ÉCOLE POLYTECHNIQUE FÉDÉRALE DE LAUSANNE

SUPER-RESOLUTION OFF THE GRID

SEMESTER PROJECT IN COMPUTER SCIENCE

Sébastien Ollquist

7th June 2022

Abstract

Super-resolution is the tool that fundamentally allows us to increase the resolution of low resolution images. The original problem has both a practical and a theoretical meaning, the latter in which we wish to recover a superposition of point sources on a d-dimensional plane having merely access to bandlimited Fourier measurements. The goal of this project is to study a recent paper on theoretical super-resolution and discuss the main procedure that aims at recovering a collection of source points on a plane. Ultimately, some intuition will be presented on how the procedure could be improved such that its sample complexity is reduced.

ACKNOWLEDGEMENTS

I would like to take the opportunity to thank the people that contributed in some way to the progression of this project. First of all, my thanks go to my project supervisor Prof. Michael Kapralov who suggested me this very interesting paper and who supervised all my work during this semester. He managed to push me in some way to give the very best of me and even though theory is hard, I would without hesitation do this again. I would also like to express my gratitude to my co-supervisor Navid Nouri who supported me from the very begining of the project and who gave me some very helpful advice all along. Finally, I would like to thank Kshiteej Sheth and Mikhail Makarov for some very interesting discussions towards the end of the project.

> *In theory, theory and practice are the same. In practice, they are not.*
>
> Albert Einstein

Contents

CHAPTER 1

INTRODUCTION

With the increasing amount of data nowadays, it has become extremely important to develop tools that allow us to treat it better. In the case of images, today's technology often neglects important information located in the image. For example, a microscope's hardware will usually act as a low pass filter so that all the high frequency points of some image will be ignored. In essence, the resulting image will contain some noise and thus sometimes miss fine critical details. Recently however, new algorithms that are capable of recovering the missing details of some fine image given only access to some coarse measurements of a signal, have been designed. This process is called super-resolution.

The paper we will study here is called "Super-Resolution Off the Grid" and has been published by Qingqing Huang and Sham M. Kakade in September 2015. From a theoretical point of view, super-resolution is the problem of recovering a superposition of point sources using bandlimited measurements, that often may be corrupted by noise. Essentially, suppose we are given a collection of point sources that lie on the d-dimensional plane, to which we have no access whatsoever. Those point sources can often be seen as brightness spikes in a given image. We use measurements from the Fourier domain that we wish to super-resolve to recover good enough estimates of the point sources.

The goal of this project is to get familiar with the theory of super-resolution along with all important mathematical preliminaries and study the main result presented in the aforementioned paper. The idea is then to compare this work with other related ones and discuss the efficiency of the main algorithm. Ultimately, it would be a nice addition to start working on an improvement of the main algorithm in order to reduce its sample complexity from quadratic space to the information theoretical linear one.

CHAPTER 2

MATHEMATICAL REFRESHER

2.1 A MATHEMATICAL THEORY OF SUPER-RESOLUTION

MATHEMATICAL ABSTRACTION We consider k point sources in d dimensions, where the points are separated by a distance at least Δ (in Euclidean distance). The d-dimensional signal $x(t)$ can be modeled as a weighted sum of k Dirac measures in $\mathbb{R}^d$ as

$$x(t) = \sum_{j=1}^{k} w_j \delta_{\mu^{(j)}},$$

where the $\mu^{(j)}$'s are the point sources in $\mathbb{R}^d$ and $w_j \in \mathbb{C}$ the weights such that $|w_j| < C$ for every $j \in [k]$ and some absolute constant $C > 0$. Δ is by definition the minimal possible distance between any two pair of points of the d-dimensional plane. Formally, we regard Δ in terms of Euclidean distance as

$$\Delta = \min_{j \neq j'} ||\mu^{(j)} - \mu^{(j')}||_2.$$

MEASUREMENT FUNCTION There are two main factors that represent our signal: the weights w_j and the complex low pass point spread functions $e^{i\pi\langle s,t\rangle}$ for some measurement s. The former is simply a weighting factor that is not too relevant to us here. The latter however represents the response of an imaging system to a point source. That is, it determines the overall performance level of our system and will be of crucial interest to our study.

We define the measurement function $f(s) : \mathbb{R}^d \to \mathbb{C}$ as being the convolution of the point source $x(t)$ with the point spread function $e^{i\pi\langle s,t\rangle}$. Formally,

$$f(s) = \sum_{j \in [k]} w_j e^{i\pi\langle \mu^{(j)}, s\rangle}.$$

The function f is in essence a sum of weighted Fourier coefficients. Note that the measurements s in the noisy setting are corrupted with a uniformly bounded perturbation z so that the noisy recovery problem becomes

$$\widetilde{f}(s) = f(s) + z(s),$$

in which $|z(s)| \leq \epsilon_z$ for every s and a constant $\epsilon_z \in (0, 1/2)$. The idea is that given access to the signal $x(t)$, we wish to generate a set of random bandlimited Fourier measurements and evaluate the noisy function $\widetilde{f}$ on every measurement s in order to recover the parameters $\{w_j, \mu^{(j)} : j \in [k]\}$ of the point source signal as best as we can. The problem statement will be described in more detail in Chapter 4.

2.2 MAIN TOOLS

The main mathematical tools that are needed to understand the paper essentially lie around the subject of linear algebra. They include notions related to vectors, matrices and tensors. We also require some probabilistic analysis tools since the algorithm is partly random. In this chapter, we introduce those tools, prove the more important results and closely relate them to the paper.

2.2.1 GENERALISED EIGENVALUE PROBLEM

Before introducing the generalised eigenvalue problem, it is important to recall what a condition number is for a particular matrix. Suppose for instance we have a matrix $X \in \mathbb{R}^{m\times n}$. We let $\lambda_1, \dots, \lambda_n$ be the eigenvalues of $X^T X$ (with repetitions) and arrange them so that $\lambda_1 \geq \dots \geq \lambda_n \geq 0$. Then, the $\sigma_1 \geq \dots \geq \sigma_n \geq 0$ such that $\sigma_i = \sqrt{\lambda_i}$ are called the *singular values* of X. Define $\sigma_{max}(X) = \sigma_1$ and $\sigma_{min}(X) = \sigma_n$. We then define the condition number of a matrix to be the ratio between the largest and the smallest singular value of X. That is

$$\text{cond}_2(X) = \sigma_1/\sigma_n = \frac{\sigma_{max}(X)}{\sigma_{min}(X)}. \tag{2.1}$$

The above factor governs the noise tolerance of the generalised eigenvalue problem, i.e. it is the measure of sensitiveness of a matrix to arbitrary perturbations. Taking its limiting value will allow us to state whether or not the main algorithm achieves stable recovery of the point sources.

The goal of an eigenvalue problem is to simply find the eigenvalues of a particular matrix A. To do so, we generally solve the equation $AU = UV$, where U is an orthogonal matrix. Multiplying each side by U^T yields

$$AUU^T = UVU^T \Rightarrow A = UVU^T = UVU^{-1},$$

where $\mathbb{R}^{d\times d} \ni U := (u_1, \dots, u_d)$ are the eigenvectors and $\mathbb{R}^{d\times d} \ni V := Diag\left[(\lambda_1, \dots, \lambda_d)^T\right]$ the eigenvalues. However, in the generalised version of this problem, we add another random matrix B such that the problem becomes

$$\begin{aligned} AU = BUV &\Rightarrow AUU^T = BUVU^T \\ &\Rightarrow A = BUVU^T = BUVU^{-1}, \end{aligned}$$

where U again are the eigenvectors and V the eigenvalues. Note that A and B are both symmetric and that B is generated randomly with the only condition being that it is positive definite. We form the pair (A, B) called the *pencil* and the pair (U, V) called the *eigenpair*. Therefore, we introduce a first version of what is called the matrix pencil method, in which given a pair of matrices (A, B), we wish to find the generalised eigenvalues λ for which there is a vector $\mathbf{x}$ such that $A\mathbf{x} = \lambda B\mathbf{x}$. Notice that the eigenvalues of A are the solution to the generalised eigenvalue problem where $B = I$.

2.2.2 COMMON NORMS

VECTORS Norms are a powerful linear algebra tool that allows us to retreive important information regarding a particular matrix or vector. The most common vector norm is the L^p-norm defined for a vector X of size n as

$$||X||_p = \left(\sum_{i\in[n]} x_i^p\right)^{1/p}. \tag{2.2}$$

Note that p can take any value in $\mathbb{N}^*$ but the most common ones are 1, 2 or ∞. In essence, the L^1-norm is the sum of absolute values of our vector, the L^2-norm represents the Euclidean distance of the vector from the origin and the L^∞ norm is by definition the largest element in the vector, in absolute value.

MATRICES The same way we define a norm operator for vectors, we can define a similar operator for matrices, that will however have a slightly different meaning. Essentially, a matrix norm is a vector norm in a vector space whose elements are matrices. Let A be a matrix of size $m \times n$. We define the following important norms for A:

$$||A||_1 = \max_{j \in [n]} \sum_{i \in [m]} |A_{i,j}| \tag{2.3}$$

$$||A||_2 = \sqrt{\lambda_{max}(A^*A)} = \sigma_{max}(A) \tag{2.4}$$

$$||A||_\infty = \max_{i \in [m]} \sum_{j \in [n]} |A_{i,j}|, \tag{2.5}$$

where $\lambda_{max}(A^*A)$ is the largest eigenvalue of A^*A in which A^* denotes the conjugate transpose of A. Note that equality 2.4 is sometimes called the *spectral norm* of a matrix. We also consider another important matrix norm called the *Frobenius* norm, which is an equivalent version of the vector L^2-norm. It essentially represents the size of the matrix A, and we define it as

$$||A||_F = \sqrt{\sum_{i \in [m]} \sum_{j \in [n]} A_{i,j}^2}. \tag{2.6}$$

2.2.3 GERSHGORIN'S DISK THEOREM

A powerful result of linear algebra allows us to relate the eigenvalues of some matrix A to points in the complex plane. Namely, the Gershgorin's disk theorem gives us a pretty good and efficient way to find the eigenvalues of a particular matrix by drawing disks in the complex plane and relating them to the eigenvalues. For the purpose, let $Y \in \mathbb{C}^{k \times k}$ be a complex valued matrix. For every $j \in [k]$, let

$$\mathcal{D}_j = \left\{ x \in \mathbb{C} : ||x - Y_{j,j}|| \leq \sum_{j' \neq j} |Y_{j,j'}| \right\}$$

be the disk associated with the j'th row of Y, defined on the complex plane. Then, Gershgorin's disk theorem states that all the eigenvalues of Y must lie within the union of all the above defined disks. The intuition behind this theorem starts with the definition of an eigenvalue. Suppose for instance that λ is an eigenvalue of Y. Then we know we can write $Yv = \lambda v$, where v is an eigenvector of Y. Select j to be the index whose entry has largest magnitude in v. Writing the matrix-vector product as a sum yields $[Yv]_j = \sum_{j' \in [n]} Y_{j,j'} v_{j'} = \lambda v_j$. We can split this sum into two distinct sums as $\sum_{j' \in [n]} Y_{j,j'} v_{j'} = Y_{j,j} v_j + \sum_{j' \neq j} Y_{j,j'} v_{j'}$. Putting the factors of $v_{j'}$ on the same side gives us

$$\sum_{j' \neq j} Y_{j,j'} v_{j'} = (\lambda - Y_{j,j}) v_j.$$

We can divide by v_j on both sides and take the absolute value to obtain

$$\left| \sum_{j' \neq j} Y_{j,j'} v_{j'} / v_j \right| = \left| \lambda - Y_{j,j} \right|.$$

Using the triangle inequality for absolute values, we get

$$\begin{aligned} \left| \lambda - Y_{j,j} \right| &\leq \sum_{j' \neq j} \left| Y_{j,j'} v_{j'} / v_j \right| \\ &\leq \sum_{j' \neq j} \left| Y_{j,j'} \right| \left| v_{j'} / v_j \right| \text{ since } |ab| = |a||b| \text{ for any } a, b \\ &\leq \sum_{j' \neq j} \left| Y_{j,j'} \right|, \end{aligned}$$

where in the last line we used the fact that v_j has the largest magnitude in v such that $v_j \geq v_{j'}$ for every j. Hence, the Gershgorin disk centered at some diagonal j has radius always at most the sum of all non-diagonal entries of Y of the same row j.

2.2.4 PROBABILISTIC INEQUALITIES

Here we present the main probability bounds that are used to prove the correctness of the main algorithm. The first lemma is a generalisation of the Hoeffding bound to random matrices. It tells us that with high probability, the values of a random Hermitian matrix concentrate around the mean. The second considers the projection of a high dimensional random vector onto a lower dimension plane. Namely, suppose that we consider a random vector that lies on the m-dimensional unit sphere and another fixed vector. Then, the lemma states that with high probability, the absolute value of their dot product is large. We now state the aforementionned lemmas:

Lemma 1 *(Matrix Hoeffding) Let $\{X^{(1)}, \ldots, X^{(m)}\}$ be a set of independent, random, Hermitian matrices of dimension $k \times k$, with identical distribution X. Assume that $E[X]$ is finite, and $X^2 \preceq \sigma^2 I$ for some positive constant σ almost surely. Then, for all $\epsilon \geq 0$,*

$$Pr\left(\left|\left|\frac{1}{m}\sum_{i=1}^{m} X^{(i)} - E[X]\right|\right|_2 \geq \epsilon\right) \leq ke^{-\frac{m^2\epsilon^2}{8\sigma^2}}.$$

Lemma 2 *(Vector random projection) Let $a \in \mathbb{R}^m$ be a random vector distributed uniformly over the m-dimensional unit sphere and fix $v \in \mathbb{C}^m$. Then, for $\delta \in (0, 1)$,*

$$Pr\left(|\langle a, v\rangle| \leq \frac{||v||_2}{\sqrt{em}}\delta\right) \leq \delta.$$

Proof (for vector random projection) We give an intuitive proof of the above lemma, which is mostly based on lemma 2.2 from [1]. Let $X_1, \ldots, X_d$ be independent samples from a standard Gaussian distribution and let $Y = (X_1, \ldots, X_d)/||X||$ be a vector formed by the normalised Gaussians. It is straightforward to see that Y is uniformly distributed on the d-dimensional sphere. Let now $Z \in \mathbb{R}^k$ be the projection of Y onto the first k coordinates and let $L = ||Z||^2$. Observe that $E[L] = d/k$ for $d < k$. Then, it is true that

$$Pr\big(L \leq \beta k/d\big) \leq \beta e^{\frac{k}{2}(1-\beta)}, \quad \beta < 1 \tag{2.7}$$

$$Pr\big(L \geq \beta k/d\big) \leq \beta e^{\frac{k}{2}(1-\beta)}, \quad \beta > 1 \tag{2.8}$$

Both bounds together tell us that the squared spectral norm of the projection is in fact fairly concentrated around its expectation. Now to apply the above to our lemma, set L to be $|\langle a, v\rangle|^2$ so that the bound becomes

$$Pr\left(|\langle a, v\rangle| \leq \frac{\sqrt{\beta}||v||_2}{\sqrt{m}}\right) \leq \beta e^{\frac{k}{2}(1-\beta)}.$$

Setting $\beta = \delta^2/e$ yields

$$Pr\left(|\langle a, v\rangle| \leq \frac{||v||_2}{\sqrt{em}}\delta\right) \leq \frac{\delta^2}{e}e^{\frac{k}{2}(1-\delta^2/e)}.$$

Hence to prove the bound, it suffices to show that $\frac{\delta^2}{e}e^{\frac{k}{2}(1-\delta^2/e)} \leq \delta$ or equivalently $\delta e^{\frac{k}{2}(1-\delta^2/e)} \leq e$. Since by assumption $\delta < 1$ and $1 < e^{k(1+\delta^2/e)/2}$, the proof is completed. ■

2.3 TENSOR DECOMPOSITION

A tensor is a generalisation of a matrix to more than two dimensions. We can think of a tensor as a point in $\mathbb{C}^{m_1 \times \ldots \times m_p}$ where p is the order of the tensor. Most of the time here, $p = 3$ since three dimensions suffice for our analysis. Note that if T is an order three tensor of dimensions $m \times n \times p$, we can view it as a collection of p matrices of size $m \times n$ stacked on top of each other. For example, the entry $A_{i,j,k}$ of a 3-tensor A will simply be the (i, j)'th entry of the k'th matrix [2]. Furthermore, we define the *rank* of a tensor T as the minimum k such that we can write T as the sum of rank one tensors.

2.3.1 CANONICAL POLYADIC DECOMPOSITION

Canonical Polyadic (CP) decomposition (or *tensor rank decomposition*) is a procedure for tensors equivalent to that of the SVD for matrices in which we wish to express a p-tensor as an outer product of p vectors. We first recall that the outer product of two vectors $u \in \mathbb{C}^m$ and $v \in \mathbb{C}^n$ gives a matrix $M \in \mathbb{C}^{m \times n}$ such that $[M]_{r,c} = u_r v_c$ for $r \in [m]$ and $c \in [n]$. From there, suppose that we are working with a rank k 3-tensor $V \in \mathbb{C}^{m \times n \times p}$. CP decomposition allows us to express V as

$$V = \sum_{n \in [k]} [A]_{:,n} \circ [B]_{:,n} \circ [C]_{:,n}, \tag{2.9}$$

where $\circ$ denotes the outer product, $A \in \mathbb{C}^{m \times k}$, $B \in \mathbb{C}^{n \times k}$, $C \in \mathbb{C}^{p \times k}$ and where $[M]_{:,n}$ denotes the vector formed by the n'th column of M. We then say that the tensor V admits the decomposition $V = A \otimes B \otimes C$, where $\otimes$ denotes the tensor product operation. Equivalently, element-wise we have

$$V_{j_1,j_2,j_3} = \sum_{n \in [k]} A_{j_1,n} B_{j_2,n} C_{j_3,n}, \tag{2.10}$$

for $j_1 \in [m], j_2 \in [n], j_3 \in [p]$.

2.3.2 TENSOR DECOMPOSITION AS A MULTILINEAR MAPPING

We can use an alternative definition of tensor decomposition using the notion of a multi-linear mapping. Namely, for given dimensions m_A, m_B, m_C, the mapping $V(\cdot, \cdot, \cdot) : \mathbb{C}^{m \times m_A} \times \mathbb{C}^{m \times m_B} \times \mathbb{C}^{m \times m_C} \to \mathbb{C}^{m_A \times m_B \times m_C}$ is defined as:

$$[V(X_A, X_B, X_C)]_{i_1,i_2,i_3} = \sum_{j_1,j_2,j_3 \in [m]} V_{j_1,j_2,j_3} [X_A]_{j_1,i_1} [X_B]_{j_2,i_2} [X_C]_{j_3,i_3}. \tag{2.11}$$

We can verify that for a particular vector $a \in \mathbb{C}^m$, the projection $V(I, I, a)$ of V along the 3rd dimension is $V(I, I, a) = ADiag(C^T a)B^T$ as long as V admits a tensor decomposition $V = A \otimes B \otimes C$. Indeed,

$$\begin{aligned}
[V(I, I, a)]_{i_1,i_2} &= \sum_{j_1,j_2,j_3 \in [m]} V_{j_1,j_2,j_3} [I]_{j_1,i_1} [I]_{j_2,i_2} [a]_{j_3} \\
&= \sum_{j_1,j_2,j_3 \in [m]} \sum_{n \in [k]} A_{j_1,n} B_{j_2,n} C_{j_3,n} [I]_{j_1,i_1} [I]_{j_2,i_2} [a]_{j_3} \\
&= \sum_{j_3 \in [m]} \sum_{n \in [k]} A_{i_1,n} B_{i_2,n} C_{j_3,n} [a]_{j_3} \\
&= \underbrace{\sum_{n \in [k]} A_{i_1,n} B_{i_2,n}}_{=AB} \underbrace{\sum_{j_3 \in [m]} C_{j_3,n} [a]_{j_3}}_{=Diag(C^T a)} \\
&= \left[ADiag(C^T a)B^T \right]_{i_1,i_2}.
\end{aligned}$$

Since the above is true for every i_1, i_2, and we do not consider the i_3 coordinate since a is a vector, we get the desired equality.

In a more general case, we can consider the projection of the tensor V on three m column matrices X, Y and Z. Again we suppose that V admits the decomposition $A \otimes B \otimes C$. In that case, it is true that

$$\begin{aligned}
[V(X,Y,Z)]_{i_1,i_2,i_3} &= \sum_{j_1,j_2,j_3 \in [m]} V_{j_1,j_2,j_3} [X]_{j_1,i_1} [Y]_{j_2,i_2} [Z]_{j_3,i_3} \\
&= \sum_{j_1,j_2,j_3 \in [m]} \sum_{n \in [k]} A_{j_1,n} B_{j_2,n} C_{j_3,n} [X]_{j_1,i_1} [Y]_{j_2,i_2} [Z]_{j_3,i_3} \\
&= \sum_{j_1,j_2,j_3 \in [m]} \sum_{n \in [k]} [X^T]_{i_1,j_1} A_{j_1,n} [Y^T]_{i_2,j_2} B_{j_2,n} [Z^T]_{i_3,j_3} C_{j_3,n} \\
&= \sum_{n \in [k]} [X^T A]_{i_1,n} [Y^T B]_{i_2,n} [Z^T C]_{i_3,n} \\
&= [(X^T A) \otimes (Y^T B) \otimes (Z^T C)]_{i_1,i_2,i_3}.
\end{aligned}$$

Hence, we conclude that $V(X,Y,Z)$ admits the decomposition $(X^T A) \otimes (Y^T B) \otimes (Z^T C)$ for any arbitrary matrices X, Y and Z.

2.3.3 TENSOR FROBENIUS NORM

We can also extend the matrix norm definition to that of a tensor. However in our work, we will only need the equivalent Frobenius norm which will be useful in the main algorithm. For the purpose, let V be a rank k p-tensor of dimensions $m \times n \times p$. Then, the Frobenius norm of V is simply the root of the sum of all squares, i.e.

$$||V||_F = \sqrt{\sum_{i \in [m]} \sum_{j \in [n]} \sum_{l \in [p]} [V]^2_{i,j,l}}. \tag{2.12}$$

2.3.4 JENNRICH'S ALGORITHM FOR TENSOR DECOMPOSITION

As previously stated, the goal of tensor decomposition is to, given a tensor T of rank k, decompose it as a sum of rank 1 tensors of appropriate dimensions. Jennrich's algorithm is commonly used for tensor decomposition. We denote the following theorem:

Theorem 3 *Let $T = \sum_{j=1}^{k} u_i \otimes v_i \otimes w_i$ be a tensor in which each set of vectors $\{u_i\}_i$ and $\{v_i\}_i$ are linearly independent. Moreover, each pair of vectors in $\{w_i\}_i$ are also linearly independent. Then, the above decomposition is unique up to rescaling, and there is an efficient algorithm to find it.*

The aforementioned algorithm is described in Algorithm 1.

Algorithm 1 Jennrich's algorithm for tensor decomposition

Input: a tensor $\widetilde{F} \in \mathbb{C}^{m \times m \times 3}$ of rank k.
Choose random unit vectors $a, b \in \mathbb{R}^m$.
Compute $\widetilde{F}(I, I, a) = U D_a V^T$, where $D_a = Diag(\langle w^{(i)}, a \rangle)$.
Compute $\widetilde{F}(I, I, b) = U D_b V^T$, where $D_b = Diag(\langle w^{(i)}, b \rangle)$.
Compute the diagonalisations $\widetilde{F}(I, I, a)\widetilde{F}(I, I, b)^{-1}$ and $\widetilde{F}(I, I, b)\widetilde{F}(I, I, a)^{-1}$.
Solve the linear system to recover the w_j's.
Return U, V, W.

Observe that

$$\widetilde{F}(I,I,a)\widetilde{F}(I,I,b)^{-1} = UD_aV^T(V^T)^{-1}D_b^{-1}U^{-1} = UD_aD_b^{-1}U^{-1},$$

and similarly

$$\widetilde{F}(I,I,a)^{-1}\widetilde{F}(I,I,b) = (V^T)^{-1}D_a^{-1}U^{-1}UD_bV^T = (V^T)^{-1}D_a^{-1}D_bV^T.$$

The correctness of the algorithm follows from the uniqueness of eigendecomposition of a matrix when the eigenvalues are distinct. For a random choice of a and b (in our case we choose the basis vectors e_1 and e_2), with high probability the eigenvalues are unique so we can recover the u_i's and the v_i's easily by simply recovering the columns of U and V, respectively [3].

CHAPTER 3

INITIAL WORK

3.1 PRONY'S METHOD

In 1795, the french mathematician Gaspard de Prony came up with a very useful trick to solve a recovery problem, which aims to reconstruct functions from their values at given points. Essentially, we consider those functions as n'th degree polynomials from which we want to recover the roots. This is today called *Prony's method* and can easily be applied to the theory of super-resolution [4].

GENERAL FORM In a general sense, consider functions of the form $z = f(x) = \sum_{j\in[n]} \mu_j \rho_j^x$, where $\mu_j \in \mathbb{R}$ and $\rho_j \in \mathbb{R}_+$ for every $1 \leq i, j \leq n$. We want to recover those parameters from the samples $z_k := f(x_k)$ of f. We are faced with what is called Prony's condition, which states that given an undetermined number of measurements or sampling points $z_0, z_1, \ldots$, we impose that

$$(z * p)_k = \sum_{j=0}^{n} z_{k+1} p_j = 0,$$

for $k = 0, 1, 2, \ldots$ and we wish to solve the above equations for the p_j's. We see that the solutions are given by $p(x) = (x - \widehat{\rho}_1) \cdots (x - \widehat{\rho}_n)$. In a more algorithmic sense, we seek to

1. solve the above linear system for p and
2. find the zeroes of $p(x)$ to obtain the ρ_j's.

LINK WITH SUPER-RESOLUTION Prony's method can in fact be applied to many domains. For example in microscopy, images often contain brightness spikes on the plane but the optical system that contains all the hardware to create the picture essentially acts as a low pass filter: it will keep the low frequencies of a signal and attenuate the frequencies that are higher than a pre-determined cutoff frequency. By doing so, those spikes are turned into what are called *point spread functions*, that determine the performance of an imaging system. Similarly to our problem, the goal of Prony's method is to reconstruct the image by recovering estimates of the original spikes.

We can view the plane of an image as part of $\mathbb{R}^2$. Let its spikes be at positions $X \subset \mathbb{R}^2$ and let them have amplitude a_x. Note that X is sparse. Taking the Fourier transform yields

$$z_\alpha := z_{\alpha_1,\alpha_2} = \sum_{x\in X} a_x e^{\alpha_1 x_1 + \alpha_2 x_2},$$

where $\alpha = (\alpha_1, \alpha_2) \in \mathbb{Z}^2$. Through the low pass filter, we obtain a finite amount of measurements

$$z_\alpha = \sum_{x \in X} a_x \rho_x^\alpha,$$

for $|\alpha_1|, |\alpha_2| \leq n$ where $\rho_x^\alpha = \rho_{x,1}^{\alpha_1} \rho_{x,2}^{\alpha_2}$ and $\rho_{x,j} = e^{ix_j}$ for $j = 1, 2$. Note that the recovery of the z_α's is exactly the 2D version of Prony's problem: the a's are our weights and the ρ_x^α's are our point sources. It is also important to note that **the spatial resolution of an imaging device may be measured by how closely lines can be resolved**. In other words, the closer we can correctly resolve the point sources, the better the resolution will be.

The following sections are dedicated to helper methods that help us better understand both the tensor decomposition and the main procedure.

3.2 UNIVARIATE CASE: MATRIX-PENCIL METHOD

As an extension of Prony's method which is not stable, we consider the main algorithm in the univariate case, the setting of which is as follows. We construct two $m \times m$ complex valued Hankel matrices H_0 and H_1, that is, matrices such that their skew-diagonals[1] are constants. We have $D_w \in \mathbb{C}^{k \times k}$ where $[D_w]_{j,j} = w_j$ and $D_\mu \in \mathbb{C}^{k \times k}$ where $[D_\mu]_{j,j} = e^{i\pi\mu^{(j)}}$. We furthermore have a Vandermonde matrix $V_m \in \mathbb{C}^{m \times k}$ defined such that $[V_m]_{r,c} = \left(e^{i\pi\mu^{(c)}}\right)^r$ for $r = 0, \ldots, m-1$ and $c \in [k]$. We construct a 3rd order tensor $F \in \mathbb{C}^{m \times m \times 2}$ in which $F_{i,i',j} = [H_{j-1}]_{i,i'}$ for $j = 1, 2$ and $i, i' \in [m]$. We denote the following fact:

Fact 4 *In the setting above, F admits the unique rank k tensor decomposition $F = V_m \otimes V_m \otimes (V_2 D_w)$.*

Proof We start by computing the product $V_2 D_w$. By definition of V_m, we have that $[V_2]_{r,c} = \left(e^{i\pi\mu^{(c)}}\right)^{r-1}$ for $r \in [2]$ and $c \in [k]$. Multiplying with the diagonal matrix D_w yields

$$[V_2 D_w]_{r,c} = w_c \left(e^{i\pi\mu^{(c)}}\right)^{r-1}, \quad \forall r \in [2], c \in [k].$$

We aim to show that

$$\begin{aligned} F_{i_1,i_2,i_3} &= \sum_{n=1}^{k} [V_m]_{i_1,n} [V_m]_{i_2,n} [V_2 D_w]_{i_3,n} \\ &= \sum_{n=1}^{k} \left(e^{i\pi\mu^{(n)}}\right)^{i_1-1} \left(e^{i\pi\mu^{(n)}}\right)^{i_2-1} w_n \left(e^{i\pi\mu^{(n)}}\right)^{i_3-1} \\ &= \sum_{n=1}^{k} w_n \left(e^{i\pi\mu^{(n)}}\right)^{i_1+i_2+i_3-3}, \end{aligned}$$

for $i_1, i_2 \in [m]$ and $i_3 \in [2]$. It can be verified that the measurements that form both Hankel matrices are given by

$$f(s) = \sum_{j \in [k]} w_j \left(e^{i\pi\mu^{(j)}}\right)^s,$$

with $s = 0, \ldots, 2m-1$ for H_0 and $s = 1, \ldots, 2m$ for H_1. For the details of the proof, see Appendix A. Then, since $F_{i_1,i_2,i_3} = [H_{i_3-1}]_{i_1,i_2}$, for a fixed i_3 we have either $F_{i_1,i_2,1} = [H_0]_{i_1,i_2}$ or $F_{i_1,i_2,2} = [H_1]_{i_1,i_2}$. Note that since we are dealing with Hankel matrices, $[H_0]_{i_1,i_2} = f(i_1 + i_2 - 2)$. Hence,

$$F_{i_1,i_2,i_3} = \sum_{n \in [k]} w_n \left(e^{i\pi\mu^{(n)}}\right)^{i_1+i_2+i_3-3} = f(i_1 + i_2 + i_3 - 3),$$

[1] A skew-diagonal is the diagonal in the North-East direction.

as required. ■

Recall that Jennrich's algorithm computes the simultaneous diagonalisations of $F(I, I, v_1)$ and $F(I, I, v_2)$, in which v_1 and v_2 are two random projections of $\mathbb{R}^m$. Note that setting $v_1 = e_1$ and $v_2 = e_2$ yields H_0 and H_1 exactly. Hence, we conclude that the matrix-pencil method studied here is just a special case of Jennrich's algorithm where both projections are done on the two basis vectors e_1 and e_2 [5].

3.3 MULTIVARIATE TOY CASE

This section generalises the previous one to the multi-dimensional case and thus describes a preview of the main algorithm when the measurements are not random and where we assume for simplicity that the point sources are uniformly distributed in $[-1, +1]$. We follow the definitions given in the paper. We have $D_w = Diag(w_j) \in \mathbb{C}^{k\times k}$ and $[V_d]_{r,c} = e^{i\pi\mu_r^{(c)}} \in \mathbb{C}^{d\times k}$ the latter in which $\mu_n^{(i)}$ are i.i.d. $\sim Unif([-1, +1])$. Furthermore, we construct a tensor F such that $F_{n_1,n_2,n_3} = f(s)\big|_{s=e_{n_1}+e_{n_2}+e_{n_3}}$, for all $n_1, n_2, n_3 \in [d]$. Hence, the measurements are taken from the grid $[d]^3$ and stored in F. The following fact demonstrates the correctness of the procedure.

Fact 5 *In the setting above, F admits the tensor decomposition $F = V_d \otimes V_d \otimes (V_d D_w)$.*

Proof We wish to show that $f(s)\big|_{s=s^{(n_1)}+s^{(n_2)}+s^{(n_1)}} = \sum_{j=1}^{k} w_j e^{i\pi(\mu_{n_1}^{(j)}+\mu_{n_2}^{(j)}+\mu_{n_3}^{(j)})}$. To do so, we first compute the matrix product $V_d D_w$. Since $[V_d]_{r,c} = e^{i\pi\mu_r^{(c)}}$ for $r \in [d], c \in [k]$ and $D_w = Diag(w_j)$, we directly have that

$$[V_d D_w]_{r,c} = w_c e^{i\pi\mu_r^{(c)}}, \quad r \in [d], c \in [k],$$

so that by definition of tensor decomposition, we get

$$\begin{aligned}
F_{n_1,n_2,n_3} &= \sum_{j=1}^{k} [V_d]_{n_1,j}[V_d]_{n_2,j}[V_d D_w]_{n_3,j} \\
&= \sum_{j=1}^{k} e^{i\pi\mu_{n_1}^{(j)}} e^{i\pi\mu_{n_2}^{(j)}} w_j e^{i\pi\mu_{n_3}^{(j)}} \\
&= \sum_{j=1}^{k} w_j e^{i\pi(\mu_{n_1}^{(j)}+\mu_{n_2}^{(j)}+\mu_{n_3}^{(j)})} \\
&= f(s)\big|_{s=s^{(n_1)}+s^{(n_2)}+s^{(n_1)}} = f(e_{n_1} + e_{n_2} + e_{n_3}),
\end{aligned}$$

as required. ■

The key idea here is to notice that since the point sources are uniform, the entries $e^{i\pi\mu_n^{(j)}}$ are i.i.d. and uniformly distributed over the unit circle of the complex plane. This implies that the factor V_d almost surely has full column rank and therefore that the tensor decomposition exists and is unique (since each vector contained in the tensor's matrices are linearly independent). Again by uniformity assumption, the condition number of V_d concentrates around 1 so the above procedure achieves stable recovery.

Despite the above procedure yielding the desired decomposition, the problem is that taking measurements from the hypergrid worsen the complexity. We thus wish to find a way to generate less measurements in a better way so that we can recover good enough estimates of the point sources. The main procedure to do so is described in the next chapter.

CHAPTER 4

MAIN ALGORITHM

Let us briefly recall what the goal of the problem is. Broadly speaking, super-resolution aims to recover a superposition of point sources using bandlimited measurements that may be corrupted with noise. In this setting, the created algorithm works in the Fourier domain where each of the k points of the d-dimensional plane are separated by a distance at least Δ. Hence, the frequencies of the Fourier measurements are bounded by $O(1/\Delta)$. The idea of the procedure is to generate a sample of random bandlimited measurements such that the estimates we recover of the point sources are precise enough.

This chapter discusses the main algorithm along with its analysis. We will go through all the important results and describe in detail their implication on the main procedure to effectively recover the point sources.

4.1 ALGORITHM DESCRIPTION

4.1.1 INTRODUCTION

The basic idea of the algorithm is to efficiently solve the problem of recovering the superposition of our point sources using *coarse* Fourier measurements. Once again, we are given a signal $x(t) = \sum_{j\in[k]} w_j \delta_{\mu^{(j)}}$ from which we want to recover the w_j and $\mu^{(j)}$ coefficients. We have access to a noisy measurement function $\widetilde{f}(\cdot)$. The idea is to generate random samples s and evaluate $\widetilde{f}$ for each of them in order to find the best fit, i.e. the best fitting $\mu^{(j)}$'s. More formally, for all s we compute

$$\widetilde{f}(s) = \sum_{j\in[k]} w_j e^{i\pi\langle \mu^{(j)}, s\rangle} + z(s),$$

where $z(s)$ is the noise upper bounded by ϵ_z for every s and store those results in a 3-way tensor. Compared to previous work, the algorithm guarantees a certain stability of recovery and runs in quadratic time. Namely, the measurements are bounded in frequency by $O(1/\Delta)$ and the algorithm runs in time $O((k\log k + d)^2)$. The key recovery guarantee relies on the condition number of the random Vandermonde matrix. This is however further discussed in the next sections.

4.1.2 DEFINITIONS

The following objects are the ones that play an essential role in the main algorithm:

- $[V_S]_{r,c} = e^{i\pi\langle \mu^{(c)}, s^{(r)}\rangle}$ is the *characteristic matrix*, that is, the matrix that contains all the point spread functions generated by the random Gaussian measurements.

- $[V_d]_{r,c} = e^{i\pi\mu_r^{(c)}}$ is the matrix that contains all the complex valued coefficients that lie on the unit circle of the complex plane. Note that this is again, due to the uniformity of the μ's.
- $V_{S'} = \left[V_S,\, V_d,\, [1]^k\right]^T$ is the matrix that will be returned from the tensor decomposition procedure.
- Finally, define V_2 such that $[V_2]_{r,c} = e^{i\pi\langle\mu^{(c)},v^{(r)}\rangle}$ where we let $v^{(3)} = 0$ for simplicity of notation. Note that the exponent being 0 will yield a last row of only ones, useful for normalisation purposes.

4.1.3 MAIN RESULT

INPUT AND OUTPUT The algorithm takes as input a cutoff frequency R, the defined number of measurements m and a noisy measurement function $\widetilde{f}(\cdot)$ and outputs the set of estimates

$$\{\widehat{w}_j, \widehat{\mu}^{(j)} : j \in [k]\},$$

where the $\widehat{w}_j$'s are the complex weight coefficients and the $\widehat{\mu}^{(j)}$'s are the estimates of the point sources. Note that in case the noise is non-existant (i.e. when $\epsilon_z = 0$), the parameters are recovered exactly. Otherwise, stable recovery implies that the estimates over all permutations π on $[k]$ are such that

$$\min_{\pi} \max\left\{||\widehat{\mu}^{(j)} - \mu^{(\pi(j))}||_2 : j \in [k]\right\} \leq poly(d,k)\epsilon_z.$$

In words, in the L^2 sense, the estimates of the point sources differ from their real values by at most the noise ϵ_z scaled by a polynomial function that depends on the number of dimensions d and the number of point sources k. Interestingly, the stability of recovery does not depend on the minimal distance between the point sources, but rather focuses on how they are concentrated on the plane.

MEASUREMENTS We generate a set of measurements $\mathcal{S} = \{s^{(1)}, \ldots, s^{(m+d+1)}\}$ randomly in which

1. the $s^{(1)}, \ldots, s^{(m)}$ are m i.i.d. samples from the Gaussian distribution $\mathcal{N}(0, R^2 I_{d\times d})$,
2. $s^{(m+n)} = e_n$ for all $n \in [d]$ and
3. $s^{(m+d+1)} = 0$.

We also let $m' = m + d + 1$ for writing simplicity. Note that each measurement is a d-dimensional vector, i.e. a point in the d-dimensional plane. Therefore, the Gaussian measurements are such that $[s^{(i)}]_n$ are non-correlated 0-mean R^2-variance gaussians for every $n \in [d]$. We then take a sample v from the unit sphere and set $v^{(1)} = v$ and $v^{(2)} = 2v$. Since again we are in dimension d, the sample v is such that $v_1^2 + \ldots + v_d^2 = 1$.

TENSOR DECOMPOSITION We now construct a tensor

$$\widetilde{F} \in \mathbb{C}^{m'\times m'\times 3} : \widetilde{F}_{n_1,n_2,n_3} = \widetilde{f}(s)|_{s=s^{(n_1)}+s^{(n_2)}+v^{(n_3)}}$$

containing all the possible combinations of measurements from $\mathcal{S} \oplus \mathcal{S} \oplus \{v^{(1)}, v^{(2)}, 0\}$, where $\oplus$ is the set addition operator. Note that there are $O((m')^2)$ such measurements. We apply an adaptation of Jennrich's algorithm on $\widetilde{F}$ (see Algorithm 2) to obtain the estimates $\widehat{V}_{S'}$ and $\widehat{D}_w$. Note that the estimate $\widehat{V}_{S'}$ is normalised so that its last line is all ones, in order to simplify the decomposition.

READ OF ESTIMATES We finish by recovering the real part of the estimates of the point sources by setting $\widehat{\mu}^{(j)} = Real(\log([\widehat{V}_{S'}]_{[m+1:m+d,j]})/(i\pi))$ for every $j \in [k]$ and find the best possible matching weight coefficients by setting $\widehat{W} = \arg\min_{W\in\mathbb{C}^k} ||\widehat{F} - \widehat{V}_{S'} \otimes \widehat{V}_{S'} \otimes \widehat{V}_d D_w||_F$, where $||\cdot||_F$ is the Frobenius norm of $(\cdot)$. Here, observe that by definition $[\widehat{V}_{S'}]_{[m+1:m+d,j]} = [\widehat{V}_d]_{[1:d,j]}$, where $\log(\widehat{V}_d)$ is performed element-wise in $\widehat{V}_d$ to recover the list of all estimates.

Algorithm 2 Adaptation of Jennrich's algorithm

Input: a tensor $\widetilde{F} \in \mathbb{C}^{m \times m \times 3}$ of rank k.
Project $\widetilde{F}$ along the direction of the first basis vector.
Compute the truncated SVD of the above projection that yields $\widetilde{F}(I, I, e_1) = \widehat{P}\widehat{\Lambda}\widehat{P}^T$.
Set $\widehat{E} = \widetilde{F}(P, P, I)$ and set $\widehat{E}_i = \widehat{E}(I, I, e_i)$ for $i = 1, 2$.
Let the columns of $\widehat{U}$ be the eigenvectors of $\widehat{E}_1\widehat{E}_2^{-1}$.
Return $\widehat{V} = \widehat{P}\widehat{U}$.

4.1.4 WHY WE USE TENSOR DECOMPOSITION

Jennrich's algorithm consists of a tensor decomposition that allows us to recover the estimates of the point sources. Namely, given a noisy tensor $\widetilde{F}$, it will spit out a decomposition that consists of three matrices, one of which we are most interested in. It is important to note that the crucical part in the adaptation of the original algorithm is the fact that we reduce the dimensionality of the original tensor in order to compute the decomposition faster. Indeed, the truncated SVD operation will allow us to recover the k largest singular values such that the newly constructed tensor E is of rank k and its projection on e_1 becomes invertible. We can then apply the original algorithm on E and spit out the desired decomposition.

We denote the following fact:

Fact 6 *In the exact case (i.e. when $\epsilon_z = 0$), the constructed tensor F admits a rank-k decomposition*

$$F = V_{S'} \otimes V_{S'} \otimes (V_2 D_w).$$

Proof As usual, we start by computing the product $V_2 D_w$. Since D_w is a diagonal matrix, we have

$$V_2 D_w = \begin{pmatrix} w_1 e^{i\pi\langle \mu^{(1)}, v^{(1)}\rangle} & \dots & w_k e^{i\pi\langle \mu^{(k)}, v^{(1)}\rangle} \\ w_1 e^{i\pi\langle \mu^{(1)}, v^{(2)}\rangle} & \dots & w_k e^{i\pi\langle \mu^{(k)}, v^{(2)}\rangle} \\ w_1 & \dots & w_k \end{pmatrix}.$$

Notice that once again, we can think of multiplying the last row of w's with an exponential whose exponent is 0. Then, write $[V_d]_{r,c} = e^{i\pi\mu_r^{(c)}} = e^{i\pi\langle \mu^{(c)}, e_r = s(m+r)\rangle}$. Using this, we can express the r'th row and the c'th column of $V_{S'}$ as

$$[V_{S'}]_{r,c} = e^{i\pi\langle \mu^{(c)}, s^{(r)}\rangle}, \quad r \in [m'], c \in [k].$$

From there for $n_1, n_2 \in [m']$ and $n_3 \in [3]$, it follows easily that

$$\begin{aligned} F_{n_1,n_2,n_3} &= \sum_{n\in[k]} [V_{S'}]_{n_1,n}[V_{S'}]_{n_2,n}[V_2 D_w]_{n_3,n} \\ &= \sum_{n\in[k]} e^{i\pi\langle \mu^{(n)}, s^{(n_1)}\rangle} e^{i\pi\langle \mu^{(n)}, s^{(n_2)}\rangle} w_n e^{i\pi\langle \mu^{(n)}, v^{(n_3)}\rangle} \\ &= \sum_{n\in[k]} w_n e^{i\pi\left(\langle \mu^{(n)}, s^{(n_1)}\rangle + \langle \mu^{(n)}, s^{(n_2)}\rangle + \langle \mu^{(n)}, v^{(n_3)}\rangle\right)} \\ &= \sum_{n\in[k]} w_n e^{i\pi\langle \mu^{(n)}, s^{(n_1)} + s^{(n_2)} + v^{(n_3)}\rangle} \\ &= \widetilde{f}(s)\big|_{s = s^{(n_1)} + s^{(n_2)} + v^{(n_3)}}, \end{aligned}$$

as required. ∎

The above fact demonstrates why we use tensors. Indeed, having access to $\widehat{V}_{S'}$ allows us to recover the estimates of the point sources. The next sections are dedicated to the multiple stability guarantees that makes this algorithm both useful and powerful.

4.2 STABILITY GUARANTEES

The main guarantee on the stability of the recovery of the point sources is given by

$$\min_{\pi} \max \left\{ ||\widehat{\mu}^{(j)} - \mu^{(\pi(j))}||_2 : j \in [k] \right\} \leq C \frac{\sqrt{d} k^5}{\Delta \delta_v} \frac{w_{max}}{w_{min}^2} \left(\frac{1 + 2\epsilon_x}{1 - 2\epsilon_x} \right)^{5/2} \epsilon_z,$$

where C is a universal constant such that the bound holds with probability at least $(1 - \delta_s)$ over the random sampling of $\mathcal{S}$ and with probability at least $(1 - \delta_v)$ over the random projections in Jennrich's algorithm. We will see in the following section that the stability guarantees of the algorithm essentially rely on two important results: the stability of Jennrich's algorithm for tensor decomposition, and the tight variation bound that comes from the choice of the measurements.

4.3 ANALYSIS

4.3.1 STABILITY OF TENSOR DECOMPOSITION

A main factor of stability resides in the tensor decomposition procedure. Since the measurements are linearly independent, the tensor decomposition exists and is unique up to column permutation and rescaling. The next few paragraphs go through steps that allow us to prove the stability guarantee that Jennrich's algorithm offers us.

SETUP IN THE NON-NOISY CASE Suppose that F admits the decomposition $F = V \otimes V \otimes (V_2 D_w) \in \mathbb{C}^{m \times m \times 3}$ and that $\widetilde{F}$ is element-wise close to F, that is, each of their elements differ by at most a constant ϵ_z in absolute value. If $\widetilde{F}$ is passed as input to Jennrich's algorithm, it happens that $\widehat{V}$ is close enough to V with high probability. To show this, denote $D_1 = diag([V_2]_{1,:} D_w)$ and $D_2 = diag([V_2]_{2,:} D_w)$ where $[V_2]_{i,:}$ is the vector formed by the i'th row of V_2. Let $F_1 = F(I, I, e_1)$. By previous computations, we easily find out that

$$\begin{aligned}
[F_1]_{n_1,n_2} = F(I, I, e_1)_{n_1,n_2} &= \sum_{j_1,j_2,j_3 \in [m]} F_{j_1,j_2,j_3} [I]_{j_1,n_1} [I]_{j_2,i_2} [e_1]_{j_3} \\
&= \sum_{n \in [k]} [V]_{n_1,n} [V]_{n_2,n} [V_2 D_w]_{1,n} \\
&= [V Diag([V_2]_{1,:} D_w) V^T]_{n_1,n_2},
\end{aligned}$$

which implies that the above projection can be decomposed as $F_1 = V D_1 V^T$. Now consider its SVD $F_1 = P \Lambda P^T$. Additionally, define the whitened[1] rank-k tensor $E = F(P, P, I)$. Observe that again, by previous computations, we can write E as

$$\begin{aligned}
E_{n_1,n_2,n_3} = F(P, P, I)_{n_1,n_2,n_3} &= \sum_{j_1,j_2 \in [m']} \sum_{j_3 \in [3]} F_{j_1,j_2,j_3} [P]_{j_1,n_1} [P]_{j_2,n_2} [I]_{j_3,n_3} \\
&= \sum_{j_1,j_2 \in [m']} \sum_{n \in [k]} [P^T]_{n_1,j_1} V_{j_1,n} [P^T]_{n_2,j_2} V_{j_2,n} [V_2 D_w]_{n_3,n} \\
&= \sum_{n \in [k]} [P^T V]_{n_1,n} [P^T V]_{n_2,n} [V_2 D_w]_{n_3,n},
\end{aligned}$$

so that $E = (P^T V) \otimes (P^T V) \otimes (V_2 D_w) = U \otimes U \otimes (V_2 D_w)$ when writing $U := P^T V \in \mathbb{C}^{k \times k}$.

[1] A whitening transformation has the objective of linearly transforming a vector of correlated random variables into a new vector, the elements of which are uncorrelated. Hence, the random elements that form the new tensor E will have zero correlation.

Now slice the tensor E in two such that we get its projections on the first two basis vectors of the plane. Formally, we have $E_1 = E(I, I, e_1)$ and $E_2 = E(I, I, e_2)$. $E_1 = UD_1U^T$ and $E_2 = UD_2U^T$. The proof for this is the same as that of F_1 and comes from the definition of linear mappings. Finally, computing $E_1E_2^{-1}$ yields the following eigendecomposition:

$$\begin{aligned} E_1E_2^{-1} &= UD_1U^T(U^TD_2U)^{-1} = UD_1U^T(U^T)^{-1}D_2^{-1}U^{-1} \\ &= UD_1D_2^{-1}U^{-1} = UDU^{-1}, \end{aligned}$$

by setting $D = D_1D_2^{-1}$. Observe that in the exact case,

$$D = diag(e^{i\pi\langle\mu^{(j)},v^{(1)}-v^{(2)}\rangle} : j \in [k]).$$

To show this, we first have to note that taking the inverse of a diagonal matrix only requires us to take the reciprocal of the diagonal elements and store them in a resulting matrix which is itself diagonal. Now recall that $[V_2D_w]_{r,c} = w_ce^{i\pi\langle\mu^{(c)},v^{(r)}\rangle}$. We can express D_1 and D_2 a little differently. Namely, write $[D_1]_{j,j} = w_je^{i\pi\langle\mu^{(j)},v^{(1)}\rangle}$ and $[D_2]_{j,j} = w_je^{i\pi\langle\mu^{(j)},v^{(2)}\rangle}$. Using the above fact allows us to express the inverse of D_2 using compact form as $[D_2^{-1}]_{j,j} = \frac{1}{w_j}e^{-i\pi\langle\mu^{(j)},v^{(2)}\rangle}$. Hence,

$$[D]_{j,j} = [D_1D_2^{-1}]_{j,j} = e^{i\pi\langle\mu^{(j)},v^{(1)}-v^{(2)}\rangle},$$

as required.

Finally write $M := E_1E_2^{-1}$. Note that since the columns of U are set to be the eigenvectors of M, by definition it suffices to compute the product PU to return our estimates since $PU = PP^TV = V$. Hence, the decomposition works as expected.

The above shows us why the dimensionality reduction is essential for our procedure. Indeed, tensor decomposition is in general a heavy operation, and it does not shy getting heavier as the tensor's dimensions grow. Here however, the decomposition is done on the smaller tensor E that, thanks to the truncated SVD, keeps most of the important information that F carries.

PROOF SKETCH OF STABILITY We can now start giving the general ideas of the proof that the above tensor decomposition procedure is stable.

1. **Noise affects the estimates mildly**: Jennrich's algorithm takes a large tensor as input and computes its decomposition. However, a key step in this decomposition is the creation of a lower dimension tensor that allows us to do the computations faster whilst keeping the stability guarantees. Here, matrix perturbation bounds show that the noise in the input tensor $\widetilde{F}$ propagates the estimates $\widehat{P}$ and $\widehat{E}$ in a mild way, as long as there is a tight enough upper bound on the condition number of V.
2. **Perturbation affects eigendecomposition mildly**: Suppose we write $\widehat{M} = \widehat{E}_1\widehat{E}_2^{-1}$ in terms of some perturbation matrices H and G defined as $H = -Z_2(I + E_2^{-1}Z_2)^{-1}E_2^{-1}$ and $G = Z_1E_2^{-1}$, where Z_i is the additive noise in the estimates $\widehat{E}_i$ for $i = 1, 2$. We can use *Gershgorin's disk theorem* to show that if the perturbations are negligible (i.e. $||H||$ and $||G||$ are small) and that the minimal separation of the diagonal entries in D is large enough, then the eigendecomposition of $\widehat{M}$ is close to that of M.
3. **Bounds on the minimal separation and the perturbations**: One can use anti-concentration bounds on the random projection to show that $sep(D)$ is large enough, and from there, use the fact that $||\widehat{U}_j - U_j||_2$ is small to deduce tight enough bounds on $||H||$ and $||G||$.
4. **The estimates $\widehat{V}$ are close enough to V**: We can finally use (3.) along with perturbation bounds again to prove that the estimates $\widehat{V}_i = \widehat{P}\widehat{U}_i$ are close enough to the actual values returned in the non-noisy case and hence conclude on the confirmed stability of tensor decomposition.

4.3.2 BOUNDS ON THE CONDITION NUMBERS

None of the above happens if the condition number of the characteristic matrix cannot be bounded. We describe here the general ideas that allow us to find a tight bound for $\mathrm{cond}_2(V_S)$.

1. **The condition numbers of V_S and $V_{S'}$ are close enough**: Using the definition of the condition number of a matrix, it can easily be shown that $\mathrm{cond}_2(V_{S'}) \leq \sqrt{1+\sqrt{k}}\,\mathrm{cond}_2(V_S)$.
2. **The eigenvalues of a random Gaussian Hermitian matrix are close to 1 in expectation**: Define $X_s \in \mathbb{C}^{k\times k}$ to be such that $[X_s]_{r,c} = e^{i\pi\langle \mu^{(c)}-\mu^{(r)},s\rangle}$. Then using the MGF of a Gaussian random variable, the assumption on the minimum distance of the point sources as well as geometric series, it can be shown that for any row of $Y := \mathbb{E}_s[X_s]$, the sum of all its non-diagonal entries is never greater than ϵ_x. From there, we can use Gershgorin's disk theorem to bound all the eigenvalues of Y by $1 \pm \epsilon_x$.
3. **The random sampling of $\mathcal{S}$ affects the condition number of V_S**: Taking two samples from a Gaussian distribution allow us to express $V_S^* V_S$ as $\frac{1}{m}\sum_m X^{(m)}$. Hence using the previous result, since each element of X_s lies on the unit circle in the complex plane and thus that $X_s^2 \preceq k^2 I$ almost surely, it suffices to use the Matrix Hoeffding lemma to show that for some large enough m, the sum of each Gaussian concentrate around the mean of X_s with high probability.

The combination of all above results allow us to write that

$$\mathrm{cond}_2(V_S) \leq \sqrt{\frac{1+2\epsilon_x}{1-2\epsilon_x}},$$

which is close to 1. Since this bound is tight enough, the recovery of the point sources is stable.

CHAPTER 5

DISCUSSION

5.1 IMPROVEMENTS ALREADY MADE

The algorithm proposed in the previous chapter allows us to efficiently recover the position of some image source points using only a polynomial amount of space. Recall that the problem instance has multiple base parameters:

1. the number of dimensions d needed to represent the entirety of the fine details of the image,
2. the number of point sources k we wish to recover,
3. the minimal distance $\Delta = \min_{j' \neq j} ||\widehat{\mu}^{(j')} - \widehat{\mu}^{(j)}||_2$ between any two pair of points j and j' on the plane, and
4. the number m of coarse measurements we have to make in order to efficiently recover the point sources in the noisy case.

The cutoff frequency R of the measurements is in this setting inversely proportional to the minimum distance Δ, that is, $R = O(1/\Delta)$. Compared to previous work which assumed that R could be as large as $\Omega(\sqrt{d}/\Delta)$, data can now be super-resolved well enough with a pretty small cutoff frequency and hence induces a robust enough procedure. Moreover, the number of measurements needed for the procedure to work in a stable enough way do not depend on the distance between the points anymore, but only depend on the number of points and the overall dimension. Indeed previous results required an exponential number of measurements to be taken on the hypergrid, whereas the current procedure's computational complexity is bounded by a polynomial in both d and k. This hence results in an exponential improvement.

5.2 ROBUSTNESS OF A RANDOM VANDERMONDE MATRIX

To complement what has been stated in previous sections, we reiterate on the fact that stability guarantees of the procedure depend on one major factor: the condition number $\mathrm{cond}_2(V_S)$ of the random Vandermonde matrix that contains our measurements. Since those are stored in a tensor, we seek the stability of the tensor decomposition procedure. For it to be stable, we need $\mathrm{cond}_2(V_S)$ to be close to 1 in expectation; this happens if the measurements are sampled from a Gaussian distribution. Again, since a single measurement is made from the convolution of two Gaussians, it allows us to express the product $V_S^* V_S$ as the sample mean of random Hermitian matrices the spectrum of which is close enough to that of their average in expectation. From there, noting that each measurement lies on the unit circle of the complex plane, a Matrix-Hoeffding bound suffices to conclude on a close bound for $\mathrm{cond}_2(V_S)$ that holds with high probability.

Since our measurements are stored in a Vandermonde matrix, it is important to observe the behaviour of their condition number. Recent studies have shown that such matrices tend to be badly ill-conditioned in most scenarios, with the exception of some specific cases such as DFT matrices or as it turns out, complex Gaussian valued matrices [6]. The bounds in question get better when the randomly generated points that form the Vandermonde matrix lie closer to the unit circle of the complex plane. In our case, the measurements indeed lie on the complex circle and thus provide a strong stability guarantee of recovery.

Recall that the condition number of a matrix governs its tolerance to noisy data. Specifically, it measures how sensitive a particular matrix is to perturbations that can appear in the input data. Although an ill-conditioned matrix augments the sensitivity of its inverse, the eigenvalue problem can stay well conditioned. In other words, it can happen that a matrix is poorly conditioned for inversion whilst the eigenvalue problem is well conditioned, or vice versa. Hence, keeping an eye on the condition number of the measurement matrix is a key element to not neglect whilst wanting to improve the sample complexity of the procedure.

5.3 IMPROVEMENTS LEFT TO BE MADE

The open problem related to this algorithm is to reduce the sample complexity from quadratic to linear. Namely, we seek a reduction of the number of measurements made, from $O(2(m')^2)$ to $O(m')$. Recall that $m' = m + d + 1$ corresponds to the first two dimensions of our measurement tensor, m of which are Gaussian, the others respectively being the d basis vectors needed to recover the location of the point sources and a row of ones used for normalisation purposes. If we wish to take a linear amount of measurements, the procedure to generate them will have to change, to wit, we remove the convolution of the two Gaussians.

CHAPTER 6

CONCLUSION

The goal of this project was to read, analyse and discuss a recent paper on the subject of super-resolution. The paper in question written in 2015 by Qingqing Huang and Sham Kakade highlights an algorithm that is capable to efficiently recover the location of some fine details of a low resolution image. It describes a single main algorithm that aims at creating a sampling set randomly which we then use to construct the measurements that we store in a tensor. A tensor decomposition procedure then aims at recovering our desired estimates. The overall runs in cubic time complexity and takes a quadratic amount of space.

Although this procedure is in itself a huge improvement compared to previous results, it is information-theoretically possible to reduce the amount of space used from quadratic to linear in the number of measurements. Future work could eventually imply taking measurements from a lower dimension and tensoring them in higher dimension or simply generating a single measurement per point source and storing them in a smaller tensor.

The project was an excellent opportunity for me to improve both my research and my personal skills. Indeed, doing a 12 credits project in the theoretical computer science lab at EPFL during a busy semester given that I don't really have the same background as a standard EPFL student is a huge challenge that I very much enjoyed taking part in. In the near future, I would definitely enjoy continuing this work with Prof. Kapralov to eventually find a real improvement that could boost both the sample complexity and the robustness of the main procedure.

BIBLIOGRAPHY

[1] Sanjoy Dasgupta and Anupam Gupta. 'An elementary proof of a theorem of Johnson and Lindenstrauss'. In: *Reseach Bell Labs* (2002).

[2] Ankur Moitra. *Lecture notes in Algorithmic aspects of Machine Learning*. 2015.

[3] Tim Roughgarden and Gregory Valiant. *Lecture notes in The modern algorithmic toolbox*. 2015.

[4] Tomas Sauer. 'Prony's method: an old trick for new problems'. In: *Snapshots of modern mathematics from Oberwolfach* (2018).

[5] Ankur Moitra. 'Super-resolution, Extremal Functions and the Condition Number of Vandermonde Matrices'. In: (2015). arXiv: 1408.1681 [cs.TT].

[6] Victor Y. Pan. 'How bad are Vandermonde matrices ?' In: *Department of Mathematics and Computer Science, Lehman College of the City University of New York* (2015).

APPENDIX A

CONSTRUCTION OF HANKEL MATRICES

In this section, we prove that the Hankel matrices H_0 and H_1 defined earlier admit the respective diagonalisations $V_m D_w V_m^T$ and $V_m D_w D_\mu V_m^T$. To do so, we first compute the product $D_w V_m^T$. We have

$$
\begin{aligned}
D_w V_m^T &= \begin{pmatrix} w_1 & 0 & \dots & 0 \\ 0 & w_2 & \dots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \dots & w_k \end{pmatrix} \begin{pmatrix} 1 & e^{i\pi\mu^{(1)}} & \dots & \left(e^{i\pi\mu^{(1)}}\right)^{m-1} \\ 1 & e^{i\pi\mu^{(2)}} & \dots & \left(e^{i\pi\mu^{(2)}}\right)^{m-1} \\ \vdots & \vdots & \ddots & \vdots \\ 1 & e^{i\pi\mu^{(k)}} & \dots & \left(e^{i\pi\mu^{(k)}}\right)^{m-1} \end{pmatrix} \\
&= \begin{pmatrix} w_1 & w_1 e^{i\pi\mu^{(1)}} & \dots & w_1\left(e^{i\pi\mu^{(1)}}\right)^{m-1} \\ w_2 & w_2 e^{i\pi\mu^{(2)}} & \dots & w_2\left(e^{i\pi\mu^{(2)}}\right)^{m-1} \\ \vdots & \vdots & \ddots & \vdots \\ w_k & w_k e^{i\pi\mu^{(k)}} & \dots & w_k\left(e^{i\pi\mu^{(k)}}\right)^{m-1} \end{pmatrix},
\end{aligned}
$$

so that

$$
\begin{aligned}
H_0 = V_m D_w V_m^T &= \begin{pmatrix} 1 & \dots & 1 \\ e^{i\pi\mu^{(1)}} & \dots & e^{i\pi\mu^{(k)}} \\ \vdots & \dots & \vdots \\ \left(e^{i\pi\mu^{(1)}}\right)^{m-1} & \dots & \left(e^{i\pi\mu^{(k)}}\right)^{m-1} \end{pmatrix} \begin{pmatrix} w_1 & w_1 e^{i\pi\mu^{(1)}} & \dots & w_1\left(e^{i\pi\mu^{(1)}}\right)^{m-1} \\ w_2 & w_2 e^{i\pi\mu^{(2)}} & \dots & w_2\left(e^{i\pi\mu^{(2)}}\right)^{m-1} \\ \vdots & \vdots & \ddots & \vdots \\ w_k & w_k e^{i\pi\mu^{(k)}} & \dots & w_k\left(e^{i\pi\mu^{(k)}}\right)^{m-1} \end{pmatrix} \\
&= \begin{pmatrix} \sum_{j\in[k]} w_j & \sum_{j\in[k]} w_j e^{i\pi\mu^{(1)}} & \dots & \sum_{j\in[k]} w_j\left(e^{i\pi\mu^{(1)}}\right)^{m-1} \\ \sum_{j\in[k]} w_j e^{i\pi\mu^{(1)}} & \sum_{j\in[k]} w_j\left(e^{i\pi\mu^{(1)}}\right)^{2} & \dots & \sum_{j\in[k]} w_j\left(e^{i\pi\mu^{(1)}}\right)^{m} \\ \vdots & \vdots & \ddots & \vdots \\ \sum_{j\in[k]} w_j\left(e^{i\pi\mu^{(1)}}\right)^{m-1} & \sum_{j\in[k]} w_j\left(e^{i\pi\mu^{(1)}}\right)^{m} & \dots & \sum_{j\in[k]} w_j\left(e^{i\pi\mu^{(1)}}\right)^{2m-1} \end{pmatrix} \\
&= \begin{pmatrix} f(0) & f(1) & \dots & f(m-1) \\ f(1) & f(2) & \dots & f(m) \\ \vdots & \vdots & \ddots & \vdots \\ f(m-1) & f(m) & \dots & f(2m-1) \end{pmatrix},
\end{aligned}
$$

where $f(s) = \sum_{j\in[k]} w_j \left(e^{i\pi\mu^{(j)}}\right)^s$ for $s = 0, \dots, 2m-1$ which indeed corresponds to the defined Hankel matrix. The proof is similar for H_1 except that s varies from 1 to $2m$ since element-wise multiplication of H_0 with the diagonal matrix D_μ yields a simple scaling by a factor $e^{i\pi\mu^{(j)}}$.