Compressed sensing and dimensionality reduction for unsupervised learning

Anthony Bourrier

ANNÉE 2014

UNIVERSITÉ DE RENNES 1

THÈSE / UNIVERSITÉ DE RENNES 1
sous le sceau de l'Université Européenne de Bretagne

pour le grade de
DOCTEUR DE L'UNIVERSITÉ DE RENNES 1

Mention : Traitement du signal et Télécommunications
Ecole doctorale Matisse

présentée par

Anthony Bourrier

préparée à l'UMR 6074 IRISA
(Institut de Recherche en Informatique et Système Aléatoires)

Compressed sensing and dimensionality reduction for unsupervised learning

Thèse soutenue à l'IRISA
le 13 mai 2014

devant le jury composé de :

Francis BACH
Directeur de recherche à INRIA / rapporteur

Gilles BLANCHARD
Professeur à l'Université de Potsdam / rapporteur

Matthieu CORD
Professeur à l'Université Pierre et Marie Curie – Paris VI / examinateur

Bernard DELYON
Professeur à l'Université de Rennes 1 / examinateur

Erwan LE PENNEC
Professeur associé à l'École Polytechnique / examinateur

Patrick PÉREZ
Chercheur à Technicolor / examinateur

Rémi GRIBONVAL
Directeur de recherche à INRIA / directeur de thèse

Remerciements

Mes remerciements se dirigent en premier lieu vers mon directeur de thèse Rémi Gribonval et mon encadrant en entreprise Patrick Pérez. Je ne saurais exprimer pleinement ma gratitude pour leur encadrement de rare qualité. Leur pédagogie, leur acuité scientifique et leur passion communicative n'ont cessé de m'impressionner, de m'inspirer et de me donner goût à la recherche pendant ces trois années. J'ai particulièrement apprécié la capacité de Rémi à dégager avec une étonnante rapidité le sens profond des concepts alors que j'étais aveuglé par certains aspects techniques, ainsi que sa grande patience face à mes questions stupides, vagues ou mal formulées. Patrick m'a maintes fois passionné par sa culture gargantuesque des sciences numériques, et son sens du détail m'a été extrêmement profitable. Je leur adresse donc mes plus sincères remerciements pour ces trois années au cours desquelles ils m'ont tant appris, aussi bien au niveau technique que méthodologique.

Merci également à Hervé Jégou, avec qui j'ai eu le plaisir de travailler et qui m'a fait découvrir et comprendre[a] plusieurs techniques. Sa vision très pragmatique des concepts m'a inspiré. J'ai également une pensée amicale pour mes autres coauteurs Mike Davies, Tomer Peleg et Florent Perronnin.

Je remercie bien sûr les rapporteurs Francis Bach et Gilles Blanchard, et plus généralement l'ensemble du jury, pour l'intérêt qu'il porte à ces travaux et le temps consacré à l'écriture des rapports et à la soutenance.

J'ai enfin une pensée chaleureuse pour mes collègues de Technicolor et de l'IRISA[b] pour leur bonne humeur, leurs répliques hilarantes et les discussions de très haute volée[c] autour d'un café[d].

[a] Enfin, je crois...

[b] Le monde rennais étant petit, certains ont d'ailleurs endossé les deux casquettes successivement.

[c] Ou pas ?

[d] Ou autre boisson chaude pour les quelques braves qui ne succombent pas à l'appel de la caféine.

Résumé des contributions

Cette section rédigée en français reprend la motivation initiale de la thèse et en résume les contributions principales. Le manuscrit principal commence à la suite de ce résumé et est rédigé en anglais.

This section written in French reviews the initial motivation of the thesis work and summarizes the main contributions. The main part of the manuscript begins after this summary and is written in English.

Contexte du sujet

L'augmentation du volume des bases de données soulève des problématiques importantes d'analyse statistique. En effet, de nombreuses tâches nécessitent de pouvoir opérer des traitements statistiques sur des données volumineuses : parmi les exemples figurent la recherche par le contenu ou le filtrage collaboratif. Cependant, les méthodes usuelles de traitement statistique des données sont trop gourmandes en termes de mémoire et/ou de temps de calcul pour apporter une solution adéquate au problème. L'enjeu est donc le développement d'algorithmes efficaces de traitement de ces données volumineuses.

Cette économie de ressources matérielles dans le traitement des données est ainsi devenu l'objet de beaucoup d'algorithmes et de travaux théoriques au cours des dernières années. Elle est généralement effectuée en diminuant la taille des données tout en s'efforçant de pouvoir extraire une bonne approximation des quantités d'intérêt pour effectuer la tâche désirée en aval.

Parallèlement à cet engouement de la communauté statistique pour les algorithmes traitant des données volumineuses, la communauté de traitement du signal a connu l'essor d'une technique appelée *échantillonnage compressé*. Cette dénomination désigne à la base un ensemble de résultats principalement théoriques sur la reconstruction d'un signal à partir d'un certain nombre de mesures linéaires. En particulier, la plupart des résultats énoncent la possibilité théorique de reconstruire précisément toute une classe de signaux *parcimonieux* à partir d'un nombre de mesures bien inférieur à la dimension de ces signaux. De plus, cette reconstruction peut être effectuée par une minimisation convexe implémentable algorithmiquement. À la suite de ces résultats initiaux, la théorie s'est étoffée de nombreux théorèmes caractérisant des hypothèses sur les signaux ou des façons d'effectuer les mesures. Différents algorithmes de reconstruction ont vu le jour, accompagnés de bornes théoriques encadrant la précision de la reconstruction sous certaines hypothèses. Cette théorie, initialement utilisée sur les vecteurs parcimonieux, s'est également étendue à des signaux plus généraux. L'échantillonnage compressé permet ainsi de produire une représentation réduite aisément calculable de toute une classe de signaux, tout en étant capable de revenir au signal de départ via un algorithme.

En parallèle, des réductions similaires de dimension tout en gardant l'"information utile" ont été exploitées dans plusieurs outils de traitement de données à grande échelle qui opèrent sur les données en réduisant leur taille de façon analogue à ce qui est fait en échantillonnage compressé. Il semble ainsi que l'échantillonnage compressé ait un réel potentiel en apprentissage. Le point de départ de la thèse est donc l'exploration de ce potentiel, en examinant si des méthodes d'échantillonnage compressé peuvent s'appliquer avec succès à des problèmes d'apprentissage statistique.

La thèse comprend trois contributions principales, faisant chacune l'objet d'un chapitre du présent manuscrit.

Échantillonnage compressé pour l'estimation de densité

La première contribution est un apport immédiatement relié à la question évoquée précédemment : des méthodes d'échantillonnage compressé peuvent-elles s'appliquer dans un cadre d'apprentissage statistique ? En échantillonnage compressé, on considère généralement un vecteur $\mathbf{x}$ s'écrivant comme combinaison linéaire d'un nombre réduit de vecteurs pris dans une famille $\{\mathbf{u}_j\}$ (la famille étant généralement orthogonale ou composée de vecteurs quasiment orthogonaux). On cherche alors à reconstruire $\mathbf{x}$ à partir de mesures linéaires $\mathbf{Mx}$, où $\mathbf{M}$ est un opérateur linéaire réduisant la dimension.

Or, des modèles analogues de "signaux" apparaissent en apprentissage statistique : un problème classique d'apprentissage non supervisé est par exemple l'estimation d'un mélange de densités à partir d'un ensemble de vecteurs. Dans ce cadre, on peut considérer que les vecteurs sont indépendemment et identiquement distribués selon une loi de probabilité de densité p qui s'écrit comme combinaison linéaire d'un nombre réduit de densités prises dans une famille $\{p_{\boldsymbol{\theta}}\}$. Le but de la tâche d'estimation est alors de trouver lesquelles de ces densités apparaissent dans la décomposition de p et d'estimer les coefficients de cette décomposition.

La ressemblance entre ces deux cadres de travail suggère que l'on pourrait considérer le problème d'estimation de paramètres de mélange comme un problème d'échantillonnage compressé : il suffit en effet de construire un opérateur de mesure $\mathbf{M}$ qui, appliqué à une densité de probabilité p, en calcule une représentation compressée $\mathbf{M}p$. Le problème d'estimation des paramètres de mélange pourrait alors être interprété comme la reconstruction de la densité p comme combinaison linéaire des $\{p_{\boldsymbol{\theta}}\}$.

Dans notre première contribution (Bourrier et al., 2013b,c) :

- **Nous proposons un cadre de travail analogue à celui de l'échantillonnage compressé pour l'estimation de paramètres de mélange.**
- **Nous identifions des opérateurs $\mathbf{M}$ qui permettent de calculer une estimation de $\mathbf{M}p$ à partir d'un ensemble de vecteurs tirés selon p.**
- **Nous instancions ce cadre de travail sur un exemple simple, où les densités considérées sont des Gaussiennes isotropes et l'opérateur $\mathbf{M}$ est un opérateur d'échantillonnage de la transformée de Fourier. Nous prouvons qu'un choix judicieux de fraquences assure l'injectivité de l'opérateur de mesure associé sur l'ensemble des mélanges parcimonieux de Gaussiennes.**
- **Nous proposons un algorithme inspiré d'une méthode utilisée en échantillonnage compressé permettant d'estimer les paramètres de mélange à partir de la représentation compressée estimée sur les données.**

- **Nous proposons des résultats numériques d'expériences effectuées dans le cadre cité ci-dessus où les densités sont des Gaussiennes isotropes, et nous mesurons la qualité de l'estimation par rapport à un algorithme classique.**

L'estimation de paramètres de mélange à l'aide de Gaussiennes isotropes peut être interprété comme un problème classique d'apprentissage non supervisé : le partitionnement de données. Nos expériences montrent, sur un exemple simple, qu'il est possible d'estimer des paramètres de mélange de façon précise par rapport à un algorithme d'estimation classique, tout en travaillant à partir d'une représentation compressée, et donc moins volumineuse que l'ensemble des données de départ.

Cette analogie entre échantillonnage compressé et estimation de paramètres de mélange suggère la mise au point d'outils théoriques permettant d'étudier les problèmes inverses de reconstruction et d'estimation dans un cadre plus général que celui de l'échantillonnage compressé classique. Il s'agit du point de départ de notre deuxième contribution.

Performances théoriques des décodeurs en grande dimension

En échantillonnage compressé, et plus généralement dans le cadre de problèmes inverses, on cherche à reconstruire un signal $\mathbf{x}$ à partir de mesures $\mathbf{Mx}$, où $\mathbf{M}$ est un opérateur linéaire donné. En toute généralité, et particulièrement si $\mathbf{M}$ réduit la dimension, cette "inversion" ne va être possible qu'à l'aide d'un *a priori* sur le signal $\mathbf{x}$. L'hypothèse généralement faite est que $\mathbf{x}$ est approximativement parcimonieux, c'est-à-dire proche (au sens d'une certaine distance) d'un ensemble Σ_k de signaux parcimonieux (typiquement les signaux qui n'ont au maximum que k composantes non nulles dans la base canonique).

L'étape de reconstruction du signal est effectuée par un "décodeur" Δ qui va servir de pseudo-inverse à l'opérateur $\mathbf{M}$ relativement au modèle Σ_k : on veut décoder précisément les vecteurs $\mathbf{x}$ situés au voisinage de Σ_k, sans accorder d'importance à la précision du décodage si $\mathbf{x}$ est éloigné de Σ_k. Dans ce cadre, une propriété usuelle requise pour un décodeur est l'*instance optimality* : on veut majorer, pour tout signal $\mathbf{x}$, l'erreur de décodage $\|\mathbf{x} - \Delta(\mathbf{Mx})\|$ par un terme de la forme $d(\mathbf{x}, \Sigma_k)$ représentant la distance du vecteur au modèle. Ainsi, le décodage sera d'autant plus précis que le signal est proche du modèle. Des résultats étudiant l'existence de tels décodeurs ont été proposés pour les modèles de vecteurs parcimonieux et, plus récemment, pour les unions finies de sous-espaces.

Dans notre deuxième contribution (Bourrier et al., 2013a) :

- **Nous proposons un cadre général relaxant plusieurs hypothèses utilisées dans les travaux précédents sur l'*instance optimality*. Nous généralisons dans ce cadre les notions de ces travaux, ainsi que leurs relations entre elles. Les généralisations effectuées couvrent notamment le cadre où Σ est un modèle quelconque de signaux et où les mesures sont éventuellement bruitées.**

- **Nous généralisons un résultat classique d'*instance optimality* avec normes ℓ_2, initialement énoncé pour un modèle de vecteurs k-parcimonieux puis généralisé à des unions de sous-espaces, à un cadre tout à fait quelconque de modèle. Nous montrons que ce résultat s'applique à de nombreux modèles classiques utilisés en problèmes inverses.**

- **Nous proposons une généralisation de la Propriété d'Isométrie Restreinte (PIR), usuellement utilisée pour majorer l'erreur de reconstruction d'un**

vecteur en échantillonnage compressé, à des modèles généraux. Nous relions cette PIR à l'existence d'un décodeur *instance optimal* au sens d'une certaine norme que nous définissons. Nous proposons une majoration de cette norme sous une hypothèse de PIR. Cette majoration est étudiée dans deux cas classiques, pour lesquels nous prouvons qu'elle est équivalente à des normes usuelles.

Ce deuxième groupe de contributions fournit ainsi des outils permettant d'étudier la précision théorique que l'on peut attendre en terme d'*instance optimality* sur un problème donné. Le cadre développé englobe notamment le cas où ce que l'on cherche à reconstruire n'est pas forcément le signal de départ mais une caractéristique du signal $\mathbf{Ax}$, où $\mathbf{A}$ est un opérateur linéaire. L'étude du cas où ce $\mathbf{A}$ est non trivial (c'est-à-dire différent de l'identité) soulève de nombreuses questions ouvertes, englobant l'apprentissage sur données compressées.

L'apprentissage non supervisé sur données compressées a déjà été évoqué dans la première contribution, dans laquelle était étudiée une méthode d'estimation de paramètres de mélange sur une base de données de vecteurs compressée globalement. Un autre moyen, plus usuel, de compresser des vecteurs est de réduire individuellement leur dimension, tout en conservant leur nombre. La dernière partie du travail traite de telles méthodes dans le cadre de la recherche de plus proche voisin, problème courant en analyse statistique.

Plongements explicites pour la recherche de plus proche voisin

En recherche de plus proche voisin, on considère un espace muni d'une distance d et un ensemble $\mathcal{X} = \{\mathbf{x}_1, \ldots, \mathbf{x}_L\}$ d'éléments de cet espace. Étant donné un élément-requête $\mathbf{q}$, la recherche de plus proche voisin de $\mathbf{q}$ consiste à trouver $\operatorname{argmin}_{\mathbf{x}\in\mathcal{X}} d(\mathbf{x}, \mathbf{q})$. Si L est grand et/ou si d est une distance dont le calcul est coûteux, le recours à une recherche approximative est obligatoire. De tels schémas approximatifs de recherche sont nombreux lorsque d est la distance Euclidienne usuelle, mais sont plus rares si d sort de ce cadre. Or, des distances fréquemment utilisées pour les données textuelles, sonores ou bien visuelles, sont des distances définies à partir d'un noyau k, qui ne sont en général pas des distances Euclidiennes dans l'espace de départ.

Pour de telles distances, des méthodes approximatives de recherche de plus proche voisin ont été dérivées, par analogie avec le cas Euclidien. La majorité s'appuie sur des méthodes de hachage, ou plus généralement d'indexation : chaque vecteur $\mathbf{x}_i$ est compressé en une représentation $s(\mathbf{x}_i)$, bien moins volumineuse que celle de départ. Cette compression est typiquement faite de façon à conserver les voisinages, et donc faire en sorte que deux voisins "proches" aient des images par s similaires tout en s'assurant que des voisins "éloignés" aient des images différentes. La recherche de plus proche voisin se fait alors de manière moins coûteuse dans l'espace compressé : la requête $\mathbf{q}$ est également réduite en une représentation $s(\mathbf{q})$, que l'on compare ensuite aux quantités $s(\mathbf{x}_i)$. La recherche de plus proche voisin se fait donc dans le domaine compressé des *signatures* $s(\mathbf{x}_i)$, potentiellement de façon bien plus économe en terme de mémoire et de temps de calcul : d'une part les représentations utilisées sont plus compactes que celles de départ, d'autre part la distance entre signatures est généralement plus efficace à calculer que la distance de départ. Cette recherche approximative permet d'identifier un certain nombre de voisins proches potentiels. La recherche se termine usuellement par un réordonnancement de ces voisins approchés.

En particulier, plusieurs méthodes de hachage binaire visant à conserver les voisinages au sens d'une distance dérivée d'un noyau ont été proposées.

Dans notre troisième contribution (Bourrier et al., 2012) :

- **Nous proposons, pour la recherche de plus proche voisin au sens d'une distance dérivant d'un noyau, de s'appuyer sur une approximation Euclidienne du noyau, typiquement appelée *plongement explicite*. De telles méthodes sont éprouvées, notamment l'Analyse en Composantes Principales (ACP) à noyaux, qui a fait l'objet de travaux à la fois théoriques et pratiques sur la précision et la réduction de la complexité de calcul de l'approximation.**

- **L'approximation via l'ACP à noyaux nous permet de dériver un schéma de recherche exacte de plus proche voisin pour les noyaux normalisés, s'appuyant sur l'inégalité de Cauchy-Schwarz bornant la précision de l'approximation. Nous proposons des résultats montrant le gain de cette approche dans un cas particulier, et ses limites en général. Nous évoquons une relaxation possible de l'inégalité permettant d'apprendre des bornes théoriques de précision de l'approximation en général.**

- **Suite à cette étape de plongement explicite, il est également possible d'appliquer, si l'on veut effectuer une recherche approximative, un schéma de compression adapté à la distance Euclidienne. En particulier, nous montrons à l'aide d'expériences que la combinaison de l'ACP à noyaux et d'un hachage Euclidien offre de meilleurs résultats que des méthodes usuelles de hachage à noyaux. Il est en outre possible, à l'aide de la méthode proposée, d'appliquer suite à l'étape de plongement explicite un schéma de compression offrant une meilleure précision que le hachage binaire : la Quantification Produit.**

Utiliser un plongement explicite afin de se ramener au cas Euclidien semble donc être une étape utile, permettant notamment de dériver des schémas de recherche approximative plus précis que les méthodes de hachage existantes. Les algorithmes de recherche de plus proche voisin devraient ainsi se comparer à cette succession d'étapes simples afin de mesurer le réel apport de l'algorithme en question.

Contents

Notations

Mathematical notations

$\mathbf{x}$	Vector (bold lowercase letter)
$\mathbf{A}$	Linear operator (bold uppercase letter)
$\mathcal{M}$	Vector or operator family (curved uppercase letter)
$L^p(\mathbb{R}^n)$	Lebesgue p-integrable complex functions on $\mathbb{R}^n$
$\|\cdot\|_p$	ℓ_p-norm in $\mathbb{R}^n$ or $L^p(\mathbb{R}^n)$
$\langle\cdot,\cdot\rangle$	Scalar product in a pre-Hilbert space
$\mathcal{F}(f).\boldsymbol{\omega}$	Fourier transform of f taken at frequency $\boldsymbol{\omega}$
$\mathcal{M}_{m,n}(\mathbb{K})$	Matrices of size $m \times n$ over the field $\mathbb{K}$
$\mathbf{I}$	Identity matrix
∇f	Gradient of a functional f
$[\![\cdot,\cdot]\!]$	Integer interval
$\mathbb{E}[\cdot]$	Expected value of a random variable
$\mathbb{P}(\cdot)$	Probability of an event

Manuscript conventions

n	Signal or data dimension
m	Compressed dimension / Number of measurements
k	Sparsity level
L	Number of elements in a database
$\mathbf{M}$	Measurement operator
Σ_k	Set of k-sparse vectors
Σ	General model of signals
E	Signal space
F	Measurement space
G	Feature space

Abbreviations

CS	Compressed Sensing
GMM	Gaussian Mixture Model
i.i.d.	independently and identically distributed
IHT	Iterative Hard Thresholding
IOP	Instance Optimality Property
KL	Kullback-Leibler
KLSH	Kernelized Locality Sensitive Hashing
KPCA	Kernel Principal Component Analysis
LSH	Locality Sensitive Hashing
NSP	Null Space Property
PCA	Principal Component Analysis
RIP	Restricted Isometry Property
RMMH	Random Maximum Margin Hashing
SVM	Support Vector Machine

Introduction

For more than a decade, numerical sciences have known a fast evolution, reflecting the growing need for data processing tools. Practical interests for such tools are numerous: some examples of general tasks where statistical learning on data is critical are content-based search, automatic classification of data, or collaborative filtering. This type of general tasks is addressed by many software or online services.

The gain in interest for such tools in the past years is tied to several phenomena, which led to a large increase in the size of available data, especially concerning the Internet. This increase in the number of sources and in the volume of content contributed to the elaboration of databases comprised of billions of elements[e], and on which traditional statistical analysis tools cannot apply, principally because of hardware constraints, implying the computers or servers cannot cope with memory or computation time requirements of these algorithms on such voluminous data.

Particular statistical processing methods of such data have therefore arisen, principally consisting in a volume reduction of the data and/or in an approximative but faster way of dealing with computations. At the theoretical level, the compromise between computational time, precision of the statistical analysis and data size has also been studied.

In parallel to this evolution in data processing, has been developed in the signal processing community a theory of *compressed sensing*, a technique ensuring that under certain hypotheses on a signal, it is possible to reconstruct it precisely from a number of measures which is *a priori* far smaller than the dimension of the signal. Compressed sensing theory has developed with both theoretical results controlling the necessary and sufficient hypotheses on the signal for such a reconstruction to be possible, and practical algorithms to perform such a reconstruction, as well as additional results on the precision of the reconstruction provided by these algorithms.

Interestingly, analogous ideas to those used in compressed sensing have been successfully applied for statistical analysis of voluminous data. The goal was principally to reduce the size of the considered data (similarly to the measurement step in compressed sensing), thus reducing the memory and/or computational time required for an algorithm to perform, while still being able to approximately extract the desired information (similarly to the reconstruction step). Therefore, compressed sensing seems to have a genuine potential for statistical analysis of voluminous data.

The starting point of the Ph.D. thesis presented in this manuscript is therefore the study of the potential of compressed sensing for learning on data. More precisely, we will consider *unsupervised* learning, in which the considered data is not divided into categories before processing.

The document layout is the following: Chapter 1 provides a more thorough description of the notions and ideas at the source of this work. In particular, it draws links between compressed sensing and some techniques which have been proposed in the learning com-

[e]Databases built from Facebook or Flickr image data can comprise 10^{11} elements.

munity. Then come three chapters, each one presenting a main contribution of the thesis. These chapters all begin by a quick review of the state of the art relative to the corresponding contribution. Then come a summary of the contributions relative to the corresponding chapter. Finally, the contributions are described.

Chapter 2 contains the first contribution: a parameter estimation method for probability density mixtures on compressed data, in a similar way as what is done in compressed sensing, and which can be interpreted as a generalized compressed sensing problem. We propose to compress data to a fixed-size representation called a *sketch*, then derive an algorithm to induce mixture parameters from this sketch. We provide numerical experiments evaluating the experimental performance of this compressed framework in the case of isotropic Gaussian mixture estimation.

Chapter 3 describes the second contribution : a generalization of results relative to the theoretical performance of reconstruction methods in compressed sensing, and more generally in *inverse problems*, to models beyond the models to which the previous results apply. More particularly, we provide necessary and sufficient conditions to the existence of an *instance optimal* reconstruction function, which is optimal in a certain sense with respect to the model. We study this instance optimality property in a general case and link it with a generalized Restricted Isometry Property, which is a widely-used property in compressed sensing.

Finally, Chapter 4 presents the third main contribution: an approximate nearest neighbor search of a certain element with respect to certain distances, relying on the approximation of these distances by a Euclidean distance. We provide a simple framework combining two existing dimension-reducing techniques: *explicit embedding* and *Product Quantization*. We provide experiments showing that these frameworks allow a better precision than hashing techniques designed to work with kernels.

Chapter 1

Motivation

We introduce in this section the underlying motivation to the thesis work depicted in the present manuscript. The initial idea is to make a parallel between two *a priori* different domains: on the one hand, inverse problems, and more particularly the recent theory of compressed sensing, which are typically used in signal processing; on the other hand, somewhat general data analysis methods which exploit ideas similar to the paradigms of compressed sensing.

We will first, in Section 1.1, give a review of inverse problems with a particular focus on compressed sensing, describing the ideas at the core of this concept. Then we will talk, in Section 1.2, about some data processing methods related to statistical learning and relying on analogous schemes to those used in compressed sensing. This will suggest links between compressed sensing and learning, which will be developed in our first contribution in Chapter 2.

1.1 Inverse problems and Compressed sensing

1.1.1 Sparse signal representations

The linear abstraction of signal processing. Physical signals typically obey linear partial differential equations. Thanks to this linear behavior, the abstract linear algebra framework is particularly well-suited for signal processing. In this sense, a signal is typically represented:

- either as a function defined on an infinite domain (typically spatial or temporal), and belonging to a space of functions sharing a certain regularity property;
- or more frequently as a quantized signal, *i.e.*, a finite-dimensional vector of $\mathbb{R}^n$ or $\mathbb{C}^n$, where each entry represents the value of a quantization cell (in time for audio signals or in space for images for instance).

Many transformations undergone by signals can be modeled by linear operators, such as convolution operators. This allows the usage of powerful mathematical tools to model signal data and derive algorithms to process it.

Sparsity of a signal. The "canonical" representation of a finite-dimensional signal as a vector, such as the representation of an image as a concatenation of pixel values, is not necessarily well-suited for processing. An interesting property on the representation of a signal in a certain basis is *sparsity* (Elad, 2010; Mallat, 2008). When linearly decomposing

a signal $\mathbf{x}$ on a certain basis $\mathcal{B} = \{\mathbf{b}_1, \dots, \mathbf{b}_n\}$, we generically say that its representation is sparse if the decomposition involves only a few nonzero coefficients, that is $\mathbf{x} = \sum_{i=1}^{n} x_i \mathbf{b}_i$ with $x_i = 0$ for most subscripts i.

In this case, the basis $\mathcal{B}$ is well-suited to represent $\mathbf{x}$: it provides a compact representation (the set of couples $\{(i, x_i) : x_i \neq 0\}$ fully describes the signal), and can give a simple interpretation by describing the signal with only a few components. This sparse representation of the signal $\mathbf{x}$ also allows quick linear computations, and the *a priori* knowledge that $\mathbf{x}$ is sparse when decomposed on $\mathcal{B}$ allows one to solve linear inverse problems, as will be discussed in the following.

Classical sparsity-inducing bases. In signal processing, given a specific task, one usually considers a certain model of signals, which can contain very specific signals or on the contrary a wide range of them. To efficiently exploit sparsity, a desirable property is that there exists a basis $\mathcal{B}$ which provides an approximately sparse decomposition for every vector of the considered model. Such a basis $\mathcal{B}$ allows one to precisely approximate all (or most of) the signals in the considered model by sparse vectors.

Such classical "sparsity-inducing" bases used in signal processing include many trigonometric-based families, initially proposed by Fourier at the beginning of the XIX$^{\text{th}}$ century for periodic functions (Fourier, 1808). Indeed, natural signals often present a form of periodicity associated to particular frequencies, which will be the frequencies around which the energy of the Fourier decomposition will be concentrated. Over the years, this type of frequency decomposition has been applied to other linear objects like nonperiodic functions or discrete vectors. Such trigonometric families are typically denoted as Fourier, or frequential, bases.

The main drawback of Fourier representations is the so-called Gibbs phenomenon, which designates their difficulty to adequately represent a discontinuity in the signal. To circumvent this difficulty, another family of functions is widely used as a representation basis for signals: the *wavelet functions* (Mallat, 2008), which typically are piecewise regular functions with localized discontinuities or strong variations. With such wavelet bases, the decomposition of a large class of natural signals, such as natural images, is concentrated on a small subset of vectors in the family, while the rest of the decomposition involves small coefficients.

Both Fourier and wavelet families, which include many different variants, are usually well-structured sets of vectors, which allows one to derive efficient algorithms to compute the corresponding decomposition of a vector.

More recently, other sparsity-inducing families of vectors have been considered and are denoted as *dictionaries*. They are typically families of vectors learned from a data set to adaptively provide a sparse decomposition on data which is similar to the training set. Unlike Fourier and wavelet families, orthogonality or even linear independence is not usually required, hence their usual denomination of *redundant* dictionaries. This type of representation, while being potentially much better suited for specific data than a nonadpative family of vectors, suffers from harder learning and decomposition steps compared to the previously mentioned representations, which is mainly due to the less-structured form of the family.

1.1.2 Inverse problems

The concision and interpretability given by sparse representations make them useful for many signal processing tasks. A common considered framework in signal processing is the

case where a vector $\mathbf{s}^* \in \mathbb{R}^n$ undergoes the action of a linear operator $\mathbf{M}$, with a possible additional noise $\mathbf{e}$. This results in a vector $\mathbf{M}\mathbf{s}^* + \mathbf{e}$, from which one would want to recover $\mathbf{s}^*$. This framework applies to multiple problems, among which appear denoising, deconvolution or super-resolution[a].

In general, the operator $\mathbf{M}$ will not be invertible, so that even in the absence of noise, the problem of recovering $\mathbf{s}^*$ from the quantity $\mathbf{y} = \mathbf{M}\mathbf{s}^* + \mathbf{e}$ is ill-posed: several vectors can indeed have the same image by $\mathbf{M}$, impeding from recovering the correct preimage. To correctly define the problem, one must impose additional constraints on $\mathbf{s}^*$. One of the typical considered constraints is that $\mathbf{s}^*$ has a sparse linear representation in a certain basis, or more generally a certain dictionary $\mathcal{D}$. This assumption implies that the vector $\mathbf{s}^*$ can be written as $\mathbf{D}\mathbf{x}^*$, where $\mathbf{D}$ is a matrix containing the vectors of $\mathcal{D}$ as columns, and $\mathbf{x}^*$ is a sparse vector containing an unknown but "small" number of nonzero entries.

In this case, the problem of "inverting" the observation $\mathbf{y} = \mathbf{M}\mathbf{D}\mathbf{x}^* + \mathbf{e}$ to find $\mathbf{x}^*$ can be expressed as the search for the sparsest vector $\mathbf{x}$ such that $\mathbf{M}\mathbf{D}\mathbf{x}$ is close to the observation $\mathbf{y}$ in a certain sense. More precisely, the problem can be formulated as:

$$\underset{\mathbf{x}}{\operatorname{argmin}} \|\mathbf{x}\|_0, \text{ subject to } \|\mathbf{y} - \mathbf{M}\mathbf{D}\mathbf{x}\|_2 \leq \sigma, \tag{1.1}$$

where $\|\mathbf{x}\|_0$ is the pseudo-norm counting the number of nonzero entries of $\mathbf{x}$, and σ is the estimated order of magnitude of the noise $\mathbf{e}$, which is typically taken in the ℓ_2-norm sense.

Note that there may be multiple solutions to the problem (1.1), so that it may be considered as ill-posed. However, if $\sigma = 0$, there exist sufficient conditions so that the solution is unique. In this case, the problem is expressed as the search for the sparsest vector of the affine space $\{\mathbf{x} \in \mathbb{R}^n : \mathbf{M}\mathbf{D}\mathbf{x} = \mathbf{y}\}$. If $\mathbf{x}^*$ is sparse enough[b], then the solution to (1.1) is unique (Chapter 2 of (Elad, 2010)).

If $\sigma > 0$, uniqueness is never achieved (unless in the trivial case where $\mathbf{x}^* = 0$): indeed, if $\mathbf{x}$ is a solution of (1.1), the continuity of the operator associated to $\mathbf{M}\mathbf{D}$ ensures that any vector sufficiently close to $\mathbf{x}$ is also solution. Since there exists an infinite number of vectors with the same sparsity as $\mathbf{x}$ and arbitrarily close to $\mathbf{x}$, there is an infinite number of solutions to (1.1). Therefore, in this case where $\sigma > 0$, one simply aims at finding *one* of the many solutions of (1.1) (Chapter 5 of (Elad, 2010)).

However, even when the problem (1.1) has a unique solution, it is in general NP-hard to solve it (Natarajan, 1995; Davis et al., 1997), therefore the problem is asymptotically intractable. Two main approaches can be considered to try and solve this problem approximately:

- *Greedy approaches*: Since one searches for a sparse vector $\mathbf{x}$ such that $\mathbf{y} \approx \mathbf{M}\mathbf{D}\mathbf{x}$, $\mathbf{y}$ will be similar to a linear combination of a few columns of $\mathbf{M}\mathbf{D}$. Therefore, one can find columns of this matrix which are well-correlated with $\mathbf{y}$, so that there is a linear combination of these columns which approximately yields $\mathbf{y}$, hopefully leading to a good solution of the problem. The research for such columns can be done in one step by finding the most correlated columns (Gribonval et al., 2007), or in an iterative fashion with a *pursuit* algorithm (Mallat and Zhang, 1993; Tropp, 2004) which updates at each step a sparse approximate solution by adding a nonzero entry corresponding to a column of $\mathbf{M}\mathbf{D}$ which is well-correlated with the residual.

[a] We consider real vectors for simplicity, but all considered results can easily be applied to complex vectors.

[b] Namely, if $\|\mathbf{x}^*\|_0$ is less than half the *spark* of the matrix $\mathbf{M}\mathbf{D}$, the spark being the minimal number of linearly dependent columns of a matrix.

Even if these approaches can yield an exact solution of the problem (1.1) in particular cases, the conditions are usually very restrictive and not satisfied in practice (Chapter 4 of (Elad, 2010)). Nonetheless, these greedy algorithms can still perform well in practice and yield a good approximate solution (Chapter 3 of (Elad, 2010)).

- ℓ_1 *relaxation*: One can replace in (1.1) the term $\|\mathbf{x}\|_0$ by the convex term $\|\mathbf{x}\|_1$. The problem then becomes

$$\mathbf{x}^* = \underset{\mathbf{x}}{\operatorname{argmin}} \|\mathbf{x}\|_1, \text{ subject to } \|\mathbf{y} - \mathbf{MDx}\|_2 \leq \sigma. \tag{1.2}$$

This formulation is typically known as Basis Pursuit (Chen et al., 1998) and can be solved using standard optimization techniques by casting it as a linear program (Chapters 13 and 14 of (Nocedal and Wright, 2006)) or a second order cone program (Chapter 4 of (Boyd and Vandenberghe, 2004)). They can typically be solved using interior point methods. When $\sigma = 0$, under stronger assumptions on $\mathbf{M}$, $\mathbf{D}$ and $\|\mathbf{x}^*\|_0$ than in the $\|\cdot\|_0$ formulation, it has been proven that the solution to the problem (1.2) is unique and the same as problem (1.1) (Donoho and Huo, 2001; Donoho, 2004). When $\sigma > 0$, there also exist robustness results upper bounding the discrepancy between the solutions of (1.1) and (1.2) (Donoho et al., 2006).

Both types of methods therefore provide feasible alternatives to the intractable problem (1.1), as well as theoretical results controlling the precision of the approximate solutions with respect to the solutions of the initial problem under certain conditions.

We have introduced inverse problems in the case where the operator $\mathbf{M}$ is fixed and one aims at reconstructing a signal from its (possibly corrupted) measure by $\mathbf{M}$. For such inverse problems, one looks for an adequate model and/or an adequate optimization formulation in order to perform the inversion. The next section focuses on another way of considering inverse problems, leading to the theory of *compressed sensing*.

1.1.3 Compressed sensing paradigms

Motivation for compressed sensing. Typically, the acquisition of a signal is performed by sampling it at regularly spaced points. In this framework, the Nyquist-Shannon theorem is a well-known result precising the number of regular measurements one needs to be able to theoretically recover the signal. It states more precisely that if a function f has a low-frequency spectrum, that is the Fourier transform of f is contained in a segment $[-\omega, \omega]$, then f can be (theoretically) perfectly recovered from a discrete regular sampling of period $\frac{\pi}{\omega}$. In this case, the reconstruction process is a simple convolution step with a sinc function.

In some practical acquisition cases of a signal, there are benefits to reducing the number of measurements, even if this causes the reconstruction step to be more costly. For instance, in medical imaging, the number of measurements should be minimal to avoid a negative impact on the patient. Another example where a reduced number of measurements would be beneficial is when sensing a signal using a device with limited energy, such as a mobile device or a satellite.

This objective of reducing the number of measurements allows one to look at linear inverse problems under another angle. One now considers a given model $\Sigma \subset \mathbb{R}^n$, which is a set of signals of interest. Given these particular signals, one aims at finding linear operators $\mathbf{M} : \mathbb{R}^n \to \mathbb{R}^m$, with m "small" (hopefully way smaller than n) and such that for any $\mathbf{x} \in \Sigma$, one can recover $\mathbf{x}$ from measurements $\mathbf{Mx}$ with an algorithmic procedure

(typically numerical optimization). As well as in the usual inverse problem discussed previously, one can also study a noisy case where the measurements are corrupted with a noise $\mathbf{e}$, and where one aims at approximately recovering $\mathbf{x}$ from measurements $\mathbf{Mx} + \mathbf{e}$.

The somewhat recent but already well developed theory of *compressed sensing* (Kutyniok, 2012; Qaisar et al., 2013) gives interesting and surprising answers to this measurement reduction objective.

The initial compressed sensing setup. Early works in the compressed sensing theory (Donoho, 2006; Candès and Tao, 2006) mainly consider the noiseless recovery of sparse vectors in the canonical basis of $\mathbb{R}^n$. Let's denote by Σ_k the set containing the k-sparse vectors:

$$\Sigma_k = \{\mathbf{x} \in \mathbb{R}^n : \|\mathbf{x}\|_0 \leq k\}. \tag{1.3}$$

In this compressed sensing framework, the goal is to find operators $\mathbf{M}$ satisfying several conditions:

- *Dimension reduction:* The operators must map $\mathbb{R}^n$ in a low-dimensional space $\mathbb{R}^m$, that is $m < n$, and hopefully $m \ll n$.
- *Theoretical recovery of Σ_k:* The operators must yield *theoretical* noiseless recovery of all vectors in Σ_k. As we have seen in the previous section, the typical linear inverse problem way of casting the recovery of a signal $\mathbf{x}^* \in \Sigma_k$ from $\mathbf{y} = \mathbf{Mx}^*$ is as

  $$\underset{\mathbf{x}\in\mathbb{R}^n}{\operatorname{argmin}} \|\mathbf{x}\|_0, \text{ subject to } \mathbf{Mx} = \mathbf{y}, \tag{1.4}$$

  which is the problem (1.1) reformulated in the case where $\mathbf{D}$ is the identity matrix of dimension n and $\sigma = 0$. The problem (1.4) should have $\mathbf{x}^*$ as a unique solution for the theoretical recovery condition to be satisfied.
- *Practical recovery of Σ_k:* The operators must yield *practical* noiseless recovery of all vectors in Σ_k. We have seen that the problem (1.1) is practically intractable in general. Therefore, an additional practical recovery condition is that this problem can be recast as an equivalent problem, which is easier to solve and still yields the same solution. This equivalent problem is typically the ℓ_1 relaxation

  $$\underset{\mathbf{x}\in\mathbb{R}^n}{\operatorname{argmin}} \|\mathbf{x}\|_1, \text{ subject to } \mathbf{Mx} = \mathbf{y}. \tag{1.5}$$

An optional extra constraint is to enforce that problems (1.4) and (1.5) yield precise reconstruction (in a certain sense) of approximately sparse vectors $\mathbf{x}^*$, that is vectors which are close to Σ_k in a certain sense. This additional constraint comes from the fact that practical signals are not usually exactly sparse but rather have fast decaying coefficients in specific basis: they are therefore approximately sparse and not exactly sparse, and the "precise recovery" property should encompass these signals.

Early results of compressed sensing. Initial results of compressed sensing have been stated in a quick succession of papers from 2004 to 2006, and many theoretical results have kept on being found ever since. In the next few sections, we propose to summarily unknot the braid of this dense succession of fundamental papers, presenting the main flavor of the results in the approximate order they were proposed.

All the initial papers about compressed sensing mainly give examples of linear operators $\mathbf{M}$ satisfying the conditions mentioned in the previous section. The proposed operators have two aspects in common, even if there are slight variations between the results:

- *Drastic dimension reduction:* The results prove that there are a lot of operators $\mathbf{M}$ which drastically reduce the initial dimension n while satisfying the other compressed sensing prerequisites. More precisely, the exact reconstruction of Σ_k is usually possible with $m = \mathcal{O}(k \ln(n))$, which is negligible with respect to n when n goes to infinity[c]. Moreover, still with $m = \mathcal{O}(k \ln(n))$, the reconstruction error of certain approximately sparse vectors $\mathbf{x}^*$ is upper bounded by the ℓ_2 distance from $\mathbf{x}^*$ to Σ_k, that is the residual ℓ_2-norm if one approximates $\mathbf{x}^*$ by its best k-term approximation.

- *Randomized choice of the operator:* Interestingly, the proposed operators are not constructed in a deterministic manner. Instead, all the works consider a simple class $\mathcal{M}$ of operators, supplied with a probability distribution. The operator used to perform compressed sensing is randomly drawn in $\mathcal{M}$, and theoretical results prove that with "high probability" on the drawing of $\mathbf{M}$ in $\mathcal{M}$, $\mathbf{M}$ will satisfy the compressed sensing constraints. The term "high probability" usually refers to the fact that when $n \to \infty$, if m is chosen as $\mathcal{O}(k \ln(n))$, then the probability that $\mathbf{M}$ satisfies the reconstruction conditions is 1-$\mathcal{O}(n^{-\alpha})$, where α is a parameter depending on the considered signals, the model $\mathcal{M}$ and the desired reconstruction guarantees.

In particular, the considered classes $\mathcal{M}$ of operators, along with the corresponding probability distributions leading to compressed sensing guarantees, were:

- In (Candès et al., 2006), the authors considered the case where $\mathbf{M}$ was a submatrix of the Fourier matrix $\mathbf{F}$, defined by

$$\mathbf{F}r, s = \frac{1}{\sqrt{n}} \exp\left(-2\pi i \frac{(r-1)(s-1)}{n}\right). \tag{1.6}$$

  $\mathbf{M}$ was obtained by uniformly extracting m different lines of $\mathbf{F}$. In this case, the authors proved the theoretical and practical recovery of Σ_k with high probability provided $m = \mathcal{O}(k \ln(n))$.

- The even more groundbreaking papers (Donoho, 2006) and (Candès and Tao, 2006) prove similar results, and are often viewed as the founding works in compressed sensing. They consider the reconstruction of vectors of $\mathbb{R}^n$ with coefficients obeying a power-law decay, that is signals $\mathbf{x}^*$ for which the q^{th} largest entry is upper bounded by $Cq^{-\alpha}$ with $C, \alpha > 0$. They prove that if $m = \mathcal{O}(k \ln(n))$, then with high probability on the drawing of the linear operator $\mathbf{M}$, which will be detailed in the following, a signal $\mathbf{x}^*$ belonging to this class of signals can be reconstructed up to a precision of the order of the best k-term approximation of $\mathbf{x}^*$. The reconstruction can once again be cast as a ℓ_1 minimization problem (1.5).

  Even if their choices of operators is similar, there are slight differences between the two papers. In (Donoho, 2006), the author considers a set $\mathcal{M}$ of $m \times n$ matrices with unit-norm columns, provided with uniform distribution. In (Candès and Tao, 2006), the authors again consider Fourier subsampling matrices, as well as matrices with entries drawn *i.i.d.* either from a symmetric Bernoulli distribution (taking values ± 1 with probability $\frac{1}{2}$) or from a normal distribution.

These surprising results show that compressed sensing is feasible with particular dimension reducing operators. As has been mentioned, all these operators share in common

[c] Here and in the following, $\mathcal{O}(f(n))$ is a quantity which is upper bounded by $Af(n)$, where A is a positive constant, for n large enough.

the interesting property of being chosen randomly among a certain family. In (Candès and Tao, 2006), the authors also precise some common properties shared between the different considered families $\mathcal{M}$ of operators, which pins down the reason why they are well-suited for compressed sensing. These conditions will then be simplified into a powerful property described in the next section.

The Restricted Isometry Property The direct legacy of the aforementioned founding results is a series of two papers (Candès and Tao, 2005; Candès et al., 2006) which mainly introduce a simple sufficient condition on an operator $\mathbf{M}$ for it to satisfy the compressed sensing constraints (and even more). This property is called the *Restricted Isometry Property* (RIP) which is a property concerning a certain sparsity level t. This RIP states that for all t-sparse vectors $\mathbf{x}$,

$$(1-\delta)\|\mathbf{x}\|_2^2 \leq \|\mathbf{M}\mathbf{x}\|_2^2 \leq (1+\delta)\|\mathbf{x}\|_2^2 \tag{1.7}$$

for a certain $0 < \delta < 1$. The RIP therefore says that even if $\mathbf{M}$ is a highly dimensional-reducing operator, so that the column vectors which appear in $\mathbf{M}$ are highly linearly dependent, any subfamily of t such column vectors is nonetheless approximately orthogonal, so that $\mathbf{M}$ acts approximately as an isometry on $\frac{t}{2}$-sparse vectors (supposing t is even), hence the name "Restricted Isometry".

This property was first precised and used in (Candès and Tao, 2005) in a noncompressive setting, then applied to compressed sensing in (Candès et al., 2006). The success of the recovery procedure in a usual compressed sensing setting can be linked to the RIP:

- The previously mentioned random matrices asymptotically satisfy the RIP with high probability: this includes Gaussian, Bernoulli and Fourier subsample matrices.
- A matrix $\mathbf{M}$ satisfying some kind of RIP is well-suited for compressed sensing. Intuitively, let's suppose that $\mathbf{M}$ satisfies the RIP on the set of $2k$-sparse vectors. If $\mathbf{x}$ and $\mathbf{y}$ are k-sparse vectors, then the distances $\|\mathbf{x}-\mathbf{y}\|_2$ and $\|\mathbf{M}\mathbf{x}-\mathbf{M}\mathbf{y}\|_2$ will be close: the ratio between the two distances will be between $1-\delta$ and $1+\delta$. This makes intuitively understandable the recovery of k-sparse vectors: if δ is small, two k-sparse vectors cannot be mixed up in the compressed domain.

  This intuitive power of the RIP is theoretically justified: the results of (Candès et al., 2006) prove that if $\mathbf{M}$ satisfies certain RIPs, it satisfies the previously mentioned compressed sensing conditions, and moreover permits reconstruction from noisy measurements. Indeed, under some assumptions on RIP constants, if $\mathbf{x}^* \in \Sigma_k$ and $\mathbf{y} = \mathbf{M}\mathbf{x} + \mathbf{e}$ is a noisy measurement with $\|\mathbf{e}\|_2 \leq \sigma$, the minimization problem

  $$\underset{\mathbf{x}\in\mathbb{R}^n}{\operatorname{argmin}} \|\mathbf{x}\|_1, \text{ subject to } \|\mathbf{y}-\mathbf{M}\mathbf{x}\|_2 \leq \sigma \tag{1.8}$$

  yields a solution $\mathbf{x}^\#$ such that $\|\mathbf{x}^*-\mathbf{x}^\#\|_2 \leq C\sigma$, that is the reconstruction error is of magnitude similar to the noise of the measurement. Moreover, the reconstruction error of an approximately sparse vector with noisy measurements can also be upper bounded under RIP assumptions.

The main advantages of the RIP are its simplicity and the fact that it separates the probabilistic choice of an operator from the theoretical implications of the choice of a well-suited operator. Because of this, it has been widely used in the compressed sensing theory to prove theoretical reconstruction results. The simplicity of this property is particularly well illustrated in (Candès, 2008), where the author concisely proves noiseless

and noisy compressed reconstruction bounds for approximately sparse vectors under RIP assumptions.

Note that more recently, compressed sensing has been considered directly in a probabilistic framework, allowing one to get reconstruction results under "RIPless" conditions on $\mathbf{M}$ (Candès and Plan, 2011). The corresponding results directly give conditions on a family of operators $\mathcal{M}$ and on the random choice of these operators for compressed sensing to be feasible with high probability.

The importance of randomness. The main characteristic of compressed sensing is the random choice of the measurement operator. Indeed, even if the previously mentioned results prove the existence of multiple well-suited operators for compressed sensing, they do not provide a way to construct them in a deterministic way. There is actually no known deterministic way of building compressed sensing operators in polynomial time which have guarantees comparable to those obtained in a probabilistic manner.

The random choice of the operator, although strange at first glance, is a good way of finding a suitable operator for the initial purpose of compressively acquiring the signal: it is fairly useless to really aim for an operator which will be guaranteed to work, and one can instead pick randomly an operator among a simple model in which most of the elements are suitable. This good behavior of randomly drawn operators is theoretically linked to the *concentration of measure* phenomenon, which designates a collection of results underlying the fact that when a large number of random variables is drawn, the variability of a wide range of statistical quantities computed from these variables will be very low, so that the observed statistics will very likely be close to their expected value.

This idea of picking an element at random instead of struggling to find the "optimal" element while still having theoretical guarantees will be discussed in the next section, through examples of methods used in statistical analysis or optimization and aiming at reducing complexity and/or memory usage for different tasks.

1.2 "CS-like" techniques in statistical analysis

In parallel to the development of compressed sensing, and actually before, the idea of using randomness to save computational time or memory has been exploited in techniques more related to statistical analysis than signal processing.

We give some examples of such techniques, shedding some light on the similarities with compressed sensing. Apart from the following mentioned techniques, sketch-based methods share common points with compressed sensing and some of them will be discussed in Chapter 2.

1.2.1 Locality-Sensitive Hashing

Databases are often comprised of vectors living in the same vector space. When statistically manipulating a database, one often needs to perform a similarity search among these vectors, *i.e.*, looking for the nearest neighbors of a certain vector in the database. When the dimension and/or the number of vectors is large, it may become compulsory to rely on approximate but faster schemes. To this end, *hashing* can be exploited.

Hashing for approximate neighbor search. The core idea of hashing for computational savings is to replace the initial vectors $\mathbf{x}_i$, belonging to a metric space with metric d, by shorter *signatures* $h(\mathbf{x}_i)$ belonging to a simpler space with a metric d' which is easier

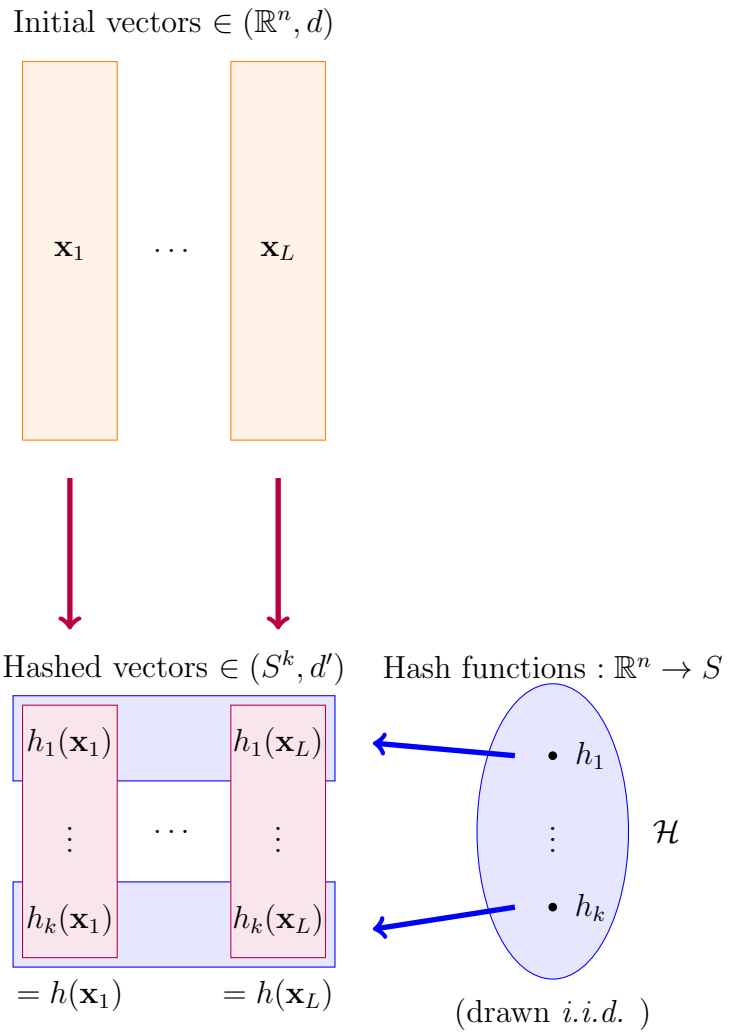

Figure 1.1: Core idea of LSH. Each initial vector is mapped to a signature for which each entry corresponds to the image of the vector by a randomly drawn hash function.

and faster to compute. Even though there is no immediate link between d and d', the hash functions are chosen so that d' approximately *preserves neighborhoods*: if $\mathbf{x}$ and $\mathbf{y}$ are "close neighbors" in the initial space, we should have $h(\mathbf{x}) \approx h(\mathbf{y})$ and conversely, if they are far apart, so should $h(\mathbf{x})$ and $h(\mathbf{y})$ be.

Locality Sensitive Hashing (LSH) (Indyk and Motwani, 1998; Gionis et al., 1999; Datar et al., 2004b) is a particular hashing scheme which was introduced to provide a fast approximate similarity search method for distances corresponding to the p-norm in dimension n:

$$\|\mathbf{x}\|_p = \left(\sum_{i=1}^{n} |x_i|^p \right)^{\frac{1}{p}}. \tag{1.9}$$

LSH relies on a family $\mathcal{H}$ of elementary hash functions mapping the initial space $\mathbb{R}^n$ to a finite set of elements S. The signature of a vector $\mathbf{x}$ is a k-tuple $h(\mathbf{x}) = (h_1(\mathbf{x}), \ldots, h_k(\mathbf{x})) \in S^k$, where the h_j are elements of $\mathcal{H}$ drawn *i.i.d.* with respect to a certain probability distribution on $\mathcal{H}$. This general framework is illustrated in Figure 1.1.

For the signature to be well-designed, the family $\mathcal{H}$ should be *locality-sensitive*, *i.e.*,

for any couple of vectors of $\mathbb{R}^n$, the probability that they are mapped to the same element in S (with respect to the drawing of the hash function) should be high if the vectors are close to one another, and low if they are far apart. More precisely, this property is usually stated as follows: for all $\mathbf{x}, \mathbf{y} \in \mathbb{R}^n$,

$$\begin{aligned} &\bullet \text{ If } d(\mathbf{x},\mathbf{y}) \leq r_1, \text{then } \mathbb{P}(h(\mathbf{x}) = h(\mathbf{y})) \geq p_1. \\ &\bullet \text{ If } d(\mathbf{x},\mathbf{y}) \geq r_2, \text{then } \mathbb{P}(h(\mathbf{x}) = h(\mathbf{y})) \leq p_2. \end{aligned} \tag{1.10}$$

Here, r_1, r_2, p_1 and p_2 are parameters of the locality sensitive property and $\mathbb{P}$ denotes the probability with respect to the drawing of the hash function h among the family $\mathcal{H}$. Obviously, for this property to have a true meaning, one must have $r_1 < r_2$ and $p_1 > p_2$. To make a parallel with compressed sensing, one may compare this property to the RIP: indeed, several hashing families satisfy this property, which is useful in showing theoretical guarantees for algorithms which use LSH, as we will mention in the following.

Among the families which satisfy this locality-sensitive property appears a family of binary functions $h_{\boldsymbol{\theta}}$ indexed on $\boldsymbol{\theta} \in \mathbb{R}^n$, with values in $\{0, 1\}$ and defined as $h_{\boldsymbol{\theta}}(\mathbf{x}) = 1$ if $\langle \boldsymbol{\theta}, \mathbf{x} \rangle \geq 0$ and $h_{\boldsymbol{\theta}}(\mathbf{x}) = 0$ otherwise (Goemans and Williamson, 1995; Charikar, 2002a). This family will be further discussed in Chapter 4, in a more general hashing setting. Other usually considered hash functions families are comprised of projections on a line followed by partitioning into segments (Datar et al., 2004b).

Note that LSH, and more generally hashing techniques aimed at reducing complexity, can be used in two different ways. The first basic way is to compute a single signature of respectable size for every database vector, then compute a full nearest neighbor search in the signature space, getting a list of nearest neighbors. This yields an easily implementable research scheme. This first way of exploiting hashing will be used in Chapter 4. On the other hand, one may compute several smaller signatures for a vector, each belonging to a different *bucket*. In this second case, a lot of vectors may have similar signatures in a given bucket. The nearest neighbor list of a "query" vector is computed by identifying the vectors which have similar signatures as the query in most buckets. This method relies on a harder implementation to be efficient, but can yield sublinear research times (Datar et al., 2004b; Andoni and Indyk, 2008). This second framework is usually the considered framework when one aims at proving theoretical guarantees for approximate nearest neighbor search.

In both cases, when a list of probable nearest neighbors is formed, an additional step of *reranking* is performed, where the true distances between the query and the considered vectors are computed in order to better identify the true nearest neighbors.

Theoretical results about LSH can be proved: if a family $\mathcal{H}$ satisfies the locality-sensitive property stated above, the main result one can obtain is that with suitable choices for the number of buckets and the dimension of the hashed vectors, LSH can be used to efficiently solve (with high probability on the drawing of the hash functions) the so-called ϵ-nearest neighbor search problem, that is given a query $\mathbf{q}$ and its nearest neighbor $\mathbf{x}$ in a database $\mathcal{X}$, find a vector $\mathbf{y} \in \mathcal{X}$ such that

$$d(\mathbf{q}, \mathbf{y}) \leq (1 + \epsilon) d(\mathbf{q}, \mathbf{x}). \tag{1.11}$$

Similarity with compressed sensing. Even though the frameworks of compressed sensing and LSH are very different, there are interesting similarities in the procedures and the theoretical study: in both cases, one randomly draws a dimension reduction operator among a certain model, and it can be proven that in suitable conditions, with high probability on the drawing of the operator, the compressed representation can be used to precisely perform a certain task.

1.2.2 Compressed methods for matrix decomposition

Complexity of matrix decompositions. Linear algebra is a core tool in statistical analysis. In particular, data can often be represented as a set of vectors of the same dimension, concatenated into a matrix. Typically, each column represents an individual and each line is a *feature*, which is a numerical quantity characterizing some aspect of the individual. When considering such a matrix, it is often useful to simplify its representation through a certain decomposition. One of the most commonly considered decomposition in the Singular Value Decomposition (SVD), which provides optimal low-rank decompositions in the Frobenius sense.

SVD and other matrix decompositions are numerically performed by iterative algorithms. They typically require extensive access to the matrix, so that the matrix should be small enough to fit in main memory. Moreover, if the matrix is of size $m \times n$, the usual complexity of the decomposition of the matrix is $\mathcal{O}\left(mn \min(m,n)\right)$. If one only needs a rank-k approximation of the matrix without performing a full decomposition, other iterative algorithms can be used which have complexity $\mathcal{O}(mnk)$. These algorithms with reduced complexity still require extensive access to the matrix.

Randomized scheme for matrix decomposition. Finding more efficient ways (either in complexity or in memory usage) than the usual iterative algorithms to perform matrix decomposition has been the topic of a lot of works ((Mahoney, 2011; Halko et al., 2011) compile numerous references). In particular, (Halko et al., 2011) describes several randomized methods for computing matrix decompositions such as Singular Value Decomposition (SVD) or QR decomposition. Typically, randomness is used to quickly compute a low-rank estimate of the matrix, and then a standard decomposition algorithm is applied to this estimate.

More precisely, if the matrix $\mathbf{N}$ to be decomposed approximately is of size $m \times n$, the randomized layout for computing the decomposition is composed of two stages:

1. *Approximation of the action of the matrix:* The first goal is to efficiently "capture" the action of the matrix, that is compute a low-rank estimate of the action of the matrix. Typically, a matrix will not have singular values of the same magnitude, so that the energy of a random vector will be mostly mapped by the matrix on the first singular directions. This phenomenon is used for randomized approximation: if one searches for a rank-k approximation, the matrix $\mathbf{N}$ is applied to a set of $k+p$ random vectors (typically Gaussian vectors), where $p > 0$ is an oversampling parameter, usually taken constant and $\ll k$. The complexity of this step is $\mathcal{O}(mnk)$, but can be reduced to $\mathcal{O}(mn\ln(k))$ by applying $\mathbf{N}$ to a structured set of random vectors, such as Subsample Random Fourier Transform (SRFT) (Woolfe et al., 2008). The output is an $m \times (k+p)$ orthogonal matrix $\mathbf{Q}$ such that

$$\|\mathbf{N} - \mathbf{Q}\mathbf{Q}^T\mathbf{N}\| \approx \|\mathbf{N} - \mathbf{N}_k\|, \tag{1.12}$$

   where $\|\cdot\|$ typically denotes the Frobenius or the ℓ_2 operator norm and $\mathbf{N}_k$ is the best rank-k approximation of $\mathbf{N}$ with respect to the norm $\|.\|$.

2. *Decomposition of the matrix:* Since the matrix $\mathbf{Q}$ allows to compute a good low-rank approximation of the main singular components of $\mathbf{N}$, the desired decomposition of $\mathbf{N}$ can directly be performed on the much smaller matrix $\mathbf{Q}^T\mathbf{N}$. For SVD, it is for instance sufficient to compute the SVD of $\mathbf{Q}^T\mathbf{N} = \mathbf{U}\mathbf{\Sigma}\mathbf{V}^*$ and the approximate SVD of $\mathbf{N}$ (up to rank k) is then given by $\mathbf{N}_k \approx (\mathbf{Q}\mathbf{U})\mathbf{\Sigma}\mathbf{V}^*$.

This framework can be adapted to other decompositions such as QR. The complexity is not different from a standard partial SVD algorithm for general dense matrices, but since this layout relies mainly on matrix-vectors multiplications to reduce the size of the data, the complexity can be significantly dropped if these multiplications can be quickly computed, typically if the matrix is sparse. Finally, the scheme can also be adapted to aim at reducing the number of passes on the matrix $\mathbf{N}$, in the case where $\mathbf{N}$ is too large to fit in main memory and the bottleneck of a decomposition algorithm is the disk access time.

Links with compressed sensing. These randomized decompositions have provable precision bounds, relying on concentration of measure tools. The results are various, but they mainly consist either in an average case analysis, upper bounding the expectation of the decomposition error over the drawing of the random vectors, or in a probabilistic bound on the error, upper bounding it with high probability on the drawing of the random vectors, similarly to compressed sensing. In these upper bounds usually appear the singular values of the matrix $\mathbf{N}$ for ranks $> k$, so that the theoretical results confirm the intuition that the randomized scheme will be useful only if the singular values of $\mathbf{N}$ decrease at a sufficient rate.

These randomized decomposition methods share certain common points with the compressed sensing framework: knowing that a matrix has sufficiently decreasing singular values (which can be compared to the sparse model of compressed sensing), it is possible to precisely estimate a low-rank decomposition from a compressed representation of the matrix obtained via its image from a randomly picked operator. However, a singular difference is that one still needs to access the initial matrix to perform the decomposition in the second step, whereas compressed sensing ensures the reconstruction of the initial vector simply from the compressed representation and the projection operator.

1.2.3 Other randomized statistical procedures

In this section, we will mention other randomized schemes that have been used in various learning frameworks, but which have fewer immediate links with compressed sensing. However, the core idea is to save computational time by "picking randomly an item" instead of "searching for the best item" to perform a task, which draws links to compressed sensing.

Linear kernel approximations Kernels (Scholkopf and Smola, 2001) are widely used functions in machine learning: they provide a considerable range of metrics between elements of a database and have nice properties leading to standard algorithms to perform learning, such as Support Vector Machines (SVM). However, since these metrics can be elaborate, the cost of computing kernel values is usually substantial, so that kernel methods cannot be applied at a very large scale.

In the past few years, there has been in the machine learning community a gain of interest for *explicit embedding* methods, which consist in finding an approximation of a certain kernel K as a scalar product, that is finding a certain (nonlinear) transformation $\tilde{\Phi}$ of the data satisfying $K(\mathbf{x}, \mathbf{y}) \approx \langle \tilde{\Phi}(\mathbf{x}), \tilde{\Phi}(\mathbf{y}) \rangle$. The benefit is that once the data has been transformed by $\tilde{\Phi}$, the approximate kernel values are fast to compute, since the computation of scalar products relies on very optimized standard linear algebra routines. This kind of approximation will be further discussed in Chapter 4.

In (Rahimi and Recht, 2007), the authors consider two random schemes for defining such a function $\tilde{\Phi}$. They consider the case where the kernel K to approximate is a shift-invariant kernel on $\mathbb{R}^n$, that is $K(\mathbf{x}, \mathbf{y})$ only depends on the difference $\mathbf{x} - \mathbf{y}$.

Their first method relies on Bochner's theorem, which allows to write $K(\mathbf{x}, \mathbf{y})$ as the Fourier transform of a probability measure, that is $K(\mathbf{x}, \mathbf{y})$ can be seen as an expected value of cosines. Therefore, they propose to approximate $K(\mathbf{x}, \mathbf{y})$ as an empirical mean of cosines drawn from the proper probability distribution, which can be considered as a scalar product. Concentration of measure results provide a uniform bound on the approximation error of the kernel on a compact subset of $\mathbb{R}^n$ with high probability on the drawing of the cosines.

The second method is less general and applies only to certain type of shift-invariant kernels. It is similar to binary hashing and consists in randomly partitioning the space into bins, ensuring the probability of collision between two vectors $\mathbf{x}, \mathbf{y}$ on a compact subset of $\mathbb{R}^n$ is proportional to $K(\mathbf{x}, \mathbf{y})$. By averaging several such hashed representations, one again gets an approximation of the kernel that can be seen as a scalar product. Moreover, a uniform bound on the approximation error can still be derived, upper bounding it with high probability on the drawing of the hash functions.

In both cases, the probability distributions used to draw the function $\tilde{\Phi}$ are adapted to the kernel one wants to approximate. These linear random approximations were shown to yield substantial savings on training time for several learning databases and problems (Rahimi and Recht, 2007).

Random Kitchen Sinks. In (Rahimi and Recht, 2008), the authors consider a binary classification problem where the classifier is sought as the sign of a linear combination of some simple classifiers taken from a family $\Omega = \{\Phi_\theta : \theta \in \Theta\}$. This framework encompasses classification models such as SVM or boosting.

Given a training set of labeled points $\{(\mathbf{x}_i, y_i)\}_{i=1}^L \subset \mathbb{R}^n \times \{-1, 1\}$, the search for such a classifier is usually expressed as the following minimization problem:

$$\underset{\theta_1,\ldots,\theta_m,\alpha_1,\ldots,\alpha_m}{\operatorname{argmin}} \sum_{i=1}^{L} \ell\left(\sum_{q=1}^{m} \alpha_q \Phi_{\theta_q}(x_i), y_i\right), \tag{1.13}$$

where m is a predetermined parameter enforcing the maximal number of terms in the classifier and $\ell(\cdot, \cdot)$ is a loss function measuring the discrepancy between the predicted class and the observed class. This optimization problem is complex, especially when finding correct values for θ_q, since the relationship between θ and Φ_θ can be complicated.

The authors therefore propose to simplify this optimization problem by relaxing the optimization on the parameters θ_q: instead of finding the best m such parameters, one could just pick m parameters $\theta_1, \ldots, \theta_m$ at random in Θ with respect to a probability density p. The optimization problem simply becomes a simple matter of minimizing over the weights $\boldsymbol{\alpha} \in \mathbb{R}^m$ the quantity

$$\sum_{i=1}^{L} \ell\left(\boldsymbol{\alpha}^T \mathbf{z}_i, y_i\right), \tag{1.14}$$

where $\mathbf{z}_i$ is the concatenated feature vector $[\Phi_{\theta_1}(x_i), \ldots, \Phi_{\theta_m}(x_i)]^T$.

Under regularity conditions on the elements of Ω and the loss function ℓ, with high probability on the drawing of the parameters θ_q, one can upper bound the difference

between the loss induced by the solution of (1.14) and the loss induced by the best classifier in a certain regular family of linear combinations of elements in Ω.

Atop this theoretical bound, experiments show that this randomized formulation yields substantial savings in training time for classification using AdaBoost.

Stochastic gradient Another interesting example where randomness can proficiently be used to reduce learning costs is the so-called *stochastic gradient* method, which relies on a method initially proposed in (Robbins and Monro, 1951). A long time after, the method has been readjusted to online learning and studied statistically (Bottou, 1998; Murata, 1998).

The stochastic gradient is typically applied to problems where one aims at minimizing with respect to $\mathbf{x}$ an objective function of the form $f(\mathbf{x}) = \sum_{i=1}^{L} \ell_i(\mathbf{x})$, where the terms $\ell_i(\mathbf{x})$ are typically loss terms of individual data of a training set. Provided all the functions ℓ_i are differentiable with respect to $\mathbf{x}$ a usual gradient descent algorithm would need to compute all terms $\nabla \ell_i$ to find the gradient descent direction ∇f. However, it has been proven that in a large-scale setting, it is sometimes preferable to reduce the cost of computing this direction by iteratively considering a descent direction of the form $\nabla \ell_i$ for a *particular* i, which is chosen randomly among the training data in the case of stochastic gradient. While such a method obviously yields a cost reduction of a single iteration of the optimization algorithm, it has been proven theoretically that it can also offer a better time/precision tradeoff at a large scale (Bottou and Bousquet, 2007).

Compressed least-squares regression. In (Maillard and Munos, 2009), the authors propose a compressive framework to perform linear regression. Given data $\mathcal{X} = \{(\mathbf{x}_r, y_r)\}_{r=1}^{L} \subset \Omega \times \mathbb{R}$ and a family $\{f_s : \Omega \to \mathbb{R}\}_{s=1}^{n}$ of functions called features, the initial goal is to find a vector $\boldsymbol{\alpha} \in \mathbb{R}^n$ such that the function $f_{\boldsymbol{\alpha}} = \sum_{s=1}^{n} \alpha_s f_s$ is a viable regression function, that is for all r,

$$y_r \approx f_{\boldsymbol{\alpha}}(\mathbf{x}_r). \tag{1.15}$$

The authors consider the case where $\boldsymbol{\alpha}$ is chosen in a penalized least-squares fashion, that is minimize a quantity of the type

$$\sum_{r=1}^{L} |y_r - f_{\boldsymbol{\alpha}}(\mathbf{x}_r)|^2 + J(\boldsymbol{\alpha}), \tag{1.16}$$

where J is a regularization function.

Instead of searching for a regression function of the form $f_{\boldsymbol{\alpha}}$, the authors propose to compress the set $\{f_s\}$ by replacing it with a set of m features $\{g_t\}_{t=1}^{m}$, obtained as random linear combinations of f_s. The proposed random combinations follow analogous probability distributions to the distributions of (Achlioptas, 2001), aimed at keeping the scalar products between a certain number of vectors while ensuring the linear combinations are somewhat sparse in order to fasten the projections computations. The regression function is then sought as a function $g_{\boldsymbol{\beta}} = \sum_{t=1}^{m} \beta_t g_t$, with $\boldsymbol{\beta} \in \mathbb{R}^m$.

The authors derive theoretical bounds on the average risk of the optimal regression function of the compressed model. For a suitable choice of the parameter m, one can obtain similar error bounds on the approximation error as other noncompressive methods, while learning the regressing function with a reduced complexity.

1.3 Conclusion

The success of compressed sensing shows us that randomization can be very robust if applied to the right model and problem. The provided examples of computational methods using randomization to solve a learning task act as an incentive to express such a task in a framework analog to compressed sensing. This will hopefully allow to develop learning methods which combine numerical efficiency and theoretical soundness.

From this observation, we will build our first contribution in the following chapter by proposing and instantiating a compressive learning framework on a usual estimation problem.

Chapter 2

Compressive density mixture estimation

This chapter will appear in a slightly modified version in the Springer/Birkhäuser book entitled *Compressed sensing and its applications*.

As we have seen in the previous chapter, methods that are close to compressed sensing have been exploited to reduce complexity and/or memory costs of several learning tasks. To further study the potential of compressed sensing to learning problems, let's first present a very conceptual view of what would be a "compressive learning" framework.

Conceptual compressive learning outline. When considering a learning problem, one is interested in performing a certain task on a certain type of data. A learning procedure will usually consist in trying to fit an underlying model to the type of data one is interested in by picking a model in a parametrized set $\mathcal{M} = \{M_{\boldsymbol{\theta}} : \boldsymbol{\theta} \in \Theta\}$, where Θ is a parameter set which is used to index the models in $\mathcal{M}$. In order to achieve this model fitting, a learning procedure will consist in finding a parameter $\boldsymbol{\theta}^*$ so that the model $M_{\boldsymbol{\theta}^*}$ is in some sense adequate to a training set $\mathcal{X} = \{\mathbf{x}_1, \dots, \mathbf{x}_L\}$ containing some data one is interested in. The computational cost of estimating such a parameter $\boldsymbol{\theta}^*$ will depend on the size of $\mathcal{X}$, on the size of the models in $\mathcal{M}$ and on the algorithm used for the estimation.

A compressive framework to perform such an estimation is outlined in Figure 2.1, which represents two main ways of compressing training data in order to apply a learning algorithm to data of reduced size. The top scheme represents the case where each vector of $\mathcal{X}$ is compressed individually: this is performed for instance in (Calderbank et al., 2009), with a method which will be described in the next section. The bottom scheme represents the case where $\mathcal{X}$ will be compressed into a single representation usually called *sketch*, of size m which should not depend on the number L of elements in the database but rather on the complexity of the model one wants to fit (represented by the parameter set Θ) and on the task one aims at achieving after the learning. This second scheme has been instantiated in a simple estimation problem in (Thaper et al., 2002), which will also be discussed in the next section.

Density mixture estimation. In this chapter, we will focus more particularly on a classical unsupervised learning problem: density mixture estimation.

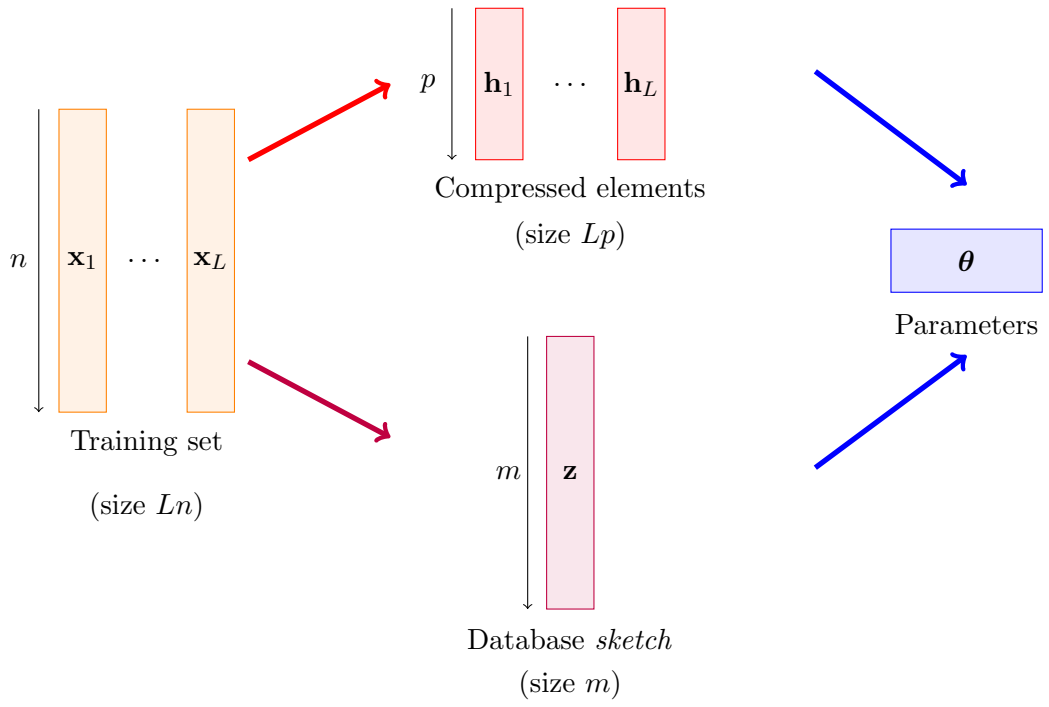

Figure 2.1: Compressive learning outline. The learning data $\mathcal{X} = \{\mathbf{x}_1, \dots, \mathbf{x}_L\}$ is compressed into a smaller representation, which can either consist in reducing the dimensions of each individual entry $\mathbf{x}_r$ or in computing a more global compressed representation of the data, called *sketch*. Parameters $\boldsymbol{\theta}$ are then inferred from such a compressed representation by an algorithm adapted to it.

Let's model the problem as follows: suppose the data set $\mathcal{X}$ comprises vectors of $\mathbb{R}^n$, and consider a set $\mathcal{P}$ of parametrized probability densities defined on $\mathbb{R}^n$, that is

$$\mathcal{P} = \left\{ p_{\boldsymbol{\theta}} : \mathbb{R}^n \to \mathbb{R}^+ \,\middle|\, \int_{\mathbb{R}^n} p_{\boldsymbol{\theta}} \, d\mu = 1, \boldsymbol{\theta} \in \Theta \right\}, \tag{2.1}$$

where μ is the Lebesgue measure on $\mathbb{R}^n$.

The goal of density mixture estimation is to find a density function p on $\mathbb{R}^n$ which satisfies two prerequisites:

- p must be written as a linear combination, or *mixture*, of functions in $\mathcal{P}$, that is $p = \sum_{s=1}^{k} \alpha_s p_{\boldsymbol{\theta}_s}$, where the quantities α_s are typically positive *weights* satisfying $\sum_{s=1}^{k} \alpha_s = 1$ and $\boldsymbol{\theta}_s$ are the *parameters* of the mixture. Let's notice the similarity of this condition with the sparsity condition discussed in the previous chapter.
- $\mathcal{X}$ can "reasonably well" be considered as a set of vectors drawn *i.i.d.* with respect to a probability law of density p. This condition is typically enforced by optimizing a certain quantity representing the consistency between the data $\mathcal{X}$ and the probability p. Said quantity is usually closely related to the parametrization of the probability densities, such as the likelihood of the parameters $\mathbb{P}(\mathcal{X}|(\alpha_s, \boldsymbol{\theta}_s)_{s=1}^k)$, representing the probability of the drawing of the data in $\mathcal{X}$ knowing the probability density p. Alternatively, this condition can be formulated by supposing the data $\mathcal{X}$ is drawn *i.i.d.* from a probability distribution of density f, and that one searches for the best approximation p of f (in a certain sense), p being an element of the set of sparse linear combinations of functions in $\mathcal{P}$.

Since p must be written as a linear combination of a few functions of $\mathcal{P}$, one can consider it as a "sparse vector" over the family $\mathcal{P}$. In that sense, several works have considered the density mixture estimation problem in a linear inverse problem fashion, and will be discussed in the next section. Drawing our inspiration from these works, our goal in this chapter will be to propose an instantiation of the compressive scheme represented in bottom part of Figure 2.1, applying it to a density mixture estimation problem.

In the next section, on top of discussing the works relative to density mixture estimation in a linear inverse problem fashion, we will also mention some other "compressive learning" works which can also be considered as instantiations of the outline in Figure 2.1.

2.1 State of the art: density mixture estimation and compressed learning

In this section, we present different sets of works that motivate our contribution: on the one hand, works on density mixture estimation expressed as a linear inverse problem; on the other hand, works on "compressed learning" aiming at performing learning tasks on compressed data, some of which coming from the database literature with a concern in analyzing data streams thanks to global compressed representations called *sketches*. Our approach will combine both aspects.

2.1.1 Density mixture estimation as a linear inverse problem

To express the density mixture estimation problem in a linear inverse fashion, the works (Bunea et al., 2010; Bertin et al., 2011) consider $\mathcal{P}$ as a finite set $\{p_1, \ldots, p_k\}$. Moreover, they also consider the alternative formulation of density mixture estimation mentioned

earlier, that is $\mathcal{X}$ is drawn *i.i.d.* from a probability distribution of density f. Finally, they consider the case where all the densities p_s and the density f belong to the Hilbert space $L^2(\mathbb{R}^n)$.

In this case, the density mixture estimation consists in finding a sparse vector $\boldsymbol{\alpha} \in \mathbb{R}^k$, such that $f \approx f_{\boldsymbol{\alpha}}$, where $f_{\boldsymbol{\alpha}} = \sum_{s=1}^{k} \alpha_s p_s$. This objective can be specified in different ways in a linear inverse problem fashion. They all share the following observation: if $\mathbb{E}$ is the expectation with respect to the density f, then the scalar product between f and p_s, defined by

$$\langle f, p_s \rangle = \int_{\mathbb{R}^n} p_s f \, d\mu = \mathbb{E}[p_s(X)], \tag{2.2}$$

can be approximated by an empirical mean obtained from the data $\mathcal{X}$. This empirical estimator is defined as

$$\hat{\mathbb{E}}[p_s(X)] = \frac{1}{L} \sum_{r=1}^{L} p_s(x_r), \tag{2.3}$$

where $\hat{\mathbb{E}}$ is the expectation with respect to the empirical probability distribution of the data $\mathcal{X}$.

In (Bunea et al., 2010), the authors aim at minimizing over $\boldsymbol{\alpha}$ the quantity

$$\|f - f_{\boldsymbol{\alpha}}\|_2^2 + J(\boldsymbol{\alpha}), \tag{2.4}$$

where J is a penalty function that promotes sparsity, defined as a weighted version of ℓ_1-norm depending on the family $\mathcal{P}$.

The first term can be developed using the scalar product, and can be replaced without changing the solution to problem (2.4) by the term

$$-2\mathbb{E}[f_{\boldsymbol{\alpha}}(X)] + \|f_{\boldsymbol{\alpha}}\|_2^2. \tag{2.5}$$

From there, the authors simply replace the theoretical expectation $\mathbb{E}$ in (2.5) by its empirical counterpart $\hat{\mathbb{E}}$. This defines the "SPADES" estimator as

$$\hat{\boldsymbol{\alpha}} = \underset{\boldsymbol{\alpha} \in \mathbb{R}^k}{\operatorname{argmin}} \; -\frac{2}{L} \sum_{r=1}^{L} f_{\boldsymbol{\alpha}}(\mathbf{x}_r) + \|f_{\boldsymbol{\alpha}}\|_2^2 + J(\boldsymbol{\alpha}). \tag{2.6}$$

They further propose oracle inequalities for the estimation error under different assumptions on the family $\mathcal{P}$ and the choice of the penalty term J.

In (Bertin et al., 2011), the authors apply the Dantzig selector (Candès and Tao, 2007) to the problem by expressing it as

$$\underset{\boldsymbol{\alpha} \in \mathbb{R}^k}{\operatorname{argmin}} \; \|\boldsymbol{\alpha}\|_1, \text{ subject to } |(\mathbf{G}\boldsymbol{\alpha})_s - \hat{\mathbb{E}}[p_s(X)]| \leq \eta_s \text{ for all } 1 \leq s \leq k. \tag{2.7}$$

In this formulation, the matrix $\mathbf{G}$ is the Gram matrix of the family $\mathcal{P}$, for which the (s_1, s_2) entry is equal to $\langle p_{s_1}, p_{s_2} \rangle$. The quantity η_s is an adaptive threshold relative to each density p_s. The objective aims at finding a sparse solution via ℓ_1 relaxation, while the constraints ensure that the theoretical correlation $\langle p_s, f_{\boldsymbol{\alpha}} \rangle = (\mathbf{G}\boldsymbol{\alpha})_s$ is close to the empirical estimator of $\langle p_s, f \rangle$ defined as $\hat{\mathbb{E}}[p_s(X)]$.

These two methods have been successfully applied to density mixture estimation for several models in dimension $n = 1, 2$ in (Bunea et al., 2010) and $n = 1$ in (Bertin et al., 2011). However, they suffer from two major drawbacks if one aims at applying them to higher-dimensional models. Let's give somewhat intuitive insights into the reasons for these drawbacks:

- *Finiteness of the density model:* The family $\mathcal{P}$ is supposed finite. However, several density models typically used in estimation are infinite, such as Gaussian Mixture Models (GMMs), which consider Gaussian densities indexed on a certain number of parameters depending on n. If one aims for instance at applying these methods to GMM estimation, one needs to discretize the continuous model of Gaussian densities. Since a number of $\left(\frac{R}{h}\right)^n$ is required to sample a cube of side R with a mesh step h in each direction, the number of centers in a discrete representation will grow exponentially fast with n, which will not be viable computationally.

- *Incoherence of the density model:* Both methods rely on the fact that $\mathcal{P}$ is a family of *incoherent* densities, that is the quantities

$$\frac{\langle p_{s_1}, p_{s_2} \rangle}{\|p_{s_1}\|_2 \|p_{s_2}\|_2} \tag{2.8}$$

  are not too close to 1 if $s_1 \neq s_2$. This incoherence necessity prevents from decomposing f on a refined model containing many similar densities.

Therefore, if these approaches are viable and theoretically sound for density mixture estimation in small dimension ($n = 1, 2, 3$), they may not be applied in all generality for moderate dimensions (say, even $n = 5, 10$). For such dimensions, it would be more interesting to consider a continuous model of densities $p_{\boldsymbol{\theta}}$ indexed by a continuous parameter $\boldsymbol{\theta}$, such as a vector of a certain space $\mathbb{R}^d$. However, such a continuous model cannot be exploited by the aforementioned methods.

Keeping these limitations in mind, let's present the other set of inspiring contributions for our work, which can be viewed as compressive learning instances.

2.1.2 Learning with randomized dimensionality reduction and sketching

Learning with individual dimension reduction. In (Calderbank et al., 2009), the authors address the classical linear Support Vector Machine (SVM) classification problem : labeled data $(\mathcal{X}, \mathcal{Y}) = \{(\mathbf{x}_r, y_r)\}_{r=1}^L \subset \mathbb{R}^n \times \{-1, 1\}$ must be separated by a hyperplane with normal vector $\mathbf{u}$ satisfying $y_r = \text{sign}(\langle \mathbf{u}, \mathbf{x}_r \rangle)$ for all $r \in [\![1, L]\!]$. Since this condition usually cannot be satisfied, one often considers a relaxation of this problem where $\mathbf{u}$ is sought as the minimizer of a loss ℓ

$$\sum_{r=1}^{L} \ell(\langle \mathbf{v}, \mathbf{x}_r \rangle, y_r) \tag{2.9}$$

over all vectors $\mathbf{v}$ of $\mathbb{R}^n$. A reformulation of this problem in the case where ℓ is the so-called Hinge loss proves that $\mathbf{v}$ can be looked for as a linear combination of the vectors $\mathbf{x}_r$. Therefore, the objective function is expressed in terms of scalar products $\langle \mathbf{x}_{r_1}, \mathbf{x}_{r_2} \rangle$.

In order to reduce the learning complexity of optimizing the objective function (2.9), the authors propose to reduce the dimension of the vectors $\mathbf{x}_r$ by replacing them with vectors $\mathbf{M}\mathbf{x}_r$, with $\mathbf{M} : \mathbb{R}^n \to \mathbb{R}^m$ being a dimensionality-reducing linear operator. Since the vectors $\mathbf{x}_r$ are supposedly sparse, we have seen in Chapter 1 that it is possible to randomly choose $\mathbf{M}$ in order to get with high probability a RIP so that $\mathbf{M}$ acts approximately as an isometry on $\mathcal{X}$. Therefore, we will have

$$\langle \mathbf{x}_{r_1}, \mathbf{x}_{r_2} \rangle \approx \langle \mathbf{M}\mathbf{x}_{r_1}, \mathbf{M}\mathbf{x}_{r_2} \rangle. \tag{2.10}$$

The learning problem can thus approximately be tackled with the lower-dimensional vectors $\mathbf{M}\mathbf{x}_r$ instead of the vectors $\mathbf{x}_r$. The authors derive upper bounds on the error incurred by the low-dimensional projection.

Data stream sketches. Data streams consist in a flow of data $(\mathbf{x}_r)_{r\geq 1}$. In some models, the data stream is important in itself; in others, the data $\mathbf{x}_r$ act simply as modification step of an underlying item $\mathbf{x}$, which is the object of interest (for instance, $\mathbf{x}$ could be a vector and the elements $\mathbf{x}_r$ could encode modifications of this vector such as adding or subtracting a unit in a particular entry).

Standard statistical problems involving data streams include the search for frequent items among the $\mathbf{x}_r$, usually called *heavy hitters* (Gilbert et al., 2007; Cormode and Hadjieleftheriou, 2010), or more generally estimation of quantiles of the content of the stream. When aiming at obtaining such information at a given time without having to store all the data flow up to this point, it is necessary to maintain a compressed representation of the data stream which will allow one to perform the desired estimations. Such compressed representations can be deterministically built, but are sometimes built similarly to hash functions. In this case, they are called *sketches*.

A sketch $\mathbf{z}$ is usually updated each time a new element $(\mathbf{x}_r)$ is streamed. Examples of usual sketches include the Count Sketch (Charikar et al., 2002) and the Count-Min Sketch (Cormode and Muthukrishnan, 2004). They both rely on updating $\mathbf{z}$ thanks to randomly chosen hash functions. They are mainly used to estimate the heavy hitters.

Compressed histogram estimation. Interestingly, (Thaper et al., 2002) proposes a sketching procedure which can be linked to compressed sensing and to density mixture estimation. In this work, the authors consider a data stream $(\mathbf{x}_r)_{r\geq 1}$, where each $\mathbf{x}_r$ is a n-dimensional vector taken in a finite set $A \subset \mathbb{R}^n$. The goal is to obtain, at a given point r_0, a histogram H_{r_0} approximating the distribution of vectors $\mathbf{x}_r$ for $r \leq r_0$.

In order to avoid storing and updating a complete histogram of the data stream as it flows, the authors propose instead to build and update a sketch of such a histogram. This sketch is obtained by considering a low-dimensional projection of a histogram H by a randomly built linear operator $\mathbf{M}$ designed to approximately keep distances between histograms, still in a way similar to (Johnson and Lindenstrauss, 1984; Achlioptas, 2001). The sketch can be updated at each time r by considering $\mathbf{x}_r$ as a histogram H_r which is null everywhere except in the bin corresponding to $\mathbf{x}_r$, where it is 1.

The benefit of this framework is to reduce memory costs, since the whole histogram needs not be updated, while still being able to compute at any time a good approximation of the histogram. However, the main drawback is the complexity of the recovery procedure, which is exponential in the dimension n of the data. This prevents from applying this method to even moderate dimensions (say, $n = 10$).

2.2 Layout of the chapter

In Section 2.3, we build an instantiation of the second scheme of Figure 2.1 to a density mixture estimation problem. We identify several conditions that such a compressive framework must satisfy to succeed in proposing a correct answer to the problem of compressive mixture estimation. We provide a precise framework to tackle this problem in the case where the considered probability densities are isotropic Gaussians. We will also prove that it is possible to (deterministically) linearly compress any sparse mixture of k isotropic Gaussians in dimension n with less than $8k^3n$ measurements.

In Section 2.4, we propose an algorithm derived from Iterative Hard Thresholding IHT (Blumensath and Davies, 2009a) to solve the considered compressive estimation problem. We propose an analysis of the memory usage of the algorithm and discuss its computational cost.

In Section 2.5, we apply our compressive algorithm to synthetic data. We compare the precision of the compressive estimation to the estimation obtained with a standard EM algorithm (Dempster et al., 1977) with respect to two usual pseudo-distance between densities. We experimentally show that the compressive scheme achieves similar precision as an EM algorithm while not requiring to store the learning data, thus leading to memory savings when the data is numerous.

2.3 Compressive estimation framework

Let's recall that we want to consider the following problem in a compressive way: let $\mathcal{X} = \{\mathbf{x}_r\}_{r=1}^{L}$ be vectors of $\mathbb{R}^n$, supposedly *i.i.d.* from a certain probability distribution of density $p \in L^1(\mathbb{R}^n)$. In the rest of this chapter, the notation $\Sigma_k^+(\mathcal{P})$ will denote the positive linear combinations of k densities in $\mathcal{P}$, that is

$$\Sigma_k^+(\mathcal{P}) = \left\{ \sum_{s=1}^{k} \lambda_s p_{\boldsymbol{\theta}_s} : \lambda_s \in \mathbb{R}_+, \boldsymbol{\theta}_s \in \Theta \right\}. \tag{2.11}$$

Let's note that $\mathcal{P} \subset \Sigma_k^+(\mathcal{P})$. Our goal is to find a good estimate of p in the set $\Sigma_k^+(\mathcal{P})$.

To simplify the problem, we will suppose that p is an exact k-sparse mixture of densities taken in $\mathcal{P}$, defined in (2.1), that is $p = \sum_{s=1}^{k} \alpha_s p_{\boldsymbol{\theta}_s}$, with $\alpha_s \geq 0$, $\sum_{s=1}^{k} \alpha_s = 1$ and $\boldsymbol{\theta}_s \in \Theta$. In this case, the most natural way to approximate p with a density of $\Sigma_k^+(\mathcal{P})$ is to try and estimate $\boldsymbol{\alpha} = (\alpha_1, \ldots, \alpha_k)$ and $\boldsymbol{\theta}_1, \ldots, \boldsymbol{\theta}_k$ from $\mathcal{X}$. We further want to perform this estimation in a compressive fashion. By analogy with compressed sensing, let's derive a conceptual method.

2.3.1 The compressive operator.

The unknown "signal" p is a sparse linear combination of the $p_{\boldsymbol{\theta}}$'s. One would like to reconstruct p from a "compressive" measure $\mathbf{M}p$, where $\mathbf{M}$ is a linear operator which transforms a density function into a finite-dimensional representation, so that one is able to manipulate it. To reconstruct p from $\mathbf{M}p$, one will then look for an element q of $\Sigma_k^+(\mathcal{P})$ satisfying $\mathbf{M}q \approx \mathbf{M}p$. The main raised issue for now is to design an adequate linear measurement operator $\mathbf{M}$. This operator should satisfy the following requirements:

- *Estimation of* $\mathbf{M}p$*:* The empirical representation of the density p is the discrete density associated to the collection of vectors $\mathcal{X}$. Since p is unknown, one cannot compute directly $\mathbf{M}p$, and so the operator $\mathbf{M}$ must be such that $\mathbf{M}p$ can be estimated through this empirical distribution.

- *Computation of* $\mathbf{M}\mathcal{P}$*:* Intuitively, the reconstruction algorithm will aim at finding densities of $\Sigma_k^+(\mathcal{P})$ which will have an image by $\mathbf{M}$ similar to the empirical value of $\mathbf{M}p$ computed from $\mathcal{X}$. To reconstruct the density, one should therefore be able to compute "easily" the value of $\mathbf{M}f$ for any $f \in \Sigma_k^+(\mathcal{P})$ (or for any $f \in \mathcal{P}$, which is equivalent). "Easily" essentially means one must have a closed-form expression of $\mathbf{M}f$.

Suppose the operator $\mathbf{M}$ transforms a function into a compressed representation of dimension m. Operator $\mathbf{M}$ can be seen as the concatenation of m linear forms $\mathbf{M}_1, \ldots, \mathbf{M}_m$. Since we made the simplifying assumption that p belonged to $\Sigma_k^+(\mathcal{P}) \subset \langle \mathcal{P} \rangle$, where $\langle \mathcal{P} \rangle$ denotes the complex span of $\mathcal{P}$, one only needs to define the linear forms $\mathbf{M}_j$ on $\langle \mathcal{P} \rangle$.

Considering the complex span instead of the real positive span will allow us to simplify the expressions of the linear forms we consider in the following, which are based on Fourier transform.

These linear forms must satisfy the two above conditions. Simple linear forms on $\langle \mathcal{P} \rangle$ can be defined as

$$\mathbf{M}_g : f \mapsto \int_{\mathbb{R}^n} fg \; d\mu, \tag{2.12}$$

where g is a bounded measurable function on $\mathbb{R}^n$. In particular, the required conditions are easily interpreted for $\mathbf{M}_g$:

- *Estimation of* $\mathbf{M}_g p$*:* Since p is a probability density on $\mathbb{R}^n$,

$$\mathbf{M}_g p = \int_{\mathbb{R}^n} gp \; d\mu = \mathbb{E}[g(X)], \tag{2.13}$$

  where $\mathbb{E}[\cdot]$ is the expectation with respect to the probability law of density p. Therefore, the value of $\mathbf{M}_g p$ can be approximated by the empirical estimate

$$\hat{\mathbf{M}}_g(\mathcal{X}) = \frac{1}{L} \sum_{r=1}^{L} g(\mathbf{x}_r) = \hat{\mathbb{E}}[g(X)], \tag{2.14}$$

  where $\hat{\mathbb{E}}[\cdot]$ is the expectation with respect to the empirical distribution with L equal masses $\frac{1}{L}$ at each vector of $\mathcal{X} = \{\mathbf{x}_1, \ldots, \mathbf{x}_L\}$. For such a choice of linear forms, it is therefore possible to estimate $\mathbf{M}_g p$ for virtually any g. Moreover, concentration of measure results such as Hoeffding's inequality or McDiarmid's inequality provide confidence intervals on the estimation error.

- *Computation of* $\mathbf{M}\mathcal{P}$*:* The functions g should be chosen so that the value of $\mathbf{M}_g p_{\boldsymbol{\theta}}$ is computable in closed form for any $\boldsymbol{\theta} \in \Theta$.

A compression scheme analog to compressed sensing will consist in considering a family $\mathcal{G}$ of functions so that $g.p_{\boldsymbol{\theta}}$ is integrable for any $\theta \in \Theta$ and $g \in \mathcal{G}$. Having defined a probability distribution on the family $\mathcal{G}$, one will be able to randomly choose a compressive operator $\mathbf{M}$ by drawing *i.i.d.* m functions $g_1, \ldots, g_m \in \mathcal{G}$ and defining $\mathbf{M} = (\mathbf{M}_{g_1}, \ldots, \mathbf{M}_{g_m})$.

2.3.2 Proposed instantiation: isotropic Gaussians.

Let's now propose a particular instantiation of this framework which we will use in the rest of this chapter. Let's consider $\sigma \in \mathbb{R}_+^*$, which will be fixed for the rest of the chapter. Given $n \in \ltimes^*$, let's define the considered family of densities as

$$\mathcal{P}_n = \left\{ p_{\boldsymbol{\mu}} : \mathbf{x} \mapsto \frac{1}{(2\pi)^{\frac{n}{2}} \sigma^n} \exp\left(-\frac{\|\mathbf{x} - \boldsymbol{\mu}\|_2^2}{2\sigma^2} \right), \boldsymbol{\mu} \in \mathbb{R}^n \right\}. \tag{2.15}$$

This family contains all isotropic Gaussians of $\mathbb{R}^n$ with variance σ^2, indexed by their n-dimensional mean $\boldsymbol{\mu}$, which uniquely characterizes them. As before, we define $\Sigma_k^+(\mathcal{P}_n)$ as the linear combinations of k (or less) functions of $\mathcal{P}_n$ and we adopt for the rest of the chapter the simplifying notation $\Sigma_k^+(\mathcal{P}_n) = \Sigma_{k,n}^+$.

Natural linear forms associated to this type of functions are Fourier measurements. Each Fourier measurement can be indexed by a frequency vector $\boldsymbol{\omega} \in \mathbb{R}^n$ and the corresponding function g can be defined as $g_\omega(\mathbf{x}) = \exp\left(-i\langle \mathbf{x}, \boldsymbol{\omega} \rangle\right)$. The corresponding linear form, denoted as $\mathbf{M}_{\boldsymbol{\omega}}$, is therefore:

$$\mathbf{M}_{\boldsymbol{\omega}} : q \mapsto \int_{\mathbb{R}^n} q(\mathbf{x}) e^{-i\langle \mathbf{x}, \boldsymbol{\omega} \rangle} d\mathbf{x}. \tag{2.16}$$

Given data $\mathcal{X} = \{\mathbf{x}_1, \ldots, \mathbf{x}_L\}$, the empirical counterpart of this linear form is

$$\hat{\mathbf{M}}_{\boldsymbol{\omega}}(\mathcal{X}) = \frac{1}{L}\sum_{r=1}^{L} \exp\left(-i\langle x_r, \boldsymbol{\omega}\rangle\right). \tag{2.17}$$

A compressive operator can therefore be defined by choosing m frequencies $\boldsymbol{\omega}_1, \ldots, \boldsymbol{\omega}_m$ and posing $\mathbf{M} = (\mathbf{M}_{\boldsymbol{\omega}_1}, \ldots, \mathbf{M}_{\boldsymbol{\omega}_m})$. These frequencies will typically be randomly drawn, so that for a probability density $f \in \langle \mathcal{P}_n \rangle$, $\mathbf{M}f$ can be interpreted as the sampling of the characteristic function of f at m random frequencies. We will make precise the random choice of the frequencies we retained in section 2.5.

The value of $\mathbf{M}_{\boldsymbol{\omega}} p_{\boldsymbol{\mu}}$ is explicitly computable: one has

$$\mathbf{M}_{\boldsymbol{\omega}} p_{\boldsymbol{\mu}} = \exp\left(-\frac{\sigma^2}{2}\|\boldsymbol{\omega}\|_2^2\right) \exp\left(-i\langle \boldsymbol{\mu}, \boldsymbol{\omega}\rangle\right). \tag{2.18}$$

2.3.3 Injectivity of the compressive operator.

Let's now prove that for some deterministic choices of less than $8k^3n$ frequencies, the corresponding sketching operator $\mathbf{M}$ is injective on $\Sigma_{k,n}^+$, and even on $\Sigma_{k,n}$, which is the set of all linear combinations of k densities of $\mathcal{P}_n$ (non-necessarily with positive coefficients).

Even though the frequencies we will consider in our experiments will not be deterministically chosen (and will be far less numerous), the following results prove that reconstructing a function of $\Sigma_{k,n}^+$ from a finite number of Fourier measurements is a theoretically well-posed problem provided the frequencies are well-chosen. The proofs of the following theorems can be found in Appendix A.

In dimension 1. We first consider the case where $n = 1$, from which we can deduce the generalization for larger dimensions. In this case, the injectivity can be obtained with $4k^2$ well-chosen frequencies (which is indeed inferior to the number of $8k^3$ announced earlier):

Theorem 1. *Let $\omega_1, \ldots, \omega_{2k} \in \mathbb{R}\backslash\{0\}$ be such that $\forall p \neq q$, $\frac{\omega_p}{\omega_q} \notin \mathbb{Q}$. Let's define $\mathbf{M}$ be the linear operator on $L^1(\mathbb{R})$ which performs a Fourier sampling at the following $4k^2$ frequencies:*

- *$2k$ first multiples of ω_1: $\omega_1, 2\omega_1, \ldots, 2k\omega_1$,*
- *$2k$ first multiples of ω_2: $\omega_2, 2\omega_2, \ldots, 2k\omega_2$,*
- $\vdots$
- *$2k$ first multiples of ω_{2k}: $\omega_{2k}, 2\omega_{2k}, \ldots, 2k\omega_{2k}$.*

Then $\mathbf{M}$ is injective on $\Sigma_{k,1}$, and in particular on $\Sigma_{k,1}^+$.

Let's note that for any k, there exists $2k$ frequencies ω_s satisfying the condition $\frac{\omega_p}{\omega_q} \neq \mathbb{Q}$ for all $p \neq q$. This comes from the fact that $\mathbb{R}$ is a vector space of infinite dimension over $\mathbb{Q}$ (if it were finite-dimensional, $\mathbb{R}$ would be isomorphic to $\mathbb{Q}^L$ as a $\mathbb{Q}$ vector space for a certain integer L, and thus would be countable).

In dimension $n > 1$. For $n > 1$, let's keep the choice of $\omega_1, \dots, \omega_{2k}$ of Theorem 1. The following theorem states that there exists $m \leq 2kn$ vectors $(\mathbf{u}_j)_{j=1}^m$ of $\mathbb{R}^n$ such that taking the measurements at frequencies $\omega_s \mathbf{u}_j$ with $(s, j) \in [\![1, 2k]\!] \times [\![1, m]\!]$ yields injectivity on $\Sigma_{k,n}$. The total number of measurements is therefore $4k^2 m \leq 8k^3 n$.

Theorem 2. *Let $n > 1$ and $\omega_1, \dots, \omega_{2k} \in \mathbb{R}\backslash\{0\}$ be such that $\forall p \neq q$, $\frac{\omega_p}{\omega_q} \notin \mathbb{Q}$. There exists $m \leq 2kn$ vectors $(\mathbf{u}_j)_{j=1}^m$ of $\mathbb{R}^n$ such that if $\mathbf{M}$ is the linear operator on $L^1(\mathbb{R}^n)$ which performs a Fourier sampling at frequencies $\omega_s \mathbf{u}_j$ with $(s, j) \in [\![1, 2k]\!] \times [\![1, m]\!]$, then $\mathbf{M}$ is injective on $\Sigma_{k,n}$, and in particular on $\Sigma_{k,n}^+$.*

2.3.4 Recovery problem formulation.

Given the data $\mathcal{X}$, we will denote $\hat{\mathbf{z}}$ the empirical sketch of $\mathcal{X}$, which is the m-dimensional vector $\hat{\mathbf{z}} = (\hat{\mathbf{M}}_{\boldsymbol{\omega}_1}(\mathcal{X}), \dots, \hat{\mathbf{M}}_{\boldsymbol{\omega}_m}(\mathcal{X}))$. One can express the recovery of the initial density p from $\hat{\mathbf{z}}$ as the following minimization problem:

$$\hat{p} = \operatorname*{argmin}_{q \in \Sigma_k} \frac{1}{2} \|\hat{\mathbf{z}} - \mathbf{M}q\|_2^2. \tag{2.19}$$

Despite being nonconvex, this kind of formulation of the problem is addressed in a regular compressed sensing setting by greedy algorithms. In the next section, we will derive from such a standard method an algorithm aimed at solving (2.19).

2.4 Compressive reconstruction algorithm

To address the estimation problem (2.19), we propose an algorithm analogous to Iterative Hard Thresholding (IHT) (Blumensath and Davies, 2009a).

2.4.1 Reminder of Iterative Hard Thresholding

IHT is a standard greedy method aimed at solving sparse inverse problems, as mentioned in Section 1.1.2. Consider a k-sparse signal $\mathbf{x}$ of dimension n and a measurement matrix $\mathbf{M}$ of size $m \times n$ (with $m < n$). Denoting $\mathbf{y} = \mathbf{M}\mathbf{x}$ the measurement of $\mathbf{x}$, IHT considers the minimization

$$\operatorname*{argmin}_{\mathbf{z} \in \Sigma_k} \|\mathbf{y} - \mathbf{M}\mathbf{z}\|_2^2, \tag{2.20}$$

where Σ_k is this time the set of k-sparse vectors of $\mathbb{R}^n$. At each iteration, IHT updates an estimate $\hat{\mathbf{x}}$ of $\mathbf{x}$, decreasing the objective function $\phi : \hat{\mathbf{x}} \mapsto \frac{1}{2}\|\mathbf{y} - \mathbf{M}\hat{\mathbf{x}}\|_2^2$ while ensuring the k-sparsity of $\hat{\mathbf{x}}$. The quantity $\mathbf{r} = \mathbf{y} - \mathbf{M}\hat{\mathbf{x}}$ is named the *residual*. The update step is performed in two steps:

1. The n-dimensional gradient of ϕ at current iterate $\hat{\mathbf{x}}$, noted $\nabla\phi$, is computed.
2. The update is given by $\hat{\mathbf{x}} \leftarrow H_k(\hat{\mathbf{x}} - \lambda\nabla\phi)$, where λ is a descent step and H_k is a hard thresholding operator which keeps only the k entries of the vector with largest module and sets the others to 0.

This algorithm has been proven to converge to the global minimum of (2.20) under a RIP assumption on $\mathbf{M}$ (Blumensath and Davies, 2009a). We now adapt this scheme of work to our reconstruction problem (2.19).

2.4.2 Proposed continuous case algorithm

We also adopt an iterative greedy method to perform the reconstruction of p. We therefore iteratively update an estimate $\hat{p}$, which is parametrized by a vector $\hat{\mathbf{a}} \in \mathbb{R}^k$ of positive weights and by the support $\hat{\Gamma} = \{\hat{\boldsymbol{\mu}}_1, \dots, \hat{\boldsymbol{\mu}}_k\} \subset \mathbb{R}^n$ corresponding to the means of the current estimated Gaussians. The current residual is defined by $\hat{\mathbf{r}} = \hat{\mathbf{z}} - \mathbf{M}\hat{p}$. In our case, the function ϕ takes $\hat{p}$ as an argument and is defined as

$$\phi(\hat{p}) = \frac{1}{2}\|\hat{\mathbf{z}} - \mathbf{M}\hat{p}\|_2^2. \tag{2.21}$$

There are some differences between our density estimation problem and the problem addressed by IHT which require modifications of the procedure. They are explained in the following sections. The algorithm is then more precisely described.

The "gradient": continuous version. In IHT, the signal one wants to reconstruct is supposed to be sparse in a finite basis of vectors. The gradient computed in the first step is a finite-dimensional vector, and each entry corresponds to the infinitesimal shift of the objective function ϕ when a certain entry of the vector is shifted.

In our case, the density is supposed to be sparse in the infinite basis $\mathcal{P}_\sigma$, which is parametrized by $\mathbb{R}^n$. The "canonical" directions in which $\hat{p}$ can be shifted are therefore also parametrized by $\mathbb{R}^n$, and the "gradient" is the collection of these possible shifts, which are noted $\nabla_{\boldsymbol{\mu}}\phi$ for all $\boldsymbol{\mu} \in \mathbb{R}^n$ and defined as follows:

$$\nabla_{\boldsymbol{\mu}}\phi(\hat{p}) = \left(\frac{\partial}{\partial t}\frac{1}{2}\|\hat{\mathbf{z}} - \mathbf{M}(\hat{p} + t p_{\boldsymbol{\mu}})\|_2^2\right)_{t=0} = -\langle \mathbf{M}p_{\boldsymbol{\mu}}, \hat{\mathbf{r}}\rangle. \tag{2.22}$$

Again, this quantity represents the local variation of the objective function (2.19) when an infinitesimal fraction of the density $p_{\boldsymbol{\mu}}$ is added to the current estimate. Since we cannot compute these values for every $\boldsymbol{\mu} \in \mathbb{R}^n$, we must only choose a finite number of $\boldsymbol{\mu}$ for which we will compute $\nabla_{\boldsymbol{\mu}}\phi$.

Since we aim at decreasing ϕ, these directions should be chosen so that $\nabla_{\boldsymbol{\mu}}\phi(\hat{p})$ is negatively minimal, so that $p_{\boldsymbol{\mu}}$ is a seemingly good candidate to be added to the current estimate $\hat{p}$. Therefore, we seek instead a certain number M of local minima of $\boldsymbol{\mu} \mapsto \nabla_{\boldsymbol{\mu}}\phi(\hat{p})$, which will typically be chosen as $\mathcal{O}(k)$. These local minima parametrize elements of $\mathcal{P}$ which are the best correlated elements to the residual $\hat{\mathbf{r}}$. They are the best directions in which to "move" locally the estimate $\hat{p}$ in order to decrease the objective function ϕ. These local minima are searched for by a randomly initialized minimization algorithm. When they are found, they are added to the current support $\hat{\Gamma}$, increasing its size up to $M + k$ elements.

Hard Thresholding. The second step in IHT consists in choosing a descent step used to shift the current estimate in the direction of the gradient, and enforcing sparsity through hard thresholding. In our case, we have at this point an updated collection of candidate means $\hat{\Gamma}$, and we want to keep only k of these means.

In order to do this, we aim at decomposing the empirical sketch $\hat{\mathbf{z}}$ as a positive linear combination of vectors in $\mathbf{M}p_{\boldsymbol{\nu}}$, with $\boldsymbol{\nu} \in \hat{\Gamma}$, that is to project the sketch $\hat{\mathbf{z}}$ on the cone generated by the sketches of the functions $\{p_{\boldsymbol{\nu}} : \boldsymbol{\nu} \in \hat{\Gamma}\}$. This cone can be noted $C(\mathbf{M}\hat{\Gamma})$ and defined by

$$C(\mathbf{M}\hat{\Gamma}) = \left\{ \sum_{j=1}^{K} \lambda_j \mathbf{M}p_{\boldsymbol{\nu}_j} : N > 0, \lambda_j \geq 0, \boldsymbol{\nu}_j \in \hat{\Gamma} \right\}. \tag{2.23}$$

The aforementioned projection of $\hat{\mathbf{z}}$ on $C(\mathbf{M}\hat{\Gamma})$ is expressed as the following minimization problem, supposing $\hat{\Gamma}$ contains K vectors $\boldsymbol{\nu}_1, \ldots, \boldsymbol{\nu}_K$:

$$\underset{\boldsymbol{\beta} \in \mathbb{R}_+^K}{\operatorname{argmin}} \|\hat{\mathbf{z}} - \mathbf{N}\boldsymbol{\beta}\|_2, \tag{2.24}$$

where $\mathbf{N}$ is the concatenation of the sketches of the functions parametrized by $\hat{\Gamma}$, that is

$$\mathbf{N} = \left[\mathbf{M}p_{\boldsymbol{\nu}_1} \ \ldots \ \mathbf{M}p_{\boldsymbol{\nu}_K}\right]. \tag{2.25}$$

The hard thresholding step is then performed by keeping the k largest coefficients and the k corresponding parameters of $\hat{\Gamma}$ found in (2.24). Note that in the framework we consider (isotropic Gaussians with Fourier measurements), the quantity $\|\mathbf{M}p_{\boldsymbol{\nu}}\|_2$ do not depend on $\boldsymbol{\nu}$, that is the sketches all have the same energy. In the case where they do not have the same energy, one should keep the k coefficients such that $\|\boldsymbol{\beta}_j \mathbf{M}p_{\boldsymbol{\nu}_j}\|_2$ is maximal.

Gradient descent step. In IHT, an iteration stops when hard thresholding is performed. In our case, we can still perform an additional step, which consists in decreasing further the objective function φ.

At this point in the iteration, $\hat{p}$ is defined as

$$\sum_{s=1}^{k} \hat{\alpha}_s p_{\hat{\boldsymbol{\mu}}_s}, \tag{2.26}$$

where the parameters $\hat{alpha}_s$ and $\hat{\boldsymbol{\mu}}_s$ hopefully estimate the real parameters α_s and μ_s of p.

Since the family $\mathcal{P}_\sigma$ is extremely coherent, the local minima we found in the previous steps may be shifted from the true mean vectors because of the imprecision induced by the other components of the mixture. This imprecision on the $\boldsymbol{\mu}_s$ obviously also imply imprecision on the α_s. However, there may exist a better estimate for p in the vicinity of $\hat{p}$.

To find it, we simply consider φ as a slightly different version of ϕ: φ represents the same cost function, but takes the parameters $\alpha_1, \ldots, \alpha_k$ and $\boldsymbol{\mu}_1, \boldsymbol{\mu}_k$ as arguments. Therefore, it is defined as:

$$\begin{array}{rccl} \varphi: & \mathbb{R}^k \times (\mathbb{R}^n)^k & \to & \mathbb{R} \\ & (\boldsymbol{\alpha}, \boldsymbol{\mu}_1, \ldots, \boldsymbol{\mu}_k) & \mapsto & \frac{1}{2}\|\hat{\mathbf{z}} - [\mathbf{M}p_{\boldsymbol{\mu}_1} \ \ldots \ \mathbf{M}p_{\boldsymbol{\mu}_k}]\boldsymbol{\alpha}\|_2^2. \end{array} \tag{2.27}$$

Initializing the parameters to the current estimators $\hat{\boldsymbol{\alpha}}$ and $\hat{\boldsymbol{\mu}}_1, \ldots, \hat{\boldsymbol{\mu}}_k$, we can apply a gradient descent algorithm on φ to find, in the vicinity of $\hat{p}$, a better estimate in the sense that it has a smaller image by ϕ.

Algorithmic scheme. Algorithm 1 summarizes the overall procedure. It mainly consists of three steps by iteration:

1. M local minima of $\boldsymbol{\mu} \mapsto \nabla_{\boldsymbol{\mu}}(\hat{p})$ are sought with a gradient descent algorithm with random initialization and are added to the current support $\hat{\Gamma}$.
2. The sketch $\hat{\mathbf{z}}$ is projected on $\mathbf{M}\hat{\Gamma}$ with a positivity constraint on the coefficients. Only the k highest coefficients and the corresponding vectors of the support are kept.
3. A gradient descent algorithm is applied to further decrease the objective function with respect to the weights and support vectors.

Algorithms 2, 3, 4 and 5 describe the subfunctions.

Algorithm 1 Compressive isotropic Gaussian mixture parameter estimation

Input: Sketch $\hat{\mathbf{z}}$, operator $\mathbf{M}$, target sparsity k, integer M.
Initialize $\hat{\Gamma} = \emptyset, \hat{\mathbf{r}} = \hat{\mathbf{z}}$.
repeat
  Set $\boldsymbol{\nu}_1, \ldots, \boldsymbol{\nu}_M \leftarrow \text{Find\_min}(\mathbf{M}, \hat{\mathbf{r}}, M)$.
  Set $\hat{\Gamma}' \leftarrow \hat{\Gamma} \cup \{\boldsymbol{\nu}_1, \ldots, \boldsymbol{\nu}_M\}$.
  Set $\hat{\mathbf{a}}' \leftarrow \text{Proj\_cone}(\hat{\mathbf{z}}, \hat{\Gamma}')$.
  Set $\hat{\mathbf{a}}, \hat{\Gamma} \leftarrow \text{Hard\_threshold}(\hat{\mathbf{a}}', \hat{\Gamma}', k)$.
  Set $\hat{\mathbf{a}}, \hat{\Gamma} \leftarrow \text{Shift\_support}(\hat{\mathbf{z}}, \hat{\mathbf{a}}, \hat{\Gamma})$.
  Set $\hat{\mathbf{r}} \leftarrow \hat{\mathbf{z}} - \sum_{j=1}^{k} \hat{\alpha}_j \mathbf{M} p_{\hat{\boldsymbol{\mu}}_j}$.
until Stopping criterion is satisfied
Return $\hat{\mathbf{a}}$, $\hat{\Gamma}$.

Algorithm 2 Find_min($\mathbf{M}, \hat{\mathbf{r}}, M$)

For $i = 1$ to M
  Find a local minimum $\boldsymbol{\nu}_i$ of the function:
  $\boldsymbol{\nu} \in \mathbb{R}^n \quad \mapsto \quad -\langle \mathbf{M} p_{\boldsymbol{\nu}}, \hat{\mathbf{r}} \rangle$
  with a gradient descent algorithm, initialized randomly.
End For
Return $\boldsymbol{\nu}_1, \ldots, \boldsymbol{\nu}_M$.

2.4.3 Memory usage

Let's now estimate the order of magnitude of the memory required by the compressive algorithm to estimate p from $\hat{\mathbf{z}}$. Let's consider that n, k and m are much larger than 1. If we suppose that optimization algorithms only use first-order quantities, their memory costs are dominated by $\mathcal{O}(kn)$. The computation of the cost function of Algorithm 5 requires $\mathcal{O}(km)$. The storage of the operator $\mathbf{M}$ (via the frequencies $\boldsymbol{\omega}_j$) requires $\mathcal{O}(mn)$.

Therefore, the total memory usage is $\mathcal{O}((k+n)m + kn)$ and does not depend on the number L of vectors. In comparison, the memory requirement of EM is $\mathcal{O}(L(k+n))$ to store both the vectors and their probabilities to belong to each current component of the mixture. The compressed algorithm allows memory savings as soon as $m + \frac{kn}{k+n} \lesssim L$. Since $kn \lesssim m$, this condition is nearly equivalent to $m \lesssim L$.

This suggests that one will be able to make memory savings if the number of vectors in the training set $\mathcal{X}$ is larger than the size of the sketch required to performed the reconstruction.

2.4.4 Computational complexity

Computational complexity is the main drawback of the compressed procedure, since this procedure relies on several optimization steps, which can involve many variables for large k and n.

More precisely, the computational bottleneck is the last step of the iteration where a gradient descent is performed. This optimization procedure involves $k(n+1)$ variables and the cost for computing the function at a certain point is $\mathcal{O}(mk)$. Therefore, since a first order optimization algorithm requires the computation of the gradient for each variable, the complexity of a simple gradient descent implementation is $\mathcal{O}(mk^2(n+1))$. Moreover, the cost of computing the sketch from the training data is $\mathcal{O}(mL)$, and must be taken into account unless the data is streamed so that the sketch can be computed "on the fly" before

Algorithm 3 Proj_cone($\mathbf{v}, \Gamma = \{\boldsymbol{\nu}_1 \dots \boldsymbol{\nu}_K\}$)

Solve the following convex optimization problem:

$\mathbf{a} = \underset{\boldsymbol{\beta} \in \mathbb{R}_+^K}{\operatorname{argmin}} ||\mathbf{v} - \mathbf{N}\boldsymbol{\beta}||_2$, with $\mathbf{N} = [\mathbf{M}p_{\boldsymbol{\nu}_1}, \dots, \mathbf{M}p_{\boldsymbol{\nu}_K}]$.

Return $\mathbf{a}$.

Algorithm 4 Hard_threshold($\mathbf{a}, \Gamma = \{\mathbf{u}_1, \dots, \mathbf{u}_K\}, k$)

Let $a_{i_1}, \dots, a_{i_k}$ be the k highest entries of $\mathbf{a}$.

Return $(a_{i_1}, \dots, a_{i_k}), \{\mathbf{u}_{i_1}, \dots, \mathbf{u}_{i_k}\}$.

Algorithm 5 Shift_support($\hat{\mathbf{z}}, \mathbf{a}, \Gamma = \{\boldsymbol{\mu}_1, \dots, \boldsymbol{\mu}_k\}$)

Find a local minimum $(\mathbf{a}', \boldsymbol{\mu}'_1, \dots, \boldsymbol{\mu}'_k)$ of the function:

$$\begin{array}{rcl} \mathbb{R}^k \times (\mathbb{R}^n)^k & \to & \mathbb{R}_+ \\ (\mathbf{b}, \boldsymbol{\nu}_1, \dots, \boldsymbol{\nu}_k) & \mapsto & \|\hat{\mathbf{z}} - [\mathbf{M}p_{\boldsymbol{\nu}_1}, \dots, \mathbf{M}p_{\boldsymbol{\nu}_k}]\mathbf{b}\|_2, \end{array}$$

using a gradient descent algorithm initialized at $\mathbf{a}, \boldsymbol{\mu}_1, \dots, \boldsymbol{\mu}_k$.

Return $\mathbf{a}', \{\boldsymbol{\mu}'_1, \dots, \boldsymbol{\mu}'_k\}$.

the estimation procedure *per se*. The overall cost is virtually $\mathcal{O}(mL + mk^2(n+1)n_{comp})$, where n_{comp} is the number of iterations performed in the compressed estimation procedure.

This must be compared to a standard EM algorithm, which has complexity $\mathcal{O}(kLn_{EM})$, where n_{EM} is again the number of iterations performed in the EM algorithm. In the case the sketch is not computed on the fly, there will be a gain in complexity in the compressed case only if $m \ll kn_{EM}$. This was not the case in our experiments (described in the next section), since the EM algorithm converged quickly enough so that $m \sim kn_{EM}$. We will further discuss the computational outlooks in Section 2.6.

2.5 Experiments

2.5.1 Experimental setup

To evaluate the behavior of the compressive reconstruction algorithm, we conducted experiments on vectors drawn from a mixture of k isotropic Gaussians with identity covariance matrices ($\sigma = 1$). In each case, we drew weights uniformly on the simplex[a], and we chose the Gaussian means $\boldsymbol{\mu}_j$ by drawing random vectors, each entry being drawn from a probability law of density $\mathcal{N}(0, 1)$.

The experiments were performed in the following way: after the choice of the probability distribution p, we drew L random vectors from this probability distribution and computed the empirical sketch of the distribution in one pass of the data. The training samples were then discarded from hard memory. We chose the sketching operator $\mathbf{M}$ randomly, following the scheme described in Section 2.5.2. We then applied the reconstruction algorithm to the empirical sketch $\hat{\mathbf{z}}$ to get an approximated mixture $\hat{p}$. The random initialization for the reconstruction algorithm is detailed in section 2.5.3.

To evaluate the quality of the estimation, we relied on two usual discrepancy measures between probability density functions. They were used to quantify the difference between the true mixture p and the estimated mixture $\hat{p}$. The two considered quantities are defined

[a] We also performed experiments where all the weights were equal to $\frac{1}{k}$ and this didn't alter the conclusions drawn from the experiments.

by integrals, which in our case could not be computed explicitly. Therefore, we approximated the integrals by empirical means: we drew $N = 10^5$ points $(\mathbf{y}_i)_{i=1}^N$ *i.i.d.* from p and computed the empirical estimates described below. The two chosen measures were:

- *Kullback-Leibler (KL) divergence:* A symmetrized version of KL divergence can be defined as

$$D_{KL}(p, \hat{p}) = \int_{\mathbb{R}^n} \left[\ln\left(\frac{p(\mathbf{x})}{\hat{p}(\mathbf{x})}\right) p(\mathbf{x}) + \ln\left(\frac{\hat{p}(\mathbf{x})}{p(\mathbf{x})}\right) \hat{p}(\mathbf{x}) \right] d\mathbf{x}. \tag{2.28}$$

  The empirical estimate we considered is defined as

$$\hat{D}_{KL}(p, \hat{p}) = \frac{1}{N} \sum_{r=1}^{N} \left[\ln\left(\frac{p(\mathbf{y}_r)}{\hat{p}(\mathbf{y}_r)}\right) + \frac{\hat{p}(\mathbf{y}_r)}{p(\mathbf{y}_r)} \ln\left(\frac{\hat{p}(\mathbf{y}_r)}{p(\mathbf{y}_r)}\right) \right]. \tag{2.29}$$

  The KL divergence ranges from 0 to $+\infty$, lower values meaning closer distributions.

- *Hellinger distance:* The Hellinger distance can be defined as

$$D_H(p, \hat{p}) = 1 - \int_{\mathbb{R}^n} \sqrt{p(\mathbf{x})\hat{p}(\mathbf{x})} d\mathbf{x}. \tag{2.30}$$

  The empirical estimate we considered is defined as

$$\hat{D}_H(p, \hat{p}) = 1 - \frac{1}{N} \sum_{r=1}^{N} \sqrt{\frac{\hat{p}(\mathbf{y}_r)}{p(\mathbf{y}_r)}}. \tag{2.31}$$

  The Hellinger distance ranges from 0 to 1. Here again, lower values mean closer distributions.

2.5.2 Choice of the frequencies

Let's now describe the heuristic we considered to randomly choose the compressive operator $\mathbf{M}$. Let's recall that $p \in \Sigma_k$, so that $p = \sum_{s=1}^k \alpha_s p_{\boldsymbol{\mu}_s}$, with the α_s positive which sum to 1. Denoting by $\mathcal{F}(f).\boldsymbol{\omega}$ the Fourier transform of a function f taken at frequency $\boldsymbol{\omega}$, we have

$$\begin{aligned} |\mathcal{F}(p).\boldsymbol{\omega}| &= \left| \sum_{s=1}^k \alpha_s p_{\boldsymbol{\mu}_s} \right| \leq \sum_{s=1}^k \alpha_s |\mathcal{F}(p_{\boldsymbol{\mu}_s}).\boldsymbol{\omega}| \\ &= \sum_{s=1}^k \alpha_s \exp\left(-\frac{\sigma^2}{2}\|\boldsymbol{\omega}\|_2^2\right) = \exp\left(-\frac{\sigma^2}{2}\|\boldsymbol{\omega}\|_2^2\right). \end{aligned} \tag{2.32}$$

This upper bound on the value of $\mathcal{F}(p)$ gives a hint on the way to design the random choice of frequencies. Indeed, we want to sample frequencies which are likely to be "energetic", so that $|\mathcal{F}(p).\boldsymbol{\omega}|$ is not "too low". Designing a random choice with a density proportional to the upper bound found in (2.32) seems like a reasonable choice. Since $\sigma = 1$, the random sampling we chose for the frequencies $\boldsymbol{\omega}_j$ is from a probability law of density $\mathcal{N}(0, \mathbf{Id})$.

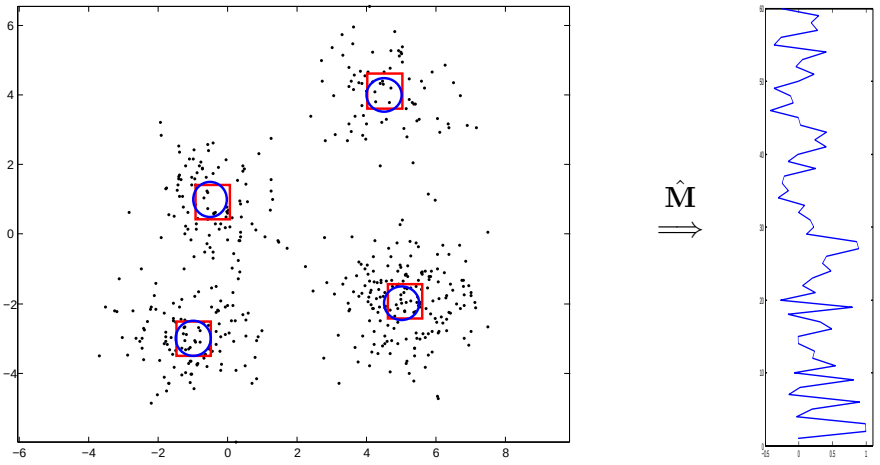

Figure 2.2: Real and reconstructed centroids (respectively represented as circles and squares) of 4 Gaussians in dimension 2 from 10^3 points drawn from the mixture. To estimate the 12 real parameters of the mixture, the data was compressed to a complex-valued sketch of dimension 30, represented to the right as a 60-dimensional real signal.

2.5.3 Heuristic for random initialization

The search for local minima in Algorithm 2 was initialized randomly by exploiting a measure performed during the construction of the sketch: during the single pass on the data, the norms of the vectors $\mathbf{x}_r$ are computed and the maximum of the norms, $R = \max_{\mathbf{x}\in\mathcal{X}} ||\mathbf{x}||_2$, is computed. These calculations have a negligible impact on the computation time of the sketch, and on its size (it only adds one component which "completes" the sketch). The knowledge of R allows us to delimit a ball in which the centers of the Gaussians are very probably contained.

We performed the random initialization by drawing a direction uniformly on the unit sphere and multiplying this unit vector by a scalar uniformly drawn in $[0; R]$.

2.5.4 Results

Figure 2.2 visually illustrates the behavior of the algorithm on a simple mixture of 4 Gaussians in dimension 2. $L = 10^3$ points were drawn from this mixture and used to compute a $m = 30$-dimensional sketch. As shown in the figure, the mixture parameters are precisely estimated without referring to the initial data. The symmetric KL divergence and Hellinger distance are respectively 0.026 and 0.003.

Figure 2.3 illustrates the reconstruction quality of our algorithm in dimension 10 for different values of mixture components k and sketch sizes m in terms of Hellinger distance. For each sketch size m ranging from 200 to 2000 with step size 200, k was chosen to range from $m/200$ to $m/10$ with step $m/200$. For each choice of parameters, 10 experiments were performed and the depicted value is the Hellinger distance such that 80% of the experiments lead to a smaller Hellinger distance. We can essentially observe that the Hellinger distance gradually decreases as the number of mixture components rises. For the considered parameters range, choosing $m = 10kn$, *i.e.*, choosing m so that it contains 10 times more values than the number of parameters to estimate, leads to a Hellinger distance smaller than 0.03 for 80% of the cases. Note that the number of measurements we took for the experiments is way below the deterministic number of $8k^3n$ provided in

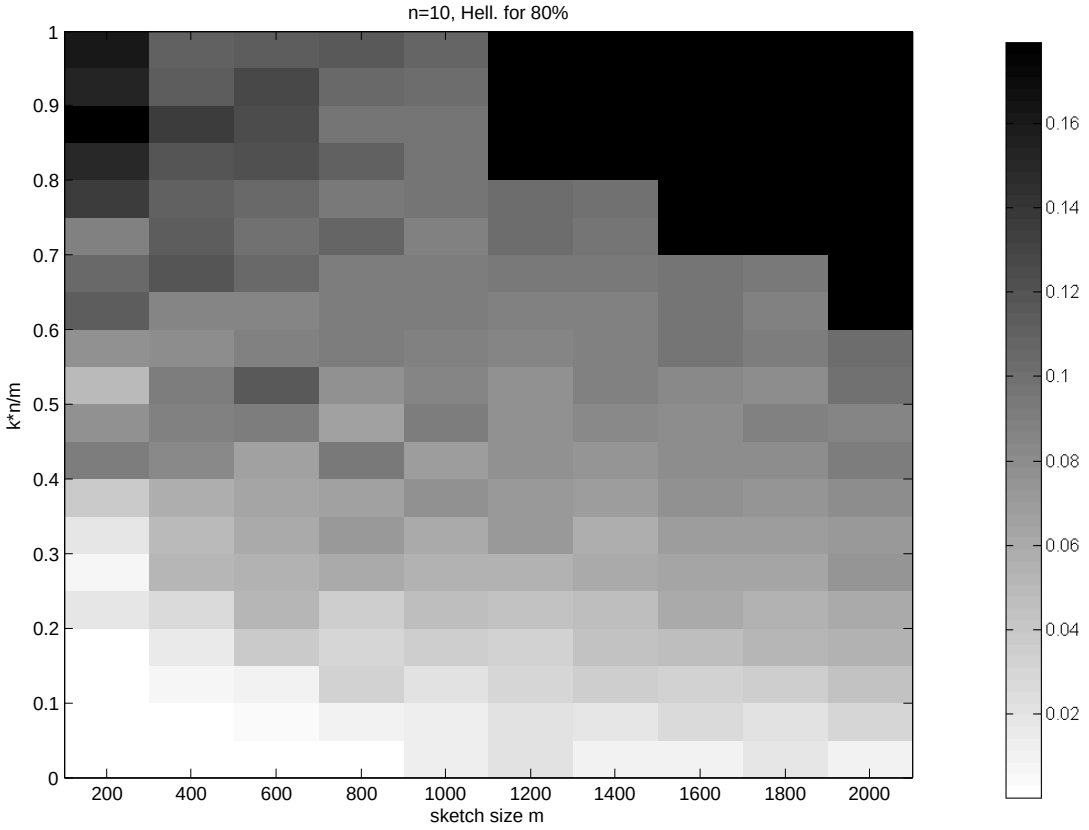

Figure 2.3: Quality of reconstruction in dimension $n = 10$, with $N = 10^4$ points, measured as a Hellinger distance. Each square corresponds to 10 experiments, and the depicted values are the values of the Hellinger distance under which 80% of performed experiments are placed.

Section 2.3.3, suggesting that a random choice of frequencies can be proven as a robust choice allowing the drastic reduction of the number of measurements compared to the proposed deterministic choice.

Table 2.1 compares our algorithm with a standard EM algorithm (Dempster et al., 1977) in the case where $n = 20$, $k = 10$, $m = 1000$ for values of dataset size L ranging from 10^3 to 10^5. For each case, we can see that the precision of the estimation increases with the number of samples. In the compressed case, this can be explained by the fact that the components of the sketch are better estimated with more points. We notice that the memory used for EM is proportional to the number L of samples in the dataset, while the memory required by the compressed algorithm does not depend on this parameter, which leads to a substantial improvement in memory usage for $L \geq 10^4$. Even with this reduced memory cost, the compressed algorithm is able to provide a precision comparable to the precision of the EM algorithm.

2.6 Conclusion and outlooks

In this chapter, we first proposed a review of techniques reminiscent of inverse problems and compressed sensing applied to certain learning tasks, mainly to density estimation. Our contribution consisted in a framework for density mixture estimation, which was instantiated to isotropic Gaussians and random Fourier sampling. We could derive an

L	Compressed			L	EM		
	KL div.	Hell.	Mem.		KL div.	Hell.	Mem.
10^3	0.68 ± 0.28	0.06 ± 0.01	0.6	10^3	0.68 ± 0.44	0.07 ± 0.03	0.24
10^4	0.24 ± 0.31	0.02 ± 0.02	0.6	10^4	0.19 ± 0.21	0.01 ± 0.02	2.4
10^5	0.13 ± 0.15	0.01 ± 0.02	0.6	10^5	0.13 ± 0.21	0.01 ± 0.02	24

Table 2.1: Comparison between our compressed estimation algorithm and an EM algorithm in terms of precision of the estimation and memory usage (in megabytes). Experiments were performed with $n = 20$, $k = 10$, $m = 1000$. In each cell, the value is a median on 10 experiments with the standard deviation for the precision measures.

algorithm which experimentally shows good reconstruction properties with respect to a standard estimation algorithm. Let's now mention some outlooks related to these results.

2.6.1 Extension to richer families of densities

Density mixture estimation using isotropic Gaussians can be seen as a clustering problem. It would be particularly interesting to extend the experimental results to more general families of Gaussians, for instance with diagonal covariance matrices, to allow for variations in the form of the "clusters".

Considering larger families of densities would probably require finer choices for the sketching operator $\mathbf{M}$, in order to be able to separate the compressed representations of all vectors of the enriched family. For this purpose, it seems interesting to investigate multiscale sketches, where coarser frequencies may be used at the beginning of the reconstruction algorithm to avoid getting too many local minima, and higher frequencies can be added throughout the algorithm to approach the solution.

2.6.2 Computational savings

If such a compressive framework has the potential to reduce memory requirements by computing the sketch on the fly with streamed data, the computational complexity of the density reconstruction is still large, especially due to the last stage of the algorithm (Algorithm 5), which is an optimization step involving $(n + 1)k$ variables and which is performed at each iteration in the current state of the algorithm. Algorithmic savings could be made by finding a faster alternative to this step. The cost of computing the sketch may also be reduced by finding better sketching operators or procedures. In particular, adopting a "multiscale" paradigm, where several sketches are computed at different ranges of frequencies may provide faster convergence to an acceptable solution.

Let's also note that the size of the sketch does not depend on the number of training vectors, that is the complexity of the estimation do not increase in the case where the sketch is updated using more training vectors (this does not include the time which is necessary to build the sketch). Moreover, the algorithm should be more precise when the number L of training vectors increases, since the empirical sketch $\hat{\mathbf{z}}$ is in this case a more reliable estimate of the theoretical sketch $\mathbf{M}p$. The general approach described in this chapter may therefore be compared to conceptual results suggesting that for a certain learning task, if an estimation algorithm is well chosen in an abstract collection of algorithms, then the computational complexity required to achieve a given estimation error ratio should *decrease* when the size of the training set increases (Shalev-Shwartz et al., 2012).

2.6.3 Theoretical analysis

We have proved that some deterministic choices of the frequencies yielded the injectivity of the sketching operator $\mathbf{M}$ on Σ_k, so that the reconstruction of a density from a sketch is theoretically well-posed. The next step in studying the well-posedness of such a problem would be a finer study of Σ_k and the action of $\mathbf{M}$ when the frequencies are chosen at random, with a Gaussian distribution such as the distribution chosen for the frequencies.

Investigating the geometry of Σ_k and using "union bound" techniques such as in classical compressed sensing, it seems possible to study the average amount of energy of the function one is able to keep by randomly sampling the Fourier transform. This type of result typically allows deduction of robustness results about the operator $\mathbf{M}$.

In particular, the regularity of Gaussians seems interesting to study this "conservation of energy" and it may be conclusive to study the behavior of $\mathbf{M}$ on the Schwartz space.

Chapter 3

Performance of decoders in linear inverse problems

Note: The main part of this chapter is taken from the preprint *Fundamental performance limits for ideal decoders in high-dimensional linear inverse problems*, which was submitted to *IEEE Transactions on Information Theory* on November 2013. A version of this preprint can be found at: http://arxiv.org/abs/1311.6239.

Inverse problems have been introduced in Chapter 1 in the case where one wants to invert a linear operator applied to a sparse vector in a certain basis. On top of this standard case, inverse problems concern a wide range of other signals: one can be interested in adding hypotheses to the usual sparse model if they have more information than just sparsity about the considered signals or conversely relax some hypotheses on the sparse model to build a model containing more objects. Inverse problems can even be considered for models which are structured differently than the set of sparse vectors.

Models beyond sparsity for inverse problems. Even if the sparse vectors model still attracts a lot of attention, other works on inverse problems or compressed sensing considering more exotic signal models have multiplied over the years (Baraniuk et al., 2010). In these "generalized" models, the signals of interest live in or close to a subset Σ of the space. Such models are more adequate to describe particular type of signals or objects, and some of them are represented in Figure 3.1. A list of various models which have been considered in inverse problem or compressed sensing setting is:

- **Block-sparse signals** (Eldar et al., 2010) are vectors divided into blocks and for which only a few blocks are nonzero. In practice, this model can be used to model signals with a multi-band spectrum (Mishali and Eldar, 2009), or the difference of gene expression levels between two samples (Parvaresh et al., 2008).

- **Dictionary-sparse signals** (Rauhut et al., 2008). Sparse signals models can be considered relatively to a certain basis, but also to a dictionary, with possibly linearly redundant vectors but which represent more adequately the considered signals. In particular, these models can be used for various signal processing tasks such as denoising or deblurring.

- **Cosparse signals** (Nam et al., 2013) are signals which are sparse after undergoing a linear transform Ω, typically dimension-increasing. This framework can model overcomplete transforms of a signal which supposedly yields sparsity, such as the shift invariant wavelet transform (Mallat, 2008) or the finite difference operator for an image.

- **Low-rank matrices** (Recht et al., 2010; Candès and Plan, 2011) often appear when considering data which live near a subspace of limited dimension relatively to the ambient dimension. Moreover, instead of only considering matrices which are approximately low-rank, one can model the signals of interest as the sum of a low-rank matrix and of a sparse matrix (Zhou et al., 2010). In order to circumvent the ambiguity of such a decomposition (a matrix can be low-rank *and* sparse), a nonsparsity constraint on the low-rank matrix can be considered (Candès et al., 2011a).

- **Unions of subspaces** (Blumensath and Davies, 2009b; Blumensath, 2011) can be considered in a general setting, which encompasses a lot of other models. In particular, the usual k-sparse model in dimension n is the union of $\binom{n}{k}$ subspaces corresponding to all the possible supports of a k-sparse vector. Most other sparse models can be considered as finite union of subspaces. Furthermore, some models of low-rank matrices can be viewed as an infinite union of subspaces (Blumensath, 2011).

- **Low-dimensional manifolds** have been considered in a compressed sensing setting (Baraniuk and Wakin, 2006; Eftekhari and Wakin, 2013). Such models can apply to the representation of small variations of an image (Wakin et al., 2005; Wakin, 2009).

- **Symmetric definite positive square matrices with k-sparse inverse**. High-dimensional Gaussian graphical models have a covariance matrix of this type: the numerous pairwise conditional independences that characterize the structure of such models, and make them tractable, translate into zeros entries of the inverse covariance matrix (the concentration matrix). Combining sparsity prior on the concentration matrix with maximum likelihood estimation of covariance from data, permits to learn jointly the structure and the parameters of Gaussian graphical models (so called "covariance selection" problem) (Yuan and Lin, 2007; Yuan, 2010).

- **Low-dimensional embedding** of a point cloud consists in finding a projection of the points onto a low-dimensional subspace while approximately keeping distances between points. This is asymptotically feasible while substantially reducing the dimension, as proved by Johnson-Lindenstrauss lemma (Johnson and Lindenstrauss, 1984). As in compressed sensing, such a projection can be achieved with high probability with usual random drawings of matrices (Achlioptas, 2001).

All these models share a common point: they typically contain far fewer vectors than the whole space. Indeed, these models serve as *a priori* information on the considered signals in order to enable their reconstruction from a non-injective operator $\mathbf{M}$. The well-posedness of an inverse problem thus comes from the information that the signal to reconstruct $\mathbf{x}$ is sufficiently close to a subset Σ of the signal space. For this information to be useful, the model Σ needs to be a substantially small part of the space for the inverse problem to make sense, otherwise the number of solutions of the problem will be too large to have a meaning.

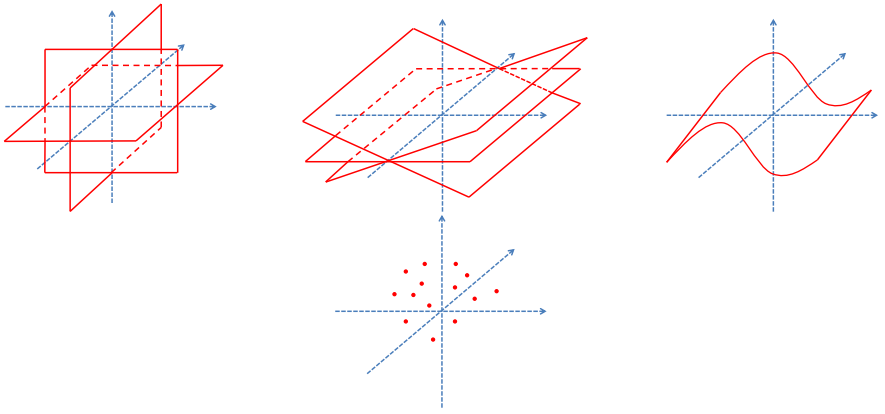

Figure 3.1: Illustration of several CS models. *From left to right:* k-sparse vectors, union of subspaces, smooth manifold and point cloud.

Since these models generalize the sparse model, the following question arises: can they be considered under a general framework, sharing common reconstruction properties? The following discussion precises what type of results would be expected in this generalized framework.

What should we expect for an inverse problem to be well-posed? The well-posedness of an inverse problem is not a universal property. There are numerous ways to define when a problem is well-posed: either theoretically or practically, either uniformly or probabilistically. A common requirement for well-posedness is that $\mathbf{M}$ is injective on Σ, so that the problem is *theoretically* well-posed: if $\mathbf{x} \in \Sigma$, the knowledge of $\mathbf{Mx}$ yields the knowledge of $\mathbf{x}$ since there is one, and one only, element of Σ having the image $\mathbf{Mx}$ by $\mathbf{M}$.

However, this theoretical well-posedness given by the single injectivity on Σ is usually not sufficient practically for several reasons:

- *Stability to the model*: The model Σ do not contain exactly the vectors one is interested in recovering. It is only comprised of approximations of the signals of interest. Therefore, $\mathbf{M}$ should be such that one can recover sufficiently precisely vectors which do not exactly belong to Σ but "live near" Σ in a certain sense.

- *Robustness to noise*: The measure $\mathbf{Mx}$ cannot be known with infinite precision. Aside from the obvious fact that this quantity is usually considered numerically, so that a quantization step occurs and involves a precision loss, linear operators $\mathbf{M}$ considered in inverse problems represent in general physical filters applied to a signal, such as transformations that a signal undergoes when captured or processed by a device. In this case, chances are that the measure is corrupted by a certain amount of noise, so that the measure is of the form $\mathbf{Mx} + \mathbf{e}$, where $\mathbf{e}$ is the additive noise term.

- *Practical recovery algorithm*: Probably the most important issue, one needs to be capable of practically solve the inverse problem with all the aforementioned constraints. The existence of a unique solution is therefore not sufficient to ensure the reconstruction: one needs an algorithm which will find the signal $\mathbf{x}$ within a given precision.

Expressing conditions under which a signal model can be recovered with these additional constraints is much more difficult and involves a part of subjectivity. In this chapter, we first review a well-posedness formulation previously proposed (Cohen et al., 2009) and called *Instance Optimality*. This condition was originally defined for the model of sparse vectors and characterizes a uniform stability to the model. The contributions of this chapter concern the generalization of these results to other models as those previously mentioned, as well as the study of Instance Optimality in the general case.

3.1 State of the art: Instance Optimality in the sparse case

Let's consider the vector space $E = \mathbb{R}^n$ and the set of k-sparse vectors Σ_k. Define on E a linear measurement operator $\mathbf{M}$ mapping the signal space in $\mathbb{R}^m$. The problem of inverting $\mathbf{M}$ for signals in Σ_k can be seen as the definition of a decoder $\Delta : \mathbb{R}^m \to \mathbb{R}^n$ such that $\forall \mathbf{x} \in \Sigma_k, \Delta(\mathbf{M}\mathbf{x}) = \mathbf{x}$, thus making $\mathbf{M}$ a linear encoder associated to the (typically nonlinear) decoder Δ.

As has already been mentioned in the previous section, a good decoder Δ is certainly expected to have nicer properties than simply reconstructing Σ_k, the first of it being the stability to the model: the signal $\mathbf{x}$ to be reconstructed may not belong exactly in Σ_k but "live near" Σ_k under a distance d, meaning that $d(\mathbf{x}, \Sigma_k) = \inf_{\mathbf{z}\in\Sigma_k} d(\mathbf{x}, \mathbf{z})$ is "small" in a certain sense. In this case, one wants to be able to build a sufficiently precise estimate of $\mathbf{x}$ from $\mathbf{M}\mathbf{x}$, that is a quantity $\Delta(\mathbf{M}\mathbf{x})$ such that $\|\mathbf{x} - \Delta(\mathbf{M}\mathbf{x})\|$ is "small" for a certain norm $\|\cdot\|$. This stability to the model has been formalized into the so-called *Instance Optimality* assumption on Δ. Δ is said to be instance optimal if:

$$\forall \mathbf{x} \in \mathbb{R}^n, \|\mathbf{x} - \Delta(\mathbf{M}\mathbf{x})\| \leq C d(\mathbf{x}, \Sigma_k), \tag{3.1}$$

for a certain choice of norm $\|\cdot\|$ and distance d. For this property to be meaningful, the constant C must not scale with n and typically "good" instance optimal decoders are decoders which involve a constant which is the same for all n (note that this implicitly relies on the fact that a sparse set $\Sigma_k \subset \mathbb{R}^n$ can be defined for any n). When the norm is ℓ_2 or ℓ_1 and the distance is ℓ_1, such good instance optimal decoders exist and can be implemented as the minimization of a convex objective (Donoho, 2006; Candès and Tao, 2006; Candès, 2008) under assumptions on $\mathbf{M}$ such as the Restricted Isometry Property (RIP). Note that Instance Optimality is a uniform upper bound on the reconstruction error, and that other types of bounds on decoders can be studied, particularly from a probabilistic point of view (Chandrasekaran et al., 2012). Other early work include upper bounds on the reconstruction error from noisy measurements with a regularizing function when the signal belongs exactly to the model (Engl et al., 1996).

In (Cohen et al., 2009), the authors considered the following question: *Given the encoder* $\mathbf{M}$*, is there a simple characterization of the existence of an instance optimal decoder?* Their goal was not to find implementable decoders that would have this property, but rather to identify conditions on $\mathbf{M}$ and Σ_k under which the reconstruction problem is ill-posed if one aims at finding an instance optimal decoder with small constant. The existence of a decoder Δ which satisfies (3.1) will be called the *Instance Optimality Property* (IOP). The authors proved that this IOP is closely related to a property of the kernel of $\mathbf{M}$ with respect to Σ_{2k}, called the *Null Space Property* (NSP). This relation allowed them to study the existence of stable decoders under several choices of norm $\|.\|$ and distance $d(.,.)$.

More precisely, the authors considered two norms $\|\cdot\|_G$ and $\|\cdot\|_E$ defined on the signal space $\mathbb{R}^n$, the distance derived from $\|\cdot\|_E$ being denoted d_E. Instance Optimality with

respect to these two norms rewrites as follows: a decoder $\Delta : \mathbb{R}^m \to \mathbb{R}^n$ is said to be instance optimal for k-sparse signals if

$$\forall \mathbf{x} \in \mathbb{R}^n, \|\mathbf{x} - \Delta(\mathbf{M}\mathbf{x})\|_G \leq C d_E(\mathbf{x}, \Sigma_k), \tag{3.2}$$

for some constant $C > 0$.

This property on Δ upper bounds the reconstruction error of a vector, measured by $\|\cdot\|_G$, by the distance from the vector to the model, measured by d_E. The authors prove that the existence of an instance optimal decoder, called IOP, is closely related to the NSP of $\mathbf{M}$ with respect to the set Σ_{2k} of $2k$-sparse vectors. Noting $\mathcal{N} = \ker(\mathbf{M})$, this NSP states

$$\forall \mathbf{h} \in \mathcal{N}, \|\mathbf{h}\|_G \leq D d_E(\mathbf{h}, \Sigma_{2k}) \tag{3.3}$$

for some constant D.

The relationship between the IOP and the NSP is the following: if there exists an instance optimal decoder Δ satisfying (3.2), then (3.3) holds with $D = C$. Conversely, if (3.3) holds, then there exists a decoder Δ such that (3.2) holds with $C = 2D$. Such a decoder can be defined as follows, supposing $\mathbf{M}$ is onto:

$$\Delta(\mathbf{M}\mathbf{x}) = \underset{\mathbf{z} \in (\mathbf{x} + \mathcal{N})}{\operatorname{argmin}} d_E(\mathbf{z}, \Sigma_k), \tag{3.4}$$

$\mathbf{x} + \mathcal{N}$ denoting the set $\{\mathbf{x} + \mathbf{h}, \mathbf{h} \in \mathcal{N}\}$. The well-posedness of this definition is discussed in Appendix B.1, in the more general setting where the model is a finite union of subspaces in finite dimension. Note that for generalized models, such a decoder may not necessarily exist since the infimum of $d_E(\mathbf{z}, \Sigma)$ may not be achieved, as we will discuss in the next section.

This result can be seen as an "equivalence" between the IOP and the NSP, with similar constants.

On top of this fundamental relationship between IOP and NSP, a question addressed in (Cohen et al., 2009) is that of the fundamental limits of dimension reduction: *Given the target dimension m and desired constant C, is there an encoder $\mathbf{M}$ with an associated instance optimal decoder?* They particularly showed that there is a fundamental trade-off between the size of the constant C in (3.1) (with ℓ_2 norm and ℓ_2 distance) and the dimension reduction ratio m/n.

In (Peleg et al., 2013), the theoretical results of (Cohen et al., 2009) are generalized in the case where one aims at stably decoding a vector living near a finite union of subspaces (UoS). They also show in this case the impossibility of getting a good ℓ_2/ℓ_2 instance optimal decoder with substantial dimensionality reduction. Their extension also covers the case where the quantity one wants to decode is not the signal itself but a linear measure of the signal.

In our thesis work, we further extended the study of the IOP to general models of signals. The global summary of our contributions is the subject of the next section.

3.2 Summary of the contributions

In our work, we consider signals of interest living in or near a subset Σ of a vector space E, without further restriction, and show that instance optimality can be generalized for such models. In fact, we consider the following generalizations:

- **Robustness to noise:** noise-robust instance optimality is characterized, showing somewhat surprisingly the equivalence between the existence of two flavors of noise-robust decoders (*noise-aware* and *noise-blind*);

- **Infinite dimension:** signal spaces E that may be infinite dimensional are considered. For example E may be a Banach space such as an L^p space, the space of signed measures, etc. This is motivated by recent work on infinite dimensional compressed sensing (Adcock et al., 2013; Hansen and Adcock, 2011) or compressive density estimation (Bourrier et al., 2013b), discussed in Chapter 2;
- **Task-oriented decoders:** the decoder is not constrained to approximate the signal $\mathbf{x}$ itself but rather a linear feature derived from the signal, $\mathbf{Ax}$, as in (Peleg et al., 2013); in the usual inverse problem framework, $\mathbf{A}$ is the identity. Examples of problems where $\mathbf{A} \neq \mathbf{I}$ include:
  - Medical imaging of a particular region of the body: as in Magnetic Resonance Imaging, one may acquire Fourier coefficients of a function defined on the body, but only want to reconstruct properly a particular region. In this case, $\mathbf{A}$ would be the orthogonal projection on this region.
  - Partial source separation: given an audio signal mixed from several sources whose positions are known, as well as the microphone filters, the task of isolating one of the sources from the mixed signal is a reconstruction task where E is the space of concatenated sources, and $\mathbf{A}$ orthogonally projects such a signal in a single source signal space.

We now summarize more precisely our main contributions which fall under two categories: on the one hand the characterization of the existence of an instance optimal decoder given a linear measurement operator $\mathbf{M}$ and a model Σ, on the other hand the limits of dimensionality reduction.

3.2.1 Instance optimality for inverse problems with general models

In the noiseless case, we express a concept of Instance Optimality which does not necessarily involve homogeneous norms and distances but some pseudo-norms instead. Such a generalized Instance Optimality can be expressed as follows:

$$\forall \mathbf{x} \in E, \|\mathbf{Ax} - \Delta(\mathbf{Mx})\|_G \leq C d_E(\mathbf{x}, \Sigma), \tag{3.5}$$

where $\|\cdot\|_G$ is a pseudo-norm and d_E is a distance the properties of which will be specified in due time, and $\mathbf{A}$ is a linear operator representing the feature one wants to estimate from $\mathbf{Mx}$. Our first contribution is to prove that the existence of a decoder Δ satisfying (3.5), which is a generalized IOP, can be linked with a generalized NSP, similarly to the sparse case. This generalized NSP can be stated as:

$$\forall \mathbf{h} \in \ker(\mathbf{M}), \|\mathbf{Ah}\|_G \leq D d_E(\mathbf{h}, \Sigma - \Sigma), \tag{3.6}$$

where the set $\Sigma - \Sigma$ is comprised of all differences of elements in Σ, that is $\Sigma - \Sigma = \{\mathbf{z}_1 - \mathbf{z}_2 | \mathbf{z}_1, \mathbf{z}_2 \in \Sigma\}$. The constants C and D are related by a factor no more than 2, as will be stated in Theorems 3 and 4 characterizing the relationships between these two properties. In particular, all previously mentioned low-dimensional models can fit in this generalized framework.

3.2.2 Noise-robust instance optimality

Our second contribution (Theorems 5 and 6) is to provide a noise-robust extension of instance optimality and link it to a property called the Robust NSP. Section 3.4 regroups

these noiseless and noise-robust results after a review of the initial IOP/NSP results of (Cohen et al., 2009). We show somewhat surprisingly that the existence of *noise-aware* instance optimal decoders for all noise levels implies the existence of a *noise-blind* decoder.

3.2.3 Infinite-dimensional inverse problems

The generalization to arbitrary vector spaces allows us to consider infinite dimensional inverse problems. Here things are not always as straightforward as in the finite dimensional setting. For example, in the theory of generalized sampling (Adcock and Hansen, 2012), even when the signal model Σ is simply a finite dimensional subspace, it can be necessary to oversample by some factor in order to guarantee stable recovery. In fact Theorem 4.1 of (Adcock and Hansen, 2011) can be read as a statement of ℓ_2/ℓ_2 instance optimality for a specific (linear) decoder given in terms of the NSP constant of the measurement operator. The results presented here therefore provide an extension of generalized sampling for linear models beyond ℓ_2.

In (Adcock and Hansen, 2011) the authors also combine generalized sampling theory with compressed sensing. However, rather than developing a uniform instance optimality theory which seems the most natural extension of generalized sampling, they adopt a non-uniform approach based on (Candès and Plan, 2011). Our results should enable the development of a uniform infinite dimensional CS theory.

3.2.4 Limits of dimensionality reduction with generalized models

The reformulation of IOP as an NSP allows us to consider the ℓ_2/ℓ_2 instance optimality for general models in Section 3.5. In this case, the NSP can be interpreted in terms of scalar product and we precise the necessity of the NSP for the existence of an instance optimal decoder. This leads to the proof of Theorem 8 stating that, just as in the sparse case, *one cannot expect to build an ℓ_2/ℓ_2 instance optimal decoder if* $\mathbf{M}$ *reduces substantially the dimension and the model is "too large"* in a precise sense. In particular, we will see that the model is "too large" when the set $\Sigma - \Sigma$ contains an orthonormal basis. This encompasses a wide range of standard models where a consequence of our results is that ℓ_2/ℓ_2 IOP with dimensionality reduction is impossible:

- **k-sparse vectors**. In the case where $\Sigma = \Sigma_k$ is the set of k-sparse vectors, Σ contains the null vector and the canonical basis, so that $\Sigma - \Sigma$ contains the canonical basis. Note that the impossibility of good ℓ_2/ℓ_2 IOP has been proved in (Cohen et al., 2009).

- **Block-sparse vectors** (Eldar et al., 2010). The same argument as above applies in this case as well, implying that imposing a block structure on sparsity does not improve ℓ_2/ℓ_2 feasibility.

- **Low-rank matrices** (Recht et al., 2010; Candès and Plan, 2011). In the case where $E = \mathcal{M}_n(\mathbb{R})$ and Σ is the set of matrices of rank $\leq k$, Σ also contains the null matrix and the canonical basis.

- **Low-rank + sparse matrices** (Zhou et al., 2010; Candès et al., 2011a). The same argument applies to the case where the model contains all matrices that are jointly low-rank *and* sparse, which appear in phase retrieval (Oymak et al., 2012; Ohlsson et al., 2011; Candès et al., 2011b).

- **Low-rank matrices with non-sparsity constraints**. In order to reduce the ambivalence of the low-rank + sparse decomposition of a matrix, (Candès et al., 2011a) introduced non-sparsity constraints on the low-rank matrix in order to enforce its entries to have approximately the same magnitude. However, as shown in Lemma 3, an orthonormal Fourier basis of the matrix space can be written as differences of matrices which belong to this model.

- **Reduced union of subspace models** (Baraniuk et al., 2010) obtained by pruning out the combinatorial collection of k-dimensional subspaces associated to k-sparse vectors. This covers block-sparse vectors (Eldar et al., 2010), tree-structured sparse vectors, and more. Despite the fact that these unions of subspaces may contain much fewer k-dimensional subspaces than the combinatorial number of subspaces of the standard k-sparse model, the same argument as in the k-sparse model applies to these signal models, provided they contain the basis collection of 1-sparse signals. This contradicts the naive intuition that ℓ_2/ℓ_2 IOP could be achievable at the price of substantially reducing the richness of the model through a drastic pruning of its subspaces.

- **k-sparse expansions in a dictionary model** (Rauhut et al., 2008). More generally, if the model is the set of vectors which a linear combination of at most k elements of a dictionary $\mathbf{D}$ which contains an orthogonal family or a tight frame, then Theorem 8 applies.

- **Cosparse vectors with respect to the finite difference operator** (Nam et al., 2013; Peleg et al., 2013). As shown in (Peleg et al., 2013), the canonical basis is highly cosparse with respect to the finite difference operator, hence it is contained in the corresponding union of subspaces.

- As shown in Lemma 2, this is also the case for **symmetric definite positive square matrices with k-sparse inverse**. The covariance matrix of high-dimensional Gaussian graphical models is of this type: the numerous pairwise conditional independences that characterize the structure of such models, and make them tractable, translate into zeros entries of the inverse covariance matrix (the concentration matrix). Combining sparsity prior on the concentration matrix with maximum likelihood estimation of covariance from data, permits to learn jointly the structure and the parameters of Gaussian graphical models (so called "covariance selection" problem) (Yuan and Lin, 2007; Yuan, 2010). In very high-dimensional cases, compressive solutions to this problem would be appealing.

- Johnson-Lindenstrauss embedding of **point clouds** (Achlioptas, 2001). Given a set $\mathcal{X}$ of L vectors in $\mathbb{R}^n$ and $\epsilon > 0$, there exists a linear mapping $f : \mathbb{R}^n \to \mathbb{R}^m$, with $m = \mathcal{O}(\ln(L)/\epsilon^2)$ and

$$(1-\epsilon)\|\mathbf{x}-\mathbf{y}\|_2 \leq \|f(\mathbf{x})-f(\mathbf{y})\|_2 \leq (1+\epsilon)\|\mathbf{x}-\mathbf{y}\|_2 \tag{3.7}$$

  holds for all $\mathbf{x}, \mathbf{y} \in \mathcal{X}$. The fact that the point cloud contains a tight frame is satisfied if it "spreads" in a number of directions which span the space. In this case, *one cannot guarantee precise out-of-sample reconstruction* of the points in $\mathbb{R}^n$ in the ℓ_2-sense, except for a very limited neighborhood of the point cloud. This is further discussed in Section 3.7.

3.2.5 Generalized Restricted Isometry Property

Our last contribution, in Section 3.6, is to study the relations between the NSP and a generalized version of the Restricted Isometry Property (RIP), which encompasses classical or recent RIP formulations, such as

- the **D**-RIP (Candès et al., 2011) for the dictionary model;
- the RIP for low-rank matrices (Candès and Plan, 2011);
- the **Ω**-RIP (Giryes et al., 2013) for the cosparse model.

This generalized RIP bounds $\|\mathbf{Mx}\|_F$ from below and/or above on a certain set V, and can be decomposed in:

$$\text{Lower} - \text{RIP} : \forall \mathbf{x} \in V, \alpha\|\mathbf{x}\|_G \leq \|\mathbf{Mx}\|_F \tag{3.8}$$
$$\text{Upper} - \text{RIP} : \forall \mathbf{x} \in V, \|\mathbf{Mx}\|_F \leq \beta\|\mathbf{x}\|_G, \tag{3.9}$$

where $\|\cdot\|_G$ and $\|\cdot\|_F$ are norms defined respectively on the signal space and on the measure space, and $0 < \alpha \leq \beta < +\infty$. We prove particularly in Theorem 9 that a generalized lower-RIP on $\Sigma - \Sigma$ implies the existence of instance optimal decoders in the noiseless and the noisy cases for a certain norm $\|\cdot\|_E$ we call the "M-norm"[a].

Furthermore, we prove that under an upper-RIP assumption on Σ, this M-norm can be upper bounded by an atomic norm (Chandrasekaran et al., 2012) defined using Σ and denoted $\|\cdot\|_\Sigma$. This norm is easier to interpret than the M-norm: it can in particular be upper bounded by usual norms for the k-sparse vectors and low-rank matrices models. We have the following general result relating generalized RIP and IOP (Theorem 10): if $\mathbf{M}$ satisfies a lower-RIP (3.8) for $V = \Sigma - \Sigma$ and an upper-RIP (3.9) for $V = \Sigma$, then for all $\delta > 0$, there exists a decoder Δ_δ satisfying $\forall \mathbf{x} \in E, \forall \mathbf{e} \in F$,

$$\|\mathbf{x} - \Delta_\delta(\mathbf{Mx} + \mathbf{e})\|_G \leq 2\left(1 + \frac{\beta}{\alpha}\right) d_\Sigma(\mathbf{x}, \Sigma) + \frac{2}{\alpha}\|\mathbf{e}\|_E + \delta, \tag{3.10}$$

which is a particular case of Robust instance optimality, as described in Section 3.4.

3.3 Structure of the chapter

The structure of the chapter is as follows: Section 3.4 first contains a quick review of the relationship between IOP and NSP in the usual sparse case, then exposes the more general setting considered in this paper, for which these properties and their relationship are extended, both in noiseless and noisy settings. Section 3.5 then focuses on the particular case of ℓ_2/ℓ_2 IOP, proving the impossibility for a certain class of models to achieve such IOP with decent precision in dimension reducing scenarii. In particular, we show that this encompasses a wide range of usual models. Finally, in Section 3.6, we get back to the problem of IO with general norms and prove that a generalized version of the lower-RIP implies the existence of an instance optimal decoder for a certain norm we call the "M-norm". We propose an upper-bound on this norm under a generalized upper-RIP assumption to get an IOP with simpler norms, illustrating the result in standard cases.

The proofs of most of the results presented in this chapter are found in Appendix B.

[a]The prefix "M-" should be thought as "Measurement-related norm" since in other works the measurement matrix may be denoted by other letters.

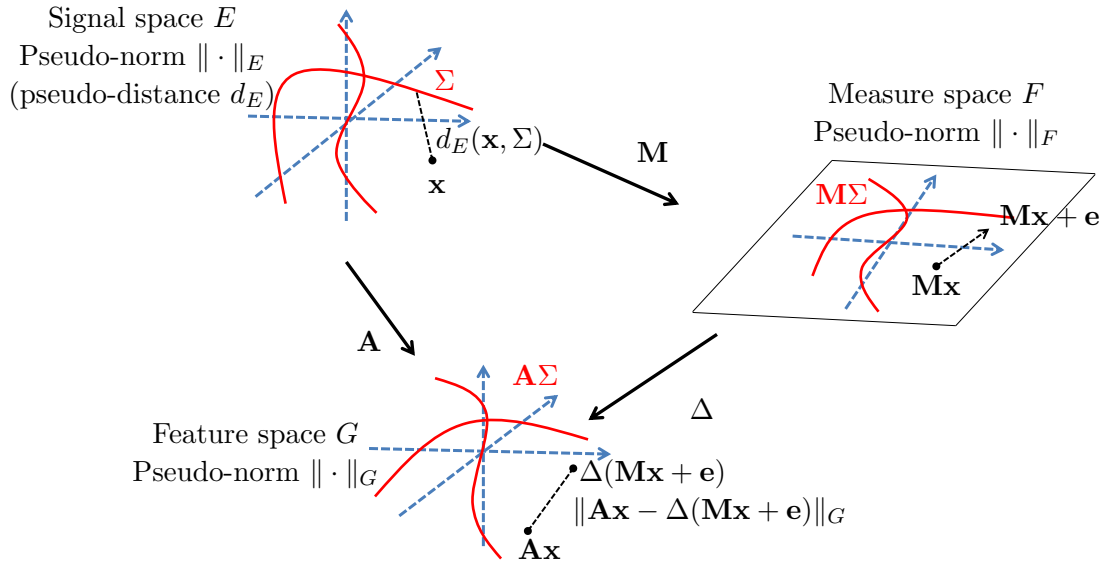

Figure 3.2: Illustration of the proposed generalized setting. The signals belong to the space E, supplied with a pseudo-norm $\|\cdot\|_E$ used to measure the distance from a vector to the model Σ containing the signals of interest. E is mapped in the measure space F by the operator $\mathbf{M}$ and the measure is perturbed by an additive noise $\mathbf{e}$. The space F is supplied with a pseudo-norm $\|\cdot\|_F$. The feature space G, supplied with a norm $\|\cdot\|_G$, is composed of vectors obtained by applying a linear operator $\mathbf{A}$ to the signals in E. These feature vectors are the vectors one wants to reconstruct from the measures in $\mathbf{M}$ by applying a decoder Δ. The reconstruction error for the vector $\mathbf{x}$ and noise $\mathbf{e}$ is therefore $\|\mathbf{Ax}-\Delta(\mathbf{Mx}+\mathbf{e})\|_G$. Note that in the case where $E = G$ and $\mathbf{A} = \mathbf{I}$, the decoder is aimed at reconstructing exactly the signals.

3.4 Generalized IOP and NSP equivalences

In this section, we extend the initial IOP/NSP relationship in several ways.

3.4.1 Proposed extensions

The framework we consider is more general. The signal space is a vector space E, possibly infinite-dimensional. In particular, E may be a Banach space such as an L^p space, the space of signed measures, etc. On this space is defined a linear operator $\mathbf{M} : E \to F$, where F is the measurement space, which will most likely be finite-dimensional in practice. We assume that $\mathbf{M}$ is onto. We further define a signal model $\Sigma \subset E$ comprising the signals which we want to be able to "reconstruct" from their images by $\mathbf{M}$. In the framework we consider, this "reconstruction" is not necessarily an inverse problem where we want to recover $\mathbf{x}$ from $\mathbf{Mx}$. More precisely, as in (Peleg et al., 2013), we consider a case where we want to recover from $\mathbf{Mx}$ a quantity $\mathbf{Ax}$, where $\mathbf{A}$ is a linear operator mapping E into a space G. When $G = E$ and $\mathbf{A} = \mathbf{I}$, we are brought back to the usual case where we want to reconstruct $\mathbf{x}$. This generalized framework is illustrated in Figure 3.2.

In this generalized framework, we are now interested in the concepts of IOP and NSP, as well as their relationship. A decoder $\Delta : F \to G$ will aim at approximating $\mathbf{Ax}$ from $\mathbf{Mx}$.

The approximation error will be measured by a function $\|\cdot\|_G : G \to \mathbb{R}_+$. This

	Triangle Inequality	Symmetry	$\|0\| = 0$	Definiteness	Homogeneity
$\|\cdot\|_E$	X	X	X	-	-
$\|\cdot\|_F$	X	X	X	-	-
$\|\cdot\|_G$	X	X	-	-	-

Table 3.1: Summary of the hypotheses on the pseudo-norms $\|\cdot\|_E$, $\|\cdot\|_F$ and $\|\cdot\|_G$. A cross means the property is required, a horizontal bar means it is not.

function needs not be a norm in order to state the following results. It still must satisfy the following properties:

$$\textbf{Symmetry} : \|\mathbf{x}\|_G = \|-\mathbf{x}\|_G \tag{3.11}$$

$$\textbf{Triangle inequality} : \|\mathbf{x}+\mathbf{y}\|_G \leq \|\mathbf{x}\|_G + \|\mathbf{y}\|_G. \tag{3.12}$$

The differences with a regular norm is that neither definiteness nor homogeneity is required: $\|\mathbf{x}\|_G = 0$ needs not imply $\mathbf{x} = 0$ and $\|\lambda\mathbf{x}\|_G$ needs not equal $|\lambda|\|\mathbf{x}\|_G$. We provide two examples of such pseudo-norms in the case where $G = \mathbb{R}^n$:

- $\|\cdot\|_G$ can be defined as a "non-normalized" ℓ_p-quasinorm for $0 \leq p \leq 1$, that is $\|\mathbf{x}\|_G = \sum_{i=1}^n |x_i|^p$. In this case, $\|\lambda\mathbf{x}\|_G = |\lambda|^p\|\mathbf{x}\|_G$.

- More generally, if $f : \mathbb{R}_+ \to \mathbb{R}_+$ is a concave function such that $f(x) = 0 \Leftrightarrow x = 0$, then $\|\cdot\|_G$ can be defined as the f-(pseudo-)norm $\|\mathbf{x}\|_f = \sum_{i=1}^n f(|x_i|)$, see (Gribonval and Nielsen, 2007).

In order to measure the *distance from a vector to the model*, we also endow E with a pseudo-norm $\|\cdot\|_E : E \to \mathbb{R}_+$ which satisfies the same properties as $\|\cdot\|_G$ with the additional requirement that $\|0\|_E = 0$. The pseudo-distance d_E is defined on E^2 by $d(\mathbf{x},\mathbf{y}) = \|\mathbf{x}-\mathbf{y}\|_E$. Yet again, $\|\cdot\|_E$ can be defined as a non-normalized ℓ_p-norm or an f-norm.

We will also consider a noisy framework where the measure $\mathbf{Mx}$ is perturbed by an additive noise term $\mathbf{e}$. To consider IOP and NSP in this context, we *measure the amount of noise* with a pseudo-norm in the measurement space F, which we will denote by $\|\cdot\|_F$. The assumptions we make on $\|\cdot\|_F$ are the same as the assumptions on $\|\cdot\|_E$.

To sum up, here are the extensions we propose compared to the framework of (Cohen et al., 2009; Peleg et al., 2013) :

- The measure $\mathbf{Mx}$ can be perturbed by an additive noise $\mathbf{e}$.

- The model set Σ can be any subset of E.

- E is not necessarily $\mathbb{R}^n$ but can be any vector space, possibly infinite-dimensional.

- The reconstruction of $\mathbf{Ax}$ is targeted rather than that of $\mathbf{x}$.

- The functions $\|\cdot\|_E$, $\|\cdot\|_F$ and $\|\cdot\|_G$ need not be norms but can be pseudo-norms with relaxed hypotheses. In particular, Table 3.1 summarizes the requirements on these functions.

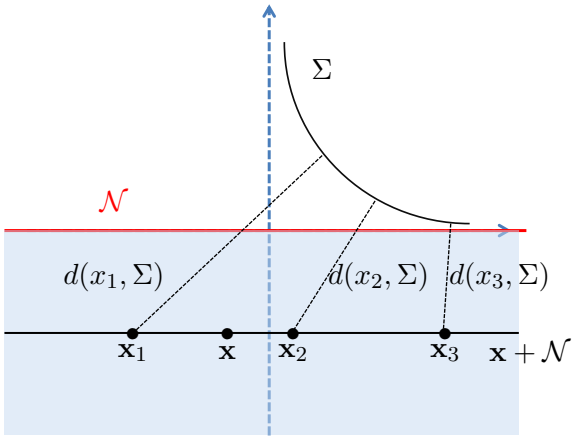

Figure 3.3: Necessity of the additive δ term in a simple case. For each $\mathbf{x}$ in the blue half-plane, the distance $d_E(x+\mathcal{N},\Sigma)$ is never reached at a particular point of $\mathbf{x}+\mathcal{N}$: the distance strictly decreases as one goes right along the affine plane $\mathbf{x}+\mathcal{N}$ ($d(\mathbf{x}_1,\Sigma) < d(\mathbf{x}_2,\Sigma) < d(\mathbf{x}_3,\Sigma)$), so that the minimal distance is reached "at infinity".

The noiseless case

We first consider the same framework as (Cohen et al., 2009; Peleg et al., 2013), where one measures $\mathbf{Mx}$ with infinite precision. In our generalized framework, instance optimality for a decoder Δ reads:

$$\forall \mathbf{x} \in E, \|\mathbf{Ax} - \Delta(\mathbf{Mx})\|_G \leq C d_E(\mathbf{x},\Sigma).$$

We will prove that if IOP holds, *i.e.*, if the above holds for a certain decoder Δ, then a generalized NSP is satisfied, that is:

$$\forall \mathbf{h} \in \mathcal{N}, \|\mathbf{Ah}\|_G \leq D d_E(\mathbf{h}, \Sigma - \Sigma),$$

with $D = C$. Note that the set Σ_{2k} has been replaced by $\Sigma - \Sigma = \{\mathbf{x}-\mathbf{y}, \mathbf{x} \in \Sigma, \mathbf{y} \in \Sigma\}$. When $\Sigma = \Sigma_k$, we have indeed $\Sigma - \Sigma = \Sigma_{2k}$.

The construction of an instance optimal decoder from the NSP is more complicated and the form of the instance optimality we get depends on additional assumptions on Σ and $\mathbf{M}$. Let's first suppose that for all $\mathbf{x} \in E$, there exists $\mathbf{z} \in (\mathbf{x}+\mathcal{N})$ such that $d_E(\mathbf{z},\Sigma) = d_E(\mathbf{x}+\mathcal{N},\Sigma)$. Then the NSP (3.6) implies the existence of an instance optimal decoder satisfying (3.5) with $C = 2D$. If this assumption is not true anymore, then the NSP implies a slightly modified IOP, which states, for any $\delta > 0$, the existence of a decoder Δ_δ such that:

$$\forall \mathbf{x} \in E, \|\mathbf{Ax} - \Delta_\delta(\mathbf{Mx})\|_G \leq C d_E(\mathbf{x},\Sigma) + \delta, \tag{3.13}$$

reflecting the fact that one cannot necessarily consider the exact quantity

$$\underset{\mathbf{z} \in (\mathbf{x}+\mathcal{N})}{\operatorname{argmin}} \; d_E(\mathbf{z},\Sigma)$$

but rather a certain vector $\mathbf{z} \in (\mathbf{x}+\mathcal{N})$ satisfying $d_E(\mathbf{z},\Sigma) \leq d_E(\mathbf{x}+\mathcal{N},\Sigma) + \delta$. A similar positive "projection error" appears in (Blumensath, 2011).

Remark 1. *To understand the necessity of such an additive error term when Σ is a general set, we can consider the following toy example depicted in Figure 3.3 where $E = \mathbb{R}^2$,*

$\mathcal{N} = \mathbb{R} \times \{0\}$, $\Sigma = \{(x_1, x_2) \in (\mathbb{R}_+)^2 : x_2 = \frac{1}{x_1}\}$ and $\|\cdot\|_G / \|\cdot\|_E$ are the ℓ_2 norm. In this case, the minimal distance between $\mathbf{x} + \mathcal{N}$ and Σ is not reached at any point, making it necessary to add the δ term for the decoder to be well-defined.

In this setting, the NSP (3.6) implies the existence of instance optimal decoders in the sense of (3.13) for all $\delta > 0$. Moreover, this weak IOP formulation still implies the regular NSP with $D = C$. This is summarized in Theorems 3 and 4.

Theorem 3. *Suppose $\forall \delta > 0$, there exists a decoder Δ_δ satisfying (3.13):*

$$\forall \mathbf{x} \in E, \|\mathbf{A}\mathbf{x} - \Delta_\delta(\mathbf{M}\mathbf{x})\|_G \leq C d_E(\mathbf{x}, \Sigma) + \delta.$$

Then $\mathbf{M}$ satisfies the NSP (3.6):

$$\forall \mathbf{h} \in \mathcal{N}, \|\mathbf{A}\mathbf{h}\|_G \leq D d_E(\mathbf{h}, \Sigma - \Sigma),$$

with constant $D = C$.

Theorem 4. *Suppose that $\mathbf{M}$ satisfies the NSP (3.6):*

$$\forall \mathbf{h} \in \mathcal{N}, \|\mathbf{A}\mathbf{h}\|_G \leq D d_E(\mathbf{h}, \Sigma - \Sigma).$$

Then $\forall \delta > 0$, there exists a decoder Δ_δ satisfying (3.13):

$$\forall \mathbf{x} \in E, \|\mathbf{A}\mathbf{x} - \Delta_\delta(\mathbf{M}\mathbf{x})\|_G \leq C d_E(\mathbf{x}, \Sigma) + \delta,$$

with $C = 2D$.

If we further assume that

$$\forall \mathbf{x} \in E, \exists \mathbf{z} \in (\mathbf{x} + \mathcal{N}), d_E(\mathbf{z}, \Sigma) = d_E(\mathbf{x} + \mathcal{N}, \Sigma), \tag{3.14}$$

then there exists a decoder Δ satisfying (3.5):

$$\forall \mathbf{x} \in E, \|\mathbf{A}\mathbf{x} - \Delta(\mathbf{M}\mathbf{x})\|_G \leq C d_E(\mathbf{x}, \Sigma) \tag{3.15}$$

with $C = 2D$.

Note that this result is similar to the result proven in (Peleg et al., 2013), which was stated in the case where Σ is a finite union of subspaces in finite dimension. In this framework, condition (3.14) is always satisfied as soon as $\|\cdot\|_E$ is a norm, by the same argument as in usual CS (see Appendix B.1).

Let's also note the following property: if $\|\cdot\|_E$ is definite, that is $\|\mathbf{x}\|_E = 0 \Rightarrow \mathbf{x} = 0$, then d_E is a distance. In the following proposition, we prove that if we further suppose that the set $\Sigma + \mathcal{N}$ is a closed set with respect to d_E, then the NSP (3.6) implies for any $\delta > 0$ the existence of a decoder Δ_δ satisfying (3.5) with $C = (2 + \delta)D$. This assumption therefore allows us to suppress the additive constant in (3.13) and replace it by an arbitrarily small increase in the multiplicative constant of (3.5).

Proposition 1. *Suppose that $\mathbf{M}$ satisfies the NSP (3.6), that d_E is a distance and that $\Sigma + \mathcal{N}$ is a closed set with respect to d_E. Then $\forall \delta > 0$, there exists a decoder Δ_δ satisfying:*

$$\forall \mathbf{x} \in E, \|\mathbf{A}\mathbf{x} - \Delta_\delta(\mathbf{M}\mathbf{x})\|_G \leq (2 + \delta) D d_E(\mathbf{x}, \Sigma). \tag{3.16}$$

The noisy case

In practice, it is not likely that one can measure with infinite precision the quantity $\mathbf{Mx}$. This measure is likely to be contaminated with some noise, which will be considered in the following as an additive term $\mathbf{e} \in F$, so that the measure one gets is $\mathbf{y} = \mathbf{Mx} + \mathbf{e}$. In this case, a good decoder should be robust to noise, so that moderate values of $\mathbf{e}$ should not have a severe impact on the approximation error. We are interested in the existence of similar results as before in this noisy setting.

We first need to define a noise-robust version of instance optimality. The robustness to noise of practical decoders is in fact a problem that has been considered by many authors. A first type of result considers *noise-aware decoders*, where given the noise level $\epsilon \geq 0$ a decoder Δ fulfills the following property: $\forall \mathbf{x} \in E, \forall \mathbf{e} \in F$,

$$\|\mathbf{e}\|_F \leq \epsilon \Rightarrow \|\mathbf{Ax} - \Delta(\mathbf{Mx} + \mathbf{e})\|_G \leq C_1 d_E(\mathbf{x}, \Sigma) + C_2 \epsilon. \tag{3.17}$$

Here, the upper bound on the approximation error gets a new term measuring the amplitude of the noise. For example, this noise-robust instance optimality holds for a noise-aware ℓ_1 decoder in the sparse case with bounded noise (Candès, 2008) for $\|\cdot\|_G = \|\cdot\|_2$ and $\|\cdot\|_E = \|.\|_1/\sqrt{k}$, provided $\mathbf{M}$ satisfies the RIP on Σ_{2k}.

In practical settings, it is hard to assume that one knows precisely the noise level. To exploit the above guarantee with a noise-aware decoder, one typically needs to overestimate the noise level. This loosens the effective performance guarantee and potentially degrades the actual performance of the decoder. An apparently stronger property for a decoder is to be robust *even without knowledge of the noise level*: $\forall \mathbf{x} \in E, \forall \mathbf{e} \in F$,

$$\|\mathbf{Ax} - \Delta(\mathbf{Mx} + \mathbf{e})\|_G \leq C_1 d_E(\mathbf{x}, \Sigma) + C_2 \|\mathbf{e}\|_F. \tag{3.18}$$

Further on, such decoders will be referred to as *noise-blind*. Guarantees of this type have been obtained under a RIP assumption for practical decoders such as iterative hard thresholding, CoSAMP, or hard thresholding pursuit, see e.g. (Foucart, 2011, Corollary 3.9).

Of course, the existence of a *noise-blind* noise-robust decoder in the sense of (3.18) implies the existence of a *noise-aware* noise-robust decoder in the sense of (3.17) for any noise level ϵ. We will see that, somewhat surprisingly, the converse is true in a sense, for both are equivalent to a noise-robust NSP.

Just as in the noiseless case, dealing with an arbitrary model Σ and possibly infinite dimensional E requires some caution. For $\delta > 0$, the noise-robust (and noise-blind) instance optimality of a decoder Δ_δ is defined as: $\forall \mathbf{x} \in E, \forall \mathbf{e} \in F$,

$$\|\mathbf{Ax} - \Delta_\delta(\mathbf{Mx} + \mathbf{e})\|_G \leq C_1 d_E(\mathbf{x}, \Sigma) + C_2 \|\mathbf{e}\|_F + \delta. \tag{3.19}$$

One can see that Δ_δ necessarily also satisfies the noiseless instance optimality (3.13) by setting $\mathbf{e} = 0$.

As we show below, if for every $\delta > 0$ there exists a noise-robust instance optimal decoder Δ_δ satisfying (3.19), then a generalized NSP for $\mathbf{M}$ relatively to $\Sigma - \Sigma$, referred to as Robust NSP, must hold:

$$\forall \mathbf{h} \in E, \|\mathbf{Ah}\|_G \leq D_1 d_E(\mathbf{h}, \Sigma - \Sigma) + D_2 \|\mathbf{Mh}\|_F, \tag{3.20}$$

with $D_1 = C_1$ and $D_2 = C_2$. This property appears e.g. in (Foucart and Rauhut, 2013) (Chap. 4) with $\|\cdot\|_G = \|\cdot\|_E = \|\cdot\|_1$ and $\|\cdot\|_F$ any norm. Note that this Robust NSP

concerns every vector of E and not just the vectors of the null space $\mathcal{N} = \ker(\mathbf{M})$[b]. In the case where $\mathbf{h} \in \mathcal{N}$, one retrieves the regular NSP. For other vectors $\mathbf{h}$, another additive term, measuring the "size" of $\mathbf{Mh}$, appears in the upper bound.

Conversely, the Robust NSP implies the existence of noise-robust instance optimal decoders Δ_δ satisfying (3.19) with $C_1 = 2D_1$ and $C_2 = 2D_2$ for all $\delta > 0$. These results are summarized in Theorems 5 and 6.

Theorem 5. *Suppose $\forall \delta > 0$, there exists a decoder Δ_δ satisfying* (3.19)*: $\forall \mathbf{x} \in E, \forall \mathbf{e} \in F$,*

$$\|\mathbf{Ax} - \Delta_\delta(\mathbf{Mx} + \mathbf{e})\|_G \leq C_1 d_E(\mathbf{x}, \Sigma) + C_2 \|\mathbf{e}\|_F + \delta.$$

Then $\mathbf{M}$ satisfies the Robust NSP (3.20)*:*

$$\forall \mathbf{h} \in E, \|\mathbf{Ah}\|_G \leq D_1 d_E(\mathbf{h}, \Sigma - \Sigma) + D_2 \|\mathbf{Mh}\|_F,$$

with constants $D_1 = C_1$ and $D_2 = C_2$.

Theorem 6. *Suppose that $\mathbf{M}$ satisfies the Robust NSP* (3.20)*:*

$$\forall \mathbf{h} \in E, \|\mathbf{Ah}\|_G \leq D_1 d_E(\mathbf{h}, \Sigma - \Sigma) + D_2 \|\mathbf{Mh}\|_F.$$

Then $\forall \delta > 0$, there exists a decoder Δ_δ satisfying (3.19)*: $\forall \mathbf{x} \in E, \forall \mathbf{e} \in F$,*

$$\|\mathbf{Ax} - \Delta_\delta(\mathbf{Mx} + \mathbf{e})\|_G \leq C_1 d_E(\mathbf{x}, \Sigma) + C_2 \|\mathbf{e}\|_F + \delta,$$

with constants $C_1 = 2D_1$ and $C_2 = 2D_2$.

We conclude this section by discussing the relation between noise-aware and noise-blind decoders. A noise-aware version of noise-robust instance optimality can be defined where for $\epsilon \geq 0, \delta > 0$ we require $\forall \mathbf{x} \in E, \forall \mathbf{e} \in F$,

$$\|\mathbf{e}\|_F \leq \epsilon \Rightarrow \|\mathbf{Ax} - \Delta_{\delta,\epsilon}(\mathbf{Mx} + \mathbf{e})\|_G \leq C_1 d_E(\mathbf{x}, \Sigma) + C_2 \epsilon + \delta. \tag{3.21}$$

Of course, the existence of a noise-blind instance optimal decoder implies that of noise-aware decoders for every $\epsilon \geq 0$. The converse is indeed essentially true, up to the value of the constants C_i:

Theorem 7. *Suppose $\forall \epsilon, \delta > 0$, there exists a* noise-aware *decoder $\Delta_{\delta,\epsilon}$ satisfying* (3.21)*: $\forall \mathbf{x} \in E, \forall \mathbf{e} \in F$,*

$$\|\mathbf{e}\|_F \leq \epsilon \Rightarrow \|\mathbf{Ax} - \Delta_{\delta,\epsilon}(\mathbf{Mx} + \mathbf{e})\|_G \leq C_1 d_E(\mathbf{x}, \Sigma) + C_2 \epsilon + \delta.$$

Then $\mathbf{M}$ satisfies the Robust NSP (3.20) *with constants $D_1 = C_1$ and $D_2 = 2C_2$. Therefore, by Theorem 6, there exists an instance optimal* noise-blind *decoder satisfying: $\forall \mathbf{x} \in E, \forall \mathbf{e} \in F$,*

$$\|\mathbf{Ax} - \Delta_\delta(\mathbf{Mx} + \mathbf{e})\|_G \leq 2C_1 d_E(\mathbf{x}, \Sigma) + 4C_2 \|\mathbf{e}\|_F + \delta.$$

[b] In fact, unlike the NSP (3.6), (3.20) is not purely a property of the null space $\mathcal{N}$ even though it implies the NSP. The name Robust NSP is thus somewhat improper, but has become a standard for this type of property.

3.5 ℓ_2/ℓ_2 Instance Optimality

In this section, we suppose that E is a Hilbert space equipped with the norm $\|\cdot\|_2$ and scalar product $\langle\cdot,\cdot\rangle$, that $F = \mathbb{R}^m$ and we consider a finite-dimensional subspace V of dimension n, on which we define the measure operator $\mathbf{M}$. We are interested in the following question in the noiseless framework: *Is it possible to have a "good" noiseless instance optimal decoder with $\|\cdot\|_G = \|\cdot\|_E = \|\cdot\|_2$ in a dimensionality reducing context where $m \ll n$?*

A result of (Cohen et al., 2009) states that in the usual sparse setting, one cannot expect to get a good instance optimal decoder if $\mathbf{M}$ performs a substantial dimensionality reduction, the best corresponding constant being $\sqrt{\frac{n}{m}}$. In (Peleg et al., 2013), the authors prove that this lower bound on the constant holds in the case where Σ is a finite union of subspaces in finite dimension. Here, we are interested in a version of this result for the general case where Σ can be a more general subset of E. More precisely, we will give a sufficient condition on Σ under which the optimal ℓ_2/ℓ_2 instance optimalityconstant is of the order of $\sqrt{\frac{n}{m}}$, thus preventing the existence of a ℓ_2/ℓ_2 instance optimal decoder with small constant if $m \ll n$.

3.5.1 Homogeneity of the NSP

In the case where $\|\cdot\|_G$, $\|\cdot\|_E$ and $\|\cdot\|_F$ are actual norms, the general NSP can be rewritten as an NSP holding on the cone $\mathbb{R}(\Sigma - \Sigma)$ generated by $\Sigma - \Sigma$, i.e., the set $\{\lambda\mathbf{z} | \lambda \in \mathbb{R}, \mathbf{z} \in \Sigma - \Sigma\}$.

Lemma 1. *If $\|\cdot\|_G$ and $\|\cdot\|_E$ are norms, we have an equivalence between the NSP on $\Sigma - \Sigma$:*

$$\forall \mathbf{h} \in \mathcal{N}, \|\mathbf{A}\mathbf{h}\|_G \leq D d_E(\mathbf{h}, \Sigma - \Sigma), \tag{3.22}$$

and the NSP on $\mathbb{R}(\Sigma - \Sigma)$:

$$\forall \mathbf{h} \in \mathcal{N}, \|\mathbf{A}\mathbf{h}\|_G \leq D d_E(\mathbf{h}, \mathbb{R}(\Sigma - \Sigma)). \tag{3.23}$$

Similarly, if $\|\cdot\|_G$, $\|\cdot\|_E$ and $\|\cdot\|_F$ are norms, we have an equivalence between the robust NSP on $\Sigma - \Sigma$:

$$\forall \mathbf{h} \in E, \|\mathbf{A}\mathbf{h}\|_G \leq D_1 d_E(\mathbf{h}, \Sigma - \Sigma) + D_2 \|\mathbf{M}\mathbf{h}\|_F, \tag{3.24}$$

and the robust NSP on $\mathbb{R}(\Sigma - \Sigma)$:

$$\forall \mathbf{h} \in E, \|\mathbf{A}\mathbf{h}\|_G \leq D_1 d_E(\mathbf{h}, \mathbb{R}(\Sigma - \Sigma)) + D_2 \|\mathbf{M}\mathbf{h}\|_F. \tag{3.25}$$

This lemma, which is valid even in the case where $\mathbf{A}$ is not the identity, shows that the NSP imposes a constraint on the whole linear cone spanned by the elements of $\Sigma - \Sigma$ and not only on the elements themselves. Note that this equivalence is trivial in the case where Σ is a union of subspaces since $\Sigma - \Sigma$ is already a cone in this case.

3.5.2 The optimal ℓ_2/ℓ_2 NSP constant

Remark 2. *In the subsequent sections of the paper, we will assume that $\mathbf{A} = \mathbf{I}$ (this implies $G = E$), so that one aims at reconstructing the actual signal.*

In the ℓ_2/ℓ_2 case, one can give a simple definition of the optimal NSP constant D_*, that is the minimal real positive number D such that the ℓ_2/ℓ_2 NSP is satisfied with constant D:

$$D_* = \inf \{D \in \mathbb{R}_+ | \forall \mathbf{h} \in \mathcal{N}, \|\mathbf{h}\|_2 \leq D d_2(\mathbf{h}, \Sigma - \Sigma)\}. \tag{3.26}$$

This definition assumes that there exists some constant so that the NSP is satisfied. Using the NSP definition and Lemma 1, we get that

$$D_* = \sup_{\substack{\mathbf{h} \in \mathcal{N} \\ \mathbf{z} \in \mathbb{R}(\Sigma - \Sigma)}} \frac{\|\mathbf{h}\|_2}{\|\mathbf{h} - \mathbf{z}\|_2} = \sup_{\substack{\mathbf{h} \in \mathcal{N} \\ \mathbf{z} \in \mathbb{R}(\Sigma - \Sigma)}} \frac{1}{\|\frac{\mathbf{h}}{\|\mathbf{h}\|_2} - \frac{\mathbf{z}}{\|\mathbf{h}\|_2}\|_2}. \tag{3.27}$$

Denoting $\mathcal{N}_1$ and Σ_1 respectively the set of unit-norm vectors (in the ℓ_2 sense) of $\mathcal{N}$ and $\Sigma - \Sigma$, we can rewrite the expression above as:

$$D_* = \sup_{\substack{\mathbf{h} \in \mathcal{N}_1 \\ \mathbf{z} \in \mathbb{R}(\Sigma - \Sigma)}} \frac{1}{\|\mathbf{h} - \mathbf{z}\|_2} = \sup_{\substack{\mathbf{h} \in \mathcal{N}_1 \\ \mathbf{z} \in \Sigma_1 \\ \lambda \in \mathbb{R}}} \frac{1}{\|\mathbf{h} - \lambda \mathbf{z}\|_2}. \tag{3.28}$$

A simple study gives that if $\|\mathbf{h}\|_2 = \|\mathbf{z}\|_2 = 1$, then $\sup_{\lambda \in \mathbb{R}} \frac{1}{\|\mathbf{h} - \lambda \mathbf{z}\|_2} = \frac{1}{\sqrt{1 - \langle \mathbf{h}, \mathbf{z} \rangle^2}}$, so that:

$$D_* = \sup_{\substack{\mathbf{h} \in \mathcal{N}_1 \\ \mathbf{z} \in \Sigma_1}} \frac{1}{\sqrt{1 - \langle \mathbf{h}, \mathbf{z} \rangle^2}}. \tag{3.29}$$

The contraposition of Theorem 3 gives the following result : if the NSP (3.6) is not satisfied for a certain constant D, then no decoder Δ_δ can satisfy instance optimality (3.13) with constant D. In the ℓ_2/ℓ_2 case, considering $D < D_*$, $\mathbf{h} \in \mathcal{N} \cap \mathcal{B}_2$ and $\mathbf{z} \in \mathbb{R}(\Sigma - \Sigma) \cap \mathcal{B}_2$ such that $\langle \mathbf{h}, \mathbf{z} \rangle^2 \geq 1 - \frac{1}{D^2}$, we can construct two vectors such that for any decoder, instance optimality with constant $< \sqrt{D^2 - 1}$ can only be satisfied for at most one of them. This will shed light on the link between NSP and IOP. We have $\mathbf{z} = \frac{\mathbf{z}_1 - \mathbf{z}_2}{\|\mathbf{z}_1 - \mathbf{z}_2\|_2}$ for some $\mathbf{z}_1, \mathbf{z}_2 \in \Sigma$. Let Δ be a decoder. If $\Delta(\mathbf{M}\mathbf{z}_1) \neq \mathbf{z}_1$, then this vector prevents Δ from being instance optimal. The same goes for $\mathbf{z}_2$ if $\Delta(\mathbf{M}\mathbf{z}_2) \neq \mathbf{z}_2$. Now, let's suppose that $\mathbf{z}_1$ and $\mathbf{z}_2$ are correctly decoded. In this case, $(\mathbf{z}_1 + \mathbf{z}_2)/2$ is decoded with a constant worse than $\sqrt{D^2 - 1}$, as depicted in Figure 3.4. Indeed, noting $\mathbf{p} = (\mathbf{z}_1 + \mathbf{z}_2)/2$ and defining the vectors $\mathbf{p}_1$ and $\mathbf{p}_2$ respectively as the orthogonal projections of $\mathbf{z}_1$ and $\mathbf{z}_2$ on the affine plane $\mathbf{p} + \mathcal{N}$, we must have $\Delta(\mathbf{M}\mathbf{p}_1) = \Delta(\mathbf{M}\mathbf{p}_2)$. Denoting as $p_{\mathcal{N}^\perp}$ the orthogonal projection on $\mathcal{N}^\perp$, we have $d_2(\mathbf{p}_1, \Sigma) \leq d_2(\mathbf{p}_1, \mathbf{z}_1) = \|p_{\mathcal{N}^\perp}(\mathbf{z}_2 - \mathbf{z}_1)\|_2/2$. Similarly, $d_2(\mathbf{p}_2, \Sigma) \leq \|p_{\mathcal{N}^\perp}(\mathbf{z}_2 - \mathbf{z}_1)\|_2/2$. The fact that $\Delta(\mathbf{M}\mathbf{p}_1) = \Delta(\mathbf{M}\mathbf{p}_2)$ implies that there exists $i \in \{1, 2\}$ such that $\|\mathbf{p}_i - \Delta(\mathbf{M}\mathbf{p}_i)\|_2 \geq \|\mathbf{p}_1 - \mathbf{p}_2\|_2/2 = \|p_{\mathcal{N}}(\mathbf{z}_1 - \mathbf{z}_2)\|_2/2$. Therefore, $\frac{\|\mathbf{p}_i - \Delta(\mathbf{M}\mathbf{p}_i)\|_2}{d_2(\mathbf{p}_i, \Sigma)} \geq \frac{\|p_{\mathcal{N}}(\mathbf{z}_2 - \mathbf{z}_1)\|_2}{\|p_{\mathcal{N}^\perp}(\mathbf{z}_2 - \mathbf{z}_1)\|_2} \geq D\sqrt{1 - \frac{1}{D^2}} = \sqrt{D^2 - 1}$. This illustrates the closeness between NSP and IOP: a vector of $\mathbb{R}(\Sigma - \Sigma)$ which is correlated with $\mathcal{N}$ can be used to define a couple of vectors such that for any decoder, one of the vectors will not be well decoded.

3.5.3 ℓ_2/ℓ_2 IO with dimensionality reduction

Main theorem

Let's now exploit the expression of D_* to state the main result of this section: if $\mathbb{R}(\Sigma - \Sigma)$ contains an orthonormal basis of the finite-dimensional subspace $V \subset E$ (or even a family of vectors that is sufficiently correlated with every vector of V), then one cannot expect to

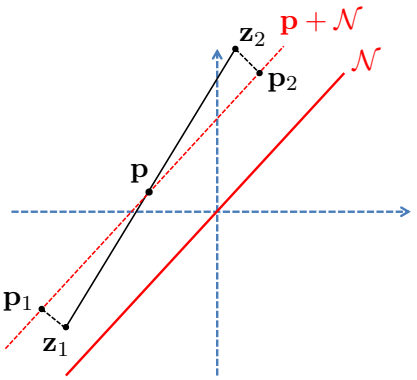

Figure 3.4: Illustration of the impact of the correlation between $\mathcal{N}$ and $\Sigma - \Sigma$ on instance optimality. Here, $\mathbf{z}_1$ and $\mathbf{z}_2$ are two vectors in Σ such that $\mathbf{z}_1 - \mathbf{z}_2$ is well correlated with $\mathcal{N}$, implying that at least one of the two vectors $\mathbf{p}_1$ and $\mathbf{p}_2$, which are close to Σ but far from one another, will not be well decoded.

get a ℓ_2/ℓ_2 instance optimal decoder with a small constant while $\mathbf{M}$ substantially reduces the dimension of V. The fact that $\mathbb{R}(\Sigma - \Sigma)$ contains such a tight frame implies that the dimension of $\mathcal{N}$ cannot be too big without $\mathcal{N}$ being strongly correlated with $\Sigma - \Sigma$, thus yielding the impossibility of a good instance optimal decoder.

Before showing examples where this theorem applies, let's first state it and prove it.

Theorem 8. *Suppose V is of dimension n and $\Sigma - \Sigma$ contains a family $\mathbf{z}_1, \dots, \mathbf{z}_n$ of unit-norm vectors of E satisfying $\forall \mathbf{x} \in V$, $\sum_{i=1}^n \langle \mathbf{z}_i, \mathbf{x} \rangle^2 \geq K \|\mathbf{x}\|_2^2$. Then to satisfy the NSP on V, $\mathbf{M}$ must map V into a space of dimension at least $\left(1 - \frac{1}{K}\left(1 - \frac{1}{D_*^2}\right)\right) n$.*

If the number of measurements m is fixed, then an ℓ_2/ℓ_2 IO decoder must have a constant at least $\frac{1}{\sqrt{1-K\left(1-\frac{m}{n}\right)}}$.

In particular, if $\Sigma - \Sigma$ contains an orthonormal basis of V, then $K = 1$ and the minimal number of measures to achieve NSP with constant D_* is n/D_*^2. Similarly, if m is fixed so that $m \ll n$, then a ℓ_2/ℓ_2 instance optimal decoder has constant at least $\sqrt{\frac{n}{m}}$.

Examples

As discussed in the introduction, there is a wide range of standard models where $\Sigma - \Sigma$ contains an orthonormal basis, and so where ℓ_2/ℓ_2 IOP with dimensionality reduction is impossible. We provide here less trivial examples, where $E = V$ is finite-dimensional.

Symmetric definite positive matrices with sparse inverse.

Lemma 2. *Consider E is the space of symmetric n-dimensional matrices, and $\Sigma \subset E$ the subset of symmetric positive-definite matrices with sparse inverse and with sparsity constant $k \geq n + 2$ (note that $k \geq n$ is necessary for the matrix to be invertible). The set $\Sigma - \Sigma$ contains an orthonormal basis of E.*

Proof. This orthonormal basis we consider is made of the $n(n+1)/2$ matrices: $\mathbf{E}_{i,i}$ and $\frac{1}{\sqrt{2}}(\mathbf{E}_{i,j} + \mathbf{E}_{j,i})_{i \neq j}$, where $\mathbf{E}_{i,j}$ is the matrix where the only nonzero entry is the (i,j) entry which has value 1.

First, consider $\mathbf{B}_i = \mathbf{I} + \mathbf{E}_{i,i}$, where $\mathbf{I}$ is the identity matrix. Since $\mathbf{B}_i^{-1} = \mathbf{I} - \frac{1}{2}\mathbf{E}_{i,i}$ is n-sparse, we have $\mathbf{B}_i \in \Sigma$. Since, $\mathbf{I} \in \Sigma$, we have $\mathbf{E}_{i,i} = \mathbf{B}_i - \mathbf{I} \in \Sigma - \Sigma$.

Now, consider the matrix $\mathbf{C}_{i,j} = 2\mathbf{I} + \mathbf{E}_{i,j} + \mathbf{E}_{j,i}$. This matrix is symmetric and for $\mathbf{x} = (x_1, \ldots, x_n) \in \mathbb{R}^n$, we have $\mathbf{x}^T\mathbf{C}_{i,j}\mathbf{x} = 2(\|\mathbf{x}\|_2^2 - x_i x_j) \geq 0$, so that $\mathbf{C}_{i,j}$ is semi-definite positive. We can remark that $\mathbf{C}_{i,j}$ is invertible and that its inverse is $\frac{1}{2}\mathbf{I} + \frac{1}{6}(\mathbf{E}_{i,i} + \mathbf{E}_{j,j}) - \frac{1}{3}(\mathbf{E}_{i,j} + \mathbf{E}_{j,i})$, which is $n+2$-sparse. The fact that $\mathbf{C}_{i,j}$ is invertible implies that it is definite, so that $\mathbf{C}_{i,j} \in \Sigma$. Therefore, we can write $\mathbf{E}_{i,i} + \mathbf{E}_{j,j} = \mathbf{C}_{i,j} - 2\mathbf{I} \in \Sigma - \Sigma$. Since Σ is a positive cone, multiplying this equality by $\frac{1}{\sqrt{2}}$ yields the desired result. □

Low-rank and nonsparse matrices. In (Candès et al., 2011a), the authors consider a matrix decomposition of the form $\mathbf{L} + \mathbf{S}$, where $\mathbf{L}$ is low-rank and $\mathbf{S}$ is sparse. In order to give meaning to this decomposition, one must avoid $\mathbf{L}$ to be sparse. To this end, a "nonsparsity model" for low-rank matrices was introduced.

Let E be the space of complex matrices of size $n_1 \times n_2$. Given $\mu \geq 1$ and $r \leq \min(n_1, n_2)$, let $\Sigma_{\mu,r}$ be the set of matrices of E of rank $\leq r$ satisfying the two following conditions (denoting the SVD of such a matrix by $\sum_{k=1}^{r} \sigma_k \mathbf{u}_k \mathbf{v}_k^*$, where $\sigma_k > 0$ and the $\mathbf{u}_k$ and $\mathbf{v}_k$ are unit-norm vectors) :

1. $\forall k, \|\mathbf{u}_k\|_\infty \leq \sqrt{\frac{\mu r}{n_1}}$ and $\|\mathbf{v}_k\|_\infty \leq \sqrt{\frac{\mu r}{n_2}}$.
2. Denoting $\mathbf{U}$ and $\mathbf{V}$ the matrices obtained by concatenating the vectors $\mathbf{u}_k$ and $\mathbf{v}_k$, $\|\mathbf{U}\mathbf{V}^*\|_\infty \leq \sqrt{\frac{\mu r}{n_1 n_2}}$.

These two conditions aim at "homogenizing" the entries of $\mathbf{U}$ and $\mathbf{V}$. Note that we necessarily have $\mu \geq 1$.

Lemma 3. *Let $E = \mathcal{M}_{n_1,n_2}(\mathbb{C})$ and $\Sigma_{\mu,r}$ be the subset of E containing the matrices satisfying the two above conditions (with $\mu \geq 1$ and $r \geq 1$). Then $\Sigma_{\mu,r} - \Sigma_{\mu,r}$ contains an orthonormal basis.*

Proof. Since $\Sigma_{\mu,r}$ contains the null matrix, it is sufficient to prove that $\Sigma_{\mu,r}$ contains an orthonormal basis. Let $\{\mathbf{e}_k\}_{k=1}^{n_1}$ and $\{\mathbf{f}_\ell\}_{\ell=1}^{n_2}$ be the discrete Fourier bases of $\mathbb{C}^{n_1}$ and $\mathbb{C}^{n_2}$, that is

$$\mathbf{e}_k = \frac{1}{\sqrt{n_1}}\left[1, \exp\left(\frac{2i\pi k}{n_1}\right), \ldots, \exp\left(\frac{2i\pi(n_1-1)k}{n_1}\right)\right]^T$$
$$\mathbf{f}_\ell = \frac{1}{\sqrt{n_2}}\left[1, \exp\left(\frac{2i\pi \ell}{n_2}\right), \ldots, \exp\left(\frac{2i\pi(n_2-1)\ell}{n_2}\right)\right]^T.$$

Then the $n_1 n_2$ rank-1 matrices of the form $\mathbf{e}_k\mathbf{f}_\ell^*$ are elements of $\Sigma_{\mu,r}$ since they obviously satisfy the two above conditions. But they also form an orthonormal basis of E, since each entry of $\mathbf{e}_k\mathbf{f}_\ell^*$ is of module $\frac{1}{\sqrt{n_1 n_2}}$ and that, denoting $\langle \cdot, \cdot \rangle$ the Hermitian scalar product on E, we have

$$\begin{aligned}
&\langle \mathbf{e}_k\mathbf{f}_\ell^*, \mathbf{e}_{k'}\mathbf{f}_{\ell'}^* \rangle \\
&= \sum_{u=0}^{n_1-1} \exp\left(2i\pi u \frac{k-k'}{n_1}\right) \sum_{v=0}^{n_2-1} \exp\left(2i\pi v \frac{\ell-\ell'}{n_2}\right) \\
&= \delta_k^{k'}\,\delta_\ell^{\ell'},
\end{aligned}$$

proving that these matrices form an orthonormal basis of E. □

3.6 The NSP and its relationship with the RIP

In the following section, we consider $E = G$ (and $\mathbf{A} = \mathbf{I}$). As we have seen, one cannot expect to get ℓ_2/ℓ_2 instance optimality in a dimensionality reduction context. This raises the following question: given pseudo-norms $\|\cdot\|_G$ and $\|\cdot\|_F$ defined respectively on E and F, is there a pseudo-norm $\|\cdot\|_E$ such that IOP holds? We will see that this property is closely related to the RIP on $\mathbf{M}$.

3.6.1 Generalized RIP and its necessity for robustness

The Restricted Isometry Property is a widely-used property on the operator $\mathbf{M}$ which yields nice stability and robustness results on the recovery of vectors from their compressive measurements. In the usual CS framework, the RIP provides a relation of the form $(1-\delta)\|\mathbf{x}\|_G \leq \|\mathbf{M}\mathbf{x}\|_F \leq (1+\delta)\|\mathbf{x}\|_G$ for any vector $\mathbf{x}$ in Σ_{2k}. The norms $\|\cdot\|_G$ and $\|\cdot\|_F$ are usually both taken as the ℓ_2-norm. A form of RIP can easily be stated in a generalized framework: we will say that $\mathbf{M}$ satisfies the RIP on $\Sigma - \Sigma$ if there exists positive constants α, β such that

$$\forall \mathbf{z} \in \Sigma - \Sigma, \alpha\|\mathbf{z}\|_G \leq \|\mathbf{M}\mathbf{z}\|_F \leq \beta\|\mathbf{z}\|_G. \tag{3.30}$$

Similarly to the sparse case, it is possible to make a distinction between *lower-RIP* (left inequality) and *upper-RIP* (right inequality). Let's remark that this definition has been stated for vectors of $\Sigma - \Sigma$: this choice is justified by the links between this formulation and the NSP, which will be discussed later in this section. Let's also note that this form of RIP encompasses several generalized RIP previously proposed: the $\mathbf{\Omega}$-RIP (Giryes et al., 2013), the $\mathbf{D}$-RIP (Candès et al., 2011) and the Union of Subspaces RIP (Blumensath, 2011).

Let's now suppose the existence of decoders robust to noise, that is for all $\delta > 0$, (3.19) is satisfied for a certain Δ_δ. This property implies the Robust NSP (3.20) with the same constants according to Theorem 5. By considering $\mathbf{h} \in \Sigma - \Sigma$, the Robust NSP reads:

$$\forall \mathbf{h} \in \Sigma - \Sigma, \|\mathbf{h}\|_G \leq D_2\|\mathbf{M}\mathbf{h}\|_F. \tag{3.31}$$

This is the lower-RIP on $\Sigma - \Sigma$, with constant $1/D_2$. The stability to noise therefore implies the lower-RIP on the set of differences of vectors of Σ, which is therefore necessary if one seeks the existence of a decoder robust to noise.

3.6.2 M-norm instance optimality with the RIP

The lower-RIP is necessary for the existence of a Robust instance optimal decoder, but what can we say this time if we suppose that $\mathbf{M}$ satisfies the lower-RIP on $\Sigma - \Sigma$ with constant α, that is $\forall \mathbf{z} \in \Sigma - \Sigma, \alpha\|\mathbf{z}\|_G \leq \|\mathbf{M}\mathbf{z}\|_F$? We will prove that in both the noiseless and the noisy cases, this implies the IOP with norms $\|\cdot\|_G$ and $\|\cdot\|_\mathbf{M}$, the latter being called "M-norm"[c] and involving $\|\cdot\|_G$ and $\|\cdot\|_F$.

Let's define the M-norm on E as the following quantity, extending its definition for ℓ_2 norms in (Peleg et al., 2013) and its implicit appearance in the proof of early results of the field (Candès, 2008):

$$\forall \mathbf{x} \in E, \|\mathbf{x}\|_\mathbf{M} = \|\mathbf{x}\|_G + \frac{1}{\alpha}\|\mathbf{M}\mathbf{x}\|_F. \tag{3.32}$$

[c] to highlight its dependency on the Measurement operator

Note that the term M-norm should be understood as M-pseudo-norm in the general case: if $\|\cdot\|_F$ and $\|\cdot\|_G$ satisfy the properties listed in Table 3.1, then $\|\cdot\|_{\mathbf{M}}$ satisfies the same properties as $\|\cdot\|_G$. However, when $\|\cdot\|_G$ and $\|\cdot\|_F$ are norms, $\|\cdot\|_{\mathbf{M}}$ is also a norm. We will note $d_{\mathbf{M}}(.,.)$ its associated (pseudo-)distance. The following theorem states that this $\|\cdot\|_{\mathbf{M}}$ allows one to derive an NSP from the lower-RIP on $\Sigma - \Sigma$.

Theorem 9. *Let's suppose that* $\mathbf{M}$ *satisfies the lower-RIP on* $\Sigma - \Sigma$ *with constant* α *(left inequality of* (3.30)*). Then the following Robust NSP is satisfied:*

$$\forall \mathbf{h} \in E, \|\mathbf{h}\|_G \leq d_{\mathbf{M}}(\mathbf{h}, \Sigma - \Sigma) + \frac{1}{\alpha}\|\mathbf{M}\mathbf{h}\|_F. \tag{3.33}$$

In particular, the following regular NSP is satisfied:

$$\forall \mathbf{h} \in \mathcal{N}, \|\mathbf{h}\|_G \leq d_{\mathbf{M}}(\mathbf{h}, \Sigma - \Sigma). \tag{3.34}$$

Therefore, if $\mathbf{M}$ satisfies the lower-RIP on $\Sigma - \Sigma$ with constant α, then for all $\delta > 0$, there exists a noise-robust instance optimal decoder Δ_δ satisfying the following property (Theorem 6): $\forall \mathbf{x} \in E, \forall \mathbf{e} \in F$,

$$\|\mathbf{x} - \Delta_\delta(\mathbf{M}\mathbf{x} + \mathbf{e})\|_G \leq 2d_{\mathbf{M}}(\mathbf{x}, \Sigma) + \frac{2}{\alpha}\|\mathbf{e}\|_F + \delta. \tag{3.35}$$

Note that in (Blumensath, 2011), the author explored the implication of a lower-RIP on $\Sigma - \Sigma$ for the case where Σ is an arbitrary UoS and $\|.\|_G/\|.\|_F$ are the ℓ_2 norm. He proved that this generalized lower-RIP implies the following IOP: for all $\delta > 0$, there exists a decoder Δ_δ such that $\forall \mathbf{x} \in E, \forall \mathbf{e} \in F, \forall \mathbf{z} \in \Sigma$,

$$\|\mathbf{x} - \Delta_\delta(\mathbf{M}\mathbf{x} + \mathbf{e})\|_2 \leq \|\mathbf{x} - \mathbf{z}\|_2 + \frac{2}{\alpha}\|\mathbf{M}(\mathbf{x} - \mathbf{z}) + \mathbf{e}\|_2 + \delta. \tag{3.36}$$

In this set-up, the instance optimality in equation (3.35) can be reformulated as: $\forall \mathbf{x} \in E, \forall \mathbf{e} \in F, \forall \mathbf{z} \in \Sigma$,

$$\|\mathbf{x} - \Delta_\delta(\mathbf{M}\mathbf{x} + \mathbf{e})\|_2 \leq 2\|\mathbf{x} - \mathbf{z}\|_2 + \frac{2}{\alpha}\|\mathbf{M}(\mathbf{x} - \mathbf{z})\|_2 + \frac{2}{\alpha}\|\mathbf{e}\|_2 + \delta. \tag{3.37}$$

Comparing these two instance optimality results, we can remark that the one in (Blumensath, 2011) is slightly tighter. This is merely a consequence of the difference in our method of proof, as we add the NSP as an intermediate result to prove instance optimality. The upper bound in (Blumensath, 2011) can also be derived in our case with the same proof layout if we suppose the lower-RIP. Compared to (Blumensath, 2011), our theory deals with general (pseudo-)norms and sets Σ beyond Union of Subspaces.

3.6.3 Upper-bound on the M-norm by an atomic norm

As we have seen, provided a *lower-RIP* on $\Sigma - \Sigma$, an NSP can be derived with the M-norm as $\|\cdot\|_E$. However, this may look like a tautology since the M-norm explicitly depends on $\mathbf{M}$. Hence, one may wonder if this NSP is of any use. We will prove in the following that provided an *upper-RIP* on a certain cone Σ' (which can be taken as $\mathbb{R}\Sigma$), a more natural upper bound can be derived by bounding the M-norm with an atomic norm (Chandrasekaran et al., 2012). In particular, this type of inequality applied to the usual k-sparse vectors and low-rank matrices models give, under standard RIP conditions, instance optimality upper bounds with typical norms.

We will suppose in this section that $\|\cdot\|_G$ is a norm.

The atomic norm $\|\cdot\|_{\Sigma'}$

Let Σ' be a subset of E and let E' be the closure of span(Σ') with respect to the norm $\|\cdot\|_G$. For $\mathbf{x} \in E'$, one can define the "norm" $\|\mathbf{x}\|_{\Sigma'}$ by:

$$\|\mathbf{x}\|_{\Sigma'} = \inf \left\{ \sum_{k=0}^{+\infty} \|\mathbf{x}_k\|_G \mid \forall k, \mathbf{x}_k \in \mathbb{R}\Sigma' \text{ and } \|\mathbf{x} - \sum_{k=0}^{K} \mathbf{x}_k\|_G \to_{K\to+\infty} 0 \right\}. \tag{3.38}$$

Remark that there may be some vectors $\mathbf{x}$ for which $\|\mathbf{x}\|_{\Sigma'} = +\infty$, if $\sum_{k=0}^{+\infty} \|\mathbf{x}_k\|_G = +\infty$ for any decomposition of $\mathbf{x}$ as an infinite sum of elements of $\mathbb{R}\Sigma'$. However, the set $V = \{\mathbf{x} \in E | \|\mathbf{x}\|_{\Sigma'} < +\infty\}$ is a normed subspace of E which contains Σ' (DeVore and Temlyakov, 1996). In the following, we assume that $V = E$. Note that this norm can be linked to atomic norms defined in (Chandrasekaran et al., 2012) by considering $\mathcal{A}$ as the set of normalized elements of Σ' with respect to $\|\cdot\|_G$.

Now suppose $\mathbf{M}$ satisfies an upper-RIP on Σ', so that

$$\forall \mathbf{x}' \in \Sigma', \|\mathbf{M}\mathbf{x}'\|_F \leq \beta \|\mathbf{x}'\|_G. \tag{3.39}$$

For $\mathbf{x} \in E$ admitting a decomposition $\sum_{k=0}^{+\infty} \mathbf{x}_k$ on $\mathbb{R}\Sigma'$, we can therefore upper bound $\|\mathbf{M}\mathbf{x}\|_F$ by $\sum_{k=0}^{+\infty} \|\mathbf{M}\mathbf{x}_k\|_F \leq \beta \sum_{k=0}^{+\infty} \|\mathbf{x}_k\|_G$. This inequality is valid for any decomposition of $\mathbf{x}$ as a sum of elements of $\mathbb{R}\Sigma'$, so that $\|\mathbf{M}\mathbf{x}\|_F \leq \beta \|\mathbf{x}\|_{\Sigma'}$. Therefore, under these hypotheses,

$$\forall \mathbf{x} \in E, \|\mathbf{x}\|_{\mathbf{M}} \leq \|\mathbf{x}\|_G + \frac{\beta}{\alpha}\|\mathbf{x}\|_{\Sigma'} \leq \left(1 + \frac{\beta}{\alpha}\right) \|\mathbf{x}\|_{\Sigma'}. \tag{3.40}$$

In particular, we have the following result:

Theorem 10. *Suppose* $\mathbf{M}$ *satisfies the lower-RIP on* $\Sigma - \Sigma$ *with constant* α *and the upper-RIP on* Σ *with constant* β*, that is*

$$\forall \mathbf{x} \in \Sigma - \Sigma, \alpha\|\mathbf{x}\|_G \leq \|\mathbf{M}\mathbf{x}\|_F \tag{3.41}$$

and

$$\forall \mathbf{x} \in \Sigma, \|\mathbf{M}\mathbf{x}\|_F \leq \beta\|\mathbf{x}\|_G. \tag{3.42}$$

Then for all $\delta > 0$*, there exists a decoder* Δ_δ *satisfying:* $\forall \mathbf{x} \in E, \forall \mathbf{e} \in F$,

$$\|\mathbf{x} - \Delta_\delta(\mathbf{M}\mathbf{x} + \mathbf{e})\|_G \leq 2\left(1 + \frac{\beta}{\alpha}\right) d_\Sigma(\mathbf{x}, \Sigma) + \frac{2}{\alpha}\|\mathbf{e}\|_E + \delta, \tag{3.43}$$

where d_Σ *is the distance associated to the norm* $\|\cdot\|_\Sigma$.

Remark 3. *Note that these results can be extended with relative ease to the case where* $\|\cdot\|_G$ *is not necessarily homogeneous but p-homogeneous, that is* $\|\lambda\mathbf{x}\|_G = |\lambda|^p\|\mathbf{x}\|_G$.

Study of $\|\cdot\|_\Sigma$ in two usual cases

We now provide a more thorough analysis of the norm $\|\cdot\|_\Sigma$ for usual models which are sparse vectors and low-rank matrices. In particular, we give a simple equivalent of this norm involving usual norms in the case where $\|\cdot\|_G = \|\cdot\|_2$ (for matrices, this is the Frobenius norm).

The norm $\|\cdot\|_\Sigma$ relies on the decomposition of a vector as a sum of elements of $\mathbb{R}\Sigma$. When Σ is the set of k-sparse vectors or the set or matrices of rank k, there are particular decompositions of this type:

- In the case where Σ is the set of k-sparse vectors, a vector $\mathbf{x}$ can be decomposed as $\sum_{j=1}^{\infty} \mathbf{x}_j$, where all $\mathbf{x}_j$ are k-sparse vectors with disjoint supports, which are eventually zero, and such that any entry of $\mathbf{x}_j$ does not exceed any entry of $\mathbf{x}_{j-1}$ (in magnitude). This is a decomposition of $\mathbf{x}$ into disjoint supports of size k with a nonincreasing constraint on the coefficients.

- Similarly, in the case where Σ is the set of matrices of rank k and $\mathbf{N}$ is a matrix, the SVD of $\mathbf{N}$ gives a decomposition of the form $\mathbf{N} = \sum_{j=1}^{\infty} \mathbf{N}_j$, where the $\mathbf{N}_j$ are rank k, eventually zero matrices such that any singular value of $\mathbf{N}_j$ does not exceed any singular value of $\mathbf{N}_{j-1}$.

For $j \geq 2$, we can upper bound the quantity $\|\mathbf{x}_j\|_2$ using the assumption on the particular decomposition: $\|\mathbf{x}_j\|_2 \leq \sqrt{k}\|\mathbf{x}_j\|_\infty \leq \sqrt{k}\frac{\|\mathbf{x}_{j-1}\|_1}{k} = \frac{\|\mathbf{x}_{j-1}\|_1}{\sqrt{k}}$. Similarly, $\|\mathbf{N}_j\|_2 \leq \frac{\|\mathbf{N}_{j-1}\|_*}{\sqrt{k}}$, where $\|\cdot\|_*$ is the trace norm, defined as the sum of singular values. We can therefore, in both cases, upper bound the norm $\|\cdot\|_\Sigma$. In the case of k-sparse vectors, this gives:

$$\|\mathbf{x}\|_\Sigma \leq \|\mathbf{x}_1\|_2 + \sum_{j\geq 1} \frac{\|\mathbf{x}_j\|_1}{\sqrt{k}} \leq \|\mathbf{x}\|_2 + \frac{\|\mathbf{x}\|_1}{\sqrt{k}}. \tag{3.44}$$

In the case of matrices of rank k, this gives:

$$\|\mathbf{N}\|_\Sigma \leq \|\mathbf{N}_1\|_2 + \sum_{j\geq 1} \frac{\|\mathbf{N}_j\|_1}{\sqrt{k}} \leq \|\mathbf{N}\|_F + \frac{\|\mathbf{N}\|_*}{\sqrt{k}}. \tag{3.45}$$

We can also upper bound the right hand side of these equations by $\mathcal{O}(\|\cdot\|_\Sigma)$ with a small constant, which will prove that the norms defined in these equations are of the same order. Indeed, a simple application of the triangle inequality gives us first that $\|\mathbf{x}\|_2 \leq \|\mathbf{x}\|_\Sigma$ and $\|\mathbf{N}\|_F \leq \|\mathbf{N}\|_\Sigma$. Then, considering a decomposition of $\mathbf{x}$ as a sum of k-sparse vectors $\sum_{j\geq 1} \mathbf{x}_j$, we get

$$\frac{\|\mathbf{x}\|_1}{\sqrt{k}} \leq \sum_{j\geq 1} \frac{\|\mathbf{x}_j\|_1}{\sqrt{k}} \leq \sum_{j\geq 1} \|\mathbf{x}_j\|_2 \tag{3.46}$$

(indeed, each $\mathbf{x}_j$ can be viewed as a k-dimensional vector and we have for such a vector $\|\mathbf{x}_j\|_1 \leq \sqrt{k}\|\mathbf{x}_j\|_2$). Similarly,

$$\frac{\|\mathbf{N}\|_*}{\sqrt{k}} \leq \sum_{j\geq 1} \|\mathbf{N}_j\|_F. \tag{3.47}$$

Since these upper bounds are satisfied for any decomposition, they can be replaced respectively by $\|\mathbf{x}\|_\Sigma$ and $\|\mathbf{N}\|_\Sigma$. Finally, we have

$$\|\mathbf{x}\|_2 + \tfrac{\|\mathbf{x}\|_1}{\sqrt{k}} \leq 2\|\mathbf{x}\|_\Sigma \quad \text{and} \quad \|\mathbf{N}\|_F + \tfrac{\|\mathbf{N}\|_*}{\sqrt{k}} \leq 2\|\mathbf{N}\|_\Sigma. \tag{3.48}$$

In these two cases, the norm $\|\cdot\|_\Sigma$ is therefore equivalent (with constants bounded by 2) to the norm $\|\cdot\|_2 + \frac{\|\cdot\|_1}{\sqrt{k}}$ (which is $\|\cdot\|_F + \frac{\|\cdot\|_*}{\sqrt{k}}$ for matrices).

We have thus shown:

Lemma 4. *When Σ is the set of k-sparse vectors, the norm $\|\cdot\|_\Sigma$ satisfies*

$$\|\cdot\|_\Sigma \leq \|\cdot\|_2 + \frac{\|\cdot\|_1}{\sqrt{k}} \leq 2\|\cdot\|_\Sigma. \tag{3.49}$$

When Σ is the set of rank-k matrices, it satisfies

$$\|\cdot\|_\Sigma \leq \|\cdot\|_F + \frac{\|\cdot\|_*}{\sqrt{k}} \leq 2\|\cdot\|_\Sigma. \tag{3.50}$$

We can therefore remark that for these two standard models, the norm $\|\cdot\|_\Sigma$ can easily be upper bounded by usual norms under RIP conditions, yielding an IOP with a usual upper bound. We can also note that stronger RIP conditions can yield a stronger result: in (Candès, 2008), the author proves that under upper and lower-RIP on $\Sigma - \Sigma$ with Σ being the set of k-sparse vectors, an instance optimal decoder can be defined as the minimization of a convex objective: the ℓ_1 norm, which appears as strongly connected to the norm $\|\cdot\|_\Sigma$. One may then wonder if a generalization of such a result is possible: when can an instance optimal decoder be obtained by solving a convex minimization problem with a norm related to $\|\cdot\|_\Sigma$?

3.7 Discussion and outlooks on Instance Optimality

Let's now review the results presented in this chapter and give some insights on interesting future work. As has been detailed throughout the chapter, Instance Optimality is a property presenting several benefits:

- It can be defined in a very general framework, for any signal space, signal model and pseudo-norms, as well as for both noiseless and noisy settings.
- It is a nice uniform formulation of the "good behavior" of a decoder and thus of the well-posedness of an inverse problem.
- It can be linked to Null Space Property and Restricted Isometry Property, which provide necessary and/or sufficient conditions for the existence of an instance optimal decoder.

We now present some immediate outlooks and interesting open questions related to Instance Optimality and to the results presented in this paper.

3.7.1 Condition for the well-posedness of the "optimal" decoder.

We have seen that for general models Σ, an additionnal term δ appears in the right hand side term of the instance optimality inequality ((3.13),(3.19)), reflecting the fact that the minimal distance of the "optimal" decoder (B.7) may not be reached at a specific point. However, as mentioned in Property 1, this additive constant can be dropped in the noiseless case provided $\Sigma + \mathcal{N}$ is a closed set. One can then wonder if there exists a similar condition (e.g., a sort of local compactness property) in the noisy case for which one can drop the constant δ and get a more usual instance optimality result.

3.7.2 Compressed graphical models.

As has been mentioned in Section 3.2.4, the case where Σ is the set of symmetric definite positive square matrices with sparse inverse is related to high-dimensional Gaussian graphical models. In Lemma 2, we showed this type of models fits in our theory since we could apply Theorem 8 in this case, proving the impossibility of ℓ_2/ℓ_2 IOP in a dimension-reduction case. Yet, as for other signal models, can Gaussian graphical models satisfy some IOP/NSP with different norms in a compressive framework?

3.7.3 Guarantees for signal-space reconstructions and more.

When $\mathbf{D}$ is a redundant dictionary of size $d \times n$ and the signals of interest are vectors of the form $\mathbf{z} = \mathbf{D}\mathbf{x}$, where $\mathbf{x}$ is a sparse vector, traditional reconstruction guarantees from $\mathbf{y} = \mathbf{M}\mathbf{z}$ assume the RIP on the matrix $\mathbf{MD}$. This is often too restrictive: for example when $\mathbf{D}$ has strongly correlated columns, failure to identify $\mathbf{x}$ from $\mathbf{y}$ does not necessarily prevent one from correctly estimating $\mathbf{z}$. Recent work on *signal-space* algorithms (Davenport et al., 2013) has shown that the $\mathbf{D}$-RIP assumption on $\mathbf{M}$ is in fact sufficient.

The framework presented in this paper offers two ways to approach this setting:

- Considering $\Sigma = \Sigma_k$ as the set of k-sparse vectors of dimension n and $\mathbf{A} = \mathbf{D}$, the upper bound on the reconstruction error is of the form $d_E(\mathbf{x}, \Sigma_k)$. Signal-space guarantees can be envisioned by choosing a metric $\|\cdot\|_E = \|\mathbf{D}\cdot\|$.

- Considering $\Sigma = \mathbf{D}\Sigma_k$ as the set of d-dimensional vectors that have a k-sparse representation in the dictionary $\mathbf{D}$ and $\mathbf{A} = \mathbf{I}$, the upper bound is of the form $d'(\mathbf{z}, \mathbf{D}\Sigma_k)$.

In (Needell and Ward, 2013), the authors propose a result similar to instance optimality by upper bounding, for a Total Variation decoder, the reconstruction error of an image $\mathbf{X}$ from compressive measurement by a quantity involving $d_1(\nabla\mathbf{X}, \Sigma_k)$, where ∇ is the gradient operator, Σ_k the k-sparse union of subspaces (in the gradient space) and d_1 is the ℓ_1 distance. This quantity is therefore the distance between the gradient of the image and the k-sparse vectors model. Can such a bound be interpreted in our framework, and possibly be generalized to other types of signals?

3.7.4 Task-oriented decoders versus general purpose decoders.

We already mentioned two very different application set-ups, in medical imaging and audio source separation, where only parts of the original signals need to be recovered. One can think of other, more dramatic, cases where only task-oriented linear features should be reconstructed. One such situation is met in current image classification work-flows. Indeed, most recent state-of-art image classification methods rely on very high-dimensional image representation (e.g., so called Fisher vectors, of dimension ranging from 10,000 to 200,000) and conduct supervised learning on such labeled signals by means of linear SVMs (Sanchez et al., 2013). Not only this approach yields top-ranking performance in terms of classification accuracy on challenging image classification benchmarks, but it also permits very large scale learning thanks to the low complexity of linear SVM training and its efficient implementations, e.g., with stochastic gradient descent. For each visual category to recognize, a linear classifier $\mathbf{w}$ is learned, which associates to an input image with representation $\mathbf{x}$ the score $\mathbf{w}^T\mathbf{x}$. The single or multiple labels that are finally assigned to $\mathbf{x}$ by the system depend on the scores provided by all trained classifiers (typically from 10 to 100), hence on a vector of the form $\mathbf{A}\mathbf{x}$, where each row of $\mathbf{A}$ is one one-vs-all linear SVM. In this set-up, the operator $\mathbf{A}$ implies a dramatic dimension reduction. For very large scale problems of this type, storing and manipulating original image signatures in the database can become intractable. The theoretical framework proposed in this paper might help designing new solutions to this problem in the future. In particular, it will provide tools to answer the following questions:

- $\mathbf{A}$ being given (learned on a labeled subset of the database): can one design a compressive measurement operator $\mathbf{M}$ such that the "classifiers" scores can be recovered

directly from the compressed image signature $\mathbf{Mx}$, hence avoiding the prior reconstruction of the high-dimensional signal $\mathbf{x}$?

- $\mathbf{M}$ being given ("legacy" compressed storing of image signatures): what are the linear classifier collections that can be precisely emulated in the compressed domain thanks to a good decoder Δ?

Note that this classification-oriented set-up might call for a specific norm $\|\cdot\|_G$ on the output of a linear score bank.

Another important domain of application that might benefit from both aspects (general purpose and task-oriented) of our work is data analysis under privacy constraints. Two scenarii can be envisioned, where our framework could help decide whether or not such constraints are compatible with the analysis of interest:

- *General purpose scenario*: given a linear measurement operator $\mathbf{M}$ of interest for further analysis, can one guarantee that there is no decoder permitting good enough recovery of original signals?

- *Task-oriented scenario*: the operator $\mathbf{M}$ serving as a means to obfuscate original signals such that critical information can't be recovered, let's consider a specific analysis task on original signals requiring the application of the feature extractor $\mathbf{A}$. Can this task be implemented on obfuscated signals instead, via a good decoder Δ, hence in a privacy-preserving fashion?

3.7.5 Worst case versus average case instance optimality.

The raw concept of Instance Optimality has a major drawback: the uniformity of the bound may impose, in some settings, a large global instance optimality constant whereas the inverse problem is well posed for the vast majority of signals. Let's consider the example depicted in Figure 3.5, where the signal space E is of dimension 2, the signal model Σ is a point cloud mostly concentrated along the line $\mathcal{D}$ and the measurement operator $\mathbf{M}$ is the orthogonal projection on D. The figure depicts the ratio (approximation error)/(distance to model) for each $\mathbf{x} \in \mathbb{R}^2$. The optimal constant, which is the supremum of these ratios, is infinite: the ratio actually goes to infinity in the vicinity of the point p. However, for the vast majority of vectors, the ratio is rather low (the blue section covers most of the space).

An interesting outlook to circumvent this pessimistic "worst-case" phenomenon is to consider a probabilistic formulation of instance optimality, as in (Cohen et al., 2009): given Ω a probability space with probability measure P, and considering $\mathbf{M}$ as a random variable on Ω, is there a decoder $\Delta(.|\mathbf{M})$ (which computes an estimate given the observation *and* the particular draw of the measurement operator $\mathbf{M}$) such that for any $\mathbf{x} \in E$, the instance optimality inequality

$$\|\mathbf{x} - \Delta(\mathbf{Mx}|\mathbf{M})\|_G \leq C d_E(\mathbf{x}, \Sigma) \tag{3.51}$$

holds with high probability on the drawing of $\mathbf{M}$? A particular challenge would be to understand in which dimension reduction scenarii there exists both a probability measure and a decoder with the above property. Another possible formulation of probabilistic instance optimality is to define a probability distribution on the signal space and to upper bound the average reconstruction error of the vectors, as in (Yu and Sapiro, 2011).

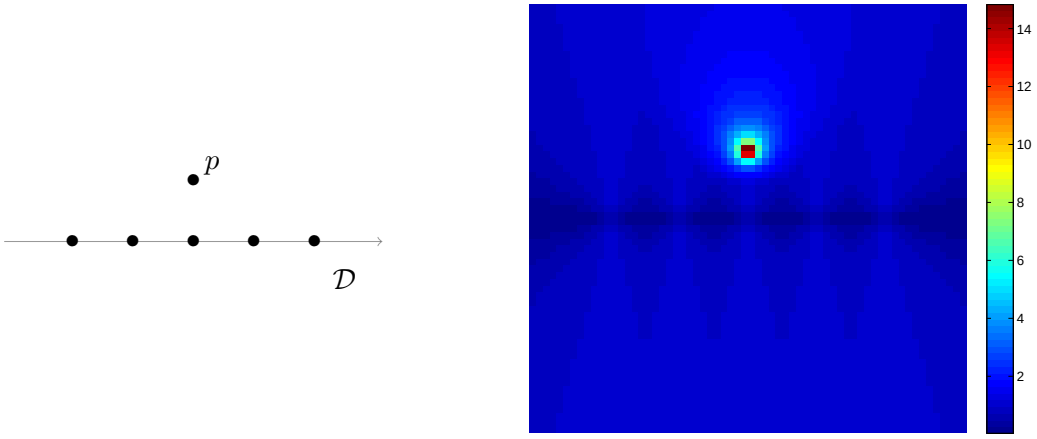

Figure 3.5: Drawback of uniform instance optimality in a simple case: the model Σ (*Left*) is the set of black points including those on $\mathcal{D}$ and the point p and the operator $\mathbf{M}$ is the 1-dimensional orthogonal projection on the horizontal axis. If we choose Δ as the pseudo-inverse of $\mathbf{M}$, the depicted IO ratio (*Right*) is low on most of the space, but the uniform constant is infinite.

Chapter 4

Explicit embeddings for nearest neighbor search

As has been mentioned in Chapters 1 and 2 and pictured in Figure 2.1, individual dimension reduction of vectors can be used to reduce the cost of a learning task. In this chapter, we will consider a problem which is not itself a learning problem as the beginning of Chapter 2 depicts them[a], but which can constitute a particular step in some global learning task.

The envisioned task is nearest neighbor (NN) search. Let Ω be a set supplied with a distance d and $\mathcal{X} = \{\mathbf{x}_1, \dots, \mathbf{x}_L\}$ be a database of elements of Ω. Given a new element $\mathbf{q} \in \Omega$, typically called a *query*, a nearest neighbor of $\mathbf{q}$ in $\mathcal{X}$ is a solution of the problem

$$\operatorname*{argmin}_{\mathbf{x} \in \mathcal{X}} d(\mathbf{q}, \mathbf{x}). \tag{4.1}$$

Usually, this solution will be unique.

Such a problem arises in applications such as image search (Sivic and Zisserman, 2003; Torralba et al., 2008) and classification (Grauman and Darrell, 2005; Boiman et al., 2008; Deng et al., 2009; Perronnin et al., 2010). In such practical applications, one is usually interested in computing not only the nearest neighbor of a query but its k nearest neighbors. In this chapter, we will develop methods and perform experiments in a framework where we only seek the nearest neighbor of a query for simplicity, but the contributions described in this chapter can be easily extended to the case where one searches for several nearest neighbors of a query.

Approximate search. The most basic way to find a nearest neighbor is to compute all the distances $d(\mathbf{q}, \mathbf{x})$ for all $\mathbf{x} \in \mathcal{X}$ and to find the minimal values. However, these computations come at a cost of $\mathcal{O}(LC_d)$, where C_d is the cost of computing one distance value between two elements of Ω. This cost can be prohibitive when L is large, which motivates the search for less costly approximate nearest neighbor (ANN) search schemes. These schemes typically rely on the computation of small-dimensional *signatures*, which are compressed representations of the elements of Ω. The signatures will be outputs of an encoding function $s : \Omega \to S$, where the signature space S is provided with a distance which is much faster to compute than the distance of Ω.

The ANN search of a query $\mathbf{q}$ is therefore performed in this signature space. Note that as discussed in Section 1.2.1, an obvious way of using these signatures is to perform a full

[a] That is a problem where one wants to infer some parameters from a training data set.

NN search in the signature space by computing all the distances between the signature of the query and the signatures of the database elements. In this chapter, we will consider methods applied in this "full-search" framework.

An ANN search pipeline will typically divide into *offline* steps, which will regroup the "fixed costs" and prepare the database $\mathcal{X}$ to the actual search step, and *online* steps, which are the steps where one gets a query $\mathbf{q}$ and needs to perform the search. The offline meta-procedure will usually be as follows:

- *Choice of the encoding function*: The encoding function s which will compute signatures is determined. In methods such as LSH, this step is generally quick since the function is a concatenation of randomly chosen hash functions. However, as we will see in Section 4.1, the choice of the signatures can be the result of a learning step involving the data, in order to choose adequate signatures.

- *Computation of the signatures*: The signatures $s(\mathbf{x}_1), \ldots, s(\mathbf{x}_L)$ are computed and stored. For a fixed database, this operation is only required to be performed once, since the same signatures can be used to perform as many NN searches as needed.

Given a query $\mathbf{q}$ for which one wants to perform an ANN search among the elements in $\mathcal{X}$, the online meta-procedure will be as follows:

- *Full NN search in the signature space*: The signature $s(\mathbf{q})$ is computed, and the distances between $s(\mathbf{q})$ and every $s(\mathbf{x}_r)$ is computed. The N nearest neighbors in the signature space are identified, where N is a predetermined integer, supposedly far smaller than the size of $\mathcal{X}$. This step may be costly since it linearly scales with the size of $\mathcal{X}$, but the signatures are chosen so that it is far less costly than a full search in the space Ω.

  As an alternative, in the case where $S \subset \Omega$, one can perform the full NN search in an asymmetric way, that is by computing the distances $d(\mathbf{q}, s(\mathbf{x}_r))$ instead of the distances $d(s(\mathbf{q}), s(\mathbf{x}_r))$. This case, although non-intuitive at first glance, will be instantiated in this chapter via the Product Quantization (Jégou et al., 2011), in which the authors argue that an asymmetric scheme yields better precision than a symmetric scheme.

- *Reranking*: The N approximate nearest neighbors identified in the previous step are then reranked, that is the effective distances between $\mathbf{q}$ and the corresponding $\mathbf{x}_i$ are computed. If the signature function s and the number N are well designed, the nearest neighbor of $\mathbf{q}$ is likely to be among these N elements.

In this case, if C_d is the cost of computing a distance in Ω and C_s is the cost of computing a distance in the signature space (one should have $C_s \ll C_d$), then the full-search cost in Ω is $\mathcal{O}(LC_d)$ whereas the cost of the described approximate scheme is $\mathcal{O}(LC_s + NC_d)$. The approximate scheme will yield computational savings if

$$\frac{LC_s + NC_d}{LC_d} = \frac{C_s}{C_d} + \frac{N}{L} \ll 1. \tag{4.2}$$

In practice, these two fractions are $\ll 1$, so that the approximate scheme yields substantial complexity reduction.

ANN search methods. Significant progress has been achieved on ANN search in the last decade, in particular with the substantial contribution of the LSH approach (Datar et al., 2004b), described in Chapter 1. Variants of this method have been developed, consisting of different types of low-dimensional signatures (Muja and Lowe, 2009; Weiss et al., 2008; Jégou et al., 2011). These methods typically allows ANN search systems to scale to datasets comprising 10^9 elements.

However, the aforementioned methods only consider simple metrics such as ℓ_p norms, and mainly the Euclidean norm. In this chapter, we will consider the case where the distance d derives from a Mercer kernel, which is a broad generalization of the case where the distance is Euclidean. This case has less been studied in the ANN literature, although kernels provide a wide range of distances which are more elaborate than the Euclidean distance. Kernels are reviewed in the next paragraph.

Mercer kernels and nearest neighbors. Let's first recall standard definitions and properties of Mercer kernels (Schölkopf and Smola, 2002). A positive semi-definite (PSD) kernel on a space Ω is an application $K : \Omega^2 \to \mathbb{R}$ such that for any collection $\mathcal{X} = \{\mathbf{x}_1, \ldots, \mathbf{x}_L\}$ of elements of Ω, the Gram matrix

$$\mathbf{K}(\mathcal{X}) = \begin{pmatrix} K(\mathbf{x}_1, \mathbf{x}_1) & \ldots & K(\mathbf{x}_1, \mathbf{x}_L) \\ \vdots & & \vdots \\ K(\mathbf{x}_L, \mathbf{x}_1) & \ldots & K(\mathbf{x}_L, \mathbf{x}_L) \end{pmatrix} \tag{4.3}$$

is symmetric semi-definite positive. For any such kernel, there exists a unique (up to isomorphism) Hilbert space $\mathcal{H}$ and application $\Psi : \Omega \to \mathcal{H}$ such that $\forall \mathbf{x}, \mathbf{y} \in \Omega$,

$$K(\mathbf{x}, \mathbf{y}) = \langle \Psi(\mathbf{x}), \Psi(\mathbf{y}) \rangle_{\mathcal{H}}, \tag{4.4}$$

where $\langle \cdot, \cdot \rangle_{\mathcal{H}}$ is the scalar product of $\mathcal{H}$. $\mathcal{H}$ is usually named *implicit space*, since it is a space which allows the implicit interpretation of the action of K as a scalar product, provided one knows the *implicit* function Ψ which embeds Ω into $\mathcal{H}$. Usually, one does not know this function but some interesting properties can still be derived, relying on the so-called *kernel trick*: one does not need to explicitly know the values of $\Phi(\mathbf{x})$ and $\Phi(\mathbf{y})$ to compute their scalar product in $\mathcal{H}$, since this scalar product is equal to $K(\mathbf{x}, \mathbf{y})$.

In particular, K induces a distance d_K on Ω by posing

$$d_K(\mathbf{x}, \mathbf{y}) = K(\mathbf{x}, \mathbf{x}) + K(\mathbf{y}, \mathbf{y}) - 2K(\mathbf{x}, \mathbf{y}). \tag{4.5}$$

Let's notice that if we define

$$S_K(1) = \{\mathbf{x} \in \Omega : K(\mathbf{x}, \mathbf{x}) = 1\}, \tag{4.6}$$

then for all $\mathbf{x}, \mathbf{y} \in S_K(1)$,

$$d_K(\mathbf{x}, \mathbf{y}) = 2\left(1 - K(\mathbf{x}, \mathbf{y})\right). \tag{4.7}$$

This property ensures that when considering data $\mathcal{X} \subset S_K(1)$ and a query $\mathbf{q} \in S_K(1)$, the problem (4.1) with $d = d_K$ is equivalent to the following problem:

$$\operatorname*{argmax}_{\mathbf{x} \in \mathcal{X}} K(\mathbf{q}, \mathbf{x}), \tag{4.8}$$

that is searching for the nearest neighbor in the sense of d_K is equivalent to searching for the most correlated vector in the sense of K. This constitutes a good incentive to

normalize data, that is project it onto $S_K(1)$, when it is possible: it allows one to consider kernels instead of distances. This will be used in our contribution, where we will perform our experiments on normalized data.

In the particular case where Ω is a compact Hausdorff space provided with a finite measure μ with support Ω, the kernel K is a Mercer kernel (König, 1986), that is $\mathcal{H} = \ell^2(\mathbb{R})$ and for all $i \in \mathbb{N}$, there exists $\lambda_i \in \mathbb{R}$ and $\Psi_i : \Omega \to \mathbb{R}$ satisfying

$$\Psi(\mathbf{x}) = \left[\sqrt{\lambda_i}\Psi_i(\mathbf{x})\right]_{i=0}^{\infty}, \tag{4.9}$$

with $\lambda_i \downarrow 0$ and $\int_\Omega \Psi_i \Psi_j \, d\mu = \delta_i^j$. The λ_i's and Ψ_i's are respectively the *eigenvalues* and the *eigenfunctions* of K. In this case, the embedding Ψ therefore transforms elements of Ω into square-summable sequences with decreasing average energy. This is particularly important in order to approximate the kernel values, as we will review when describing our contribution.

Quick overview of the contribution. In this chapter, we propose two methods for NN search when the considered distance is derived from a Mercer kernel K. The described methods rely on *explicit embeddings*, that is approximating the unknown function Ψ by an explicit function $\tilde{\Psi}$ which maps Ω to the Euclidean space $\mathbb{R}^E$ ($E > 0$ is the *embedding dimension*), and satisfying

$$\langle \Psi(\mathbf{q}), \Psi(\mathbf{x}_i) \rangle_{\mathcal{H}} \approx \langle \tilde{\Psi}(\mathbf{q}), \tilde{\Psi}(\mathbf{x}_i) \rangle, \tag{4.10}$$

where $\langle \cdot, \cdot \rangle$ is the scalar product in $\mathbb{R}^E$. We will underline two benefits from exploiting explicit embeddings when designing ANN search methods with kernel distances:

- *Exact search scheme*: As shown in our first proposed method, one can derive error bounds from an explicit embedding such as KPCA, which can be used to derive an exact search scheme while performing the NN search in $\mathbb{R}^E$ instead of performing it in Ω. This can yield slightly lower search times while still be ensured to obtain the nearest neighbor of the query.

- *Approximate search scheme*: More importantly, our second proposed method aims at computing low-dimensional signatures for efficient large-scale ANN search in two steps: first the data $\mathcal{X}$ is mapped from Ω to $\mathbb{R}^E$ with an explicit embedding $\tilde{\Psi}$, then small-dimensional signatures are computed using methods adapted to an Euclidean space, and which are therefore not applicable directly in Ω. This two-step procedure is experimentally shown to perform better than previously proposed methods which aim at finding signatures directly in the implicit space, and are detailed in the following section.

4.1 State of the art: ANN search for kernels and explicit embeddings

In this section, we first describe two previously existing methods developed in order to compute signatures specifically aimed at solving problem (4.1) when d is a Mercer kernel distance. The two presented methods both produce binary signatures, which are then compared with respect to the Hamming distance in the signature space. Note that other methods aimed at producing small-size signatures for kernels exist, such as (He et al.,

2010; Gorisse et al., 2012), but were not considered in our work because the framework presented in these papers is different from ours and closer to a classification context.

Then is reviewed the main explicit embedding method: the Kernel Principal Component Analysis (KPCA), which will be used in our approach. We also briefly mention several other explicit embeddings procedures, which may also be used in the framework we consider, provided they are applied to the proper kernels.

4.1.1 Kernel ANN methods

Kernelized Locality Sensitive Hashing

In LSH, a typical choice for the hash functions for n-dimensional vectors are functions of the type $h : \mathbf{x} \mapsto \text{sign}(\langle \mathbf{x}, \mathbf{r} \rangle)$, where $\mathbf{r}$ is usually chosen as a random draw with respect to a normal distribution $\mathcal{N}(0, \mathbf{I})$. Kernelized Locality Sensitive Hashing (KLSH) (Kulis and Grauman, 2009) aims at imitating this Gaussian random choice in the implicit space $\mathcal{H}$.

To this end, the authors select M data points $\mathbf{y}_1, \ldots, \mathbf{y}_M$ among $\mathcal{X}$ and approximate the space $\mathcal{H}$ by the finite-dimensional space F spanned by $\Psi(\mathbf{y}_1), \ldots, \Psi(\mathbf{y}_M)$. The hash functions are determined by randomly choosing subsets of $Q < M$ points among the $\Psi(\mathbf{y}_i)$ and taking the empirical mean of these Q points. The image of a vector of Ω by such a hash function is computable thanks to the kernel trick. After a proper rescaling aimed at correcting variance deviations, this random choice yields a vector of F which is approximately drawn with respect to an isotropic Gaussian distribution $\mathcal{N}(0, \mathbf{I})$, according to the Central Limit Therorem.

Moreover, since the considered directions belong to the space F spanned by elements of $\mathcal{X}$, the image by the corresponding hash functions of any vector $\Psi(\mathbf{x})$ with $\mathbf{x} \in \Omega$ can explicitly be computable since the calculation only requires to compute kernel values between $\mathbf{x}$ and the $\mathbf{y}_i$.

Random Maximum Margin Hashing

Random Maximum Margin Hashing (RMMH) (Joly and Buisson, 2011) aims at finding hash functions of the form $h : \mathbf{x} \mapsto \text{sign}(\langle \Psi(\mathbf{x}), \mathbf{r} \rangle_{\mathcal{H}} + b)$, where $\mathbf{r}$ is once again a vector of $\mathcal{H}$ and b is a real number. The authors express the problem of finding such hash functions as an SVM problem.

Each hash function is chosen as follows: an even number M of vectors $\mathbf{y}_1, \ldots, \mathbf{y}_M$ are randomly drawn from $\mathcal{X}$ and half of these vectors are labeled with label $+1$, the other half with label -1. The corresponding hash function is given by

$$h(\mathbf{x}) = \text{sign}\left(\sum_{i=1}^{M} \alpha_i \epsilon_i K(\mathbf{x}, \mathbf{y}_i) + b \right), \tag{4.11}$$

where ϵ_i is the label of $\mathbf{y}_i$ and $(\alpha_i)_{i=1}^M$ and b maximize the usal C-SVM cost:

$$\frac{1}{2} \sum_{i,j=1}^{M} \epsilon_i \epsilon_j \alpha_i \alpha_j K(\mathbf{y}_i, \mathbf{y}_j) + C \sum_{i=1}^{M} \xi_i, \tag{4.12}$$

subject to

$$\forall i,\ \xi_i \geq 0 \text{ and } 1 - \xi_i \leq \epsilon_i \left(\sum_{j=1}^{M} \epsilon_j \alpha_j K(\mathbf{y}_i, \mathbf{y}_j) + b \right). \tag{4.13}$$

Intuitively, this formulation finds hash functions which prevent close neighbors from having a different image by the hash function, since the hyperplane is chosen by margin maximization. This method has been presented as yielding better precision than KLSH for approximate search.

4.1.2 Explicit embeddings

In this section, we provide a quick review of the standard KPCA method, which can be considered as the standard explicit embedding procedure. We also give a brief list of other embedding methods which have been proposed.

Kernel Principal Component Analysis

KPCA was introduced in (Schölkopf et al., 1998) as a way to perform explicitly PCA in the implicit space $\mathcal{H}$, thus being a generic way to compute a finite-dimensional embedding of the data satisfying the condition (4.10).

Performing PCA on the set of vectors $\Psi(\mathcal{X}) = (\Psi(\mathbf{x}_i))_{i=1}^M$ in implicit space $\mathcal{H}$ consists in projecting them onto the E-dimensional subspace V that minimizes the following mean-squared error:

$$\sum_{i,j=1}^{M} \left(K(\mathbf{x}_i, \mathbf{x}_j) - \langle P_W \Psi(\mathbf{x}_i), P_W \Psi(\mathbf{x}_j)\rangle_{\mathcal{H}}\right)^2 \tag{4.14}$$

over all E-dimensional subspaces W (P_W being the orthogonal projection onto W).

It is possible to compute explicitly this projection for any vector of the form $\Psi(\mathbf{x})$ by exploiting the Gram matrix $\mathbf{K}(\mathcal{X}) = (K(\mathbf{x}_i, \mathbf{x}_j))_{i,j=1,\dots M}$. Let us denote its E largest eigenvalues by $(\sigma_i^2)_{i=1}^E$ and by $(\mathbf{u}_i)_{i=1}^E$ the corresponding eigenvectors. Let $K(\mathbf{x}, :)$ be the vector $[K(\mathbf{x}, \mathbf{x}_1), \dots, K(\mathbf{x}, \mathbf{x}_M)]^T$. Then if $(\mathbf{v}_i)_{i=1}^E$ denote the E first eigenvectors of the covariance operator of $\Psi(\mathcal{X})$, one has (Schölkopf et al., 1998):

$$\varphi_i = \langle P_V(\Psi(\mathbf{x})), \mathbf{v}_i\rangle_{\mathcal{H}} = \frac{1}{\sigma_i}\langle K(\mathbf{x}, :), \mathbf{u}_i\rangle. \tag{4.15}$$

This result allows us to define by PCA approximation an explicit embedding $\tilde{\Psi} : \Omega \to \mathbb{R}^E$ such that $\tilde{\Psi}(\mathbf{x}) = [\varphi_i]_{i=1}^E$.

This result allows us to define by PCA approximation an explicit embedding $\tilde{\Psi}(\mathbf{x}) = P_V(\Psi(\mathbf{x}))$ and an approximate kernel $\tilde{K}(\mathbf{x}, \mathbf{y}) = \langle \tilde{\Psi}(\mathbf{x}), \tilde{\Psi}(\mathbf{y})\rangle$. We therefore have

$$\tilde{K}(\mathbf{x}, \mathbf{y}) = \langle P_V(\Psi(\mathbf{x})), P_V(\Psi(\mathbf{y}))\rangle_{\mathcal{H}}. \tag{4.16}$$

Note that KPCA is *adaptive*, that is the explicit embedding $\tilde{\Phi}$ it produces is learned on training data $\mathcal{X}$, and thus depend on it.

Other explicit embeddings

Other embedding methods have been proposed for specific kernels in a classification context. In (Maji and Berg, 2009), the authors describe an explicit embedding for the histogram intersection kernel, defined for $\mathbf{x}, \mathbf{y} \in \mathbb{R}^n$ by

$$K(\mathbf{x}, \mathbf{y}) = \sum_{i=1}^{n} \min(x_i, y_i). \tag{4.17}$$

Dataset	Number of vectors	Dim	Descriptor	URL
SIFT1M	1M	128	HesAff + SIFT	http://corpus-texmex.irisa.fr
BIGANN	1M–200M	128	DOG + SIFT	http://corpus-texmex.irisa.fr
Imagenet	1.261M	1,000	Bag-of-words	http://www.image-net.org

Table 4.1: Datasets used for the evaluation of nearest neighbor search

In (Vedaldi and Zisserman, 2010, 2012), the authors propose a nonadaptive (thus requiring no input data nor learning step) embedding algorithm for additive or multiplicative kernels, that is kernels defined on $\mathbb{R}^n$ as $K(\mathbf{x},\mathbf{y}) = \sum_{i=1}^{n} k(x_i, y_i)$ or $K(\mathbf{x},\mathbf{y}) = \prod_{i=1}^{n} k(x_i, y_i)$, where k is a kernel on $\mathbb{R}$.

In (Perronnin et al., 2010), the authors suggest applying KPCA independently in each dimension if the kernel K is additive. This yields a computational gain at the cost of a loss of precision in the embedding approximation.

As we will see, the methods we propose heavily relies on explicit embeddings. In all our experiments, we used KPCA as the embedding method, mainly because we are aiming at a generic kernel-independent method. In that sense, the KPCA has the benefit that it does not make any assumption on K other than the fact it is a Mercer kernel. Moreover, KPCA allows one to derive an embedding in $\mathbb{R}^E$ without constraints on E, whereas for instance, (Maji and Berg, 2009; Vedaldi and Zisserman, 2010; Perronnin et al., 2010) require the output dimensionality to be greater than or equal to that of the initial space. Finally, KPCA yields in practice good approximations on real datasets.

4.2 Layout of the chapter

Our contribution consists in two methods relying on explicit embedding for ANN search. Since the description of each method is directly followed by practical experiments, the datasets used in the experiments are first described in Section 4.3.

The first proposed method is an exact search method and is described in Section 4.4, along with an analysis of its benefits and an experimental illustration.

The second proposed method is an approximate scheme and is described in Section 4.5. It is experimentally compared to the previously mentioned KLSH and RMMH schemes in terms of precision.

4.3 Datasets and experimental protocol

For the sake of reproducibility, we have used publicly available datasets of vectors that are large enough to evaluate the interest of NN search techniques. Table 4.1 gives a brief description of these benchmarks, which all consist of image descriptors extracted from real images.

In our case, each of these datasets is separated into three disjoint sets: a query set, a training set and a validation set, this separation being random for SIFT1M and BIGANN which do not have such explicit prior separation of the full dataset.

- *SIFT1M*: This benchmark introduced in (Jégou et al., 2011) consists of one million local SIFT descriptors (Lowe, 2004) computed on patches extracted with the Hessian-Affine detector (Mikolajczyk et al., 2005).

- *BIGANN* (Jégou et al., 2011): This set provides a NN ground-truth for subsets of increasing size, from one million to one billion vectors. These descriptors were

extracted using the Difference of Gaussians (DoG) detector and are therefore not equivalent to those of the SIFT1M benchmark.

- *Imagenet* (Deng et al., 2009): This set contains the pre-computed 1,000-dimensional bag-of-words (BOW) vectors provided with the images of the 2010 Large-Scale Visual Recognition (ILSVRC'2010). For our search problem, the larger "train" set is used as the database to be searched, the "val" set to learn the parameters, and we randomly selected a subset of 1,000 queries from the "test" set to act as queries to be processed by our methods.

For these three datasets, we are looking for the most similar vectors with respect to the χ^2 kernel, as done in related works (Joly and Buisson, 2011; Vedaldi and Zisserman, 2012). The χ^2 kernel is defined on $\Omega = \mathbb{R}^n$ by

$$K(\mathbf{x}, \mathbf{y}) = K_{\chi^2}(\mathbf{x}, \mathbf{y}) = \sum_{i=1}^{n} \frac{x_i y_i}{x_i + y_i}. \tag{4.18}$$

In our experiments, all the data vectors were ℓ_1-normalized. This is the normalization for which $K_{\chi^2}(\mathbf{x}, \mathbf{x}) = 1$, and thus it is the required normalization to get the equivalence between nearest neighbor and highest kernel value, as discussed at the beginning of the chapter.

In the experiments we describe, the search performance is measured by the Recall@R, which measures the rate of queries for which the nearest neighbor is present among the R closest neighbors in the signature/embedded space.

4.4 Exact search method

The first method we propose relies on an explicit embedding in order to cast the search problem in a Euclidean space. The benefit is that in such a space, the scalar products are matrix multiplications and such computations can be performed by heavily optimized routines.

In order to derive an exact NN search scheme, one needs to be able to retain every potential nearest neighbor, that is upper bounding the precision of the explicit embedding. Since we are using KPCA, the next section is dedicated on the derivation of a standard error bound for KPCA.

4.4.1 Error bound for the KPCA embedding

Precision bounds on the KPCA provided in the literature are usually probabilistic bounds on the error between the empirical eigenvalues/eigenfunctions computed from the matrix $\mathbf{K}(\mathcal{X})$ (denoted σ_i^2 and $\mathbf{u}_i$ in Section 4.1.2) and their continuous counterparts, defined in the preliminary section of this chapter. Such bounds are difficult to obtain and they are not tight or require oracle knowledge about the implicit embedding (Shawe-Taylor et al., 2005; Braun, 2006).

However, it is possible to get tighter precision bounds for our problem by exploiting the data we consider. As shown in Section 4.1.2, KPCA is an orthogonal projection onto a subspace V in the implicit space $\mathcal{H}$. The approximation error $\epsilon(\mathbf{x}, \mathbf{y}) = K(\mathbf{x}, \mathbf{y}) - \tilde{K}(\mathbf{x}, \mathbf{y})$ can thus be written as a scalar product in the implicit space:

$$\begin{aligned}\epsilon(\mathbf{x},\mathbf{y}) &= K(\mathbf{x},\mathbf{y}) - \tilde{K}(\mathbf{x},\mathbf{y}) \\ &= \langle \Psi(\mathbf{x}), \Psi(\mathbf{y}) \rangle_{\mathcal{H}} - \langle P_V \Psi(\mathbf{x}), P_V \Psi(\mathbf{y}) \rangle_{\mathcal{H}} \\ &= \langle P_{V^\perp} \Psi(\mathbf{x}), P_{V^\perp} \Psi(\mathbf{y}) \rangle_{\mathcal{H}}, \end{aligned} \tag{4.19}$$

where $P_{V^\perp}$ is the orthogonal projection on the subspace orthogonal to V. Let us denote

$$R(\mathbf{x}) = ||P_{V^\perp} \Psi(\mathbf{x})||_{\mathcal{H}} = \left(K(\mathbf{x},\mathbf{x}) - ||\tilde{\Psi}(\mathbf{x})||^2 \right)^{\frac{1}{2}}. \tag{4.20}$$

This quantity can be easily computed in practice. The Cauchy-Schwarz inequality provides a bound on $\epsilon(\mathbf{x},\mathbf{y})$ in terms of $R(\mathbf{x})$ and $R(\mathbf{y})$:

$$|\epsilon(\mathbf{x},\mathbf{y})| \leq R(\mathbf{x}) R(\mathbf{y}). \tag{4.21}$$

This bound can be used for nearest neighbor search. Despite its simplicity, we believe it has never been used in such a context. Let us assume that we look for the nearest neighbor of a query $\mathbf{q}$ in the vector dataset $\mathcal{X}$ and let us define

$$\mathbf{x}_1 = \underset{\mathbf{x} \in \mathcal{X}}{\operatorname{argmax}}\, K(\mathbf{q},\mathbf{x}) \quad \text{and} \quad \tilde{\mathbf{x}}_1 = \underset{\mathbf{x} \in \mathcal{X}}{\operatorname{argmax}}\, \tilde{K}(\mathbf{q},\mathbf{x}), \tag{4.22}$$

that is respectively the nearest neighbor and approximate nearest neighbor of the query $\mathbf{q}$. We can get a lower bound for $K(\mathbf{q},\mathbf{x}_1)$ by noticing:

$$K(\mathbf{q},\mathbf{x}_1) \geq K(\mathbf{q},\tilde{\mathbf{x}}_1) \geq \tilde{K}(\mathbf{q},\tilde{\mathbf{x}}_1) - R(\mathbf{q}) R(\tilde{\mathbf{x}}_1). \tag{4.23}$$

For any vector $\mathbf{x} \in \mathcal{X}$, we can get an upper bound on $K(\mathbf{q},\mathbf{x})$:

$$K(\mathbf{q},\mathbf{x}) \leq \tilde{K}(\mathbf{q},\mathbf{x}) + R(\mathbf{q}) R(\mathbf{x}). \tag{4.24}$$

Combining (4.23) and (4.24), we obtain that, if for some $\mathbf{x}$ we have:

$$\tilde{K}(\mathbf{q},\mathbf{x}) + R(\mathbf{q}) R(\mathbf{x}) < \tilde{K}(\mathbf{q},\tilde{\mathbf{x}}_1) - R(\mathbf{q}) R(\tilde{\mathbf{x}}_1), \tag{4.25}$$

then $K(\mathbf{q},\mathbf{x}) < K(\mathbf{q},\mathbf{x}_1)$ and $\mathbf{x}$ cannot be the nearest neighbor of $\mathbf{q}$. The following section describes a procedure that uses this result to avoid kernel computations while still being sure to find the nearest neighbor of a query.

4.4.2 Exact search procedure: description and illustration

Assuming that the quantities $R(\mathbf{x})$ are computed and stored while performing the embedding of all vectors in dataset $\mathcal{X}$, the nearest neighbor of the query $\mathbf{q}$ is retrieved exactly by using the following procedure, which principle is illustrated in Figure 4.1 (left):

1. Compute $\tilde{\Psi}(\mathbf{q})$ and $R(\mathbf{q})$ using Equation (4.15).
2. Compute $\tilde{K}(\mathbf{q},\mathbf{x}) = \langle \tilde{\Psi}(\mathbf{q}), \tilde{\Psi}(\mathbf{x}) \rangle$ for all $\mathbf{x} \in \mathcal{X}$.
3. Find $\tilde{\mathbf{x}}_1 = \underset{\mathbf{x} \in \mathcal{X}}{\operatorname{argmax}}\, \tilde{K}(\mathbf{q},\mathbf{x})$
4. Find all elements $\mathbf{y} \in \mathcal{X}$ such that:
   $\tilde{K}(\mathbf{q},\tilde{\mathbf{x}}_1) - R(\mathbf{q}) R(\tilde{\mathbf{x}}_1) \leq \tilde{K}(\mathbf{q},\mathbf{y}) + R(\mathbf{q}) R(\mathbf{y})$.
   Let us denote by $\mathcal{Y}$ this set of vectors.
5. The true nearest neighbor of the query is $\mathbf{x}_1 = \underset{\mathbf{y} \in \mathcal{Y}}{\operatorname{argmax}}\, K(\mathbf{q},\mathbf{y})$.

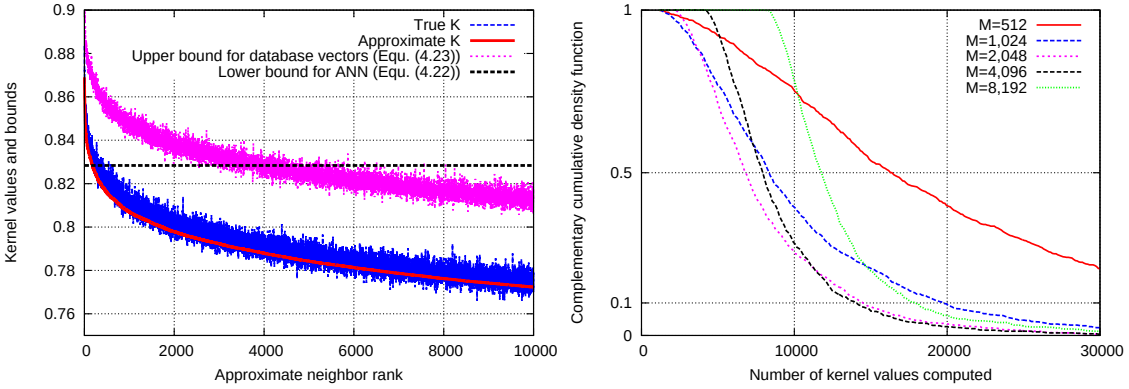

Figure 4.1: Exact nearest neighbor search in a database of 10M 128-dimensional SIFT vectors (extracted from the BIGANN database), embedded in dimension 128 with KPCA on χ^2 kernel. *Left:* Procedure for a given query. The reranked vectors are the neighbors which have an upper bound value higher than the lower bound of the approximate nearest neighbor. *Right:* one minus the cumulative density function of the number of kernel values computed to find the exact nearest neighbor of a query (for different values of M). The M kernel calculations required to compute the embedding of the query are taken into account.

4.4.3 Complexity Analysis and Experiments

Let L be the size of the base $\mathcal{X}$, E be the embedding dimension, M be the number of vectors used for KPCA, N be the cardinality of $\mathcal{Y}$, C_K be the cost of computing the kernel between two vectors in the initial space and $C_{\tilde{K}}$ be the cost of computing the approximate kernel between two embedded vectors (which is the cost of a scalar product in the embedded space). The cost of the exact search in the initial space has complexity $\mathcal{O}(C_K L)$.

The exact search in the embedded space as described above has complexity $\mathcal{O}(MC_K)$ for step 1 (scalar products are negligible if $E \ll M$), $\mathcal{O}(NC_{\tilde{K}})$ for steps 2 to 4 (steps 3 and 4 have constant complexity per data vector, which is negligible compared to $C_{\tilde{K}}$ for high-dimensional data), and $\mathcal{O}(NC_K)$ for step 5.

Hence this method is guaranteed to find the nearest neighbor of $\mathbf{q}$ with complexity

$$\mathcal{O}\left((M+N)C_K + LC_{\tilde{K}}\right). \tag{4.26}$$

If $C_{\tilde{K}} \ll C_K$ (which is likely to be the case, especially for complicated kernels), and if $M+N \ll L$, we can be sure to retrieve the nearest neighbor while performing much fewer calculations.

Figure 4.1 (right) illustrates this method on 10^6 SIFT descriptors taken from the dataset BIGANN and shows that it reduces the number of kernel values computed. Indeed, for $M = 2,048$, 90% of the tested queries require less than 15,000 kernel computations, which is a comparison with less than 0.15% of the base with respect to the distance we are interested in. These comparisons are of negligible complexity, the exact search process in the embedded space thus roughly requires 10^6 scalar products in dimension 128 instead of 10^6 χ^2 kernel computations in dimension 128 for the exact search in the initial space, which leads to a speedup considering that scalar products have more optimized implementations than the χ^2 kernel.

A tradeoff has to be found on M: a high value will tighten the Cauchy-Schwarz bound but will result in more kernel computations to perform the embedding on the query. In

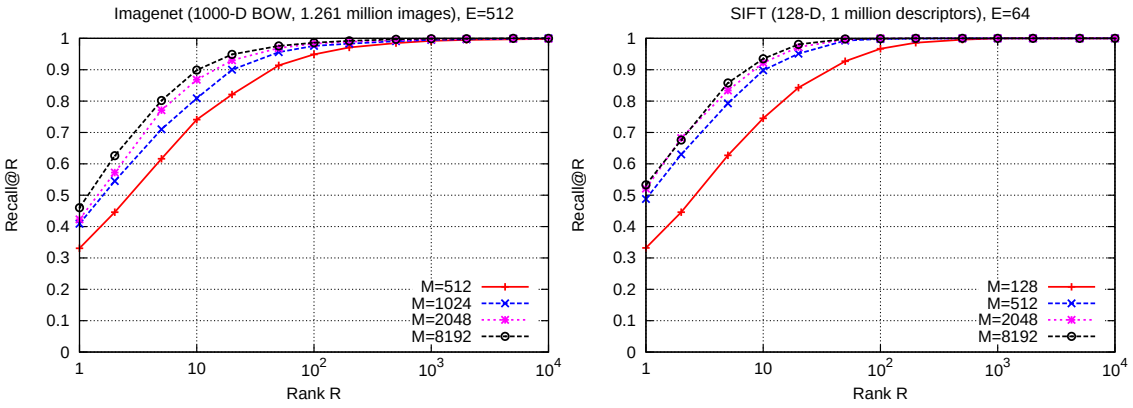

Figure 4.2: Impact of the size M of the learning set on the ranking performance for the exact search procedure, measured using recall@R curves. For the two datasets considered (Imagenet and SIFT1M), the vector output by the explicit embedding is about one half of the original descriptor dimensionality (E=512 and E=64, respectively).

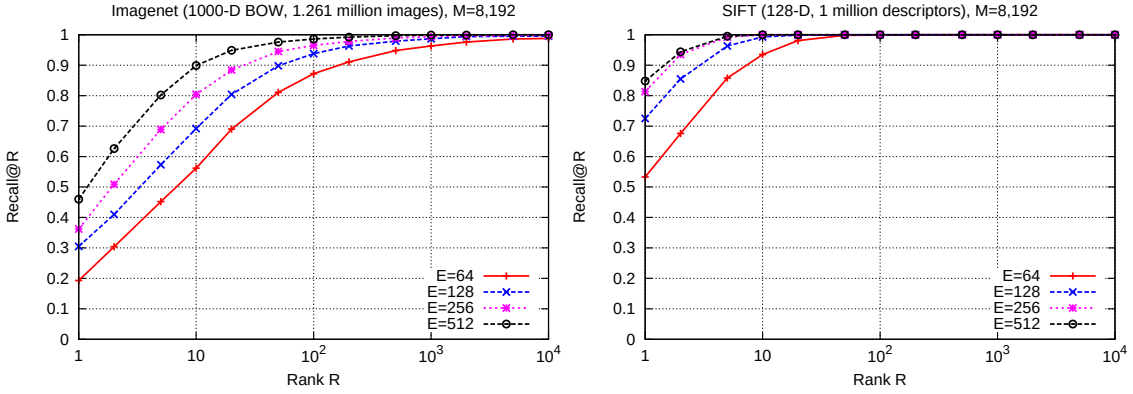

Figure 4.3: Ranking quality (Recall@R curves) as a function of the embedded vector size E for the exact search procedure. The number of learning vectors is here fixed to $M = 8,192$. As expected, the longer BOW vectors from Imagenet require a higher embedding dimensionality E.

this example, $M = 2,048$ performs on average better than $M = 8,192$ (8,000 average kernel computations vs. more than 11,000). See also Figures 4.2 and 4.3 for an analysis of the impact of parameters M and E on the search quality (independently of the number of kernel computations).

On the other hand, this approach does not perform well on the Imagenet database. For this dataset, the Cauchy-Schwarz inequality is too pessimistic to provide a good selection of the vectors to be reranked. Indeed, experiments showed that the number of reranked vectors reprensented 30% of the database. In this case, it is still possible to select fewer vectors by relaxing the Cauchy-Schwarz inequality and considering that

$$\alpha R(\mathbf{x})R(\mathbf{y}) \leq \epsilon(\mathbf{x},\mathbf{y}) \leq \beta R(\mathbf{x})R(\mathbf{y}), \tag{4.27}$$

where α and β are simply lower and upper bounds for $\cos(\mathbf{x},\mathbf{y})$. By empirically estimating values for α and β that work for many vectors, one can use concentration of measure inequalities to derive theoretical guarantees learned from data. However, one is not able then to precisely ensure the retrieval of the true nearest neighbor of a vector.

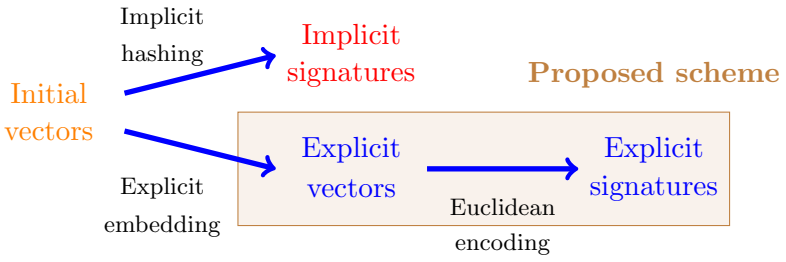

Figure 4.4: Proposed approximate search scheme. As opposed as the existing *implicit* hashing methods, we first compute explicit vectors belonging to an Euclidean space, then apply an encoding method designed for Euclidean distances.

4.5 Approximate search method

Even though the previous method allows one to perform an exact search at a slightly reduced cost, it may still be too costly to perform the approximate search in the embedded space. In this case, it is preferable to compute very small-dimensional signatures from the database elements, as described in the preliminary section of this chapter.

The binary schemes described in Section 4.1.1 are *implicit* schemes: they aim at finding the encoding function directly as a hashing function in the implicit space $\mathcal{H}$, expressing it as a function of the kernel K. Instead, the approximate scheme we propose relies on two separate steps:

1. The data is embedded in a Euclidean space $\mathbb{R}^E$ using a function $\tilde{\Psi}$ corresponding to KPCA[b].

2. The signatures are computed by applying an Euclidean signature method on the $\tilde{\Psi}(\mathbf{x}_i)$ in $\mathbb{R}^E$.

This scheme is illustrated in Figure 4.4.

Note that at step 2, if the data is normalized with respect to the kernel, any compression technique which aims at approximating the dot-product, the Euclidean distance or the cosine similarity could be employed. This includes among others LSH (Charikar, 2002b; Datar et al., 2004a) which approximates the cosine or Spectral Hashing (Weiss et al., 2008) and Locality Sensitive Binary Code (Raginsky and Lazebnik, 2010) which approximate the Euclidean distance. In this section, we particularly focus on a recently proposed technique known as Product Quantization (PQ) which has been shown to be state-of-the-art for the Euclidean distance (Jégou et al., 2011). It is discussed briefly in the next section.

4.5.1 Product quantization and variations

Most of the techniques based on short signatures for ANN search consider binary codes (Kulis and Grauman, 2009; Joly and Buisson, 2011; Gong and Lazebnik, 2011). In (Jégou et al., 2011), the authors proposed instead the PQ algorithm which produces non-binary codes allowing the estimation of Euclidean distances in the signature domain. For this purpose, a product quantizer is used to decompose the feature space into a Cartesian product of D subspaces. A given database vector $\mathbf{x}$ is decomposed into D subvectors of equal lengths, each of which is quantized separately using a k-means quantizer with a limited number of centroids. This number is typically set to 256 such that each subvector is quantized

[b] Any other explicit embedding method can be used if applicable, but we used KPCA in our experiments.

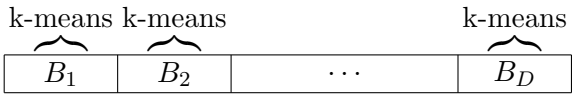

Figure 4.5: Illustration of the PQ procedure on one vector. The vector is considered as a product of D subvectors of equal length, and each subvector is quantized separately using a k-means algorithm on the database $\mathcal{X}$.

as a 1-byte code (8 bits) and therefore the vector is represented by a D-bytes code. This quantization is illustrated in Figure 4.5. The estimation of $d(\mathbf{q}, \mathbf{x})$ is done in an asymmetrical manner, *i.e.*, the query is not quantized to prevent it from being approximated. Computing a distance estimate requires D table look-ups and additions, which remains competitive compared with operations on binary vectors at a given signature size.

4.5.2 Experiments on approximate search

In this section, we provide and discuss experimental results of our approximate search scheme compared to the other previously described binary methods. Let's first discuss a particular pre-processing which has been chosen for the PQ technique in our experiments.

Pre-processing for energy balance.

Since we are applying PQ after a KPCA, the energy is not equally distributed in the embedded vectors $\tilde{\Psi}(\mathbf{x}_i)$: the first components have high energy while the last components have smaller values. This has a negative impact on PQ since the last subvectors will have less variance than the first while the k-means clustering still produces the same number of centroids for each subvector. Intuitively, one should find a way to balance energy between the subvectors so that all the quantizers have a subvector of comparable energy to encode.

In (Jégou et al., 2010), the authors faced the same phenomenon after a regular PCA. In this context, they argue that the PQ technique is more effective if a rotation matrix is applied on the vectors produced by the PCA and prior to PQ encoding. The drawback is that the individual product quantizers partially encode the same information, since the decorrelation performed by PCA is lost.

In our contribution, we consider the use of a random permutation of the components of the vectors $\tilde{\Psi}(\mathbf{x}_i)$ as an alternative to applying a random rotation. The random permutation is cheaper to compute and it ensures that the components remain uncorrelated.

In order to choose the appropriate pre-processing for PQ, let's study the impact of three pre-processing techniques:

(i) No pre-processing is done at the output of the KPCA.

(ii) A random rotation is applied to the $\tilde{\Psi}(\mathbf{x}_i)$.

(iii) The components of the $\tilde{\Psi}(\mathbf{x}_i)$ are randomly permuted.

Table 4.2 shows that the choice of the pre-processing stage has an impact on the search quality in terms of Recall@R. On SIFT1M, the random permutation is clearly the best choice. On Imagenet, the difference is less clear but the random permutation still yields better results. We will therefore adopt this choice in all subsequent experiments.

Dataset	SIFT1M			Imagenet		
Pre-processing	None	RR	RP	None	RR	RP
Recall@1	0.17	0.15	**0.19**	0.04	0.04	**0.05**
Recall@10	0.42	0.39	**0.51**	0.12	0.12	**0.14**
Recall@100	0.75	0.74	**0.85**	0.30	0.33	**0.37**
Recall@1000	0.96	0.97	**0.99**	0.60	0.67	**0.69**

Table 4.2: Choice of the pre-processing: performance of KPCA+PQ (Recall@R) with no pre-processing (Jégou et al., 2011), Random rotation (RR) (Jégou et al., 2010) and Random permutation (RP). Parameters: $M = 1,024$, $E = 64$ and $D = 8$ for the two datasets considered. Averages on 10 experiments.

Comparison with the state of the art.

We now compare the proposed coding scheme based on KPCA+PQ with the state-of-the-art techniques described earlier: KLSH (Kulis and Grauman, 2009) and RMMH (Joly and Buisson, 2011). We note that the proposed approach differs in two significant ways from KLSH and RMMH:

(i) As previously mentioned, KLSH and RMMH rely on an implicit embedding while we rely on an explicit KPCA embedding.

(ii) The signatures produced in KLSH and RMMH are binary while PQ encodes an embedded vector $\tilde{\Psi}(\mathbf{x}_i)$ as a succession of bytes.

To better understand the respective impact of these two differences, we also experimented with a coding scheme based on KPCA+LSH which performs an explicit embedding of the data (as in the first step of the proposed scheme) but performs a standard binary coding (as in the second step). Comparing this scheme with KLSH and RMMH allows one to measure exactly the gain of adding the explicit embedding step since the produced binary codes are very similar to those produce by KLSH and RMMH.

We performed experiments for different signature sizes (noted B in bits). The number of learning vectors for the KPCA was chosen as $M = 1,024$, so that the number of kernel computations when processing a query is limited. A similar number of learning vectors was chosen as an input to KLSH and RMMH algorithms. For the embedding dimension E, we set it differently depending on the scheme. For KPCA+PQ, we set $E = 128$. For KPCA+LSH, we set $E = B$ which led to optimal or near optimal results in preliminary experiments.

We report results for SIFT1M on Figure 4.6 and for Imagenet on Figure 4.7. We repeated all experiments 10 times, with different random draws for any step that relies on randomness. We show the average and standard deviation.

We can first observe that, as reported in (Joly and Buisson, 2011), RMMH outperforms KLSH on the BOW Imagenet features. However, for SIFT features, KLSH performs significantly better than RMMH. For instance, for $B = 128$ bits, the improvement in recall@100 is on the order of 10% absolute. Second, we can see that the simple KPCA+LSH scheme outperforms KLSH and RMMH at all ranks and code sizes. This difference can be significant (on the order of 10% absolute on SIFT1M for 128 bits). This shows the practical superiority of explicit embedding over implicit embedding for approximate search on these datasets. Finally, we can observe that KPCA+PQ performs significantly better than KPCA+LSH, which confirms the superiority of a non-binary encoding scheme over a binary coding scheme (Jégou et al., 2011).

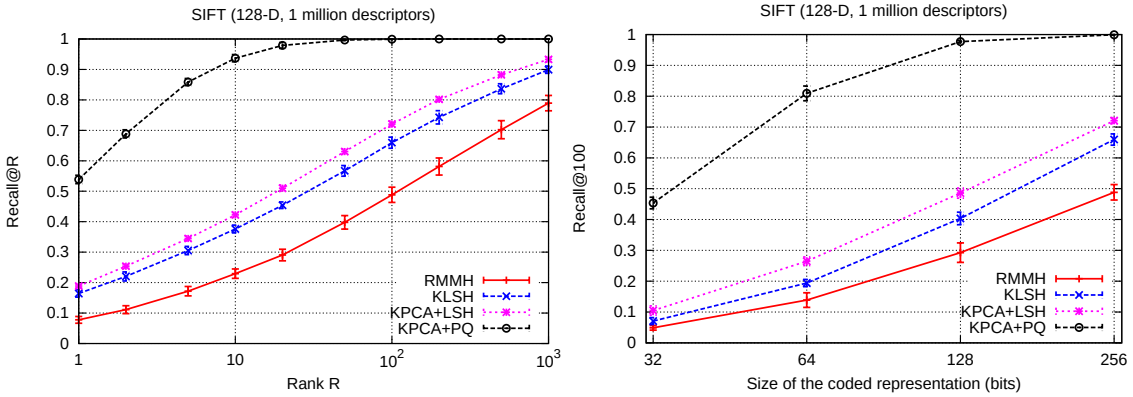

Figure 4.6: Comparison with the state-of-the-art: search quality for the χ^2 kernel with coded representations for 1M SIFT. *Left*: recall@R curves for a fixed size of B=256 bits. *Right:* recall@100 as a function of the number of bits (32–256). Parameters: KPCA+LSH $\rightarrow E = B$ and M=1,024; KPCA+PQ $\rightarrow E$=128 and M=1,024.

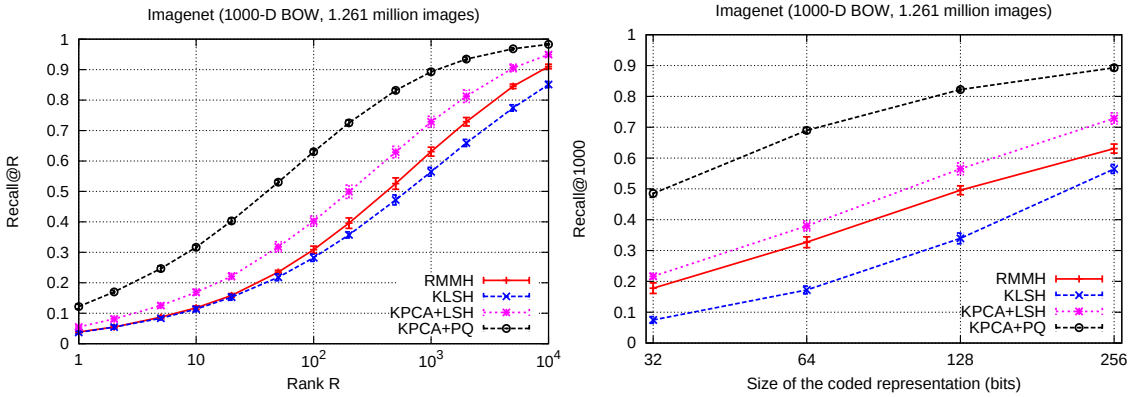

Figure 4.7: Comparison with the state-of-the-art: search quality for the χ^2 kernel with coded representations for 1.2M BOW. *Left*: recall@R curves for a fixed size of B=256 bits. *Right:* recall@1000 as a function of the number of bits (32–256). Parameters: KPCA+LSH $\rightarrow E = B$ and $M = 1,024$; KPCA+PQ $\rightarrow E = 128$ and $M = 1,024$.

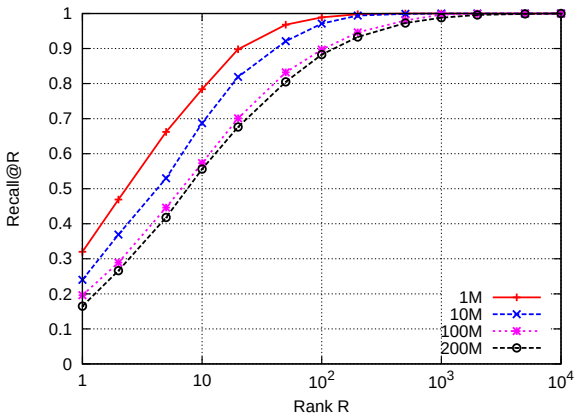

Figure 4.8: Performance as a function of the database size (1M to 200M vectors). Parameters for KPCA+PQ: $D = 16$, $E = 64$ and $M = 1,024$.

Overall, the improvement of KPCA+PQ with respect to KLSH and RMMH can be considerable, especially for small codes. For instance, for SIFT1M and for $B = 64$, the recall@100 is about 80% for KPCA+PQ while it is 20% for KLSH. For Imagenet and for the same number B of bits, the recall@1000 of RMMH is somewhat above 30% while KPCA+PQ achieves on the order of 70%.

Large scale experiments.

Figure 4.8 reports the Recall@R measured on subsets of 1M–200M SIFT descriptors taken from the BIGANN dataset. These results show that the proposed KPCA+PQ scheme can scale to very large datasets without a drastic loss in performance.

4.6 Conclusion

This study shows that when dealing with a distance derived from a Mercer kernel, the use of explicit embeddings can benefit nearest-neighbor search both for exact and more generally for approximate search.

The proposed exact search algorithm based on the Cauchy-Schwarz inequality works very well on the SIFT descriptors and allows to reduce the cost of the exact search roughly to the cost of computing the scalar products in the embedded space. However, it is not applicable to the Imagenet high-dimensional descriptors due to the rough precision of the Cauchy-Schwarz inequality in a high-dimensional space. Nonetheless, it may be possible in this context to sacrifice the exact search for a quasi-exact search with probability bounds by learning the discrepancy between Cauchy-Schwarz inequality and the actual value of the approximation error for a range of training vectors.

The proposed approximate scheme, despite its simplicity, performs better than other encoding methods in this framework. Performing KPCA followed by a standard binary LSH encoding yields better precision than standard methods which directly mimic binary LSH in the implicit space, thus giving a first incentive to use explicit embeddings. Furthermore, mapping the initial elements into an Euclidean space allows one to use a wide range of schemes designed at computing signatures, and especially Product Quantization, which performs way better than binary schemes at a given signature size. Therefore, using an explicit embedding prior to an efficient Euclidean coding scheme should be considered as

a baseline when measuring the performance of ANN search methods with small signatures in a kernelized framework.

Chapter 5

Conclusion

In this short chapter, we first propose to wrap up the principal contributions presented in this manuscript and to draw additional links between them, deriving some lessons from these different works. We then suggest some global research directions which seem interesting to pursue in order to benefit from the most conceptual part of this thesis work.

5.1 Summary of the contributions

The proof of concept of Chapter 2 focuses on estimating the parameters of a mixture of densities, instantiating it on isotropic Gaussian mixture models. Even though these are a simple problem and model, the estimation of such Gaussian mixture parameters acts as a step in many practical learning or data processing mechanisms. Therefore, this proof of concept is meaningful and acts as an incentive to come up with extensions of the method to more general Gaussian mixture models.

Aside from the experimental results, the originality of the approach itself motivates the search for alternate algorithms designed to perform learning with more data and less work (Shalev-Shwartz et al., 2012). The standard algorithmic procedure, updating in an iterative way some parameters using all available data, is replaced in our approach by a two-step procedure in which the data is compressed to a sketch of fixed-size representation prior to the estimation step. The way this sketching step is built may heavily depend on the task one wishes to perform afterwards: one could either consider building a sketch particularly fitted to a specific parameter estimation or more generally building a (probably higher-dimensional) sketch aimed at different estimation or processing tasks. In all cases, this sketching procedure can be viewed as a "storage step", where the only amount of information that is kept on the data is the amount one needs to be able to exploit it.

Since the estimation procedure we proposed can be interpreted as a generalized compressed sensing problem, it seems natural to be inspired by the rich signal processing literature on the subject. In particular, we showed in Chapter 3 that standard theoretical results can be generalized to a broad class of models. The assumptions we made to derive our results are very mild, so that the studied guarantees are potentially applicable to the study of many linear compression schemes for learning. More particularly, the stated theorems give tools aimed at studying both the performance one will not be able to achieve and the performance one can achieve if a generalized RIP is satisfied. This gives an angle for tackling the theoretical study of linear sketching tools: one may aim at finding frameworks where these tools satisfy the RIP in order to get the developed instance optimality guarantees.

The work in Chapter 4 shows that a simple scheme for nearest neighbor search exploiting a well-used technique (namely, the KPCA) can still achieve better performance than more elaborated techniques. One may see there an incentive to make the most of existing techniques.

5.2 Perspectives

Some short-term perspectives have already been presented at the end of Chapters 2 and 3. In this section, we will elaborate on some of them and evoke more general lines of research (possibly very conceptual in their current form) inspired by the work presented in this manuscript. These perspectives mainly focus on compressed density estimation (and more generally compressed learning), linear inverse problems for general models and algorithmic methods to solve such problems.

5.2.1 Short-term perspectives

Robustness analysis of compressed Gaussian mixture estimation. In Chapter 2, we proved a theorem stating the injectivity of the sketching operator on the set of sparse mixtures of isotropic Gaussians, provided the frequencies were (deterministically) well-chosen. However, the operators we used in our practical experiments were randomly drawn, and the method was experimentally shown to work with far fewer measurements than stated in the theorem. In order to better estimate the number of required measurements needed to reconstruct accurately the mixture, we certainly need, as in compressed sensing theory, to consider a probabilistic framework where $\mathbf{M}$ is randomly drawn. This should allow us to find a number of measurements so that with high probability on the draw of $\mathbf{M}$, a robustness result is satisfies (such as a RIP, so that the results of Chapter 3 are satisfied). In particular, it seems interesting to consider a method analog to (Baraniuk et al., 2007) by studying the average conservation of the ℓ_2 norm between f and $\mathbf{M}f$ for $f \in \Sigma_{k,n}^+$, then using concentration of measure and union bound arguments to obtain a RIP-like result.

Reduction of the complexity of the compressed estimation algorithm. The compressed approach we developed in Chapter 2 allows memory savings in the case of numerous data, but the complexity of the algorithm is still high due to the numerous optimization steps required by the estimation. It is therefore important to overcome these complexity limitations in the case of numerous parameters to estimate. The bottleneck of the algorithm is the last step of the iteration, which performs a gradient descent on the objective function taken as a function of the parameters. To reduce the computational cost of this step, one may consider trying not to perform it at each iteration, or update in this fashion only a fraction of the parameters. The complexity will also be reduced if the algorithm is modified so that it converges faster: in that sense, it may be interesting to consider a hierarchical procedure where only low-frequency components of the sketch are used at the beginning of the optimization, leading to a rough estimation of the parameters, then high-frequency components are considered to refine the estimation.

Practical extension of the compressed framework to a broader class of Gaussians. As has been mentioned in the summary of the contributions, Gaussian models are widely used in learning frameworks because of their simplicity. It is therefore an interesting outlook to extend the proof of concept of Chapter 2 to richer families of Gaussians than

isotropic Gaussians. In particular, Gaussians with diagonal covariance matrices are often used since they allow more variability than isotropic Gaussians without changing the order of magnitude of the number of parameters needed to describe them, which is $\mathcal{O}(kn)$ for k Gaussians in dimension n. An experimental study of the compressed framework instantiated on this enriched class of Gaussians is therefore motivated by potentially important applications. Moreover, the practical challenges can help finding a better way to build sketching operators and design algorithms to solve the compressed estimation problem.

Theoretical analysis of the difference between explicit and implicit hashing. We have experimentally shown in Chapter 4 that performing an explicit embedding then applying a Euclidean hashing scheme yielded better precision than using an implicit embedding which directly tries to mimic a hashing technique in the implicit space and use the kernel trick to define the hashing functions. It may be insightful to perform a theoretical analysis of this behavior. For instance, under assumptions on the probability distribution of the data and the decrease of the eigenvalues of K, can theoretical results be exploited to compare the expected precision of the different methods?

5.2.2 Mid-term perspectives

Generalized theoretical recovery results for compressive density mixture estimation. The compressive framework of Chapter 2 could be refined in the case where the mixture model is generic. In particular, a crucial question is the existence of key properties on the probability family $\mathcal{P}$ and the linear sketching operator $\mathbf{M}$ such that there exist theoretical recovery guarantees of a mixture of $\Sigma_k^+(\mathcal{P})$ given its image by $\mathbf{M}$. More precisely, finding conditions ensuring the generalized RIP for $\mathbf{M}$ stated in Chapter 3 would yield an IOP for $\mathbf{M}$. If a probabilistic analysis of the robustness of the sketching operator has already been performed in the case of isotropic Gaussians, it may serve as a starting point to identify the prerequisites on the density families allowing to generalize the robustness results.

Characterization of the existence of "well-behaved" decoders. In Chapter 3, we studied the existence of instance optimal decoders without constraints on these decoders. The theorems in Section 3.4 provide decoders which are minimizers of possibly nonconvex and irregular functions. In order to link the theory to practical application, we need to identify the cases when it is possible to regularize these decoders (in particular, make them minimizers of convex objectives) and still getting instance optimality. A starting point can be the study of the difference between the decoders mentioned in the proofs of Section 3.4 theorems and the ℓ_1 decoder used in classical compressed sensing. Can the method used to prove the $\ell_1 \Leftrightarrow \ell_0$ minimization equivalence for sparse vectors be adapted to more general norms and models?

5.2.3 Long-term perspectives

Randomized methods for non-convex optimization and sparse decomposition. In order to solve the non-convex problem (2.19), we designed an algorithm based on random initialization in the parameter space to search for good candidates to add to the current estimate. Such methods, aimed at reconstructing a sparse signal on a very coherent (and even continuous) dictionary, may be studied more thoroughly and improved. An important theoretical question is the existence of probabilistic recovery guarantees for such algorithms, which would certainly need to identify some particular required properties

on the random step for the algorithm to be robust. Given a dictionary $\mathcal{D}$, is there an "optimal" parametrization of $\mathcal{D}$ which essentially makes decomposition algorithms more robust?

The enriched tradeoffs of large-scale learning. In (Bottou and Bousquet, 2008), the authors studied the impact of modifying the expression of learning objectives to sacrifice optimization precision for computational savings. In particular, they proved that at a given target training time, the required time to reach an asymptotic generalization error can be reduced by sacrificing the precision of the iteration steps for computational gains. However, they do not take account the memory used by the training algorithm. Moreover, they consider algorithms aimed at minimizing a single objective.

A deep question is therefore the tradeoff between generalization error, required memory and computational time in a learning procedure. It would be particularly appealing to come with the relationship between memory and computational time at a target precision for a class of algorithms which compute low-dimensional representations of training data as a step before the actual estimation of the training parameters. The consideration of computational tradeoffs for a class of algorithms aimed at performing a certain task is linked to (Shalev-Shwartz et al., 2012) and is a thrilling problematic. It has the potential to answer some profound questions about the optimal way to perform learning depending on the limitations of a particular practical framework in terms of available memory, computational time, and number of available training samples.

Bibliography

Achlioptas, D. (2001). Database-friendly random projections. In *Proceedings of the twentieth ACM SIGMOD-SIGACT-SIGART symposium on Principles of database systems*, pages 274–281.

Adcock, B. and Hansen, A. C. (2011). Generalized sampling and infinite-dimensional compressed sensing. *DAMTP Tech. Rep.*

Adcock, B. and Hansen, A. C. (2012). A generalized sampling theorem for stable reconstructions in arbitrary bases. *J. Fourier Anal. Appl.*, 18(4):685–716.

Adcock, B., Hansen, A. C., Poon, C., and Roman, B. (2013). Breaking the coherence barrier: asymptotic incoherence and asymptotic sparsity in compressed sensing. *arXiv*, pages 1–44.

Andoni, A. and Indyk, P. (2008). Near-optimal hashing algorithms for approximate nearest neighbor in high dimensions. *Commun. ACM*, 51(1):117–122.

Baraniuk, R., Davenport, M., Devore, R., and Wakin, M. (2007). A simple proof of the restricted isometry property for random matrices. *Constr. Approx*, 2008.

Baraniuk, R. G., Cevher, V., and Wakin, M. B. (2010). Low-dimensional models for dimensionality reduction and signal recovery: A geometric perspective. *Proceedings of the IEEE*, 98(6):959–971.

Baraniuk, R. G. and Wakin, M. B. (2006). Random projections of smooth manifolds. In *Foundations of Computational Mathematics*, pages 941–944.

Bertin, K., Pennec, E. L., and Rivoirard, V. (2011). Adaptive Dantzig density estimation. *Annales de l'Institut Henri Poincaré (B) Probabilités et Statistiques*, 47(1):43–74.

Blumensath, T. (2011). Sampling and reconstructing signals from a union of linear subspaces. *IEEE Transactions on Information Theory*, 57(7):4660–4671.

Blumensath, T. and Davies, M. E. (2009a). Iterative hard thresholding for compressed sensing. *Applied and Computational Harmonic Analysis*, 27(3):265–274.

Blumensath, T. and Davies, M. E. (2009b). Sampling theorems for signals from the union of finite-dimensional linear subspaces. *IEEE Transactions on Information Theory*, 55(4):1872–1882.

Boiman, O., Shechman, E., and Irani, M. (2008). In defense of nearest neighbor based image classification. In *CVPR*.

Bottou, L. (1998). Online algorithms and stochastic approximations. In Saad, D., editor, *Online Learning and Neural Networks*. Cambridge University Press, Cambridge, UK. revised, oct 2012.

Bottou, L. and Bousquet, O. (2007). The tradeoffs of large scale learning. In *NIPS*.

Bottou, L. and Bousquet, O. (2008). The tradeoffs of large scale learning. In *Neural Information Processing Systems Conference*.

Bourrier, A., Davies, M. E., Peleg, T., Pérez, P., and Gribonval, R. (2013a). Fundamental performance limits for ideal decoders in high-dimensional linear inverse problems. Submitted to IEEE Trans. Inf. Theory.

Bourrier, A., Gribonval, R., and Pérez, P. (2013b). Compressive Gaussian Mixture Estimation. In *2013 IEEE International Conference on Acoustics, Speech and Signal Processing*.

Bourrier, A., Gribonval, R., and Pérez, P. (2013c). Estimation de mélange de Gaussiennes sur données compressées. In *XXIVème Colloque Gretsi*.

Bourrier, A., Perronnin, F., Gribonval, R., Pérez, P., and Jégou, H. (2012). Nearest neighbor search for arbitrary kernels with explicit embeddings. Research Report RR-8040, INRIA.

Boyd, S. and Vandenberghe, L. (2004). *Convex Optimization*. Cambridge University Press, New York, NY, USA.

Braun, M. (2006). Accurate error bounds for the eigenvalues of the kernel matrix. *Journal of Machine Learning Research*, 7:2303–2328.

Bunea, F., Tsybakov, A. B., Wegkamp, M., and Barbu, A. (2010). Spades and mixture models. *Annals of Statistics*, 38(4):2525–2558.

Calderbank, R., Schapire, R., and Jafarpour, S. (2009). Compressed learning : Universal sparse dimensionality reduction and learning in the measurement domain. *Preprint*.

Candès, E. J. (2008). The restricted isometry property and its implications for compressed sensing. *C. R. Acad. Sci. Paris S'er. I Math.*, 346:589–592.

Candès, E. J., Eldar, Y. C., Needell, D., and Randall, P. (2011). Compressed sensing with coherent and redundant dictionaries. *Applied and Computational Harmonic Analysis*, 31(1):59–73.

Candès, E. J., Li, X., Ma, Y., and Wright, J. (2011a). Robust principal component analysis? *J. ACM*, 58(3):11.

Candès, E. J. and Plan, Y. (2011). A probabilistic and ripless theory of compressed sensing. *IEEE Transactions on Information Theory*, 57(11):7235–7254.

Candès, E. J. and Plan, Y. (2011). Tight oracle inequalities for low-rank matrix recovery from a minimal number of noisy random measurements. *IEEE Trans. Inf. Theor.*, 57(4):2342–2359.

Candès, E. J., Romberg, J. K., and Tao, T. (2006). Robust uncertainty principles: exact signal reconstruction from highly incomplete frequency information. *IEEE Transactions on Information Theory*, 52(2):489–509.

Candès, E. J., Romberg, J. K., and Tao, T. (2006). Stable signal recovery from incomplete and inaccurate measurements. *Communications on Pure and Applied Mathematics*, 59(8):1207–1223.

Candès, E. J., Strohmer, T., and Voroninski, V. (2011b). Phaselift: Exact and stable signal recovery from magnitude measurements via convex programming. *CoRR*, abs/1109.4499.

Candès, E. J. and Tao, T. (2005). Decoding by linear programming. *IEEE Transactions on Information Theory*, 51(12):4203–4215.

Candès, E. J. and Tao, T. (2006). Near-optimal signal recovery from random projections: Universal encoding strategies? *IEEE Transactions on Information Theory*, 52(12):5406–5425.

Candès, E. J. and Tao, T. (2007). The Dantzig selector: Statistical estimation when p is much larger than n. *The Annals of Statistics*, 35(6):2313–2351.

Chandrasekaran, V., Recht, B., Parrilo, P. A., and Willsky, A. S. (2012). The convex geometry of linear inverse problems. *Foundations of Computational Mathematics*, 12(6):805–849.

Charikar, M. (2002a). Similarity estimation techniques from rounding algorithms. In *STOC*, pages 380–388.

Charikar, M. (2002b). Similarity estimation techniques from rounding algorithms. In *ACM STOC*.

Charikar, M., Chen, K., and Farach-Colton, M. (2002). Finding frequent items in data streams. In *ICALP*, pages 693–703.

Chen, S. S., Donoho, D. L., and Saunders, M. A. (1998). Atomic decomposition by basis pursuit. *SIAM Journal on Scientific Computing*, 20:33–61.

Cohen, A., Dahmen, W., and Devore, R. (2009). Compressed sensing and best k-term approximation. *J. Amer. Math. Soc*, pages 211–231.

Cormode, G. and Hadjieleftheriou, M. (2010). Methods for finding frequent items in data streams. *VLDB Journal*, 19(1):3–20.

Cormode, G. and Muthukrishnan, S. (2004). An improved data stream summary: The count-min sketch and its applications. In *LATIN*, pages 29–38.

Datar, M., Immorlica, N., Indyk, P., and Mirrokni, V. (2004a). Locality-sensitive hashing scheme based on p-stable distributions. In *Proceedings of the Symposium on Computational Geometry*, pages 253–262.

Datar, M., Immorlica, N., Indyk, P., and Mirrokni, V. S. (2004b). Locality-sensitive hashing scheme based on p-stable distributions. In *Symposium on Computational Geometry*, pages 253–262.

Davenport, M. A., Needell, D., and Wakin, M. B. (2013). Signal space cosamp for sparse recovery with redundant dictionaries. *IEEE Transactions on Information Theory*, 59(10):6820–6829.

Davis, G., Mallat, S., and Avellaneda, M. (1997). Adaptive greedy approximations. *Constructive Approximation*, 13(1):57–98.

Dempster, A. P., Laird, N. M., and Rubin, D. B. (1977). Maximum likelihood from incomplete data via the em algorithm. *Journal of the Royal Statistical Society*, 39(1):1–38.

Deng, J., Dong, W., Socher, R., Li, L.-J., Li, K., and Fei-Fei, L. (2009). ImageNet: A Large-Scale Hierarchical Image Database. In *CVPR*.

DeVore, R. A. and Temlyakov, V. N. (1996). Some remarks on greedy algorithms. *Adv. Comp. Math.*, 5(2-3):173–187.

Donoho, D. L. (2004). For most large underdetermined systems of linear equations the minimal ℓ_1-norm solution is also the sparsest solution. *Comm. Pure Appl. Math*, 59:797–829.

Donoho, D. L. (2006). Compressed sensing. *IEEE Trans. Inform. Theory*, 52:1289–1306.

Donoho, D. L., Elad, M., and Temlyakov, V. N. (2006). Stable recovery of sparse overcomplete representations in the presence of noise. *IEEE TRANS. INFORM. THEORY*, 52(1):6–18.

Donoho, D. L. and Huo, X. (2001). Uncertainty principles and ideal atomic decomposition. *IEEE Transactions on Information Theory*, 47(7):2845–2862.

Eftekhari, A. and Wakin, M. B. (2013). New analysis of manifold embeddings and signal recovery from compressive measurements. *CoRR*, abs/1306.4748.

Elad, M. (2010). *Sparse and Redundant Representations - From Theory to Applications in Signal and Image Processing*. Springer.

Eldar, Y. C., Kuppinger, P., and Bölcskei, H. (2010). Block-sparse signals: uncertainty relations and efficient recovery. *IEEE Transactions on Signal Processing*, 58(6):3042–3054.

Engl, H. W., Hanke, M., and Neubbauer, A. (1996). *Regularization of Inverse Problems*. Springer.

Foucart, S. (2011). Hard thresholding pursuit: An algorithm for compressive sensing. *SIAM J. Numerical Analysis*, 49:2543–2563.

Foucart, S. and Rauhut, H. (2013). *A mathematical introduction to compressive sensing*. Applied and Numerical Harmonic Analysis. Springer.

Fourier, J. (1808). Mémoire sur la propagation de la chaleur dans les corps solides. *Nouveau Bulletin des sciences par la Société philomatique de Paris I*, 6.

Gilbert, A. C., Strauss, M. J., Tropp, J. A., and Vershynin, R. (2007). One sketch for all: fast algorithms for compressed sensing. In *STOC*, pages 237–246.

Gionis, A., Indyk, P., and Motwani, R. (1999). Similarity search in high dimensions via hashing. In *VLDB*, pages 518–529.

Giryes, R., Nam, S., Elad, M., Gribonval, R., and Davies, M. E. (2013). Greedy-Like Algorithms for the Cosparse Analysis Model. partially funded by the ERC, PLEASE project, ERC-2011-StG-277906.

Goemans, M. X. and Williamson, D. P. (1995). Improved approximation algorithms for maximum cut and satisfiability problems using semidefinite programming. *J. ACM*, 42(6):1115–1145.

Gong, Y. and Lazebnik, S. (2011). Iterative quantization: A procrustean approach to learning binary codes. In *CVPR*.

Gorisse, D., Cord, M., and Precioso, F. (2012). Locality-sensitive hashing for chi2 distance. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, 34(2):402–409.

Grauman, K. and Darrell, T. (2005). The pyramid match kernel: Discriminative classification with sets of image features. In *ICCV*.

Gribonval, R., Mailhé, B., Rauhut, H., Schnass, K., and Vandergheynst, P. (2007). Average Case Analysis of Multichannel Thresholding. In *Acoustics, Speech and Signal Processing, 2007. ICASSP 2007. IEEE International Conference on*, pages II–853 – II–856, Honolulu, Hawai, United States. IEEE.

Gribonval, R. and Nielsen, M. (2007). Highly sparse representations from dictionaries are unique and independent of the sparseness measure. *Appl. Comp. Harm. Anal.*, 22(3):335–355.

Halko, N., Martinsson, P.-G., and Tropp, J. A. (2011). Finding structure with randomness: Probabilistic algorithms for constructing approximate matrix decompositions. *SIAM Review*, 53(2):217–288.

Hansen, A. C. and Adcock, B. (2011). Generalized sampling and infinite dimensional compressed sensing. *Magnetic Resonance Imaging*.

He, J., Liu, W., and Chang, S.-F. (2010). Scalable similarity search with optimized kernel hashing. In *Proceedings of the 16th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, KDD '10, pages 1129–1138, New York, NY, USA. ACM.

Indyk, P. and Motwani, R. (1998). Approximate nearest neighbors: Towards removing the curse of dimensionality. In *STOC*, pages 604–613.

Jégou, H., Douze, M., and Schmid, C. (2011). Product quantization for nearest neighbor search. *Trans. PAMI*, 33(1):117–128.

Jégou, H., Douze, M., Schmid, C., and Pérez, P. (2010). Aggregating local descriptors into a compact image representation. In *CVPR*.

Jégou, H., Tavenard, R., Douze, M., and Amsaleg, L. (2011). Searching in one billion vectors: re-rank with source coding. In *ICASSP*, Prague Czech Republic.

Johnson, W. and Lindenstrauss, J. (1984). Extensions of Lipschitz mappings into a Hilbert space. In *Conference in modern analysis and probability (New Haven, Conn., 1982)*, volume 26 of *Contemporary Mathematics*, pages 189–206. American Mathematical Society.

Joly, A. and Buisson, O. (2011). Random maximum margin hashing. In *CVPR*.

König, H. (1986). Eigenvalues of compact operators with applications to integral operators. *Linear Algebra and its Applications*, 84(0):111 – 122.

Kulis, B. and Grauman, K. (2009). Kernelized locality-sensitive hashing for scalable image search. In *ICCV*.

Kutyniok, G. (2012). Compressed sensing: Theory and applications. *CoRR*, abs/1203.3815.

Lowe, D. (2004). Distinctive image features from scale-invariant keypoints. *IJCV*, 60(2):91–110.

Mahoney, M. W. (2011). Randomized algorithms for matrices and data. *Foundations and Trends in Machine Learning*, 3(2):123–224.

Maillard, O.-A. and Munos, R. (2009). Compressed least squares regression. In Bengio, Y., Schuurmans, D., Lafferty, J., Williams, C. K. I., and Culotta, A., editors, *Advances in Neural Information Processing Systems 22*, pages 1213–1221.

Maji, S. and Berg, A. (2009). Max-margin additive models for detection. In *ICCV*.

Mallat, S. (2008). *A Wavelet Tour of Signal Processing, Third Edition: The Sparse Way*. Academic Press, 3rd edition.

Mallat, S. and Zhang, Z. (1993). Matching pursuit with time-frequency dictionaries. *IEEE Transactions on Signal Processing*, 41:3397–3415.

Mikolajczyk, K., Tuytelaars, T., Schmid, C., Zisserman, A., Matas, J., Schaffalitzky, F., Kadir, T., and Gool, L. V. (2005). A comparison of affine region detectors. *IJCV*, 65(1/2):43–72.

Mishali, M. and Eldar, Y. C. (2009). Blind multiband signal reconstruction: Compressed sensing for analog signals. *IEEE Transactions on Signal Processing*, 57(3):993–1009.

Muja, M. and Lowe, D. G. (2009). Fast approximate nearest neighbors with automatic algorithm configuration. In *VISAPP*.

Murata, N. (1998). A statistical study on on-line learning. In *Online Learning and Neural Networks*. Cambridge University Press.

Nam, S., Davies, M. E., Elad, M., and Gribonval, R. (2013). The Cosparse Analysis Model and Algorithms. *Applied and Computational Harmonic Analysis*, 34(1):30–56.

Natarajan, B. K. (1995). Sparse approximate solutions to linear systems. *SIAM J. Comput.*, 24(2):227–234.

Needell, D. and Ward, R. (2013). Stable image reconstruction using total variation minimization. *SIAM J. Imaging Sciences*, 6(2):1035–1058.

Nocedal, J. and Wright, S. J. (2006). *Numerical Optimization*. Springer, New York, 2nd edition.

Ohlsson, H., Yang, A. Y., and Sastry, S. S. (2011). Compressive phase retrieval from squared output measurements via semidefinite programming. *CoRR*, abs/1111.6323.

Oymak, S., Jalali, A., Fazel, M., Eldar, Y. C., and Hassibi, B. (2012). Simultaneously structured models with application to sparse and low-rank matrices. *CoRR*, abs/1212.3753.

Parvaresh, F., Vikalo, H., Misra, S., and Hassibi, B. (2008). Recovering sparse signals using sparse measurement matrices in compressed dna microarrays. *J. Sel. Topics Signal Processing*, 2(3):275–285.

Peleg, T., Gribonval, R., and Davies, M. E. (2013). Compressed sensing and best approximation from union of subspaces: Beyond dictionaries. In *EUSIPCO*.

Perronnin, F., Sánchez, J., and Liu, Y. (2010). Large-scale image categorization with explicit data embedding. In *CVPR*.

Qaisar, S., Bilal, R. M., Iqbal, W., Naureen, M., and Lee, S. (2013). Compressive sensing: From theory to applications, a survey. *Journal of Communication and Networks*, 15(5):443–456.

Raginsky, M. and Lazebnik, S. (2010). Locality-sensitive binary codes from shift-invariant kernels. In *NIPS*.

Rahimi, A. and Recht, B. (2007). Random features for large-scale kernel machines. In *NIPS*.

Rahimi, A. and Recht, B. (2008). Weighted sums of random kitchen sinks: Replacing minimization with randomization in learning. In *NIPS*, pages 1313–1320.

Rauhut, H., Schnass, K., and Vandergheynst, P. (2008). Compressed sensing and redundant dictionaries. *IEEE Transactions on Information Theory*, 54(5):2210–2219.

Recht, B., Fazel, M., and Parrilo, P. A. (2010). Guaranteed minimum-rank solutions of linear matrix equations via nuclear norm minimization. *SIAM Review*, 52(3):471–501.

Robbins, H. and Monro, S. (1951). A stochastic approximation method. *Annals of Mathematical Statistics*, 22:400–407.

Sanchez, J., Perronnin, F., Mensink, T., and Verbeek, J. (2013). Image classification with the fisher vector: Theory and practice. *Int. J. Computer Vision*, 105(3):22–245.

Schölkopf, B. and Smola, A. (2002). *Learning with Kernels: Support Vector Machines, Regularization, Optimization and Beyond*. MIT Press, Cambridge, MA.

Schölkopf, B., Smola, A., and Müller, K.-R. (1998). Nonlinear component analysis as a kernel eigenvalue problem.

Scholkopf, B. and Smola, A. J. (2001). *Learning with Kernels: Support Vector Machines, Regularization, Optimization, and Beyond*. MIT Press, Cambridge, MA, USA.

Shalev-Shwartz, S., Shamir, O., and Tromer, E. (2012). Using more data to speed-up training time. In *AISTATS*, pages 1019–1027.

Shawe-Taylor, J., Williams, C., Cristianini, N., and Kandola, J. (2005). On the eigenspectrum of the gram matrix and the generalization error of kernel pca. IEEE *Transactions on Information Theory*, 51(7):2510–2522.

Sivic, J. and Zisserman, A. (2003). Video Google: A text retrieval approach to object matching in videos. In *ICCV*.

Thaper, N., Guha, S., Indyk, P., and Koudas, N. (2002). Dynamic multidimensional histograms. In *ACM SIGMOD International conference on Management of data*.

Torralba, A., Fergus, R., and Weiss, Y. (2008). Small codes and large databases for recognition. In *CVPR*.

Tropp, J. A. (2004). Greed is good: Algorithmic results for sparse approximation. *IEEE Trans. Inform. Theory*, 50:2231–2242.

Vedaldi, A. and Zisserman, A. (2010). Efficient additive kernels via explicit feature maps. In *CVPR*.

Vedaldi, A. and Zisserman, A. (2012). Efficient additive kernels via explicit feature maps. *Trans. PAMI*, 34(3):480–492.

Wakin, M. B. (2009). A manifold lifting algorithm for multi view compressive imaging. In *in Proc. Picture Coding Symposium*.

Wakin, M. B., Donoho, D. L., Choi, H., and Baraniuk, R. G. (2005). The multiscale structure of non-differentiable image manifolds. In *in Proc. Wavelets XI at SPIE Optics and Photonics*.

Weiss, Y., Torralba, A., and Fergus, R. (2008). Spectral hashing. In *NIPS*.

Woolfe, F., Liberty, E., Rokhlin, V., and Tygert, M. (2008). A fast randomized algorithm for the approximation of matrices. *Applied and Computational Harmonic Analysis*, 25(3):335 – 366.

Yu, G. and Sapiro, G. (2011). Statistical compressed sensing of gaussian mixture models. *IEEE Transactions on Signal Processing*, 59(12):5842–5858.

Yuan, M. (2010). High dimensional inverse covariance matrix estimation via linear programming. *Journal of Machine Learning Research*, 11:2261–2286.

Yuan, M. and Lin, Y. (2007). Model selection and estimation in the Gaussian graphical model. *Biometrika*.

Zhou, Z., Li, X., Wright, J., Candès, E. J., and Ma, Y. (2010). Stable principal component pursuit. In *ISIT*, pages 1518–1522.

List of Figures

Appendix A

Proofs of Chapter 2 theorems

This appendix is aimed at proving Theorems 1 and 2. Let's recall that the standard deviation of the Gaussians, denoted σ, is a fixed positive real number in all the subsequent sections of this appendix. For $\boldsymbol{\omega} \in \mathbb{R}^n$, $\mathbf{M}_{\boldsymbol{\omega}}$ will denote the Fourier sampling operator on $L^1(\mathbb{R}^n)$ at frequency $\boldsymbol{\omega}$. Recall that $L^1(\mathbb{R}^n)$ denotes the complex vector space of Lebesgue-integrable functions mapping $\mathbb{R}^n$ into $\mathbb{C}$.

A.1 Preliminary lemmas

In this section, we consider n as any positive integer and prove two elementary lemmas which will be used in the proofs of Theorems 1 and 2.

The first lemma states a simple result:

Lemma 5. *For any integer $n > 0$, the family*

$$\mathcal{P}_n = \left\{ p_{\boldsymbol{\mu}} : \mathbf{x} \mapsto \frac{1}{(2\pi)^{\frac{n}{2}}\sigma^n} \exp\left(-\frac{\|\mathbf{x}-\boldsymbol{\mu}\|_2^2}{2\sigma^2}\right), \boldsymbol{\mu} \in \mathbb{R}^n \right\} \tag{A.1}$$

is linearly independent in $L^1(\mathbb{R}^n)$.

Proof. Let $N > 0$, $(\lambda_j)_{j=1}^N$ be complex numbers and $(\boldsymbol{\nu}_j)_{j=1}^N$ be real n-dimensional distinct vectors such that $f = \sum_{j=1}^N \lambda_j p_{\boldsymbol{\nu}_j} = 0$. Let's prove that we necessarily have $\lambda_j = 0$ for all j, which will prove the lemma.

Denoting by $\mathcal{F}$ the Fourier transform, we get that for any $\boldsymbol{\omega} \in \mathbb{R}^n$:

$$\mathcal{F}(f).\boldsymbol{\omega} = \sum_{j=1}^N \lambda_j \mathcal{F}(p_{\boldsymbol{\nu}_j}).\boldsymbol{\omega} = \sum_{j=1}^N \lambda_j e^{-i\langle \boldsymbol{\omega}, \boldsymbol{\nu}_j \rangle} \mathcal{F}(p_0).\boldsymbol{\omega} = 0. \tag{A.2}$$

Since $\mathcal{F}(p_0)$ takes no zero value, this is equivalent to

$$\sum_{j=1}^N \lambda_j e^{-i\langle \boldsymbol{\omega}, \boldsymbol{\nu}_j \rangle} = 0. \tag{A.3}$$

Denoting f_j the linear form $\boldsymbol{\omega} \mapsto \langle \boldsymbol{\omega}, \boldsymbol{\nu}_j \rangle$ on $\mathbb{R}^n$, there exists $\mathbf{u} \in \mathbb{R}^n$ such that $\mathbf{u} \notin \ker(f_r - f_s)$ for all distinct $r, s \in [\![1, N]\!]$. A straightforward argument to justify this claim is to invoke Baire's theorem, since $\mathbb{R}^n$ is complete and a hyperplane is a closed set of empty interior.

Denoting $c_j = \langle \mathbf{u}, \boldsymbol{\nu}_j \rangle$, the hypothesis on $\mathbf{u}$ implies that the c_j are distinct. We then have for any $x \in \mathbb{R}$, by replacing $\boldsymbol{\omega}$ by $x\mathbf{u}$ in (A.3),

$$\sum_{j=1}^{N} \lambda_j e^{-ic_j x} = 0. \tag{A.4}$$

The functions $x \mapsto e^{-ic_j x}$ defined on $\mathbb{R}$ are eigenfunctions of the linear derivation operator on $C^\infty(\mathbb{R})$ associated to distinct eigenvalues $-ic_1, \ldots, -ic_N$. Therefore, they form a linearly independent family and the λ_j are all zero. Finally, the family $\mathcal{P}_n$ is linearly independent.

□

Lemma 5 will principally be used in the following way: if a linear combination $f = \sum_{s=1}^{k} \lambda_s p_{\boldsymbol{\mu}_s}$ of distinct functions of $\mathcal{P}_n$ is the zero function, then all the λ_s are equal to 0.

The next lemma relates the injectivity of an operator $\mathbf{M}$ on $\Sigma_{k,n}$ with the preimage of $\{0\}$ by $\mathbf{M}$ in $\Sigma_{2k,n}$.

Lemma 6. *Let $n > 0$ and $\mathbf{M}$ be a linear operator on $L^1(\mathbb{R}^n)$. Then the following properties are equivalent for any integer $k > 0$:*

1. *$\mathbf{M}$ is injective on $\Sigma_{k,n}$.*
2. *$\ker(\mathbf{M}) \cap \Sigma_{2k,n} = \{0\}$.*

Proof. 1. $\Rightarrow$ 2. Suppose $\mathbf{M}$ is injective on $\Sigma_{k,n}$ and let $f \in \ker(\mathbf{M}) \cap \Sigma_{2k,n}$, that is $\mathbf{M}f = 0$ and

$$f = \sum_{s=1}^{2k} \lambda_s p_{\boldsymbol{\mu}_s}, \tag{A.5}$$

with $\lambda_s \in \mathbb{R}$ and $p_{\boldsymbol{\mu}_s} \in \mathbb{R}^n$ for all s. Since $\mathbf{M}f = 0$, we can apply $\mathbf{M}$ to (A.5) to obtain

$$\mathbf{M}\left(\sum_{s=1}^{k} \lambda_s p_{\boldsymbol{\mu}_s}\right) = \mathbf{M}\left(\sum_{s=k+1}^{2k} -\lambda_s p_{\boldsymbol{\mu}_s}\right). \tag{A.6}$$

The left and right-hand side of (A.6) are images by $\mathbf{M}$ of functions of $\Sigma_{k,n}$. Since $\mathbf{M}$ is injective on $\Sigma_{k,n}$, these functions are equal, which yields

$$f = \sum_{s=1}^{2k} \lambda_s p_{\boldsymbol{\mu}_s} = 0. \tag{A.7}$$

Therefore, $\ker(\mathbf{M}) \cap \Sigma_{2k,n} = \{0\}$.

2. $\Rightarrow$ 1. Let's suppose $\ker(\mathbf{M}) \cap \Sigma_{2k,n} = \{0\}$. Let $\boldsymbol{\mu}_1, \ldots, \boldsymbol{\mu}_k, \boldsymbol{\nu}_1, \ldots, \boldsymbol{\nu}_k$ be vectors of $\mathbb{R}^n$ and $\alpha_1, \ldots, \alpha_k, \beta_1, \ldots, \beta_k$ be real numbers such that

$$\mathbf{M}\left(\sum_{s=1}^{k} \alpha_s p_{\boldsymbol{\mu}_s}\right) = \mathbf{M}\left(\sum_{s=1}^{k} \beta_s p_{\boldsymbol{\nu}_s}\right). \tag{A.8}$$

By subtracting the right hand side in (A.8), we get

$$\mathbf{M}\left(\sum_{s=1}^{k} \alpha_s p_{\boldsymbol{\mu}_s} - \sum_{s=1}^{k} \beta_s p_{\boldsymbol{\nu}_s}\right) = 0. \tag{A.9}$$

The argument is a function of $\ker(\mathbf{M}) \cap \Sigma_{2k,n}$, so it is zero by hypothesis. Therefore, we necessarily have

$$\sum_{s=1}^{k} \alpha_s p_{\boldsymbol{\mu}_s} = \sum_{s=1}^{k} \beta_s p_{\boldsymbol{\nu}_s} \tag{A.10}$$

and $\mathbf{M}$ is injective on $\Sigma_{k,n}$. □

In the proofs of Theorems 1 and 2, we will actually prove that $\ker(\mathbf{M}) \cap \Sigma_{2k,n} = \{0\}$. Lemma 6 states that this implies the injectivity result we want to obtain. The converse will also be used to prove Theorem 2 from Theorem 1.

A.2 Proof of Theorem 1

In this section, we fix $n = 1$. The numbers $\omega_1, \ldots, \omega_{2k}$ defined in the statement of the theorem are also fixed, and we note $\boldsymbol{\omega} = (\omega_1, \ldots, \omega_{2k}) \in \mathbb{R}^{2k}$. We define $\mathbf{M}$ as the linear Fourier sampling operator on $L^1(\mathbb{R})$ associated to the $4k^2$ frequencies defined as the $2k$ first nonzero multiples of the ω_j.

Let's now define a family of finite-dimensional linear operators which will be injective iff $\mathbf{M}$ is injective, as stated by the following lemma. Given real numbers ω and μ, define the function

$$h_\omega(a) = \exp\left(-\frac{\sigma^2}{2}\omega^2\right) \exp\left(-i\omega\mu\right). \tag{A.11}$$

Since $h_\omega(\mu) = \mathbf{M}_\omega(p_\mu)$, this new notation will be used to simplify the expressions. Given $\omega \in \mathbb{R}$ and an ℓ-dimensional vector $\boldsymbol{\mu} = (\mu_1, \ldots, \mu_\ell)$, we denote by $\mathbf{N}_{\omega,\boldsymbol{\mu}}$ the following matrix:

$$\mathbf{N}_{\omega,\boldsymbol{\mu}} = \begin{pmatrix} h_\omega(\mu_1) & \ldots & h_\omega(\mu_\ell) \\ h_{2\omega}(\mu_1) & \ldots & h_{2\omega}(\mu_\ell) \\ \vdots & & \vdots \\ h_{2k\omega}(\mu_1) & \ldots & h_{2k\omega}(\mu_\ell) \end{pmatrix} \in \mathcal{M}_{2k,\ell}(\mathbb{R}). \tag{A.12}$$

We note $\mathbf{N}_{\boldsymbol{\mu}}$ the matrix:

$$\mathbf{N}_{\boldsymbol{\mu}} = \begin{pmatrix} \mathbf{N}_{\omega_1,\boldsymbol{\mu}} \\ \vdots \\ \mathbf{N}_{\omega_{2k},\boldsymbol{\mu}} \end{pmatrix} \in \mathcal{M}_{4k^2,\ell}(\mathbb{R}). \tag{A.13}$$

The following lemma links the injectivity of $\mathbf{M}$ and the injectivity of the $\mathbf{N}_{\boldsymbol{\mu}}$.

Lemma 7. *The operator $\mathbf{M}$ is injective on $\Sigma_{k,n}$ if and only if the operator $\mathbf{N}_{\boldsymbol{\mu}}$ is injective for any vector $\boldsymbol{\mu} \in \mathbb{R}^{2k}$ composed of distinct entries.*

Proof. Let's first remark that for all coefficients $\boldsymbol{\alpha} = (\alpha_1, \ldots, \alpha_k) \in \mathbb{R}^k$ and distinct means $\boldsymbol{\mu} = (\mu_1, \ldots, \mu_k) \in \mathbb{R}^k$, we have $\mathbf{M}\left(\sum_{s=1}^{k} \alpha_s p_{\mu_s}\right) = \mathbf{N}_{\boldsymbol{\mu}}\boldsymbol{\alpha}$. The matrix $\mathbf{N}_{\boldsymbol{\mu}}$ thus allows us to interpret $\mathbf{M}$ as a finite-dimensional linear map for linear combinations of Gaussians in the support $\boldsymbol{\mu}$. Let's now prove the equivalence.

Necessary condition. Suppose $\mathbf{M}$ is injective on $\Sigma_{k,n}$. Let $\boldsymbol{\mu} \in \mathbb{R}^{2k}$ be a vector composed of distinct entries and $\boldsymbol{\gamma} \in \ker(\mathbf{N}_{\boldsymbol{\mu}})$. This implies

$$\mathbf{M}\left(\sum_{s=1}^{2k} \gamma_s p_{\mu_s}\right) = 0. \tag{A.14}$$

Since $\mathbf{M}$ is injective on $\Sigma_{k,n}$, Lemma 6 ensures that

$$\sum_{s=1}^{2k} \gamma_s p_{\mu_s} = 0. \tag{A.15}$$

Since the μ_s are distinct, Lemma 5 implies that all γ_s are 0, so that $\boldsymbol{\gamma} = 0$. Finally, $\ker(\mathbf{N}_{\boldsymbol{\mu}}) = \{0\}$ and $\mathbf{N}_{\boldsymbol{\mu}}$ is injective.

Sufficient condition. Suppose $\mathbf{N}_{\boldsymbol{\mu}}$ is injective for all $\boldsymbol{\mu} \in \mathbb{R}^{2k}$ composed of distinct entries. Let $\boldsymbol{\alpha} = (\alpha_1, \dots, \alpha_k)$, $\boldsymbol{\beta} = (\beta_1, \dots, \beta_k)$, $\boldsymbol{\theta} = (\theta_1, \dots, \theta_k)$ and $\boldsymbol{\nu} = (\nu_1, \dots, \nu_k)$ be such that:

$$\mathbf{M}\left(\sum_{s=1}^{k} \alpha_s p_{\theta_s}\right) = \mathbf{M}\left(\sum_{s=1}^{k} \beta_s p_{\nu_s}\right). \tag{A.16}$$

With a proper reordering of the terms of the two sums, we can put this last inequality in the form:

$$\mathbf{M}\left(\sum_{s=1}^{\ell} (\alpha_s - \beta_s) p_{\theta_s} + \sum_{s=\ell+1}^{k} \alpha_s p_{\theta_s} + \sum_{s=\ell+1}^{k} -\beta_s p_{\mu_s} + \sum_{s=1}^{\ell} 0.p_{\xi_s}\right) = 0, \tag{A.17}$$

where the first sum is the gathering of the common functions between the two sums in (A.16), the second and third are the distinct functions, and the fourth is composed of any other functions p_{ξ_s} chosen so that the linear combination comprises $2k$ terms composed of distinct functions.

The injectivity of $\mathbf{N}_{\boldsymbol{\mu}}$ for $\boldsymbol{\mu} = (\theta_1, \dots, \theta_k, \nu_{\ell+1}, \nu_k, \xi_1, \dots, \xi_\ell)$ gives that $\alpha_s = \beta_s$ for all $s \in [\![1, \ell]\!]$ and $\alpha_s = \beta_s = 0$ for all $s \in [\![\ell + 1, k]\!]$, so that

$$\sum_{s=1}^{k} \alpha_s p_{\theta_s} = \sum_{s=1}^{k} \beta_s p_{\nu_s}, \tag{A.18}$$

which gives the injectivity of $\mathbf{M}$ on $\Sigma_{k,n}$.

□

We are now ready to prove Theorem 1. Lemma 7 casts the initial claim as the injectivity of $\mathbf{N}_{\boldsymbol{\mu}}$ for all $\boldsymbol{\mu} \in \mathbb{R}^{2k}$ with distinct entries. For such a vector $\boldsymbol{\mu}$, we have

$$\ker(\mathbf{N}_{\boldsymbol{\mu}}) = \bigcap_{s=1,\dots,2k} \ker(\mathbf{N}_{\omega_s, \boldsymbol{\mu}}). \tag{A.19}$$

Let's therefore study $\ker(\mathbf{N}_{\omega,\boldsymbol{\mu}})$ for $\omega \in \mathbb{R}$. For $s \in \{1, \dots, 2k\}$, in order to simplify notations, let's note $\beta_s = \exp\left(-\frac{\sigma^2}{2} s^2 \omega^2\right)$ and $x_s = \exp(-i\omega\mu_s)$.

$\mathbf{N}_{\omega,\boldsymbol{\mu}}$ can be written as:

$$\mathbf{N}_{\omega,\boldsymbol{\mu}} = \begin{pmatrix} \beta_1 x_1 & \dots & \beta_1 x_{2k} \\ \beta_2 x_1^2 & \dots & \beta_2 x_{2k}^2 \\ \vdots & & \vdots \\ \beta_{2k} x_1^{2k} & \dots & \beta_{2k} x_{2k}^{2k} \end{pmatrix}. \tag{A.20}$$

The determinant of this matrix is proportional (within a nonzero factor) to a Vandermonde determinant. Therefore, this matrix has a nonzero kernel if and only if $\exists s \neq t$ such that $x_s = x_t$, which is equivalent to $\mu_s = \mu_t \left[\frac{2\pi}{\omega}\right]$.

The structure of the matrix gives the form of $\ker(\mathbf{N}_{\omega,\boldsymbol{\mu}})$ which is determined by the equivalence classes of the μ_s in $\mathbb{R}\backslash\frac{2\pi}{\omega}\mathbb{Z}$. More precisely, if there are $P \leq 2k$ such classes and if we note $I_1, \ldots, I_P$ the partition of $[\![1, 2k]\!]$ such that

$$\forall j, \forall s, t \in I_j,\ \mu_s = \mu_t \left[\frac{2\pi}{\omega}\right] \tag{A.21}$$

then

$$\ker(\mathbf{N}_{\omega,\boldsymbol{\mu}}) = \bigoplus_{j=1,\ldots,P} U(I_j), \tag{A.22}$$

where

$$U(I_j) = \left\{ \mathbf{u} = (u_1, \ldots, u_{2k}) \in \mathbb{R}^{2k} : \sum_{s\in I_j} u_s = 0,\ \forall s \notin I_j,\ u_s = 0 \right\}. \tag{A.23}$$

Let's remark that $U(I_j) = \{0\}$ if and only if I_j is a singleton. Let's now show that the choice taken for the frequencies is sufficient to yield the injectivity of $\mathbf{N}_{\boldsymbol{\mu}}$.

For $t = 1, \ldots, 2k$, we note $I_1^t, \ldots, I_{P_t}^t$ the partition induced on $[\![1, 2k]\!]$ by the equivalence relation of equality modulo $\frac{2\pi}{\omega_t}$ for the $\mu_s, s \in [\![1, 2k]\!]$. To prove the theorem, it is sufficient to prove that:

$$\forall s \in [\![1, 2k]\!],\ \exists t \in [\![1, 2k]\!],\ \exists j \in [\![1, P_t]\!],\ I_j^t = \{s\}. \tag{A.24}$$

Indeed, this will prove that $\forall s \in [\![1, 2k]\!]$, $\exists t \in [\![1, 2k]\!]$, $\ker(\mathbf{N}_{\omega_t,\boldsymbol{\mu}}) \perp \langle \mathbf{e}_s \rangle$, where $\mathbf{e}_s$ denotes the s^{th} vector of the canonical basis of $\mathbb{R}^{2k}$. We will thus have:

$$\forall s \in [\![1, 2k]\!],\ \left(\bigcap_{t\in[\![1,2k]\!]} \ker(\mathbf{N}_{\omega_t,\boldsymbol{\mu}}) \right) \perp \langle \mathbf{e}_s \rangle, \tag{A.25}$$

which will yield that

$$\bigcap_{t\in[\![1,2k]\!]} \ker(\mathbf{N}_{\omega_t,\boldsymbol{\mu}}) = \{0\}. \tag{A.26}$$

Let's therefore prove (A.24). The proof relies on the following fact: if $x \neq y$ are in I_ℓ^s for some ℓ and s, then for all $t \neq s$, x and y cannot belong to a same I_j^t. Indeed, by contraposition, if x and y belong to I_ℓ^s and I_j^t for $s \neq t$, we have $\mu_x - \mu_y = 0\left[\frac{2\pi}{\omega_s}\right] = 0\left[\frac{2\pi}{\omega_t}\right]$. By hypothesis, $\frac{\omega_s}{\omega_t} \notin \mathbb{Q}$, so that $\frac{2\pi}{\omega_s}\mathbb{Z} \cap \frac{2\pi}{\omega_t}\mathbb{Z} = \{0\}$. Therefore, $\mu_x = \mu_y$ and $x = y$ since $\boldsymbol{\mu}$ is composed of distinct entries.

Let $s \in [\![1, 2k]\!]$. The above result tells us that for any $t \neq s$, t and s are in the same subset in at most one of these partitions. Since there are $2k$ partitions and only $2k - 1$ elements t distinct from s, there is necessarily at least one partition in which $\{s\}$ is a subset. This yields (A.24), and therefore the result.

A.3 Proof of Theorem 2

We fix $\omega_1, \ldots, \omega_{2k}$ as real frequencies satisfying the hypotheses of Theorem 1.

Given a family $\mathcal{U} = \{\mathbf{u}_1, \ldots, \mathbf{u}_m\} \subset \mathbb{R}^n$, we note $\mathbf{M}(\mathcal{U})$ the functional operator which samples the Fourier transform of a function at every frequency $x\omega_y\mathbf{u}_j$ for $(x, y, j) \in [\![1, 2k]\!]^2 \times [\![1, m]\!]$.

Let's first give a sufficient condition on $\mathcal{U}$ such that $\mathbf{M}(\mathcal{U})$ is injective on $\Sigma_{k,n}$.

Lemma 8. *Suppose $\mathcal{U}$ is a family of m vectors in $\mathbb{R}^n$ satisfying the following property: for all distinct vectors $\mathbf{x}_1, \dots, \mathbf{x}_{2k}$ of $\mathbb{R}^n$, for all $t \in [\![1, 2k]\!]$, there exists $\mathbf{u} \in U$ satisfying*

$$\forall s \in [\![1, 2k]\!],\ s \neq t \Rightarrow \langle \mathbf{x}_s - \mathbf{x}_t, \mathbf{u} \rangle \neq 0. \tag{A.27}$$

Then $\mathbf{M}(\mathcal{U})$ is injective on $\Sigma_{k,n}$.

Proof. The assumption on $\mathcal{U}$ means that for any collection $\mathcal{X}$ of $2k$ vectors and any vector $\mathbf{x} \in \mathcal{X}$, there exists a direction $\mathbf{u}$ in $\mathcal{U}$ such that the orthogonal projection of $\mathbf{x}$ on $\mathbf{u}$ is different from the orthogonal projection of any other vector of $\mathcal{X}$ on $\mathbf{u}$. Since $\mathcal{U}$ is fixed in the current proof, let's note $\mathbf{M} = \mathbf{M}(\mathcal{U})$. Without loss of generality, let's suppose the vectors of $\mathcal{U}$ are normalized in the ℓ_2 sense.

Using Lemma 6, it is sufficient to prove that for all collection of vectors $\boldsymbol{\mu}_1, \dots, \boldsymbol{\mu}_{2k} \in \mathbb{R}^n$ and all real numbers $\lambda_1, \dots, \lambda_{2k}$,

$$\mathbf{M}\left(\sum_{s=1}^{2k} \lambda_s p_{\boldsymbol{\mu}_s}\right) = 0 \Rightarrow \sum_{s=1}^{2k} \lambda_s p_{\boldsymbol{\mu}_s} = 0. \tag{A.28}$$

Without loss of generality, we can suppose the $\boldsymbol{\mu}_s$ are distinct (if some of them are equal, we can gather the common values and add any other $p_{\boldsymbol{\mu}}$ with a coefficient 0 until we get $2k$ terms). We note $p = \sum_{s=1}^{2k} \lambda_s p_{\boldsymbol{\mu}_s}$.

We will now prove that every λ_t is 0, which will prove the result. Let $t \in [\![1, 2k]\!]$. From the hypothesis, there exists $\mathbf{u} \in \mathcal{U}$ such that (A.27) holds. In the expression of $\mathbf{M}p$ as a vector of dimension $4k^2m$, let's consider the $4k^2$ entries $\mathbf{M}_{x\omega_j}p$, with $(x, j) \in [\![1, 2k]\!]^2$. Using (2.18), we can get their expressions in terms of p:

$$0 = \mathbf{M}_{x\omega_j \mathbf{u}}(p) = \sum_{s=1}^{2k} \lambda_s \mathcal{F}(p_{\boldsymbol{\mu}_s}).(x\omega_j \mathbf{u}) = \sum_{s=1}^{2k} \lambda_s \exp\left(-\frac{\sigma^2}{2} x^2 \omega_j^2\right) \exp(-ix\omega_j \langle \mathbf{u}, \boldsymbol{\mu}_s \rangle). \tag{A.29}$$

Let's now remark that the above quantity is the Fourier transform at frequency $x\omega_j$ of the mixture of 1-dimensional Gaussians

$$q = \sum_{s=1}^{2k} \lambda_s p_{\langle \mathbf{u}, \boldsymbol{\mu}_s \rangle}. \tag{A.30}$$

Let $\mathbf{N}$ be the operator defined on $L^1(\mathbb{R})$ which samples the Fourier transform of a function at frequencies $x\omega_j$ for $(x, j) \in [\![1, 2k]\!]^2$. From the above reasoning, we have $\mathbf{N}q = 0$, so $q \in \ker(\mathbf{N})$.

Theorem 1 ensures that $\mathbf{N}$ is injective on $\Sigma_{k,1}$ and therefore Lemma 6 ensures that $\ker(\mathbf{N}) \cap \Sigma_{2k,1} = \{0\}$. Thus, $q = 0$. But the hypothesis on $\mathbf{u}$ ensures that $\langle \mathbf{u}, \boldsymbol{\mu}_t \rangle \neq \langle \mathbf{u}, \boldsymbol{\mu}_s \rangle$ for all $s \neq t$. By regrouping the same functions $p_{\langle \mathbf{u}, \boldsymbol{\mu}_s \rangle}$ in the expression (A.30), we get that

$$q = \lambda_t p_{\langle \mathbf{u}, \boldsymbol{\mu}_t \rangle} + \sum_s \xi_s p_{\langle \mathbf{u}, \boldsymbol{\mu}_s \rangle}, \tag{A.31}$$

where the second sum contains at most $2k - 1$ terms and the $\boldsymbol{\mu}_s$ and $\boldsymbol{\mu}_t$ are distinct. Lemma 5 ensures that, since $q = 0$, we have $\lambda_t = 0$. This is valid for all $t \in [\![1, 2k]\!]$, which proves the result.

□

Let's now prove a final lemma, which states the existence of certain remarkable families of vectors in $\mathbb{R}^n$ which will be used in the rest of the proof.

Lemma 9. *For any integer $q \geq n$, there exists a family $\mathcal{B}_q$ of vectors of $\mathbb{R}^n$ such that any subfamily of n vectors of $\mathcal{B}_q$ is a basis of $\mathbb{R}^n$.*

Proof. We prove this by induction on q. For $q = n$, the property is trivial: it is sufficient to define $\mathcal{B}_q$ as a basis of $\mathbb{R}^n$.

Let's suppose the property is satisfied for $q \geq n$. Consider the hyperplanes of $\mathbb{R}^n$ spanned by all the collections of $n-1$ vectors of $\mathcal{B}_q$: there are finitely many such hyperplanes. The union of these hyperplanes cannot recover $\mathbb{R}^n$: a straightforward argument to justify this claim is to invoke Baire's theorem, since $\mathbb{R}^n$ is complete and a hyperplane is a closed set of empty interior. Therefore, there exists a vector $\mathbf{x} \in \mathbb{R}^n$ which is not in this union.

Let's define $\mathcal{B}_{q+1} = \mathcal{B}_q \cup \{\mathbf{x}\}$, and let $\mathcal{U}$ be a subfamily of n vectors of $\mathcal{B}_{q+1}$. If $\mathcal{U} \subset \mathcal{B}_q$, then $\mathcal{U}$ is a basis of $\mathbb{R}^n$ by hypothesis. If not, then $\mathcal{U}$ is composed of $\mathbf{x}$ and $n-1$ vectors of $\mathcal{B}_q$. These $n-1$ vectors are linearly independent: if not, adding one other vector of $\mathcal{B}_q$ to them would produce a subfamily of n vectors of $\mathcal{B}_q$ which is not a basis. Moreover, by hypothesis on $\mathbf{x}$, it is not included in the hyperplane spanned by these $n-1$ vectors, so that $\mathcal{U}$ is a linearly independent family of $\mathbb{R}^n$ composed of n vectors, that is a basis of $\mathbb{R}^n$. This proves the result. □

Let's now finish the proof of the theorem. Let m be an integer $> (2k-1)(n-1)$ and pose $\mathcal{U} = \mathcal{B}_m$, where $\mathcal{B}_m$ is the corresponding set in Lemma 9. We will prove that $\mathcal{U}$ satisfies the hypothesis of Lemma 8, which will prove the theorem.

Let $\mathbf{x}_1, \ldots, \mathbf{x}_{2k}$ be a family of distinct vectors of $\mathbb{R}^n$, and $t \in [\![1, 2k]\!]$. For any $s \neq t$, there are at most $n-1$ vectors $\mathbf{u}$ of $\mathcal{U}$ such that $\langle \mathbf{u}, \mathbf{x}_t \rangle = \langle \mathbf{u}, \mathbf{x}_s \rangle$: indeed, if there are n such vectors, they form a basis and we would therefore have $\mathbf{x}_t = \mathbf{x}_s$, which contradicts the fact that the vectors are distinct. Therefore, the set

$$\bigcup_{s \neq t} \{\mathbf{u} \in \mathcal{U} : \langle \mathbf{u}, \mathbf{x}_t \rangle = \langle \mathbf{u}, \mathbf{x}_s \rangle\} \tag{A.32}$$

is of cardinal $\leq (2k-1)(n-1)$. Since $m > (2k-1)(n-1)$, there exists a vector $\mathbf{u} \in U$ such that for all $s \neq t$, $\langle \mathbf{u}, \mathbf{x}_t \rangle \neq \langle \mathbf{u}, \mathbf{x}_s \rangle$. This is valid for all t, so that $\mathcal{U}$ satisfies the hypothesis of Lemma 8, which proves the result, since we can pose $m = (2k-1)(n-1) + 1 \leq 2kn$.

Appendix B

Proofs of Chapter 3 theorems

B.1 Well-posedness of the finite UoS decoder

In this section, we will prove that if Σ is a finite union of subspaces in $\mathbb{R}^n$ and $\|.\|$ a norm on $\mathbb{R}^n$, then the quantity $\arg\min_{\mathbf{z}\in(\mathbf{x}+\mathcal{N})} d(\mathbf{z},\Sigma)$, where d is the distance relative to $\|.\|$, is defined for all $\mathbf{x}\in\mathbb{R}^n$.

Let's first prove the following lemma:

Lemma 10. *Let V and W be two subspaces of $\mathbb{R}^n$ and $\|.\|$ a norm on $\mathbb{R}^n$. Then $\forall \mathbf{x}\in\mathbb{R}^n, \exists \mathbf{y}\in(\mathbf{x}+V)$ such that $d(\mathbf{y},W)=d(\mathbf{x}+V,W)$, where d is the distance derived from $\|.\|$.*

Proof. Let Φ be defined on $V+W$ by $\Phi(\mathbf{u})=\|\mathbf{u}-\mathbf{x}\|$. Since $\Phi(\mathbf{u})\geq\|\mathbf{u}\|-\|\mathbf{x}\|$, we have $\lim_{\|\mathbf{u}\|\to+\infty}\Phi(\mathbf{u})=+\infty$, so that $\exists M>0$ such that $\|\mathbf{u}\|>M\Rightarrow\Phi(\mathbf{u})\geq\|\mathbf{x}\|$. The set $B=\{\mathbf{u}\in V+W,\|\mathbf{u}\|\leq M\}$ is a closed ball of $V+W$ and is thus a compact. Since Φ is continuous, Φ has a minimizer $\mathbf{v}$ on B. $0\in B$, so that $\Phi(0)=\|\mathbf{x}\|\geq\Phi(\mathbf{v})$. For all $\mathbf{u}$ such that $\|\mathbf{u}\|>M$, we have $\Phi(\mathbf{u})\geq\|\mathbf{x}\|\geq\Phi(\mathbf{v})$, so that $\mathbf{v}$ is a global minimizer of Φ.

We therefore have $\forall(\mathbf{u},\mathbf{w})\in V\times W,\|\mathbf{x}-\mathbf{v}\|\leq\|\mathbf{x}-(\mathbf{u}+\mathbf{w})\|$. The vector $\mathbf{v}$ can be written $\mathbf{f}+\mathbf{g}$ with $\mathbf{f}\in V$ and $\mathbf{g}\in W$, so that the vector $\mathbf{y}=\mathbf{x}-\mathbf{f}$, which belongs to $\mathbf{x}+V$, satisfies $d(\mathbf{x}-\mathbf{f},W)=\|(\mathbf{x}-\mathbf{f})-\mathbf{g}\|=d(\mathbf{x},V+W)=d(\mathbf{x}+V,W)$, which proves the result.

□

Let $\Sigma=\cup_{i\in[\![1,p]\!]}V_i$, where V_i are subspaces of $\mathbb{R}^n$. Lemma 10 applied to $V=\mathcal{N}$ and $W=V_i$ ensures the existence of $\mathbf{x}_i\in(\mathbf{x}+\mathcal{N})$ such that $d_E(\mathbf{x}_i,V_i)=d_E(\mathbf{x}+\mathcal{N},V_i)$. Therefore, $\Delta(\mathbf{Mx})$ can be defined as

$$\underset{\{\mathbf{x}_i,i\in[\![1,p]\!]\}}{\operatorname{argmin}} d_E(\mathbf{x}_i,V_i)$$

and satisfies $d_E(\Delta(\mathbf{Mx}),\Sigma)=d_E(\mathbf{x}+\mathcal{N},\Sigma)$, so that the decoder

$$\Delta(\mathbf{Mx})=\underset{\mathbf{z}\in(\mathbf{x}+\mathcal{N})}{\operatorname{argmin}} d(\mathbf{z},\Sigma)$$

is properly defined. In particular, this applies to the decoder (3.4).

B.2 Proof of Theorem 3

Let $\delta > 0$ and Δ_δ and C be such that (3.13) holds $\forall \mathbf{x} \in E$. Let $\mathbf{h} \in \mathcal{N}$. Then $\exists \mathbf{h}_0 \in \Sigma - \Sigma$ such that $d_E(\mathbf{h}, \mathbf{h}_0) \leq d_E(\mathbf{h}, \Sigma - \Sigma) + \delta$. Let $\mathbf{h}_0 = \mathbf{h}_1 - \mathbf{h}_2$ with $\mathbf{h}_1, \mathbf{h}_2 \in \Sigma$, and $\mathbf{h}_3 = \mathbf{h} - \mathbf{h}_0$. Since $\mathbf{h} \in \mathcal{N}$, we have:

$$\mathbf{M}(\mathbf{h}_1 + \mathbf{h}_3) = \mathbf{M}\mathbf{h}_2. \tag{B.1}$$

Applying (3.13) to $\mathbf{x} = \mathbf{h}_2 \in \Sigma$ and using the fact that $\|0\|_E = 0$, we get:

$$\|\mathbf{A}\mathbf{h}_2 - \Delta_\delta(\mathbf{M}\mathbf{h}_2)\|_G \leq \delta. \tag{B.2}$$

Let's now find an upper bound for $\|\mathbf{A}\mathbf{h}\|_G$:

$$\begin{aligned}
\|\mathbf{A}\mathbf{h}\|_G &= \|\mathbf{A}(\mathbf{h}_1 - \mathbf{h}_2 + \mathbf{h}_3)\|_G \\
&= \|\mathbf{A}(\mathbf{h}_1 + \mathbf{h}_3) - \Delta_\delta(\mathbf{M}(\mathbf{h}_1 + \mathbf{h}_3)) - \mathbf{A}\mathbf{h}_2 + \Delta_\delta(\mathbf{M}(\mathbf{h}_1 + \mathbf{h}_3))\|_G \\
&\leq \|\mathbf{A}(\mathbf{h}_1 + \mathbf{h}_3) - \Delta_\delta(\mathbf{M}(\mathbf{h}_1 + \mathbf{h}_3))\|_G + \|\mathbf{A}\mathbf{h}_2 - \Delta_\delta(\mathbf{M}(\mathbf{h}_1 + \mathbf{h}_3))\|_G
\end{aligned} \tag{B.3}$$

where we have used (3.11) and (3.12) for the last inequality. Combining (B.1) and (B.2), we get that:

$$\|\mathbf{A}\mathbf{h}_2 - \Delta_\delta(\mathbf{M}(\mathbf{h}_1 + \mathbf{h}_3))\|_G \leq \delta. \tag{B.4}$$

Applying (3.13) to $\mathbf{x} = \mathbf{h}_1 + \mathbf{h}_3$, we get:

$$\begin{aligned}
&\|\mathbf{A}(\mathbf{h}_1 + \mathbf{h}_3) - \Delta_\delta(\mathbf{M}(\mathbf{h}_1 + \mathbf{h}_3))\|_G \\
&\leq C d_Y(\mathbf{h}_1 + \mathbf{h}_3, \Sigma) + \delta \leq C\|\mathbf{h}_3\|_E + \delta \\
&= C d_Y(\mathbf{h}, \mathbf{h}_0) + \delta \leq C d_Y(\mathbf{h}, \Sigma - \Sigma) + (C+1)\delta.
\end{aligned} \tag{B.5}$$

Combining (B.3), (B.4) and (B.5) gives:

$$\|\mathbf{A}\mathbf{h}\|_G \leq C d_Y(\mathbf{h}, \Sigma - \Sigma) + (C+2)\delta. \tag{B.6}$$

(B.6) is valid for all $\delta > 0$, so it is valid for $\delta = 0$. This gives us the property (3.6) with $D = C$.

B.3 Proof of Theorem 4

Let's first assume that (3.14) holds and define the following decoder on F:

$$\Delta'(\mathbf{M}\mathbf{x}) = \underset{\mathbf{z} \in (\mathbf{x} + \mathcal{N})}{\operatorname{argmin}} d_E(\mathbf{z}, \Sigma). \tag{B.7}$$

Note that the decoder is well defined, since $\mathbf{M}\mathbf{x}_1 = \mathbf{M}\mathbf{x}_2 \Rightarrow \mathbf{x}_1 + \mathcal{N} = \mathbf{x}_2 + \mathcal{N}$.

For $\mathbf{x} \in E$, we have $\mathbf{x} - \Delta'(\mathbf{M}\mathbf{x}) \in \mathcal{N}$, so that (3.6) yields:

$$\begin{aligned}
\|\mathbf{A}\mathbf{x} - \mathbf{A}\Delta'(\mathbf{M}\mathbf{x})\|_G &\leq D d_E(\mathbf{x} - \Delta'(\mathbf{M}\mathbf{x}), \Sigma - \Sigma) \\
&\leq D d_E(\mathbf{x}, \Sigma) + D d_E(\Delta'(\mathbf{M}\mathbf{x}), \Sigma) \\
&\leq 2D d_E(\mathbf{x}, \Sigma),
\end{aligned} \tag{B.8}$$

where we have used (3.12) for the second inequality. The last inequality comes from (B.7), which yields $d_E(\Delta'(\mathbf{M}\mathbf{x}), \Sigma) \leq d_E(\mathbf{x}, \Sigma)$. Therefore, by posing $\Delta = \mathbf{A}\Delta'$, we get (3.5).

Let's return to the general case, and consider $\nu > 0$. We define the following decoder on F:

$$\Delta'_\nu(\mathbf{M}\mathbf{x}) \in \{\mathbf{u} \in (\mathbf{x} + \mathcal{N}) | d_E(\mathbf{u}, \Sigma) \leq d_E(\mathbf{x} + \mathcal{N}, \Sigma) + \nu\}. \tag{B.9}$$

Note that this set may not contain a unique element and thus this definition relies on the axiom of choice.

For $\mathbf{x} \in E$, we have again $\mathbf{x} - \Delta'_\nu(\mathbf{Mx}) \in \mathcal{N}$, so that by (3.6):

$$\begin{aligned}\|\mathbf{Ax} - \mathbf{A}\Delta'_\nu(\mathbf{Mx})\|_G &\leq D d_E(\mathbf{x} - \Delta'_\nu(\mathbf{Mx}), \Sigma - \Sigma) \\ &\leq D d_E(\mathbf{x}, \Sigma) + D d_E(\Delta'_\nu(\mathbf{Mx}), \Sigma) \\ &\leq 2D d_E(\mathbf{x}, \Sigma) + D\nu, \end{aligned} \tag{B.10}$$

where we have used (3.12) again for the second inequality. The last inequality comes from (B.9), which yields $d_E(\Delta'_\nu(\mathbf{Mx}), \Sigma) \leq d_E(\mathbf{x}, \Sigma) + \nu$. Therefore, by posing $\Delta_\delta = \mathbf{A}\Delta'_{\delta/D}$, we get (3.13).

B.4 Proof of Proposition 1

Let $\mathbf{x} \in E$ and $\nu > 0$. If $0 = d_E(\mathbf{x} + \mathcal{N}, \Sigma) = d_E(\mathbf{x}, \Sigma + \mathcal{N})$, then since $\Sigma + \mathcal{N}$ is a closed set, $\mathbf{x} \in \Sigma + \mathcal{N}$, and therefore $(\mathbf{x} + \mathcal{N}) \cap \Sigma \neq \emptyset$. In this case, we define $\Delta'_\nu(\mathbf{Mx})$ as any element of $(\mathbf{x} + \mathcal{N}) \cap \Sigma$.

If $d_E(\mathbf{x} + \mathcal{N}, \Sigma) > 0$, then we define

$$\Delta'_\nu(\mathbf{Mx}) \in \{\mathbf{u} \in (\mathbf{x} + \mathcal{N}) | d_E(\mathbf{u}, \Sigma) \leq (1 + \nu) d_E(\mathbf{x} + \mathcal{N}, \Sigma)\}.$$

This provides a consistent definition of Δ'_ν.

Let's remark that for all $\mathbf{x} \in E$,

$$\begin{aligned} d_E(\Delta'_\nu(\mathbf{Mx}), \Sigma) &\leq (1 + \nu) d_E(\mathbf{x} + \mathcal{N}, \Sigma) \\ &\leq (1 + \nu) d_E(\mathbf{x}, \Sigma) \end{aligned}$$

For $\mathbf{x} \in E$, $\mathbf{x} - \Delta'_\nu(\mathbf{Mx}) \in \mathcal{N}$, so that (3.6) gives:

$$\begin{aligned}\|\mathbf{Ax} - \mathbf{A}\Delta'_\nu(\mathbf{Mx})\|_G &\leq D d_E(\mathbf{x} - \Delta'_\nu(\mathbf{Mx}), \Sigma - \Sigma) \\ &\leq D d_E(\mathbf{x}, \Sigma) + D d_E(\Delta'_\nu(\mathbf{Mx}), \Sigma) \\ &\leq (2 + \nu) D d_E(\mathbf{x}, \Sigma). \end{aligned} \tag{B.11}$$

Defining $\Delta_\delta = \mathbf{A}\Delta'_\nu$, we get the desired result.

B.5 Proof of Theorem 5 and Theorem 7

Let's first remark that applying (3.19) (resp. (3.21)) with $\mathbf{x} = \mathbf{z} \in \Sigma$ and $\mathbf{e} = 0$ yields $\|\mathbf{Az} - \Delta_\delta(\mathbf{Mz})\|_G \leq \delta$ and $\|\mathbf{Az} - \Delta_{\delta,\epsilon}(\mathbf{Mz})\|_G \leq C_2\epsilon + \delta$ for any $\mathbf{z} \in \Sigma$, $\epsilon \geq 0$, where we have used the fact that $\|0\|_F = 0$.

Let $\mathbf{h} \in E$ and $\mathbf{z} \in \Sigma$. We apply (3.19) (resp. (3.21)) with $\mathbf{x} = \mathbf{z} - \mathbf{h}$, $\mathbf{e} = \mathbf{Mh}$, and $\epsilon = \|\mathbf{Mh}\|_F$, which yields:

$$\begin{aligned}\|\mathbf{Az} - \mathbf{Ah} - \Delta_\delta(\mathbf{Mz})\|_G &\leq C_1 d_Y(\mathbf{z} - \mathbf{h}, \Sigma) + C_2\|\mathbf{Mh}\|_F + \delta. \\ \|\mathbf{Az} - \mathbf{Ah} - \Delta_{\delta,\epsilon}(\mathbf{Mz})\|_G &\leq C_1 d_Y(\mathbf{z} - \mathbf{h}, \Sigma) + C_2\|\mathbf{Mh}\|_F + \delta.\end{aligned}$$

Let's remark that (3.11) and (3.12) imply for all $\mathbf{x}, \mathbf{y} \in G$ $\|\mathbf{y}\|_G \leq \|\mathbf{x} - \mathbf{y}\|_G + \|\mathbf{x}\|_G$. Therefore, since $\|\mathbf{Az} - \Delta_\delta(\mathbf{Mz})\|_G \leq \delta$ (resp. $\|\mathbf{Az} - \Delta_{\delta,\epsilon}(\mathbf{Mz})\|_G \leq C_2\|\mathbf{Mh}\|_F + \delta$), we have:

$$\begin{aligned}\|\mathbf{Ah}\|_G &\leq C_1 d_Y(\mathbf{z} - \mathbf{h}, \Sigma) + C_2\|\mathbf{Mh}\|_F + 2\delta. \\ (\text{resp. } \|\mathbf{Ah}\|_G &\leq C_1 d_Y(\mathbf{z} - \mathbf{h}, \Sigma) + 2C_2\|\mathbf{Mh}\|_F + 2\delta.)\end{aligned}$$

This last inequality is valid for all $\mathbf{z} \in \Sigma$, therefore (3.19) implies:

$$\begin{aligned}\|\mathbf{Ah}\|_G &\leq C_1 \inf_{\mathbf{z}\in\Sigma} d_Y(\mathbf{z}-\mathbf{h},\Sigma) + C_2\|\mathbf{Mh}\|_F + 2\delta \\ &= C_1 \inf_{\mathbf{z}\in\Sigma,\mathbf{u}\in\Sigma} \|\mathbf{z}-\mathbf{h}-\mathbf{u}\|_E + C_2\|\mathbf{Mh}\|_F + 2\delta \\ &= C_1 d_Y(\mathbf{h},\Sigma-\Sigma) + C_2\|\mathbf{Mh}\|_F + 2\delta, \end{aligned} \tag{B.12}$$

where we have used (3.11) for the last inequality. Similarly, (3.21) implies

$$\|\mathbf{Ah}\|_G \leq C_1 d_Y(\mathbf{h},\Sigma-\Sigma) + 2C_2\|\mathbf{Mh}\|_F + 2\delta. \tag{B.13}$$

We conclude by using the fact that (B.12) and (B.13) hold for all $\delta > 0$.

B.6 Proof of Theorem 6

Let's suppose (3.20) and define for $\delta > 0$ the decoder $\Delta'_\delta : F \to E$ such that $\forall \mathbf{y} \in F$:

$$D_1 d_Y(\Delta'_\delta(\mathbf{y}),\Sigma) + D_2 d_Z(\mathbf{M}\Delta'_\delta(\mathbf{y}),\mathbf{y}) \leq \inf_{\mathbf{u}\in E} [D_1 d_Y(\mathbf{u},\Sigma) + D_2 d_Z(\mathbf{Mu},\mathbf{y})] + \delta. \tag{B.14}$$

Let's prove that this decoder meets property (3.19).

Let $\mathbf{x} \in E$ and $\mathbf{e} \in F$. Applying (3.20) with $\mathbf{h} = \mathbf{x} - \Delta'_\delta(\mathbf{Mx}+\mathbf{e})$, we get:

$$\begin{aligned}&\|\mathbf{A}(\mathbf{x}-\Delta'_\delta(\mathbf{Mx}+\mathbf{e}))\|_G \\ &\leq D_1 d_Y(\mathbf{x}-\Delta'_\delta(\mathbf{Mx}+\mathbf{e}),\Sigma-\Sigma) + D_2\|\mathbf{M}(\mathbf{x}-\Delta'_\delta(\mathbf{Mx}+\mathbf{e}))\|_F \\ &\leq D_1 d_Y(\mathbf{x},\Sigma) + D_1 d_Y(\Delta'_\delta(\mathbf{Mx}+\mathbf{e}),\Sigma) + D_2 d_Z(\mathbf{M}\Delta'_\delta(\mathbf{Mx}+\mathbf{e}),\mathbf{Mx}+\mathbf{e}) + D_2\|\mathbf{e}\|_F \\ &\leq 2D_1 d_Y(\mathbf{x},\Sigma) + 2D_2\|\mathbf{e}\|_F + \delta, \end{aligned} \tag{B.15}$$

where we have used (3.11) and (3.12) for the second inequality and the last inequality is a consequence of (B.14).

Posing $\Delta_\delta = \mathbf{A}\Delta'_\delta$ proves (3.19) with $C_1 = 2D_1$ and $C_2 = 2D_2$.

B.7 Proof of Lemma 1

The two equivalences are very similar to prove, so that we will only prove the first. (3.23) $\Rightarrow$ (3.22) is obvious. Let's now suppose (3.22), so that:

$$\forall \mathbf{h} \in \mathcal{N}, \forall \mathbf{z} \in \Sigma-\Sigma, \|\mathbf{Ah}\|_G \leq D\|\mathbf{h}-\mathbf{z}\|_E. \tag{B.16}$$

We also have:

$$\forall \lambda \in \mathbb{R}^*, \forall \mathbf{h} \in \mathcal{N}, \forall \mathbf{z} \in \Sigma-\Sigma, \|\lambda\mathbf{Ah}\|_G \leq D\|\lambda\mathbf{h}-\mathbf{z}\|_E, \tag{B.17}$$

so that:

$$\forall \lambda \in \mathbb{R}^*, \forall \mathbf{h} \in \mathcal{N}, \forall \mathbf{z} \in \Sigma-\Sigma, \|\mathbf{Ah}\|_G \leq D\|\mathbf{h}-\mathbf{z}/\lambda\|_E. \tag{B.18}$$

This last inequality yields (3.23).

B.8 Proof of Theorem 8

Let's note $\widetilde{\mathbf{M}} = \mathbf{M}_{|V}$ and $\widetilde{\mathcal{N}} = \mathcal{N} \cap V$. Let m be the dimension of the range of $\widetilde{\mathbf{M}}$, so that $\widetilde{\mathcal{N}}$ is of dimension $n - m$. Let $\mathbf{h}_1, \ldots, \mathbf{h}_{n-m}$ be an orthonormal basis of $\widetilde{\mathcal{N}}$. We have:

$$n - m = \sum_{j=1}^{n-m} \|\mathbf{h}_j\|_2^2 \leq \frac{1}{K} \sum_{j=1}^{n-m} \sum_{i=1}^{n} \langle \mathbf{h}_j, \mathbf{z}_i \rangle^2. \tag{B.19}$$

Using (3.29), we get that, for all $\mathbf{h} \in \mathcal{N}$ and unit-norm vector $\mathbf{z} \in \Sigma - \Sigma$, $\langle \mathbf{h}, \mathbf{z} \rangle^2 \leq \left(1 - \frac{1}{D_*^2}\right) \|\mathbf{h}\|_2^2$. If we denote by $p_{\widetilde{\mathcal{N}}}$ the orthogonal projection on $\widetilde{\mathcal{N}}$ and apply this inequality with $\mathbf{h} = p_{\widetilde{\mathcal{N}}}(\mathbf{z}_i) = \sum_{j=1}^{n-m} \langle \mathbf{h}_j, \mathbf{z}_i \rangle \mathbf{h}_j$ and $\mathbf{z} = \mathbf{z}_i$, we get that $\|p_{\widetilde{\mathcal{N}}}(\mathbf{z}_i)\|_2^4 \leq \left(1 - \frac{1}{D_*^2}\right) \|p_{\widetilde{\mathcal{N}}}(\mathbf{z}_i)\|_2^2$, which can be simplified to $\|p_{\widetilde{\mathcal{N}}}(\mathbf{z}_i)\|_2^2 = \sum_{j=1}^{n-m} \langle \mathbf{h}_j, \mathbf{z}_i \rangle^2 \leq \left(1 - \frac{1}{D_*^2}\right)$ even if $\|p_{\widetilde{\mathcal{N}}}(\mathbf{z}_i)\|_2 = 0$.

Using this relation in (B.19), we get:

$$n - m \leq \frac{n}{K} \left(1 - \frac{1}{D_*^2}\right), \tag{B.20}$$

so that:

$$m \geq n \left(1 - \frac{1}{K} \left(1 - \frac{1}{D_*^2}\right)\right). \tag{B.21}$$

We get the lower bound on D_*^2 by isolating it in the inequality.

B.9 Proof of Theorem 9

Let $\mathbf{h} \in E$ and $\mathbf{z} \in \Sigma - \Sigma$. We have the following inequalities:

$$\|\mathbf{h}\|_G \leq \|\mathbf{h} - \mathbf{z}\|_G + \|\mathbf{z}\|_G \leq \|\mathbf{h} - \mathbf{z}\|_G + \frac{1}{\alpha} \|\mathbf{M}\mathbf{z}\|_F, \tag{B.22}$$

where we have used the lower-RIP for the second inequality.

A similar consideration on $\mathbf{M}\mathbf{z}$ yields:

$$\|\mathbf{M}\mathbf{z}\|_F \leq \|\mathbf{M}(\mathbf{z} - \mathbf{h})\|_F + \|\mathbf{M}\mathbf{h}\|_F. \tag{B.23}$$

Substituting (B.23) into (B.22), we get:

$$\begin{aligned} \|\mathbf{h}\|_G &\leq \|\mathbf{h} - \mathbf{z}\|_G + \frac{1}{\alpha} \|\mathbf{M}(\mathbf{h} - \mathbf{z})\|_F + \frac{1}{\alpha} \|\mathbf{M}\mathbf{h}\|_F \\ &= \|\mathbf{h} - \mathbf{z}\|_{\mathbf{M}} + \frac{1}{\alpha} \|\mathbf{M}\mathbf{h}\|_F. \end{aligned} \tag{B.24}$$

Taking the infimum of the right hand-side quantity over all $\mathbf{z} \in \Sigma - \Sigma$, one gets the desired Robust NSP:

$$\|\mathbf{h}\|_G \leq d_{\mathbf{M}}(\mathbf{h}, \Sigma - \Sigma) + \frac{1}{\alpha} \|\mathbf{M}\mathbf{h}\|_F. \tag{B.25}$$