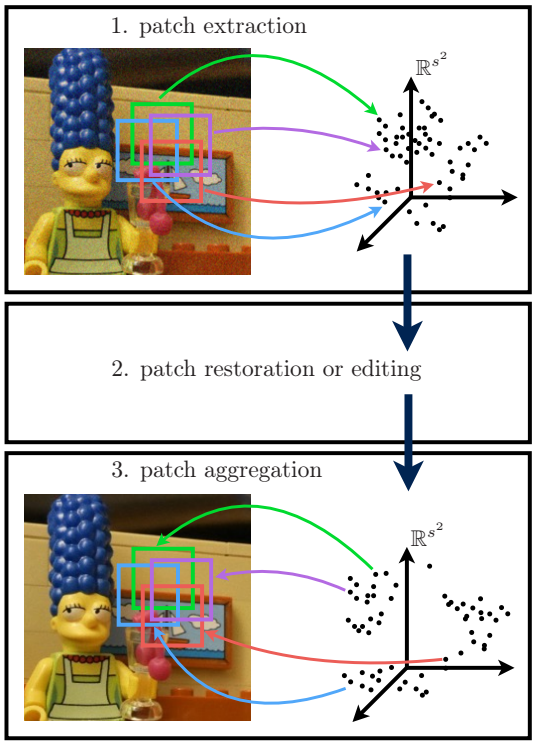


FIGURE 1.6: Illustration of the 3 steps of patch-based image processing. Patches are small image pieces, they can be seen as vectors of a high dimensional space. Patch-based methods decompose images in overlapping patches (step 1) and make these patches collaborate for restoration, synthesis or editing purposes (step 2). At this point, the processed overlapping patches do not necessarily share the same values on their common pixels. Aggregation techniques aim at combining all these different overlapping patches into a single image (step 3).

3. Patch aggregation : reform the image from the restored patches from step 2.

We will present in the following these steps more in details. Let $\mathbf{\Omega}$ be a discrete rectangular grid of size $s_x \times s_y$ in $\mathbb{R}^2$ and let $u : \mathbf{\Omega} \to \mathbb{R}$ be a grey level image on $\mathbf{\Omega}$.

Patch extraction

A patch of an image u can be written as a sub-image $u_{|\Omega} \in \mathbb{R}^{\Omega}$ where $\Omega \subset \mathbf{\Omega}$ is the domain of the patch. The number of pixels $|\Omega|$ is called the *size* of the patch. Patches usually considered in the literature have connected domains. For example, if $\Omega = [\![1, \sqrt{d}]\!]^2$, $u_{|\Omega}$ is the square patch of size d at the top left corner of u. The image u can be considered itself as a (large) patch of size $s_x \times s_y$, and contains $N = (s_x - \sqrt{d} + 1) \times (s_y - \sqrt{d} + 1)$ overlapping square patches of size d.

The *patch extraction* is characterized by an extraction operator χ which gives a set of patches from an image. In most applications, this set is composed of all overlapping square patches of size d of u. Assimilating u to a $s_x \times s_y$ matrix and patches as vectors of size d (read column-wise), this extraction operator can for instance be written

$$\chi : \mathbb{R}^{s_x \times s_y} \to \mathbb{R}^{d \times N},$$

where the i^{th} column y_i of the matrix $\chi(u)$ is the i^{th} patch of u. Since χ is a linear operator, $\mathrm{Im}(\chi)$ is a linear subspace of $\mathbb{R}^{d \times N}$ of dimension $s_x \times s_y$ at most. Therefore, $\mathrm{Im}(\chi) \neq \mathbb{R}^{d \times N}$. For commodity, we shall write $\chi = (\chi_i)_{i \in I}$ with $(\chi_i)_{i \in I}$ the set of linear operators such that $\chi_i(u)$ refers to the i^{th} patch.

In the general case, χ is not surjective, and an element of $\mathrm{Im}(\chi)$ has lots of redundancies, since each pixel may appear in many different patches.

Patch-based editing or restoration

Given an extraction operator χ returning a set of N patches, patch-based signal processing consists in processing the set of patches $\chi(u)$ instead of the signal u. However, after this processing, the set of patches is usually not in $\mathrm{Im}(\chi)$ anymore. It means that a pixel which belongs to several patches can have different values in all these processed patches.

For example, in image restoration, we have access to $\widehat{u}$, a distorted version of the true signal u. In order to construct $\widetilde{u}$, an estimate of u, we first extract the patches $(\widehat{y}_i)_{i \in [\![1,N]\!]} := \chi(\widehat{u})$ and we try to infer their restored versions $\widehat{y}_i$. For instance, in a Bayesian framework, if we have access to a posterior probability distribution for each patch, we can estimate each $\widehat{y}_i$ by

$$\widetilde{y}_i = \arg\max_y p(y|\widehat{y}_i).$$

However, after this estimation, there is no guaranty that $(\widetilde{y}_i)_{i \in [\![1,N]\!]} \in \mathrm{Im}(\chi)$, *i.e.* no guaranty that we can find $\widetilde{u}$ such that $\chi(\widetilde{u}) = (\widetilde{y}_i)_{i \in [\![1,N]\!]}$.

Patch aggregation

The patch aggregation is the action of recovering an image from a set of patches. It is characterized by an aggregation operator ξ, which reconstructs an image from a set of patches. Most of the time, it satisfies $\xi \circ \chi = \mathrm{I}_{s_x \times s_y}$, but it is not mandatory.

If χ extracts all N overlapping square patches of size d from u, ξ can be seen as a map from $\mathbb{R}^{d \times N}$ to $\mathbb{R}^{s_x \times s_y}$. In this case, the most common aggregations are *the central*

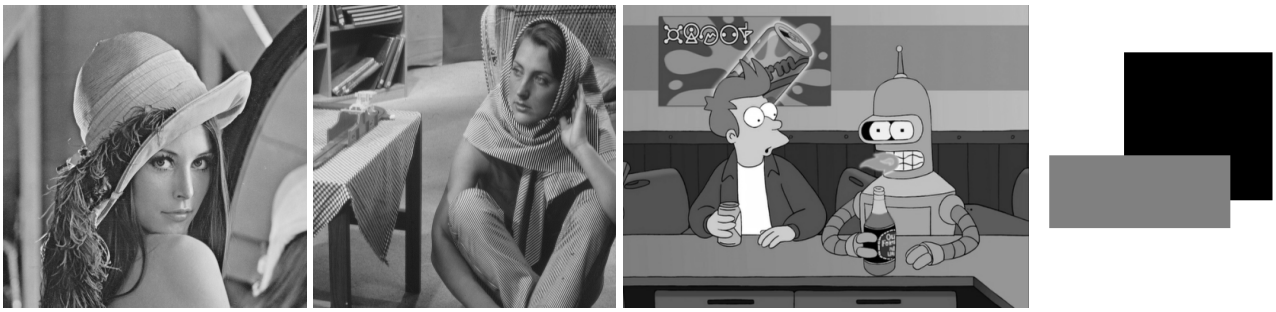


FIGURE 1.7: Test images, from left to right: *Lena*, *Barbara*, *Cartoon*, *Squares*.

pixel aggregation, which consists in keeping only the central pixel of each patch and *the uniform (resp. weighted) aggregation* which consists in taking, at each pixel, the uniform (resp. weighted) average of the d different values provided by the patches it belongs to.

1.2.4 Detail on some patch-based algorithms

The framework of patch based methods presented in Section 1.2.3 is very general and has led to a large variety of algorithms in the literature. They sometimes deviate a bit from this formalization (like in Section 1.2.4, where the two last steps are merged), but they always contain in one way or another these three steps. We will present in this section some particular patch-based algorithms, with their results on a bench of images presented on Figure 1.7 and for three different noise standard variation: $\sigma = 10, \sigma = 30$ and $\sigma = 50$.

NL-Means and NL-Bayes

NL-Means, for Non-Local Means, was introduced in 2005 by Buades, Coll, and Morel, 2005. It is considered to be the first patch-based denoising algorithm. At the time, this algorithm stroke by its simplicity and its obvious potential and yet its good PSNR results.

The idea of the algorithm is to first extract all the overlapping square patches of size k (the author found best results by setting $k = 3$) and then for each patch, average it with all the patches in a search window (used for computational time purposes) weighted by the (minus exponential of the) square Euclidean distance to the considered patch. The idea is that we perform a noise reduction by averaging each patch with its most similar ones. The complete process is detailed in Algorithm 2.

NL-means gives quite good PSNR results because it works well in constant zones. Even if the authors studied and proved some consistency results (see Buades et al 2006), its strategy remains basic. It implicitly assumes that the noise is Gaussian but does not use any model on the patches which, as we shall see, limits the restoration.

A direct extension of NL-means is NL-Bayes (introduced by Lebrun, Buades, and Morel, 2013), which keeps almost the same idea, simply adding a model to the patches coherent with the one of the noise. It is presented in Algorithm 3. Instead of averaging directly the patch of the search window, the algorithm infers a Gaussian model with all the "close" patches and uses this model to compute the most likely restored patch. The authors also adds some general denoising tricks, in particular an oracle step and a constant zone enforcement (see Section 1.2.2) to improve the

Algorithm 2 NL-means

Input: Noisy image $\widehat{u}$, standard deviation σ
Output: Denoised image $\widetilde{u}$

Set parameter size of the patch : k = 3
Set parameter size of the search window : $\lambda = 31$
Set parameter bandwith filtering parameter : $C = 0.6\sigma$
for each i pixel of the image **do**
    P : patch of size $k \times k$ whom i is the center
    for each Q in the search window of size $\lambda \times \lambda$ **do**
        $d(P,Q)^2 \leftarrow \frac{1}{k^2}||P - Q||^2$
        $w_Q \leftarrow \exp(-\frac{d(P,Q)^2}{C^2})$
    end for
    $\widetilde{P} \leftarrow \frac{\sum_Q w_Q \times Q}{\sum_Q w_Q}$
end for
for each pixel i **do**
    Set $\widetilde{i}$ as the average of all the patches whome it belongs
end for

performances of the algorithm. NL-Bayes remains basic: the model is simple and the use of the l_2 norm to compare similar patches is not ideal (a patch almost identical but with a slight shift is considered far although it is similar for our eyes), but its results are way better than those of NL-means: it shows how powerful and crucial it is to have a model on the patches and by extension on the image. The visual results of these two algorithms are presented on Figure 1.8 and Figure 1.9.

HDMI

NL-means has shown that patch-based methods have potential, and NL-Bayes that they need a model to be efficient. In the recent years, lots of powerful statistical tools have been developed to process data and can almost directly be applied to the patches. This new paradigm have led to many methods in the denoising literature. The idea is to consider the whole set of patches of the image (or a transformation of the image, like the Fourier transform) as a single data set and infer a single model on it. Introduced by Houdard, Bouveyron, and Delon, 2017, HDMI is one of the most recent and efficient one, and makes use of HDDC (see Section 1.1.4).

Even if NL-means uses very small patches (3×3), they can be taken larger (10×10 for instance). Considering all overlapping patches of an image, this constitutes a large data set of reasonable dimension (compared to the image) which remains high for statistical models. This makes HDDC particularly useful, as it is adapted to high dimensional data. Besides, the underlying assumptions of the model are besides justified in the case of patch data :

- The noise is the same in all patches, and therefore in all the clusters
- The patches inside a cluster live in a lower dimensional space: real world images seem to satisfy this criterion, as demonstrated by the success of patch based and of dictionary method in denoising.
- This model uses a lot of information redundancy, which fits the "self-similarity" principle.

Algorithm 3 NL-bayes

Input: Noisy image $\widehat{u}$, standard deviation σ
Output: Denoised image $\widetilde{u}$

Set parameter size of the patch k, Set parameter size of the search window λ, Set parameter close patches C (see Buades, Lebrun, and Morel, 2012)

for each i pixel of the image **do** ▷ First loop to compute the oracle
  P : patch of size $k \times k$ whom i is the center
  $\mathcal{P}(P) \leftarrow \{Q; ||Q - P|| \leq C\}$
  Compute C_P and $\overline{P}$ with
  $C_P = \frac{1}{|\mathcal{P}(P)|-1} \sum_{Q \in \mathcal{P}(P)} (Q - \overline{P})(Q - \overline{P})^T$ and $\overline{P} = \frac{1}{|\mathcal{P}(P)|} \sum_{Q \in \mathcal{P}(P)} Q$
  $\widetilde{P}^1 \leftarrow \overline{P} + (C_P - \sigma^2 \widehat{u}) C_P^{-1} (P - \overline{P})$
end for
Obtain a first denoised version $\widetilde{u}^1$ by averaging all the patches

for each i pixel of the image **do** ▷ Second loop to denoise the image
  P : patch of size $k \times k$ whom i is the center extracted from the noisy image, and $\widetilde{P}^1$ the corresponding patch in $\widetilde{u}^1$
  $\mathcal{P}(P) \leftarrow \{Q$ patch of $\widehat{u}$; such as the corresponding $\widetilde{Q}^1$ in $\widetilde{u}^1$ verifies $||\widetilde{Q}^1 - \widetilde{P}^1|| \leq C\}$ and $\widetilde{\mathcal{P}}(P) \leftarrow \{\hat{Q}^1$ patch of $\widetilde{u}^1; ||\widetilde{Q}^1 - \widetilde{P}^1|| \leq C\}$
  Compute $\widetilde{C_P}$ and $\widetilde{\overline{P}}$ with
  $\widetilde{C_P} = \frac{1}{|\widetilde{\mathcal{P}}(P)|-1} \sum_{\widetilde{Q} \in \widetilde{\mathcal{P}}(P)} (\widetilde{Q} - \widetilde{\overline{P}})(\widetilde{Q} - \widetilde{\overline{P}})^T$ and $\widetilde{\overline{P}} = \frac{1}{|\widetilde{\mathcal{P}}(P)|} \sum_{\widetilde{Q} \in \mathcal{P}(P)} \widetilde{Q}$
  $\widetilde{P}^2 \leftarrow \widetilde{\overline{P}} + (\widetilde{C_P} - \sigma^2 \widehat{u}) \widetilde{C_P}^{-1} (P - \widetilde{\overline{P}})$
end for
Obtain the final denoised version $\widetilde{u}$ by averaging all the patches $\widetilde{P}^2$

FIGURE 1.8: Experimental results of NL-Means on the test images. The parameters used the standard parameters recommended by the authors and explicited in Alogrithm 2.

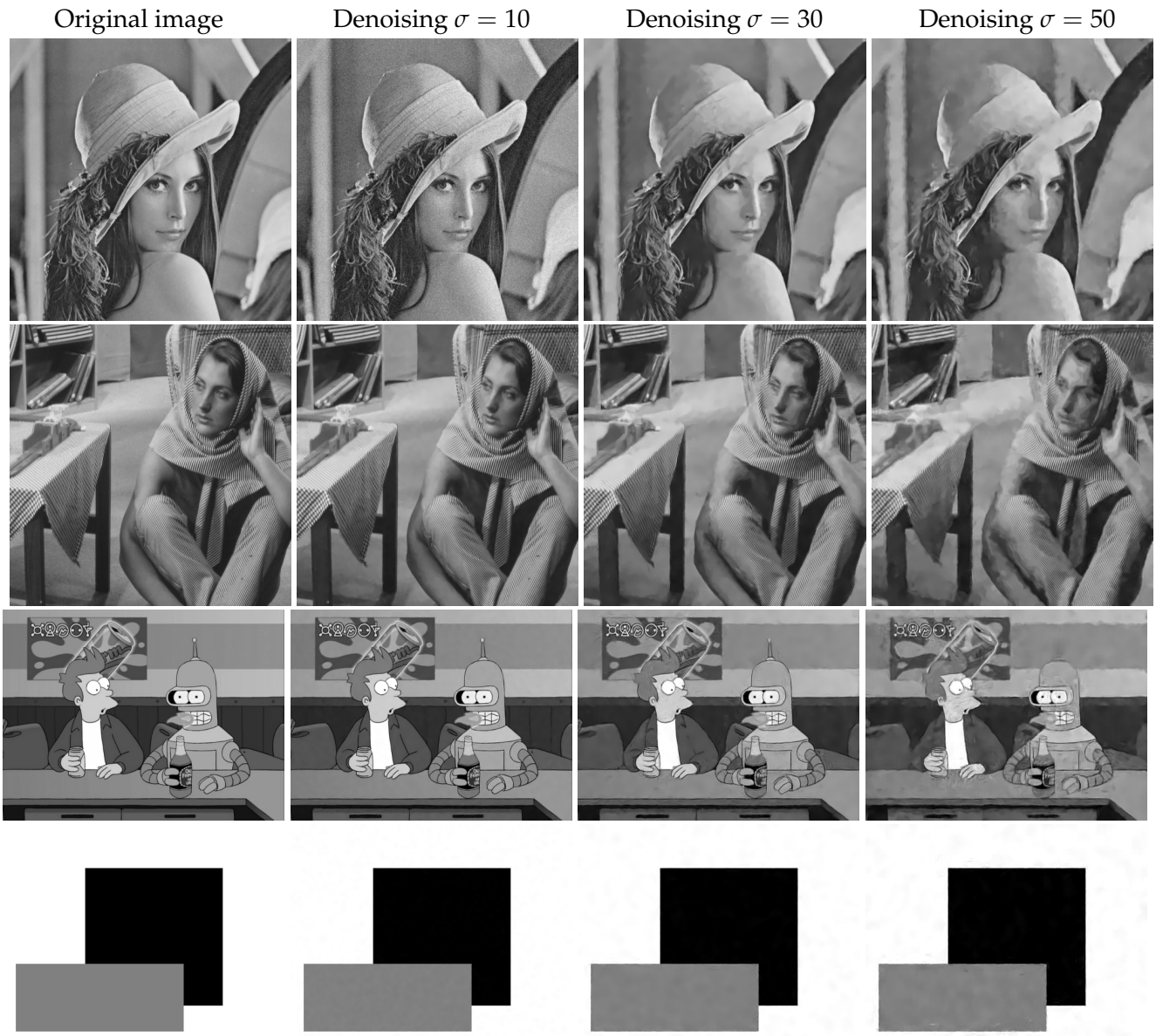


FIGURE 1.9: Experimental results of NL-Bayes on the test images. The parameters choice follows the recommandation in Buades, Lebrun, and Morel, 2012.

In practice, HDMI follows the 3 steps of patch-based methods presented in Section 1.2.3, using HDDC to infer and restore the patches in the patch edition step. However, the estimation of the intrinsic dimension d_i of the cluster is a capital problem. But, as we said in Section 1.1.4, the noise is closely related to the intrinsic dimensions and it is here supposed to be known. So the dimension of each cluster is chosen in order to fit the value of the noise with the one implied by the choice of the intrinsic dimensions of the clusters $d_1, \dots, d_K$. We have therefore this formula:

$$\forall k \in [\![1, K]\!], d_k = \arg\min_{\delta}\left(\frac{1}{p-\delta}\sum_{i=\delta+1}^{p}\lambda_i^{(k)} - \sigma\right). \tag{1.10}$$

The complete algorithm is presented in Algorithm 4 and its visual results are presented on Figure 1.10.

Algorithm 4 HDMI

Input: Noisy image $\widehat{u}$, standard deviation σ
Output: Denoised image $\widetilde{u}$

1: Extract the patches of I
2: Learn HDDC using (1.10) and estimate the dimension
3: **for** each noisy patch P **do**
4: Find the cluster k whose P belongs
5: $\widetilde{P} \leftarrow \sum_k \pi_k \left(\mu_k + Q_k(\mathrm{I} - \sigma^2\Delta_k^{-1})Q_k^t(P - \mu_k)\right)$, where (π, μ, Q, Δ) are the inferred parameters of HDDC (see Equation (1.8)).
6: **end for**
7: Average all patches $\widetilde{P}$ to obtain the denoised image $\widetilde{u}$

EPLL

Until now, we only saw methods using the so-called standard or uniform aggregation. More complex strategies including both patch restoration and aggregation into a single variational formulation have also been considered in the literature. This idea was first introduced by Zoran and Weiss, 2011 with the Expected Patch Log Likelihood (EPLL) and remains, to our knowledge, largely unexploited. Similarly to NL-Means, they introduced this new strategy into a basic framework, showing the potential of it. Some work has extended their algorithm, for instance addind a multiscale framework like Papyan and Elad, 2015, or using more general prior models like Deledalle, Parameswaran, and Nguyen, 2018, but these extensions remained focused on the edition step and have left the aggregation step untouched, although it was one of the main innovation of EPLL.

In details, starting from a noisy image $\widehat{u}$, the authors reconstruct a restored version of u as one of the solutions of

$$\arg\min_{u} \frac{\lambda}{2\sigma^2}||\widehat{u} - u||^2 - EPLL_f(u), \tag{1.11}$$

where $EPLL_f(u) = \sum_j \log f(\chi_i(u))$, with χ_i the patch extraction operators presented in Section 1.2.3 and f a given prior density on the image patches.

In their paper, Zoran and Weiss used as a prior a huge GMM on patches learned previously on a big data set of noise-free natural images, which is their second main

FIGURE 1.10: Experimental results of HDMI on the test images. For the experiment, we used patch of size 10×10 and 80 clusters.

idea derived from machine learning. The rest of the method consists of an optimization step, made by a 'half-quadratic splitting', which makes use of an auxiliary variable $Z = (z_i)_{i \in I}$ corresponding to the set of extracted patches. Their goal is then to optimize the quantity

$$\lambda ||\widehat{u} - u||^2 + \sum_i \beta \left(||\chi_i(u) - z_i||^2 \right) - \sum_i \log f(z_i), \tag{1.12}$$

on both u and Z, where β is an additional hyperparameter (ratio between prior and data term) chosen heuristically. Algorithm 5 presents the details of the method, and more information can be found in the original paper. The visual results on the test images are presented in Figure 1.11.

The authors of EPLL interpret the quantity $EPLL_f(u)$ as the empirical expectation of the log-likelihood of a patch (up to a multiplicative factor $\frac{1}{N}$ with N the number of patches). We believe that this interpretation is reductive and we will show in Chapter 2 that it has another intuitive interpretation.

Algorithm 5 EPLL

Input: Noisy image $\widehat{u}$, standard deviation σ
Output: Denoised image $\widetilde{u}$

Learn a GMM on a large patch dataSet

for A certain amount of time while inscreasing β **do**
  $\widetilde{u} \leftarrow \left(\lambda + \beta \sum_i \chi_i^T \chi_i\right)^{-1} \left(\lambda \widehat{u} + \beta \sum_i z_i\right)$ ▷ Solving (1.12) according to u

  for Each patch $P_i = \chi_i(u)$ **do** ▷ Approximate the solution of (1.12) according to z
    $k_i \leftarrow \max_k \tau_{i,k}$ ▷ choosing the most likely component of the GMM
    $z_i \leftarrow \left(\Sigma_{k_i} + \sigma^2 \mathrm{I}\right)^{-1} \left(\Sigma_{k_i} \widehat{u} + \sigma^2 \mu_{k_i}\right)$
  end for
end for

BM3D

BM3D, introduced by Dabov et al., 2007, is one of the most famous and competitive patch-based algorithm for denoising. It relies on the NL-Means structure, but instead of adding a Bayesian framework like NL-Bayes, it makes use of a transform thresholding, applied on a 3D group of similar patches (all the 2D-patches are piled up in a third direction), in order to restore the sparsity the structure is supposed to satisfy, with an additional second step (the oracle trick, see 1.2.2). This leads to a quite complicated algorithm, presented on Algorithm 6.

Even if it is quite efficient, BM3D is optimized and contains lots of heuristic and empirical parameters tuning. Another paradigm in the patch-based denoising literature is the use of a sparse dictionary. It can be either a learned dictionary, like in Aharon, Elad, and Bruckstein, 2006 or a fixed basis, like the Fourier or Wavelet transform, like in Yu and Sapiro, 2011. Among those methods, which are very technical, one of the most competitive is BM3D.

FIGURE 1.11: Experimental results of EPLL on the test images. These results have been made with the original code that the authors of Zoran and Weiss, 2011 provided.

Algorithm 6 BM3D

Input: Noisy image $\widehat{u}$, standard deviation σ

Output: Denoised image $\widetilde{u}$

First iteration or Oracle iteration to obtain a basic estimate of the image

Set parameters for the first round : $\kappa, \lambda, N_{MAX}, s, \lambda_{3D}, \tau$, see original paper for more details

for Each patch P of size $\kappa \times \kappa$ with a step s in rows and columns **do**

  $(Q_i) \leftarrow$ Set of square patches in the square neighborhood of P of size $\lambda \times \lambda$ such that $||Q_i - P|| \leq \tau$

  if There are more than N_{MAX} similar patches **then**

    Keep only the N_{MAX} closest

  else

    Keep 2^p patches, with p as big as possible.

  end if

  Construct a 3D group $\mathcal{P}(P)$ from the patches (Q_i)

  Apply a biorthogonal spline wavelet on every patch in $\mathcal{P}(P)$

  Apply a Walsh-Hadamard transform along the third dimension of $\mathcal{P}(P)$

  Apply a hard thresholding with λ_{3D} to $\mathcal{P}(P)$ and set

$$w_P = \begin{cases} N_P^{-1} & \text{If } N_P \geq 1, \\ 1 & \text{Otherwise.} \end{cases},$$

  where N_P is the number of non-zero coefficients remaining

  Apply an inverse Walsh-Hadamard transform along the third dimension, and inverse biorthogonal spline wavelet on every patch Q_i of $\mathcal{P}(P)$ and make it an estimate $\widetilde{Q_i}$ for the patch Q_i associated with the weight w_P.

end for

Reconstruct the image by averaging all the estimator $\hat{Q}_j$ created with their associated weights.

Second and final iteration of the algorithm

Set parameters for second round : $\kappa, \lambda, N_{MAX}, s, \lambda_{3D}, \tau$, see original paper for more details

for Each patch P **from the basic estimate** of size $\kappa \times \kappa$ with a step s in rows and columns **do**

  Select (Q_i) a set of close patches to P from **the basic estimate** like in first round

  Construct two 3D group from the patches (Q_i), $\mathcal{P}(P)$ and $\mathcal{P}(P)$ from the noisy image and the basic estimate

  Apply a 2D-DCT and then a Walsh-Hadamard transform along the third axis on every patch in $\mathcal{P}(P)$ and $\mathcal{P}(P)$

  $\omega_P \leftarrow \frac{|\mathcal{P}(P_1)|^2}{|\mathcal{P}(P_1)|^2 + \sigma^2}$

  Apply a Wiener collaborative filtering by mulptiplicating element-wise $\tau_{3D}(\mathcal{P}(P))$ with the Wiener coefficient ω_P

$$w_P \leftarrow \begin{cases} ||\omega_P||_2^{-2} & \text{If } ||\omega_P||_2 > 0, \\ 1 & \text{Otherwise.} \end{cases},$$

  Apply an inverse Walsh-Hadamard transform along the third dimension, and inverse 2D DCT on every patch Q_i of $\mathcal{P}(P)$ and make it an estimate $\widetilde{Q_i}$ for the patch Q_i associated with the weight w_P.

end for

Reconstruct the image by averaging all the estimators $\hat{Q}_j$ created with their associated weights.

FIGURE 1.12: Experimental results of BM3D on the test images. The parameters choice follow the recommandation of **lebrun12analysis**.

		NL-Means	*NL-Bayes*	*EPLL*	*HDMI*	*BM3D*
$\sigma = 10$	Lena	34.30	35.84	35.15	35.90	**35.91**
	Barbara	33.28	35.17	33.26	**35.18**	35.09
	Cartoon	36.12	**37.90**	36.39	37.86	37.45
	Squares	43.21	47.22	46.68	**54.85**	50.70
$\sigma = 30$	Lena	29.45	31.20	30.44	31.08	**31.24**
	Barbara	27.06	**29.83**	27.07	29.68	29.76
	Cartoon	28.92	30.45	28.84	**30.64**	30.40
	Squares	38.39	40.73	39.47	**46.54**	41.25
$\sigma = 50$	Lena	27.41	28.81	28.04	28.68	**28.88**
	Barbara	25.58	25.97	24.10	26.76	**27.10**
	Cartoon	25.11	26.88	26.65	**27.18**	26.76
	Squares	37.66	36.79	35.78	**42.04**	37.31

FIGURE 1.13: PSNR of the different aggregation methods with NL-Bayes inference, on the test image of Figure 1.7, with standard deviation of $\sigma = 10$, $\sigma = 30$ and $\sigma = 50$. This corresponding visual results can be found in Figure 1.8, 1.9, 1.10, 1.11 and 1.12

Performance comparison

We present in Figure 1.13 the PSNR comparison of the different algorithms presented in this section, on the test images of Figure 1.7, with noises of standard deviation $\sigma = 10$, $\sigma = 30$ and $\sigma = 50$. Visual results can be found in Figure 1.8, 1.9, 1.10, 1.11 and 1.12.

As we can see, the quality of the results highly depends on the images. Simple images like squares are easier to denoise.

The results at low noise ($\sigma = 10$) are almost perfect, and it is hard for the eye to differentiate with the ground truth. Indeed, natural images like *Lena* or *Barbara* are already a bit noisy. For higher noise ($\sigma = 50$), the problem becomes very difficult as there are almost more noise than information. This enables to see the differences between these algorithms. Even if they have similar PSNR, their results on the same image differ a lot visually. EPLL has very poor performance for high noise in terms of visual quality (see *Barbara* in Figure 1.11) although its PSNR remains competitive. This highlights the idea mentioned in Section 1.2.2, that the PSNR is not an ideal criterion and must be considered with care.

It is also interesting to see that EPLL and NL-Means, even if they remain competitive, are a bit less effective (of 1 or 2 dB) than the other algorithms, which are more optimized and tuned. This shows that relevant improvements in efficiency come more from the paradigms than from the technical optimizations and from the tuning, even if they are important to exploit an idea to its maximum potential.

1.2.5 Conclusion

We reviewed some basics on Bayesian models and how to apply them to patch-based algorithms. As we saw, these models participate a lot in the success of patch-based methods.

These methods are an interesting approach to tackle the denoising problem, as they seem well adapted to natural images and the recent computer modeling development. Even if they are all based on the same principle, there is a huge variety of

approaches. Small details in the models and in how they are learned and inferred can change a lot the results.

Among them, the most competitive have very good results, but as the noise increases, lots of visual flaws appear in their estimations: they can still be greatly improved and these flaws are a good starting point to analyze and improve the models and the frameworks. Indeed, some similarities and pattern can be observed in the defects and artifacts in the image. By studying and classifying them, one can hope to better understand their origin and to remove them. This will be the main concern of the next chapter.

Chapter 2

Patch aggregation

This chapter is based on the publication:

Alexandre Saint-Dizier, Julie Delon, and Charles Bouveyron (2020). "A unified view on patch aggregation". In: *Journal of Mathematical Imaging and Vision* 62.2, pp. 149– 168.

Introduction

As we saw in Chapter 1, patch-based methods have shown promising results for image denoising. They all rely on three steps: the patch extraction step, the patch editing step and the aggregation step (see Section 1.2.3). In most of the literature, all the attention has been put into the patch editing step, which benefits from the recent progress in statistics, machine learning and signal processing. While powerful and complicated models have been used for patch editing, most methods share the same basic ideas for the patch extraction and the patch aggregation steps.

For the patch extraction, most methods simply take all overlapping square patches of a given size. The question of how to choose properly the patch size and more generally of the patch extraction has been less studied. This question could be approached with hyper-parameters estimation (see Section 1.1.6), but different patch size implies different patches and thus different data, which makes the model selection even more intricate. Some work has also been made to desing methods with adaptive patch size, like in Deledalle, Duval, and Salmon, 2012 and Kervrann and Boulanger, 2006. Besides, taking all the patches of an image can form very large data set. Recent study in texture synthesis have shown that choosing carefully the patches can improve greatly the speed of an algorithm for almost no loss of performance, by using determinantal point process for instance (see Launay and Leclaire, 2019). These ideas could be applied to image denoising to speed up the computation time. I decided in this PhD to focus on the aggregation step, as I believe that this step deserves more attention in patch-based methods.

In the aggregation step, the processed patches are merged together into a single image. While much attention has been paid on statistical or geometrical patch representations and interpretation, little work has been dedicated to explore this merging or aggregation step. Going from the image space to the patch space is a linear and straightforward operation, but recovering an image from a set of overlapping patches is straightforward only if all of these patches share the same values on their common pixels. Even for patches coming from the same image, this property is lost as soon as the patches undergo non trivial operations. For patches of size d, each pixel belongs to d different patches (neglecting the borders) and these patches yield d different estimates for the pixel value, as illustrated by Figure 1.6. In the literature, there are essentially four ways to answer the aggregation question:

1. For each pixel, keep only the estimator provided by the patch centered at this pixel (*central aggregation*);
2. For each pixel, average the d estimators with uniform weights (*uniform aggregation*);
3. For each pixel, average the d estimators with adapted weights (*weighted aggregation*);
4. Reconstruct the image from the patches as a solution of a variational problem.

The first solution is the one chosen in the first version of NL-Means (see Section 1.2.4). This approach ignores the information available in the rest of the patches. As a result, when applied in the context of image denoising for instance, residual noise can often be observed around edges or rare regions. A majority of methods tackle this issue by averaging the d estimates of the pixel, either with uniform weights Kervrann and Boulanger, 2006 or with weights taking into account the precision of each estimator Dabov et al., 2007; Salmon and Strozecki, 2010; Talebi, Zhu, and Milanfar, 2013, in order to minimize the variance of the aggregated estimator. A recent approach Romano and Elad, 2015, called SOS boosting, proposes to improve iteratively a denoising algorithm by reducing the gap between each restored patch and its value after uniform aggregation. The BM3D algorithm Dabov et al., 2007 uses weights which are chosen inversely proportional to the total variance of the sample of noisy patches used to estimate the denoised patch. More recently, the DCT-based denoising approaches Guleryuz, 2007; Pierazzo, Morel, and Facciolo, 2017 use weights chosen inversely proportional to the number of non-zero coefficients of the DCT after thresholding, giving more weights to patches that have a lot of coefficients set to 0 (flat patches for example). Other approaches draw on similar ideas to derive optimal weights Dengwen and Xiaoliu, 2009; Sezer and Altunbasak, 2009; Kervrann, 2014; Feng et al., 2015. Instead of the variance, some authors also attempt to minimize the risk of the final estimator at each pixel, by making use of Stein's Unbiased Risk Estimator (SURE) Deledalle, Duval, and Salmon, 2012; Van De Ville and Kocher, 2009. In Carrera et al., 2017, a comparison is led between global optimization and weighted aggregation for denoising purposes.

The last solution for patch aggregation, explored for instance in Elad and Aharon, 2006; Zoran and Weiss, 2011, consists of a global variational formulation of the restoration problem, including a global prior. These global formulations intrinsically include the aggregation problem, which is treated iteratively during the optimization process. In Zoran and Weiss, 2011, the log of the global prior (the expected patch log likelihood, or EPLL) is a sum of local priors on the patches and interpreted, up to a scalar, as "the expected log likelihood of a randomly chosen patch in the image". However, it can also be interpreted, up to a constant, as (the log of) a global image probability distribution, as already noted by Tabti et al., 2014. Other attempts Roth, Lempitsky, and Rother, 2009; Cho et al., 2008 have been made to construct a global image probability distribution from local patch priors, such as the field of experts Roth, Lempitsky, and Rother, 2009 which uses Markov Random Fields priors on pixels. We will see that the approach developed in the current chapter has strong links with these global interpretations. In a related direction, the fact that patches should coincide on their intersections can also be written as a hard constraint that can be included in any variational framework, as explored in the recent paper Paulino, 2018.

In texture synthesis, alternatives to aggregation have been considered, such as Efros and Freeman, 2001 which finds a minimal error boundary cut between two overlapping patches, or Raad, Desolneux, and Morel, 2016 which uses conditioning to force the new patches to be coherent with the part of the image which has already been synthesized.

In this chapter, we propose a novel perspective on this aggregation stage. To this aim, we focus on the case where each image patch is given a stochastic model on $\mathbb{R}^d$, for instance a Gaussian distribution or a mixture of Gaussian models. This situation is quite classical in Bayesian image restoration, where each patch is restored with a prior model (see Section 1.2.3). It is usual that these different models do not coincide on overlapping patches. In order to overcome this limitation, we introduce the notion of *patch fusion*, which draws on all the prior models to construct a global model on the whole image (up to a normalization), by taking into account the fact that these models should coincide on their overlaps. At the end, the final models for overlapping patches coincide but are not generally the same than the ones prescribed as input. As we shall see, the classical aggregation techniques described above can be interpreted as special cases of our fusion framework. Our notion of patch fusion also reconciles the point of view developed in EPLL Zoran and Weiss, 2011 and the conditioning approach suggested in Raad, Desolneux, and Morel, 2016 for texture synthesis.

2.1 Motivations

2.1.1 Directions of improvement

The different algorithms presented in Section 1.2.3 have good results in term of PSNR, but can still be improved, especially visually. By comparing their differences and similarities among the different images and methods, we can have an idea of which aspects of the methods work and which aspects limit the performances (see Section 1.2.3). As we saw, for low standard deviation of the noise, the results are almost perfect, and thus difficult to compare. The reconstruction issues become more and more visible as the standard deviation of the noise increases. At $\sigma = 50$, they are particularly noticeable while still being representative of the general behavior of the algorithms. This makes this level of noise a good choice to study these issues. The main reconstruction issues that we identified for $\sigma = 50$ of all the algorithms are presented in Figures 2.1, 2.2, 2.3, 2.4 and 2.5.

NL-Means (see Figure 2.1) is unique among the presented patch-based methods, as it is the only algorithm which uses *central aggregation* (some version using the uniform has been developed). This gives usually good results, but with some sort of blurry noise. This is due to a lack of structure, one of the main problem of central aggregation, which considers the pixel independently and does not favor local coherence. Besides, NL-Means uses very small patches, which does not help restoring border and textures as well. With a closer look, we can see that, in addition to the general noise impression remaining in the restored version, a bigger artifact remains next to the border and in some particular areas. This effect called the "rare patch effect", is due to the fact that some patches in the image are too singular (like the eyes of *Barbara* for instance), so that the algorithm does not find enough "close patches" to perform a large enough average in the case of NL-Means. This effect is inherent to patch-based methods and difficult to tackle since it is a counter example to the self-similarity on which they all rely. One way to tackle this issue it is to use a larger data set, like in EPLL, but it does not really solve the problem and raise

FIGURE 2.1: Issues on the reconstruction of *Barbara* and *Cartoon* with the NL-Means algorithm for $\sigma = 50$. As we can see, the images contains strong artifacts, especially around the borders.

other issues (computational time, over fitting, etc...). Eventually, we can see on the arm in *Cartoon* that NL-Means added some extra lines to it. This is also a general problem of patch-based methods that we call "wrong patch effect", when different patches are associated together by the algorithm (also as a consequence of the "rare patch effect"). This is one of the main source of creation of "artifacts" (a nonsense texture addition of the algorithm) by patch-based methods. This effect is particularly present in NL-Means and NL-Bayes because they use l_2 norm to compare patches.

The result of NL-Bayes on *Cartoon* (see Figure 2.2) shows a good example of "artifacts" : the weird contrasts on the table or on the hands. It can as well be explained by the "rare patch effect". This phenomenon occurs mainly around borders, and especially when the borders are not straight (like in this example). The right image of Figure 2.2 shows an opposite phenomenon, the algorithm has removed a whole part of the image (the antenna on the robot), but explained by the same rare patch effect : the algorithm has associated most of the patch of this area to patches from the nearby constant zone, having no better candidates. This leads to assimilate this zone to its constant surroundings. Yet, the algorithm works well for uniform areas, contrary to NL-Means. This shows the power of the Gaussian model, which is the main improvement of NL-Bayes. Yet, the transition between the two gray zones is imprecise, although it is still a simple type of frontier. I think that it can be explained by the use of small patches and the simplicity of the limitations of the single Gaussian model.

The result of EPLL are also very special. This method has two main differences with the others : its aggregation step, which is the (approximated) solution of a variational problem and the learning of its model, which is not performed on the image itself but rather beforehand on a large data set. We can see on the left picture of Figure 2.3, that the stripes of the trousers of *Barbara* are really poorly restored. This is probably due to the lack of such a particular pattern in the training data of EPLL (or in insufficient quantity). Besides, the algorithm has a strange behavior, it seems that it applies by default some uniform areas in regions where the model has a low variance. This leads to these visually unpleasant flat tints, which are very bad in term of visual quality but do not decrease too much the PSNR. Indeed, a stripe shifted of one pixel would have a smaller PSNR than a uniform gray area, which shows one again the limitation of this criteria. This phenomenon may be caused by

FIGURE 2.2: Issues on the reconstruction of *Cartoon* with the NL-Bayes algorithm for $\sigma = 50$. We can see on the left lots of artifacts on the table, and the disparition of the antenna of Bender on the right.

FIGURE 2.3: Issues on the reconstruction of *Barbara* and *Cartoon* with the EPLL algorithm for $\sigma = 50$. We can see on the left some gray flat tints instead of the stripes. On squares, the estimation is quite accurate, with still some fluffy effect in the gray square.

the 'Half quadratic splitting' of the aggregation procedure, which may be stuck in a local maxima, but it is difficult to say with all the different steps involved in the algorithm involved. On the image *Squares*, all the patches seem to be successfully restored by the model. However, in the grey square, we can see some low-frequency variation in this region, supposed to be uniform. This is what we call a *fluffy effect* (see Section 2.1.2). It may have the same origin as the flat tints, and also be due to the presence of the data term which drags the estimation toward the noisy image and to the low-frequency noise that the regularization by the prior is enable to compensate.

As we have seen in Figure 1.10, HDMI performs very well on artificial images like *Cartoon*. Figure 2.4 shows its main limitations. As we can see, the stripes of *Barbara* are very well denoised, since this pattern occurs a lot in the image, it has been well integrated to the model. However, the stripes are also visible on the floor next to the face of *Barbara* and even in the door. This problem is due to the predominance of the stripes in the learned model on the image. This could be tackled with more

FIGURE 2.4: Issues on the reconstruction of *Barbara* and *Cartoon* with the HDMI algorithm for $\sigma = 50$. We can see on *Barbara* that the stripes are well-denoised, but they are still (wrongly) visible on the floor and in the door. On *Lena*, their is no such defect, but a strong fluffy effect.

careful patch extraction which keeps some balance between the patterns of the image. The other main issue of HDMI is the fluffy effect, which becomes very present in textured zones, like in the face of *Lena*.

BM3D leads to different kinds of flaws. Like for HDMI, it performs well on the stripes of *Barbara*, but it (almost) does not add stripes on the constant zones. Yet, it creates lots of artifacts, like in the arm of *Barbara* or in the hat of *Lena*. The algorithm also behaves like if it would favor piecewise constant zone in its estimation (but with more general geometry than EPLL). Yet, because of the complexity of BM3D, it is difficult to track the origin of the encountered issues.

2.1.2 Fluffy effects and artifacts

In the previous section, we have seen that patch-based methods face two main difficulties : the fluffy effect and the presence of artifacts. We call fluffy effect the effect visually similar to cotton, that we can see for instance in the gray area of *Squares* in Figure 2.3. HDMI has a very strong fluffy effect and few artifacts while BM3D, on the contrary, has almost no fluffy effect and lots of artifacts. The origin of artifacts is hard to identify especially for complex algorithm like BM3D. On the other hand, the fluffy effect seems related to the low-frequency of the noise. Indeed, even if the noise is perfectly i.i.d., there are inevitably some local effects, for instance some spots where all the noise is higher or lower than its expectation. This effect is highlighted on Figure 2.6. This phenomenon justifies the use of large patches, since the probability of such a spot decreases greatly with its size. Besides, it questions the validity of uniform aggregation. Indeed, the low-frequency of the noise induces a small bias on each patch on a dark spot, and averaging all these patches does not have any effect on this bias. This is the motivation behind the next section.

2.1.3 Independence of the patches

The main assumption behind the inference of the models used in patch-based denoising is the independence of the patches, and thus of the noise added to it. Even if

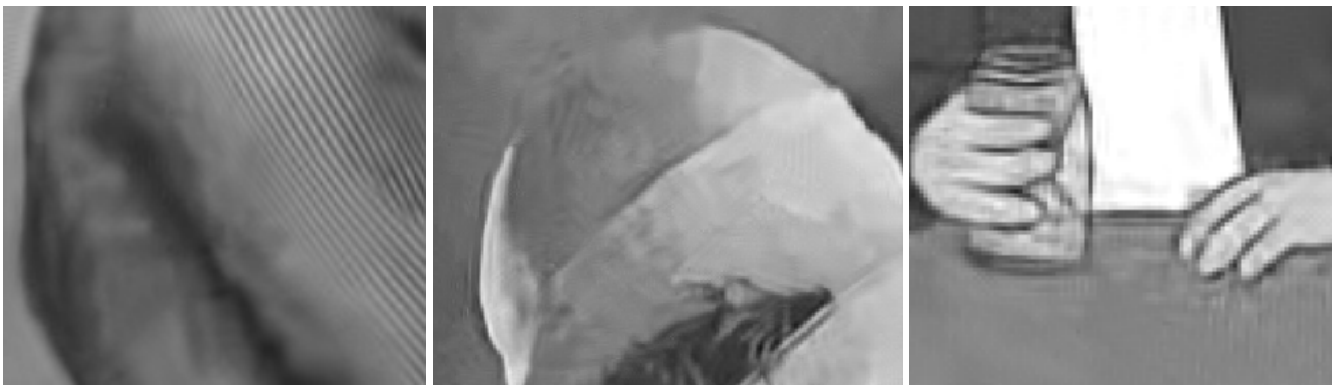


FIGURE 2.5: Issues on the reconstruction of *Barbara*, *Lena* and *Cartoon* with the BM3D algorithm for $\sigma = 50$. On *Barbara* and *Lena*, we can see some important artifacts, and also some piecewise constant effect in the hat of *Lena*. The results have also some edge artifacts, similar to those of NL-Bayes, that we can see in *Cartoon* around and on the hand.

FIGURE 2.6: Illustration of the fluffy effect. From left to right: the noisy image ($\sigma = 30$), the result of HDMI and the original image. We can see a good example of the fluffy effect on the shoulder of *Lena* in the central image. If we pay attention to the bottom left of these images (under the curls), we can see a small dark spot in the denoised image which is completely absent from the original image, but seems to exist in the noisy one. This is due to low-frequency noise, when the local expectation of the noise is higher than the global one. This is one of the main origin of the fluffy effect that patch-based algorithms struggle to avoid creating.

it is obviously false, it does seem realistic, since the value a pixel does not appear at the same place in all the patches it belongs to. It is well-known that this assumption is mandatory for inferring, but it is often forgotten that it is also the case for the aggregation. This step is less formalized and is most of the time justified by heuristics, but the main idea behind it is that the "i.i.d." values of the patches averaged at each pixel will decrease the noise. And as we saw in Section 2.1.2, this is not the case.

In a real world scenario, it is impossible to achieve real independence of the noise. Yet, in an experimental study, it is possible to add the noise directly to the patch after the patch extraction instead of adding it to the image. This enables to obtain data which are genuinely i.i.d. and to see how the algorithms behave on it. We have then four possibilities : we can learn the model with or without this oracle data set, and then restore and aggregate the patches with or without it. The result of this experiment with HDMI (patches of size 10×10) is presented on Figure 2.7. It is important to mention that in order to reconstruct the image using the oracle data set, each pixel was given the value of the patch of which it is the center, instead of averaging its value on all the patches it belongs. Otherwise, this experiment would have been useless, since averaging 100 version of the same image with different noise divides the noise standard deviation by 100.

Unsurprisingly, the utilization of the oracle data set has no influence on the model inference. As we said, this assumption is quite reasonable for learning purposes: the patch extraction takes the pixel values apart. On the contrary, the aggregation brings pixel values together, and the non-independence becomes problematic. This is clearly shown in the second column of Figure 2.7. With the oracle data set, the estimations look almost perfect, whatever the learning procedure.

2.1.4 Conclusion

As we saw, patch-based methods show a lot of potential, but are far to be perfect. The patch edition step has been widely studied and discussed in the literature, to the detriment of the two other steps of the framework : patch extraction and patch aggregation.

More careful patch extraction could benefit to the computation time of the patch-based methods, and help solving some small learning over-fitting. But one of the main limitation to patch-based methods appears to be the aggregation. Indeed, it seems irrelevant to perform a simple averaging on data on which much energy has been spent to model precisely. This is yet what most methods do, as if there were no relation between the patches and the whole image. This appears to be a loss of precious information that could be used to better aggregate. This following study is a first attempt to incorporate the aggregation fully into the Bayesian framework used in patch edition.

2.2 Patch model, agreement and fusion

In this section, we define what we call a *patch model*, which extends the classical definition of a deterministic patch in a stochastic setting. This model will be used thereafter to define a notion of *patch fusion*, motivated by the situation described in Section 2.1.

(A) **Learning** : real data, **aggregation** : real data.

(B) **Learning** : real data, **aggregation** : oracle data.

(C) **Learning** : oracle data, **aggregation** : real data.

(D) **Learning** : oracle data, **aggregation** : oracle data.

FIGURE 2.7: Experiment on *Lena* with HDMI ($\sigma = 30$) using real or oracle data on the learning and/or on the restoration and with a central aggregation, so that the comparison remains relevant. Each picture corresponds to one of the four different possibilities.

2.2.1 Patch model: a probabilistic patch representation

Let us define a patch model of size d on the discrete grid $\mathbf{\Omega}$. The notion is illustrated by Figure 2.8.

Definition 3. A patch model P on the grid $\mathbf{\Omega}$ is a couple (Ω, ν), where $\Omega \subset \mathbf{\Omega}$ and where ν is a probability distribution on $\mathbb{R}^\Omega$. We refer to ν as the *distribution* of the patch model, and to Ω as its *domain*. We denote by $\mathcal{P}$ the set of all patch models on $\mathbf{\Omega}$.

This definition is a generalization of the classical definition of a patch on a grid.

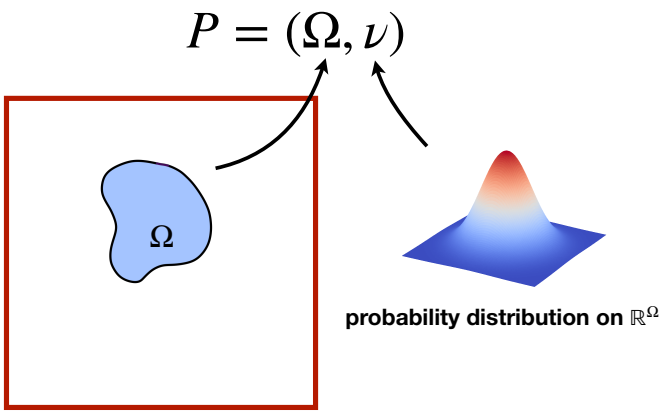


FIGURE 2.8: A patch model on $\mathbf{\Omega}$ is composed of a domain Ω (a subset of $\mathbf{\Omega}$) and a probability distribution ν on $\mathbb{R}^{\Omega}$.

Indeed, a deterministic patch P can be assimilated to a Dirac distribution on $\mathbb{R}^{\Omega}$. We do not impose any connectedness of Ω in our definition.

We are now in the position to define the notion of agreement between two patch models (see Figure 2.9), which says that two patch models agree if they share the same distribution on their intersection.

Definition 4 (Patch model agreement)**.** Let $P_1 = (\Omega_1, \nu_1)$ and $P_2 = (\Omega_2, \nu_2)$ be two patch models in $\mathcal{P}$. We say that these two patch models agree and we write $P_1 \hat{=} P_2$ if and only if

$$\nu_{1|\Omega_1 \cap \Omega_2} = \nu_{2|\Omega_1 \cap \Omega_2}.$$

Therefore, two disjoint patch models (P_1 and P_2 such that $\Omega_1 \cap \Omega_2 = \emptyset$) agree automatically. The $\hat{=}$ relation is reflexive and symmetric, but not transitive.

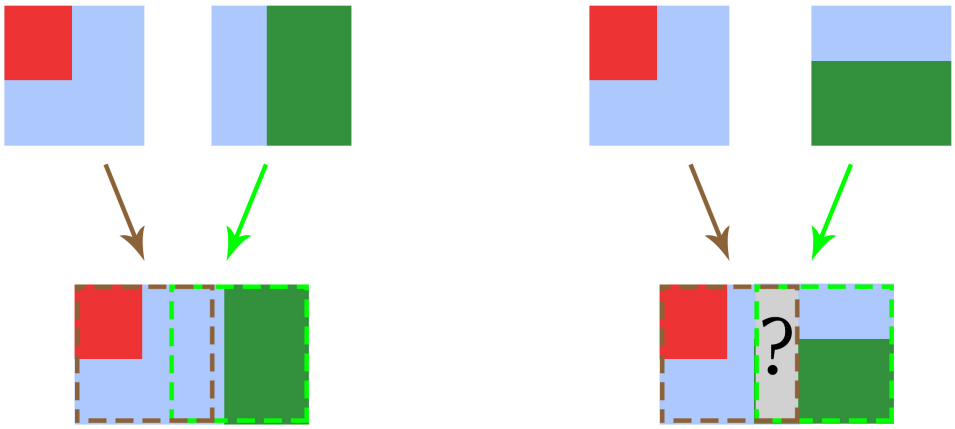
Observe that this definition can also be applied to deterministic patches. We say that they agree if their values on their overlap coincide. We will also denote this with the symbol $\hat{=}$. We now define the notion of compatibility between patch models, which is much less restrictive that the patch agreement, and will be important to introduce the notion of patch fusion in the next section.

Definition 5 (Patch model compatibility)**.** Let $(P_n)_{n \in [\![1,N]\!]} = (\Omega_n, f_n(x)dx)_{n \in [\![1,N]\!]}$ be a set of N patch models with bounded densities $f_1, \dots, f_n$. We say that these patch models are compatible if

$$\int_{z \in \mathbb{R}^{\cup_{n=1}^{N} \Omega_n}} \prod_{n=1}^{N} f_n(z_{|\Omega_n}) dz > 0.$$

2.2.2 Patch model fusion

We can now define the fusion of two patch models. As explained before, this definition is motivated by the situation where we end up with one or several distributions on the different patches. The fusion operation permits to construct directly a distribution for the whole image from the different patch models. It simply consists in



(A) The two patches agree, the aggregation is straightforward

(B) The two patches do not agree, the aggregation is ambiguous

FIGURE 2.9: Illustration of the notion of agreement between two deterministic patches.

aggregating patch models by merging their domains, and defining a novel distribution on this merged domain as a (specific) product of their original distributions.

Definition 6 (Patch model fusion)**.** Let $P_1 = (\Omega_1, \nu_1)$ and $P_2 = (\Omega_2, \nu_2)$ be two compatible patch models. We suppose that the distributions ν_1 and ν_2 have bounded densities f_1 and f_2.

The fusion $P_1 \odot P_2$ is the patch model defined by (Ω, ν) where $\Omega = \Omega_1 \cup \Omega_2$ and $\nu(dx) = f(x)dx$, with

$$\forall x \in \mathbb{R}^\Omega, \quad f(x) = \frac{f_1(x_{|\Omega_1}) f_2(x_{|\Omega_2})}{\int_{z \in \mathbb{R}^\Omega} f_1(z_{|\Omega_1}) f_2(z_{|\Omega_2}) dz}.$$

Remark 7.

- *For the sake of simplicity, we restrict ourselves to the set of patch models with bounded densities. This strong assumption is convenient because it is stable for the fusion operation, and it is always satisfied with the distributions we consider, but it could be relaxed. In practice, we only need to ensure that*

$$\int_{z \in \mathbb{R}^\Omega} f_1(z_{|\Omega_1}) f_2(z_{|\Omega_2}) dz < +\infty.$$

- *With this definition, the notion of patch fusion does not directly apply to deterministic patches if we see them as Dirac distributions. However, as we shall see in Section 2.3, the notion of fusion extends well to deterministic patches, if they are modeled by Gaussian distributions with their value as expectation, and with a covariance proportional to the identity.*

This fusion definition has a very intuitive motivation, as we shall see in the next proposition.

Proposition 8 (Interpretation of the fusion)**.** *Let $P_1 = (\Omega_1, \nu_1)$ and $P_2 = (\Omega_2, \nu_2)$ be two compatible patch models and define $P_1 \odot P_2 = (\Omega, \nu)$. Assume that the distributions*

ν_1 and ν_2 have bounded densities f_1 and f_2. Let $Z_1 \sim \nu_1$ and $Z_2 \sim \nu_2$ be two independent random vectors. We write $Z_1 = \begin{pmatrix} X_1 \\ Y_1 \end{pmatrix}$, where X_1 corresponds to the coordinates of Z_1 on $\Omega_1 \cap \Omega_2$ (so $X_1 \sim (\nu_1)_{|\Omega_1 \cap \Omega_2}$) and Y_1 to the coordinates on $\Omega_1 \setminus \Omega_2$. We write $Z_2 = \begin{pmatrix} X_2 \\ Y_2 \end{pmatrix}$ in the same way, where X_2 corresponds to the coordinates of Z_2 on $\Omega_1 \cap \Omega_2$ and Y_2 on $\Omega_2 \setminus \Omega_1$ (Y_1 and Y_2 may not have the same dimension). Then ν is the conditional probability distribution of the vector $\begin{pmatrix} X_1 \\ Y_1 \\ Y_2 \end{pmatrix}$ given $X_1 = X_2$.

Proof. In the following, we denote by $p(X = x)$ the value of the density of the random variable X at x. For $z = (x_1, y_1, y_2) \in \mathbb{R}^{\Omega_1 \cap \Omega_2} \times \mathbb{R}^{\Omega_1 \setminus \Omega_2} \times \mathbb{R}^{\Omega_2 \setminus \Omega_1}$, we want to calculate the conditional density

$$p\left(\begin{pmatrix} X_1 \\ Y_1 \\ Y_2 \end{pmatrix} = \begin{pmatrix} x_1 \\ y_1 \\ y_2 \end{pmatrix} | X_1 = X_2\right).$$

This conditional density can be written

$$\frac{p((X_1, Y_1, X_2, Y_2) = (x_1, y_1, x_1, y_2))}{p(X_1 - X_2 = 0)}$$

where

$$\begin{aligned} p((X_1, Y_1, X_2, Y_2) = (x_1, y_1, x_1, y_2)) &= p((X_1, Y_1) = (x_1, y_1)) \times p((X_2, Y_2) = (x_1, y_2)) \\ &= f_1\left(\begin{pmatrix} x_1 \\ y_1 \end{pmatrix}\right) \times f_2\left(\begin{pmatrix} x_1 \\ y_2 \end{pmatrix}\right) = f_1(z_{|\Omega_1}) \times f_2(z_{|\Omega_2}), \end{aligned}$$

by independence of Z_1 and Z_2. Moreover,

$$p(X_1 - X_2 = 0) = \int p(X_1 = x_1, X_2 = x_1) dx_1 = \int p(X_1 = x_1) \times p(X_2 = x_1) dx_1.$$

Since

$$p(X_1 = x_1) = \int f_1\left(\begin{pmatrix} x_1 \\ y_1 \end{pmatrix}\right) dy_1 \text{ and } p(X_2 = x_1) = \int f_2\left(\begin{pmatrix} x_1 \\ y_2 \end{pmatrix}\right) dy_2,$$

we conclude that

$$p(X_1 - X_2 = 0) = \int f_1(z_{|\Omega_1}) \times f_2(z_{|\Omega_2}) dz > 0.$$

□

The fusion operation is therefore a way to combine two patch models while imposing these models to be equal on the intersection of their domains.

In order to extend this fusion operation to larger sets of patches, we need the following proposition.

Proposition 9. *For any compatible patch models with bounded densities P_1, P_2 and P_3 in $\mathcal{P}$, the fusion operation $\odot$ is well-defined and satisfies*

- *$P_1 \odot P_2 \in \mathcal{P}$ and has a bounded density.*

- $(P_1 \odot P_2) \odot P_3 = P_1 \odot (P_2 \odot P_3)$.
- $P_1 \odot P_2 = P_2 \odot P_1$

Proof. Let $P_1 = (\Omega_1, f_1(x)dx)$, $P_2 = (\Omega_2, f_2(x)dx)$, $P_3 = (\Omega_3, f_3(x)dx)$ and $(\hat{\Omega}, \hat{f}(x)dx) = P_1 \odot P_2$. We have $\hat{\Omega} = \Omega_1 \cup \Omega_2$ and

$$\hat{f}(x) \propto f_1(x_{|\Omega_1}) \times f_2(x_{|\Omega_2}),$$

which clearly shows the commutativity. So $P_1 \odot P_2$ has also a bounded density and it is straightforward from the definition that $P_1 \odot P_2$ is compatible with P_3. Besides, if we have $(\bar{\Omega}, \bar{f}dx) = (P_1 \odot P_2) \odot P_3$, we get

$$\bar{f}(x) \propto f_1(x_{|\Omega_1}) \times f_2(x_{|\Omega_2}) \times f_3(x_{|\Omega_3}),$$

which clearly shows the associativity.

□

Remark 10. *This proposition ensures the stability and coherence of the operation, which can therefore be extended to any number of compatible patch models without ambiguity. For any set of compatible patch models with bounded densities written*

$$(P_n)_{n\in[\![1,N]\!]} = (\Omega_n, f_n(x)dx)_{n\in[\![1,N]\!]},$$

we will denote this fusion by

$$\bigodot_n P_n = (\Omega, f(x)dx), \text{ with } \Omega = \bigcup_n \Omega_n \text{ and}$$

$$\forall x \in \mathbb{R}^\Omega, f(x) \propto \prod_n f_n(x_{|\Omega_n}).$$

Merging patch models in any order will always yield the same result (under the condition of compatibility and bounded densities). This operation can be used to propagate and connect all patch models to obtain a single image model.

2.2.3 Fused image model

The previous fusion operation can be used to define a global model on the whole image space from a set of local patch models.

Definition 11. Let E be a set of patch models. We say that E *covers the image support* if every pixel of $\mathbf{\Omega}$ belongs to the domain of at least one patch model of E, i.e.

$$\forall i \in \mathbf{\Omega}, \exists P = (\Omega, \nu) \in E \text{ such that } i \in \Omega.$$

We say that E *is coherent* if all patch models in E agree, i.e.

$$\forall\, (P_1, P_2) \in E^2, P_1 \hat{=} P_2.$$

We say that E *represents an image* if E covers the image support and is coherent.

For a set E of compatible patch models which covers the image support, Proposition 9 ensures that it is possible to fuse all the patch models of E to obtain a global model $(\mathbf{\Omega}, \boldsymbol{\nu}) = \bigodot_{P\in E} P$ on the image. As a by-product, this constructs a new set $\hat{E}$ by

$$\hat{E} := \{(\Omega, \boldsymbol{\nu}_{|\Omega}) \text{ with } P = (\Omega, \nu) \in E\}.$$

For each patch (Ω, ν) in E, there is a patch $\left(\Omega, \boldsymbol{\nu}_{|\Omega}\right)$ in $\hat{E}$ with the same domain, but with the marginal of $\boldsymbol{\nu}$ on Ω as a distribution instead of ν. Therefore, $\hat{E}$ covers the image, according to the previous definition.

Observe that this coherent set $\hat{E}$ is generally different from the set E, even in the case where E is obtained as all the marginals of a patch model on the whole image. Indeed, even if they agree, fusing two patch models does not preserve their common distribution on their intersection. Indeed, the fusion is not *stable*, in the sense that in general $P \odot P \neq P$. Fusing a patch model P adds some information: having two patches with the same distribution gives more confidence in this distribution than having only one. Furthermore, this property is not compatible with the associativity and the commutativity: if P and Q are two patch models such that $P \odot P = P$, then one must have $(P \odot Q) \odot P = Q \odot P$, which is not desirable because $P \odot Q \neq Q$ in general. Still, the fusion ensures a weaker stability, but well-suited for our application: $MLE(P \odot P) = MLE(P)$ where MLE is the Maximum Likelihood Estimator.

In practice, the previous definitions lead to generic algorithms which consist in fusing all patch models iteratively, in any order. This is justified by proposition 9, but is not necessarily efficient. How the fusion is performed in practice should depend on the considered distributions.

In the case of normal or uniform patch models, we will see in the next section that the fusion has a closed-form solution. We did not investigate more involved models, but we think that approximate schemes could be used for more complex distributions.

2.3 Application to particular distributions

2.3.1 Uniform distribution

A very simple example of patch model fusion can be derived in the case of uniform distributions.

Proposition 12 (Fusion of uniform patch models)**.** *Let $A \subset \mathbb{R}^{\Omega_A}$ and $B \subset \mathbb{R}^{\Omega_B}$ be two bounded borelian sets, and $P_A = (\Omega_A, \nu_A), P_B = (\Omega_B, \nu_B)$ be two patch models with uniform distribution on A and B,* i.e. *such that $\nu_A = \frac{1}{|A|}\mathbb{1}_A$, $\nu_B = \frac{1}{|B|}\mathbb{1}_B$. Let $\Omega = \Omega_A \cup \Omega_B$ and $C = \{x \in \mathbb{R}^{\Omega}; x_{|\Omega_A} \in A \text{ and } x_{|\Omega_B} \in B\}$.*

If C is of strictly positive Lebesgue measure in $\mathbb{R}^{\Omega}$, then P_A and P_B are compatible and denoting $P_A \odot P_B$ by (Ω, ν), ν is a uniform distribution on C.

In other terms, the fusion of two uniform patch models is also a uniform patch model. Its distribution is the only uniform distribution on $\mathbb{R}^{\Omega}$ whose marginal distributions on Ω_A and Ω_B are P_A and P_B.

This illustrates the behavior of the fusion operation, which forces patch models to agree on their intersection. As a consequence, a patch model with a peaked distribution will impose its opinion to the other patch models: we expect a confident model to be given more credit in the final aggregation. As we shall see, the Gaussian case keeps this behavior, but in a softer way.

2.3.2 Gaussian distributions

The Gaussian distribution also yields a closed form expression for the fusion operation.

Proposition 13 (Fusion of Gaussian patch models). *Let $P_1 = (\Omega_1, \nu_1)$ and $P_2 = (\Omega_2, \nu_2)$ be two Gaussian patch models with positive definite covariances. We write x the variable representing the common pixels of the two patch models (*i.e. *those in $\Omega_1 \cap \Omega_2$) and y for the others (*i.e. *those in $\Omega_1 \setminus \Omega_2$ for P_1, and those in $\Omega_2 \setminus \Omega_1$ for P_2), and we write*

$$\nu_1 = \mathcal{N}\left(\begin{pmatrix}\mu_x\\ \mu_y\end{pmatrix}, \begin{pmatrix}\Sigma_x & \Sigma_{xy}\\ \Sigma_{xy}^T & \Sigma_y\end{pmatrix}\right) \text{ and } \nu_2 = \mathcal{N}\left(\begin{pmatrix}\mu'_x\\ \mu'_y\end{pmatrix}, \begin{pmatrix}\Sigma'_x & \Sigma'_{xy}\\ (\Sigma'_{xy})^T & \Sigma'_y\end{pmatrix}\right).$$

Then (Ω_1, ν_1) and (Ω_2, ν_2) are compatible and the distribution of $(\Omega_1, \nu_1) \odot (\Omega_2, \nu_2)$ is Gaussian with parameters

$$\boldsymbol{\mu} = \begin{pmatrix}\mu_x\\ \mu_y\\ \mu'_y\end{pmatrix} + \begin{pmatrix}\Sigma_x (\Sigma_x + \Sigma'_x)^{-1}\\ (\Sigma_{xy})^T (\Sigma_x + \Sigma'_x)^{-1}\\ -(\Sigma_{xy})^T (\Sigma_x + \Sigma'_x)^{-1}\end{pmatrix} (\mu_x - \mu'_x)$$

and

$$\boldsymbol{\Sigma} = \begin{pmatrix}\begin{pmatrix}\Sigma_x & \Sigma_{xy}\\ (\Sigma_{xy})^T & \Sigma'_y\end{pmatrix} & \begin{matrix}0\\ \end{matrix}\\ \begin{matrix}0 & \end{matrix} & \Sigma'_y\end{pmatrix} - \begin{pmatrix}\Sigma_x (\Sigma_x + \Sigma'_x)^{-1}\\ (\Sigma_{xy})^T (\Sigma_x + \Sigma'_x)^{-1}\\ -(\Sigma'_{xy})^T (\Sigma_x + \Sigma'_x)^{-1}\end{pmatrix} \begin{pmatrix}\Sigma_x\\ \Sigma_{xy}\\ -\Sigma'_{xy}\end{pmatrix}^T.$$

Proof. Let $Z_1 = \begin{pmatrix}X_1\\ Y_1\end{pmatrix} \sim \nu_1$ and $Z_2 = \begin{pmatrix}X_2\\ Y_2\end{pmatrix} \sim \nu_2$ be two independent Gaussian random vectors. From proposition 8, we know that the distribution we are looking for is the conditional probability distribution of $\begin{pmatrix}X_1\\ Y_1\\ Y_2\end{pmatrix}$ knowing $X_1 = X_2$. The random variable $W = X_1 - X_2$ follows a Gaussian distribution with expectation $\mu_x - \mu'_x$ and covariance $\Sigma_x + \Sigma'_x$. Similarly, we know that $\begin{pmatrix}Z_1\\ Y_2\\ W\end{pmatrix}$ is a Gaussian random vector with parameters $\hat{\mu}, \hat{\Sigma}$ such that

$$\hat{\mu} = \begin{pmatrix}\mu_x\\ \mu_y\\ \mu'_y\\ \mu_x - \mu'_x\end{pmatrix} \text{ and } \hat{\Sigma} = \begin{pmatrix}\Sigma_x & \Sigma_{xy} & 0 & \Sigma_x\\ \Sigma_{xy}^T & \Sigma_y & 0 & \Sigma_{xy}^T\\ 0 & 0 & \Sigma'_y & -\Sigma'^T_{xy}\\ \Sigma_x^T & \Sigma_{xy} & -\Sigma'_{xy} & \Sigma_x + \Sigma'_x\end{pmatrix}.$$

Indeed, since $Z_1 = \begin{pmatrix}X_1\\ Y_1\end{pmatrix}$ and $Z_2 = \begin{pmatrix}X_2\\ Y_2\end{pmatrix}$ are independent, the covariance between Z_1 and W can be written

$$\mathrm{Cov}(Z_1, W) = \mathrm{Cov}(Z_1, X_1 - X_2) = \mathrm{Cov}(Z_1, X_1) = \begin{pmatrix}\Sigma_x\\ \Sigma_{xy}^T\end{pmatrix},$$

and the covariance between Y_2 and W is

$$\mathrm{Cov}(Y_2, W) = -\mathrm{Cov}(Y_2, X_2) = -\Sigma'^T_{xy}.$$

It follows that the conditional density of $\begin{pmatrix} X_1 \\ Y_1 \\ Y_2 \end{pmatrix}$ knowing $W = 0$ is a normal distribution with expectation

$$\begin{aligned} \boldsymbol{\mu} &= \mathbb{E}\begin{pmatrix} Z_1 \\ Y_2 \end{pmatrix} + \begin{pmatrix} \mathrm{Cov}(Z_1, W) \\ \mathrm{Cov}(Y_2, W) \end{pmatrix} \mathrm{Cov}(W, W)^{-1}(0 - \mathbb{E}(W)) \\ &= \begin{pmatrix} \mu_x \\ \mu_y \\ \mu'_y \end{pmatrix} + \begin{pmatrix} \Sigma_x \left(\Sigma_x + \Sigma'_x\right)^{-1} \\ \Sigma_{xy}^T \left(\Sigma_x + \Sigma'_x\right)^{-1} \\ -\Sigma_{xy}'^T \left(\Sigma_x + \Sigma'_x\right)^{-1} \end{pmatrix} (\mu'_x - \mu_x) \end{aligned}$$

and covariance matrix

$$\begin{aligned} \boldsymbol{\Sigma} &= \mathrm{Cov}\left(\begin{pmatrix} Z_1 \\ Y_2 \end{pmatrix}, \begin{pmatrix} Z_1 \\ Y_2 \end{pmatrix}\right) - \begin{pmatrix} \mathrm{Cov}(Z_1, W) \\ \mathrm{Cov}(Y_2, W) \end{pmatrix} \mathrm{Cov}(W, W)^{-1} \begin{pmatrix} \mathrm{Cov}(Z_1, W)^T \\ \mathrm{Cov}(Y_2, W)^T \end{pmatrix} \\ &= \begin{pmatrix} \begin{pmatrix} \Sigma_x & \Sigma_{xy} \\ (\Sigma_{xy})^T & \Sigma'_y \end{pmatrix} & \begin{matrix} 0 \\ \end{matrix} \\ \begin{matrix} 0 & \end{matrix} & \Sigma'_y \end{pmatrix} - \begin{pmatrix} \Sigma_x \left(\Sigma_x + \Sigma'_x\right)^{-1} \\ (\Sigma_{xy})^T \left(\Sigma_x + \Sigma'_x\right)^{-1} \\ -(\Sigma'_{xy})^T \left(\Sigma_x + \Sigma'_x\right)^{-1} \end{pmatrix} \begin{pmatrix} \Sigma_x \\ \Sigma_{xy} \\ -\Sigma'_{xy} \end{pmatrix}^T. \end{aligned}$$

□

Remark 14. *We have defined the fusion only for distributions with densities, but in this case we see that we could extend the fusion to singular Gaussian distributions such that $\Sigma_x + \Sigma'_x$ is invertible.*

As a consequence, the set of all Gaussian patch models is stable by fusion. So if we have a set E of Gaussian patch models which covers the image, the resulting fusion of all the patch models from E will be a huge Gaussian model on the whole image support.

If we merge N Gaussian patch models $(\Omega_n, \nu_n)_{n\in[\![1,N]\!]}$ with expectations $(\mu_n)_{n\in[\![1,N]\!]}$ and precision matrices [1] $(\Lambda_n)_{n\in[\![1,N]\!]}$, a very simple formula can be derived to link the parameters (μ, Λ) of the fused Gaussian model and the set $(\mu_n, \Lambda_n)_{n\in[\![1,N]\!]}$. Before giving this formula in the next proposition, note that we see the expectations μ_n as vectors of $\mathbb{R}^\Omega$ and matrices Λ_n as matrices of $\mathbb{R}^{\Omega\times\Omega}$, which means that $\mu_n(i)$ is the expectation of the patch n at the pixel i, and is thus defined only if i belongs to Ω_n.

Proposition 15. *Let $(\Omega_n, \nu_n)_{n\in[\![1,N]\!]}$ be N Gaussian patch models, with expectations $(\mu_n)_{n\in[\![1,N]\!]}$ and precision matrices $(\Lambda_n)_{n\in[\![1,N]\!]}$. Let $P = (\Omega, \nu) = \bigodot_{n\in[\![1,N]\!]}(\Omega_n, \nu_n)$ the patch model obtained by fusing all these patch models.*

Then P is a Gaussian patch model, whose precision matrix Λ and expectation μ satisfy, for all $(i, j) \in \Omega \times \Omega$,

$$\begin{aligned} \Lambda(i,j) &= \sum_{1\le n\le N, i\in\Omega_n, j\in\Omega_n} \Lambda_n(i,j). \\ (\Lambda\mu)(i) &= \sum_{1\le n\le N, i\in\Omega_n} (\Lambda_n\mu_n)(i). \end{aligned} \tag{2.1}$$

[1] The precision matrix is the inverse of the covariance matrix.

Proof. From Proposition 13, we know that ν is a Gaussian distribution $\mathcal{N}(\mu, \Lambda^{-1})$. Denoting the density of this distribution by f, we have

$$-\log f(x) = \frac{1}{2}(x-\mu)^T \Lambda (x-\mu) + \text{cte}.$$

According to remark 10, we also have

$$\begin{aligned} -\log f(x) &= -\sum_n \log f_n(x_{|\Omega_n}) + \text{cte} \\ &= \frac{1}{2}\sum_{n=1}^{N} (x_{|\Omega_n} - \mu_n)^T \Lambda_n (x_{|\Omega_n} - \mu_n) + \text{cte}. \end{aligned}$$

Equations for Λ and μ follow by identifying the covariance matrices and expectations of these Gaussian distributions.

□

Remark 16. *Observe that while the precision matrices can be easily derived, the value of the expectation of the whole Gaussian is not directly accessible from* (2.1)*, since the precision matrix needs to be inverted.*

2.3.3 Fusion algorithm for Gaussian distributions

From the previous results, we can derive a simple and explicit fusion algorithm for normally distributed patches. In practice, if we aim at merging a set E of Gaussian patch models covering an image, keeping in memory and computing the covariance of the global Gaussian model is not tractable, since it requires to deal with a $(s_x \times s_y)^2$ matrix. Thanks to Proposition 15, we know that the precision matrix is sparse, but we have no such result on the covariance matrix. Still, if necessary, we can approximate the global covariance matrix by noticing that pixels which are far enough from each other do not much influence each other. For instance, using standard Gaussian models for the image *Lena*, we observe that beyond a distance of $2\sqrt{d}$, patch models do not influence each other anymore, as illustrated by Figure 2.10. It means that the covariance matrix of the whole image is almost sparse. This gives us the possibility to compute and store this covariance matrix much more easily, as described in algorithm 7 and figure 2.11.

In practice, this algorithm permits to compute the whole fused model. The computation is however quite slow: for a 512×512 image and 10×10 patches, fusing all patch models takes several minutes on a recent computer.

2.4 Link with classical aggregation methods

2.4.1 Standard aggregations

In the previous section, we have seen how to construct a distribution on a whole image from a set of compatible patch models. This construction, while theoretical, actually contains the main aggregation procedures used in the literature as special cases. More precisely, we shall see that these aggregation procedures can be seen as special cases of the fusion of Gaussian patch models with diagonal covariances.

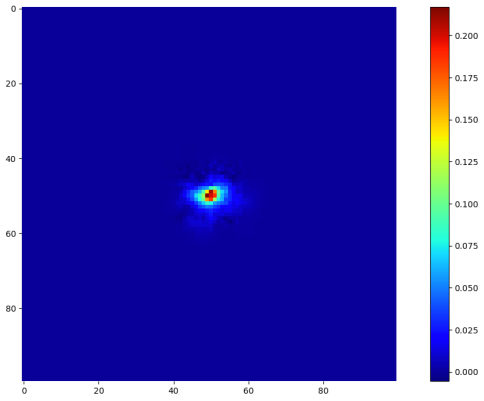


FIGURE 2.10: In this experiment, we compute a complete Gaussian model for the image *Lena* thanks to our fusion algorithm. The figure shows the resulting (absolute values of the) correlation map of a pixel in a 100×100 patch. As we can see, the correlation decreases to 0 very fast when we move away from the center. This experiment was made using 10×10 patches, which justifies to approximate by 0 the correlation between pixels at distance greater than $2\sqrt{d}$.

Algorithm 7 Approximation of the fusion procedure for Gaussian models with sparsity hypotheses on the covariance matrix

Input: Set $\mathcal{P}$ of square patches of size d, block-size b
Output: Aggregated image $\widetilde{u}$

1: Compute $\mathcal{B}$, partition of the image domain composed of disjoint blocks of size $b \times b$.
2: $s \leftarrow 2 \times \sqrt{d}$ (sparsity parameter)
3: **for** $B \in \mathcal{B}$ **do**
4: $\quad \widetilde{B} \leftarrow$ block of size $(b+s) \times (b+s)$ centered in B
5: $\quad \mathcal{P}_{\widetilde{B}} \leftarrow \{P \in \mathcal{P} | P \subset \widetilde{B}\}$
6: $\quad$ Compute $u_{\widetilde{B}}$ by fusing iteratively all patches from $\mathcal{P}_{\widetilde{B}}$ using proposition 13
7: $\quad \widetilde{u}_{|B} \leftarrow u_{\widetilde{B}|_B}$
8: **end for**

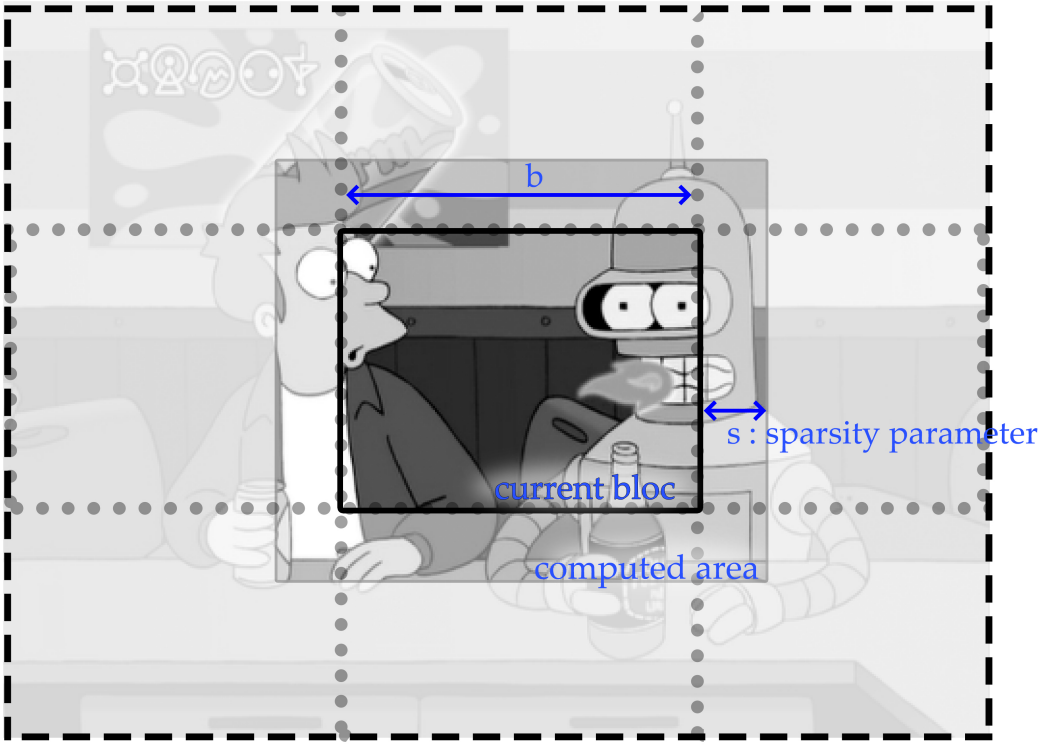


FIGURE 2.11: Illustration of algorithm 7. The image is divided into blocks of size b. For each block B, we extend this block by a distance s into a larger block $\tilde{B}$. The fusion of all patch models in $\tilde{B}$ is computed, but only the values of pixels belonging to B are kept.

Proposition 17. *Let (Ω_1, ν_1) and (Ω_2, ν_2) be two Gaussian patch models with diagonal positive definite covariances*

$$\nu_1 = \mathcal{N}\left(\begin{pmatrix}\mu_x\\ \mu_y\end{pmatrix}, \begin{pmatrix}\Sigma_x & 0\\ 0 & \Sigma_y\end{pmatrix}\right) \text{ and } \nu_2 = \mathcal{N}\left(\begin{pmatrix}\mu'_x\\ \mu'_y\end{pmatrix}, \begin{pmatrix}\Sigma'_x & 0\\ 0 & \Sigma'_y\end{pmatrix}\right),$$

where the variable x represents the common coordinates of the two patch models (μ_y and μ'_y may not have the same dimension). Then the patch models (Ω_1, ν_1) and (Ω_2, ν_2) are compatible and the distribution of $(\Omega_1, \nu_1) \odot (\Omega_2, \nu_2)$ is a Gaussian distribution with parameters

$$\boldsymbol{\mu} = \begin{pmatrix}(\Sigma_x^{-1} + \Sigma'^{-1}_x)^{-1}(\Sigma_x^{-1}\mu_x + \Sigma'^{-1}_x\mu'_x)\\ \mu_y\\ \mu_{y'}\end{pmatrix} \text{ and } \boldsymbol{\Sigma} = \begin{pmatrix}(\Sigma_x^{-1} + \Sigma'^{-1}_x)^{-1} & 0 & 0\\ 0 & \Sigma_y & 0\\ 0 & 0 & \Sigma'_y\end{pmatrix}.$$

Moreover, the matrix $(\Sigma_x^{-1} + \Sigma'^{-1}_x)^{-1}$ is diagonal, and so is $\boldsymbol{\Sigma}$.

Proof. This proposition is a direct application of proposition 13. □

The previous proposition states that if covariance matrices are all supposed diagonal, then the resulting fused image has also a diagonal covariance. This boils down to assume that all the pixels are independent.

In the final image model, the mean at each pixel is simply a weighted average of all the expectations of the patches containing this pixel. The weights are given by the precisions of the marginals at these pixel. We recognize here a special case of the *weighted aggregation procedure* described in the introduction. The more precise an estimate is, the more it counts in the final estimate.

A more specific case is the one obtained when all covariance matrices are identical and proportional to the identity matrix. In this case, the covariance of the resulting image model will be simply a diagonal, counting for each pixel the number of patches it belongs to. The resulting expectation at a given pixel will be a simple average of all the expectations of the patches containing this pixel. This corresponds to the widely used *uniform aggregation*.

Finally, the limit case where each patch model has a covariance with infinite values except for its central pixel corresponds to the *central aggregation*.

2.4.2 Expected Patch Log Likelihood

More complex strategies including both patch restoration and aggregation into a single variational formulation have been considered in the literature. This is the case of the Expected Patch Log Likelihood (EPLL) of Zoran and Weiss, 2011. Starting from an image

$$\widehat{u} = Au + \epsilon, \tag{2.2}$$

degraded by a linear operator A and an i.i.d. Gaussian noise $\epsilon \sim \mathcal{N}(0, \sigma^2 Id)$, the authors reconstruct a restored version of $\widetilde{u}$ as one of the solutions of

$$\arg\min_u \frac{\lambda}{2\sigma^2}||Au - \widehat{u}||^2 - EPLL_f(u), \tag{2.3}$$

where $EPLL_f(u) = \sum_j \log f(x_j)$, with $\{x_j\}$ the set of all square patches of size $\sqrt{d} \times \sqrt{d}$ extracted from the image u and f a given prior density on the image patches.

The authors of Zoran and Weiss, 2011 interpret the quantity $EPLL_f(u)$ as the empirical expectation of the log-likelihood of a patch (up to a multiplicative factor $\frac{1}{N}$ with N the number of patches). This quantity has another intuitive interpretation, as highlighted in the following proposition, whose proof follows directly from Remark 10.

Proposition 18. *Let u be an image on the domain $\mathbf{\Omega}$ and assume that $\nu(dx) = f(x)dx$ is a prior on all square patches of size $\sqrt{d} \times \sqrt{d}$ with domain inside $\mathbf{\Omega}$. Define E the set of all of of these square patch models sharing the same distribution $f(x)dx$. Then if $\bar{P} = \bigodot_{P \in E} P = (\bar{\Omega}, \bar{f}(x)dx)$ is well defined, there is a constant c such that*

$$EPLL_f(u) = \log \bar{f}(u) + c.$$

The function $EPLL_f$ is the log of the density obtained by fusing all square patch models on the grid with the same prior $f(x)dx$. Up to a constant, it can thus be interpreted as the *log* of a prior $p(u)$ on the whole image u. Consequently, by choosing $\lambda = 1$ in equation (2.3), the solution of (2.3) can be interpreted as a maximum a posteriori and be written $\operatorname{argmax}_u \log p(u|\widehat{u})$ on the whole image, since the term $-\frac{\lambda}{2\sigma^2}||Au - \widehat{u}||^2$ is, up to a constant, equal to $\log p(\widehat{u}|u)$ under the white Gaussian noise assumption.

Propositions 8 and 18 also clarify the link between the $EPLL$ approach and the iterative conditioning strategies used for instance in Raad, Desolneux, and Morel, 2016 for texture synthesis. Indeed, the fused image prior used in $EPLL$ can be interpreted as a probability distribution of a global random image obtained by fusing all patch distributions, conditioning by their equality on all their intersections.

Now, consider the pure denoising case ($A = Id$). In this case, the solution of (2.3) can also be interpreted as a maximum likelihood for another fused distribution $\bar{\bar{f}}(x)dx$ on the whole image, as shown in the following distribution.

Proposition 19. *Keeping the notations of proposition 18, let $\bar{P} = \bigodot_{P \in E} P$ be the image model obtained by fusing all patch models of E. Let $P_{\widehat{u}} = \left(\mathbf{\Omega}, \mathcal{N}(\widehat{u}, \frac{\sigma^2}{\lambda} Id)\right)$ be an image model on the whole grid.*

Then if $(\mathbf{\Omega}, \tilde{f}(x)dx) := \bar{P} \odot P_{\widehat{u}}$, we have

$$\arg\min_u \frac{\lambda}{2\sigma^2}||u - \widehat{u}||^2 - EPLL_f(u) = \arg\max_u \tilde{f}(u).$$

Proof. We just have to remark that

$$\begin{aligned}\log \tilde{f}(u) &= \log \bar{f}(u) + \log\left(e^{-\lambda \frac{\|u-\widehat{u}\|^2}{2\sigma^2}}\right) + cst \\ &= EPLL_f(u) - \frac{\lambda}{2\sigma^2}||u - \widehat{u}||^2 + cst.\end{aligned}$$

□

In the light of this proposition, the result of the EPLL algorithm in the denoising case is simply the maximum likelihood of the probability distribution obtained by merging all the patch models with a large Gaussian model centered on the noisy image and with variance $\frac{\sigma^2}{\lambda}$.

Under the full degradation model (2.2), a last interpretation of (2.3) is possible, using the fusion of posterior patch models. To this aim, we have to assume that the degradation operator A is diagonal, which means that it acts separately on pixels.

The restriction of A to a domain Ω can thus be written $A_{|\Omega}$ and the model (2.2) restricted to Ω becomes

$$\widehat{u}_{|\Omega} = A_{|\Omega} u_{|\Omega} + \epsilon_{|\Omega}.$$

For a given patch model $P = (\Omega, f(x)dx)$ in E, the corresponding posterior patch model is just $(\Omega, f_{ap}(x)dx)$ where $f_{ap}(x)dx$ is the posterior obtained under this degradation model on Ω and the prior $f(x)dx$. For the sake of simplicity, we assume in the following proposition that each pixel is covered by exactly the same number of patch models. This is true if we assume that the image is periodic. In practice, it is satisfied for all pixels except those lying close to the image borders.

Proposition 20. *Keeping the notations of proposition 18, assume that each pixel of $\mathbf{\Omega}$ is covered by exactly d patch models of E. For each patch model $P = (\Omega, f(x)dx)$ in E, we define the corresponding posterior patch model as $P_{ap} = (\Omega, f_{ap}(x)dx)$ with*

$$f_{ap}(x) \propto f(x) \frac{1}{(2\pi)^{d/2}\sigma^d} e^{-\frac{1}{2\sigma^2}\|A_{|\Omega}x - \widehat{u}_{|\Omega}\|^2}.$$

We define E_{ap} the set of all these posterior patch models,

$$E_{ap} = \{(\Omega, f_{ap}(x)dx), \text{such that } (\Omega, f(x)dx) \in E\}.$$

Then, if $\bar{P}_{ap} = \bigodot_{P \in E_{ap}} P = (\mathbf{\Omega}, \bar{f}_{ap})$ is well defined, we have

$$\log \bar{f}_{ap}(u) = EPLL_f(u) - \frac{d}{2\sigma^2}||Au - \widehat{u}||^2 + cst.$$

Proof. We just have to remark that

$$\sum_{P \in E} \|A_{|\Omega} u_{|\Omega} - \widehat{u}_{|\Omega}\|^2 = d||Au - \widehat{u}||^2.$$

□

In other words, for $\lambda = d$, the solution of (2.3) is a maximum of a fused posterior model on the whole image, assuming that all patches have the same prior $f(x)dx$.

2.4.3 Bayesian Model Averaging

We can ask ourselves the question of the link between the fusion operation introduced in this chapter and the notion of Bayesian Model Averaging (BMA) Hoeting et al., 1999, which also attempts to combine information provided by different models on data. For the sake of simplicity, assume that we have two patch models P_1 and P_2 on the same domain Ω, and an observed degraded patch y on Ω. In the BMA framework, the a posteriori distribution of the (unknown) clean patch x can be written

$$p(x|y) = \sum_{k=1,2} p(x|y, P_k) p(P_k|y),$$

where each $p(x|y, P_k)$ is simply the a posteriori distribution of x knowing y for the patch model P_k. Since $p(P_k|y)$ is a scalar, the BMA of two posterior models is merely a linear combination of these models. It can be interpreted as a generalization of the weighted aggregation, but is different from the fusion operation.

2.5 Experiments

In this section, we illustrate the behavior of the fusion operation on different examples. We start with toy examples showing the main difference between the fusion and the classical uniform and weighted aggregations. Then we focus on the particular case of patch-based image denoising and we give some insight on the advantages and limitations of the fusion. We conclude with two simple extensions: the first one consists in mixing the fusion and the uniform aggregation in order to keep the best of both worlds, and the second one consists of a sparse fusion relying on a very few number of patch models.

Throughout this experimental section, we focus on the case of Gaussian distributions and we compute the expectations and covariances of the fused models explicitly, as explained in Section 2.3.3. However, let us underline that this is usually not the most efficient way to take advantage of the fused model. Indeed, we have seen in Remark 10 that the logarithm of the fused density can be written directly by summing the logarithms of these densities. As a consequence, it is very easy to integrate such a model in any variational framework, without any explicit computation of the fused model, even if this won't necessary yield a convex formulation (it will be convex for normal densities though).

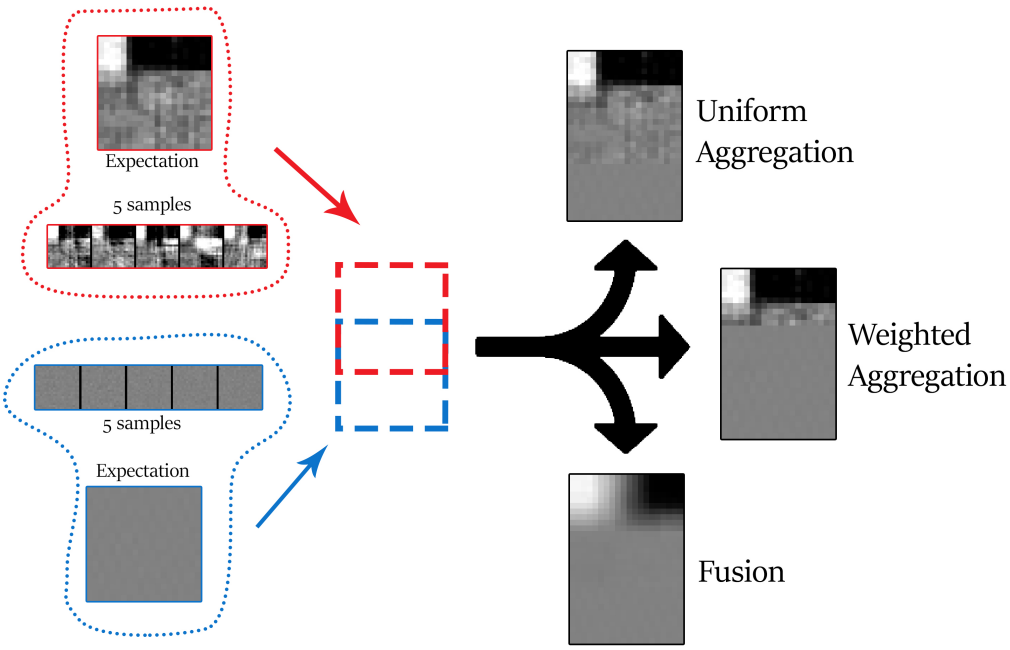
2.5.1 A toy example

Figure 2.12 shows two toy examples clarifying the difference between the fusion defined in this article on the one hand, and the weighted and uniform aggregation on the other hand. In these examples (a) and (b), two Gaussian patch models (shown respectively in red and blue on the left) are fused and the three aggregation strategies lead to quite different results. In both examples, the red model has a high variance and the blue model is more precise (or more confident, if we see patch models as persons with more or less solid opinions). On the right, we show only the expectations of the fused models.

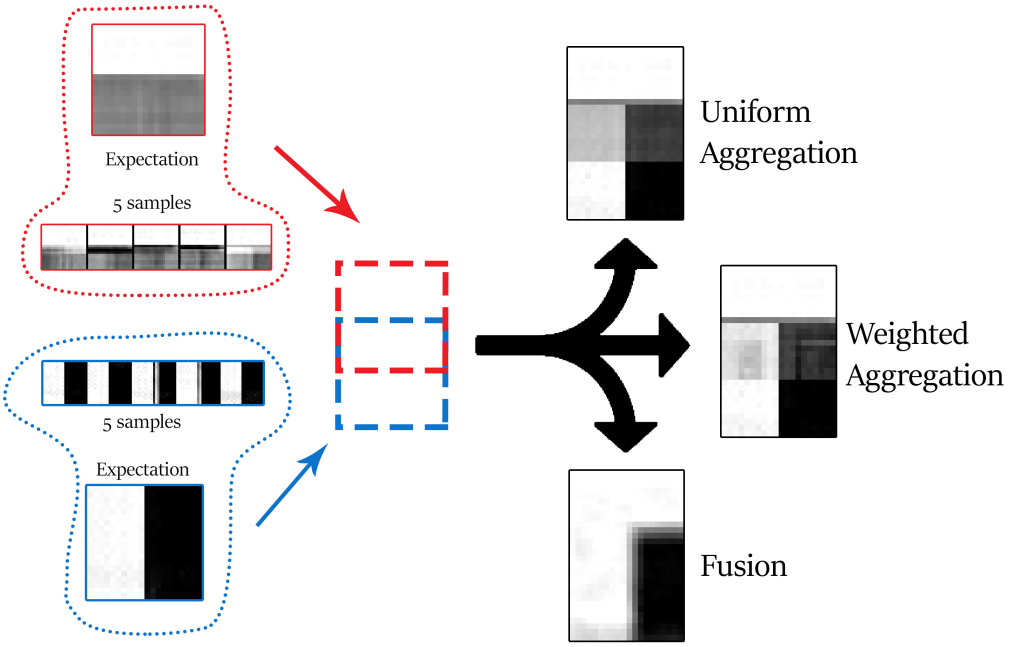
In both examples, the uniform aggregation gives the same credit to both patch models, whatever their covariances. In (a), although both patch models seem to almost agree on their overlap, this results in a quite noisy result on the patches overlap, even if the blue model has a very high precision in this region. The weighted aggregation takes into account this precision and yields a more satisfying result. The fusion operation also gives much more credit to the blue patch model than to the red one and yields a much smoother result. As we shall see in the section devoted to image denoising, this behavior permits to obtain very regular results, free from the usual artifacts created by standard aggregation procedures, but at the price of some blur.

In example (b), both patch models strongly disagree on their overlap. The uniform aggregation yields a result which can be seen as a compromise between their opinions but is in contradiction with both of them. The weighted aggregation takes into account the greater precision of the blue model but still yields a result which is quite unlikely from both models point of views. Again the fusion yields a quite smooth result, which is likely for both patch models (even if it is more likely for the blue model than for the red one).

Notice that the uniform and weighted aggregations do not update the pixels outside of the patch overlap zone, while the fusion operation also affects these pixels, as shown in Proposition 13.



(A) In this case, the two patch models almost agree, but the red one has a very large variance compared to the blue one.



(B) In this case, the two patch models completely disagree.

FIGURE 2.12: Illustration of the behavior of the different aggregation schemes for two adjacent Gaussian patch models. On both figures, the two patch models on the left are aggregated in three different ways to form the patches on the right, either with the uniform aggregation, the weighted aggregation (taking into account the precision of each pixel) and the fusion operation.

NL-Bayes				
Aggregation	*Uniform*	*Weighted*	*Fusion* $\lambda = d$	*Fusion* $\lambda = 10d$
Lena	30.58	30.49	30.28	**30.66**
Barbara	28.99	28.94	28.83	**29.04**
Cartoon	30.04	29.98	29.57	**30.35**
Squares	45.28	46.87	**47.35**	46.54
EPLL Model				
Aggregation	*Uniform*	*Weighted*	*Fusion* $\lambda = d$	*Half Quadratic Splitting*
Lena	30.69	30.42	29.88	**30.71**
Barbara	26.56	26.18	25.45	**27.55**
Cartoon	29.89	29.62	28.65	**30.49**
Squares	37.38	39.09	36.96	**39.51**
HDMI				
Aggregation	*Uniform*	*Weighted*	*Fusion* $\lambda = d$	*Fusion* $\lambda = 10d$
Lena	**31.12**	31.10	28.16	29.96
Barbara	**29.55**	29.54	25.57	28.72
Cartoon	**30.55**	30.52	25.67	29.34
Squares	44.24	**48.77**	46.37	35.62

FIGURE 2.13: PSNR of the different aggregation methods with NL-Bayes inference.

2.5.2 Application to denoising

For the sake of simplicity, we restrict the rest of our experiments to denoising problems. We also restrict our experiments to the case where the patches of E are all square patches of size $\sqrt{d} \times \sqrt{d}$ in $\mathbf{\Omega}$.

We recall here the patch-based restoration framework applied to denoising. In image denoising, in order to restore an unknown image u from its noisy version $u + \varepsilon$, we usually start by extracting all square patches $\{y_k,\ k \in \{1, \dots, |\mathbf{\Omega}|\}\}$ from $\widehat{u} = u + \epsilon$. The noise model on patches can be written

$$y_k = x_k + \epsilon_k,$$

with x_k the (unknown) patch before degradation. As explained in Section 1.2.2, we will assume that the noise is i.i.d Gaussian of variance σ^2.

In this situation, Bayesian patch-based methods use a common restoration framework to restore u from $u + \epsilon$:

1. **Estimation**: estimate a prior density f_k for each clean patch x_k.
2. **Restoration**: compute a denoised version $\hat{x}_k$ from y_k using the knowledge of the noise model and the prior f_k.
3. **Aggregation**: reconstruct a whole denoised image $\hat{u}$ from the set of denoised patches $\{\hat{x}_k,\ k \in \{1, \dots, |\mathbf{\Omega}|\}\}$.

The restoration step can for instance take the form of a maximum a posteriori

$$\widetilde{x_k} = \operatorname{argmax}_x \frac{1}{2\sigma^2} \|y_k - x\|^2 - \log f_k(x).$$

Several methods in the literature use the previous restoration scheme, with slight variations. In the following sections, we will focus on three of them, which are representative of different choices in the three previously mentioned steps:

- NL-Bayes (Lebrun, Buades, and Morel, 2013), which estimates a specific Gaussian model $\mathcal{N}(\mu_k, \Sigma_k)$ for each patch x_k (see Section 1.2.4).
- HDMI (Houdard, Bouveyron, and Delon, 2017), which estimates a low-dimensional Gaussian Mixture model for the whole set of patches x_k, $k \in \{1, \dots, |\mathbf{\Omega}|\}$ (see Section 1.2.4).
- EPLL (Zoran and Weiss, 2011), which estimates a Gaussian Mixture model for patches on an external database, and replaces steps 2 and 3 above by the variational problem (2.3) and solves it by Half Quadratic Splitting. (see Section 1.2.4).

All of these methods yield a prior model f_k for each patch x_k. In the case of Gaussian Mixture Models, for the sake of simplicity, we choose to keep as a prior for x_k the Gaussian component which is the most likely for x_k.

Since the noise model is also Gaussian, these methods also yield Gaussian posterior models for each patch. We write these posteriors $\tilde{f}_k$, and

$$\tilde{f}_k(x|y_k) \propto f_k(x) e^{-\frac{\|x-y_k\|^2}{2\sigma^2}}.$$

In the following, we will illustrate how these prior or posterior models can be fused using the framework introduced in the previous sections. If we compute a fused prior model, the maximum a posteriori under the noise degradation model can be used to restore the image. In other words, if $\bar{f}$ is the fused image model density, the restored image is computed as the solution of

$$\operatorname{argmin}_u \frac{1}{2\sigma^2}\|u - \widehat{u}\|^2 - \log \bar{f}(u). \tag{2.4}$$

If instead we compute a fused posterior model $\bar{f}(u|\widehat{u})$, the restored image can be computed directly as the maximum of this posterior, *i.e.*

$$\operatorname{argmax}_u \bar{f}(u|\widehat{u}).$$

Now, writing x_k for the patches of u,

$$\begin{aligned} -\log \bar{f}(u|\widehat{u}) &= -\log \prod_{k=1}^{|\mathbf{\Omega}|} f_k(x_k|y_k) \\ &= -\sum_{k=1}^{|\mathbf{\Omega}|} \log f_k(x_k) + \sum_{k=1}^{|\mathbf{\Omega}|} \frac{\|x_k - y_k\|^2}{2\sigma^2} \\ &= -\log \bar{f}(u) + d\frac{\|u - \widehat{u}\|^2}{2\sigma^2}. \end{aligned}$$

Thus, both strategies boil down to minimize an energy of the same form

$$\operatorname{argmin}_u \frac{\lambda}{2\sigma^2}\|u - \widehat{u}\|^2 - \log \bar{f}(u), \tag{2.5}$$

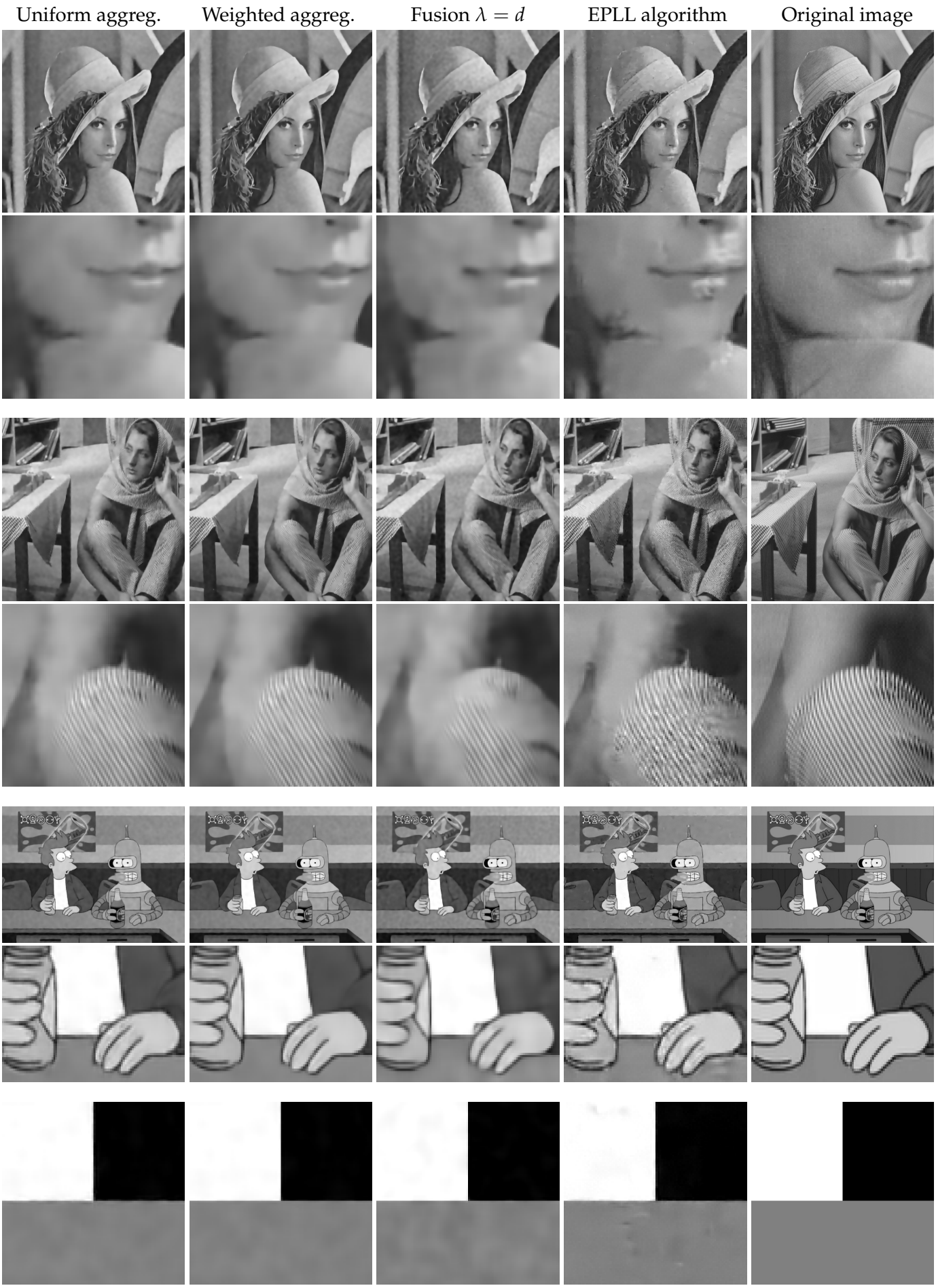


FIGURE 2.14: Comparison of the different aggregation procedures on the 4 test images using EPLL for the inferrence of Gaussian models. Images are degraded by a noise of standard deviation $\sigma = 30$.

with different values of λ. The value $\lambda = 0$ corresponds to the fusion of the prior models and $\lambda = d$ corresponds to the fusion of the posterior models. Fusing with higher values of λ gives much more weight to the noisy image $\widehat{u}$.

2.5.3 Results for the three different inference methods

Experiments are led on four different 512×512 images, *Lena*, *Barbara*, *Cartoon* and *Squares*. We will see that the behaviors of the different aggregation procedures strongly depend both on the image content and on the way patch Gaussian models are inferred from the noisy data.

For each of the three inference methods described in the previous section, we provide different visual results illustrating the visual effects of all aggregation strategies. PSNR values are also provided in Table 2.13. As we shall see, while the complete fusion operation is not really competitive PSNR-wise, it leads interesting visual results, quite different from the simpler aggregation strategies. Our goal here is to study and highlight these different behaviors.

NL-Bayes

The NL-Bayes algorithm infers a specific Gaussian models for each patch and uses small patches (5×5). As a consequence, most Gaussian covariances are quite well approximated by their diagonal, and the different aggregation procedures only display minor differences on natural images. Table 2.13 shows that the fusion slightly improves the PSNR results, but the difference is more significant for very simple geometric images like *squares*, even if the visual differences are quite subtle and concentrated around the junctions and edges of the rectangles.

EPLL

The EPLL model (Zoran and Weiss, 2011) makes use of 8×8 patches and learns a Gaussian mixture model with 200 groups on a large external set of images. In the original paper, patches are centered (their DC component is removed) before processing and all the Gaussian models from the GMM are assumed to have zero means. Additionally, in order to minimize (2.3), the authors of Zoran and Weiss, 2011 introduce an auxiliary variable and make use of Half-Quadratic Splitting, which means that the restoration iterates between patch estimation and image reconstruction (by a uniform aggregation). In consequence, while the $EPLL_f$ term is a particular case of the fusion operation, their model is not directly comparable to our framework.

First, in order to use the closed-form expressions of Proposition 13, we need a Gaussian model for each patch and not a full GMM. As explained above, we chose to keep as a prior for each patch the Gaussian of the mixture which is the more likely for it. However, observe that by making this choice we loose some of the information of the full GMM model. Second, we need a model on patches, and not on centered patches. To cope with this limitation, we remove the mean value of each patch, select the most likely Gaussian component in the GMM, and give the mean value of the original patch to this Gaussian model. As a result, the expectations of the different Gaussian models contain an important low frequency noise. For these different reasons, the comparison of the different aggregation strategies with the EPLL algorithm (which makes the fusion on the full GMM model) should be made with caution. Figure 2.14 provides the result of these different strategies for the images *Lena* and *Barbara* with $\sigma = 30$. The results of the fusion operation on these

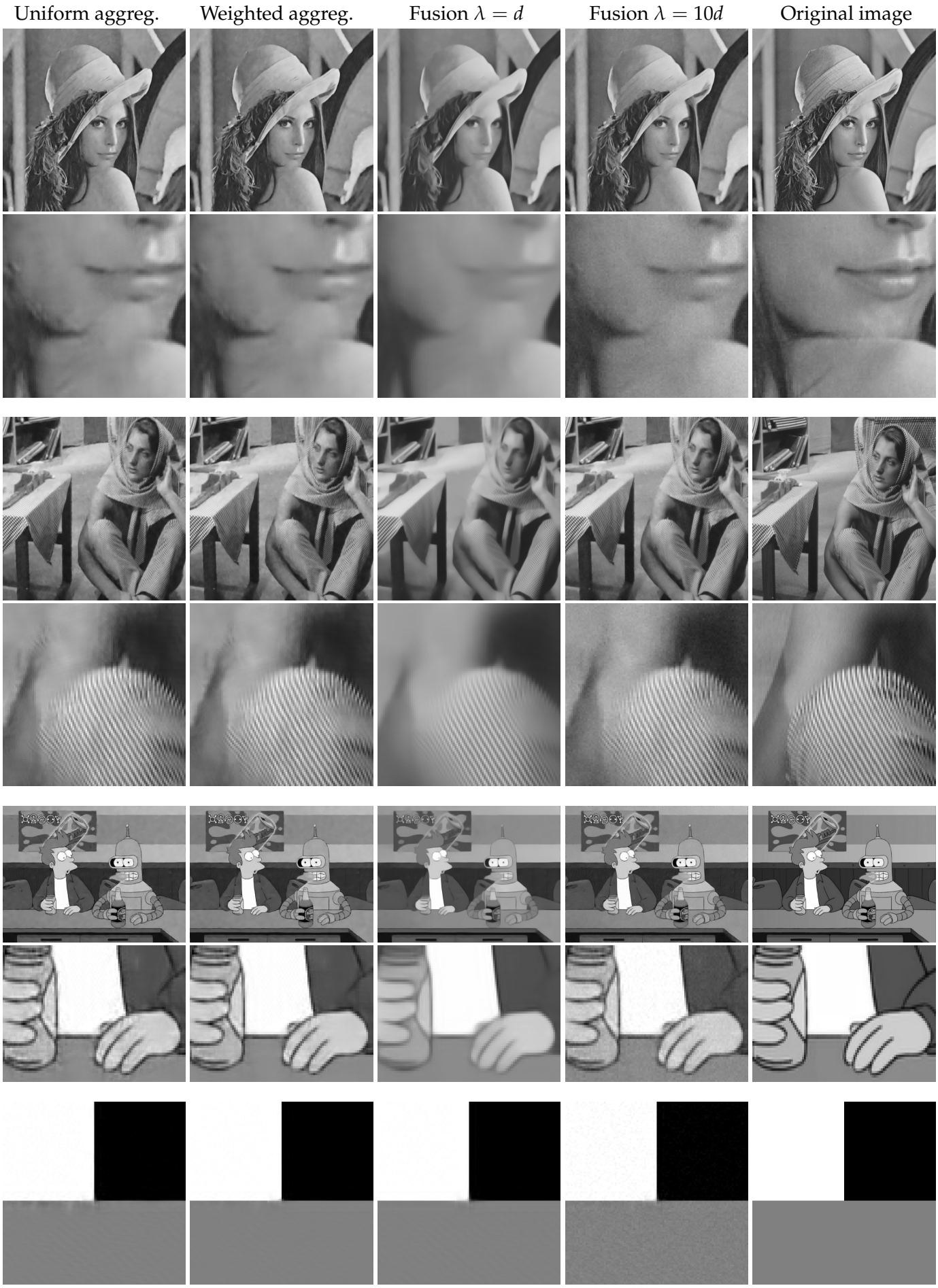


FIGURE 2.15: Comparison of the different aggregation procedures on the 4 test images, using HDMI for the inferrence of Gaussian models. Images are degraded by a noise of standard deviation $\sigma = 30$.

models are very smooth but present what we call a "fluffy" effect (see Section 2.5.4 and Figure 2.16), due to the way the Gaussian means are handled.

HDMI

In the HDMI algorithm (Houdard, Bouveyron, and Delon, 2017), a GMM is learned on 10×10 patches, with only a few dozens of low-dimensional Gaussian models in the mixture. Again, we keep as a prior for each patch the Gaussian of the mixture which is the more likely for it, so we loose part of the richness of the original model in our experiments. Figure 2.15 provides the result of the different strategies for the images *Lena* and *Barbara* with $\sigma = 30$. In this case, the different aggregations procedures produce quite important differences. The uniform aggregation is efficient PSNR-wise, but suffers from numerous artifacts (see Section 2.5.4). Using the whole fused model provides results which are below PSNR-wise but are also much smoother, and removes numerous artifacts. Some of the results suffer from a loss of contrast, explained in Section 2.5.4.

It is noticeable that the fusion operation tends to improve the results for the models inferred by NL-Bayes while it does not for the HDMI and EPLL models, at least PSNR-wise. We think that it can be partly explained by the fact that at this point, we are able to take fully into account the Gaussian models inferred by NL-Bayes and that it is not the case for the GMM in HDMI and EPLL.

2.5.4 Visual effects

Fluffy effect

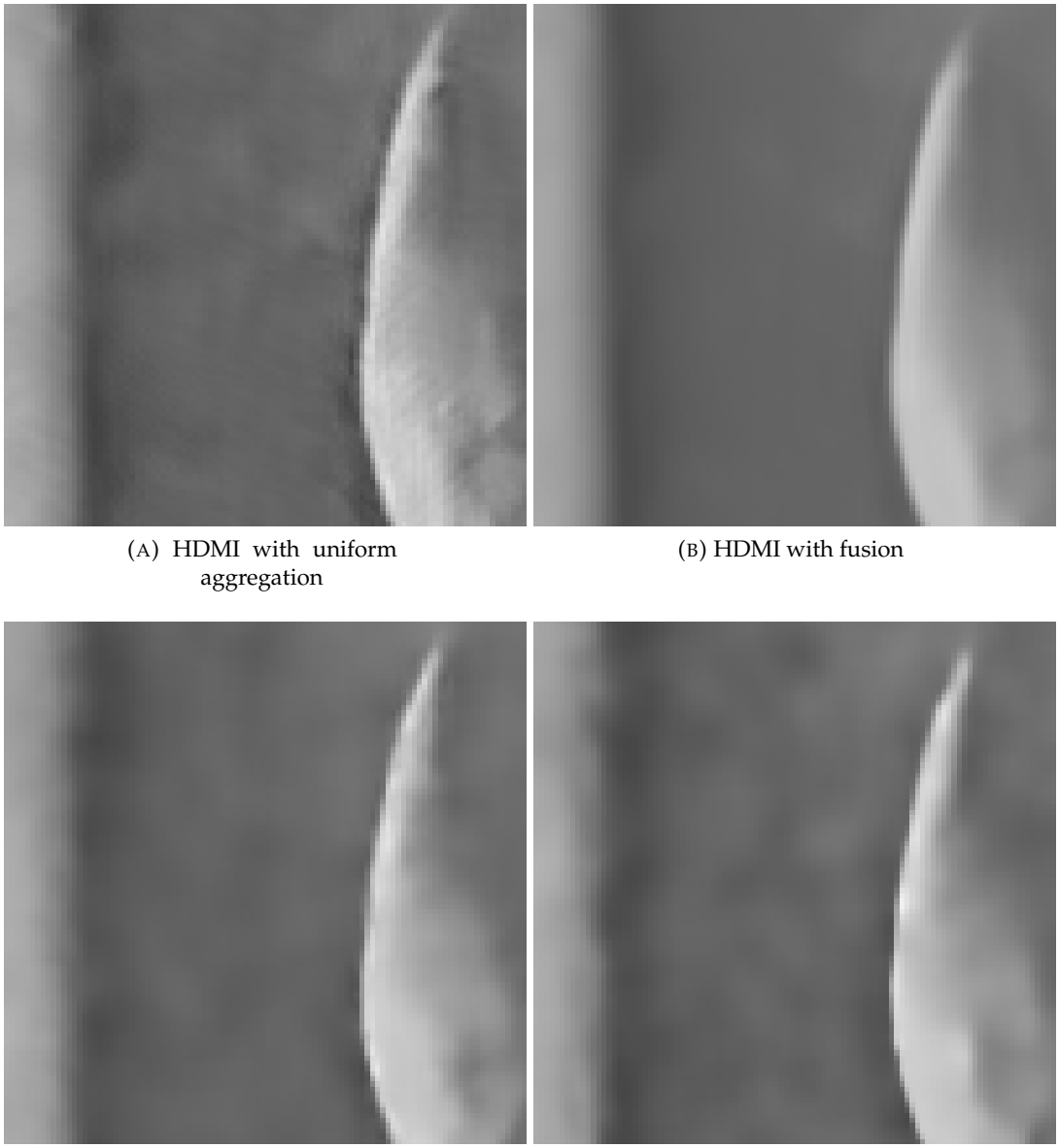
We call *fluffy effect* the effect visually similar to cotton, appearing in constant regions when using single scale patch-based methods. This effect was presented in Section 2.1.2.

As we can see in Figure 2.16, when using HDMI Houdard, Bouveyron, and Delon, 2017, the fusion aggregation clearly reduces this defect, whereas it does increase when using EPLL Zoran and Weiss, 2011. We can explain these results as follows: in HDMI, patch priors have (almost) independent expectations, since they are inferred using numerous different patches on the whole image. The remaining inconsistencies between overlapping patch models are thus removed by the fusion. With EPLL, since each noisy patch has its own DC component as a model expectation, and since these DC components are not independent for overlapping patches, the white noise low frequencies are reinforced by the fusion and the results show a very pronounced fluffy effect.

When the model is appropriate, the fusion aggregation is a solution to the problems raised in Section 2.1.2, namely the fluffy effect and the visual artifacts as shown in Figure 2.16. It still assumes that patches are independent, but it takes into account the influence of their overlap on their reconstruction, by forcing them to agree. Nevertheless, this constrains the patch models to reduce their possibilities, which results in a blur and a loss of contrast, as developed in Section 2.5.4 (blur and contrast).

Artifacts

The main advantage of the fusion aggregation is to reduce the artifacts. This is quite understandable, since the method creates a model for which all overlapping patches



(A) HDMI with uniform aggregation

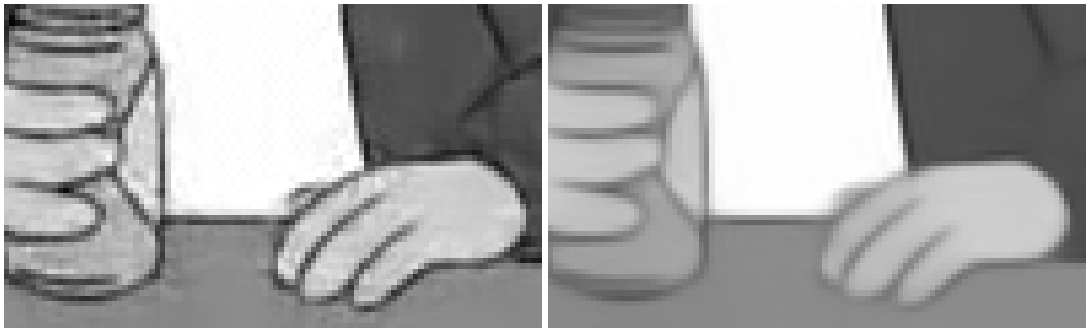
(B) HDMI with fusion

(C) EPLL model with uniform aggregation

(D) EPLL model with fusion

FIGURE 2.16: Influence of the fusion on the fluffy effect (low frequency noise visually similar to cotton and visible in constant regions after patch-based processing). On the first line, when using the HDMI algorithm, the fluffy effect is highly reduced by the fusion, since the hypothesis that the patch models are independent is almost fulfilled. On the contrary, when using EPLL, the average of the noisy patch becomes the expectation of its prior model. In this case, the fluffy effect is amplified by the fusion.

have to agree. An artifact is created when one or several of the original overlapping patch models are badly estimated. In this case, even if the uniform aggregation averages several correct estimates with this wrong estimate, the artifact can remain noticeable. When using the fusion approach, if this artifact is inconsistent with the other models, it will completely disappear. This is illustrated by Figure 2.17.



(A) HDMI with uniform aggregation

(B) HDMI with fusion

FIGURE 2.17: Examples of artifacts created in patch-based image denoising. On the left, we can see that the uniform aggregation creates numerous artifacts, for instance around the fingers. These artifacts, which are inconsistent across overlapping patch models, are not present in the fusion result.

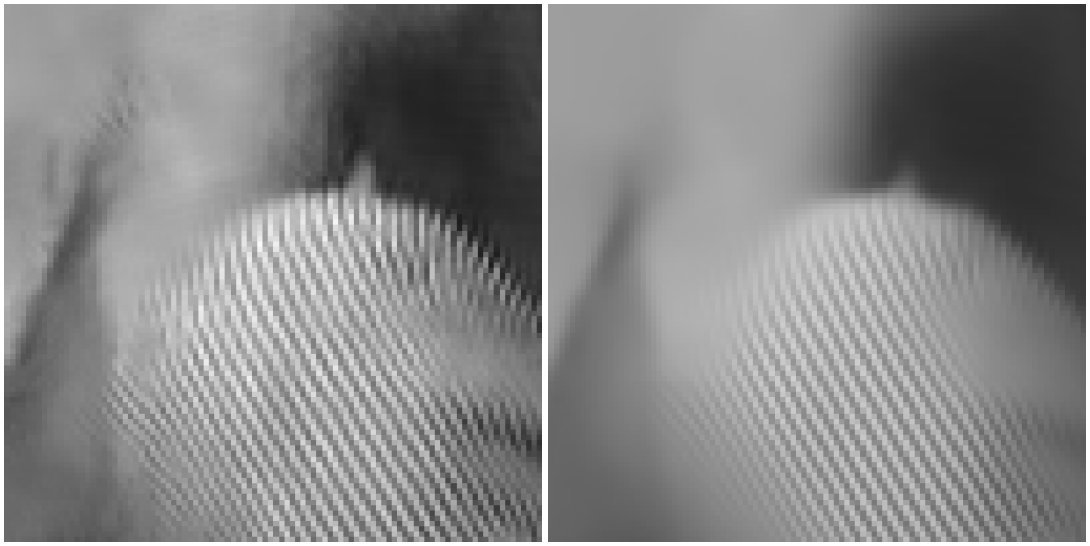
Blur and constrast

The main drawback of the fusion operation is a loss of constrast and sharpness around some geometric structures, which makes the PSNR decrease. This is particularly striking in regions where the patch models are not well learned. In practice, flat patch models tend to come with higher precisions than patch models representing geometric structures or contrasted textures. If, across an edge or a geometric structure, some patches are wrongly attributed to a flat patch model, this model will count significantly more than others in the fusion operation, and will result in an important contrast loss. These shortcomings can be reduced by increasing the weight λ of the data term in the final restoration (Equation 2.5), at the cost of a slight increase of noise. Besides, if a texture or an edge is not captured by the model, then the fusion cannot restore it properly and the resulting texture will appear blurry. This effect is illustrated on Figure 2.18 and can be reduced for instance by using the information of the fused model, see Section 2.5.5.

2.5.5 Possible extensions

Precision estimate

As we have seen, the fusion yields good results in regions where the estimated model is confident and has been well trained. This "confidence" level can be accessed through the covariance of the fused model. A simple way to exploit it is to consider the precision of the marginal at a given pixel. If this precision is high, we can consider to keep the estimate provided by the fusion, and use another estimate otherwise, like the uniform aggregation. This way, we can construct the *precision estimate*, defined as an average of the images obtained by the uniform aggregation and the fusion, weighted by the precision of the marginals for each pixel. This idea is illustrated on Figure 2.19. The figure shows the precision map obtained on *Lena* with the Gaussian models of the HDMI algorithm, and the resulting *precision estimate*, which clearly keeps the best of both worlds, reducing the artifacts of the uniform aggregation but providing a much less blurry result than the sole fusion.



(A) HDMI with uniform aggregation

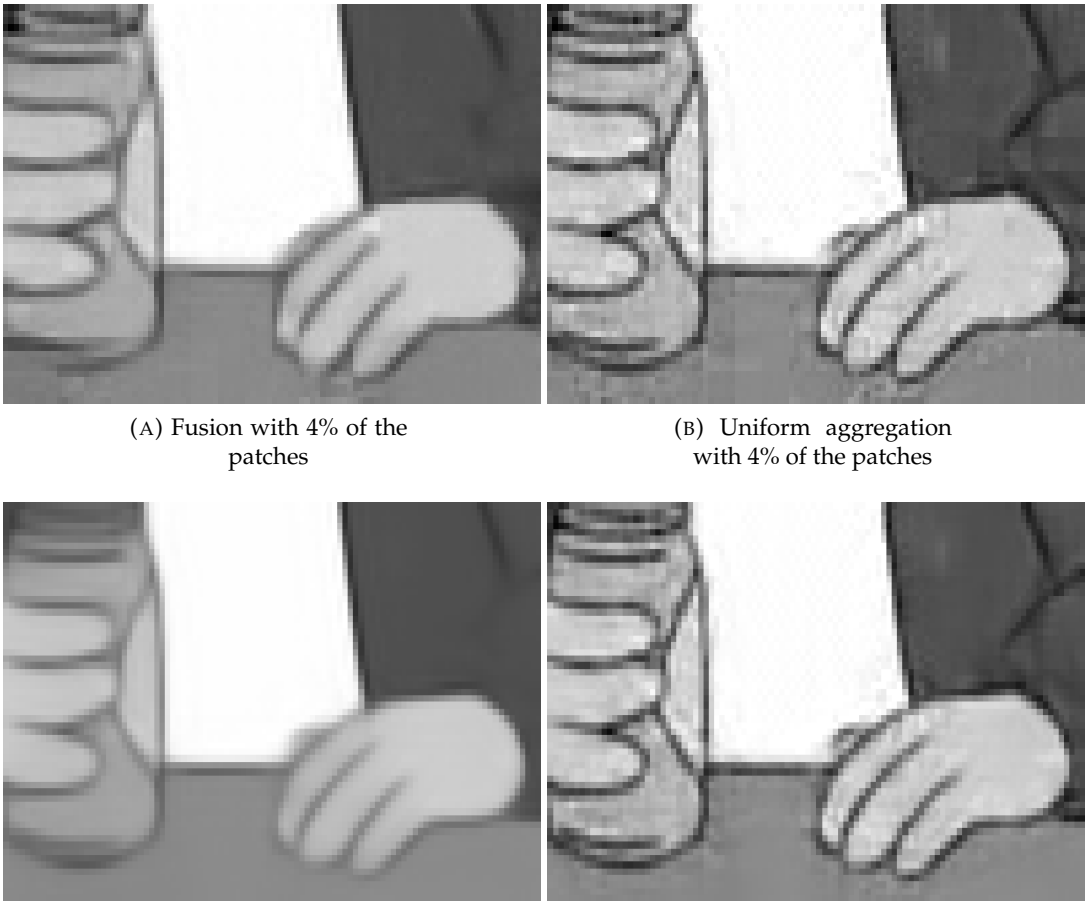
(B) HDMI with fusion

FIGURE 2.18: Illustration of the loss of sharpness and constrast due to the fusion operation. As we can see, the stripes of the legs of Barbara are perfectly restored by the fusion aggregation, since the model is well-trained on this region. However, on the sides of the leg, the texture looks blurry. This is explained by the lack of patch examples for this geometry. The bound of the shadow on the arm is also well-restored by the fusion, but the "dark spot" on the elbow is another good example of contrast loss: many patch models of this region are considered to be uniform and highly reduce the obscurity of the area.

(A) Precision map (B) Uniform aggregation

(C) Fusion (D) Precision estimate, mix of (b) and (c)

FIGURE 2.19: The inverse of the diagonal of the covariance matrix gives us the precision of the marginal of each pixel. This is a basic estimate of how confident the model is for each pixel. This enables to compute the precision estimate, which tries to keep the best of both worlds.



(A) Fusion with 4% of the patches

(B) Uniform aggregation with 4% of the patches

(C) Fusion with all the patches

(D) Uniform aggregation with all the patches

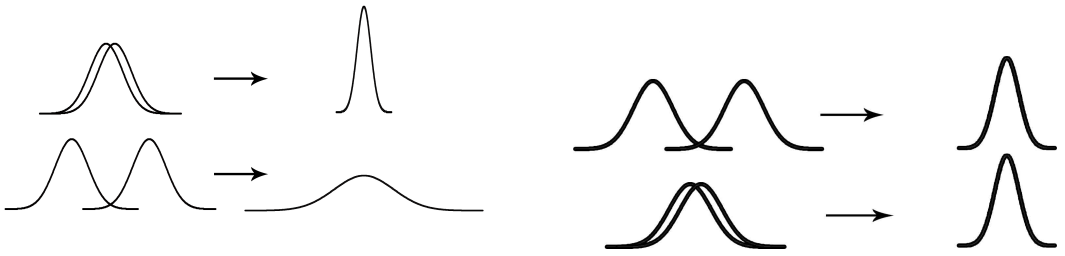
FIGURE 2.20: Comparison of different aggregation procedures on the *Cartoon* image. On the top row, only 4% of the patches are used, and on the bottom row, all the patches are used. On the left the fusion, and on the right the uniform aggregation

Sparse aggregation

The fusion does not need numerous patches at each pixel to achieve visually smooth results. The image can therefore be reconstructed using a reduced number of patches, chosen either at random or using some heuristics to select the best model among them, as in Tabti et al., 2014 for instance. This could be a way to speed up the learning phase, or to spend more time learning more complicated models. Figure 2.20 shows an example of a simple sparse aggregation, using only 4% of the patches (of size 10×10), so that each pixel belongs to only 4 patches.

2.5.6 Limitations

If we look closely at the formula to merge Gaussian patches presented in Proposition 13, we can see that the resulting covariance matrix of the fusion does not depend on the expectations of the two Gaussian distributions. This is problematic. Intuitively, a compromise made between converging opinions (patches) should be considered with more assurance than one made between diverging opinions. This would be the behavior of an ideal fusion, which would more efficiently deal with "isolated"



(A) Ideal behavior

(B) Behavior of the proposed fusion

FIGURE 2.21: Schematic comparison of the behavior of the proposed fusion and the ideal behavior of the patch model fusion.

opinions. This behavior is presented in Figure 2.21. In comparison, the proposed fusion does not adapt to the proximity of opinion of the patches. This would result in an influence of the original expectations on the value of the resulting covariance matrix when merging two Gaussian patches.

2.6 Conclusion

We have presented a new way of aggregating patches in the Bayesian framework of patch-based methods. This study was motivated by the problem raised in Section 2.1 and the experiment of Figure 2.7. We have shown that Bayesian patch-based methods were mainly limited by the non independence of the patches during the aggregation step. We constructed an aggregation procedure based on more realistic hypothesis than the independence of the patches and on a deeper utilization of the model inferred in the editing step. This has led us to introduce a new formal definition of a patch model, and the notion of agreement between overlapping patches. We have built on this notion to propose a general common framework for the aggregation operation, seen as a fusion of different overlapping patch models. As we have shown, this common framework includes all previous aggregation schemes used in the literature, and reduces the design of new ones to the design of a fusion operation.

Our approach also permits to compute a fused image model which generalizes the Expected Patch Log Likelihood introduced by Zoran and Weiss, 2011. When patches are assumed to follow Gaussian distributions, this fused model is also Gaussian, with tractable expectation and covariance. This whole fused model can in turn be used to restore the whole image. In practice, the fusion operation can be used for any model which leads to tractable computations.

We have compared experimentally several special cases of this fusion operation for patch-based image denoising. As we have seen, using the fusion to aggregate does not necessarily improve the result PSNR-wise, but highly reduces the visual artifacts and the fluffy effects, identified as the main reconstruction issues of standard aggregation procedures in Section 2.1. On some images, it sometimes outperforms the standard uniform and weighted aggregations. The fusion is preferable if the model is well trained, since it takes advantage of all the provided information. However, it has some severe limitations, like the fact that estimates are less contrasted and sharp, and the computational time.

The proposed patch fusion is a first idea toward more general and efficient aggregation schemes. It was motivated by the prior knowledge that the patches should agree. Even if this fusion has some nice properties and intuitive interpretations, its

behavior is not ideal, as explained in Section 2.5.6. This suggests the introduction of a new fusion operation that would follow a more suitable behavior, and naturally led us to the optimal transport theory, which is a powerful and well-studied way to interpolate and compare distributions. Merging patch models, as formalized in this Chapter, with optimal transport theory can be formulated as a generalization of Wasserstein barycenter. Even if some applications of optimal transport theory to tomographic reconstruction led to similar considerations (see Abraham et al., 2017), it remains a particular case of this new problem, which has not yet been addressed in the literature. The next part present a study of this problem, that we called *Generalized Wasserstein Barycenter* (GWB).

Part II

Generalized Wasserstein Barycenter

Chapter 3

Requirements: Optimal Transport

Introduction

This chapter presents some basis of the optimal transport theory which will be useful to expose the work presented in Chapter 4. I will mostly present the problems motivating the theory and the main theoretical and computational tools to solve them. This includes the Kantorovich duality, the network simplex algorithm, the entropic regularization and finally the multi-marginal formulation of optimal transport.

3.1 General optimal transport

3.1.1 Kantorovitvh formulation

Optimal transport is the mathematical answer to a very basic and practical question: how to displace mass from a pile of soil to another with the lowest cost possible, as roughly illustrated on Figure 3.1. There are several ways to formulate this mathematically.

Before everything, we need two spaces X and Y where the piles are located. A natural way to mathematically model a pile of soil is using a distribution. The pile we want to move will be denoted by μ and the target pile we want to obtain by ν. In order to be able to move μ to ν, we must ensure that they have the same mass, so we have to assume that

$$\mu[X] = \nu[Y]$$

So without loss of generality, we can assume that μ and ν have a unitary mass, and hence that they are probability distributions. The space of Borel probability distributions of a space E is denoted by $\mathcal{P}(E)$.

Then, we have to define what we mean by "displace". We are only interested in the theoretical procedure of moving the soil and therefore what really matters is the

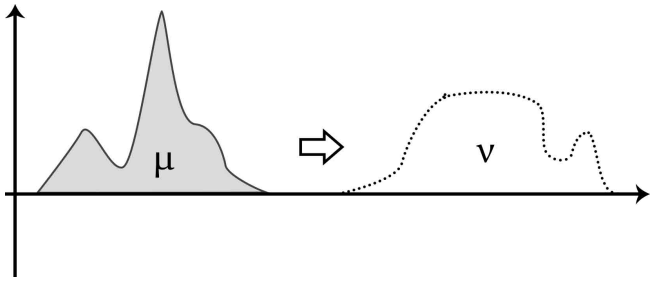


FIGURE 3.1: The goal of optimal transport is to find a way to displace the first distribution μ to the target distribution ν with the lowest possible cost.

cost of displacement. Hence, we model the displacement (or rather its consequence) by a cost function

$$c : X \times Y \to \mathbb{R}_+.$$

In the real world, $c(x, y)$ would correspond to the total amount of energy needed to load a unity (1kg for instance) of soil at the location $x \in X$, to move it to $y \in Y$ and unload it there, or the price that we would pay someone to do it, in a more capitalist mindset.

We then need to model our goal, i.e. the procedure to follow in order to move the first pile to the other. A natural way to do it is to consider, for each location $x \in X$, to which location $y \in Y$ we have to displace it. This implies defining a function

$$T : X \to Y,$$

that we call the *displacement map*. This modelization leads to the Monge formulation, presented in Section 3.1.2. However, this formulation excludes the possibility to divide the mass. For instance, if all the soil lies in $x \in X$, this would imply that we can only move this pile to another single point pile. The Monge formulation is therefore a bit restrictive, and we shall relax it to the so-called Kantorovich formulation. This is the same idea, with the additional assumption that we should be able to divide the mass. Instead of asking us where to move the mass lying in $x \in X$, we shall instead ask how much mass lying in x should we move to $y \in Y$. This implies having what we call a *transport plan*, i.e. a distribution γ on $X \times Y$. For $x \in X$ and $y \in Y$, $\gamma(x, y)$ would correspond to the amount of mass displaced from x to y.

But, so that the transport plan γ remains acceptable, we need to ensure that we would actually end up having all the mass of μ exactly displaced to ν if we follow the plan γ. This means that γ should satisfy

$$\forall A \subset X, \gamma[A, Y] = \mu[A] \text{ and } \forall B \subset Y, \gamma[X, B] = \nu[B] \tag{3.1}$$

Formally, we say that μ and ν are the *marginals* of γ. We can also write it by

$$\forall (\phi, \psi) \in L^1(d\mu) \times L^1(d\nu), \int_{X\times Y} \phi(x) d\gamma(x, y) = \int_X \phi(x) d\mu(x) \\ \text{and } \int_{X\times Y} \psi(y)\gamma(x, y) = \int_Y \psi(y) d\nu(y) \tag{3.2}$$

and also more succinctly by

$$\forall (\phi, \psi) \in L^1(d\mu) \times L^1(d\nu), \int_{X\times Y} (\phi(x) + \psi(y)) d\gamma(x, y) = \int_X \phi(x) d\mu(x) + \int_Y \psi(y) d\nu(y) \tag{3.3}$$

The set of all acceptable γ satisfying either (3.1), (3.2) or (3.3) is denoted by $\Pi(\mu, \nu)$.

Finally, "to transport" means to move all the mass of μ to ν according to the displacement procedure or transport plan γ. The cost of the displacement is the sum of all the infinitesimal costs of moving the mass $\gamma(x, y)$ from x to y. The transport cost of a transport plan γ is therefore

$$\int_{(x,y)\in X\times Y} c(x, y) d\gamma(x, y).$$

We can now define formally the Kantorovich formulation of the optimal transport problem :

Problem 1 (Kantorovich formulation)**.** Given two probability measures $\mu \in \mathcal{P}(X)$ and $\nu \in \mathcal{P}(Y)$, and a cost function $c : X \times Y \to \mathbb{R}_+$, find

$$\inf_{\gamma \in \Pi(\mu,\nu)} \int_{X\times Y} c d\gamma \tag{3.4}$$

with $\Pi(\mu,\nu)$ the *set of admissible transport plans*, defined by

$$\Pi(\mu,\nu) = \left\{ \gamma \in \mathcal{M}(X\times Y); \begin{cases} \forall A \subset X, \gamma[A\times Y] = \mu[A] \\ \forall B \subset Y, \gamma[X\times B] = \nu[B] \end{cases} \right\} \tag{3.5}$$

The value of (3.4) is called the *transport cost* and will be denoted by $\mathcal{L}_c(\mu,\nu)$.

Problem 2 (Probabilistic interpretation)**.** Given two probability measures $\mu \in \mathcal{P}(X)$ and $\nu \in \mathcal{P}(Y)$, and a cost function $c : X \times Y \to \mathbb{R}_+$, the Kantorovich optimal transport problem can equivalently be formulated as

$$\inf_{(U,V)\in\widetilde{\Pi(\mu,\nu)}} \mathbb{E}[c(U,V)]$$

with $\widetilde{\Pi(\mu,\nu)}$ is the set of all pairs of random variables (U,V) such that $U \sim \mu$ and $V \sim \nu$.

3.1.2 Monge formulation

The first formulation of the optimal transport problem was introduced by Monge. He gave his name to the so-called Monge formulation. As we saw in the previous section, it consists in assuming that we cannot split mass and implies defining a displacement map $T : X \to Y$.

Again, for a given displacement map, we need to ensure that it displaces correctly μ to ν. It means that the total mass displaced to y following the procedure implied by T corresponds to the actual mass we need to move to the location:

$$\forall B \subset Y, \nu[B] = \mu[T^{-1}(B)]. \tag{3.6}$$

We write equality (3.6) as

$$\nu = T\#\mu$$

and we say that ν is the *push-forward* of μ by T. This leads to the Monge formulation of the optimal transport problem

Definition 21. Given two probability measure $\mu \in \mathcal{P}(X)$ and $\nu \in \mathcal{P}(Y)$, and a cost function $c : X \times Y \to \mathbb{R}_+$, find

$$\inf_{T;\nu=T\#\mu} \int_{x\in X} c(x,T(x)) d\mu(x). \tag{3.7}$$

The Monge formulation is finally a particular and degenerated case of the Kantorovitch formulation. We can indeed notice that every displacement map T induce a transport plan defined by $\forall (x,y) \in X\times Y, \gamma_T(x,y) = \delta_{y=T(x)}\mu(x)$, which indicates to move the mass $\mu(x)$ from x to y if $y = T(x)$ and nothing otherwise. This is

equivalent to

$$\forall \phi \in L(d\gamma), \int_{X\times Y} \phi(x,y) d\gamma_T(x,y) = \int_X \phi(x, T(x)) d\mu(x).$$

We could define other problems as well, for instance assuming that each target location $y \in Y$ should receive the mass from only one location. The Monge formulation is often presented for historical reasons and because it remains quite intuitive, but most of the results and development of optimal transport rely on the Kantorovich formulation.

Since the Kantorovich formulation is a relaxation of the Monge formulation, we have always

$$\mathcal{L}_c(\mu,\nu) \leq \mathcal{L}_c^{\text{Monge}}(\mu,\nu).$$

But it appears that the solutions of the two problems coincide in some (quite general) cases, which mainly depends on c. For instance, if c is the power of a distance, or if c is stricly concave and μ vanishes on supp ν, then the Monge and Kantorovich formulations are equivalent. A general result proven by Pratelli, 2007 is presented in the following proposition.

Proposition 22. *If X and Y are Polish spaces, μ and ν are nonatomic measures (their mass can be split indefinitely) and c is continuous, then the Monge and Kantorovich formulation of the optimal transport problem have the same solution.*

When this occurs, the Monge formulation is very useful to characterize solutions. For example, for the quadratic cost, it can be shown that the displacement maps are gradients of a convex function.

We will only consider the Kantorovitch formulation in the rest of this thesis.

3.1.3 Particular costs

The solution of the problem will highly depend on the properties of the cost (convexity, regularity, ...) and the characteristic of the input distributions (discrete, continuous, Gaussian, ...). For some special cases, the optimal transport problem takes particular forms that can more easily be solved.

The case $X = Y$ and $c(x,y) = \mathbf{1}_{x\neq y}$

This cost basically means that the cost of displacing mass from x to y is independent of the distance from x to y. For a given transport plan γ, we have

$$\int_{(x,y)\in X\times Y} c(x,y)\gamma(x,y) = 1 - \int_{x\in X} \gamma(x,x)$$

In this setup, the problem boils down to leave as much mass as possible immobile. The only mass that we can afford to leave in place is the one already at the target location. The total cost of the transport will be half of the total mass of the difference of μ and ν, which is exactly the total variation. This is a consequence of the Strassen's theorem (Valadier, 1974). We have then

$$\mathcal{L}_c(\mu,\nu) = \frac{1}{2}||\mu - \nu||_{TV}$$

Wasserstein metric : case where c is the power of a distance

When c is the power of a distance d (and therefore when $X = Y$), the transport cost enables to define a distance on the space of the bounded measures, called the Wasserstein distance. It is one of the reasons which made optimal transport very popular, as it becomes an efficient tool to define the distances between distributions. These distances are widely used in machine learning.

Theorem 23 (Wasserstein distances). *We denote by $\mathcal{P}_p(X)$ the set of probability measures on X with finite moments of order p, i.e. the set of measures μ such that for some x_0 we have:*

$$\int_{x\in X} d(x,x_0)^p d\mu(x) < +\infty.$$

Then we have

- *For all $p \geq 1$, $W_p = \mathcal{L}_{d^p}^{1/p}$ defines a metric on $\mathcal{P}_p(X)$.*
- *For all $p \in [0,1[$, $W_p = \mathcal{L}_{d^p}$ defines a metric on $\mathcal{P}_p(X)$.*

The transport cost is in general symmetric. The point-separation property relies on the one of d, so that the cost is null only on its diagonal. The triangle inequality is a consequence of the Minkowski inequality and the gluing lemma. More details can be found in Villani, 2008.

One dimensional transport

When the space X is one-dimensional and $c = \psi(d)$, with d a distance and ψ a convex function, the Wasserstein distance has a closed form expression.

Definition 24. For a measure α on $\mathbb{R}$, we define the cumulative distribution function $C_\alpha : \mathbb{R} \to [0,1]$ by

$$\forall x \in \mathbb{R}, C_\alpha(x) = \int_{-\infty}^{x} d\alpha,$$

and its pseudo inverse $C_\alpha{-}1 : [0,1] \to \mathbb{R}$ by

$$\forall t \in [0,1], C_\alpha^{-1}(t) = \min\{x \in \mathbb{R} | C_\alpha(x) \geq t\}.$$

Proposition 25. *Then, for $p \geq 1$, we have*

$$W_p(\alpha,\beta)^p = \int_0^1 |C_\alpha^{-1}(t) - C_\beta^{-1}(t)|^p dt.$$

The particularity of the one dimensional case is that the space can be ordered. With this in mind, the introduction of the cumulative function becomes quite natural.

Quadratic cost

Among all costs, there is one particularly convenient to work with, as it behave nicely with the problem. The transport cost associated to the quadratic cost will be denoted by $\mathcal{L}_2$. Here is for information a part of Theorem 11 of Villani, 2003.

Theorem. *Let μ and ν be two probability measures on $\mathbb{R}^n$ with finite second moment. Then,*

1. *$\gamma \in \Pi(\mu,\nu)$ is optimal if and only if there exists a convex lower semi-continuous function ϕ such that*
$$\operatorname{supp}\gamma \subset \operatorname{graph}(\partial\phi),$$
where $\partial\phi$ is the subdifferential of ϕ.

2. *If μ does not give mass to sets of Hausdorff dimension less than $n-1$, then there is a unique optimal γ, which is*
$$d\gamma(x,y) = d\mu(x)\delta_{y=\nabla\phi(x)},$$
where $\nabla\phi$ is the unique gradient of a convex function which pushes forward μ onto ν, i.e. $\nabla\phi\#\mu = \nu$.

3. *As a corollary, $\nabla\phi$ is the unique solution to the Monge transportation problem:*
$$\int |x-\nabla\phi(x)|^2 d\mu(x) = \inf_{T\#\mu=\nu}\int_X |x-T(x)|^2 d\mu(x).$$

$\nabla\phi$ is called the Brenier's map.

Gaussian with quadratic cost

In case of a quadratic cost, we have a closed-form solution for Gaussian distributions, which behave in general nicely in l_2-norm optimizations. It is also a good example of the advantages of the probabilistic formulation.

Proposition 26. *Let $\mu_X = \mathcal{N}(m_X,\Sigma_X)$ and $\mu_Y = \mathcal{N}(m_Y,\Sigma_Y)$ be two Gaussian distribution. Then, the optimal transport quadratic cost between these two distribution is*
$$\mathcal{L}_2(\mu_X,\mu_Y) = |m_X-m_Y|^2 + Tr(\Sigma_X+\Sigma_Y-2\sqrt{\Sigma_X^{\frac{1}{2}}\Sigma_Y\Sigma_X^{\frac{1}{2}}}),$$
And in the case where Σ_X and Σ_Y commute, we have
$$\mathcal{L}_2(\mu_X,\mu_Y) = |m_X-m_Y|^2 + ||\sqrt{\Sigma_X}-\sqrt{\Sigma_Y}||^2_{Frob}.$$

Proof. The proof uses the probabilistic interpretation of the Kantorovich formulation :
$$\mathcal{L}_2(\mu_X,\mu_Y) = \min_{(X,Y),X\sim\mu_X,Y\sim\mu_Y} \mathbb{E}\left[|X-Y|^2\right].$$

First we reduce to the case where $m_X = m_Y = 0$. As we shall see, the reasoning is very general, and we do not need to assume that we work with Gaussian distributions. We just have to remark that

$$\begin{aligned}\mathbb{E}\left[|(X-\mathbb{E}[X])-(Y-\mathbb{E}[Y])|^2\right] &= \mathbb{E}[|X-Y|^2] + \mathbb{E}[|\mathbb{E}[X]-\mathbb{E}[Y]|^2] - 2\langle\mathbb{E}[X-Y],\mathbb{E}[X]-\mathbb{E}[Y]\rangle \\ &= \mathbb{E}[|X-Y|^2] - |\mathbb{E}[X]-\mathbb{E}[Y]|^2,\end{aligned}$$

So we have
$$\mathbb{E}[|X-Y|^2] = |\mathbb{E}[X]-\mathbb{E}[Y]|^2 + \mathbb{E}[|(X-\mathbb{E}[X])-(Y-\mathbb{E}[Y])|^2],$$
which reduces the calculation to the case where the variables have 0 mean.

Then we prove that the optimal case is attained for a Gaussian. We know from the Kantorovich theorem that the infimum is attained (see Theorem 33). Let suppose

that it is attained for a coupling (a probability distribution on $X \times Y$) γ. Then let γ' be a Gaussian distribution with the same covariance as γ. Then clearly we have

$$\mathbb{E}_{\gamma'}\left[|X-Y|^2\right] = \mathbb{E}_{\gamma}\left[|X-Y|^2\right].$$

All this together reduces the problem to look for a Gaussian random variable $Z = (X, Y)$, with marginal μ_X and μ_Y and with covariance

$$Z \sim \mathcal{N}(m_Z, \Sigma_Z) = \mathcal{N}\left(\begin{pmatrix} m_X \\ m_Y \end{pmatrix}, \begin{pmatrix} \Sigma_X & C \\ C^T & \Sigma_Y \end{pmatrix}\right),$$

where C is such that $\Sigma_Z \geq 0$, which is equivalent to $\Sigma_X - C^T\Sigma_Y^{-1}C \geq 0$ by the Shur complement. This transitions to

$$\mathbb{E}\left[||X-Y||^2\right] = \mathbb{E}[X^TX] + \mathbb{E}[Y^TY] - 2\mathbb{E}[X^TY].$$

We have $\mathbb{E}[X^TX] = \mathbb{E}[Tr(XX^T)] = Tr(\Sigma_X) + m_X^Tm_X$ and $\mathbb{E}[Y^TY] = Tr(\Sigma_Y) + m_Y^Tm_Y$ and $\mathbb{E}[X^TY] = Tr(C) + m_X^Tm_Y$. So the minimization is equivalent to

$$\max_{C, \Sigma_Y - C^T\Sigma_X^{-1}C \geq 0} Tr(C)$$

Let $S = \Sigma_Y - C^T\Sigma_X^{-1}C$. Then $\Sigma_Y - S = C^T\Sigma_X^{-1}C \geq 0$. We can write $\Sigma_Y - S = UD^2U = U_rD_r^2U_r^T$ with U an unitary matrix, r the rank of $\Sigma_Y - S$ and U_r of size $n \times r$, $U = [U_r, U_{n-r}]$. U_{n-r} corresponds to the eigenvectors of $\Sigma_Y - S$ for the eigenvalue 0, so $CU_{n-r} = 0$ because $\Sigma_X^{-1} > 0$.

Then we have the identity

$$\left(\Sigma_X^{-\frac{1}{2}}CU_rD_r^{-1}\right)^T\left(\Sigma_X^{-\frac{1}{2}}CU_rD_r^{-1}\right) = I_r.$$

So we have $CU_r = \Sigma_X^{\frac{1}{2}}OD_r$ for O a given orthonormal matrix. Furthermore, we have $CU_{n-r} = 0$. So

$$C = CUU^T = CU_rU_r^T = \Sigma_X^{\frac{1}{2}}OD_rU_r^T.$$

We just showed that for each C such that $S = \Sigma_Y - C^T\Sigma_X^{-1}C \geq 0$, we have $S = \Sigma_Y - C'^T\Sigma_X^{-1}C' \geq 0$ for all $C' = \Sigma_X^{\frac{1}{2}}OD_rU_r^T$ with O an $n \times r$ matrix.

Now, let us fix S. We want to calculate

$$\sup_{O^TO=I_r} Tr(\Sigma_X^{\frac{1}{2}}OD_rU_r^T) = \sup_{O^TO=I_r} Tr(O^T\Sigma_X^{\frac{1}{2}}U_rD_r) = \sup_{O^TO=I_r} Tr(O^TB) \tag{3.8}$$

with $B = \Sigma_X^{\frac{1}{2}}U_rD_r$.

We can express the constraint with a Lagrangian, denoting $O = [v_1, ..., v_r]$, we have $v_i^Tv_j = \delta_{i,j}$, expressed by $\sum_{i,j}\lambda_{i,j}(v_i^Tv_j - \delta_{i,j}) = Tr(\Lambda(O^TO - I))$. The dual problem is

$$\inf_{\Lambda \in S_r} \sup_{O \in M_r} Tr(O^TB) + Tr(\Lambda(O^TO - I))$$

The optimum exists because of the Kantorovich theorem (see Theorem 33), and the admissibility condition is

$$B^T + 2\Lambda O^T = 0 \text{ and } O^T O = I_r,$$

so we have $O\Lambda = -2B$. Furthermore, O is of rank r and B as well, so Λ is invertible, and $O = -2B\Lambda^{-1}$. So we have

$$B^T B = 4\Lambda^T O^T O\Lambda = 4\Lambda^2$$

We conclude that Λ is a square root of $\frac{1}{4}B^T B$, so $Tr(O^T B) = -2Tr(\Lambda^{-1}B^T B) \leq Tr(\sqrt{B^T B}^+)$, where $\sqrt{\cdot}^+$ is the square root with only positive eigenvalues. Equation 3.8 is therefore maximized for $\Lambda = -\sqrt{B^T B}^+$.

What we have shown is that

$$\sup_{O^T O = I_r} Tr(O^T B) = Tr(\sqrt{B^T B}^+) = Tr\left(\sqrt{BB^T}^+\right),$$

since $B^T B$ and BB^T share the same eigenvalues (except 0).

Replacing B by its value, we find

$$Tr\left(\sqrt{BB^T}\right) = Tr\left(\sqrt{\Sigma_X^{\frac{1}{2}} U_r D_r^2 U_r^T \Sigma_X^{\frac{1}{2}}}\right) = Tr\left(\sqrt{\Sigma_X^{\frac{1}{2}}(\Sigma_Y - S)\Sigma_X^{\frac{1}{2}}}\right).$$

So for a given $S \geq 0$, we have

$$\sup_{C;\Sigma_Y - C^T\Sigma_X C = S} Tr(C) = Tr\left(\sqrt{\Sigma_X^{\frac{1}{2}}(\Sigma_Y - S)\Sigma_X^{\frac{1}{2}}}\right).$$

Then we have, $\forall x \in \mathbb{R}^n, x^T\Sigma_X^{\frac{1}{2}}(\Sigma_Y - S)\Sigma_X^{\frac{1}{2}}x \leq x^T\Sigma_X^{\frac{1}{2}}(\Sigma_Y - 0)\Sigma_X^{\frac{1}{2}}x$ because S is symmetric positive. Therefore, the eigenvalues of $\Sigma_X^{\frac{1}{2}}(\Sigma_Y - S)\Sigma_X^{\frac{1}{2}}$ are lower than the eigenvalues of $\Sigma_X^{\frac{1}{2}}(\Sigma_Y)\Sigma_X^{\frac{1}{2}}$, thanks to the Courant-Fisher theorem stating that

$$\mu_{k+1} = \min_{V_k} \max_{x \in V_k^\perp} \frac{x^T M x}{x^T x}.$$

So the maximum is attained for $S = 0$, which gives

$$\sup_{C;\Sigma_Y - C^T\Sigma_X^{-1}C \geq 0} Tr(C) = 2Tr\left(\sqrt{\Sigma_X^{\frac{1}{2}}\Sigma_Y\Sigma_X^{\frac{1}{2}}}\right),$$

and we can conclude that

$$\inf_{X\sim\mu_X, Y\sim\mu_Y} \mathbb{E}\left[|X - Y|^2\right] = Tr(\Sigma_X) + Tr(\Sigma_Y) - 2Tr\left(\sqrt{\Sigma_X^{\frac{1}{2}}\Sigma_Y\Sigma_X^{\frac{1}{2}}}\right).$$

□

Proposition 27. *Let $\mu_X = \mathcal{N}(m_X, \Sigma_X)$ and $\mu_Y = \mathcal{N}(m_Y, \Sigma_Y)$ be two Gaussian distributions. Then, the Brenier's map $\nabla\phi$ for the quadratic cost that pushes μ_X to μ_Y is linear and we have*

$$\nabla\phi(x) = \Sigma_Y^{1/2}\left(\Sigma_Y^{1/2}\Sigma_X\Sigma_Y^{1/2}\right)^{-1/2}\Sigma_Y^{1/2}.$$

A proof of this result can be found in Knott and Smith, 1984.

3.2 Discrete optimal transport

3.2.1 Discrete formulation

The optimal transport problem remains relevant when the distribution are discrete, i.e. can be expressed as a (finite) sum of Dirac masses. We have in this case $\mu = \sum_{i=1}^{n} a_i \delta_{x_i}$ and $\nu = \sum_{j=1}^{m} b_j \delta_{y_j}$ with $a \in \mathbb{R}_+^n$ and $b \in \mathbb{R}_+^m$.

In this framework, admissible transport plans are also discrete. Indeed, for $\gamma \in \Pi(\mu, \nu)$, from condition (3.1), we have

$$\gamma(\{a_1, \dots, a_n\} \times Y) = \mu(\{a_1, \dots, a_n\}) = 1 \text{ and similarly } \gamma(X \times \{b_1, \dots, b_m\}) = 1.$$

So $\gamma(\{a_1, \dots, a_n\} \times \{b_1, \dots, b_m\}) = \gamma((\{a_1, \dots, a_n\} \times Y) \cap (X \times \{b_1, \dots, b_m\}) = 1$. As a consequence, we can write

$$\gamma = \sum_{i,j} P_{i,j} \delta_{(x_i, y_j)}$$

and γ is fully characterized by the matrix P. Condition (3.1) becomes

$$\begin{cases} \sum_{i=1}^{n} P_{i,j} = b_j \\ \sum_{j=1}^{m} P_{i,j} = a_i \end{cases},$$

which can succinctly be rewritten as

$$P \mathbb{1}_m = a \text{ and } \mathbb{1}_n^T P = b. \tag{3.9}$$

where $\mathbb{1}_n$ is the vector of 1 of dimension n.

Then, the transportation cost for a given transport plan can be written

$$\int_{X \times Y} c d\gamma = \sum_{i,j} P_{i,j} c(x_i, y_j). \tag{3.10}$$

We see that the spaces X and Y, and the cost function c disappear. The input can be reduced to two vectors representing the distributions $a = (a_i)_{i \in [\![1,n]\!]}$ and $b = (b_j)_{j \in [\![1,m]\!]}$, and a matrix C such that

$$\forall (i,j) [\![1,n]\!] \times [\![1,m]\!], C_{i,j} = c(x_i, y_j).$$

This matrix is called the *cost matrix*. The transport plans can also be represented by a matrix, and the sum in Equation (3.10) becomes a matrix scalar product. This leads to the discrete formulation:

Definition 28 (Discrete optimal transport)**.** Given a cost matrix $C \in \mathbb{R}_+^{n \times m}$, and two vectors $a \in \mathbb{R}^n$ and $b \in \mathbb{R}^m$, find

$$\min_{P \in \Pi(a,b)} \langle C, P \rangle,$$

where $\Pi(a,b) \subset \mathbb{R}_+^{n \times m}$ is the set of admissible (discrete) transport plan satisfying (3.9).

Remark 29. *The cost matrix can also go to $+\infty$ and take negative values, as adding a constant to it does not change the optimal transport plan. This also applies in the continuous setting, the cost function can take negative values as long as it remains bounded below.*

As definition 28 suggests, the discrete optimal transport problem, besides being convex, is linear. Birkhoff's theorem, a fundamental theorem of linear programming states that any linear program with a non-empty and bounded feasible set attains its minimum at an extreme point of the feasible set. In our case, $\Pi(a, b)$ is non empty and bounded, so we can restrain the search for solutions to these extreme points, which have interesting structure.

3.2.2 Graph interpretation

We will recall here some basic definitions of graph theory that will be useful to characterize our graphical representation of the optimal transport problem.

Definition 30. A *bipartite graph* (or bigraph) is a graph whose vertices can be divided into two disjoint and independent sets S and T, such that each edge of G connects a vertex of S to a vertex of T. We will here only consider undirected bipartite graphs. A bipartite graph is said to be *complete* if each vertex of S is connected to each vertex of T.

As we saw, the original motivation behind optimal transport was the problem of moving a pile of soil to another, which led to the continuous formulation. The discrete problem is more intuitive.

A Dirac mass can be seen as a location, e.g. as a shop or a factory, which needs/provides a certain amount of supply, represented by the mass of the given Dirac mass. A sum of Dirac masses is by extension a natural representation of a set of locations with different needs/productions.

In our classical optimal transport setup, we shall see the first distribution $\mu = \sum_{i=1}^{n} a_i \delta_{x_i}$ as a representation of n factories at the locations $x_1, \dots, x_n$, and $\nu = \sum_{j=1}^{m} b_j \delta_{y_j}$ as a representation of m shops located at $y_1, \dots, y_m$. The factory i produces a_i copies of a product, and the shop j sells b_j copies of it. We suppose that the total production of all the factories equals the total consummation of all the shops. Transporting 1 product from x_i to y_j costs $c(x_i, y_j)$. In order to launch the business, we have to find a plan, saying how many products the factory i delivers to the shop j. This is exactly a transport plan, and the optimal transport plan corresponds to the cheapest way to supply the shops. This interpretation is illustrated on Figure 3.2.

In this setup, it becomes natural to introduce a bipartite graph to represent the problem. The source distribution is represented by a set S of n vertices and the target distribution a set T of m other vertices. Two vertices i and j are connected by an edge $e \in E$ if and only if $P_{i,j} > 0$. In this case, we associate the flow $P_{i,j}$ to this edge.

This gives a bipartite graph $G(P) = ((S, T), E)$. In this setup, the cost matrix naturally induces a cost on the edges of G. A transport plan can be represented by a flow on this graph, i.e. a value on each edge (i, j) referring how much "flow" goes from vertex $i \in S$ to $j \in T$. A transport plan is admissible when the capacity of the vertices are saturated, when the total flow going out of the source vertices of S equals their capacity and the same for the total flow going in the target vertices T. This is illustrated on Figure 3.3.

The discrete optimal transport is very different of the well-known maximum flow problem. It consists in finding the admissible flow with the least cost. This representation does not enable to use directly a graph algorithm to solve the problem (like for linear solvers in Section 3.4.1), but will turn out to be very useful when

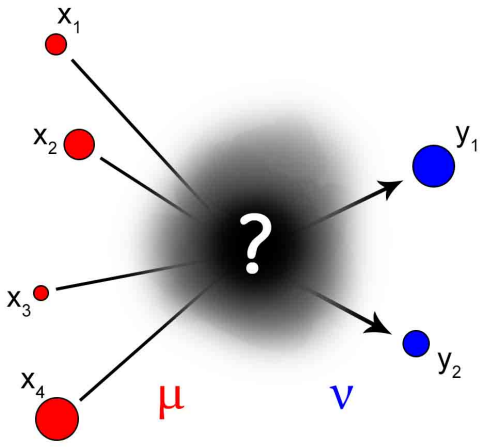


FIGURE 3.2: Illustration of the discrete optimal transport problem. The mass of μ (the products of the factories) must be sent to ν (the shops) with the lowest cost possible.

combined to the dual representation to use the general network simplex algorithm (see Section 3.4.3). For now, it will help us characterize nicely the extreme points of the polytope $\Pi(a,b)$ and thus to have a better idea of the form of the potential solutions.

Proposition 31. *An extreme point of $\Pi(a,b)$ is a flow with no non-null cycle. In particular, it cannot have more than $n+m-1$ zeros.*

Proof. Let X be an extreme point of $\Pi(a,b)$, and E the set of edges of $G(X)$, i.e. edges with (strictly) positive flow. Let suppose that we have a cycle, so we have a path $i_1, j_1, \dots, i_k, j_k, i_1$ with positive edges. Let $\epsilon > 0$ be a positive number lower than all the flows on the path, e.g. $\epsilon = \min_{(i,j)\in E} X_{i,j}$. Then we can define X^+ by augmenting X by ϵ the flow on the odd edges of the cycle, so $\forall l \leq k, X^+_{i_l,j_l} =_{i_l,j_l} +\epsilon$ and $\forall l \leq k, X^+_{j_l,i_{l+1}} = X_{i_l,j_l} - \epsilon$ and X^- by augmenting by ϵ the even edges of the cycle by the same way. So we have clearly $X^+ \in \Pi(a,b)$ and $X^- \in \Pi(a,b)$ and $X = \frac{1}{2}(X^+ + X^-)$, which contradicts the extremality of X. □

Remark 32. *This proof is quite insightful and, as we shall see in Section 3.4.3, its main idea will be central in the network simplex algorithm. However, this result can simply be seen as a consequence of a general linear programming result, stating that if a linear program with M constraints has an optimal solution, then it has an optimal solution with at most M nonnegative entries. As we shall see in Section 3.4.1, the optimal transport problem can be recasted as a linear problem with $n+m-1$ constraints, and has an optimal solution by theorem 33.*

3.2.3 The affectation problem

In case of discrete distributions with uniform weights and the same number of points, we can assume that $\forall i \in [\![1,n]\!], a_i = b_i = 1$.

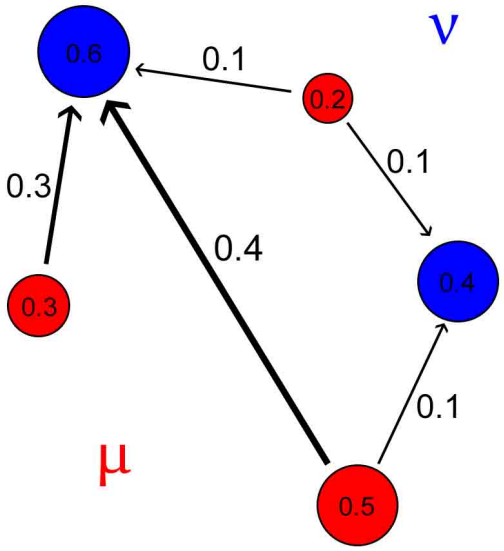


FIGURE 3.3: Example of a transport plan between two discrete distributions, modeled by the black arrows. As we see, this transport is not extreme, as it has 5 nonnegative values and $5 > 2 + 3 - 1$. It may be optimal, but it can be modified to a (non necessarly strictly) better extreme transport plan.

A coupling P would in this case be a positive square matrix whose columns and lines sum to 1. The resulting optimal transport problem becomes

$$\min_{P \in \Pi(a,b)} \sum_{i,j} P_{i,j} C_{i,j}.$$

As we saw in Section 3.2.2, the solution of the problem is necessarily an extremal point of $\Pi(a,b)$. In this case, it can easily be shown that P must be binary, and thus have exactly one non null entry equal to 1 for each line and column. Such a matrix naturally encodes a permutation. The problem becomes therefore

$$L_C(a,b) = \min_{\sigma \in S(n)} \sum_i C_{i,\sigma(i)},$$

which coincides with the *matching problem* or *affectation problem* (also corresponds to the discrete Monge formulation).

This problem, originally formulated on graphs, as the problem of finding an optimal perfect matching on a bipartite graph is well-known in Informatics and can efficiently be solved by the Hungarian algorithm (see Kuhn, 1955).

3.3 Kantorovitch duality

The main result in optimal transport theory is the Kantorovich duality, which under weak conditions on the cost c states the equality with the dual problem. It serves as a basis for most of the further results and study of optimal transport.

3.3.1 General Kantorovich duality

Theorem 33 (Kantorovich duality). *Let X,Y be two (Polish) space with probability measure μ and ν, and $c : X \times Y \to \mathbb{R}_+ \cup \{+\infty\}$ a lower semi-continous function, then:*

$$\min_{\gamma \in \Pi(\mu,\nu)} \int_{X\times Y} c(x,y)d\gamma = \sup_{\Phi_c} \int_X \phi(x)d\mu + \int_Y \psi(y)d\nu,$$

where $\Phi_c = \{(\phi,\psi) \in L^1(d\mu) \times L^1(d\nu); \forall (x,y) \in X \times Y, \phi(x) + \psi(y) \leq c(x,y)\}$.

A detailled proof of this theorem can be found in Villani, 2003. We will present here a formal proof which helps understanding the underlying process of the duality.

Proof. The idea is to rewrite the constraints of the problems as indicator functions and see that they can be derived from each other. The difficulty of the real proof lies in the careful verification of the condition of a general duality theorem.

We have

$$\inf_{\gamma \in \Pi(\mu,\nu)} \int cd\gamma = \inf_{\gamma \geq 0} \left(\int cd\gamma + \begin{cases} 0 \text{ if } \gamma \in \Pi(\mu,\nu) \\ +\infty \text{ otherwise} \end{cases} \right) \tag{3.11}$$

Recalling the compact definition of $\Pi(\mu,\nu)$ of Equation (3.3), we can write the indicator function of (3.11) as

$$\left(\begin{cases} 0 \text{ if } \gamma \in \Pi(\mu,\nu) \\ +\infty \text{ otherwise} \end{cases} \right) = \sup_{\phi,\psi} \left[\int \phi d\mu + \int \psi d\nu - \int (\phi(x) + \psi(y)) d\gamma(x,y) \right]$$

which gives :

$$\begin{aligned} \mathcal{L}_c(\mu,\nu) &= \inf_{\gamma \geq 0} \sup_{\phi,\psi} \left[\int cd\gamma + \int \phi d\mu + \int \psi d\nu - \int [\phi(x) + \psi(y)] d\gamma(x,y) \right] \\ &\equiv \sup_{\phi,\psi} \inf_{\gamma \geq 0} \left[\int cd\gamma + \int \phi d\mu + \int \psi d\nu - \int [\phi(x) + \psi(y)] d\gamma(x,y) \right] \\ &= \sup_{\phi,\psi} \left[\int \phi d\mu + \int \psi d\nu - \sup_{\gamma \geq 0} \int_{X\times Y} [\phi(x) + \psi(y) - c(x,y)] d\gamma(x,y) \right]. \end{aligned}$$

The inversion sup/inf above is formal and needs to be justified properly using a duality theorem.

If it exists (x_0,y_0) such that $\phi(x_0) + \psi(y_0) - c(x_0,y_0) > 0$, then for $\gamma_\lambda = \lambda\delta_{(x_0,y_0)}$, we have

$$[\phi(x) + \psi(y) - c(x,y)] d\gamma_\lambda(x,y) \underset{\lambda \to +\infty}{\to} +\infty.$$

Otherwise, if $\forall (x,y) \in X \times Y, \phi(x) + \psi(y) \leq c(x,y)$, the supremum is obviously 0 and is obtained for $\gamma = 0$. So we have

$$\sup_{\gamma \geq 0} \int_{X\times Y} [\phi(x) + \psi(y) - c(x,y)] d\gamma(x,y) = \begin{cases} 0 \text{ if } (\psi,\psi) \in \Phi_c \\ +\infty \text{ otherwise} \end{cases}.$$

Therefore, this gives

$$\mathcal{L}_c(\mu,\nu) = \sup_{(\phi,\psi)\in\Phi_c} \left[\int \phi d\mu + \int \psi d\nu \right].$$

□

3.3.2 Discrete duality

The duality problem in the discrete setting is a direct application of the general result, but can also be proven using a similar reasoning and a min-max theorem.

Proposition 34. *Let $a \in \mathbb{R}^n$ and $b \in \mathbb{R}^m$ be two vectors and $C \in \mathbb{R}_+^{n\times m}$ be a cost matrix. Then,*

$$L_C(a,b) = \max_{(f,g)\in\Phi_C} \langle f,a\rangle + \langle g,b\rangle, \tag{3.12}$$

where $\Phi_C = \{(f,g) \in \mathbb{R}^n \times \mathbb{R}^m : \forall (i,j) \in [\![1,n]\!] \times [\![1,m]\!], f_i + g_j \leq C_{i,j}\}$ is the set of admissible potentials.

Duality plays a crucial role in computation and characterization of solutions.

3.3.3 Interpretation

We follow here and extend the graphical interpretation given in Section 3.2.2.

We saw that the source distribution can be interpreted as a set of factories that must deliver their products to a set of shops, modeled by the target distribution. A transport plan is a procedure telling how to dispatch to production while minimizing the transportation cost.

Now, we imagine that the operator of the business described above doesn't know how to compute such an optimal plan (he doesn't have access to Section 3.4) but still wants to make sure that he doesn't lose money. He has no choice but to subcontract the transportation. Since he doesn't know the optimal plan, he can't pay for each transportation, the problem would remain the same (the truck company could use a suboptimal plan and charge him more than needed). So he finds an idea to be sure that he does not lose money : he offers the truck company the choice in the pricing, at the condition that it would only charge the loading and unloading of the goods (with the vectors f and g), and that the price f_i of loading a product at factory i and g_j of unloading it at shop j cost less than if he would have done it himself, i.e. that $f_i + g_j \leq C_{i,j}$. The operator is then sure to make a good deal (by the so-called weak duality) or at least to avoid making a bad one. But the duality theorem states that if the truck company finds the right pricing, then the operator will pay exactly as much as we would have with an optimal transport plan.

3.3.4 C-transform

The form of the dual problem suggests further study on the optimal couple of potentials. Let consider (f,g) an admissible pair to the dual problem. Even if the pair (f,g) is not optimal, we can try to check if at least g is optimal for this given f.

In this case, we have to maximize $\langle g,b\rangle$ while keeping the constraints satisfied, i.e. that $\forall i,j, g_j \leq C_{i,j} - f_i$. Since b is positive, it means increasing g until the constraints are saturated, which implies, by definition of the min, by choosing $\forall j \in [\![1,m]\!], g^* : \min_i C_{i,j} - f_i$. This is what we call the C-transform.

Definition 35. Given a dual pair of potential (f, g) and a cost matrix C, the *C-transform* of f, denoted by f^C, is defined by

$$\forall j \in [\![1, m]\!], f_j^C = \min_i C_{i,j} - f_i.$$

Similarly, the *C-transform* of g, also denoted by g^C (by a slight abuse of notation) is defined by :

$$\forall i \in [\![1, n]\!], g_i^C = \min_j C_{i,j} - g_j$$

Remark 36. *We can define a continuous version of the C-transform: for c a cost function and ϕ a function, we define*

$$\phi^c : y \to \inf_y c(x, y) - \phi(x)$$

and similarly for the second variable. Most of the properties developed in the discrete case have their equivalent in the continuous case, but they are difficult to apply directly to the optimal transport problem because of definition and convergence issues that we will not develop here. Still, they are central in the theory and the c-transform will play a big role in the barycenter dual (see Section 3.6.3). For now, it is worth mentioning that the function $\phi \to \phi^c$ is concave.

The c-transform (resp. C-transform) has a lot of nice properties, which makes it a useful tool to study discrete (resp. continuous) optimal transport solutions.

Proposition 37. *If (f, g) is an admissible pair of the discrete dual problem, then (f, f^C) and (g^C, g) are better ones, namely:*

$$\langle f, a\rangle + \langle f^C, b\rangle \geq \langle f, a\rangle + \langle g, b\rangle \text{ and } \langle g^C, a\rangle + \langle g, b\rangle \geq \langle f, a\rangle + \langle g, b\rangle.$$

Proposition 38. *We have the following properties for the C-transform :*

- $f \leq f' \implies f^C \geq f'^C$
- $f^{CC} \geq f$
- $f^{CCC} = f^C$

The same properties apply for the C-transform of g, as the problem is perfectly symmetric.

Proof. The first statement is trivial. For the second one, we write

$$f_k^{CC} = \min_j C_{k,j} - f_j^C = \min_j C_{k,j} - (\min_i C_{i,j} - f_i)$$

for $k \in [\![1, n]\!]$. Then, using $\min_i Ci, j - f_i \leq C_{k,j} - f_k$, we find

$$f_k^{CC} \geq \min_j C_{k,j} - C_{k,j} + f_k = f_k$$

which gives the second statement. Combining it with the first one, we get $f^{CCC} \leq f^C$ and we have also, from the second applied to f^C, $f^{CCC} \geq f^C$, which gives the third one. □

Proposition 37 enables reducing the admissible pair of potentials to the set of potentials of the form (f, f^C) and even of the form (f^{CC}, f^C). This will be very useful in practice to solve the discrete problem. In the continuous setting, this is known as the double convexification trick, to assume that the pair are of the form (ϕ^{cc}, ϕ^c) (the c-transform of a function is often more regular). It is widely used in the proofs, but can rarely be extended to the results because of the passage to the limit.

A similar work can be done for the primal problem and induces the notion of c-cyclical monotony:

Definition 39. Let c be a cost function. A subset $E \subset X \times Y$ is said to be *c-cyclically monotone* if, for any $K \in \mathbb{N}$, and any family $(x_1, y_1), \dots, (x_K, y_K)$ of points in E, the following inequality holds:

$$\sum_{k=1}^{K} c(x_k, y_k) \leq \sum_{k=1}^{K} c(x_k, y_{k+1}) \tag{3.13}$$

with $y_{K+1} = y_1$.

A transport plan is said to be *c-cyclically monotone* if its support is a c-cyclically monotone set.

Intuitively, a c-cyclically monotone set is a set of correspondence (like a continuous graph) that cannot be improved by shifting the correspondence along a cycle. It can be shown that an optimal transport plan is necessarily c-cyclical monotone, and the inverse holds under particular conditions.

This notion, central in the continuous setting, by its relation with the sub-differential of convex functions, has also an equivalent in the discrete setting, as a condition to the edges of the graph $G(P)$. It is rather inefficient because of the impossibility to be checked in practice.

Definition 40. Let C be a cost matrix. A bipartite graph G is said to be *C-cyclically monotone* if, for any $K \in [\![1, \min(n, m)]\!]$, and any family $(i_1, j_1), \dots, (i_K, j_K)$ of vertices of G, the following inequality holds:

$$\sum_{k=1}^{K} C_{i_k, j_k} \leq \sum_{k=1}^{K} C_{i_k, k_{k+1}} \tag{3.14}$$

with $y_{K+1} = y_1$.

A discrete transport plan P is said to be *C-cyclically monotone* if its graph $G(P)$ is a C-cyclically monotone graph.

It roughly says that we cannot construct a cycle with negative cost using edges of G, which already gives the intuition of the crucial role of such cycle in the resolution of the discrete optimal transport problem.

3.4 Solving optimal transport

3.4.1 Linear programming

We have

$$L_C(a, b) = \min_{P \in \Pi(a,b)} \sum_{i,j} P_{i,j} C_{i,j}.$$

with

$$P \in \Pi(a,b) \Leftrightarrow \begin{cases} \forall (i,j) \in [\![1,n]\!] \times [\![1,m]\!], P_{i,j} \geq 0 \\ \forall j \in [\![1,m]\!], \sum_i P_{i,j} = b_j \\ \forall i \in [\![1,n]\!], \sum_j P_{i,j} = a_i \end{cases} . \tag{3.15}$$

The objective function to optimize is linear, as well as the constraints on P. We can therefore rewrite the above equation in a standard form for the linear program literature, and use blindly their algorithms to solve it. We need for this to express the constraints in terms of equality and nonnegativity using vectors, and therefore to encode P in a vector fashion. For instance, we can transform $P \in \mathbb{R}^{n \times m}$ to $p \in \mathbb{R}^{nm}$ by saying that $p_{i+n(j-1)} = P_{i,j}$. This simply consists in putting all the column of P one after another in a vector.

In (3.15), there are $n+m$ linear equality constraints, on $n \times m$ variables, which can therefore be expressed with a matrix $A \in \mathbb{R}^{(n+m)\times(m\times n)}$. With the encodage of P, we need to define A by

$$A = \begin{pmatrix} \mathbb{1}_n^T \otimes \mathrm{I}_m \\ \mathrm{I}_n \otimes \mathbb{1}_m^T \end{pmatrix},$$

where $\otimes$ is the Kronecker's product (see Notations for the definition).

With these definitions, the constraints of (3.15) can be written as

$$P \in \Pi(a,b) \Leftrightarrow p \in \mathbb{R}_+^{nm}, Ap = \begin{pmatrix} a \\ b \end{pmatrix}$$

which gives

$$L_C(a,b) = \min_{\substack{p \in \mathbb{R}_+^{nm} \\ Ap = \begin{pmatrix} a \\ b \end{pmatrix}}} \langle c, p \rangle \tag{3.16}$$

This enables to solve exactly the discrete optimal transport problem, with any linear programming solver, like the simplex algorithm. Even if this gives the right solution and is simple to code, the use of a general solution to linear program is quite inefficient, as we have more information on the problem (like the duality relation) that we could take advantage of. This is the case of the network simplex algorithm, presented in the Section 3.4.3.

Remark 41. *The $n+m$ constraints defined above are redundant, because the fact has been omitted that $\sum_i a_i = \sum_j b_j$. The problem has in reality $n+m-1$ constraints as stated in Remark 32. For practical purposes, one needs to remove one line (any of them) of the constraints to avoid degeneracy in the computation.*

3.4.2 Characterization of the solutions

As we saw in Section 3.3.4, we have some independent characterizations on the optimal solutions of the primal and dual problem. In addition, we can also characterize their relationship.

Let us recall the interpretation with the shops and factories of Section 3.3.3. For the truck company, moving a product from i to j costs $C_{i,j}$ and is charged $f_i + g_j$ to the operator. So if the constraints are saturated, i.e. if we have $f_i + g_j = C_{i,j}$, then the truck company can safely supply the shop j with the factory i. On the contrary, if we have $f_i + g_j < C_{i,j}$, the company would suffer some loss that it will never be able to compensate with another trip.

In conclusion, if the truck company wants to make the deal work, it must transport products only on the edges of the graph where the constraints is saturated, i.e. either it must have $f_i + g_j = C_{i,j}$, or $P_{i,j} = 0$. This notion is called the complementary slackness. It corresponds to a situation (not necessarily optimal) where the operator and the company have a fair deal, i.e. when what the operator pays to the company with pricing (f, g) corresponds to the cost of the company following the procedure P.

Proposition 42. *Let P^* and (f^*, g^*) be optimal solutions for the primal and the dual problem. Then we have*

$$\forall (i,j) \in [\![1,n]\!] \times [\![1,m]\!], P^*_{i,j}(c_{i,j} - f^*_i - g^*_j) = 0.$$

This condition is called the complementary slackness, and if it is satisfied, we say that P and (f, g) are complementary.

Proof. From the strong duality relation we have

$$\langle P^*, C \rangle = \langle f^*, a \rangle + \langle g^*, b \rangle.$$

So

$$\begin{aligned} \langle P^*, C \rangle &= \langle f^*, P^* \mathbb{1} \rangle + \langle g^*, \mathbb{1}^T P^* \rangle \\ &= \langle f^* \mathbb{1}^T, P^* \rangle + \langle \mathbb{1} g^*, P^* \rangle \end{aligned}$$

This leads to

$$\langle P^*, C - f^* \oplus g^* \rangle = 0.$$

where $(f \oplus g)_{i,j} = f_i + g_j$. And, by positivity of P^* and $C - f^* \oplus g^*$, this gives the result. □

Proposition 43. *If P and (f, g) are complementary and feasible solutions of the primal and the dual problem, then both are optimal.*

Proof. From the proof of proposition 42, we know that P and (f, g) are complementary and feasible solutions implies

$$\langle C, P \rangle = \langle f, a \rangle + \langle g, b \rangle.$$

Therefore, we have

$$L_C(a,b) \leq \langle C, P \rangle = \langle a, f \rangle + \langle g, b \rangle \leq L_C(a,b)$$

which gives

$$L_C(a,b) = \langle C, P \rangle = \langle a, f \rangle + \langle g, b \rangle = L_C(a,b).$$

□

3.4.3 The network simplex

General solvers of linear problem use the convexity of the problem, but do not take advantage of the strong duality relation and in particular of the complementary slackness. The network simplex is a general iterative algorithm, alternating on the

primal and dual formulation of the problem to reach the solution. We will present here the (almost complete) adaptation to the optimal transport problem.

Starting from an extreme point of the simplex of the primal admissible solutions, the algorithm is based on two principles :

- For the current admissible primal solution, we associate a complementary dual solution. If this solution is admissible, then the primal solution is optimal (and therefore the dual as well).
- If the dual solution is not admissible, we use it to improve the primal, and get closer to the solution.

Obtaining a complementary dual solution from a primal solution

We will use here the graph formulation of Section 3.2.2 to explain the algorithm. We associate, for a primal solution candidate P, a bipartite graph $G(P) = ((S,T),E)$, S being indexed by i, T by j and E being the set of nonnegative edges of P, i.e.

$$(i,j) \in E \Leftrightarrow P_{i,j} > 0.$$

During the algorithm, we work with a graph G, that we initialize as $G(P)$ with P a starting candidate solution (extreme point of the polytope), but, as we shall see, G can deviate from $G(P)$ by having some additional edges. We will, however, always ensure that $G(P) \subset G$ and that G (and $G(P)$) has no cycle.

Since P is assumed to be an extreme point of the polytope, we have $|E| < n+m$ and $G = G(P)$ has no cycle.

We are looking for a complementary solution (but not necessarily feasible), so we just have to find f and g such that

$$\forall (i,j) \in E, f_i + g_j = C_{i,j}.$$

This is a system of $|E| < n+m$ equations with $n+m$ unknown. It is always solvable, but undetermined, and any solution of this system is suitable.

Yet, there is a very practical way to compute a solution in practice. Since G has no cycle, it is a forest (union of disconnected trees). Let consider $H \subset G$ a tree of G. Each vertex of H corresponds to a variable, and each edge to a constraint, which involves only two variables. Starting from the root r, we set $f_r \leftarrow 0$, and we follow the edges (i,j) of the tree and set to the unassigned variable the value given by the constraint, either $f_i \leftarrow C_{i,j} - g_j$ or $g_j \leftarrow C_{i,j} - f_i$, until the leaves of the tree.

By doing this to each tree of G, we obtain (f,g) a pair of potentials complementary to P. This procedure is efficient but quite naive. In practice, a little more care must be taken to solve the system to ensure that the algorithm remains robust to degeneracy. See Remark 45 for more details.

Network simplex update

Following the procedure of the previous section, we are now in possession of a primal solution candidate P, which is an extreme point of the polytope, and (f,g), a complementary pair of potentials.

If (f,g) is admissible, it means from Proposition 43 that both P and (f,g) are optimal.

Let suppose that (f, g) is not admissible. It means that we have (i_0, j_0) such that

$$f_{i_0} + g_{j_0} > C_{i_0,j_0}.$$

The edge (i_0, j_0) is called the *pivot*. It may not be unique, and it must be carefully chosen in order to avoid termination issues and optimize the computation time (see Remark 45). We will not discuss the strategies here, we just consider one particular pivot.

Let consider the graph G as defined earlier. We know that $(i_0, j_0) \notin G$ since all edges of G have saturated constraints (and therefore we have $P_{i_0,j_0} = 0$). So we add the edge to G. We now have two cases :

1. (i_0, j_0) connects two trees of G, which thus remains a forest. It means that P remains an extreme point of the polytope. We can then construct a new complementary solution (f, g) using the previous algorithm. This case is however degenerated, since we will get an edge of G whereas P_{i_0,j_0} remains 0, and we no longer have $G(P) = G$.

2. (i_0, j_0) creates a cycle. We have to remove an edge to get back a forest, and to modify P so that we still have $G(P) \subset G$.

In the second case, we will follow the same strategy as in the proof of Proposition 31. Adding and subtracting successively a value θ to all the edges of the cycle won't affect the admissibility of P and, if the value is chosen carefully, will make some of them vanish while improving the solution. From the truck company point of view, at this point, the edge (i_0, j_0) is very profitable ($f_{i_0} + g_{j_0} > C_{i_0,j_0}$) and yet unexploited ($P_{i_0,j_0} = 0$): we want to add on it as much traffic as possible (change P) without affecting our quality of the delivery.

Let consider the cycle we have just created: $i_0, j_0, i_1, j_1, \ldots, i_l, j_l, i_{l+1} = i_0$.

We shall add θ to the odd edges (of the form (i_k, j_k) and including the null edge (i_0, j_0)) and subtract θ to the even ones (of the form (i_{k+1}, j_k)). So, in order to keep the positivity of P and to make at least one value vanish, we have to chose

$$\theta = \min_k P_{i_{k+1},j_k},$$

which is the biggest (and only) possible value. P is then modified only on the cycle as follow:

$$\forall k \leq l, P_{i_k,j_k} \leftarrow P_{i_k,j_k} + \theta \text{ and } \forall k \leq l, P_{i_{k+1},j_k} \leftarrow P_{i_{k+1},j_k} - \theta.$$

We finally remove from G **one** edge (i, j) where $P_{i,j}$ vanished (see 45). We obtain a primal admissible solution, which is an extreme point of the polytope, since its positive graph, G, has no cycle.

This solution maximizes the earning of the truck company for the pricing given by the dual pair (f, g). Therefore, it is closer to the optimal as shown in the following proposition.

Proposition 44. *The network simplex update improves the dual solution.*

Proof. Keeping the notation of the rest of the section, we consider P and P' the former and new solution. We have

$$\langle C, P' \rangle - \langle C, P \rangle = \langle C, P' - P \rangle = \theta \times \left(\sum_{k=0}^{l} C_{i_k,j_k} - C_{i_{k+1},j_k} \right)$$

And since the constraints of the edges of P are saturated, we have

$$\begin{aligned}\sum_{k=0}^{l} C_{i_k,j_k} - C_{i_{k+1},j_k} &= C_{i_0,j_0} - C_{i_1,j_0} + \left(\sum_{k=1}^{l} C_{i_k,j_k} - C_{i_{k+1},j_k}\right) \\ &= C_{i_0,j_0} - f_{i_1} - g_{j_0} + \left(\sum_{k=1}^{l} f_{i_k} + g_{j_k} - f_{i_{k+1}} - g_{j_k}\right) \\ &= C_{i_0,j_0} - f_{i_0} - g_{j_0}.\end{aligned}$$

which gives $\langle C, P\rangle - \langle C, P'\rangle = \theta \times (C_{i_0,j_0} - (f_{i_0} + g_{j_0})) \geq 0$. □

Initialise the network simplex

The last remaining step of the simplex is the initialization. We only need an heuristic to start with an extreme point of the simplex. A simple way to do that is the West-corner heuristic, which simply consists in managing the factory and the shop in an arbitrary order.

The idea is to saturate the edges one after the other. At each step (i, j), $P_{i,j}$ take the highest possible value, i.e. $\min(a_i, b_j)$: either the factory i gives all its production to the shop j, either the shop j receives all its products from factory i. In the first case, we increase i (factory i has already given everything, we move on to the next one) and in the second case, we increase j (same idea, shop j is full). If both saturations occur simultaneously, we go directly to $(i + 1, j + 1)$.

As we move in a diagonal, we have at most $n + m - 1$ nonnegative entries, which ensures that the produced primal solution is indeed extreme.

Algorithm

The complete network simplex algorithm is presented in Algorithm 8.

Remark 45. *The proof of Proposition 44 shows that the new transport plan P is not worst than the previous one. However, θ can be null when G and $G(P)$ differ. In this case, P doesn't change, but G does. It may be problematic for the terminaison of the algorithm.*

To ensure that the algorithm terminates, one needs to fix only one root, and initialize with one tree (adding null edge at the beginning), and be careful when deleting the edge at each iteration to ensure that we keep a strongly feasible tree. We can then show that the global distance to the root strictly decreases if the cost doesn't increase. This point is quite technical and will not be developped here. See Bertsekas, 1998, chapter 5, proposition 5.2 for more details.

3.4.4 Other methods

Even if the network simplex is a good way to compute the exact solution of the optimal transport problem, it remains pretty slow and it is not very scalable in practice. Other methods have been developed to quickly approximate the transport cost, like the sliced Wasserstein distance, which project the original distributions on 1-D lines, and use Radon inverses to estimate it (see Rabin et al., 2011 for more details).

Among all these methods, the most efficient and popular is by far the entropic regularization, to which the next section is dedicated.

Algorithm 8 Network simplex for optimal transport

Input: A cost matrix C, input vectors a and b.
Output: P and (f, g) respectively a primal and dual solution.

```
Initialize an extreme primal solution P, using for instance the West-corner heuristic (see Section 3.4.3)
Construct the graph G(P) associated to P
while P is not optimal do
    Compute a complementary dual potential (f, g) as described in Section 3.4.3
    if (f, g) is admissible then
        Return P and (f, g)
    else
        Update the graph G and P as described in Section 3.4.3
    end if
end while
```

3.5 Entropic regularization

Solving the exact optimal transport problem is difficult. In practice it doesn't scale very well and in theory, only a few cases have a closed form solution. One of the main difficulty is that the problem is quite "sharp" : the solution must be sparse. It can therefore be interesting to "soften" the problem with a regularization. A classical way to do it is to penalize with a prior function (similar ideas than in the Bayesian framework and model selection of Section 1.1.6), and in particular with the entropy. This regularization was introduced and popularized by Cuturi, 2013 in the field of machine learning, but was already widely used in other fields such as economics. It leads indeed to very useful theorems and algorithms, but other regularization can be considered, like the l_2 norm (see Essid and Solomon, 2018) or any strictly convex cost (see Dessein, Papadakis, and Rouas, 2018). However this alternatives does not lead to results as good and simple as with the entropy.

3.5.1 Discrete formulation

The main interest of the entropic regularization lies in its resolution in the discrete case.

As we saw in Section 1.1.2, it is generally used in a probabilistic context, but we can extend its definition to matrices (basically by seeing them as vectors):

$$H(P) = -\sum_{i,j} P_{i,j} \log(P_{i,j})$$

with the convention $0 \times \log(0) = 0$ and $H(P) = -\infty$ if P has a nonpositive entry (this fact turns out to be crucial in the Bregman formulation of Equation (3.20)).

Remark 46. *The entropy is sometimes defined with an additional -1, but then it is no longer positive for transport plans.*

Problem 3 (Entropic regularization)**.** Given a, b a source and target vectors, C a cost matrix, and $\epsilon > 0$ a regularization parameter, we define

$$L_C^\epsilon(a, b) = \min_{P \in \Pi(a,b)} \langle P, C \rangle - \epsilon H(P). \tag{3.17}$$

Since $-H$ is strongly convex, and the scalar product is convex, the objective function of (3.17) is also strongly convex. This means that for all $\epsilon > 0$, Problem 3 has a unique optimal solution.

Remark 47. *The entropic regularization introduces a new hyper-parameter ϵ. Even if the influence of this parameter can be intuitively understood, and its limit behavior is stable, its choice still remains an issue in practice. As we shall see in the interpretation of the impact of the regularization, it represents a trade-off between efficiency and dispersion, and therefore cannot be optimized.*

3.5.2 Entropic behaviour

As we saw, solutions of the discrete optimal transport problem lies in the extreme points of $\Pi(a,b)$, and hence have a priori a pretty low entropy. Problem 3 favors solutions with high entropy. So the solution will no longer lie in the border of the polytope but will be attracted toward its center, as a trade-off between entropy and transport cost. The phenomenon is highlighted by the following proposition.

Proposition 48. *For all $\epsilon > 0$, there exists $\lambda > 0$ such that*

$$\min_{P\in\Pi(a,b)} \langle C,P\rangle - \epsilon H(P) = \min_{P\in\Pi_\lambda(a,b)} \langle C,P\rangle$$

with

$$\Pi_\lambda(a,b) = \{P \in \Pi(a,b); KL(P|a\otimes b) \leq \lambda\}.$$

The proof can be found in Cuturi, 2013. Proposition 48 shows that solving the entropic regularization consists indeed in forcing the solution to move away from the extreme point of $\Pi(a,b)$ and approach its (entropic) center, as the set of possible solutions reduces from the convex set $\Pi(a,b)$ to the strictly convex set $\Pi_\lambda(a,b)$.

Proposition 49. *The unique solution P_ϵ of Equation (3.17) converges to the optimal solution with maximal entropy within the set of all optimal solutions. So we have*

$$P_\epsilon \underset{\epsilon\to 0}{\to} \underset{\substack{P\in\Pi(a,b)\\ \langle C,P\rangle = L_C(a,b)}}{\arg\max} H(P) \tag{3.18}$$

and

$$L_C^\epsilon(a,b) \underset{\epsilon\to 0}{\to} L_C(a,b). \tag{3.19}$$

Besides,

$$P_\epsilon \underset{\epsilon\to+\infty}{\to} a\otimes b = (a_i b_j)_{(i,j)\in[\![1,n]\!]\times[\![1,m]\!]}.$$

Proof. Let ϵ_l be a sequence of strictly positive real such that $\epsilon_l \to 0$. For each l, we denote by P_l the solution of the regularized optimal transport problem. By compacity of $\Pi(a,b)$ (closed and bounded), we can assume that a subsequence of P_l (that we do not relabel) converges : $P_l \to P^* \in \Pi(a,b)$.

Let P be a solution of the regular discrete optimal transport problem. We have $\langle C,P\rangle = L_C(a,b)$. Then, for any l, by optimality of P_l, we have $\langle C,P_l\rangle - \epsilon_l H(P_l) \leq \langle C,P\rangle - \epsilon H(P)$. Similarly, by optimality of P (for $\epsilon = 0$), we obtain

$$0 \leq \langle C,P_l\rangle - \langle C,P\rangle \leq \epsilon_l \left(H(P_l) - H(P)\right).$$

Since H is continuous, taking the limit $l \to +\infty$, we have

$$\langle C, P^* \rangle = \langle C, P \rangle = L_C(a, b) \text{ and } H(P^*) - H(P) \geq 0,$$

which shows that P^* is a solution of the right-hand side of (3.18). By strong concavity of H, this solution is unique, and therefore is the only limit possible for a subsequence of (P_l). So we have $P_l \to P^*$, which proves (3.18) and (3.19).

Similarly, having a sequence $\epsilon_l \to +\infty$, (a subsequence of) $P_l \to P^*$, and $P = a \otimes b$ the solution of $\max_{P \in \Pi(a,b)} H(P)$, we have

$$\frac{\langle C, P_l \rangle - \langle C, P \rangle}{\epsilon_l} \leq H(P_l) - H(P) \leq 0$$

which, by taking the limit $l \to +\infty$, shows that $H(P^*) = H(P)$ and that P_l converges by uniqueness of P. □

Remark 50. *These results still hold for any strictly convex regularization functions.*

In the case where $\epsilon \to +\infty$, the entropy regularization forces the solution to split the mass and, at the limit, to spread it completely: each point of the source distribution sends mass to each point of the target distribution. For the shops and factories interpretation, it would correspond to the introduction of a law that would force the factories to share more their product with the rest of the shops. In the case of an infinite ϵ, it means that each factory send a proportion of its production to each shop, which gives the maximal number of edges in $G(P)$. On the contrary, when $\epsilon = 0$, the solution is an extreme point of $\Pi(a, b)$, with at most $n + m - 1$ nonnegative entries, the shops have less than 2 providers in average. For the in-between values of ϵ, the solution will be a balance between these two extremities.

The main impact of the regularization is the diffusion of the mass which "blurs" the optimal transport plans. This can be annoying when we want to calculate the actual transport cost and transport plan, but it can also be preferable in practice, when we use optimal transport with neural networks or to optimize the distance between distribution. But the main advantage of the entropic regularization is that solutions are extremely faster to compute than the exact solution as we shall see in the next section.

3.5.3 Sinkhorn algorithm

Even if the idea of entropic regularization is pretty insightful, there is another way of formulating Problem 3 which is more convenient for practical applications.

Similarly as we did for the entropy, we can define the Kullback-Leibler divergence for (nonnegative) coupling matrices:

$$KL(P|K) = \sum_{i,j} P_{i,j} \log\left(\frac{P_{i,j}}{K_{i,j}}\right).$$

With this definition in mind, we have

$$\begin{aligned} L_C^\epsilon(a,b) &= \min_{P\in\Pi(a,b)} \sum_{i,j} P_{i,j}C_{i,j} + \epsilon \sum_{i,j} P_{i,j}\log(P_{i,j}) \\ &= \min_{P\in\Pi(a,b)} \epsilon \times \left[\sum_{i,j} P_{i,j}\left(\log(P_{i,j}) - \frac{-C_{i,j}}{\epsilon}\right)\right] \\ &= \epsilon \times \min_{P\in\Pi(a,b)} \sum_{i,j} P_{i,j}\log\left(\frac{P_{i,j}}{e^{-\frac{C_{i,j}}{\epsilon}}}\right) \\ &= \epsilon \times KL(P|K), \end{aligned}$$

with K the Gibbs kernel associated to C defined by $K_{i,j} = e^{-\frac{C_{i,j}}{\epsilon}}$.

Finally, we have

$$P^* = \arg\min_{P\in\Pi(a,b)} \langle C,P\rangle - \epsilon H(P) = \arg\min_{P\in\Pi(a,b)} KL(P|K). \tag{3.20}$$

Proposition 51. *Let P_ϵ be the unique solution to the Problem 3. Then there exist $(u,v) \in \mathbb{R}_+^n \times \mathbb{R}_+^m$ such that*

$$\forall (i,j) \in [\![1,n]\!] \times [\![1,m]\!], (P_\epsilon)_{i,j} = u_i K_{i,j} v_j.$$

Proof. We introduce two Lagrangian dual variables $\alpha \in \mathbb{R}^n$ and $\beta \in \mathbb{R}^m$. The optimization problem then becomes:

$$\mathcal{L}(P,\alpha,\beta) = \min_{P\geq 0}\max_{f,g} \langle C,P\rangle - \epsilon H(P) - \langle \alpha, P\mathbb{1}_m - a\rangle - \langle \beta, P^T\mathbb{1}_n - b\rangle.$$

We know that this problem must have a solution $(P_\epsilon,\alpha^*,\beta^*)$, which leads to

$$\frac{\partial \mathcal{L}(P_\epsilon,\alpha^*,\beta^*)}{\partial P_{i,j}} = C_{i,j} - \epsilon\log((P_\epsilon)_{i,j}) - \alpha_i - \beta_j = 0.$$

This gives

$$(P_\epsilon)_{i,j} = e^{\alpha_i/\epsilon} e^{-C_{i,j}/\epsilon} e^{\beta_j/\epsilon}.$$

So we have $u = e^{\alpha^*/\epsilon}$ and $v = e^{\beta^*/\epsilon}$. □

From Proposition 51, we see that computing the matrix P_ϵ reduces to compute the two vectors u and v, which are related to the constraints Π_a and Π_b. It is tempting to use for that an iterative algorithm, which would compute u so that the matrix $(u_i K_{i,j} v_j)_{(i,j)} \in \Pi_a$ for a given v, and compute v so that $(u_i K_{i,j} v_j)_{(i,j)} \in \Pi_b$ for a given u. It turns out that these iterations are fully justified by the general Sinkhorn theorem, which gave its name to this particular procedure.

Following Sinkhorn and Knopp, 1967, the scaling procedure to compute P_ϵ is :

$$u^{(l+1)} = \frac{a}{Kv(l)} \text{ and } v^{(l+1)} = \frac{b}{K^T u^{(l+1)}}. \tag{3.21}$$

From Sinkhorn theorem, we know that the solution is unique up to a multiplicative constant (if (u,v) is a solution, $(\frac{u}{\lambda},\lambda v)$ is also one), but the algorithm will always converges to a pair (u,v) satisfying Proposition 51.

Another way of seeing the Sinkhorn theorem is from a Bregman iterative projection point of view. Indeed, as Equation (3.20) suggests, the entropic regularization of the optimal transport problem can be seen as a projection on $\Pi(a,b)$ with the KL-divergence of the Gibbs kernel of the cost matrix. This form suggests the use of a Bregman iterative projection algorithm, introduced by Bregman, 1967. The principle is to divide the constraints into simpler ones easier to handle and to alternatively project on each corresponding set. Bregman, 1967 showed that under general assumptions, this algorithm converges. It is the case for affine subconstraints. A deep study of general Bregman algorithms and their application to optimal transport can be found in Benamou et al., 2015.

In our case, the set $\Pi(a,b)$ can be seen as the addition (intersection) of two constraints $\Pi_a = \{P; P\mathbb{1}_m = a\}$ and $\Pi_b = \{P; P^T\mathbb{1} = b\}$. These two sets are affine, and therefore the iterative Bregman projections converges with the following procedure

$$P^{(2l+1)} = \underset{P\in\Pi_a}{\arg\min}\, KL(P|P^{(2l)}) \text{ and } P^{(2l+2)} = \underset{P\in\Pi_b}{\arg\min}\, KL(P|P^{(2l+1)}). \tag{3.22}$$

The two problems of Equation 3.22 can be solved using a Lagrangian multiplier, following the same scheme as the proof of Proposition 51, we define

$$\mathcal{L}(P,\alpha) = KL(P|K) - \langle \alpha, P\mathbb{1}_m - a\rangle$$

and we have

$$\frac{\partial \mathcal{L}(P^*,\alpha)}{\partial P_{i,j}} = 0 \Leftrightarrow P^*_{i,j} = \lambda_i \times K_{i,j}$$

with λ_i only depending on i. Using the constraints $P^* \in \Pi_a$, we find that

$$\forall i, \lambda_i = \frac{a_i}{\sum_j K_{i,j}},$$

which gives the result.

The second problem is solved exactly the same way, and this leads to exactly the same updates as seen previously. The two approaches are perfectly equivalent and two different ways of seeing the problem.

In practice, the Sinkhorn algorithm involves the Gibbs kernel of the cost matrix, which can create some computational problems due to the exponentiation. So it is more efficient and convenient to work in the log-domain. The detailed algorithm is presented in Algorithm 9. It makes use of a logsumexp function, that is generally implemented in standard computer science libraries, that efficiently computes the logarithm of the sum of all the exponential of an array.

3.5.4 Continuous formulation

The entropic regularization can also be formulated in a continuous setting:

Problem 4. Let $\mu, \nu \in \mathcal{P}(X)$ absolutely continuous with respect to the Lebesgue measure and c a cost function. Solve

$$\inf_{\gamma\in\Pi(\mu,\nu)} \int_{X\times X} c d\gamma + \epsilon H(\gamma|\mu\otimes\nu). \tag{3.23}$$

As for the other problems, the continuous case is more subtle because of issues of regularity and definition of limits. However, it is still relevant to study, as it is can be related to physics, and in particular to a problem formulated by Schrödinger:

Algorithm 9 Sinkhorn algorithm

Input: A cost matrix C, input vectors a and b, $\epsilon > 0$
Output: P_ϵ solution of the entropic regularization.

Initialize $\log u$ and $\log v$
for A given number of iteration **do**
  $\log X_{i,j} \leftarrow -\frac{C_{i,j}}{\epsilon} + \log v_j$
  $\log u \leftarrow \log a - \text{logsumexp}(X)$
  $\log Y_{i,j} \leftarrow -\frac{C_{j,i}}{\epsilon} + \log u_i$
  $\log v \leftarrow \log b - \text{logsumexp}(Y)$
end for
$P_\epsilon \leftarrow \exp(\log u_i - \frac{C_{i,j}}{\epsilon} + \log v_j)$
Return P_ϵ

knowing the distributions of particles at an initial time t_0 and a final time t_1, identify the most likely flow of density of particles between these two times. The problem seems indeed related to the entropy, which was originally introduced in physics to measure the "amount of chaos in the universe", and turns out to be possibly recasted in an entropic regularization optimal transport problem. It is interesting to note that in this context, the parameter ϵ can be interpreted as the temperature of the particles. See Léonard, 2013 for more details.

3.5.5 Extensions of the entropic regularization

The Sinkhorn algorithm plays a central role in the application of optimal transport. It enables to use optimal transport on problem where the network simplex algorithm doesn't scale. We presented in Section 3.5.3 the basic ideas, but it has been widely studied and improved. Many works have been made to speed up its computation using GPU (see Cuturi, 2013), multiscale approaches (see Schmitzer, 2019), convergence acceleration (see Peyré et al., 2019) and approximations (see Solomon et al., 2015 for instance). The Sinkhorn algorithm has also been generalized to a wider type of algorithms in Peyré, 2015 and Chizat et al., 2018.

Still, as we saw, the entropic regularization is more than a computational friendly application. Its use can be preferable to the regular transport cost in various practical applications, as it models better some chaotic and spreading behaviors. Besides, a similar duality relation holds for the regularized transport cost, which turns out to be differentiable with a simple formula:

Proposition 52 (Dual of the discrete entropic regularization). *For $\epsilon > 0$, we have*

$$L_C^\epsilon = \max_{f \in \mathbb{R}^n, g \in \mathbb{R}^m} \langle f, a \rangle + \langle g, b \rangle - \epsilon \langle e^{f/\epsilon}, K e^{g/\epsilon} \rangle. \tag{3.24}$$

Proposition 53. *For $\epsilon > 0$ and C a cost matrix, $(a, b) \to L_C^\epsilon$ is convex and differentiable. Its gradient reads*

$$\nabla L_C^\epsilon(a, b) = (f^*, g^*)$$

where (f^*, g^*) is the unique solution of Equation 3.24 such that $\sum_i f_i = \sum_j g_j = 0$.

Similarly, for a, b two vectors summing to 1, the function $C \to L_C^\epsilon(a, b)$ is convex and smooth and

$$\nabla_C L_C^\epsilon(a, b) = P^*$$

where P^* is the unique optimal solution of the entropic regularization of the discrete optimal transport problem.

These beautiful results make the regularized Wasserstein distance extremely useful in fields of data science and machine learning.

3.6 Multi-marginal optimal transport

3.6.1 Continuous formulation

Until now, we considered the problem of optimal transport as transporting the mass from a source distribution to a target distribution. This seemed at first unsymmetrical (with the Monge formulation), but we saw with the Kantorovich relaxation that it was more natural to search for an optimal coupling (a transport plan) instead of a transport map. This formulation does not limit the number of marginals to couple. When they are more than two, we speak of the multi-marginal optimal transport problem.

Problem 5 (Multi-marginal optimal transport)**.** Given p spaces $X_1, \dots, X_p$, p distributions $\nu_1 \in \mathcal{P}(X_1), \dots, \nu_p \in \mathcal{P}(X_p)$ called *marginals*, and $c : X_1 \times \cdots \times X_p \to \mathbb{R}_+$ a lower semi-continuous cost function, we consider the following optimization problem:

$$\inf_{\gamma \in \Pi(\nu_1, \dots, \nu_p)} \int_{X_1 \times \cdots \times X_p} c d\gamma \tag{3.25}$$

where $\Pi(\nu_1, \dots, \nu_p)$ is the set of admissible coupling having $\nu_1, \dots, \nu_p$ as marginals, i.e. such that

$$\forall i \in [\![1, p]\!], \pi_i(\gamma) = \nu_i$$

where $\pi_i : X_1 \times \cdots \times X_p \to X_i$ is the canonical projection.

The value of (3.25) is called the transport cost and is denoted by $\mathcal{L}_c(\nu_1, \dots, \nu_p)$.

For $p = 2$, the multi-marginal optimal transport problem exactly corresponds to the regular optimal transport problem. Many of the results on regular optimal transport apply to the multi-marginal case, with sometimes some technical adjustments due to the scaling of the dimension. It appears that it also leads to special results and form of solutions with no counter-part in the two marginal case. The multi-marginality character of the problem increases its difficulty, and it has been less studied. In particular, we can show that Problem 5 admits a solution, and compute its dual problem :

Problem 6 (Dual of multi-marginal optimal transport)**.** Let $\nu_1, \dots, \nu_p$ be p marginals and c a cost function. Find

$$\sup_{u_1, \dots, u_p \in \Phi_c} \sum_{i=1}^{p} \int_{X_i} \phi_i d\nu_i, \tag{3.26}$$

with

$$\Phi_c = \left\{ u_1, \dots, u_p \in L_1(d\nu_1) \times \cdots \times L_1(d\nu_p); \forall (x_1, \dots, x_p) \in X_1 \times \cdots \times X_p, \sum_i \phi(x_i) \leq c(x_1, \dots, x_p) \right\}$$

the set of admissible potentials.

The form of the dual is not surprising and can be guessed from the two marginals problem, which is obviously a particular case of this one. The same concepts can

be derived from this formulation, like the multi-marginal c-transform (also called c-conjugate) and the c-cyclically monotone set.

However, the strong duality relation is not as well-known as for the two marginals case. From Kellerer, 1984, it holds for bounded costs. Some recent work have shown that it holds for the Coulomb cost:

$$c(x_1, \dots, c_K) = \sum_{k \neq k'} \frac{1}{|x_k - x_{k'}|^2}.$$

in some particular cases.

We can define similarly the Monge formulation of the multi marginal optimal transport problem:

Problem 7 (Multi-marginal Monge formulation)**.** Let $\nu_1 \in \mathcal{P}(X_1), \dots, \nu_p \in \mathcal{P}(X_p)$ and $c : X_1 \times \cdots \times X_p \to \mathbb{R}$ a lower semi-continuous cost function. Find

$$\inf_{(T_1,\dots,T_p) \in \mathcal{T}(\nu_1,\dots,\nu_p)} \int_{(x_1,\dots,x_p) \in X_1 \times \cdots \times X_p} c(T_1(x_1), \dots, T_p(x_p)) d\nu_1(x)$$

where $\mathcal{T}(\nu_1, \dots, \nu_p)$ is the set of admissible transport maps, and is defined by

$$\mathcal{T}(\nu_1, \dots, \nu_p) = \{(T_1, \dots, T_p); T_i \# \nu_1 = \nu_i, 2 \leq i \leq p \text{ and } T_1 = \mathrm{I}\}.$$

In this formulation, we consider having one source distributions and $p-1$ target distribution. It turns out that the results of Proposition 22 can easily be scaled by recurrence to the multi-marginal case:

Proposition 54. *If $X_1, \dots, X_K$ are Polish space, $\nu_1, \dots, \nu_K$ are non-atomic probability measures, and c a continuous cost, the multi-marginal Monge formulation and the multi-marginal Kantorovich formulation have the same solution.*

See Nenna, 2016 for more details on the multi-marginal optimal transport problem.

3.6.2 Discrete formulation

The discrete multi-marginal formulation and its dual derive quite naturally from the continuous two marginals ones.

Problem 8 (Discrete multi-marginal optimal transport)**.** Let $a_1, \dots, a_p$ be vectors of $\mathbb{R}^{d_1}, \dots, \mathbb{R}^{d_p}$ and $C \in \mathbb{R}^{d_1 \times \cdots \times d_p}$ be a cost matrix. Find

$$\min_{P \in \Pi(a_1,\dots,a_p)} < C, P >= \min_{P \in \Pi(a_1,\dots,a_p)} \sum_s \sum_{i_s=1}^{d_s} P_{i_1,\dots,i_p} C_{i_1,\dots,i_p},$$

with $\Pi(a_1, \dots, a_p) = \{P \in \mathbb{R}^{d_1 \times \cdots \times d_p}; \forall s, \forall i_s, \sum_{l \neq s} \sum_{i_l=1}^{n_l} P_{i_1,\dots,i_p} = a_{s,i_s}\}$.

Problem 9 (Dual of the discrete multi-marginal OT problem)**.** Let $a_1, \dots, a_p$ be vectors of $\mathbb{R}^{d_1}, \dots, \mathbb{R}^{d_p}$ and $C \in \mathbb{R}^{d_1 \times \cdots \times d_p}$ be a cost matrix. Find

$$\max_{(f^1,\dots,f^K) \in \Phi_C} \sum_{k=1}^{K} \sum_{i_k=1}^{d_k} f^k_{i_k} a_{i_k},$$

where $\Phi_C = \{(f^1, \dots, f^K) \in \mathbb{R}^{d_1} \times \cdots \times \mathbb{R}^{d_K}; \forall (i_1, \dots, i_K), \sum_k f^k_{i_k} \leq C_{i_1,\dots,i_K}\}$.

Similarly to the two marginals problem, the multi-marginal transport problem can be formulated as a general linear problem, and solved by a solver. However, the primal (resp. the dual) problem has $d_1 \times \cdots \times d_K$ variables (resp. constraints) and $d_1 + \ldots + d_K$ constraints (resp. variables). This makes this very inefficient and enables to solve the problem only when having a few marginals with a few points.

3.6.3 Barycenter and optimal transport

Having a metric on a given space enables to define the notion of barycenter on this space. There are several ways to do so, the most popular follows the idea of the Fréchet mean:

Definition 55. Given K points $x_1, \ldots, x_K \in X$, a "pseudo-metric" d on this space and $\lambda_1, \ldots, \lambda_K \in \mathbb{R}_+$ some weights, a barycenter of $x_1, \ldots, x_K$ for d is defined by

$$B_d(x_1, \ldots, x_p) = \arg\min_{y \in \mathbb{R}^d} \sum_i \lambda_i d(x_i, y).$$

B_d is called the barycentric map and is not in general single-valued.

When X is an Euclidian space and d is the square of the l_2 norm, the barycenter of K points is unique and we have

$$\forall x_1, \ldots, x_K \in X, B_2(x_1, \ldots, x_K) = \sum_i \lambda_i x_i. \tag{3.27}$$

In this case, B_2 is single-valued and defines a proper function from X^K to X.

As we saw, optimal transport theory can define a distance on the space of distributions thanks to the the Wasserstein distance. The question of a Wasserstein barycenter arises therefore naturally.

Problem 10 (Wasserstein Barycenter)**.** Given K probability distributions $\nu_1, \ldots, \nu_K \in \mathcal{P}(X)$ and $\lambda_1, \ldots, \lambda_K \in \mathbb{R}_+$, find

$$\inf_{\nu \in \mathcal{P}(X)} \sum_i \lambda_i W_p(\nu_i, \nu) \tag{3.28}$$

where W_p is the p-Wasserstein distance.

Due to the high non-linearity of this problem, it is very difficult and it still remains an open issue to compute and characterize the Wasserstein barycenters. However, some relaxations have been considered like the transport cost barycenter:

Problem 11 (Transport cost barycenter)**.** Given K probability distributions $\nu_1, \ldots, \nu_K \in \mathcal{P}(X)$, $\lambda_1, \ldots, \lambda_K \in \mathbb{R}_+$ and a cost function c, find

$$\inf_{\nu} \sum_i \lambda_i \mathcal{L}_c(\nu_i, \nu). \tag{3.29}$$

The main advantage of this formulation is that the objective function becomes linear. In the discrete case, we have

$$\min_{a \in \Sigma_n} \min_{P_1 \in \mathbb{R}^{n \times n_1}, \ldots, P_K \in \mathbb{R}^{n \times n_K}} \left\{ \sum_k \lambda_k < P_k, C_k >; \forall k, P_k \mathbb{1}_{n_k} = a, P_k^T \mathbb{1}_n = b_k \right\}$$

it enables to solve it with a linear solver, or subgradient descent on the dual to tackle the scalability of the problem, as presented in Carlier, Oberman, and Oudet, 2015,

or can even be approximated using Bregman iterative projections, see Benamou et al., 2015. The main result on the transport cost barycenters is due to Agueh and Carlier, 2011, who have studied in depth the questions of existence and unicity of the barycenters for the square of the 2-Wasserstein distance :

$$\inf_{\nu} \sum_i \lambda_i W_2^2(\nu_i, \nu) = \inf_{\nu} \sum_i \lambda_i \mathcal{L}_2(\nu_i, \nu). \tag{3.30}$$

They have shown existence, unicity and regularity of the solution, and that the solutions of this barycenter problem can be related to the solution of the multi-marginal optimal transport problem with a particular cost function (see also Gangbo and Święch, 1998), which can be written

$$\inf_{\gamma \in \Pi(\nu_1, \nu_2, \dots, \nu_p)} \int_{\mathbb{R}^d \times \cdots \times \mathbb{R}^d} \frac{1}{2} \sum_{i,j=1}^{p} \lambda_i \lambda_j |y_i - y_j|^2 d\gamma(y_1, y_2, \dots, y_p), \tag{3.31}$$

where $\Pi(\nu_1, \nu_2, \dots, \nu_p)$ is the set of probability measures on $(\mathbb{R}^d)^p$ with $\nu_1, \nu_2, \dots, \nu_p$ as marginals. More precisely, they show the following proposition.

Proposition 56 (Agueh and Carlier, 2011)**.** *Assume that ν_i vanishes on small sets for $i = 1, \dots, p$. If (3.31) has a solution γ^*, then $\nu^* = B\#\gamma^*$ is a solution of (3.30), and the infimum of (3.31) and (3.30) are equal.*

It turns out that this equivalence remains relevant for barycenters of more general transport costs, if the barycentric map B_c is single-valued.

Proposition 57. *Let $\nu_1, \dots, \nu_K \in \mathcal{P}(X)$ be K probability distributions, $\lambda_1, \dots, \lambda_K \in \mathbb{R}_+$ and a cost function c such that B_c is single-valued. Then $B_c\#\gamma^*$ is a solution of the transport cost barycenter, where γ^* is an optimal transport plan of the multimarginal optimal transport problem with marginals $\nu_1, \dots, \nu_K$ and the cost*

$$\hat{c} : \begin{cases} X_1 \times \cdots \times X_K & \to \mathbb{R} \\ (x_1, \dots, x_K) & \to \sum_k c(x_k, B_c(x_1, \dots, x_K)) \end{cases}$$

Some other variations of the barycenter problem have been studied like replacing the transport cost by its entropic regularization (see Cuturi and Doucet, 2014 and Cuturi and Peyré, 2016), or by restraining the constraint to Gaussian mixtures (see Delon and Desolneux, 2020). This topic is still a very active area of research, as Wasserstein barycenters have numerous applications in image processing, computer graphics, statistics and machine learning. See Peyré and Cuturi, 2019 for more details.

3.6.4 Interpretation

The two marginals optimal transport problem can be intuitively understood using the shops and factory interpretation. We could try to extend it to the multi-marginal case, saying that each factory produces $K-1$ products and we have $K-1$ shops specialized in each product. This approaches a Monge formulation (due to the asymmetry of the marginals) and doesn't work directly because the cost function concerns all marginals at once, and not taken independently.

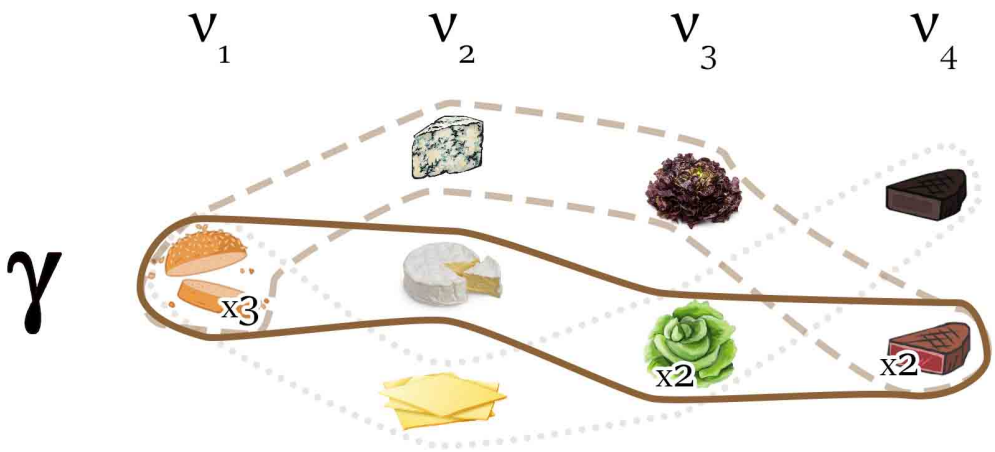


FIGURE 3.4: Illustration of the burger making interpretation of the multi-marginal optimal transport problem. The distribution can be modeled by the different ingredients of the burgers, and a multi-marginal transport plan is the set of made burgers (that can be seen as team of ingredients). Each burger has a taste according to the synergy of its components, and the goal is to optimize the global taste (the sum) of the production.

Multi-marginal interpretation:

The multi-marginal optimal transport problem is better represented in the context of a restaurant: let imagine that we want to make burgers. The distributions $\nu_1, \dots, \nu_p$ corresponds to the different ingredients needed, say steak, cheese, salad, etc.... Each distribution indicates the supply of each ingredients. For instance, if $\nu_1 = \sum_i a_i^{(1)} \delta_{x_i}$, then $x_1, \dots, x_n$ refer to the different categories of cheese, e.g. cheddar, roquefort and camembert, and a_i how much of each is available. To make one hamburger, we need exactly one piece of each ingredient and its global taste will depend on the synergy of the ingredients, given by the cost c. It is indeed a total waste to mix a high quality rare steak with some tasteless cheddar, or to mix a well done steak with some beaufort. The problem is to find the way to make the most tasteful burger production from the ingredients at disposal. This is a way to find an optimal combination among different populations, or a "team assignment" as it is called in the fields of economics (with yet a slightly more restricted problem). This is illustrated in Figure 3.4.

Transport cost barycenter problem interpretation :

As we saw, the optimal transport problem is well-described by the notion of "transportation", and the multimarginal optimal transport problem is more about assignement. The transport cost barycenter problem stands in middle ground. For two point of the map x and y, $c(x, y)$ still corresponds to the cost of transporting something from x to y. Let imagine that we are in possession of old cars that we want to recycle. But recycling a wheel is different of recycling an engine : for each part of the car, there is a specialized recycling center. The distributions $\nu_1, \dots, \nu_p$ correspond to the locations and capacities of these specialized centers, for instance ν_1 refers to the wheel centers, ν_2 the engine centers, and so on. Before recycling the cars with these

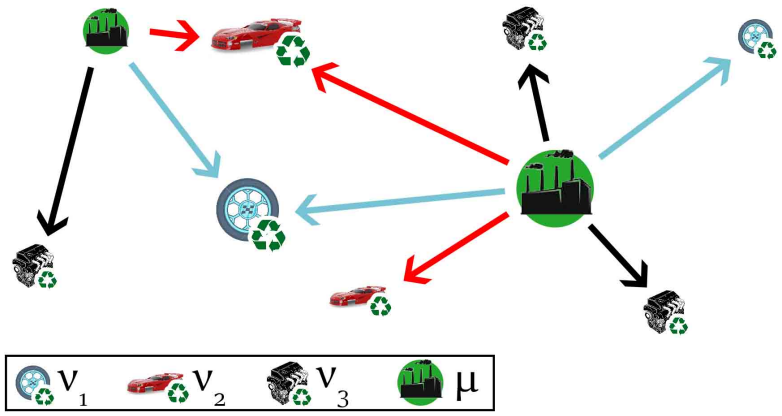


FIGURE 3.5: Illustration of the interpretation of the optimal transport barycenter problem. The input distributions $\nu_1, \dots, \nu_K$ are modeled by different recycling centers, specialized in wheel, engine or car body. In this context, the barycenter μ is a set of dissassembling centers that cut cars into its different components and send it to the recycling centers.

centers, we need first to disassemble them and send the pieces to the corresponding centers. The problem of the barycenter consists in choosing the locations of the disassembling centers. We can create as much as we want, but we want to minimize the cost of transporting the car piece to the recycling centers once they are disassembled. This is illustrated in Figure 3.5.

Link between the transport cost barycenter and the multi-marginal transport :

Finding a solution of the transport cost barycenter problem corresponds to choosing an optimal distribution of the disassembly centers following the previous interpretation. However, in order to actually recycle the cars, each disassembly center has to figure out where it has to send the disassembled pieces. The different parts are independent, so it can be done by solving a regular optimal transport problem between the disassambling centers distribution (the barycenter) and each of the corresponding specialized recycling center distribution.

From another point of view, this corresponds to the knowledge of the final distribution of the destination of the cars pieces, i.e. for each particular car, to which recycling centers go its different components. This is exactly a team assignment problem, a team being the set of recycling centers which will receive the different pieces from a same car. And this information is enough to recover the optimal location of the dissassembly centers: at the spatial barycenter of all the teams of recycling centers. This highlights the fact that the transport cost barycenter problem can be reformulated as a multimarginal optimal transport problem. This is illustrated in Figure 3.6.

3.6.5 Multi-marginal Sinkhorn iterations

The well-known Sinkhorn algorithm used to solve optimal transport regularization can be extended to the multi-marginal case. It leads to similar Bregmann projections (see Benamou et al., 2015 for more details).

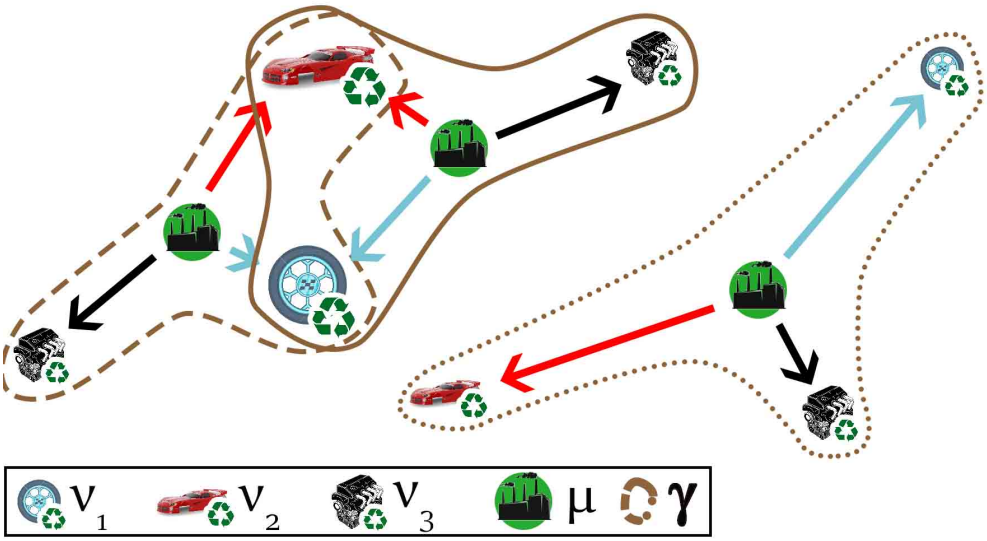


FIGURE 3.6: Illustration of the link betwwen the barycenter and the multi-marginal problem. The solution of the barycenter μ, the dissassembling centers, can be reduced to the teams of recycling centers that will work together (receiving the components of the same car). With a proper cost, these teams can be found by a multi-marginal transport plan γ.

We will use in this section multi-indices. Let $\mathcal{I} = [\![1, n_1]\!] \times \cdots \times [\![1, n_K]\!]$, and $i = (i_1, \ldots, i_K) \in \mathcal{I}$, then we shall write C_i for $C_{i_1,\ldots,i_K}$.

We consider $a_1 \in \mathbb{R}^{n_1}, \ldots, a_K \in \mathbb{R}^{n_K}$ the discrete marginals of our problem, and $C \in \mathbb{R}^{n_1 \times \cdots \times n_K}$ the cost matrix.

The regularized OT problem can be generalized to the multi-marginal case by

$$L_C(a_1, \ldots, a_K) = \inf_{P \in \Pi(a_1,\ldots,a_K)} \sum_{i \in \mathcal{I}} C_i P_i - \epsilon H(P)$$

Similarly to the two marginals case (see Equation 1.3), this problem can be reformulated as an optimization of a Kullback Leibler divergence:

$$L_C(a_1, \ldots, a_K) = \operatorname*{arg\,min}_{P \in \Pi(a_1,\ldots,a_K)} KL(P|K)$$

with $K_i = e^{-C_i/\epsilon}$.

The solution P^* can be approached by Bregman projections and can be written in this form

$$\forall i \in \mathcal{I}, P_i^* = K_i \times \prod_{k=1}^{K} (u_k)_{i_k},$$

where $u_1, \ldots, u_K$ are non-negative vectors uniquely determined up to a multiplicative constant, which can be obtained following an iterative projection procedure, similar to the Sinkhorn algorithm (see Section 3.5.3). Each u_k must be updated as follow:

$$\forall i_k \in [\![1, n_k]\!], (u_k^{(n+1)})_{i_k} = \frac{(a_k)_{i_k}}{S_k(g_k^{(n+1)})}$$

where S_k and g_k are the marginalization along dimension k and the actual k-multiplicative factor, and are defined by

$$\begin{cases} \forall i \in \mathcal{I}, (g_k^{(n+1)})_i = K_i \left(\prod_{l<k} (u_l^{(n+1)})_{i_l} \right) \left(\prod_{l>k} (u_l^{(n)})_{i_l} \right) \\ \\ \forall \boldsymbol{i_k} \in [\![1, n_k]\!], S_k(P)_{\boldsymbol{i_k}} = \sum_{i_1,\dots,i_{k-1},i_{k+1},\dots,i_K} P_{i_1,\dots,i_{k-1},\boldsymbol{i_k},i_{k+1},\dots,i_K} \end{cases}$$

These operations are a direct generalization of the 2-marginal case. They can be coded in Python and enable to approximate the solution of the multimarginal OT problem with more points and dimension that the exact linear approach. It will be used in the next chapter for several experiments.

3.7 Conclusion

Optimal transport is a very wide subject. We presented here the basic tools and notions that we will use in the next Chapter, as well as some insights on related parts of the fields, but we did not mention many other variations and extensions of optimal transport.

Another big aspect of optimal transport is its application to physics model and in particular to the fluid mecanics. Optimal transport problems also have a dynamic formulation, called the Benamou-Brenier formulation, more related to interpolation and partial differential equations. For a good insight on optimal transport, see Villani, 2003 or Villani, 2008 for a theoretical perspective, and Peyré and Cuturi, 2019 from a more computational point of view.

Chapter 4

Generalized Wasserstein Barycenter

In this chapter, we introduce a generalization of the Wasserstein barycenter, to a case where the initial probability measures live on different subspaces of $\mathbb{R}^d$. This study is motivated by the problem of patch model fusion introduced in Chapter 2 and aims at finding a way to merge patch models using optimal transport.

4.1 Introduction

In recent years, optimal transport (Villani, 2008) has received a lot of attention and has become an essential tool to compare or interpolate between probability distributions. The apparition of efficient numerical approaches has made optimal transport particularly successful in numerous applied fields such as image processing, machine learning (particularly deep learning) and computer graphics (Peyré and Cuturi, 2019), to name just a few.

An important tool derived from optimal transport is the notion of Wasserstein barycenter. In the euclidean case, the barycenter of $x_1, \dots, x_p$ with weights $\lambda_1, \dots, \lambda_p$ (positive and summing to 1) is the point x of $\mathbb{R}^d$ which minimizes $\sum_{i=1}^p \lambda_i |x - x_i|_2^2$. The Wasserstein barycenter is obtained in the same way in the space $\mathcal{P}_2(\mathbb{R}^d)$ of probability measures with second order moments, by replacing the euclidean distance by the square Wasserstein distance W_2.

In this chapter, we propose a generalization of the notion of Wasserstein barycenter, to a case where the considered probability measures live on different subspaces of $\mathbb{R}^d$. Relying on the same euclidean analogy as above, for p vectors $x_i \in \mathbb{R}^{d_i}$ and p linear transformations $P_i : \mathbb{R}^d \to \mathbb{R}^{d_i}$, $i = 1, \dots, p$, a generalized barycenter between these x_i can be defined as a minimizer in $\mathbb{R}^d$ of $\sum_{i=1}^p \lambda_i |P_i(x) - x_i|_2^2$. A solution is given by $\hat{x} = (\sum_{i=1}^p \lambda_i P_i^T P_i)^{-1} (\sum \lambda_i P_i^T x_i)$ when the matrix $\sum_{i=1}^p \lambda_i P_i^T P_i$ is full rank. Our generalized Wasserstein barycenter is obtained by replacing the vectors x_i by p probability measures ν_i on their respective subspace $\mathbb{R}^{d_i}$ and the euclidean distance by W_2. In other words, we study the minimization problem

$$\inf_{\nu} \sum_{i=1}^{p} \lambda_i W_2^2(P_i \# \nu, \nu_i), \tag{4.1}$$

where $P_i \# \nu$ denotes the push-forward measure of ν by P_i.

Observe that this formulation contains the Wasserstein barycenter problem as a special case (choosing P_i as the identity matrix on $\mathbb{R}^d$). A particular case of this problem, where all but one of the P_i are 1D projections and the last one is full rank, has been studied in Abraham et al., 2017 for tomography reconstruction.

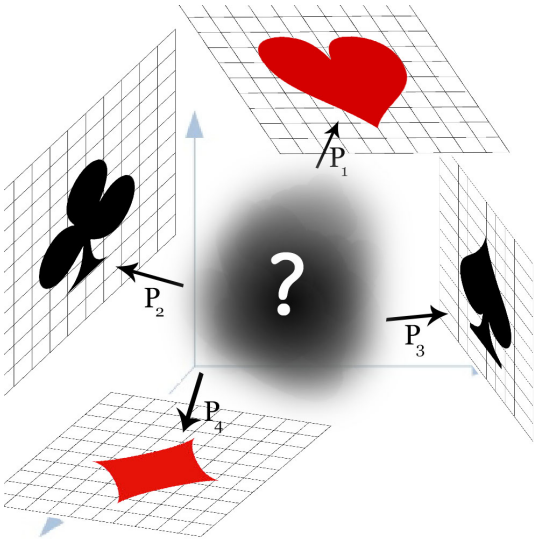


FIGURE 4.1: Illustration of the geometrical application of our generalized barycenter.

Figure 4.1 illustrates an example in $\mathbb{R}^3$ for $p = 4$ distributions, where the transformations P_1 are linear projections on four different plans. Knowing these four projections $\nu_1, \dots, \nu_4$, we look for the 3d probability measure ν which minimizes (4.1).

As a simpler example, for $p = 2$ probability measures ν_1 and ν_2 on $\mathbb{R}^2$, assume that $P_1 : \mathbb{R}^3 \to \mathbb{R}^2$ is the projection on the first two coordinates, and $P_2 : \mathbb{R}^3 \to \mathbb{R}^2$ is the projection on the last two coordinates. If ν_1 and ν_2 coincide on their only common coordinate, the minimum of (4.1) is 0 and a possible solution is given by the gluing lemma applied to ν_1 and ν_2. A more interesting case is obtained when the measures ν_1 and ν_2 do not coincide on their common coordinate. In this case, a solution of the minimization problem should realize a compromise between the marginals of ν_1 and ν_2 on this common coordinate.

A first application of this problem is the reconstruction of a measure from the mere knowledge of projections of this measure on different subspaces. These projections can be noisy or contain errors, and therefore do not necessarily coincide on their common subspaces. Another application, in image processing, is patch-based aggregation Saint-Dizier, Delon, and Bouveyron, 2020. Patches are small overlapping image pieces. Many Bayesian image restoration approaches Zoran and Weiss, 2011; Lebrun, Buades, and Morel, 2013; Wang and Morel, 2013; Yu, Sapiro, and Mallat, 2012; Teodoro, Almeida, and Figueiredo, 2015; Houdard, Bouveyron, and Delon, 2017 work at the patch level and assume stochastic prior models on these patches. Since these models are usually infered independently on all patches, they never coincide on their overlaps. In this context, the generalized Wasserstein barycenter is a way to define a stochastic model on the whole image from a set of probability distributions on patches.

The contributions of this chapter are the following. We first show the existence of solutions for the minimization problem (4.1). We also show how it is related to a multi-marginal optimal transport problem in dimension $\sum_{i=1}^{p} d_i$, and how to solve 4.1 numerically.

The chapter is organized as follows. Section 4.2 studies the dual of (4.1), shows the existence of our generalized Wasserstein barycenters and studies the link with an associated multimarginal problem. Section 4.3 illustrates the notion of generalized Wasserstein barycenters on Gaussian distributions, and finally, Section 4.4 proposes

some numerical experiments.

4.2 Generalized Wasserstein barycenters between probability measures on different subspaces

In the Wasserstein barycenter problem (see Section 3.6.3), the measures $\nu_1, \dots, \nu_p$ are assumed to live in the same space $\mathbb{R}^d$. In the multi-marginal formulation (3.31), these measures are seen as marginals on independent spaces of a probability measure on the product space $(\mathbb{R}^d)^p$. The generalized Wasserstein barycenters (GWB) can be seen as a generalization of these problems, where the space to which the ν_i live can intersect.

Definition 58 (GWB)**.** Given p positive integers $d_1, \dots, d_p$, p probability measures $(\nu_1, \dots, \nu_p) \in \mathcal{P}_2(\mathbb{R}^{d_1}) \times \cdots \times \mathcal{P}_2(\mathbb{R}^{d_p})$, $\lambda = (\lambda_1, \dots, \lambda_p)$ positive weights summing to 1 and p surjective linear applications $P_i : \mathbb{R}^d \to \mathbb{R}^{d_i}$, a solution ν of the minimization problem

$$\inf_{\nu} \sum_{i=1}^{p} \lambda_i W_2^2(\nu_i, P_i \# \nu) \tag{GWB}$$

is called generalized Wasserstein barycenter of the marginals ν_i for the applications P_i.

Observe that the previous energy is convex in ν. In the following section, we study the dual of this optimization problem and we show the existence of solutions for the primal optimization problem.

Remark 59.

- *The case where $X_1 = \cdots = X_p = X$, $P_1 = \cdots = P_p = Id$ and $c_i = \lambda_i c$ corresponds to the optimal barycenter problem introduced in Section 3.6.3.*
- *The case where $X = X_1 \times \cdots \times X_p$ and where $P_i : X \to X_i$ are the canonical projections is trivial, and any $\mu \in \Pi(\nu_1, \dots, \nu_p)$ is a marginal barycenter.*
- *We can assume without loss of generality that the projections P_i are surjective, since the spaces X_i could be replaced by $P_i(X_i)$.*

4.2.1 Study of the dual

Definitions

We denote by $\mathcal{M}(\mathbb{R}^{d_i})$ the space of bounded Radon measures on $\mathbb{R}^{d_i}$, identified with the dual of $\mathcal{C}_0(\mathbb{R}^{d_i})$, the set of continuous functions on $\mathbb{R}^{d_i}$ vanishing at infinity. We also denote by $\mathcal{M}_+^1(\mathbb{R}^d)$ the set of Radon probability measures on $\mathbb{R}^d$.

Following the same reasoning as Agueh and Carlier, 2011, we define

$$F_i = (1 + |.|^2)\mathcal{C}_0(\mathbb{R}^{d_i}) = \left\{ f_i \in \mathcal{C}(\mathbb{R}^{d_i}); \lim_{|x| \to +\infty} \frac{|f_i(x)|}{1 + |x|^2} = 0 \right\}.$$

The dual of F_i is identified with $F_i' = \left\{ \mu \in \mathcal{M}(\mathbb{R}^{d_i}); \ (1 + |.|^2)\mu \in \mathcal{M}(\mathbb{R}^{d_i}) \right\}$. We define as well F by

$$F = \left\{ f \in (1 + |.|^2)\mathcal{C}_0(\mathbb{R}^d); \ \exists (f_1, \dots, f_p) \in F_1 \times \cdots \times F_p; \sum_{i=1}^{p} f_i \circ P_i = f \right\},$$

equipped with the norm

$$\forall f \in F, ||f||_F = \sup_{x \in \mathbb{R}^d} \frac{|f(x)|}{1+|x|^2}.$$

Proposition 60. *The dual space F' of F is*

$$F' = \left\{ \nu \in \mathcal{M}(\mathbb{R}^d); \forall i, P_i \# \nu \in F_i' \right\}.$$

Proof. Let $E = \left\{ \nu \in \mathcal{M}(\mathbb{R}^d); \forall i, P_i \# \nu \in F_i' \right\}$. Let $i \in [\![1, p]\!]$ and $f_i \in F_i$. We have, for all $x \in \mathbb{R}^d$,

$$\begin{aligned} \frac{|f_i \circ P_i(x)|}{1+|x|^2} &= \frac{|f_i \circ P_i(x)|}{1+|P_i(x)|^2} \times \frac{1+|P_i(x)|^2}{1+|x|^2} \\ &\leq \frac{|f_i \circ P_i(x)|}{1+|P_i(x)|^2} \times \max(|||P_i|||^2, 1) \underset{|P_i(x)| \to +\infty}{\longrightarrow} 0. \end{aligned}$$

where $|||.|||$ is the operator norm, namely $|||P_i||| = \max_{|x|=1} |P_i(x)|$

Let $\epsilon > 0$. By the previous equation, we can choose A_1 such that $|P_i(x)| \geq A_1 \implies \frac{|f_i \circ P_i(x)|}{1+|x|^2} \leq \epsilon$.

By continuity of f_i, we know that $K = \sup_{|x| \leq A_1} |f_i(x)| < +\infty$. So we can choose A_2 such that $|x| \geq A_2 \implies \frac{K}{1+|x|^2} \leq \epsilon$.

Let x such that $|x| \geq \max(A_1, A_2)$. We have then two cases:

- If $|P_i(x)| \geq A_1$, then by definition of A_1, $\frac{|f_i \circ P_i(x)|}{1+|x|^2} \leq \epsilon$.
- If $|P_i(x)| < A_1$, then $|f_i \circ P_i(x)| \leq K$ and therefore $\frac{|f_i \circ P_i(x)|}{1+|x|^2} \leq \frac{K}{1+|x|^2} \leq \epsilon$.

This shows that $\forall x \in \mathbb{R}^d$ such that $|x| \geq \max(A_1, A_2)$, $\frac{|f_i \circ P_i(x)|}{1+|x|^2} \leq \epsilon$ and therefore

$$\frac{|f_i \circ P_i(x)|}{1+|x|^2} \underset{|x| \to +\infty}{\longrightarrow} 0.$$

So $f_i \circ P_i \in F$.

Let $\mu \in F'$, we have

$$\int_{\mathbb{R}^{d_i}} f_i d P_i \# \mu = \int_{\mathbb{R}^d} f_i \circ P_i d\mu < \infty$$

because $\mu \in F'$. As a consequence, $\forall i \in [\![1, p]\!]$, we have $P_i \# \mu \in F_i'$, and therefore $F' \subset E$.

Conversely, let $\mu \in E$. For $f \in F$, it exists $f_1, \ldots, f_p \in F_1 \times \cdots \times F_p$ such that $f = \sum_{i=1}^p f_i \circ P_i$, so we have $\mu(f) = \sum_{i=1}^p \int_{\mathbb{R}^{d_i}} f_i d(P_i \# \mu) < +\infty$. So $\mu \in F'$. □

We are interested in the following primal problem:

$$\inf_{\nu \in F' \cap \mathcal{M}_+^1(\mathbb{R}^d)} \sum_{i=1}^p \lambda_i W_2^2(\nu_i, P_i \# \nu). \tag{GWB}$$

We define, for $f_i \in F_i$ and $x_i \in \mathbb{R}^{d_i}$,

$$S_i f_i(x_i) = \inf_{y_i \in \mathbb{R}^{d_i}} \lambda_i |x_i - y_i|^2 - f_i(y_i).$$

and we consider the following maximization problem

$$\sup_{\forall i, f_i \in F_i;\ \sum_{k=1}^{p} f_k \circ P_k = 0} \sum_{i=1}^{p} \int_{\mathbb{R}^{d_i}} S_i f_i d\nu_i. \tag{GWB'}$$

Duality relation

The proof of the duality between (GWB) and (GWB') is a bit more complicated than for the regular barycenter formulation, because of the singularity of the space F. We will need the following assumption, whose validity shall be discussed later on.

Assumption 1. *Let $\epsilon > 0$. It exists $\eta_\epsilon > 0$ such that for all $f \in F$,*

$$||f|| \leq \eta_\epsilon \implies \exists (f_1, \dots, f_p) \in F_1 \times \dots \times F_p \text{ such that } \sum_i f_i \circ P_i = f \text{ and } \forall i, ||f_i|| \leq \epsilon.$$

We can now express the duality relation.

Proposition 61. *Under Assumption 1, the following duality relation holds*

$$\text{(GWB)} = \text{(GWB')}.$$

Proof. We suppose that Assumption 1 holds. The proof goes as follows: we first introduce convex functions H_is which help us express the transport cost differently. From those, we define a convex function H which enables to express the dual problem and the primal problem with its conjugate. We finally conclude thanks to a duality theorem.

Using classical optimal transport results, we first express the individual transport costs of the barycenter with respect to the function S_i:

$$\begin{aligned}
\lambda_i \mathcal{L}_{c_i}(\nu_i, P_i \# \nu) &= \sup \left\{ \int_{\mathbb{R}^{d_i}} f_i dP_i \# \nu + \int_{\mathbb{R}^{d_i}} g_i d\nu_i ; f_i, g_i \in \mathcal{C}_b(\mathbb{R}^{d_i}), f_i(x_i) + g_i(y_i) \leq \lambda_i |x_i - y_i|^2 \right\} \\
&= \sup \left\{ \int_{\mathbb{R}^d} f_i \circ P_i d\nu + \int_{x_i \in \mathbb{R}^{d_i}} \inf_{y_i \in \mathbb{R}^{d_i}} (\lambda_i |x_i - y_i|^2 - f_i(y_i)) d\nu_i, f_i \in \mathcal{C}_b(\mathbb{R}^{d_i}) \right\} \\
&= \sup \left\{ \int_{\mathbb{R}^d} f_i \circ P_i d\nu + \int_{x_i \in \mathbb{R}^{d_i}} \inf_{y_i \in \mathbb{R}^{d_i}} (\lambda_i |x_i - y_i|^2 - f_i(y_i)) d\nu_i, f_i \in F_i \right\} \\
&= \sup \left\{ \int_{\mathbb{R}^d} f_i \circ P_i(x) d\nu + \int_{\mathbb{R}^{d_i}} S_i f_i d\nu_i, f_i \in F_i \right\}
\end{aligned}$$

The switch from $\mathcal{C}_b(\mathbb{R}^{d_i})$ to F_i is valid since $\mathcal{C}_b(\mathbb{R}^{d_i}) \subset F_i$ is dense in F_i and because the expression we try to optimize is continuous in the functions f_i.

We define, for $f_i \in F_i$,

$$H_i(f_i) = -\int_{x_i \in \mathbb{R}^{d_i}} S_i f_i d\nu_i$$

.

It is then easy to show that $f_i \to S_i f_i$ is concave, and that H_i is convex. We will now study the conjugate of the H_i^*s, namely

$$H_i^*(\mu_i) = \sup_{f_i \in F_i} \left\{ \int_{\mathbb{R}^d} f_i d\mu_i - H_i(f_i) \right\}$$

Let $\mu_i \in F_i'$.

- If μ_i is not positive, then it exists $f_i \in F_i$ such that $f \leq 0$ and $\int f_i d\mu_i > 0$. Then considering tf_i, for $t > 0$, we have $tf_i \in F_i$, $H_i(tf_i) = \int_{\mathbb{R}^{d_i}} \sup_{y_i} tf_i(y_i) - \lambda_i |x_i - y_i|^2 d\nu_i \leq 0$ because $tf_i \leq 0$ and so

$$H_i^*(\mu_i) \geq t \int f_i d\mu_i - H_i(tf_i) \geq t \int f_i d\mu_i \to_{t \to +\infty} +\infty$$

- If μ_i is positive and $|\mu_i| \neq 1$, then using $f_i : x \to t, t \in \mathbb{R}$, we have $\lim_{|x_i| \to +\infty} \frac{f_i(x_i)}{1+|x_i|^2} = 0$, so $f_i \in F_i$, which gives $S_i f_i = t$ and

$$H_i^*(\mu_i) \geq \sup_t \int t d\mu_i - \int t d\nu_i = \sup_t t(|\mu_i| - 1) = +\infty.$$

- Finally, if $\mu_i \in \mathcal{P}(\mathbb{R}^d)$, then by Kantorovitch duality, we have directly that $H_i^*(\mu_i) = \mathcal{L}_{c_i}(\nu_i, \mu_i)$.

Therefore

$$H_i^*(\mu_i) = \begin{cases} \mathcal{L}_{c_i}(\nu_i, \mu_i) & \text{if } \mu_i \in \mathcal{P}(\mathbb{R}^d) \cap F_i' \\ +\infty & \text{otherwise} \end{cases}$$

And, in particular, for $\nu \in F'$:

$$H_i^*(P_i \# \nu) = \begin{cases} \mathcal{L}_{c_i}(\nu_i, P_i \# \nu) & \text{if } \nu \in \mathcal{M}_+^1(\mathbb{R}^d) \cap F' \\ +\infty & \text{otherwise} \end{cases}$$

If we denote by $K(\nu) = \sum_i H_i^*(P_i \# \nu)$, we have

$$(GWB) = \inf_{\nu \in F' \cap \mathcal{P}(\mathbb{R}^d)} \sum_i H_i^*(P_i \# \nu) = \inf_{\nu \in F'} K(\nu) = -K^*(0)$$

For $f \in (1 + |.|^2)\mathcal{C}_0(\mathbb{R}^d)$, we define

$$H(f) = \inf \left\{ \sum_i H_i(f_i); \sum_i f_i \circ P_i = f \right\}$$

with the convention $\inf \emptyset = +\infty$.

H is convex, $\text{dom}(H) = F$ and we have, for $\nu \in F'$,

$$\begin{aligned}
H^*(\nu) &= \sup_{f\in F}\left\{\int_{\mathbb{R}^d} f d\nu - \inf_{\sum_i f_i\circ P_i = f}\sum_i H_i(f_i)\right\}\\
&= \sup_{f\in F}\left\{\int_{\mathbb{R}^d} f d\nu + \sup_{\sum_i f_i\circ P_i = f} -\sum_i H_i(f_i)\right\}\\
&= \sup_{f\in F, \sum_i f_i\circ P_i = f}\left\{\int_{\mathbb{R}^d} f d\nu - \sum_i H_i(f_i)\right\}\\
&= \sup_{f_i\in F_i}\left\{\sum_i \int_{\mathbb{R}^d} f_i\circ P_i d\nu - H_i(f_i)\right\}\\
&= \sum_i \sup_{f_i\in F_i}\left\{\int_{\mathbb{R}^d} f_i dP_i\#\nu - H_i(f_i)\right\} = \sum_i H_i^*(P_i\#\nu) = K(\nu)
\end{aligned}$$

So $H^* = K$ and we have $(GWB) = -K^*(0) = -H^{**}(0)$ and $-H(0) = (P)$.
Let us show that $H(0) = H^{**}(0)$. We have, using that $\nu_i \in F_i'$,

$$H_i(f_i) = -\int_{x_i\in\mathbb{R}^{d_i}} \inf_{y_i}(\lambda_i|x_i - y_i|^2) - f_i(y_i)d\nu_i \geq f_i(0) - \lambda_i\int_{x_i\in\mathbb{R}^{d_i}}|x_i|^2 d\nu_i > -\infty$$

So H_i and therefore H cannot go to $-\infty$.

Then, using Assumption 1 for $\epsilon = \frac{1}{2}\min\{\lambda_1,\dots,\lambda_p\}$, we have $\eta_\epsilon > 0$ such that for all satisfying $||f||_F \leq \eta$, there exists $f_1,\dots,f_p$ such that $\sum_i f_i\circ P_i = f$ and $\forall i, ||f_i|| \leq \epsilon$. So we have, for such a function f:

$$\begin{aligned}
H(f) \leq \sum_i H_i(f_i) &= \sum_i \int_{x_i\in\mathbb{R}^{d_i}} \sup_{y_i}(f(y_i) - \lambda_i|x_i - y_i|^2)d\nu_i\\
&\leq \sum_i \int_{x_i\in\mathbb{R}^{d_i}} \sup_{y_i}(\frac{\lambda_i}{2}(1+|y_i|^2) - \lambda_i|x_i - y_i|^2)d\nu_i\\
&\leq \sum_i \int_{x_i\in\mathbb{R}^{d_i}} \lambda_i(1+|x_i|^2)d\nu_i \leq 1 + \sum_i \lambda_i\int_{x_i\in\mathbb{R}^{d_i}}|x_i|^2 d\nu_i < \infty
\end{aligned}$$

So H is bounded around 0.

The function $\widetilde{H} = H_{|F}$ is well-defined, convex, and $(\overset{\circ}{\text{dom}}\widetilde{H}) = F$. In particular $0 \in (\overset{\circ}{\text{dom}}\widetilde{H})$, so by standard convex analysis result (see for instance Proposition 5.2 Chapter 1 from Ekeland and Temam, 1999), $\partial\widetilde{H}(0) \neq \emptyset$.

Since $\forall f \in (1+|.|^2)\mathcal{C}^0(\mathbb{R}^d), H(f) = \begin{cases}\widetilde{H}(f) & \text{if } f\in F\\ +\infty & \text{otherwise}\end{cases}$, we have $\partial\widetilde{H}(0) \subset \partial H(0)$, so $\partial H(0) \neq \emptyset$. Therefore, by equality (5.3) from Ekeland and Temam, 1999, we have $H(0) = H^{**}(0)$.
So we have $(GWB') = (GWB)$

□

Concerning Assumption 1, we were only able to prove it in particular cases, although we conjecture that it holds in more general cases.

Proposition 62. *If at least one space $\mathbb{R}^{d_i}$ has the same dimension as $\mathbb{R}^d$, we have $F = (1+|.|^2)\mathcal{C}_0(\mathbb{R}^d)$ and Assumption 1 holds.*

This is roughly the case of study of Abraham et al., 2017 and the proof is obvious (just take $f_j = f$ with j such that $d_j = d$ and $\forall i \neq j, f_i = 0$). The following proposition gives another case of validity of Assumption 1.

Proposition 63. *If $P_1, \dots, P_p$ satisfy the coordinate projection condition,* i.e. *if there exists a basis $\mathcal{B}$ such that*

$$\forall i \in [\![1,p]\!], Ker(P_i) = Vect(B_i), B_i \subset \mathcal{B},$$

then Assumption 1 holds.

Proof. We show the result by induction on the number p of functions.

Initialization: For $p = 1$, since $f \in F$, we have $f_1 \circ P_1 = f$. We choose an arbitrary linear pseudo-inverse $\widetilde{P_1}^{-1}$ of P_1 such that $\forall x_1 \in \mathbb{R}^{d_1}, P_1(\widetilde{P_1}^{-1}(x_1)) = x_1$.

So we have,$\forall x_1 \in \mathbb{R}^{d_1}$,

$$\begin{aligned}\frac{|f_1(x_1)|}{1+|x_1|^2} &= \frac{|f_1 \circ P_1(\widetilde{P_1}^{-1}(x_1))|}{1+|x_1|^2} \\ &= \frac{|f(\widetilde{P_1}^{-1}(x_1))|}{1+|\widetilde{P_1}^{-1}(x_1)|^2} \times \frac{1+|\widetilde{P_1}^{-1}(x_1)|^2}{1+|x_1|^2} \\ &\leq ||f|| \times \max(|||\widetilde{P_1}^{-1}|||^2, 1),\end{aligned}$$

so we can take $\eta_\epsilon = \frac{\epsilon}{\max(|||\widetilde{P_1}^{-1}|||^2,1)}$.

Heredity: Let suppose the result for $p \geq 1$, and let $\epsilon > 0$ and $f \in F$.

Let $\mathcal{B} = (e_1, \dots, e_d)$ a common basis for all the $Ker(P_i)$. For all $i \in [\![1,p]\!]$, it exists $R_i \subset [\![1,d]\!]$ such that $\forall x = \sum_j \lambda_j e_j \in \mathbb{R}^d, P_i(x) = \sum_{j\in R_i} \lambda_j P_i(e_j)$, and $\{P_i(e_j); j \in R_i\}$ is a basis of $\mathbb{R}^{d_i}$ because of the surjectivity of P_i.

For $x_i = \sum_{j\in R_i} \lambda_i P_i(e_j) \in \mathbb{R}^{d_i}$, we define $\widetilde{P_i}^{-1}(x_i) = \sum_{j\in R_i} \lambda_i e_j$. We have, by construction, $\widetilde{P_i}^{-1} \circ P_i(\sum_j \lambda_j e_j) = \sum_{j\in R_i} \lambda_j e_j$.

We define $\widetilde{f_1}$ by

$$\forall x_1 \in \mathbb{R}^{d_1}, \widetilde{f_1}(x_1) = f_1(x_1) + \sum_{i>1} f_i \circ P_i \circ \widetilde{P_1}^{-1}(x_1),$$

and $\widetilde{f_i}$ for $i > 1$ by

$$\forall x_i \in \mathbb{R}^{d_i}, \widetilde{f_i}(x_i) = f_i(x_i) - f_i \circ P_i \circ \widetilde{P_1}^{-1} \circ P_1 \circ \widetilde{P_i}^{-1}(x_i).$$

Then, we have by construction,

$$\forall x_1 \in \mathbb{R}^{d_1}, f(\widetilde{P_1}^{-1}(x_1)) = \widetilde{f_1}(x_1).$$

So, following the same reasoning as in the initialization, we have the property

$$\forall \epsilon_1 > 0, \exists \eta \text{ such that } ||f|| \leq \eta \implies ||\widetilde{f_1}|| \leq \epsilon_1. \tag{4.2}$$

Besides, $\forall x \in \mathbb{R}^d$

$$\sum_i \widetilde{f}_i \circ P_i(x) = \sum_i f_i \circ P_i(x) + \sum_{i>1} \left(f_i \circ P_i \circ \widetilde{P_1}^{-1} \circ P_1(x) - f_i \circ P_i \circ \widetilde{P_1}^{-1} \circ P_1 \circ \widetilde{P_i}^{-1} \circ P_i(x) \right)$$
$$= f(x) + \sum_{i>1} \left(f_i \circ P_i \circ \widetilde{P_1}^{-1} \circ P_1(x) - f_i \circ P_i \circ \widetilde{P_1}^{-1} \circ P_1 \circ \widetilde{P_i}^{-1} \circ P_i(x) \right).$$

If $x = \sum_j \lambda_j e_j$, then

$$P_i \circ \widetilde{P_1}^{-1} \circ P_1 \circ \widetilde{P_i}^{-1} \circ P_i(x) = \sum_{j \in R_1 \cap R_i} \lambda_j P_i(e_j)$$

and

$$P_i \circ \widetilde{P_1}^{-1} \circ P_1(x) = \sum_{j \in R_1 \cap R_i} \lambda_j P_i(e_j).$$

It follows that for all i, we have

$$f_i \circ P_i \circ \widetilde{P_1}^{-1} \circ P_1(x) = f_i \circ P_i \circ \widetilde{P_1}^{-1} \circ P_1 \circ \widetilde{P_i}^{-1} \circ P_i(x),$$

so $\sum_i \widetilde{f}_i \circ P_i(x) = f(x)$. Now, $\forall x \in \mathbb{R}^d$,

$$\begin{aligned} \frac{|\sum_{i>1} \widetilde{f}_i \circ P_i(x)|}{1+|x|^2} &= \frac{|f(x) - \widetilde{f}_1 \circ P_1(x)|}{1+|x|^2} \\ &\leq \frac{|f(x)|}{1+|x|^2} + \frac{\widetilde{f}_1 \circ P_1(x)}{1+|P_1(x)|^2} \times \frac{1+|P_1(x)|^2}{1+|x|^2} \\ &\leq ||f|| + ||\widetilde{f}_1|| \times \max(1, |||P_1|||^2). \end{aligned} \tag{4.3}$$

The rest of the proof is a combination of Property (4.2), Equation (4.3) and of the induction hypothesis.

Applying the induction hypothesis for $P_2, \dots, P_p$ to $\widetilde{f} = \sum_{i>1} \widetilde{f}_i \circ P_i$, we find $\eta_\epsilon^{(p-1)}$ such that

$$\|\widetilde{f}\| < \eta_\epsilon^{(p-1)} \implies \exists \hat{f}_2, \dots, \hat{f}_p \text{ such that } \sum_i \hat{f}_i \circ P_i = \widetilde{f} \text{ and } \forall i \in [\![2, p]\!], \|f_i\| \leq \epsilon.$$

Using (4.2), we find η such that $||f|| \leq \eta \implies ||\widetilde{f}_1|| \leq \frac{\eta_\epsilon^{(p-1)}}{2\max(1, |||P_1|||^2)}$. Eventually, using (4.3), for f such that $||f|| \leq \eta_\epsilon = \min(\eta, \frac{\eta_\epsilon^{(p-1)}}{2})$,

$$||\sum_{i>1} \widetilde{f}_i \circ P_i|| \leq \frac{\eta_\epsilon^{(p-1)}}{2} + \frac{\eta_\epsilon^{(p-1)}}{2\max(1, |||P_1|||^2)} \times \max(1, |||P_1|||^2) = \eta_\epsilon^{(p-1)},$$

which concludes the induction. □

4.2.2 Existence of solutions for (GWB)

We show in the following that the primal minimization problem (GWB) has solutions and that a solution is generally not unique. The duality is not used here.

Proposition 64. *The problem (GWB) has solutions.*

Proof. First, assume that $\sum_{i=1}^p |P_i(x)|^2$ is coercive. Let ν^n be a minimizing sequence for (GWB). It follows that the whole sequence $W_2(P_i\#\nu^n, \nu_i)$ is upper bounded. For any coupling (X, Y) of $(P_i\#\nu^n, \nu_i)$, we can write (using Cauchy-Schwarz)

$$\mathbb{E}(|X-Y|^2) \geq [\sqrt{\mathbb{E}(|X|^2)} - \sqrt{\mathbb{E}(|Y|^2)}]^2.$$

It follows that $\sqrt{\mathbb{E}(|X|^2)} \leq \sqrt{\mathbb{E}(|X-Y|^2)} + \sqrt{\mathbb{E}(|Y|^2)}$ and this is valid for any coupling (X, Y) of $(P_i\#\nu^n, \nu_i)$, in particular for a coupling which attains the minimum of $\mathbb{E}(|X-Y|^2)$. For this particular coupling, we get

$$\int_{\mathbb{R}^d} |x|^2 d(P_i\#\nu^n) = \mathbb{E}(|X|^2) \leq \left(W_2(P_i\#\nu^n, \nu_i) + \sqrt{\int_{\mathbb{R}^d} |x|^2 d\nu_i} \right)^2$$

The right terms are both bounded independently of n and since i takes only a finite number of values, it can be upper bounded also independently of i. We call M such an upper bound. It follows that

$$\int_{\mathbb{R}^d} \sum_{i=1}^p |P_i(x)|^2 d\nu^n = \sum_{i=1}^p \int_{\mathbb{R}^d} |x|^2 d(P_i\#\nu^n) \leq Mp.$$

We have assumed that $\sum_{i=1}^p |P_i(x)|^2$ is coercive. It follows that for any $\epsilon > 0$, we can find a compact K such that $\forall x \notin K$, $\sum_{i=1}^p |P_i(x)|^2 \geq \frac{Mp}{\epsilon}$. Thus,

$$\nu^n(\mathbb{R}^d \setminus K) \leq \frac{\epsilon}{Mp} \int_{\mathbb{R}^d \setminus K} \sum_{i=1}^p |P_i(x)|^2 d\nu^n \leq \frac{\epsilon}{Mp} \int_{\mathbb{R}^d} \sum_{i=1}^p |P_i(x)|^2 d\nu^n \leq \epsilon.$$

Such a compact K can be found for any positive ϵ, so the sequence ν^n is tight. It follows, by Prokhorov theorem, that there exists a probability measure ν and a subsequence of (ν^n) which converges weakly to ν. Without loss of generality, we still call this sequence (ν^n). It is easy to show that for each $i = 1, \dots, p$, $P_i\#\nu^n$ also converges weakly to $P_i\#\nu$, and thus $W_2(P_i\#\nu^n, \nu_i) \longrightarrow W_2(P_i\#\nu, \nu_i)$ since W_2 metrizes the weak convergence on $\mathcal{P}_2(\mathbb{R}^d)$. Thus, ν is a solution of (GWB).

If $\sum_{i=1}^p |P_i(x)|^2$ is not coercive, it means that some directions of $\mathbb{R}^d$ (all directions in $\bigcap_{i=1}^p \mathrm{Ker} P_i$) are not seen by the projections P_i. In these directions, the minimizing sequence (ν^n) has no reason to converge and could for instance oscillate between different measures, without affecting the value of the cost $\sum_{i=1}^p \lambda_i W_2^2(\nu_i, P_i\#\nu^n)$. In this case, we can instead construct a solution ν of the problem in $(\bigcap_{i=1}^p \mathrm{Ker} P_i)^\perp$, since $\sum_{i=1}^p |P_i(x)|^2$ is coercive on this subspace. Any probability measure on the whole space $\mathbb{R}^d$ with marginal ν on $(\bigcap_{i=1}^p \mathrm{Ker} P_i)^\perp$ is a solution of (GWB). □

We don't have uniqueness of the solution in general. Even if $\sum_{i=1}^p |P_i(x)|^2$ is coercive, if ν is a solution, any probability distribution μ on $\mathbb{R}^d$ satisfying $P_i\#\nu = P_i\#\mu$ for all $i = 1, \dots, p$ is also a solution of the minimization problem. The question of the existence and uniqueness of probability measures with known and overlapping marginals is a difficult and important problem in probability, see for instance the recent paper Kazi-Tani and Rullière, 2019.

Observe that if one of the P_i is an isomorphism of $\mathbb{R}^d$ and the corresponding ν_i is absolutely continuous, we have uniqueness since $\nu \longrightarrow \sum_{i=1}^p W_2^2(\nu_i, P_i\#\nu)$ is strictly convex in this case.

Figure 4.2 shows an example where the measures $\nu_1, \dots, \nu_p$ are several 1d projections of a discrete measure (in yellow) in $\mathbb{R}^2$. In this case, the problem (GWB) has at least one solution given by the yellow distribution and for which the value of the energy is 0. We show in black the reconstruction of a probability measure with exactly the same projections. The algorithm used for this reconstruction will be explained in Section 4.4. We see that when the number of 1d projection increases, the reconstructed measure tends toward the discrete yellow measure.

4.2.3 Link between (GWB) and multi marginal optimal transport

In the following, we write $D = d_1 + \dots + d_p$ and we assume that $\sum_{i=1}^p \lambda_i P_i^T P_i$ is invertible. For $\vec{x} = (x_1, \dots, x_p) \in \mathbb{R}^{d_1} \times \dots \times \mathbb{R}^{d_p}$, we define

$$B(\vec{x}) = B(x_1, \dots, x_p) = \arg\min_{y \in \mathbb{R}^d} \sum_{i=1}^p \lambda_i |x_i - P_i(y)|^2 = \left(\sum_{i=1}^p \lambda_i P_i^T P_i \right)^{-1} \sum_{i=1}^p \lambda_i P_i^T(x_i),$$

and we also define the cost $c(\vec{x})$ by

$$c(\vec{x}) = \sum_{i=1}^p \lambda_i |x_i - P_i(B(\vec{x}))|^2.$$

We propose to study the multimarginal problem for the measures $\nu_1, \dots, \nu_p$ and cost function c, *i.e.*

$$\inf \left\{ \int_{\mathbb{R}^{d_1+\dots+d_p}} c(\vec{x}) d\gamma(\vec{x}), \gamma \in \Pi(\nu_1, \dots, \nu_p) \right\} \tag{MM}$$

Proposition 65. *The infimum in* (MM) *and* (GWB) *are equal. If γ is a solution of* (MM), *then $\nu = B\#\gamma$ is a solution of* (GWB).

Proof. Let $\gamma \in \Pi(\nu_1, \dots, \nu_p)$ and define $\nu = B\#\gamma$. For all i, we define π_i the projection from $\mathbb{R}^D = \mathbb{R}^{d_1} \times \dots \times \mathbb{R}^{d_p}$ to $\mathbb{R}^{d_i}$ such that $\pi_i(x_1, \dots, x_p) = x_i$, and define $\gamma_i = (\pi_i, P_i \circ B)\#\gamma$. We easily check that $\gamma_i \in \Pi(\nu_i, P_i\#\nu)$, since $\forall A \in \mathbb{R}^{d_i}$, $\gamma_i[A \times \mathbb{R}^{d_i}] = \gamma[(\pi_i, P_i \circ B)^{-1}(A, \mathbb{R}^{d_i})] = \gamma[\{x, x_i \in A\}] = \nu_i[A]$ and $\gamma_i[\mathbb{R}^{d_i} \times A] = (P_i \circ B)\#\gamma[A] = P_i\#(B\#\gamma)[A] = P_i\#\nu[A]$. Thus,

$$W_2^2(\nu_i, P_i\#\nu) \leq \int_{\mathbb{R}^{d_i} \times \mathbb{R}^{d_i}} |x - y|^2 d\gamma_i(x, y) = \int_{\mathbb{R}^D} |x_i - (P_i \circ B)(\vec{x})|^2 d\gamma(\vec{x})$$

As a consequence, for all $\gamma \in \Pi(\nu_1, \dots, \nu_p)$,

$$\sum_i \lambda_i W_2^2(\nu_i, P_i\#\nu) \leq \int_{\mathbb{R}^D} c(\vec{x}) d\gamma(\vec{x}).$$

This holds for any $\gamma \in \Pi(\nu_1, \dots, \nu_p)$ and thus (GWB) $\leq$ (MM).

Conversely, let $\mu \in F' \cap \mathcal{M}_+^1(\mathbb{R}^d)$ and $\eta_i \in \Pi(\nu_i, P_i\#\mu)$. By the disintegration theorem, there exists a family of probability measures $(\eta_i^y)_{y \in \mathbb{R}^{d_i}}$ such that $\eta_i = \eta_i^y \otimes$

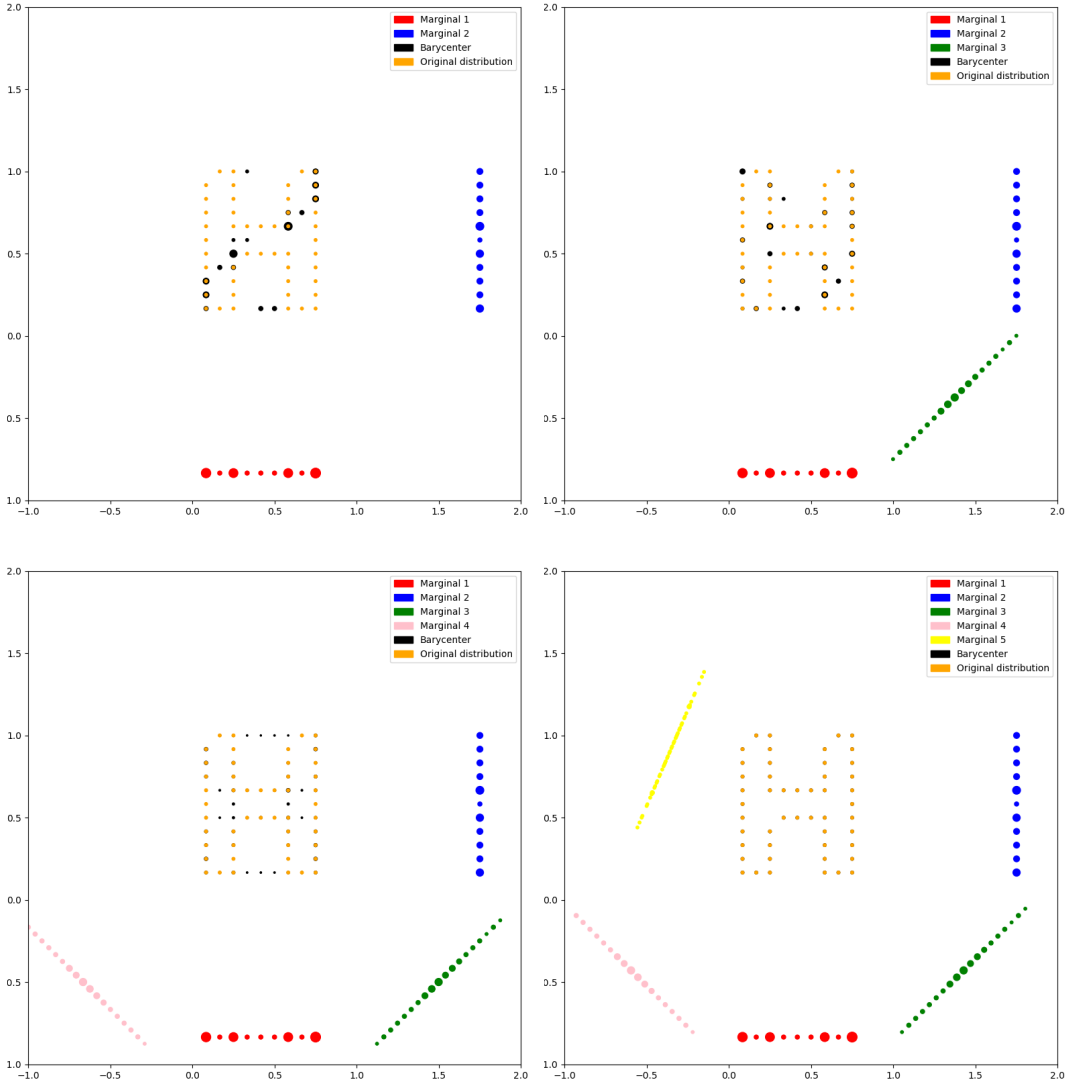


FIGURE 4.2: In this example, the measures $\nu_1, \dots, \nu_p$ are several 1d projection of a discrete measure (in yellow) in $\mathbb{R}^2$. The number of projection varies from 2 to 5. The reconstructed generalized barycenter is shown in black, illustrating the non uniqueness of the solution (this black probability measure has exactly the same projections as the yellow one). The more marginals there are, the more accurate is the reconstruction. For 5 projections, the reconstructed generalized barycenter is the same as the original distribution.

$(P_i\#\mu)$, which implies that for all f positive and measurable on $\mathbb{R}^{d_i}\times\mathbb{R}^{d_i}$

$$\begin{aligned}\int_{\mathbb{R}^{d_i}\times\mathbb{R}^{d_i}} f(x_i,y)d\eta_i(x_i,y) &= \int_{\mathbb{R}^{d_i}}\left(\int_{\mathbb{R}^{d_i}} f(x_i,y)d\eta_i^y(x_i)\right)d(P_i\#\mu)(y)\\ &= \int_{\mathbb{R}^d}\left(\int_{\mathbb{R}^{d_i}} f(x_i,P_i(y))d\eta_i^{P_i(y)}(x_i)\right)d\mu(y)\\ &== \int_{\mathbb{R}^d}\left(\int_{\mathbb{R}^D} f(x_i,P_i(y))d\eta_1^{P_1(y)}(x_1)\dots d\eta_p^{P_p(y)}(x_p)\right)d\mu(y).\end{aligned}$$

We define the probability measure θ by

$$\int_{\mathbb{R}^D} f(\vec{x})d\theta(\vec{x}) = \int_{\mathbb{R}^d}\left(\int_{\mathbb{R}^D} f(\vec{x})d\eta_1^{P_1(y)}(x_1)\dots d\eta_p^{P_p(y)}(x_p)\right)d\mu(y).$$

By construction, $\theta\in\Pi(\nu_1,\dots,\nu_p)$. Indeed,

$$\begin{aligned}\int_{\mathbb{R}^D} f(x_i)d\theta(\vec{x}) &= \int_{\mathbb{R}^d}\left(\int_{\mathbb{R}^D} f(x_i)d\eta_1^{P_1(y)}(x_1)\dots d\eta_p^{P_p(y)}(x_p)\right)d\mu(y)\\ &= \int_{\mathbb{R}^{d_i}\times\mathbb{R}^{d_i}} f(x_i)d\eta_i(x_i,y) = \int_{\mathbb{R}^{d_i}} f(x)d\nu_i(x).\end{aligned}$$

Finally, for any distribution μ and $\eta_1,\dots,\eta_p$ in $\Pi(\nu_1,P_1\#\mu),\dots,\Pi(\nu_p,P_p\#\mu)$, we have

$$\begin{aligned}\sum_i\lambda_i\int_{\mathbb{R}^{d_i}\times\mathbb{R}^{d_i}}|x_i-y_i|^2d\eta_i(x_i,y_i) &= \sum_i\lambda_i\int_{\mathbb{R}^d}\left(\int_{\mathbb{R}^D}|x_i-P_i(y)|^2)d\eta_1^{P_1(y)}(x_1)\dots d\eta_p^{P_p(y)}(x_p)\right)d\mu(y)\\ &= \int_{\mathbb{R}^d\times\mathbb{R}^D}\sum_i\lambda_i|x_i-P_i(y)|^2d\eta_1^{P_1(y)}(x_1)\dots d\eta_p^{P_p(y)}(x_p)d\mu(y)\\ &\geq \int_{\mathbb{R}^d\times\mathbb{R}^D}\sum_i\lambda_i|x_i-P_i(B(\vec{x}))|^2d\eta_1^{P_1(y)}(x_1)\dots d\eta_p^{P_p(y)}(x_p)d\mu(y)\\ &= \int_{\mathbb{R}^d}c(\vec{x})d\theta(\vec{x}) \geq \text{ (MM)}\end{aligned}$$

So we have, for any $\mu\in F'\cap\mathcal{M}_+^1(\mathbb{R}^d)$,

$$\sum_i\lambda_iW_2^2(\nu_i,P_i\#\mu) = \inf_{\eta_1,\dots,\eta_p}\sum_i\lambda_i\int_{\mathbb{R}^{d_i}\times\mathbb{R}^{d_i}}|x_i-y_i|^2d\eta_i(x_i,y) \geq \text{ (MM)}$$

It follows that (GWB) $\geq$ (MM), thus (GWB) $=$ (MM).

We have seen that if (MM) admits a solution γ, then defining $\nu = B\#\gamma$, we have

$$\text{(GWB)} \leq \sum_i\lambda_iW_2^2(\nu_i,P_i\#\nu) \leq \int_{\mathbb{R}^d}c(x)d\gamma(x) = \text{ (MM)} = \text{ (GWB)}$$

This yields

$$\text{(GWB)} = \sum_i\lambda_iW_2^2(\nu_i,P_i\#\nu)$$

□

The previous proposition clarifies the link between the (GWB) and the (MM) problems. Since the (MM) can be solved by linear programming, we can derive from this equivalence a way to solve (GWB). Before that, we give in the following

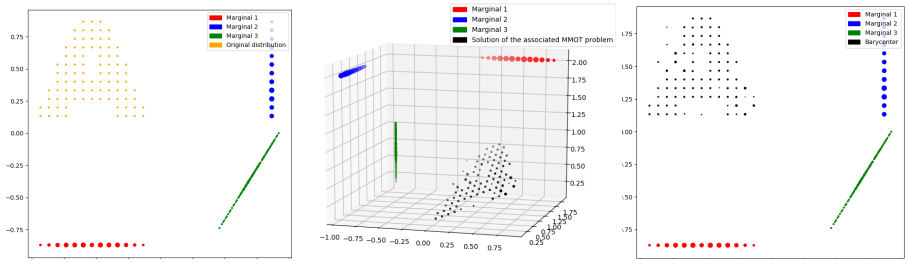


FIGURE 4.3: Left: A-shaped original distribution (in yellow) and three 1D projections. Center: Solution of the corresponding multi-marginal problem (MM) for these three projections. The solution of the multimarginal problem (MM) is supported by a plane, as shown in Proposition 66. Right: Generalized barycenter (in black).

some insights on a specific case where all the probability measures ν_i are projections from the same high dimensional probability measure ν.

Proposition 66. *Assume that ν is in $\mathcal{P}_2(\mathbb{R}^d)$ and for each i in $\{1,\dots,p\}$, $\nu_i = P_i\#\nu$. Let $P : \mathbb{R}^d \to \mathbb{R}^{d_1+\dots d_p}$ be the linear application defined by $P(x) = (P_1(x),\dots,P_p(x))$ $\forall x \in \mathbb{R}^d$. The probability measure ν is clearly a solution of (GWB), and $\gamma = P\#\nu$ is a solution of (MM). If $d < D$, γ is supported on a subspace of dimension d of $\mathbb{R}^D$.*

Proof. The fact that ν is solution of (GWB) is obvious by definition of the ν_i. Also, by definition of B, we have for each $x \in \mathbb{R}^d$, $B(P(x)) = x$. For $\gamma = P\#\nu$, clearly $\gamma \in \Pi(\nu_1,\dots,\nu_p)$ and we have

$$\int_{\mathbb{R}^{d_1+\dots+d_p}} \sum_{i=1}^{p} \lambda_i |x_i - P_i(B(x)|^2 d\gamma(x) = 0,$$

which means that $\gamma = P\#\nu$ is a solution of (MM). Since P is linear, and ν lives in $\mathbb{R}^d$, if $D > d$ then γ lives in a subspace of dimension d of $\mathbb{R}^D$.

□

For instance, for a probability measure ν on the plane ($d = 2$) and three linear projections on lines P_1, P_2, P_3, then the solution $\gamma = P\#\nu$ of the multimarginal problem (MM) on $\mathbb{R}^3$ will be supported by a plane.

4.3 Solutions of (GWB) for Gaussian distributions

When $\nu_1,\dots,\nu_p$ are normal distributions, we show below that the generalized Wasserstein barycenter can also be a normal distribution, and how its parameters can be computed in practice.

Proposition 67. *If $\nu_1,\dots,\nu_p$ are all non-degenerate Gaussian distributions with $\forall i \in [\![1,p]\!], \nu_i = \mathcal{N}(\mu_i, S_i)$ and ν is a Gaussian distribution with expectation μ and covariance S, with*

$$\mu = B(\mu_1,\dots,\mu_p)$$

and S a symmetric positive definite solution of

$$S^{1/2}\left(\sum_i \lambda_i P_i^T P_i\right) S^{1/2} = \sum_i \lambda_i \left(S^{1/2}\left(P_i^T S_i P_i\right) S^{1/2}\right)^{1/2}, \tag{4.4}$$

then ν is a generalized Wasserstein barycenter for the GWB problem associated with the ν_i.

Proof. The quadratic transport cost between two distributions η and η' is equal to

$$W_2^2(\eta,\eta') = |\mathbb{E}[\eta] - \mathbb{E}[\eta']|^2 + W_2^2\left(\eta - \mathbb{E}[\eta], \eta' - \mathbb{E}[\eta']\right).$$

Therefore, we have

$$\mathbb{E}[\nu] = \inf_x \sum_i \lambda_i |P_i(x) - \mathbb{E}[\nu_i]|^2 = B(\mu_1,\dots,\mu_p).$$

and we can assume that the ν_i have 0-mean.

Let S be a symmetric positive definite solution of (4.4) and let $\nu = \mathcal{N}(0,S)$. We know from Villani, 2003 that there exist convex potentiels ψ_i such that $\nabla\psi_i$ is the Brenier's map transporting ν_i to $P_i\#\nu$ and from Knott and Smith, 1984 that $\nabla\psi_i^*$ is a linear map represented by the matrix

$$T_i = S_i^{1/2}\left(S_i^{1/2}P_iSP_i^TS_i^{1/2}\right)^{-1/2}S_i^{1/2}.$$

Let $K_i = S_i^{1/2}$ and $L_i = S^{1/2}P_i^T$. Using the identity

$$\left(L_iK_i^2L_i^T\right)^{1/2} = L_iK_i\left(K_iL_i^TL_iK_i\right)^{-1/2}K_iL_i^T,$$

we can rewrite Equation (4.4) as

$$\sum_i \lambda_i L_iK_i\left(K_iL_i^TL_iK_i\right)^{-1/2}K_iL_i = S^{1/2}\left(\sum_i \lambda_i P_i^TP_i\right)S^{1/2},$$

which gives

$$\sum_i \lambda_i S^{1/2}P_i^TS_i^{1/2}\left(S_i^{1/2}P_iSP_i^TS_i^{1/2}\right)^{-1/2}S_i^{1/2}P_iS^{1/2} = S^{1/2}\left(\sum_i \lambda_i P_i^TP_i\right)S^{1/2},$$

and then, since S is invertible,

$$\sum_i \lambda_i P_i^TT_iP_i = \sum_i \lambda_i P_i^TP_i.$$

So we have,

$$\sum_i \lambda_i \nabla(\psi_i^*\circ P_i) = \sum_i \lambda_i \nabla\frac{|P_i(.)|^2}{2}.$$

Integrating the previous equality, we have a constant $C\in\mathbb{R}$ such that

$$\forall x\in\mathbb{R}^d, \sum_i \lambda_i\left(\frac{|P_i(x)|^2}{2} - \psi_i^*(P_i(x))\right) = C.$$

By Kantorovitch duality, we have

$$\begin{aligned}\frac{1}{2}W_2^2(\nu_i,P_i\#\nu) &= \int_{\mathbb{R}^{d_i}}\left(\frac{|x_i|^2}{2} - \psi_i(x_i)\right)d\nu_i(x_i) + \int_{\mathbb{R}^{d_i}}\left(\frac{|y_i|^2}{2} - \psi_i^*(y_i)\right)dP_i\#\nu(y_i)\\ &= \int_{\mathbb{R}^{d_i}}\left(\frac{|x_i|^2}{2} - \psi_i(x_i)\right)d\nu_i(x_i) + \int_{\mathbb{R}^{d}}\left(\frac{|P_i(x)|^2}{2} - \psi_i^*(P_i(x))\right)d\nu(x)\end{aligned}$$

Therefore, summing over i we have

$$\begin{aligned}\frac{1}{2}\sum_i \lambda_i W_2^2(\nu_i, P_i\#\nu) &= \sum_i \lambda_i \int_{\mathbb{R}^{d_i}} \left(\frac{|x_i|^2}{2} - \psi_i(x_i)\right) d\nu_i(x_i) + \int_{\mathbb{R}^d} \sum_i \lambda_i \left(\frac{|P_i(x_i)|^2}{2} - \psi_i^*(P_i(x))\right) d\nu \\ &= \sum_i \lambda_i \int_{\mathbb{R}^{d_i}} \left(\frac{|x_i|^2}{2} - \psi_i(x_i)\right) d\nu_i(x_i) + C.\end{aligned}$$

Now, let $\mu \in F' \cap \mathcal{P}(\mathbb{R}^d)$. For $i \in [\![1, p]\!]$, using the conjugate inequality $\langle x_i, y_i\rangle \leq \psi_i(x) + \psi_i^*(y)$, we have

$$\frac{|x_i - y_i|^2}{2} \geq \frac{|x_i|^2}{2} - \psi_i(x_i) + \frac{|y_i|^2}{2} - \psi_i^*(y_i)$$

which gives after integration

$$\frac{W_2^2(\nu_i, P_i\#\mu)}{2} \geq \int_{\mathbb{R}^{d_i}} \left(\frac{|x_i|^2}{2} - \psi_i(x_i)\right) d\nu_i(x_i) + \int_{R^d} \left(\frac{|P_i(x)|^2}{2} - \psi_i^*(P_i(x))\right) d\mu(x).$$

Summing over i, we obtain

$$\begin{aligned}\frac{1}{2}\sum_i \lambda_i W_2^2(\nu_i, P_i\#\mu) &\geq \sum_i \lambda_i \int_{\mathbb{R}^{d_i}} \left(\frac{|x_i|^2}{2} - \psi_i(x_i)\right) d\nu_i(x_i) + \int_{R^d} \sum_i \lambda_i \left(\frac{|P_i(x)|^2}{2} - \psi_i^*(P_i(x))\right) d\mu(x) \\ &\geq \sum_i \lambda_i \int_{\mathbb{R}^{d_i}} \left(\frac{|x_i|^2}{2} - \psi_i(x_i)\right) d\nu_i(x_i) + C \\ &\geq \frac{1}{2}\sum_i \lambda_i W_2^2(\nu_i, P_i\#\nu),\end{aligned}$$

which proves that ν is a generalized Wasserstein barycenter.

□ □

Proposition 68. *Keeping the same notations as the previous proposition, if the equality $\sum_i \lambda_i P_i^T P_i = \delta I_d$ holds, with $\delta > 0$, then Equation (4.4) has a symmetric definite positive solution.*

Proof. In this case, Equation (4.4) becomes

$$S = \frac{1}{\delta}\sum_{i=1}^p \lambda_i \left(S^{1/2}\left(P_i^T S_i P_i\right) S^{1/2}\right)^{1/2}.$$

Let α_i and β_i be the lowest and highest eigenvalues of S_i and α, β such that

$$0 < \alpha \leq \frac{\min_i \lambda_i^2 \alpha_i}{\delta} \text{ and } \left(\sum_i \sqrt{\frac{\beta_i \lambda_i}{\delta}}\right)^2 \leq \beta.$$

We define $K_{\alpha,\beta}$ the convex and compact set of symmetric matrices S such that $\alpha \mathrm{I} \leq S \leq \beta \mathrm{I}$. For $S \in K_{\alpha,\beta}$, we define

$$F(S) = \frac{1}{\delta}\sum_{i=1}^p \lambda_i \left(S^{1/2}\left(P_i^T S_i P_i\right) S^{1/2}\right)^{1/2}.$$

We know that $\sum_i \lambda_i P_i^T P_i = \delta I_d$, so we have for all $y \in \mathbb{R}^d$,

$$\sum_i \lambda_i |P_i(y)|^2 = \delta |y|^2.$$

Therefore, $\forall y \in \mathbb{R}^d, \exists j \in [\![1,p]\!]$ such that $|P_j(y)| \geq \sqrt{\delta}|y|$ and $\forall y \in \mathbb{R}^d, \forall j \in [\![1,p]\!], |P_j(y)| \leq \sqrt{\frac{\delta}{\lambda_j}}|y|$.

Now, let $x \in \mathbb{R}^d$ and j such that $|P_j(S^{1/2}x)| \geq \sqrt{\delta}|S^{1/2}x|$. We have

$$|x^T S^{1/2} \left(P_j^T S_j P_j\right) S^{1/2} x|^2 \geq |P_j(S^{1/2}x)|^2 \times \alpha_j \geq \delta \alpha_j |S^{1/2}x|^2 \geq \delta \alpha_j \alpha |x|^2.$$

So we have

$$\frac{1}{\delta} \sum_i \lambda_i \left(S^{1/2} \left(P_i^T S_i P_i\right) S^{1/2}\right)^{1/2} \geq \sqrt{\alpha \left(\frac{\min_i \lambda_i^2 \alpha_i}{\delta}\right)} I_d \geq \alpha I_d.$$

Similarly, we have, for all $x \in \mathbb{R}^d$ and $i \in [\![1,p]\!]$,

$$|x^T S^{1/2} \left(P_i^T S_i P_i\right) S^{1/2} x| \leq |P_i(S^{1/2}x)|^2 \times \beta_i \leq \frac{\beta_i \delta}{\lambda_i} |S^{1/2}x|^2 \leq \frac{\delta \beta_i \beta}{\lambda_i} |x|^2,$$

so

$$\frac{1}{\delta} \sum_i \lambda_i \left(S^{1/2} \left(P_i^T S_i P_i\right) S^{1/2}\right)^{1/2} \leq \sum_i \sqrt{\frac{\lambda_i \beta_i \beta}{\delta}} I_d \leq \beta I_d.$$

Eventually, we showed that

$$\forall S \in K_{\alpha,\beta}, \alpha \mathrm{I} \leq F(S) \leq \beta \mathrm{I}$$

So F is a continuous self-map of $K_{\alpha,\beta}$. So, from Brouwer's fixed-point theorem, it exists a solution to (4.4) which is symmetric definite positive.

□

The condition $\sum_i \lambda_i P_i^T P_i = \delta I_d$ is quite restrictive. Figure 4.7 shows an example where such a condition is satisfied. If the P_i are orthogonal coordinate projections, then it means that each coordinate must be represented the same number of times. In general, it is not satisfied. However, we can still recast Equation (4.4) into the following fixed point equation:

$$S = \left(\sum_i \lambda_i P_i^T P_i\right)^{-1} \sum_i \lambda_i S^{-1/2} \left(S^{1/2} \left(P_i^T S_i P_i\right) S^{1/2}\right)^{1/2} S^{1/2}, \tag{4.5}$$

or formally into (up to the square root unicity)

$$S = \left(\sum_i \lambda_i P_i^T P_i\right)^{-1} \sum_i \lambda_i \left(\left(P_i^T S_i P_i\right) S\right)^{1/2}. \tag{4.6}$$

using the identity

$$\left(S^{-1/2} \left(S^{1/2} \left(P_i^T S_i P_i\right) S^{1/2}\right)^{1/2} S^{1/2}\right)^2 = \left(P_i^T S_i P_i\right) S.$$

Equation (4.5) and (4.6) could both lead to an iterative fixed point algorithm. However, these fixed point equations no longer being symmetric, we were unable to prove existence of symmetric solutions. In practice, we tried to iterate on Equation (4.6) (using the algorithm of Deadman, Higham, and Ralha, 2012 to compute square root of matrices with no negative eigenvalues), alternatively with a symmetrization of S, and this seems to converge toward a satisfying result. This is the algorithm used to compute Gaussian solutions of (GWB) in Section 4.4.

4.4 Experiments

This section gathers some experiments illustrating the behavior of the generalized Wasserstein Barycenters. Thanks to Proposition 65, we can compute the Generalized Wasserstein Barcyenter for discrete distributions using the multi-marginal Sinkhorn algorithm presented in Section 3.6.5. When the considered distributions are Gaussian, we use the fixed-point equation algortihm as presented in Section 4.3. We first show several results of generalized barycenters between disagreeing projections in 3 dimensions, illustrating how (GWB) solutions find a compromise between several distributions which do not coincide on their common subspaces. The section concludes with experiments on Gaussian distributions, first in low dimension and then in larger dimension with experiments on image patches.

4.4.1 Generalized barycenters in 3 dimensions between disagreeing marginals

Figures 4.4, 4.5 and 4.6 show several generalized barycenters computed with the Sinkhorn agorithm between different sets of disagreeing marginals. In Figures 4.4 and 4.5, we show on the left the three dimensional barycenter ν (black dots) between the original two dimensional distributions ν_i (colored dots, each color corresponding to a different i). On the right, we show for each i the superposition of $P_i\#\nu$ (black) and ν_i. For instance, in the top row experiment, the red square is narrower than the blue heart in their common dimension, and the barycenter has to compromise between these two shapes on this dimension.

4.4.2 Generalized Gaussian barycenters

We are able to compute solutions for (GWB) between discrete distributions with a few marginals, and between Gaussian distributions using the results of Section 4.3. Figure 4.7 shows two examples of a three dimensional Gaussian barycenter between 2D dimensional Gaussian projections. For the 3d distribution $\nu = \mathcal{N}(\mu, S)$, we show only an ellipse corresponding to a level line of ν. In the experiment on the left, the three projections satisfy $\sum_i \lambda_i P_i^T P_i = I_d$, but the condition is not satisfied by the four projections in the experiment on the right. Both barycenters are computed by solving the fixed-point equation (4.6) which, in the left one, gives the same result as Equation (4.4).

4.4.3 From patch distributions to image distributions

As explained in the first part of this manuscript, it is usual to assume stochastic prior models on image patches (small square image pieces), and to use such priors in a Bayesian setting for image restoration or synthesis. In most situations, these models are infered independently on all overlaping patches and do not coincide on their overlaps. In order to reconstruct a distribution on the whole image domain, we

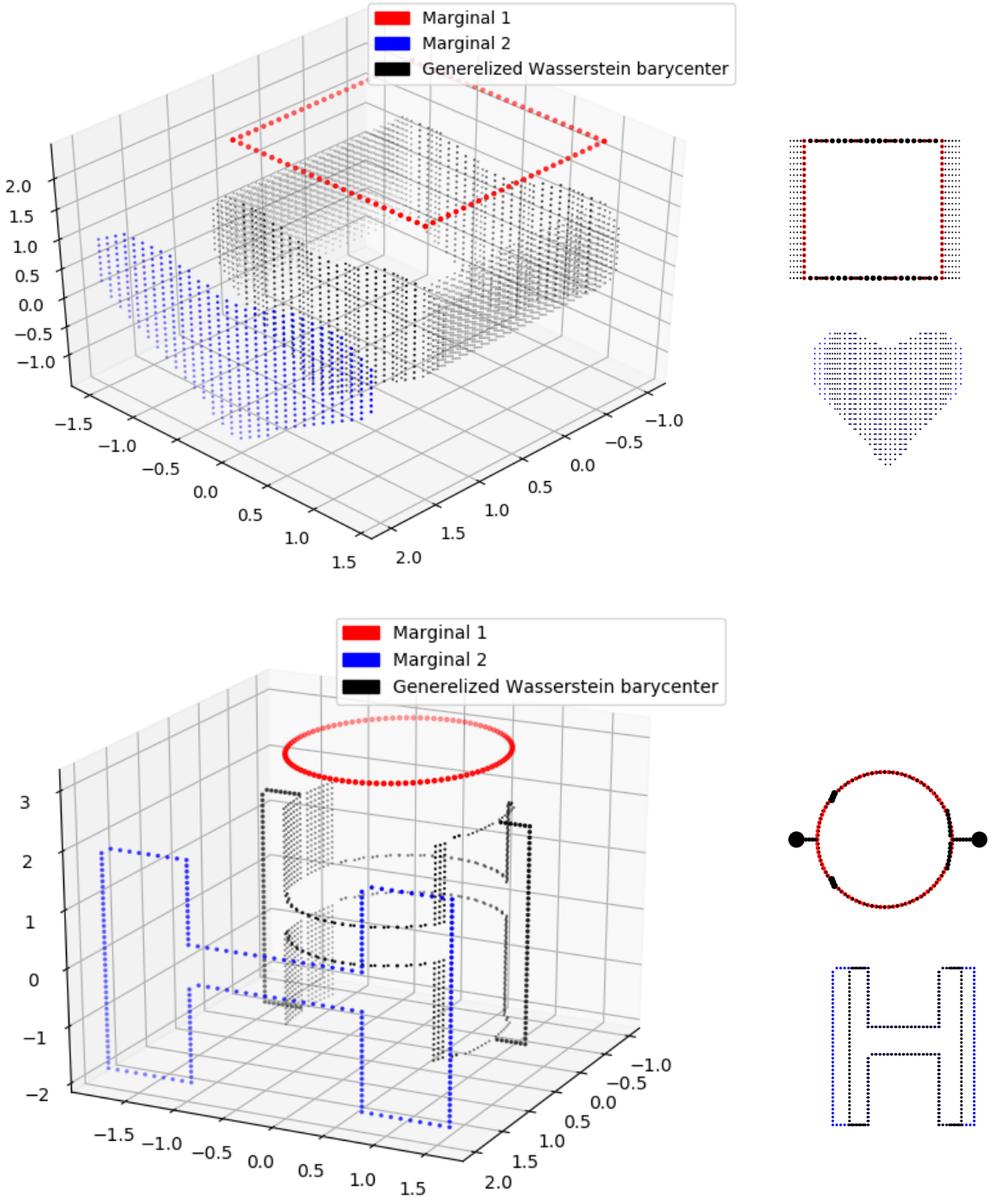


FIGURE 4.4: Generalized barycenters between disagreeing 2d distributions. Each line corresponds to an experiment. On the left, the three dimensional barycenter ν (black dots) between the original two dimensional distributions ν_i (colored dots, each color corresponding to a different i). On the right, for each i, we show the superposition of $P_i\#\nu$ (black) and ν_i.

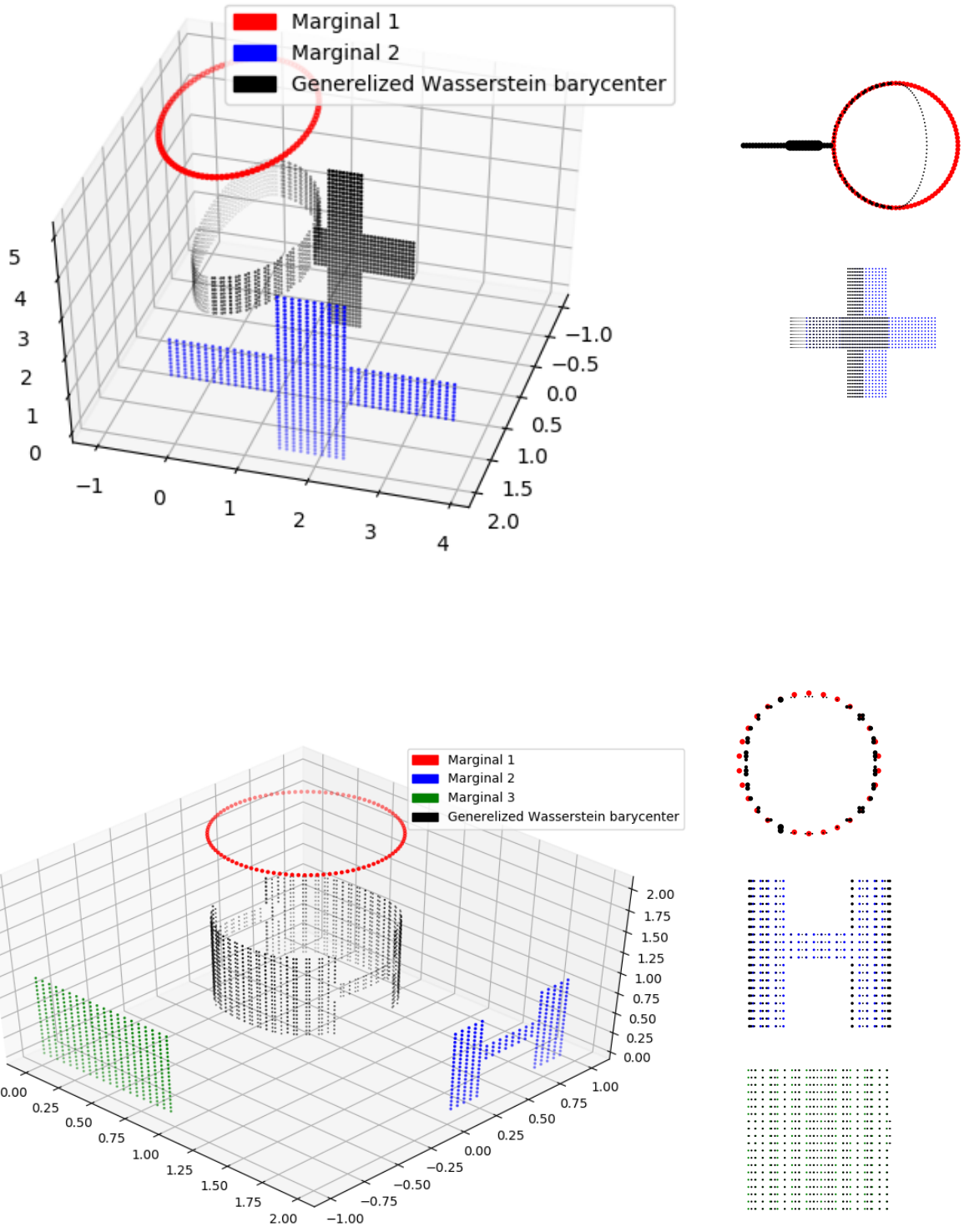


FIGURE 4.5: Generalized barycenters between disagreeing 2d distributions. Each line corresponds to an experiment. On the left, the three dimensional barycenter ν (black dots) between the original two dimensional distributions ν_i (colored dots, each color corresponding to a different i). On the right, for each i, we show the superposition of $P_i\#\nu$ (black) and ν_i.

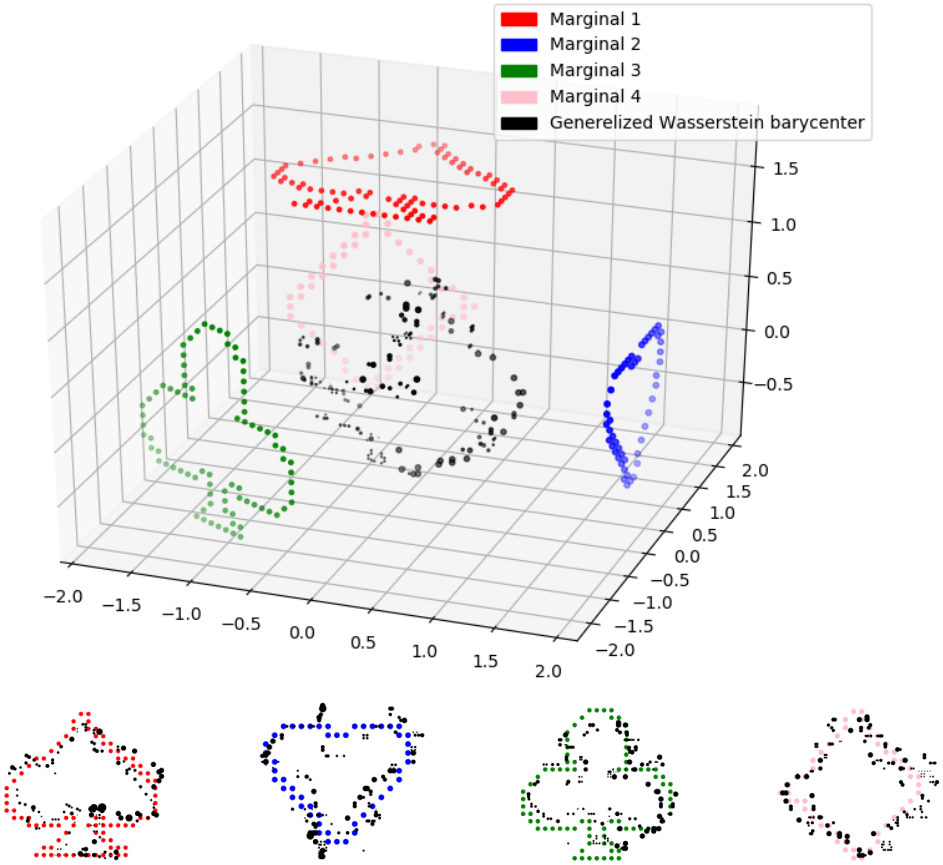


FIGURE 4.6: Exemple of 3d barycenter between 2d marginals.

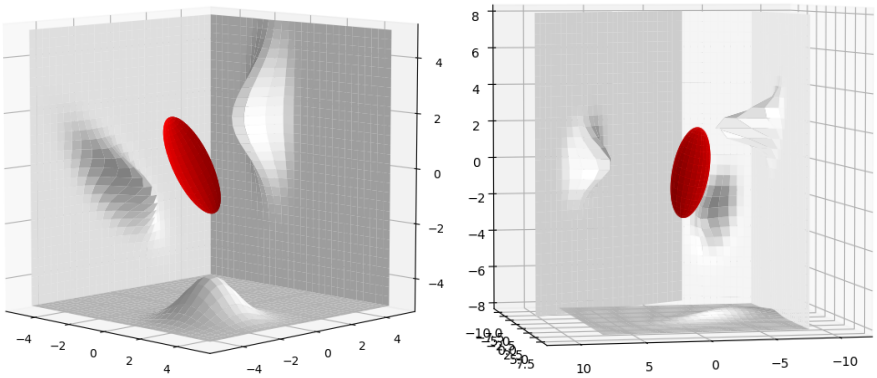


FIGURE 4.7: Exemple of generalized 3d Gaussian barycenter between 2d Gaussian distributions on different subspaces. In the experiment on the left, the three projections satisfy $\sum_i \lambda_i P_i^T P_i = I_d$, but the condition is not satisfied by the four projections in the experiment on the right. Both barycenters are computed by solving the fixed-point equation (4.6).

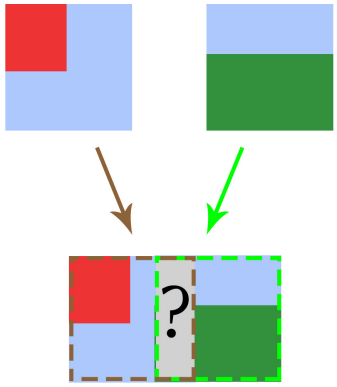


FIGURE 4.8: We wish two compute a patch distribution on the rectangle below from two patch distributions on the squares above. The rectangle is obtained here by fusioning the squares with an overlap of a few columns.

need to compromise between all of these patch distributions and we propose to use the generalized Wasserstein barycenters in this aim.

We focus here on simple experiments that can be seen as a proof of concept for this application. Assume that we have two square patches with an overlap of a few columns, as shown in Figure 4.8. Each patch i has its own Gaussian distribution ν_i, and we wish to compute the Gaussian generalized barycenter ν_{GWB} of these ν_i on the rectangle obtained by fusing the two squares on their overlap. This process shall be called the *GWB aggregation*. As a comparison, we also show the results obtained by computing the uniform average, ν_{unif} between the two distribution (the mean and covariances are averaged on their overlap) called *uniform aggregation*, and the distribution ν_{fusion} obtained by fusion between the distributions (see Chapter 2 for more details) called *fusion aggregation*.

The first experiment uses synthetic Gaussian patch models with a constant mean and covariances shown on the left of Figure 4.9. The same Figure shows the covariances computed for ν_{unif}, ν_{fusion} and ν_{GWB}. Figure 4.10 shows some samples of the same distributions.

For the second experiment, we use Gaussian models infered from real images Zoran and Weiss, 2011. Covariances and samples are shown on Figures 4.11 and 4.12.

The fusion aggregation forces the distribution to have the same value on their overlap. In Figure 4.10, it makes the right side of the patch uniform, making the horizontal border vanish, as it is the only way for the two patches to coincide. In the experiment with real models, it gives more weight to the patch with the lowest variance (the left one), leading to ignore the right one on the overlap.

The GWB aggregation and the uniform aggregation give quite similar results. It can indeed be shown that if the covariance matrices S_1 and S_2 commute, the two solutions ν_{GWB} and ν_{unif} will coincide. However, behaviour differences can still be observed. As we can see in Figures 4.9 and 4.11, the resulting covariances of the GWB aggregation has more intermediate values. This implies more smoothness and coherence in the results. In Figure 4.10 for instance, we can see that both methods divide the domain of the patch on the right into 4 blocs. As in the uniform aggregation, they all seem independent, while they tend to agree more horizontally in the GWB aggregation. This effect is however less perceptible with messier models, like the one in Figure 4.12. Yet, in this experiment, the uniform aggregation makes appear straight line that divides the patch in 3 blocs (we clearly see the distinction

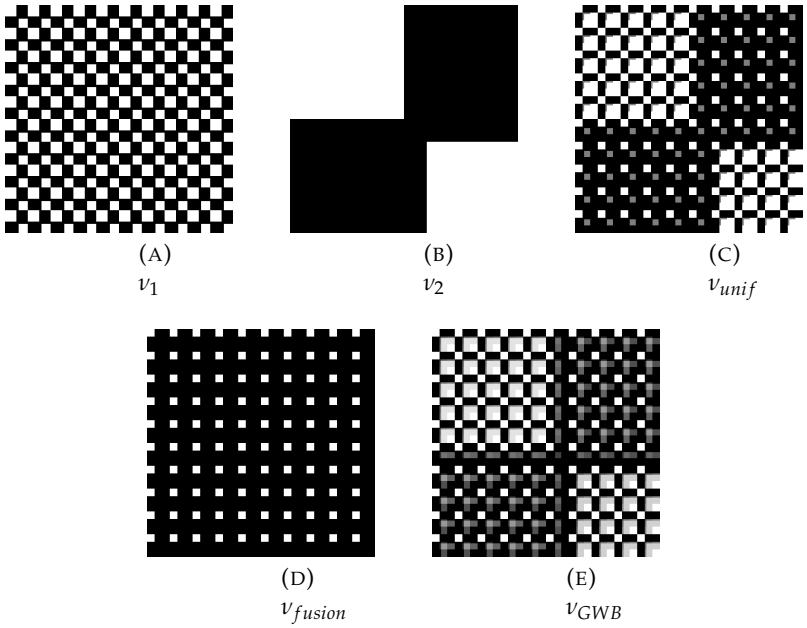


FIGURE 4.9: Covariances of the different models for the first experiment.

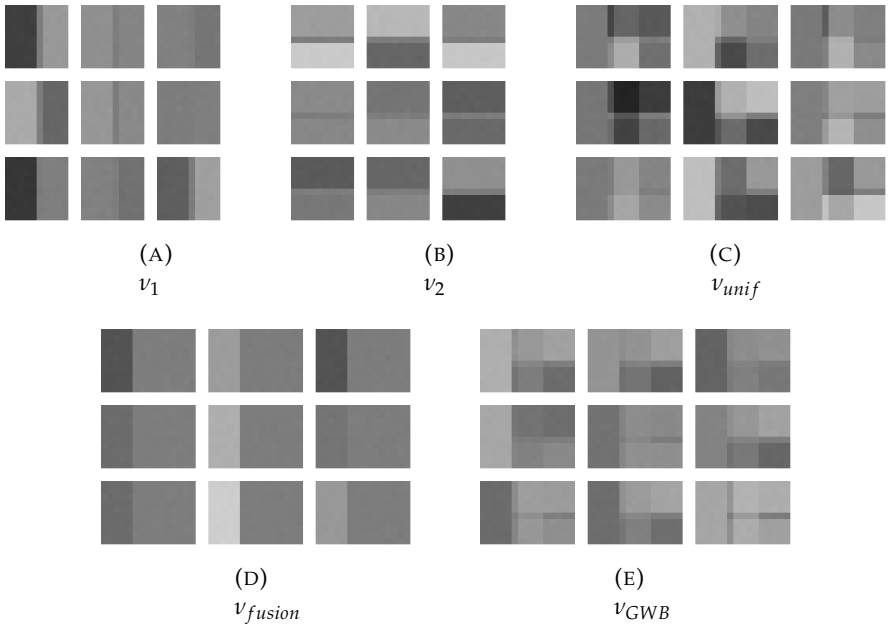
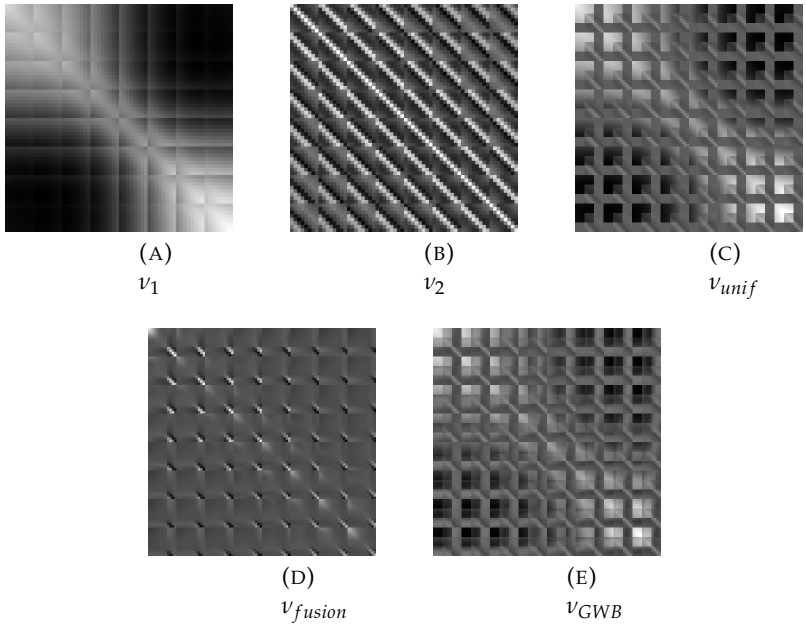


FIGURE 4.10: Set of independent samples for each distribution for the first experiment.



(A) ν_1 (B) ν_2 (C) ν_{unif}

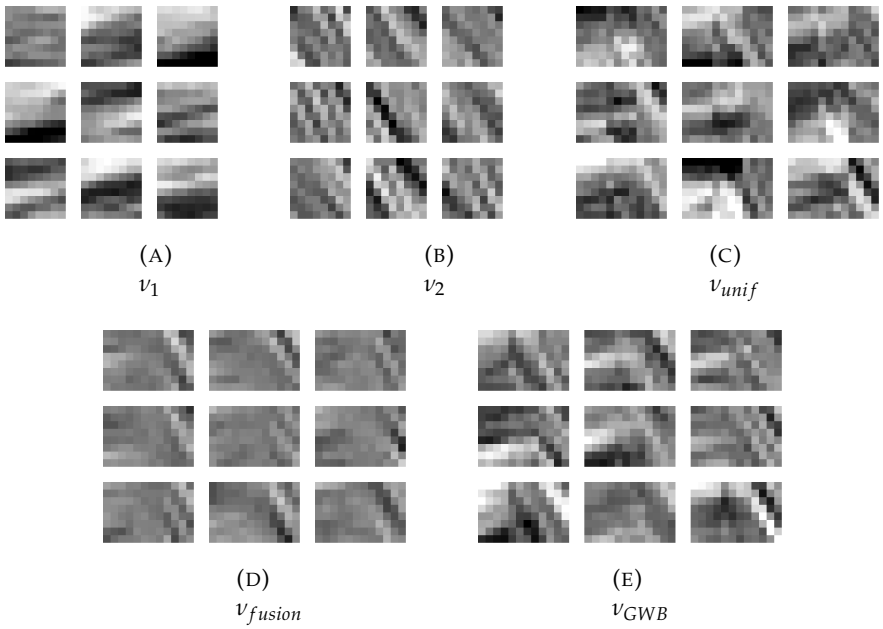
(D) ν_{fusion} (E) ν_{GWB}

FIGURE 4.11: Covariances of the different models for the second experiment.

between where the patches overlap and where they do not) that are not present in the GWB aggregation, giving a slightly nicer visual result.

4.5 Conclusion

In this chapter, we have introduced a generalization of the Wasserstein barycenters to a case where the initial probability measures live on different subspaces of $\mathbb{R}^d$. We have studied the existence of this barycenter, its dual formulation and shown how it is related to a larger multi-marginal optimal transport problem and a fixed-point equation in the Gaussian case. We applied our results on small examples related to the original motivations of this thesis. It appears that GWB behaves nicely in practice on point clouds, and gives interesting reconstruction from different marginals, but the patch application is not completely satisfying: the W_2 norm appears to be a poor choice to compare patch models. However, this study gives a basis for investigations on generalized barycenters with different norms or costs, as particular problems arise from the utilization of projections.



(A) ν_1 (B) ν_2 (C) ν_{unif}

(D) ν_{fusion} (E) ν_{GWB}

FIGURE 4.12: Set of independent samples for each distribution for the second experiment.

Conclusion and perspectives

Patch-based algorithms are an efficient solution to the image denoising problem, and more generally to image restoration. They have been widely studied in the past years and still have a lot of potential, and we believe that patch aggregation should play a big role in their future development.

We focused in this PhD on the conceptual study of patch aggregation. The first part proposed a new formalization of the aggregation step and a new way of computing it, and the second part is a direct extension of this work aiming to design new fusions with optimal transport theory.

The first proposed fusion operation has an interesting behavior but still suffers from some limitations. We intended to improve and develop the idea using quadratic optimal transport theory. However, as explained in Chapter 2, the quadratic norm is not the most appropriate norm for patch models application. Unsurprisingly, the experimental results of chapter 4 are not completely convincing. This work still represents a first step toward an aggregation theory, raising some unexpected problems, that would be interesting to explore in the future.

Wider class of restoration problems

Even if it is not fully satisfying, the proposed fusion of Chapter 2 showed some interesting behaviors. We focused on the denoising problem for the sake of simplicity, but this could be applied to a wider class of image restoration problem.

The classical linear degradation $\widehat{u} = Au + \epsilon$ does not change the setup. This extension concerns more the editing step, which similarly provides a posteriori patch models that we can merge in a similar fashion.

The strength of the fusion is its flexibility and its coherence. We think that the presented ideas could be used to merge patch models in more exotic fashions, for example when dealing with missing data and in a multi resolution framework. This would obviously depend of the considered fusion, but maybe some general ideas could emerge from these considerations and lead to interesting developments.

Development of an aggregation theory

In part II, we introduced the generalized Wasserstein problem, following the discussion of part I. As we saw, this problem has lots of similarity with the classical barycenter problem, but raises new issues and theoretical question. This could be the starting point of a hypothetical *aggregation theory* that would study the interaction of overlapping objects.

In particular, the formulation of Assumption 1 is very simple but turns out to be surprisingly tedious to prove. We managed to prove it in a particularly convenient case, but it seems that this proof can not be generalized, as the problem is completely different without the coordinate basis assumption. It appears that the problem is

deeply related to the space F (defined in Section 4.2.1). In the coordinate basis assumption, we are able to create functions of F thanks to the input functions f_i. In general, it is not possible and the problem appears to be somewhat degenerated.

In the simple case with 3 marginals and 2 variables, and with $P_1(x,y) = x$, $P_2(x,y) = y$ and $P_3(x,y) = x + y$, the space F reduces to

$$F = \{(\lambda \mathrm{I}_{X_1}, \lambda \mathrm{I}_{X_2}, -\lambda \mathrm{I}_{X_3}); \lambda \in \mathbb{R}\}$$

It means that in the conditions of Assumption 1, we have to show that

$$\forall i \in \{1,2,3\}, \exists \lambda \in \mathbb{R}; \forall x_i \in X_i = \mathbb{R}, f_i(x_i) = \lambda x_i + \widetilde{f}_i(x_i) \text{ with } ||\widetilde{f}_i|| \leq \epsilon,$$

which is already tricky and difficult to prove.

We believe that Assumption 1 holds more generally. It seems that the problem, even if it does not look difficult at first glance, might be too hard to be tackled frontally and must be formulated otherwise. We would be curious to shove in this direction.

Application to GMM

All the algorithms presented in this thesis apply to single Gaussian distributions. However, most of the models used in image restoration procedure are GMMs. We chose to select the most likely components of the mixture, which may seem reasonable (and justified by the success of NL-Bayes for instance), but we believe that this is only valid in a context of uniform aggregation. One of the strength of the patch model fusion is to keep track of the model. We believe that merging the GMMs instead of the most likely Gaussian distributions can lead to a huge improvement in performance. Indeed, it would give much more flexibility to the reconstruction and enable to give more weight to the unlikely and unprecise components which are mainly the borders. Delon and Desolneux, 2020 introduced a Wasserstein type distance restricted to GMMs, by imposing the transport plan to be a Gaussian mixture as well:

$$MW_2^2(\mu,\nu) = \inf_{\gamma \in \Pi(\mu,\nu) \cap GMM_{2d}} \int |x-y|^2 d\gamma(x,y),$$

where GMM_{2d} is the space of all Gaussian mixture models of dimension $2d$ and μ and ν belong to GMM_d. They showed that if we have $\mu = \sum_{k=1}^{K} \alpha_k \mu_k$ and $\nu = \sum_{l=1}^{L} \beta_l \nu_l$, then

$$MW_2^2(\mu,\nu) = \min_{w \in \Pi(\alpha,\beta)} \sum_{k,l} w_{k,l} W_2^2(\mu_k, \nu_k).$$

This distance could be used to perform a generalized barycenter for Gaussian mixtures, roughly by replacing W_2^2 by GWB in the previous equation. This would be more adapted to our application, since working with single Gaussian is a big limitation of patch model aggregation. However, it cannot be applied as such in the actual fusion framework, as the number of components would explode exponentially in the number of fusion. One needs to find a way to restrain the number of components of the GMMs. This is a direct path of development that I intend to explore.

Other costs

The l_2 norm is useful for its convenience and the well-understanding of its behavior. It has lots of practical and theoretical advantages, but its behavior is far to be ideal, as it for instance separates the expectation from the covariance. It was a good starting point to study the generalized Wasserstein problem, but the application that motivated it would benefit from extension to other costs, like the l_1 norm. Some results presented in Chapter 4 can be written with more general costs. For instance, the proof of the correspondence between GWB and the multimarginal formulation (Proposition 65) does not use any specificity of the l_2 norm, except for the well-definition of B. However key results like the duality relation or the existence of solutions rely on the particularity of the square norm, and we don't know how they could be generalized. This study should come along with the developments of the regular Wasserstein problem and an hypothetical aggregation theory. It would, in any case, be of major interest for the application to the aggregation step.

Bibliography

Abraham, Isabelle et al. (2017). "Tomographic reconstruction from a few views: a multi-marginal optimal transport approach". In: *Applied Mathematics & Optimization* 75.1, pp. 55–73.

Agueh, Martial and Guillaume Carlier (2011). "Barycenters in the Wasserstein space". In: *SIAM Journal on Mathematical Analysis* 43.2, pp. 904–924.

Aharon, Michal, Michael Elad, and Alfred Bruckstein (2006). "*rmk*-SVD: An algorithm for designing overcomplete dictionaries for sparse representation". In: *IEEE Transactions on signal processing* 54.11, pp. 4311–4322.

Akaike (1973). "Information Theory and an Extension of the Maximum Likelihood Principle". In:

Barnes, Connelly et al. (2009). "PatchMatch: A randomized correspondence algorithm for structural image editing". In: *ACM Trans. Graph.* 28.3, pp. 24–1.

Baudry (2009). "Selection de Modele pour la Classication Non Supervisee". PhD thesis. Universite Paris-Sud.

Baudry, Jean-Patrick, Cathy Maugis, and Bertrand Michel (2012). "Slope heuristics: overview and implementation". In: *Statistics and Computing* 22.2, pp. 455–470.

Benamou, Jean-David et al. (2015). "Iterative Bregman projections for regularized transportation problems". In: *SIAM Journal on Scientific Computing* 37.2, A1111–A1138.

Bertsekas, Dimitri P (1998). *Network optimization: continuous and discrete models*. Athena Scientific Belmont, MA.

Biernacki, Christophe, Gilles Celeux, and Gérard Govaert (2000). "Assessing a mixture model for clustering with the integrated completed likelihood". In: *IEEE transactions on pattern analysis and machine intelligence* 22.7, pp. 719–725.

— (2003). "Choosing starting values for the EM algorithm for getting the highest likelihood in multivariate Gaussian mixture models". In: *Computational Statistics & Data Analysis* 41.3, pp. 561–575.

Bishop, Christopher M (2006). *Pattern recognition and machine learning*. springer.

Bouveyron, Charles, Stéphane Girard, and Cordelia Schmid (2007). "High-dimensional data clustering". In: *Computational Statistics & Data Analysis* 52.1, pp. 502–519.

Bregman, Lev M (1967). "The relaxation method of finding the common point of convex sets and its application to the solution of problems in convex programming". In: *USSR computational mathematics and mathematical physics* 7.3, pp. 200–217.

Buades, Antoni, Bartomeu Coll, and Jean-Michel Morel (2005). "A review of image denoising algorithms, with a new one". In: *Multiscale Modeling & Simulation* 4.2, pp. 490–530.

Buades, Antoni, Marc Lebrun, and Jean-Michel Morel (2012). "Implementation of the non-local bayes image denoising algorithm". In: *Image Processing On Line*.

Carlier, Guillaume, Adam Oberman, and Edouard Oudet (2015). "Numerical methods for matching for teams and Wasserstein barycenters". In: *ESAIM: Mathematical Modelling and Numerical Analysis* 49.6, pp. 1621–1642.

Carrera, Diego et al. (2017). "Sparse overcomplete denoising: aggregation versus global optimization". In: *IEEE Signal Processing Letters* 24.10, pp. 1468–1472.

Chatterjee, Priyam and Peyman Milanfar (2011). "Patch-based near-optimal image denoising". In: *IEEE Transactions on Image Processing* 21.4, pp. 1635–1649.

Chizat, Lenaic et al. (2018). "Scaling algorithms for unbalanced optimal transport problems". In: *Mathematics of Computation* 87.314, pp. 2563–2609.

Cho, Taeg Sang et al. (2008). "The patch transform and its applications to image editing". In: *Computer Vision and Pattern Recognition, 2008. CVPR 2008. IEEE Conference on*. IEEE, pp. 1–8.

Colom, Miguel, Antoni Buades, and Jean-Michel Morel (2014). "Nonparametric noise estimation method for raw images". In: *JOSA A* 31.4, pp. 863–871.

Criminisi, Antonio, Patrick Pérez, and Kentaro Toyama (2004). "Region filling and object removal by exemplar-based image inpainting". In: *IEEE Transactions on image processing* 13.9, pp. 1200–1212.

Cuturi, Marco (2013). "Sinkhorn distances: Lightspeed computation of optimal transport". In: *Advances in neural information processing systems*, pp. 2292–2300.

Cuturi, Marco and Arnaud Doucet (2014). "Fast computation of Wasserstein barycenters". In:

Cuturi, Marco and Gabriel Peyré (2016). "A smoothed dual approach for variational Wasserstein problems". In: *SIAM Journal on Imaging Sciences* 9.1, pp. 320–343.

Dabov, Kostadin et al. (2007). "Image denoising by sparse 3-D transform-domain collaborative filtering". In: *IEEE Transactions on image processing* 16.8, pp. 2080–2095.

Danielyan, Aram, Vladimir Katkovnik, and Karen Egiazarian (2012). "BM3D frames and variational image deblurring". In: *IEEE Transactions on Image Processing* 21.4, pp. 1715–1728.

Deadman, Edvin, Nicholas J Higham, and Rui Ralha (2012). "Blocked Schur algorithms for computing the matrix square root". In: *International Workshop on Applied Parallel Computing*. Springer, pp. 171–182.

Deledalle, Charles-Alban, Vincent Duval, and Joseph Salmon (2012). "Non-local methods with shape-adaptive patches (NLM-SAP)". In: *Journal of Mathematical Imaging and Vision* 43.2, pp. 103–120.

Deledalle, Charles-Alban, Shibin Parameswaran, and Truong Q Nguyen (2018). "Image denoising with generalized Gaussian mixture model patch priors". In: *SIAM Journal on Imaging Sciences* 11.4, pp. 2568–2609.

Deledalle, Charles-Alban, Florence Tupin, and Loïc Denis (2010). "Poisson NL means: Unsupervised non local means for Poisson noise". In: *Image processing (ICIP), 2010 17th IEEE international conference on*. IEEE, pp. 801–804.

Delon, Julie and Agnès Desolneux (2020). "A wasserstein-type distance in the space of gaussian mixture models". In: *SIAM Journal on Imaging Sciences* 13.2, pp. 936–970.

Dengwen, Zhou and Shen Xiaoliu (2009). "Image denoising using weighted averaging". In: *Communications and Mobile Computing, 2009. CMC'09. WRI International Conference on*. Vol. 1. IEEE, pp. 400–403.

Dessein, Arnaud, Nicolas Papadakis, and Jean-Luc Rouas (2018). "Regularized optimal transport and the rot mover's distance". In: *The Journal of Machine Learning Research* 19.1, pp. 590–642.

Efros, Alexei A and William T Freeman (2001). "Image quilting for texture synthesis and transfer". In: *Proceedings of the 28th annual conference on Computer graphics and interactive techniques*. ACM, pp. 341–346.

Efros, Alexei A and Thomas K Leung (1999). "Texture synthesis by non-parametric sampling". In: *Computer Vision, 1999. The Proceedings of the Seventh IEEE International Conference on*. Vol. 2. IEEE, pp. 1033–1038.

Ekeland, Ivar and Roger Temam (1999). *Convex analysis and variational problems*. Vol. 28. Siam.

Elad, Michael and Michal Aharon (2006). "Image denoising via sparse and redundant representations over learned dictionaries". In: *IEEE Transactions on Image processing* 15.12, pp. 3736–3745.

Essid, Montacer and Justin Solomon (2018). "Quadratically regularized optimal transport on graphs". In: *SIAM Journal on Scientific Computing* 40.4, A1961–A1986.

Feng, Jianzhou et al. (2015). "An optimized pixel-wise weighting approach for patch-based image denoising". In: *IEEE Signal Processing Letters* 22.1, pp. 115–119.

Frigo, Oriel et al. (2016). "Split and match: Example-based adaptive patch sampling for unsupervised style transfer". In: *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 553–561.

Gangbo, Wilfrid and Andrzej Święch (1998). "Optimal maps for the multidimensional Monge-Kantorovich problem". In: *Communications on Pure and Applied Mathematics: A Journal Issued by the Courant Institute of Mathematical Sciences* 51.1, pp. 23–45.

Gepperth, Alexander and Benedikt Pfülb (2019). "Gradient-based training of Gaussian Mixture Models in High-Dimensional Spaces". In: *arXiv preprint arXiv:1912.09379*.

Guleryuz, Onur G (2007). "Weighted averaging for denoising with overcomplete dictionaries". In: *IEEE Transactions on Image Processing* 16.12, pp. 3020–3034.

He, Ji et al. (2004). "Initialization of cluster refinement algorithms: A review and comparative study". In: *2004 IEEE International Joint Conference on Neural Networks (IEEE Cat. No. 04CH37541)*. Vol. 1. IEEE, pp. 297–302.

Hoeting, Jennifer A et al. (1999). "Bayesian model averaging: a tutorial". In: *Statistical science*, pp. 382–401.

Hore, Alain and Djemel Ziou (2010). "Image quality metrics: PSNR vs. SSIM". In: *2010 20th international conference on pattern recognition*. IEEE, pp. 2366–2369.

Houdard, Antoine, Charles Bouveyron, and Julie Delon (2017). "High-Dimensional Mixture Models For Unsupervised Image Denoising (HDMI)". In:

Islam, Mohammad Tariqul et al. (2018). "Mixed Gaussian-impulse noise reduction from images using convolutional neural network". In: *Signal Processing: Image Communication* 68, pp. 26–41.

Kazi-Tani, Nabil and Didier Rullière (2019). "On a construction of multivariate distributions given some multidimensional marginals". In: *Advances in Applied Probability* 51.2, pp. 487–513.

Kellerer, Hans G (1984). "Duality theorems for marginal problems". In: *Zeitschrift für Wahrscheinlichkeitstheorie und verwandte Gebiete* 67.4, pp. 399–432.

Kervrann, Charles (2014). "PEWA: Patch-based Exponentially Weighted Aggregation for image denoising". In: *Advances in Neural Information Processing Systems*, pp. 2150–2158.

Kervrann, Charles and Jérôme Boulanger (2006). "Optimal spatial adaptation for patch-based image denoising". In: *IEEE Transactions on Image Processing* 15.10, pp. 2866–2878.

Knott, M. and C. S. Smith (1984). "On the optimal mapping of distributions". In: *Journal of Optimization Theory and Applications* 43.1, pp. 39–49. ISSN: 1573-2878. DOI: 10.1007/BF00934745. URL: https://doi.org/10.1007/BF00934745.

Kuhn, Harold W (1955). "The Hungarian method for the assignment problem". In: *Naval research logistics quarterly* 2.1-2, pp. 83–97.

Kwatra, Vivek et al. (2005). "Texture optimization for example-based synthesis". In: *ACM Transactions on Graphics (ToG)* 24.3, pp. 795–802.

Launay, Claire and Arthur Leclaire (2019). "Determinantal Patch Processes for Texture Synthesis". In:

Lebrun, Marc, Antoni Buades, and Jean-Michel Morel (2013). "A nonlocal bayesian image denoising algorithm". In: *SIAM Journal on Imaging Sciences* 6.3, pp. 1665–1688.

Lebrun, Marc et al. (2012). "Secrets of image denoising cuisine". In: *Acta Numerica* 21, pp. 475–576.

Léonard, Christian (2013). "A survey of the Schr\" odinger problem and some of its connections with optimal transport". In: *arXiv preprint arXiv:1308.0215*.

Makitalo, Markku and Alessandro Foi (2010). "Optimal inversion of the Anscombe transformation in low-count Poisson image denoising". In: *IEEE transactions on Image Processing* 20.1, pp. 99–109.

McLachlan, Geoffrey J and Thriyambakam Krishnan (2007). *The EM algorithm and extensions*. Vol. 382. John Wiley & Sons.

Nenna, Luca (2016). "Numerical methods for multi-marginal optimal transportation". PhD thesis.

Newson, Alasdair et al. (2014). "Video inpainting of complex scenes". In: *SIAM Journal on Imaging Sciences* 7.4, pp. 1993–2019.

Nishii, Ryuie (1988). "Maximum likelihood principle and model selection when the true model is unspecified". In: *Journal of Multivariate Analysis* 27.2, pp. 392–403.

Papyan, Vardan and Michael Elad (2015). "Multi-scale patch-based image restoration". In: *IEEE Transactions on image processing* 25.1, pp. 249–261.

Paulino, Ignacio Francisco Ramírez (2018). "PACO: Signal Restoration via PAtch COnsensus". In: *arXiv preprint arXiv:1808.06942*.

Peyré, Gabriel (2015). "Entropic approximation of Wasserstein gradient flows". In: *SIAM Journal on Imaging Sciences* 8.4, pp. 2323–2351.

Peyré, Gabriel, Sébastien Bougleux, and Laurent Cohen (2008). "Non-local regularization of inverse problems". In: *Computer Vision–ECCV 2008*, pp. 57–68.

Peyré, Gabriel, Marco Cuturi, et al. (2019). "Computational optimal transport". In: *Foundations and Trends® in Machine Learning* 11.5-6, pp. 355–607.

Peyré, Gabriel et al. (2019). "Quantum entropic regularization of matrix-valued optimal transport". In: *European Journal of Applied Mathematics* 30.6, pp. 1079–1102.

Pierazzo, Nicola, Jean-Michel Morel, and Gabriele Facciolo (2017). "Multi-Scale DCT Denoising". In: *Image Processing On Line* 7, pp. 288–308.

Pratelli, Aldo (2007). "On the equality between Monge's infimum and Kantorovich's minimum in optimal mass transportation". In: *Annales de l'Institut Henri Poincare (B) Probability and Statistics*. Vol. 43. 1. Elsevier, pp. 1–13.

Raad, Lara, Agnès Desolneux, and Jean-Michel Morel (2016). "A Conditional Multiscale Locally Gaussian Texture Synthesis Algorithm". In: *Journal of Mathematical Imaging and Vision* 56.2, pp. 260–279.

Rabin, Julien et al. (2011). "Wasserstein barycenter and its application to texture mixing". In: *International Conference on Scale Space and Variational Methods in Computer Vision*. Springer, pp. 435–446.

Romano, Yaniv and Michael Elad (2015). "Boosting of image denoising algorithms". In: *SIAM Journal on Imaging Sciences* 8.2, pp. 1187–1219.

Roth, Stefan, Victor Lempitsky, and Carsten Rother (2009). "Discrete-continuous optimization for optical flow estimation". In: *Statistical and Geometrical Approaches to Visual Motion Analysis*. Springer, pp. 1–22.

Rudin, Leonid I, Stanley Osher, and Emad Fatemi (1992). "Nonlinear total variation based noise removal algorithms". In: *Physica D: nonlinear phenomena* 60.1-4, pp. 259–268.

Saint-Dizier, Alexandre, Julie Delon, and Charles Bouveyron (2020). "A unified view on patch aggregation". In: *Journal of Mathematical Imaging and Vision* 62.2, pp. 149–168.

Salmon, Joseph and Yann Strozecki (2010). "From patches to pixels in non-local methods: Weighted-average reprojection". In: *Image Processing (ICIP), 2010 17th IEEE International Conference on*. IEEE, pp. 1929–1932.

Schmitzer, Bernhard (2019). "Stabilized sparse scaling algorithms for entropy regularized transport problems". In: *SIAM Journal on Scientific Computing* 41.3, A1443–A1481.

Schwarz, Gideon et al. (1978). "Estimating the dimension of a model". In: *The annals of statistics* 6.2, pp. 461–464.

Sezer, OG and Y Altunbasak (2009). "Weighted average denoising with sparse orthonormal transforms. proceedings of the IEEE International Conference on Image Processing (ICIP)". In: *Cairo, Egypt*.

Sinkhorn, Richard and Paul Knopp (1967). "Concerning nonnegative matrices and doubly stochastic matrices". In: *Pacific Journal of Mathematics* 21.2, pp. 343–348.

Solomon, Justin et al. (2015). "Convolutional wasserstein distances: Efficient optimal transportation on geometric domains". In: *ACM Transactions on Graphics (TOG)* 34.4, pp. 1–11.

Soltanayev, Shakarim and Se Young Chun (2018). "Training deep learning based denoisers without ground truth data". In: *Advances in Neural Information Processing Systems*, pp. 3257–3267.

Tabti, Sonia et al. (2014). "Modeling the distribution of patches with shift-invariance: application to SAR image restoration". In: *Image Processing (ICIP), 2014 IEEE International Conference on*. IEEE, pp. 96–100.

Talebi, Hossein, Xiang Zhu, and Peyman Milanfar (2013). "How to SAIF-ly boost denoising performance". In: *IEEE Transactions on image processing* 22.4, pp. 1470–1485.

Teodoro, Afonso M, Mariana SC Almeida, and Mário AT Figueiredo (2015). "Single-frame Image Denoising and Inpainting Using Gaussian Mixtures." In: *ICPRAM (2)*, pp. 283–288.

Ueda, Naonori and Ryohei Nakano (1998). "Deterministic annealing EM algorithm". In: *Neural networks* 11.2, pp. 271–282.

Valadier, Michel (1974). "On the Strassen theorem". In: *Analyse Convexe et Ses Applications*. Springer, pp. 203–215.

Van De Ville, Dimitri and Michel Kocher (2009). "SURE-Based Non-Local Means." In: *IEEE Signal Process. Lett.* 16.11, pp. 973–976.

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

— (2008). *Optimal transport: old and new*. Vol. 338. Springer Science & Business Media.

Wang, Yi-Qing and Jean-Michel Morel (2013). "SURE guided Gaussian mixture image denoising". In: *SIAM Journal on Imaging Sciences* 6.2, pp. 999–1034.

— (2014). "Can a single image denoising neural network handle all levels of gaussian noise?" In: *IEEE Signal Processing Letters* 21.9, pp. 1150–1153.

Wang, Zhou et al. (2004). "Image quality assessment: from error visibility to structural similarity". In: *IEEE transactions on image processing* 13.4, pp. 600–612.

Wexler, Yonatan, Eli Shechtman, and Michal Irani (2007). "Space-time completion of video". In: *IEEE Transactions on pattern analysis and machine intelligence* 29.3.

Yang, Yuhong (2005). "Can the strengths of AIC and BIC be shared? A conflict between model indentification and regression estimation". In: *Biometrika* 92.4, pp. 937–950.

Yu, Guoshen and Guillermo Sapiro (2011). "DCT image denoising: a simple and effective image denoising algorithm". In: *Image Processing On Line* 1, pp. 292–296.

Yu, Guoshen, Guillermo Sapiro, and Stéphane Mallat (2012). "Solving inverse problems with piecewise linear estimators: From Gaussian mixture models to structured sparsity". In: *IEEE Transactions on Image Processing* 21.5, pp. 2481–2499.

Zhang, Kai, Wangmeng Zuo, and Lei Zhang (2018). "FFDNet: Toward a fast and flexible solution for CNN-based image denoising". In: *IEEE Transactions on Image Processing* 27.9, pp. 4608–4622.

Zhang, Kai et al. (2017). "Beyond a gaussian denoiser: Residual learning of deep cnn for image denoising". In: *IEEE Transactions on Image Processing*.

Zoran, Daniel and Yair Weiss (2011). "From learning models of natural image patches to whole image restoration". In: *Computer Vision (ICCV), 2011 IEEE International Conference on*. IEEE, pp. 479–486.