École Polytechnique Fédérale de Lausanne

Super-Resolution Off the Grid

EPFL - Semester Project in Computer Science

Sébastien Ollquist

22nd March 2022

Abstract

Super-Resolution is the tool that allows us to upgrade the quality of images. The goal of this project is to study a recent paper on super-resolution and discuss the main algorithm in order to eventually find an improvement using information theoretical bounds.

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

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, unfortunately a lot of them are still taken with old technology leading to a weak resolution. Recently, a new theory called super-resolution has emerged. Its aim is to reconstruct an image of better quality from a poor quality image.

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, which may be corrupted with noise.

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 above mentioned 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 complexity and thus make it run faster.

Chapter 2

Mathematical refresher

2.1 A mathematical theory of super-resolution

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 $C > 0$.

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 operations 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 $\lambda_1, \ldots, \lambda_n$ be the eigenvalues of $X^T X$ (with repetitions) and arrange them so that $\lambda_1 \geq \ldots \geq \lambda_n \geq 0$. Then, the $\sigma_1 \geq \ldots \geq \sigma_n \geq 0$ 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

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

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 following equation: $AU = UV \Rightarrow AUU^T = UVU^T \Rightarrow A = UVU^T = UVU^{-1}$, 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

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

where U again are the eigenvectors and V the eigenvalues. Note that A and B are both symmetric. 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 $A\mathbf{x} = \lambda B\mathbf{x}$. Note that the eigenvalues of A are the solution to the generalised eigenvalue problem where $B = I$.

2.3 TENSOR DECOMPOSITION

2.3.1 GENTLE INTRODUCTION

A tensor is a generalisation of a matrix to more than two dimensions. We can think of a tensor as a point in $\mathbb{C}^{m_1\times\dots\times m_k}$ where k is the order of the tensor. Most of the time here, $k = 3$ since three dimensions suffice for our analysis. Note that if T is an order three tensor of dimensions $m_A \times m_B \times m_C$, we can view it as a collection of m_C matrices of size $m_A \times m_B$ stacked on top of each other.

We define the *rank* of a tensor V as the minimum r such that we can write V as the sum of rank one tensors. A rank one tensor will be decomposed in the form of a tensor product of three matrices A, B and C as $V = A \otimes B \otimes C$. The above product is element-wise defined as $V_{i_1,i_2,i_3} = \sum_{j=1}^{k} A_{i_1,j} B_{i_2,j} C_{i_3,j}$.

An alternative definition is given 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:

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

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,

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

2.3.2 JENNRICH'S ALGORITHM

CHAPTER 3

INITIAL WORK

3.1 PRONY'S METHOD

In this section, we will consider

3.2 UNIVARIATE CASE: ADAPTATION OF PRONY'S METHOD

3.3 MULTIVARIATE TOY CASE

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 have $F_{n_1,n_2,n_3} = f(s)\big|_{s=e_{n_1}+e_{n_2}+e_{n_3}}$, for all $n_1, n_2, n_3 \in [d]$. We have the following fact:

Fact 1 *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 $f(e_1 + e_2 + e_3) = \sum_{j=1}^{k} w_j e^{i\pi(\mu_1^{(j)}+\mu_2^{(j)}+\mu_3^{(j)})}$. To do so, we first compute the matrix product $V_d D_w$. We have

$$V_d D_w = \begin{pmatrix} e^{i\pi\mu_1^{(1)}} & \dots & e^{i\pi\mu_1^{(k)}} \\ \vdots & \ddots & \vdots \\ e^{i\pi\mu_d^{(1)}} & \dots & e^{i\pi\mu_d^{(k)}} \end{pmatrix} \begin{pmatrix} w_1 & 0 & \dots & 0 \\ 0 & w_2 & \dots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \dots & w_k \end{pmatrix}$$

$$= \begin{pmatrix} w_1 e^{i\pi\mu_1^{(1)}} & w_2 e^{i\pi\mu_1^{(2)}} & \dots & w_k e^{i\pi\mu_1^{(k)}} \\ w_1 e^{i\pi\mu_2^{(1)}} & w_2 e^{i\pi\mu_2^{(2)}} & \dots & w_k e^{i\pi\mu_2^{(k)}} \\ \vdots & \vdots & \ddots & \vdots \\ w_1 e^{i\pi\mu_d^{(1)}} & w_2 e^{i\pi\mu_d^{(2)}} & \dots & w_k e^{i\pi\mu_d^{(k)}} \end{pmatrix},$$

so that by definition of tensor decomposition, we get

$$\begin{aligned}
F_{n_1,n_2,n_3} &= \sum_{j=1}^{k} [V_d]_{n_1,j}[V_d]_{n_2,j}[V_d D_w]_{n_3,j} \\
&= \sum_{j=1}^{k} e^{i\pi\mu_{n_1}^{(j)}} e^{i\pi\mu_{n_2}^{(j)}} w_j e^{i\pi\mu_{n_3}^{(j)}} \\
&= \sum_{j=1}^{k} w_j e^{i\pi(\mu_{n_1}^{(j)}+\mu_{n_2}^{(j)}+\mu_{n_3}^{(j)})} = f(e_1+e_2+e_3),
\end{aligned}$$

as required. ■

Chapter 4

Main algorithm

CHAPTER 5

DICUSSION

Chapter 6

Improvement

Chapter 7

Conclusion