UNIVERSITÀ DEGLI STUDI DI TRENTO

DEPARTMENT of MATHEMATICS
Master Degree in Mathematics

HOSVD FOR MULTISPECTRAL IMAGES.

A numerical approach to plant biodiversity estimation.

Supervisor:
prof. Alessandra Bernardi
Co-advisor:
prof. Duccio Rocchini

Candidate:
Martina Iannacito

Anno Accademico 2018/2019

A chi è sempre pronto ad aiutare,
a chi compra il cioccolato e
a chi fa i fiocchetti blu,
per mamma, papà e Dani...

Contents

Introduction

Estimating plant biodiversity is a crucial challenge for the present days. Technological improvements provide new ways for such ecological issue. In particular remote sensing has already proved its power. Satellites collect photos of territories at different wavelengths, generating multispectral images. Since plants reflect and absorb in a specific way sunlight, ecologists use matrices derived from these multispectral photos to estimate plant biodiversity of a certain area. We mainly focus on two indexes:

- Rényi, which takes into account only the frequencies of matrix entries;

- Rao, which takes into account both the frequencies of matrix entries and the values themselves of entries.

However working with these data is expensive in terms of storage memory use. Similar difficulties in handling big data difficult are extremely common nowadays, but mathematical solutions have been developed in these years. Usually big data are stored in tensors, a generalisation of matrices, and treated with specific tensor techniques. In applications a very common tensor technique is *High Order Singular Value Decomposition*, HOSVD. Even if there are results of HOSVD applications over multispectral images, as far as we know this is the first attempt in the plant ecology filed. The high storage dimensions of multispectral images cause many difficulties to a not professional laptop in testing the implementations of biodiversity index computation algorithms. Therefore we apply two recent variants of HOSVD to spectral images and we compute biodiversity indexes over the compressed data generated. With the help of University cluster, we tested the HOSVD for two different datasets and provide results about the two biodiversity indexes presented previously. Moreover for completeness we also organise the starting data in two different ways, for both the datasets. Since this is a first attempt we are not able to compare our results with others. However from our data we conclude that for Rényi index compressing data is extremely advantageous since the information lost in estimating biodiversity is few. For Rao index the results are not as satisfactory, even if they are promising. Using approximatively 15% of the original data information, in the first case we show that the average error per pixel is between 5.5% and 7.5%, while in the second one error per pixel is between 17% and 19%. Moreover we present images of the elements

which realise the minimum and the maximum error for different indexes and for different datasets. It is stunning how difficult is for human eyes perceive differences comparing indexes computed from original and approximated data, even when the approximation error is not very low.

Now we briefly illustrate the thesis structure. In the first chapter we introduce briefly remote sensing history and fundaments. Moreover we illustrate the features of *MODerate-resolution Imaging Spectroradiometer* sensor, MODIS, whose data are used in the discussion. Next we present the most significant plant aspects related to spectral reflectance and absorbance. In conclusion to Chapter 1, we find the *Normalized Difference Vegetation Index*, NDVI, definition that is as follow:

Definition. Given a region R, let $RED, NIR \in \mathbb{M}^{m \times n}(\mathbb{R})$ be respectively the RED and the NIR raster band of R. The *normalized difference vegetation index* of region R is $NDVI \in \mathbb{M}^{m \times n}(\mathbb{R})$ such that

$$\text{NDVI}_{ij} = \frac{NIR_{ij} - RED_{ij}}{NIR_{ij} + RED_{ij}}$$

for every $i \in \{1, \ldots, m\}$ and for every $j \in \{1, \ldots, n\}$, when it is defined.

and the most used biodiversity indexes computed over them, as Rao and Rényi index.
Here we highlight the problem we face: multispectral images, necessary to calculate NDVIs images and consequently to estimate biodiversity, occupy huge amount of storage memory. Therefore we look for a compression technique which can reduce the information loss in term of biodiversity index estimation.
In the second chapter we describe the mathematical instruments we use to compress multispectral images. After having reminded the main results of linear algebra, we prove two central theorems:

- **Singular value decomposition theorem**
  Given $A \in \mathbb{M}^{m \times n}(\mathbb{K})$ of rank r, let $\sigma_i(A)$ be the singular values for every $i \in \{1, \ldots, r\}$. Then there exist:

  - two orthogonal matrices $U \in \mathbb{O}(m)$ and $v \in \mathbb{O}(n)$;
  - a matrix $\Sigma \in \mathbb{M}^{m \times n}(\mathbb{R})$ such that $\Sigma_{ii} = \sigma_i$ for every $i \in \{1, \ldots, r\}$ and $\Sigma_{ij} = 0$ otherwise;

  such that $A = U\Sigma V^T$;

- **Schmidt, Mirsky, Eckhart, Young theorem**
  Given a matrix $A \in \mathbb{M}^{m \times n}(\mathbb{R})$ of rank r, let $\Sigma \in \mathbb{M}^{m \times n}$, $U \in \mathbb{O}(m)$ and $V \in \mathbb{O}(m)$ the matrices of the singular value decomposition as in Theorem 2.1.11. For every $i \in \{1, \ldots, r\}$ we define the matrix $S^{(i)} \in \mathbb{M}^{m \times n}(\mathbb{R})$ such

that its (i, i) entry is the i-th singular value and the others entries are zero. Then the best rank 1 approximation of A is given by the matrix $US^{(1)}V^T$, i.e.

$$\left\| A - US^{(1)}V^T \right\| \leq \left\| A - X \right\| \text{ for every } X \in \mathbb{M}^{m \times n}(\mathbb{R}) \text{ of rank 1}.$$

As conclusion of the first section of Chapter 2, we present an example of photo compression as application of the previous theorems. However multispectral images are usually stored in tensors, a generalisation of matrices. So we revise the main properties of tensors. Then in the last section of chapter 2 we present *High order Singular Value Decomposition* which is a generalisation of *Singular Value Decomposition*, SVD. We mainly focus on two recent developments of this technique, *Truncated HOSVD*, T-HOSVD, and *Sequentially Truncated HOSVD*, ST-HOSVD. Indeed we apply these two approximation strategies in order to compress multispectral data. We prove theorems related to their error compared with respect to the best approximation error, when it is defined. Lastly we cite some recent results about generalisation to tensors of Schmidt, Mirsky, Eckhart, Young theorem, SMEY.

In the conclusive chapter we present the data chosen for testing HOSVD techniques. Next we describe `Python` implementations of T-HOSVD, ST-HOSVD and of biodiversity index algorithm. Lastly our approach is introduced: first we compute over NASA NDVI images Rao and Rényi biodiversity indexes. Then we create tensors with the spectral bands necessary to build up NDVI images. We apply T-HOSVD and ST-HOSVD to these generated tensors and we compute new NDVI images, over which we calculate biodiversity indexes. Finally we estimate the error made in computing biodiversity indexes over NASA NDVI with respect to NDVI images from approximated tensors.

From the reported statistics, we highlight that for Rényi index these tensors techniques lead to extremely good results in terms of biodiversity estimation, while for Rao index our results are not optimal, but still appreciable.

Chapter 1

The ecological framework

In this chapter we will present the ecological background of the data we will use in the following chapters.

1.1 Biodiversity and its complexity

Nowadays measuring biodiversity, particularly plant biodiversity, is a crucial issue in ecology. As a matter of fact the loss of biodiversity is of public domain, but currently we can not estimate the scale of phenomenon. The existence and the persistence of biodiversity are decisive for the present and future life of human beings. Indeed not only food, fibres, fuel, pharmaceuticals are based on it, but it also regulates climate, purifies water, affects soil formation, prevents flood. Last but not least the heterogeneity of life beings has a significant impact on aesthetic and culture, [**eco1**; **eco2**].

Let us properly define the concept of biodiversity.

Definition 1.1.1. The Convention on Biological Diversity defined *biodiversity* as the variability, i.e. the set of all the phenotypic and genotypic differences among living organisms, from all sources, including, inter alia, terrestrial, marine and other aquatic ecosystems and the ecological complexes of which they are part; this includes diversity within species, between species and of ecosystems, cf. [**eco3**].

Scientists, national and international organizations report a huge amount of evidences of biodiversity loss. The World Conservation Union, IUCN, indicated in 2002 that one tenth of all the world's bird species and a quarter of its mammals are threatened with extinction, from one to two third of fish, mussels and Crustacea share the same fate. Other studies highlighted that the shark populations in the Northwest Atlantic was reduced as much as 75% since 1986 [**eco4**], while amphibian have decreased by around 80% in 50 years [**eco5**]. More than this alarming signals come also from plant world: from 1999 up to 2015 we assisted of 3% decline of the

total forest area, [**eco6**]. It could seem that estimating the general undergoing biodiversity loss is easy, given all these information, but much of the present knowledge is too fragmentary and selective for taking long term decisions.

Besides collecting data is a quite time expensive process. Not only they must be collected, but also geo-referenced. Such information, which are mainly stored in museum collections, forestry and wildlife department archives, but also into the notebooks of amateur naturalists. Even though the cost per observation is probably very low, the time cost is not negligible.

Lastly we notice that different data lead to different forms of biodiversity estimations. For example we can focus on the changes of the ecosystem behaviour or we can count manually the different species.

Another central way of measuring biodiversity uses the extent of the habitat. Knowing the extent and the rate of change of an habitat can explain the modifications of the population and the species linked with it. Obviously collecting data related to huge territory is not practical through ground surveys: group of ecology experts must cover long distances and collect information, cf. [**eco7**]. However technological progress provided new instrument, which can help ecologists. As a matter of fact the advent of satellites has made possible having real images of a territory, even with a remarkable quality. This approach has created the remote sensing discipline. Many definitions have been proposed during the years for this subject. We refer to the one presented by [**eco8**].

Definition 1.1.2. Remote sensing is the practice of deriving information about the Earth's land and water surfaces using images acquired from an overhead perspective, using electromagnetic radiation in one or more regions of the electromagnetic spectrum, reflected or emitted from the Earth's surface.

This field of study is based on a well known physical phenomenon: different materials and different organisms absorb and reflect electromagnetic radiations differently. In function of its frequencies, measured in Hertz, the radiations or waves will interact differently with the matter and will get different names, cf. [**eco9**]:

- *Radio waves* have frequencies whose order is between 10^3Hz and 10^9Hz. Their main application is in telecommunication, as for radios and televisions;

- *Micro-waves* have frequencies whose order is between 10^9Hz and 10^{11}Hz. Their are abundantly used in scientific research and in radar telecommunications;

- *Infrared waves* have frequencies whose order is between 10^{11}Hz and 10^{14}Hz. They are internally classified in *far infrared*, *mid infrared* and *near infrared* with increasing frequencies;

- *Visible light* collects all the waves with a frequencies between $4.3 \cdot 10^{14}$Hz and $7.5 \cdot 10^{14}$Hz approximatively. We call waves of these frequencies so because human eyes can perceive them, while our brains can interpret them as colours;

- *Ultraviolet waves* have frequencies between $7.5 \cdot 10^{14}$Hz and $3 \cdot 10^{17}$Hz. The Sun produces ultraviolet radiations, which the atmosphere mostly absorbs;

- *X-rays* have frequencies between $3 \cdot 10^{17}$Hz and $5 \cdot 10^{19}$Hz. They are principally used in Medicine and in Chemistry, since the damages caused to cells;

- *Gamma-rays* have frequencies whose order is higher than 10^{18}Hz. They are related to nuclear reactions.

Thanks to the inverse relation which links frequency and wavelength, we can describe and classify radiations also with their characteristic wavelength, measured in meters. Even if human beings can perceive just the so called visible spectrum, in remote sensing other radiations are necessary and fundamental. These radiations are measured through sensors assembled on spacecraft and craft. We can imagine these sensors as special flying cameras. Similarly to our entertainment instruments, which let us collect the best moments of our lives with many modalities, also these sensors can record radiations from the Earth's surface in different ways. They can just collect radiations reflected by the surface of land and oceans. However it is also possible to measure the thermic radiations produced by our planet. Lastly using lasers or radars on the air-platform, we can illuminate the surface and collect data related to present energy. We distinguish these modalities defining the first two as *passive* remote sensing, the last one as *active* remote sensing. If we remind the example of our camera, in the first modality photos are taken without flash, i.e. without any contribution from the instrument, while in the second one the image is recorded using an artificial illumination, i.e. the flash contributes in the registration of the image.
Most of the modern sensors can acquire multiple images, divided in the so called bands.

Definition 1.1.3. *Bands* or *channels* are the recorded spectral measurements.

Depending on the sensor features, we can acquire images dived into the different bands.

Definition 1.1.4. *Multi-spectral sensor* can acquire from 4 up to 10 bands. *Hyperspectral sensor* can acquire more than 100 bands.

Usually in remote sensing we talk about *imagery* instead of images, in order to highlight the difference between common pictures and Earth's surface spectral measurements, which have a strong quantitative nature. To understand the purposes of remote sensing and how the it landed in the ecological studies, we will briefly retrace the history of this discipline. As it is evident, remote sensing relies on photography. After the chemical discoveries of Daguerre, published in 1839, the first aerial photograph was taken by Gaspard-Félix Tournachon, better known as Nadar. This artist acquired a picture of France from a tethered balloon in 1858. Obviously this was closure to an artistic experiment rather than a scientific one. Only after the advent

of airplanes, it was possible to know the speed, the altitude and the directions at which photos were captured. One of the first images was taken by W. Wright in 1909 and it showed the lands near Centocelle, in Rome. The World War I boosted the usage and the technique of aerial photography for military aims, as surveillance and reconnaissance. However even the end of the war could not stop the improvements of this discipline. The years between the two World Wars determined the naissance of photogrammetry, a discipline which analyses aerial photos, and the participation of private industry to aerial mapping collection and application. The governments promoted the development of scientific acquisition and analysation of all the data, whose number was increasing consistently. Moreover in the United States aerial surveys were used to fight the economic depression of 1929. Indeed they could better check the economical improvements in the rural areas. However the fate of remote sensing and wars will cross again. As a matter of fact World War II speeded up scientific and technological discoveries. For the first time it was possible to go beyond the visible spectrum line. In addition to the surveillance of military targets, also terrain features, as vegetation and practicability, were recorded and studied. From the Cold War Era we could observe a progressive separation between military use, as defence and national security, and civil applications. For our purposes, we have to cite the studies of R. Colwell of 1956, [eco10]. For the first time this scientist identified a small unhealthy cereal crops from a coloured infrared film. He was a pioneer in remote sensing for plant science. The advent of satellite, supported by NASA's research programs, gave new impetus to remote sensing. The launch of Landstat 1 in 1972, an Earth observing satellite, brought three great innovations. As first one, data were available for large regions of the planet every day. Moreover they were provided in digital form, stimulating consequently improvements in digital analysis. Lastly it was used as model for the next generation of land observing satellites. Remote sensing benefited also from the development of geographic information system, which unified remote sensed and other geospatial data. In addition to the hyperspectral remote sensing new technologies of 90s, we remark the launch of Terra-1, a NASA satellite. For the first time a satellite was created and used with the main aim of monitoring changes in Earth's ecosystem. Obviously the power of Internet of this century enabled the diffusion and the public distribution of remote sensed data worldwide. Near to the spread of personal navigation devices, recent studies had led improvements in acquisition of fine-resolution imagery, with a significant increase in the data dimension, cf. [eco8].

The data used in this thesis come from the National Aeronautics and Space Administration, better known as NASA, cf. [eco11]. Its mission is summed up as "to improve life here, to extend life to there, to find life beyond", [eco12]. The program which most concerns us is the Earth Observing System, structured in the 80s. The EOS program principally investigates variables as:

- radiations, clouds, water vapor, precipitations;

- oceans;

- greenhouse gases;

- surface hydrology and ecosystem processes, as land cover change;

- glaciers, sea ice, and ice sheets;

- ozone and stratospheric chemistry;

- volcanoes and climate effects of aerosols.

Part of the NASA funds are invested in improving data final product. Indeed there are many calibration process both before the launch and after it. Physical and engineering tests are conducted in laboratory to ensure the best quality in the information acquisition. On the other side after the data collection, scientists' teams collaborate to validate and publish the data. Besides to encourage international scientific cooperation the data are subjected to open access policy.

1.1.1 Chosen data source

Data used in the present thesis come from the *MODerate-Resolution Imaging Spectroradiometer* sensor, or MODIS, built on both the satellite Terra and Aqua. Terra was launched in 1999, while Aqua in 2002. Thanks to technological improvements Aqua is significantly lighter than Terra, 2934Kg insead of 5190kg. Their rotation period has a lag of six hours. As one can directly image from the satellite name, Terra main aim is observing and collecting information of lands and atmosphere, while Aqua deals with ocean sensing. MODIS measures 36 bands in the visible and infrared spectrum, at different resolutions. Firstly there are RED and NIR band with pixel size of 250km and next 5 bands, still with RED and NIR, at 500m of spatial resolution. These are extremely useful for land observation. The remaining bands with 1km of resolution consist of monitoring images from visible spectrum, MIR, NIR and TIR. The data are registered four times a day, twice daytime and twice night-time. Images usually include the entire Earth's surface.

All these data are usually stored as raster. As we will see in the following chapters each image can be seen as a matrix, whose values represent a certain amount of grey in white-black picture. A similar procedure is possible for satellite images.

Definition 1.1.5. *Raster* is a collection of numbers stored in matrix. *Pixels* are the entries of the raster. Each pixel has defined spatial attributes.

Among these spatial attributes, we find:

- *spatial resolution* which describes the dimension of each pixel in meters. Obviously higher is the spatial resolution higher will be the visible details in the image;

- *temporal resolution* which expresses the maximum theoretical rate of data availability. In other words temporal resolution estimates the frequency, at which the satellite captures a certain areas;

- *radiometric resolution* which defines the number of possible unique integer values that can be stored by the sensor in a pixel. Usually it is expressed in bit. For example a radiometric resolution of 8 bit can store 2^8 unique values per pixel, i.e. all integers between 0 and 255;

- *spectral resolution* corresponds to the number of bands recorded. We have seen that MODIS can record 36 bands, i.e. its spectral resolution is 36 bands.

All these properties of the imageries are stored as metadata, i.e. data related to the main data, which in this case are spectral images. The most common file format for rasters is `GEOTiff`.

1.2   Vegetation data

We have already highlighted that different data are linked with different ecological aims. Since the goal of this discussion is plant ecology, we will briefly describe the main aspects of vegetal organisms.
Plants, as other living beings, are *autotrophic*, in other words they can manufacture complex organic nutritive compounds from simple inorganic sources, as solar light. The crucial step for plant survival is *photosynthesis.*

Definition 1.2.1. *Photosynthesis* process lets organisms produce molecules, in which energy is stored in form of strong chemical bounds. The chemical equation of photosynthesis is:

$$6\,CO_2 + 6\,H_2O + \text{light energy} \longrightarrow C_6H_{12}O_6 + 6\,O_2.$$

Photosynthesis does not take place in every possible cell of plant. In fact plants developed specialised organic structures to boost their photosynthetic ability. The most common leaf section, shown in Figure 1.1 consists of different stratus, listed from the top:

- *cuticle* the most external upper layer, which looks like waxy and translucent;

- *upper epidermis*, characterised by dense specialised cells;

- *palisade tissue*, where there are many vertically stretched cells;

- *mesophyll tissue*, recognizable for its irregular cells and for frequent openings;

- *lower epidermis*, last layer is different from the upper one just for its *stomata.* They are surrounded by guards cells, whose mission is regulating the air movements into the leaves.
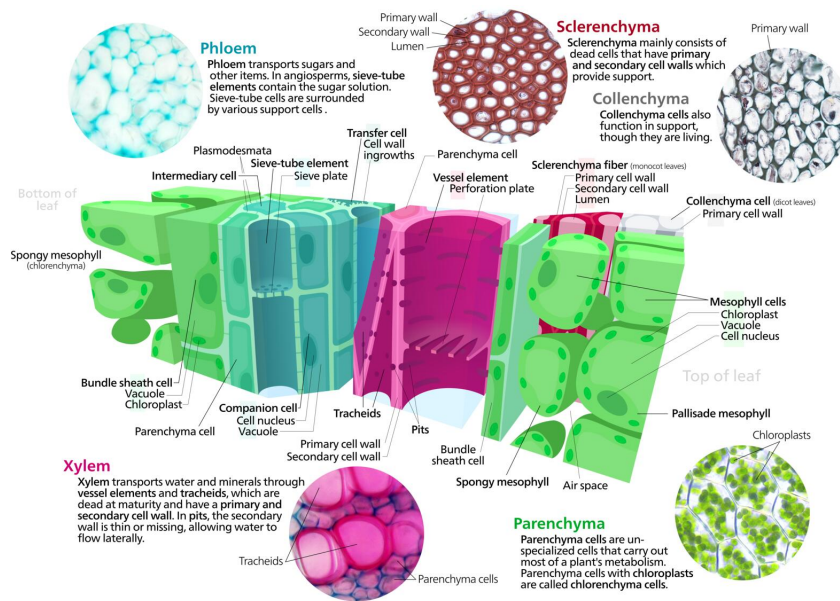
Figure 1.1: Common leaf section.

Carbon dioxide, CO_2, enters in the plant cells from the stomata. Moreover these vegetal "doors" balance the moisture concentration into the entire leaf. The mesophyll tissue cells have a large surface, evolved to facilitate CO_2 and O_2 exchange, fundamental for photosynthesis and respiration. Plants leaves developed a special *organelle*, called *chloroplast* where the photosynthesis takes place. We find chloroplasts into palisade cells. Their presence makes the upper side of a leaf looks darker than the lower ones. If water is usually collected by roots and by stem system, the last photosynthesis ingredient, sunlight, is mostly acquired by the upper cuticle. It evolved differently depending on the environmental conditions of the plant habitat. Species used to live in bright areas have a thick upper cuticle, which prevents water loss. Conversely the ones growing in shaded zones have thin cuticle, in order to maximize the amount of collected light. As proved by Philpoot in 1971 the cuticle diffuses much of the available sunlight to cells and reflects poorly. Scientists discovered that mostly visible wavelengths of sunlight reach chloroplasts. There evolution selected the most efficient molecules to store sunlight energy, i.e. *pigments*.

Definition 1.2.2. *Pigments* molecules reflect and absorb only their own characteristic wavelength, when they are hit by a light wave.

In the most common green vegetation there are different type of pigments so that the spectral region of wavelengths between $0.35\,\mu m$ and $0.70\,\mu m$ is partially absorbed, cf. [**eco13**]. The most studied plants pigments are:

- *Chlorophyll a* which absorbs blue visible light, with wavelength $0.43\,\mu m$ and red one, with wavelength $0.66\,\mu m$;

- *Chlorophyll b* which absorbs blue visible light, with wavelength $0.45\,\mu m$ and red one, with wavelength $0.65\,\mu m$;

- *Carotenes*, as the well known β-*Carotenes*, which has a strong absorption of blue wavelength region around $0.45\,\mu m$ wavelength;

- *Phycoerythrin* whose absorption region is concentrated around $0.55\,\mu m$ value of wavelength, i.e. the green visible light spectral region.

- *Phycocyanin* which absorbs mostly the spectral regions around $0.62\,\mu m$ value of wavelength, i.e. the green and red visible light spectral region.

Even though the pigments present in vegetation are different and various, usually chlorophyll a and b are dominant. We can notice that their action creates a hole in the absorption spectral region, centred around $0.45\,\mu m$ wavelength, which corresponds to the visible green spectrum. The appreciable consequence is the usual green colour of healthy leaves. Moreover we have surely observed that when our plants are overwatered, under-lighted or simply it is fall time, their leaves becomes yellow, red or brown. The biological explanation of this event is linked with pigments. Indeed in stressful condition or for some vegetal species just coming winter, the chlorophyll disappears, letting carotenes and other pigments becoming dominant.

Consequently scientists can detect seasonal senescence, stress and health of vegetation using of remote sensing instruments to control the chlorophyll absorption. Indeed it represents a significant biophysical variable, which supported by others data can describe the ecological status of a system. So far we have focused the interaction between visible spectrum and vegetal organisms. However more subtle spectral information can be collected through sensor with higher spectral resolution. Spectral images from healthy green leaves show that almost the entire NIR wavelength range is not absorbed. Reminding that we use infrared radiations to quickly defrost food, we intuitively understand this evolutionary selection. If plants could have absorbed NIR wavelengths so efficiently as they absorb visible light, their proteins and consequently their entire cellular structure would have get too hot and so chemically denatured, i.e. damaged. Evolutionary pressure leaded vegetal organisms to a reflection rate of 40 to 60 percent and to a transmission rate to the underling leaves of 40 to 60 percent of NIR wavelengths radiations. Hence, the absorption rate is usually around 5 to 10 percent. Moreover, biologists observed an interesting phenomenon of leaves reflectance, called *leaf additive reflectance*. When a plant has a great number of leaves layers, it presents higher infrared reflectance rate. On the other side if the leaves are sparse, the NIR reflectance rate will be lower with respect to the previous case. As a matter of fact if we assume that at the first layer from the top each leaf will reflect 50% of the total NIR radiation and the remaining infrared radiation will be transmitted at the second layer, the second layers leaves will reflect only half of the half original NIR wave, i.e. 25% of the original radiation. From this second stratum, reflected NIR will hit again foliage from the first layer, which again reflects only 12.5% of the original radiation. In conclusion we get that 62.5% of the total

NIR radiation is reflected, with a very small contribution from the second stratum. Therefore if the layers are abundant, the total plant NIR reflectance rate will be higher, since at each stratum the radiation will be partially reflected from bottom leaves to the upper ones, increasing the contribution of each single leaf. On the other hand if the layers are few, foliage will prevalently transmit NIR radiation, which will quickly reach the soil, and consequently the reflectance rate will be low. This effect is central when instead of single isolated organism we study vegetation canopies, as for example forest or corn field. As a matter of fact vegetation canopies include different plants, which means different leaves shapes, orientations, sizes and distinct soil coverages. The direct effect is a strong difference in NIR reflectance between a single leaf, which reflects on average 50% of the original NIR wavelength radiation, and a canopy, whose NIR reflectance rate is close to 35% of the original sunlight wave. The reflection rate of RED wavelength of hitting radiation is quite the same for single leaf, on average 10% of the original one, and for canopy, up to 5% of the original one, cf. [**eco14**].

From the 60s the scientific community highlighted two important relations between biomass and spectral measurements:

- a direct relation between NIR region and biomass, i.e. greater the biomass greater the NIR reflected radiation measured and vice versa;

- an inverse relation between RED spectral region and biomass, i.e. greater the biomass, lower the RED reflected visible spectrum measured and vice versa.

The relation between NIR and RED reflectance is central for the vegetal biomass estimation. The aim of vegetation indexes is measuring biomass or vegetative vigour on the base of digital brightness values. According to [**eco15**] and [**eco16**] the ideal vegetation index should:

- maximize sensitivity to selected plant biophysical parameters;

- be preferably linear to simplify calibration and validation;

- normalize or describe also external noise effects, as Sun hitting angle, satellite view angle and atmosphere interactions;

- normalize internal noise effects, as soil variations;

- normalize differences in senescent vegetation;

- be matched with measurable biophysical parameters. The most used to compare remote sensed data are

  - *leaf area index*, or LAI, which is the ration between the area of the upper leaf surface of the considered vegetal organisms and the surface of a pixel. It is a dimensionless value and it ranges from 0, for pure soil pixel, up to 6, typical of huge forests;

    − *absorbed photosynthetically active radiation*, or APAR, which expresses the amount of solar energy that vegetation effectively absorb;

    − *vegetation fraction*, or VF, which is the percentage of vegetation occupying each pixel;

    − *net primary production*, or NPP, measures the plants photosynthesis production of a region.

During the last half century, ecologists proposed many vegetation indexes.

1.2.1 Chosen data form

We will focus only on the *Normalized Difference Vegetation Index*, or NDVI, presented in 1974 by Rouse and his collaborators, cf. [**eco17**].

Definition 1.2.3. Given a region R, let $RED, NIR \in \mathbb{M}^{m \times n}(\mathbb{R})$ be respectively the RED and the NIR raster band of R imagery. The *normalized difference vegetation index* of region R is $NDVI \in \mathbb{M}^{m \times n}(\mathbb{R})$ such that

$$\text{NDVI}_{ij} = \frac{NIR_{ij} - RED_{ij}}{NIR_{ij} + RED_{ij}}$$

for every $i \in \{1, \ldots, m\}$ and for every $j \in \{1, \ldots, n\}$, when it is defined.

This index has been widely studied, cf. [**eco18**; **eco19**] and we can now present its advantages and disadvantages. The main benefit of using NDVI is monitoring the seasonal and inter-annual changes in the plant activity. Moreover, its mathematical form reduces many types of noise, as variations of Sun illumination, atmospheric interactions and cloud disturbances. On the oher side, it is not linear and is usually highly correlated with LAI. Consequently it is stretch in favour of low-biomass conditions and it is compressed for the high ones, as forested area. Lastly it is extremely sensitive to soil and more general background variations, cf. [**eco20**; **eco21**]. Anyway NDVI, shown in Figure 1.2 is still today widely used in ecological contexts, which include the biodiversity generalised entropy estimation.
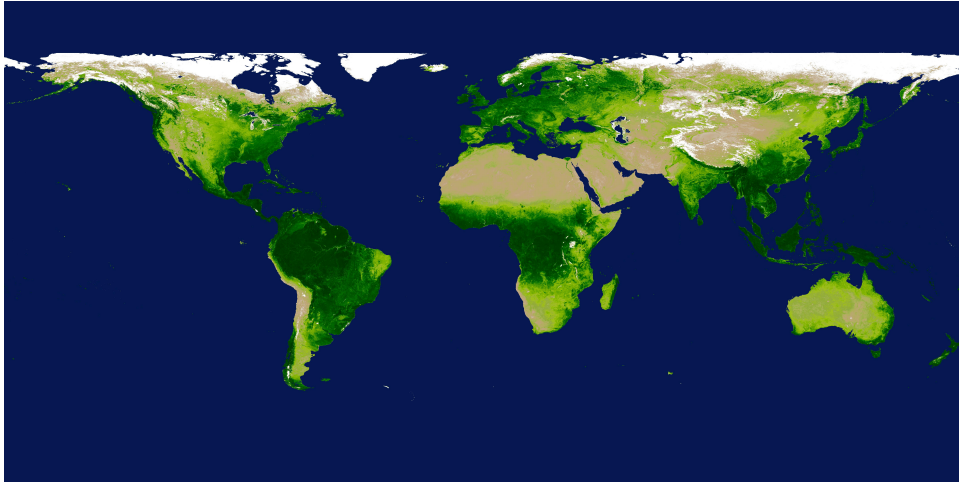
Figure 1.2: Earth NDVI image of November 2018.

1.3   Biodiversity index

So far we have discussed remote sensing, focusing the vegetal world. In this section we will explain how we can take advantage of plant remote sensed data to estimate biodiversity.
Firstly we notice that ecologists distinguish three type of biodiversity: alpha, beta and gamma biodiversity.
In 1972 reminding the studies of Lotka, Volterra, Gause, Elton and Grinnell, Whittaker highlighted that *niche* is the key concept to identify different species and consequently to model exclusion or coexistence with others.

Definition 1.3.1. *Niche* is the position that a specie occupies in terms of resources use, time of activity, vertical location, relation to horizontal pattern, manner of population interaction with other species.*Community* is a system where populations of niche-differentiated species interact.

Whittaker noticed that organisms occupying nearly niches with respect to the resource gradient, which drives evolutionary processes, will evolve in divergent ways. For example in some boreal forests, where the gradient is obviously sunlight, there are from 5 up to 7 main plants with extremely different characteristics, as canopy, smaller tree species, tall and low shrub species, tall and low herbs species and ground-level moss. However in most of the cases evolutionary pressure leaded to a complete fit of all the possible community niches. The direct consequence is a "packing" of species along the gradient, [**eco22**]. Moreover Whittaker underlined that evolutionary diversification on the ecosystem bottom level, for example plants divergent evolution, is transmitted to the all other layers, through the food-chain. Well posed this fundamental evolutionary concepts, we can define the three type of biodiversity presented by [**eco23**] and [**eco24**].

Definition 1.3.2. *Alpha diversity* is the species richness of an area. *Beta diversity* is the extent of species replacement or biotic change along environmental gradients. *Gamma diversity* is the overall diversity of an extensive area.

Since these definitions may seem quite abstract, we will provide an example.

Example 1.3.3. Let R be an extensive region, consisting of 3 specific habitats, as fields, wood and forest areas. We assume to have 8 different species:

$$\{A, B, C, D, E, F, G, H\},$$

whose presence in the different zones is reported in the following table.

Specie	A	B	C	D	E	F	G	H
Field	x	x					x	x
Wood			x			x		x
Forest			x	x	x	x	x	

Table 1.1: Species distribution in different areas of region R.

The alpha diversity for the field area has value 4, since there are 4 different species, for the wood area is equal to 3 and for the forest one is 5. The beta diversity for field and wood area is 5, since they differ for 5 species, for field and forest it is 7 and for the pair wood and forest it is equal to 4. Mathematically the beta diversity is the cardinality of the set difference between the union and the intersection of the area species sets. The gamma diversity of region R is obviously 8, i.e. the total number of present species.

Henceforth we will focus on alpha diversity. Ecologists wish to measure alpha diversity of a sample area estimating both the species richness and the species importance. Counting the number of species is one of the simplest way to measure alpha biodiversity of an area. However it does not give any estimate of the species importance. Moreover if we choose a sample from the whole area, then the specific choice may affect the result. Therefore many various indexes were developed to cross this line. In the second half of the 20th century Simpson proposed its index, cf. [**eco25**].

Definition 1.3.4. Given a sample area with N species and defined p_i the probability of randomly drowning an individual of the i-th species from the area for every $i \in \{1, \ldots, N\}$, the *Simpson index* is

$$I_S = \sum_{i=1}^{N} p_i^2.$$

The *relative abundance* of the i-th species is p_i for every $i \in \{1, \ldots, N\}$.

Each element of the summation is the probability of randomly drown two individuals of the same species from the chosen sample area. Notice that this index expresses degree of dominance. As a matter of fact greater is the presence of a species, greater will be the I_S final value. However this highlights also a limit of the Simpson index: if we assume the probabilities to be decreasingly ordered, the first ones strongly affect it.

Following the idea of Simpson, other indexes were proposed over the years. We recall two of the best known.

Definition 1.3.5. Given a sample area with N species and defined p_i the relative abundances for every $i \in \{1, \ldots, N\}$, in decreasing order, the *Berger-Parker index* is

$$I_{BP} = p_1,$$

the *Shannon-Wiener index* is

$$I_{SW} = -\sum_{i=1}^{N} p_i \log p_i,$$

while the *Rényi index* is

$$I_R = -\log \sum_{i=1}^{N} p_i^2.$$

Notice that the Rényi index is the negative logarithm of the the Simpson index. Moreover in 1961 the Hungarian mathematician Rényi proposed a parametric family of indexes, cf. [**eco26**].

Definition 1.3.6. Given a sample area with N species and defined p_i the relative abundances $i \in \{1, \ldots, N\}$, in decreasing order, the *parametric Rényi index* is

$$I_{R,\alpha} = \frac{1}{1-\alpha} \log \sum_{i=1}^{N} p_i^\alpha,$$

with $\alpha \in \mathbb{R}^+ \setminus \{1\}$.

For different values of α, the Rényi parametric index has a different relation with the previous one.

Remark 1.3.7. *For $\alpha = 0$, we get the logarithmic value of the number of species, i.e. $I_{R,0} = \log S$.*
Notice that even if $I_{R,\alpha}$ is not defined for $\alpha = 1$, we can compute the limit

$$\lim_{\alpha \to 1} \frac{1}{1-\alpha} \log \sum_{i=1}^{N} p_i^\alpha = -\sum_{i=1}^{N} p_i \log p_i$$

applying the l'Hôpital rule.

Ecological literature includes many applications and results of Rényi parametric index family. For example [**eco27**] used this family properties to estimate diversity of tropical forest communities in Sumatra. Besides there are applications of this index also in the animal ecology field, as [**eco28**] and [**eco29**].

1.3.1 Indexes for remote sensed images

These indexes are usually developed in the information theory field, cf. [**eco30**]. The innovative idea of [**eco31**] was the application of information theory studies to remote sensed images. Let u a pixel be our basic unit. Given a remote sensed image, of radiometric resolution $N \in \mathbb{N}$, for every u of this image we find two associated information:

- $s(u)$ the spatial component which expresses the position of u in the image;

- $z(u)$ the property of u, which in our case is a value $v \in \{0, \ldots, N-1\}$.

Remark 1.3.8. *If instead of an image we consider a raster $A \in \mathbb{M}^{m \times n}(\mathbb{R})$, of radiometric resolution $N \in \mathbb{N}$, the pixels are the entries of A. So we associate to a_{ij} entry of A*

- *the entry coordinates as spatial component, i.e. $s(a_{ij}) = (i, j)$;*

- *the entry value as pixel property, i.e. $z(a_{ij}) = a_{ij}$, which will belongs to $\{0, \ldots, N-1\}$*

for every $i \in \{1, \ldots, m\}$ and for every $j \in \{1, \ldots, n\}$.

We assume that the species are the values between 0 and the radiometric resolution of the image. Consequently the number of species is the radiometric resolution and the relative abundance of the i-th specie is the number of times the i-th value is present in the image. However use of Rényi parametric index presents some limits. The next example will clarify it.

Example 1.3.9. Let R be a remote sensed image with 8 different radiometric values, stored in the vector $p \in \mathbb{R}^8$. We show that for different structures of p, the Rényi index remains the same. If $p = [0, 0, 0, 1, 1, 0, 0, 0]$, the relative abundances are $p_0 = 0.75$ and $p_1 = 0.25$. If $p = [0, 0, 7, 0, 0, 0, 7, 0]$, the relative abundances are still $p_0 = 0.75$ and $p_7 = 0.25$. In terms of diversity we notice that the lag between values in the first vectors is lower than the one in the second one. However since this difference is not seen, the Rényi index will be equal to

$$I_{R,\alpha} = \frac{1}{1-\alpha} \log(\frac{1}{4}^\alpha + \frac{3}{4}^\alpha).$$

This obstacle is due to the approach used in the remote sensing: the elements which describe different species are numerical values, not classes. Therefore we can easily establish a measure of difference between them.

Another limit of Rényi and relative abundance based indexes is the difficulty in understanding if diversity comes either from the number of species or from their "distribution", cf. [**eco32**]. Therefore in ecology the aim of using Rényi parametric index is creating a diversity profile, which highlights a relation between the parameter value and the sensitivity to rare or frequent values.

To overcome the limit described in Example 1.3.9, in [**eco33**] an index new to spectral images was presented.

Definition 1.3.10. Given a spectral image of a sample area, let N be the image radiometric resolution and let p_i be the relative abundances of the i-th value for every $i \in \{1, \dots, N\}$. Fixed a distance function d, we build up a pairwise spectral difference matrix $D \in \mathbb{M}^N(\mathbb{R})$ such that

$$D_{ij} = d(i, j)$$

for every $i, j \in \{1, \dots, N\}$. The *Rao's Q index* is

$$I_{RQ} = \sum_{j=1}^{N} \sum_{i=1}^{N} p_i p_j D_{ij}.$$

Notice that Rao's Q index takes into account not only relative abundances of values but also their mutual distances. Many functional ecology applications of Rao's Q index provided significant results, cf. [**eco34**; **eco35**; **eco36**]. Although its theoretical strength, this index is still under investigation in remote sensing diversity estimation.

To conclude this section, we want to briefly describe the method of vegetal biodiversity computation based on remote sensed images. Given a raster, ecologists usually split it into small chunks, called *windows*. To estimate the value of the previously presented indexes, we firstly select l an odd number, which will corresponds to the window side. With this choice, we have a central entry in the window, i.e. the $(\frac{l+1}{2}, \frac{l+1}{2})$ entry, which will be placed as a mask, in position $(1, 1)$ over our raster. The index is therefore computed only with the values which the window covers. Notice that with this choice we will have some missing values, which will not contribute in the index computation. The obtained index value is stored in position $(1, 1)$ in the output raster. The following step is moving the window so that its central entry is over the entry $(1, 2)$ of the raster. The index value computed is stored in the corresponding position of the output raster, i.e. in $(1, 2)$. We proceed in this way until the last entry of output raster is filled.
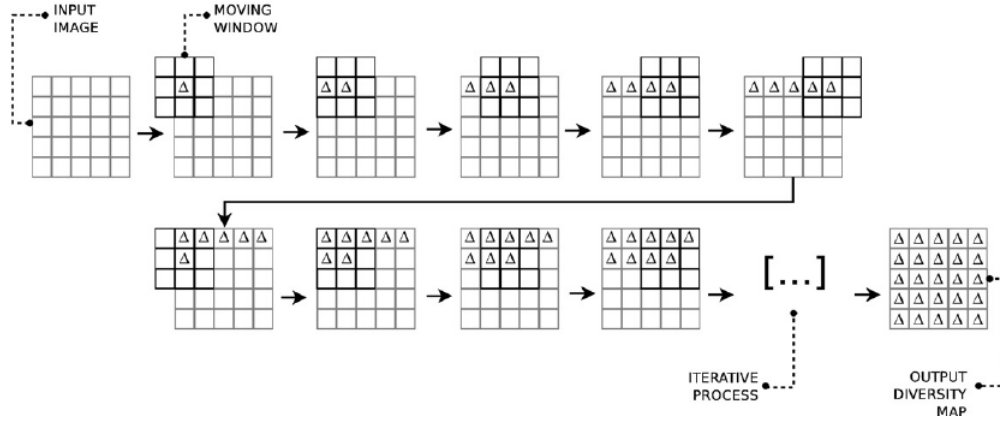
Figure 1.3: The moving window technique for biodiversity index computation, cf. [**eco31**].

This technique, visually presented in Figure 1.3, is extremely popular in ecology since it is quite simple, robust and powerful to simulate biological boundary, cf. [**eco37**; **eco38**]. Moreover notice that this technique of computing index has a suitable algorithmic structure.

Algorithm 1 Index computation with moving window

Input: a raster $R \in \mathbb{M}^{m \times n}(\mathbb{R})$
Input: an index computation function f
Input: an odd number l
Output: the raster with the index values OR
$\quad m \in \mathbb{M}^{l \times l}(\mathbb{R})$ is the moving window;
$\quad$ **for** $i = 1, 2, \ldots, m$ **do**
$\quad\quad$ **for** $j = 1, 2, \ldots, n$ **do**
$\quad\quad\quad$ we assign to m the values of R covered by a l side square mask at the step (i, j);
$\quad\quad\quad$ f computes the index value with values covered by the moving window m;
$\quad\quad\quad$ $OR_{ij} \leftarrow f(m)$;
$\quad\quad$ **end**
$\quad$ **end**

1.4 The problem

Technological improvements in remote sensors lead to a growth, in the number and in the size of the data. For example to store a band of the entire Earth's surface from the MODIS sensor with a low spectral resolution, $5600\,$m, we need around $99\,$MB. Therefore for rasters with higher spectral, spatial or radiometric resolutions the memory request increases significantly. One may point out that $99\,$MB are not

much, if compared with the features of modern machines. However if we have to analyse thousand of them the necessary memory increase quickly. Moreover we remark that in order to perform any index computation over these imageries, we have to load them into the computer RAM, which usually has lower capacity and is occupied also by the system application and by our computing script.

Since the advent of big data, which include multispectral and hyperspectral images, improving our storage and analytical techniques became fundamental. Obviously mathematicians are facing this challenge together with computer scientists, physics and engineers. The usual structure for storing multispectral images are *tensors*. Mathematicians proved the existence of an optimal technique, called *Singular Values Decomposition*, or SVD, for matrices approximation. One application of SVD is image compression. Since we need to store our data in tensors, the generalisation of matrices, we will present *High Order Singular Value Decomposition*, HOSVD, a generalisation to tensors of SVD. Even if only for some special case HOSVD provides optimal results, we are able to present an estimate of the tensor approximation errors. Indeed the core of the present thesis will be the HOSVD technique, its modern versions and the error made using them. In the last chapter we will apply some possible HOSVD implementations to tensors, in which we stored RED and NIR bands. Next from the compressed tensors we get a NDVI rasters and over them we compute biodiversity index. So we will be able to compare our biodiversity estimations from compressed data with the estimations over original data.

As far as we know, this is a first attempt to estimate biodiversity with the presented indexes from HOSVD compressed data. Moreover the obtained results are extremely promising, letting us support its efficiency in ecological framework.

Chapter 2

From SVD to HOSVD

In this second chapter we will introduce the main algebraic and numerical results related to singular value decomposition and to high order singular value decomposition.

2.1   Singular value decomposition

Before stating any theorem, we assume that $\mathbb{K}$ will denote henceforth an algebraical closed field with characteristic 0.

One of the most used results of linear algebra in pure and applied mathematics is the spectral theorem.

Theorem 2.1.1 (Spectral theorem)**.**
Let $\mathcal{V}$ be a vector space over the field $\mathbb{K}$ of finite dimension $n \in \mathbb{N}$. If $T : \mathcal{V} \mapsto \mathcal{V}$ is a linear symmetric operator, then T is a diagonalisable.

Proof. Cf. [**mate1**]. $\qquad\qquad\square$

A special case of this theorem is the following.

Theorem 2.1.2 (Real spectral theorem)**.**
Given a symmetric matrix $A \in \mathbb{M}^n(\mathbb{R})$, then A is orthogonally diagonalisable, i.e. there exist $U \in \mathbb{O}(n)$ and $D \in \mathbb{M}^n(\mathbb{R})$ diagonal such that $A = U^T D U$.

Proof. Cf. [**mate1**] $\qquad\qquad\square$

A particular case of the previous theorem is summed up in the principal axis theorem.

Theorem 2.1.3 (Principal axis theorem)**.**
Given a matrix $A \in \mathbb{M}^n(\mathbb{R})$, the following statements are equivalent:

1. *A is symmetric, i.e. $A^T = A$;*

2. *A is orthogonally diagonalisable, i.e. there exist $U \in \mathbb{O}(n)$ and $D \in \mathbb{M}^n(\mathbb{R})$ diagonal such that $A = U^T D U$;*

3. *there exist an orthonormal basis of $\mathbb{R}^n$ of eigenvectors of A, i.e. there exists $\mathscr{B} = \{v_1, \ldots, v_n\} \subseteq \mathbb{R}^n$ such that for every $i, j \in \{1, \ldots, n\}$:*

   - *$v_i \cdot v_j = \delta_{ij}$;*
   - *$Av_i = \lambda_i v_i$, with $\lambda_i \in \mathbb{R}$.*

Proof. Cf. [**mate2**]. □

Example 2.1.4. Given the matrix $A \in \mathbb{M}^2(\mathbb{R})$ defined as $A = \begin{bmatrix} 6 & 2 \\ 2 & 3 \end{bmatrix}$, it is possible to apply the principal axis theorem, since A is symmetric. After some computations, we have that the eigenvalues are $\lambda_1 = 2$ and $\lambda_2 = 7$, while the relative eigenvectors are

$$v = \frac{1}{\sqrt{5}} \begin{bmatrix} 1 \\ -2 \end{bmatrix} \quad \text{and} \quad \frac{1}{\sqrt{5}} \begin{bmatrix} 2 \\ 1 \end{bmatrix}.$$

If v and w are set column-wise to create the matrix U, we check that:

$$U^T D U = \frac{1}{5} \begin{bmatrix} 1 & -2 \\ 2 & 1 \end{bmatrix} \begin{bmatrix} 2 & 0 \\ 0 & 7 \end{bmatrix} \begin{bmatrix} 1 & 2 \\ -2 & 1 \end{bmatrix} = \frac{1}{5} \begin{bmatrix} 30 & 10 \\ 10 & 15 \end{bmatrix} = \begin{bmatrix} 6 & 2 \\ 2 & 3 \end{bmatrix} = A.$$

Everything in this case works fine, thanks to the main concept of symmetry. However in applications usually matrices are not symmetric neither square. Therefore it is necessary to introduce other more general results.
We prove an easy proposition for real rectangular matrices.

Proposition 2.1.5. *Given $A \in \mathbb{M}^{m \times n}(\mathbb{R})$, then $A^T A$ and $A A^T$ are both symmetric matrices.*

Proof. Fixed $B = A^T A$, then $B^T = (A^T A)^T = (A)^T (A^T)^T = A^T A = B$, thanks to the transpose operation properties. Similarly, fixed $C = A A^T$, we get $C^T = (A A^T)^T = (A^T)^T (A)^T = A A^T = C$, i.e. the thesis. □

Before working with this property, we remind the definition of norm.

Definition 2.1.6. Given $\mathscr{V}$ a vector space with a scalar product, the norm of a vector w is $||w|| = \sqrt{w \cdot w}$ for every $w \in \mathscr{V}$.

Thanks to the previous result, we can apply the principal axis theorem to the matrix $A^T A$ for every $A \in \mathbb{M}^{m \times n}(\mathbb{R})$. Moreover using the third statement of Theorem 2.1.3 there exist an orthonormal basis $\mathscr{B} = \{v_1, \ldots, v_n\}$ such that $(A^T A)v_i = \lambda_i v_i$ with $\lambda_i \in \mathbb{R}$ for every $i \in \{1, \ldots, n\}$. So using associativity of the scalar product and applying the orthogonal eigenvectors definition we get

$$Av_i \cdot Av_j = (Av_i)^T (Av_j) = v_i^T (A^T A v_j) = v_i^T (\lambda_j v_j) = \lambda_j v_i \cdot v_j = \lambda_j \delta_{ij}. \qquad (2.1)$$

for every $i, j \in \{1, \ldots, n\}$. Fixed $i = j$ for every $i \in \{1, \ldots, n\}$, Equation 2.1 leads to an important characteristic of these eigenvalues:

$$\lambda_i = \lambda_i v_i \cdot v_i = A v_i \cdot A v_i = ||A v_i||^2$$

for every $i \in \{1, \ldots, n\}$. Therefore for every $i \in \{1, \ldots, n\}$ the eigenvalue λ_i is not negative. Consequently the following definition is well posed.

Definition 2.1.7. Let $A \in \mathbb{M}^{m \times n}(\mathbb{R})$ be a matrix and let λ_i be eigenvalue of $A^T A$. A *left singular value* of A is $\sigma_i = \sqrt{\lambda_i}$ for every $i \in \{1, \ldots, n\}$.

Obviously one can define the *right singular values* of a matrix A by taking the square roots of the eigenvalues of $A A^T$. In the present discussion we will develop the theory with the left eigenvalues, which for simplicity we will call *singular values*. Given a matrix, we would to know the number of singular values.

Lemma 2.1.8. *Given a matrix $A \in \mathbb{M}^{m \times n}(\mathbb{R})$ of rank r, fixed $\mathcal{B} = \{v_1, \ldots, v_n\}$ an orthogonal basis of eigenvectors of $A^T A$, then $\mathcal{B}' = \{A v_1, \ldots, A v_r\}$ is an orthogonal basis of $\mathcal{I}m(A)$.*

Proof. Obviously the image through A of $\mathcal{B}$ is a subset of $\mathcal{I}m(A)$. By the definition of rank, it follows that $\mathcal{I}m(A)$ has dimension r. If we define h the cardinality of the maximal independent set contained in $A(\mathcal{B})$, then h has to be smaller of equal to r. Without loss of generality we assume that the maximal independent set is

$$\mathcal{B}' = \{A v_1, \ldots, A v_h\}.$$

Let $w \in \mathbb{R}^n$ be a vector with respect the basis $\mathcal{B}$:

$$w = \sum_{i=1}^{n} \omega_i v_i \quad \text{with } \omega_i \in \mathbb{R} \text{ for every } i \in \{1, \ldots, n\}.$$

The image of w through A is:

$$A w = A \left(\sum_{i=1}^{n} \omega_i v_i \right) = \sum_{i=1}^{n} \omega_i A v_i = \sum_{i=1}^{h} \tilde{\omega}_i A v_i \tag{2.2}$$

with $\tilde{\omega}_i \in \mathbb{R}$ for every $i \in \{1, \ldots, h\}$ by the linear property of A. Equation 2.2 shows that every $z \in \mathcal{I}m(A)$ z is a linear combination of elements of $\mathcal{B}'$, i.e. $\mathcal{B}'$ is a basis of $\mathcal{I}m(A)$. Therefore h has to be equal to r. It can be observed that for every $i, j \in \{1, \ldots, r\}$ such that $i \neq j$:

$$(A v_i) \cdot (A v_j)(A v_i)^T (A v_j) = v_i^T (A^T A v_j) = v_i^T (\lambda_j v_j) = \lambda_j v_i \cdot v_j = 0,$$

which means that $\mathcal{B}$ is an orthogonal basis of $\mathcal{I}m(A)$. $\qquad \square$

Corollary 2.1.9. *Given a matrix $A \in \mathbb{M}^{m \times n}(\mathbb{R})$ of rank r, then A has r singular values not necessary distinct.*

Proof. The singular values are defined as:

$$\sigma_i = \sqrt{\lambda_i} = ||Av_i||$$

for v_i eigenvectors of $A^T A$ for every $i \in \{1, \ldots, n\}$. By Theorem 2.1.3, it follows that $\mathcal{B} = \{v_1, \ldots, v_n\}$ is an orthonormal basis of $\mathbb{R}^n$. Without loss of generality we assume that the first r vectors have linear independent image through A. So applying the main lemma, the set $\mathcal{B}' = \{Av_1, \ldots, Av_r\}$ is a basis of $\mathcal{I}m(A)$. Therefore we get that A has r singular values by definition. $\qquad\square$

All along the thesis we will denote the i-th singular value of a matrix A by $\sigma_i(A)$. In order to state and to prove one of the central result in this discussion, it is convenient to assume that henceforth the singular values will be decreasing ordered.

Lemma 2.1.10. *Let $A, B \in \mathbb{M}^{m \times n}(\mathbb{R})$ and $q = \min\{m, n\}$, we get the following equation:*

$$\sigma_{i+j-1}(A + B) \leq \sigma_i(A) + \sigma_j(B)$$

for every $i, j \in \{1, \ldots, q\}$ such that $i + j - 1 \leq q$.

Proof. Cf. [**mate3**] $\qquad\square$

Theorem 2.1.11. *Singular value decomposition*
Given $A \in \mathbb{M}^{m \times n}(\mathbb{K})$ of rank r, let $\sigma_i(A)$ be the singular values for every $i \in \{1, \ldots, r\}$. Then there exist:

- *two orthogonal matrices $U \in \mathbb{O}(m)$ and $V \in \mathbb{O}(n)$;*

- *a matrix $\Sigma \in \mathbb{M}^{m \times n}(\mathbb{R})$ such that $\Sigma_{ii} = \sigma_i$ for every $i \in \{1, \ldots, r\}$ and $\Sigma_{ij} = 0$ otherwise;*

such that $A = U\Sigma V^T$.

Proof. Let $\mathcal{B}_2 = \{v_1 \ldots, v_n\}$ be a basis of eigenvectors of $A^T A$, by Lemma 2.1.8 we can assume, without loss of generality, that the set $\tilde{\mathcal{B}} = \{Av_1, \ldots, Av_r\}$ is an orthogonal basis of $\mathcal{I}m(A)$. For every $i \in \{1, \ldots, r\}$ we define the vector:

$$u_i = \frac{Av_i}{\sigma_i} \in \mathbb{R}^m. \tag{2.3}$$

Consequently the set $\tilde{\mathcal{B}}_1 = \{u_1, \ldots, u_r\}$ is a set of independent orthogonal vectors, which can be completed with $m - r$ vectors of $\mathbb{R}^m$. So the completed set $\mathcal{B}_1 = \{u_1, \ldots, u_m\}$ is an orthogonal basis of $\mathbb{R}^m$.
Sorting the vectors of $\mathcal{B}_1$ and $\mathcal{B}_2$ column-wise, we define two matrices U and V. By

construction $U \in \mathbb{O}(m)$ and $V \in \mathbb{O}(n)$. The matrix Σ of Theorem 2.1.11 turns out to be

$$
\Sigma = \begin{bmatrix}
\sigma_1 & 0 & 0 & 0 & 0 & \ldots & 0 \\
0 & \sigma_2 & 0 & 0 & 0 & \ldots & 0 \\
\vdots & \vdots & \ddots & \vdots & \vdots & \vdots & \vdots \\
0 & 0 & 0 & \sigma_r & 0 & \ldots & 0 \\
0 & 0 & 0 & 0 & 0 & \ldots & 0 \\
\vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\
0 & 0 & 0 & 0 & 0 & \ldots & 0
\end{bmatrix}.
$$

Showing that $A = U\Sigma V^T$ is equivalent to show that $AV = U\Sigma$, because $V^T = V^{-1}$. Computing the j-th column of $U\Sigma$ for $j \in \{1, \ldots, r\}$, we have:

$$
(U\Sigma)_{\cdot,j} = U_{\cdot,\cdot}\Sigma_{\cdot,j} = U \begin{bmatrix} \underline{0} \\ \sigma_j \\ \underline{0} \end{bmatrix} = \sigma_j u_j, \tag{2.4}
$$

while if $j \in \{r+1, \ldots, m\}$, then

$$
(U\Sigma)_{\cdot,j} = U_{\cdot,\cdot}\Sigma_{\cdot,j} = U \begin{bmatrix} 0 \\ \vdots \\ 0 \end{bmatrix} = \underline{0}. \tag{2.5}
$$

Then computing the j-th column of AV for $j \in \{1, \ldots, r\}$, we get:

$$
(AV)_{\cdot,j} = A_{\cdot,\cdot}V_{\cdot,j} = Av_j = \sigma_j u_j \tag{2.6}
$$

by Equation 2.3. On the other side, if $j \in \{r+1, \ldots, n\}$, then

$$
(AV)_{\cdot,j} = A_{\cdot,\cdot}V_{\cdot,j} = Av_j = \underline{0}
$$

because of the rank of A. $\qquad\square$

Before stating the most important consequence of the singular value decomposition theorem, we provide a useful example.

Example 2.1.12. Let $A \in \mathbb{M}^{3\times 2}(\mathbb{R})$ be:

$$
A = \begin{bmatrix} 3 & 2 \\ 2 & 3 \\ 2 & -2 \end{bmatrix},
$$

whose rank is 2. We define the matrix $B = A^T A$, which is

$$
B = A^T A = \begin{bmatrix} 3 & 2 & 2 \\ 2 & 3 & -2 \end{bmatrix} \begin{bmatrix} 3 & 2 \\ 2 & 3 \\ 2 & -2 \end{bmatrix} = \begin{bmatrix} 17 & 8 \\ 8 & 17 \end{bmatrix}.
$$

The characteristic polynomial of B in λ is:

$$p(\lambda) = \lambda^2 - 34\lambda + 225 = (\lambda - 25)(\lambda - 9).$$

Consequently the singular values of A are $\sigma_1 = 5$, $\sigma_2 = 3$. Following the proof steps, it is necessary now to compute the orthogonal eigenvectors of B, which are $v_1 = \begin{bmatrix} v \\ -v \end{bmatrix}$ and $v_2 = \begin{bmatrix} w \\ w \end{bmatrix}$ for every $v, w \in \mathbb{R}$. Fixed $v = w = \frac{1}{\sqrt{2}}$, we get $v_1 = \frac{1}{\sqrt{2}} \begin{bmatrix} 1 \\ 1 \end{bmatrix}$ and $v_2 = \frac{1}{\sqrt{2}} \begin{bmatrix} 1 \\ -1 \end{bmatrix}$. So the matrix V is:

$$V = \frac{1}{\sqrt{2}} \begin{bmatrix} 1 & 1 \\ 1 & -1 \end{bmatrix}.$$

In order to get the matrix U, we compute two mutually orthogonal vectors

$$u_1 = \frac{Av_1}{\sigma_1} = \frac{1}{5\sqrt{2}} \begin{bmatrix} 3 & 2 \\ 2 & 3 \\ 2 & -2 \end{bmatrix} \begin{bmatrix} 1 \\ 1 \end{bmatrix} = \frac{1}{\sqrt{2}} \begin{bmatrix} 1 \\ 1 \\ 0 \end{bmatrix}$$

and

$$u_2 = \frac{Av_2}{\sigma_2} = \frac{1}{3\sqrt{2}} \begin{bmatrix} 3 & 2 \\ 2 & 3 \\ 2 & -2 \end{bmatrix} \begin{bmatrix} 1 \\ -1 \end{bmatrix} = \frac{1}{3\sqrt{2}} \begin{bmatrix} 1 \\ -1 \\ 4 \end{bmatrix}.$$

To complete the matrix U it is necessary to determine a third vector u_3 such that $u_1 \cdot u_3 = u_2 \cdot u_3 = 0$ and $u_3 \cdot u_3 = 1$, as $u_1 \cdot u_1 = u_2 \cdot u_2 = 1$. The first two conditions are equivalent to state that $u_3 = \begin{bmatrix} a \\ b \\ c \end{bmatrix}$ with $a, b, c \in \mathbb{R}$ and $b = -a$ and $2c = -a$. Therefore fixed $a = 2$, the vector $u_3 = \begin{bmatrix} 2 \\ -2 \\ -1 \end{bmatrix}$ normalised becomes $u_3 = \frac{1}{3} \begin{bmatrix} 2 \\ -2 \\ -1 \end{bmatrix}$. Consequently the matrix U is

$$U = \begin{bmatrix} \frac{1}{\sqrt{2}} & \frac{1}{\sqrt{18}} & \frac{2}{3} \\ \frac{1}{\sqrt{2}} & \frac{-1}{\sqrt{18}} & \frac{-2}{3} \\ 0 & \frac{4}{\sqrt{18}} & \frac{-1}{3} \end{bmatrix}.$$

In conclusion we compute:

$$U\Sigma V^T = \frac{1}{6} \begin{bmatrix} 3 & 1 & 2\sqrt{2} \\ 3 & -1 & -2\sqrt{2} \\ 0 & 4 & -\sqrt{2} \end{bmatrix} \begin{bmatrix} 5 & 0 \\ 0 & 3 \\ 0 & 0 \end{bmatrix} \begin{bmatrix} 1 & 1 \\ 1 & -1 \end{bmatrix} = \frac{1}{6} \begin{bmatrix} 18 & 12 \\ 12 & 18 \\ 12 & -12 \end{bmatrix} = A.$$

Remark 2.1.13. *The previous example shows that the third vector u_3 that we added does not contribute to the reconstruction of the original matrix A. This evidently*

holds in general, i.e. since from the $(r+1)$-th column and row of Σ all its entries are zeros, the last columns of U and V from the $(r+1)$-th do not contribute to the final result. As a matter of fact sometimes it is used a second formulation of the singular value decomposition theorem.

Definition 2.1.14. Given a matrix $A \in \mathbb{M}^{m \times n}(\mathbb{R})$ of rank r, its SVD is $U\Sigma V^T$, with $U \in \mathbb{O}(m)$, $\Sigma \in \mathbb{M}^{m \times n}(\mathbb{R})$ and $V \in \mathbb{O}(n)$. Let $\tilde{U} \in \mathbb{M}^{m \times r}(\mathbb{R})$ and $\tilde{V} \in \mathbb{M}^{n \times r}(\mathbb{R})$ be such that $\tilde{U}_{.,j} = U_{.,j}$ and $\tilde{V}_{.,j} = V_{.,j}$ for every $j \in \{1, \ldots, r\}$. Lastly let $\tilde{\Sigma} \in \mathbb{M}^{r \times r}(\mathbb{R})$ be such that $\tilde{\Sigma}_{i,j} = \Sigma_{i,j}$ for every $i, j \in \{1, \ldots, r\}$. The *thin SVD* of A is $\tilde{U}\tilde{\Sigma}\tilde{V}^T$.

One of the most significant consequence of Theorem 2.1.11 is the following approximation result, which will be proved using the Frobenius norm.

Definition 2.1.15. Given a matrix $A \in \mathbb{M}^{m \times n}(\mathbb{K})$, the Frobenius norm of the matrix A is:
$$\left\|A\right\|_F = \sqrt{\sum_{i=1}^{m}\sum_{j=1}^{n}|a|_{ij}^2}$$

Example 2.1.16. Given the matrix $A \in \mathbb{M}^{3 \times 2}$ of Example 2.1.12, its Frobenius norm is:
$$\left\|A\right\|_F^2 = 4(2)^2 + 2(3)^2 = 34.$$

Notice that $tr(A^T A) = 2(17) = \left\|A\right\|_F^2$.

Remark 2.1.17. *We observe that given $A \in \mathbb{M}^{m \times n}(\mathbb{K})$, then*
$$\left\|A\right\|_F^2 = tr(A^H A).$$

Henceforth all along the thesis to shorten the notation we will omit to specify the norm, assuming to use the Frobenius' one.

Lemma 2.1.18. *Given $A \in \mathbb{M}^{m \times n}(\mathbb{R})$ of rank r, if $\sigma_i(A)$ is the i-th singular value of A for every $i \in \{1, \ldots, r\}$, then*
$$\left\|A\right\| = \sqrt{\sum_{i=1}^{r}\sigma_i^2(A)}.$$

Proof. We apply Theorem 2.1.11 to the matrix A, i.e. with $U \in \mathbb{O}(m)$, $V \in \mathbb{O}(n)$ and $\Sigma \in \mathbb{M}^{m \times n}(\mathbb{R})$ such that the first r diagonal elements of Σ are the decreasingly ordered singular values of A and all the others are zero. Consequently by Remark 2.1.17 it follows
$$\left\|A\right\|^2 = \left\|U\Sigma V^T\right\|^2 = tr[(U\Sigma V^T)^T(U\Sigma V^T)]. \tag{2.7}$$

By the orthogonality of U, we obtain from the previous equation that:

$$\left\|A\right\|^2 = tr[(U\Sigma V^T)^T(U\Sigma V^T)] = tr[V\Sigma U^T U\Sigma V^T] = tr(V\Sigma^T\Sigma V^T). \qquad (2.8)$$

Reminding that Σ is a diagonal matrix, as seen in Remark 2.1.17, Equation 2.8 becomes

$$\left\|A\right\|^2 = tr(V\Sigma^T\Sigma V^T) = tr(V\Sigma^2 V^T). \qquad (2.9)$$

In conclusion thanks to trace commutativity property with respect to the product, it follows from 2.9:

$$\left\|A\right\|^2 = tr(V\Sigma^2 V^T) = tr(\Sigma^2 VV^T) = tr(\Sigma^2) = \sum_{i=1}^{r}\sigma_i^2(A),$$

since the orthogonality of V and the diagonal structure of Σ. $\qquad\square$

The previous Lemma will be crucial in the proof of the next result, central for our discussion.

Theorem 2.1.19 (Schmidt, Mirsky, Eckhart, Young theorem).
Given a matrix $A \in \mathbb{M}^{m\times n}(\mathbb{R})$ of rank r, let $\Sigma \in \mathbb{M}^{m\times n}$, $U \in \mathbb{O}(m)$ and $V \in \mathbb{O}(m)$ be the matrices of the singular value decomposition as in Theorem 2.1.11. For every $i \in \{1,\ldots,r\}$ we define the matrix $S^{(i)} \in \mathbb{M}^{m\times n}(\mathbb{R})$ such that its (i,i) entry is the i-th singular value and the others entries are zero. Then the best rank 1 approximation of A is given by the matrix $US^{(1)}V^T$, i.e.

$$\left\|A - US^{(1)}V^T\right\| \leq \left\|A - X\right\| \text{ for every } X \in \mathbb{M}^{m\times n}(\mathbb{R}) \text{ of rank } 1.$$

Proof. Firstly we compute $\left\|A - US^{(1)}V^T\right\|^2$, reminding that by Theorem 2.1.11 $A = U\Sigma V^T$, then

$$\left\|A - US^{(1)}V^T\right\|^2 = \left\|U\Sigma V^T - US^{(1)}V^T\right\|^2 = \left\|U(\Sigma - S^{(1)})V^T\right\|^2, \qquad (2.10)$$

by the distributive property of the product with respect to the sum. Setting the matrix $B = \Sigma - S^{(1)}$ and recalling Remark 2.1.17, we get

$$\left\|A - US^{(1)}V^T\right\|^2 = \left\|UBV^T\right\|^2 = tr[(UBV^T)^T(UBV^T)] = tr(VB^T BV^T). \quad (2.11)$$

Applying the trace commutativity property with respect to the product and the orthogonality of V, Equation 2.11 becomes:

$$\left\|A - US^{(1)}V^T\right\|^2 = tr(VB^T BV^T) = tr(VV^T B^T B). \qquad (2.12)$$

But B is a diagonal matrix, so from the previous equation we have:

$$\left\| A - US^{(1)}V^T \right\|^2 = tr(B^2) = \sum_{i=2}^{r} \sigma_i^2. \tag{2.13}$$

It is now convenient to distinguish two cases. Firstly if $r = 1$, then we have the thesis. In fact $\left\| A - US^{(1)}V^T \right\|^2 = 0$ by the previous equation.

In the second case we assume that $r > 1$. For every $X \in \mathbb{M}^{m \times n}(\mathbb{R})$ of rank 1, we define the matrix $B' = A - X$ and $B'' = X$. Applying Lemma 2.1.10 with $i \in \{1, \ldots, r-1\}$ and $j = 2$, because obviously $i + 1 = i + j - 1 \leq r \leq q$, with $q = \min\{m, n\}$, it follows that

$$\sigma_{i+1}(B' + B'') \leq \sigma_i(B') + \sigma_2(B'') \tag{2.14}$$

which is equivalent to

$$\sigma_{i+1}(A) \leq \sigma_i(A - X) + \sigma_2(X). \tag{2.15}$$

But $rank(X) = 1$ implies that X has only one singular value, as stated in Corollary 2.1.9, so $\sigma_2(X) = 0$. Applying the 2.15 inequality to Equation 2.13 and setting $k = i + 1$ for every $i \in \{1, \ldots, r-1\}$, we obtain:

$$\left\| A - US^{(1)}V^T \right\|^2 = \sum_{k=2}^{r} \sigma_k^2 = \sum_{i=1}^{r-1} \sigma_{i+1}^2(A) \leq \sum_{i=1}^{r-1} \sigma_i^2(A - X). \tag{2.16}$$

Reminding Lemma 2.1.18 and the positivity of square real elements, from 2.16 we get:

$$\left\| A - US^{(1)}V^T \right\|^2 = \sum_{i=1}^{r-1} \sigma_i^2(A - X) \leq \sum_{i=1}^{r} \sigma_i^2(A - X) = \left\| A - X \right\|^2. \tag{2.17}$$

$\square$

Corollary 2.1.20. *Under the hypothesis of Theorem 2.1.19, the matrix $\sum_{i=1}^{k} US^{(i)}V^T$ is the best rank k approximation of A for every $k \in \{1, \ldots, r\}$, i.e.*

$$\left\| A - \sum_{i=1}^{k} US^{(i)}V^T \right\|_F \leq \left\| A - X \right\|_F \ \text{for every } X \in \mathbb{M}^{m \times n}(\mathbb{R}) \text{ of rank } k.$$

Proof. It is convenient to distinguish two cases. Firstly if $k = 1$, then the thesis follows from Theorem 2.1.19.

Otherwise set $k \in \{2, \ldots, r\}$, then we have

$$\left\| A - \sum_{i=1}^{k} US^{(i)}V^T \right\|^2 = \sum_{h=k+1}^{r} \sigma_h^2, \tag{2.18}$$

thanks to the structure of $S^{(i)}$ for every $i \in \{1, \ldots, r\}$, the orthogonality of U and V and the trace properties. Fixed $X \in \mathbb{M}^{m \times n}(\mathbb{R})$ of rank k and $h \in \{1, \ldots, r-1\}$, we apply Lemma 2.1.10 to the matrices $B' = A - X$ and $B'' = X$ with $i \in \{1, \ldots, r-k\}$ and $j = k+1$, which provides $i + j - 1 = i + k \leq r \leq q$ where $q \in \min\{m, n\}$ and

$$\sigma_{i+k}(A) = \sigma_{i+k}(B' + B'') \leq \sigma_i(B') + \sigma_{k+1}(B'') = \sigma_i(A - X) + \sigma_{k+1}(X). \quad (2.19)$$

We observe that X has rank k and consequently $\sigma_{k+1}(X) = 0$, as stated in 2.1.9. Using Inequality 2.19 in Equation 2.18 and setting $h = k+i$ for every $i \in \{1, \ldots, r-k\}$, we have:

$$\left\| A - \sum_{i=1}^{k} U S^{(i)} V^T \right\|^2 = \sum_{h=k+1}^{r} \sigma_h^2 = \sum_{i=1}^{r-k} \sigma_{i+k}^2(A) \leq \sum_{i=1}^{r-k} \sigma_i^2(A - X). \quad (2.20)$$

Recalling Lemma 2.1.18 and the positivity of square real elements, Equation 2.20 becomes:

$$\left\| A - \sum_{i=1}^{k} U S^{(i)} V^T \right\|^2 \leq \sum_{i=1}^{r-k} \sigma_i^2(A - X) \leq \sum_{i=1}^{r} \sigma_i^2(A - X) = \left\| A - X \right\|^2. \quad (2.21)$$

$\square$

Remark 2.1.21. *Notice that Theorem 2.1.19 holds also with the spectral norm and more in general for any norm $\mathbb{O}(m \times n)$ invariant, cf. [***mate4***]*

A possible application of Schmidt, Mirsky, Eckhart, Young theorem is provided by images. As a matter of fact each image of size $m \times n$ in grey scale can be stored into a machine as a matrix of mn real values. Consequently Theorem 2.1.19 leads to an approximation of the image, but also to a reduction of the necessary storage memory.
We now present an example of grey image approximation, implemented in `python`.

Example 2.1.22. Let's take a grey photo, shown at the original quality in Figure 2.1.

Figure 2.1: Cremona cathedral, from [**mate5**].

We need to import and convert a grey image into a matrix. The `Python` libraries we choose are `PIL`, [**m1**], for image importation, and `NumPy`, [**res3**] for image-matrix conversion and SVD computation. To perform singular value decomposition we use `NumPy` command `svd` from `linearalgebra` module. The code for importing, converting and decomposing is the following.

```python
#### algorithm implementation
import sys
import numpy as np
from numpy import linalg as la
#### image manegment
import PIL
from PIL import Image
#### time evaluation
import time

#### loading image
inNAME = "Duomo.JPG"
img = Image.open("Duomo.JPG")
#### converting image in matrix
imgmat = np.array(list(img.getdata(band=0)), np.float64)
dim = imgmat.shape
sizeOR = imgmat.nbytes
#### decomposing la.svd
start_time = time.monotonic()
```

```
ImDec = la.svd(imgmat)
end_time = time.monotonic()
```

In the variable `dim` we store matrix dimension, $(4608, 3456)$, while in `sizeOR` the memory necessary for storing the matrices. Notice that we add also library `time` to estimate the time necessary to compute SVD. It takes approximately $45\,\mathrm{s}$ to compute SVD of the matrix `imgmat`. Next we fix 4 target rank, we reconvert each approximations into an image, presented in Figure 2.2, with the following code.

```
size = dict()
U = ImDec[0]
sigma = ImDec[1]
V = ImDec[2]
Rank = [10,50,100,1000]
for r in Rank:
  size[str(r)]=((U[:,
  ↪ :r]).nbytes)+(np.diag(sigma[:r]).nbytes)+((V[:, :r]).nbytes)
  rec = (U[:, :r]) @ np.diag(sigma[:r]) @ (V[:r, :])
  outNAME = str(r) + inNAME
  out = Image.fromarray((rec))
  out = out.convert("L")
  out.save(outNAME)
```

For each approximation, we have calculated and stored in dictionary `size` how much memory is used to store orthogonal matrices of thin SVD and the diagonal one. Lastly in the following table we list the ratios between values of `sizeOR` and `size` for each rank, which express the rate of compression of the image.

Rank	Rate of compression
10	0.0051
50	0.0255
100	0.2689
1000	0.5692

Table 2.1: Rate of compression.

Notice that with 2.5% of the original image information, we can build up a photo where the most significant elements are easily recognisable, as the brick tower clock, the man, the tables and the bike. When we zoom to the gable inscription in rank 1000 approximation, we can easily read it. Moreover it is astonishing that at a first glance we can not say if the detail of Figure 2.2e or of Figure 2.2f comes from the original photo.

However it happens that coloured images, called RGB, are composed by three overlapping layers: red, green and blue. In this case to propose a similar compression

(a) Rank 10 compression

(b) Rank 50 compression

(c) Rank 100 compression

(d) Rank 1000 compression

(e) Calbe detail from orginal photo

(f) Calbe detail from rank 1000 compression

Figure 2.2: Approximation of Cremona cathedral photo.

strategy we have to change the mathematical structure used, matrices, and introduce tensors.

2.2   Tensors

Firstly we present a theorem which will let us define tensor spaces.

Theorem 2.2.1. *Given $\mathfrak{A}$ a commutative ring, let $\mathcal{M}$ and $\mathcal{N}$ be two $\mathfrak{A}$-modules. Then there exists a pair $(\mathcal{T}, g)$, where T is an $\mathfrak{A}$-module and $g : \mathcal{M} \times \mathcal{N} \mapsto T$ a $\mathfrak{A}$-bilinear map such that:*

- *for every $\mathfrak{A}$-module $\mathcal{P}$ and for every $\mathfrak{A}$-bilinear map $f : \mathcal{M} \times \mathcal{N} \mapsto \mathcal{P}$ there exists a unique $\mathfrak{A}$-linear map $f' : \mathcal{T} \mapsto \mathcal{P}$ such that $f = f' \circ g$:*

- *if $(\mathcal{T}, g)$ and $(\mathcal{T}', g')$ are two pairs with the previous property, then there exists a unique isomorphism $h : \mathcal{T} \mapsto \mathcal{T}'$ such that $h \circ g = g'$.*

Proof. Cf. [**mate6**]. □

Corollary 2.2.2. *Given $\mathfrak{A}$ a commutative ring, let $\mathcal{M}_1, \ldots, \mathcal{M}_n$ be two $\mathfrak{A}$-modules. Then there exists a pair $(\mathcal{T}, g)$, where $\mathcal{T}$ is an $\mathfrak{A}$-module and $g : \mathcal{M}_1 \times \mathcal{M}_2 \times \ldots \times \mathcal{M}_n \mapsto T$ a $\mathfrak{A}$-multilinear map such that:*

- *for every $\mathfrak{A}$-module $\mathcal{P}$ and for every $\mathfrak{A}$-multilinear map $f : \mathcal{M}_1 \times \mathcal{M}_2 \times \ldots \times \mathcal{M}_n \mapsto \mathcal{P}$ there exists a unique $\mathfrak{A}$-homomorphism $f' : \mathcal{T} \mapsto \mathcal{P}$ such that $f = f' \circ g$:*

- *if $(\mathcal{T}, g)$ and $(\mathcal{T}', g')$ are two pairs with the previous property, then there exists a unique isomorphism $h : \mathcal{T} \mapsto \mathcal{T}'$ such that $h \circ g = g'$.*

Proof. Cf. [**mate6**]. □

Definition 2.2.3. Given $n \in \mathbb{N}$, $\mathfrak{A}$ a commutative ring and $\mathfrak{A}$-modules $\mathcal{M}_1, \ldots, \mathcal{M}_n$, the *tensor product* of $\mathcal{M}_1, \ldots, \mathcal{M}_n$ is the pair $(\mathcal{T}, g)$ with the properties stated in Theorem 2.2.1. We will denote it by $\mathcal{M}_1 \otimes_{\mathfrak{A}} \mathcal{M}_2 \otimes_{\mathfrak{A}} \ldots \otimes_{\mathfrak{A}} \mathcal{M}_n$.
The map $g : \mathcal{M}_1 \times \mathcal{M}_2 \times \ldots \times \mathcal{M}_n \mapsto \mathcal{M}_1 \otimes_{\mathfrak{A}} \mathcal{M}_2 \otimes_{\mathfrak{A}} \ldots \otimes_{\mathfrak{A}} \mathcal{M}_n$ is such that for every $(m_1, m_2, \ldots, m_n) \in \mathcal{M}_1 \times \mathcal{M} \times \ldots \times \mathcal{M}_n$:

$$g(m_1, m_2, \ldots, m_n) = m_1 \otimes_{\mathfrak{A}} m_2 \otimes_{\mathfrak{A}} \ldots \otimes_{\mathfrak{A}} m_n.$$

When there is no ambiguity about the commutative ring, we will omit it. Moreover tensor product symbol will replace the map g.
Since vector spaces are modules, this definition of tensor product holds also for them.

Remark 2.2.4. *Given $\mathbb{K}$ a field and fixed $d, n_1, \ldots, n_d \in \mathbb{N}$, then we get the tensor product $\mathbb{K}^{n_1} \otimes \ldots \otimes \mathbb{K}^{n_d}$. By Theorem 2.2.1 it is a module over the field $\mathbb{K}$. Consequently the tensor product of the fields $\mathbb{K}^{n_i}$ for $i \in \{1, \ldots, d\}$ is a vector space. If $\mathcal{B}_{n_j} = \{e_1^j, \ldots, e_{n_j}^j\}$ is a basis of $\mathbb{K}^{n_j}$ for every $j \in \{1, \ldots, d\}$, then the set*

$$\mathcal{B} = \{e_{i_1}^1 \otimes e_{i_2}^2 \otimes \ldots \otimes e_{i_d}^d \mid e_{i_j}^j \in \mathcal{B}_{n_j} \text{ for every } i_j \in \{1, \ldots, n_j\} \text{ for every } j \in \{1, \ldots, d\}\}$$

is a basis for $\mathbb{K}^{n_1} \otimes \ldots \otimes \mathbb{K}^{n_d}$.

Thanks to this remark, we can define the canonical tensor basis.

Definition 2.2.5. Let $\mathbb{K}$ be a field and fixed $d, n_1, \ldots, n_d \in \mathbb{N}$, we choose the canonical basis of $\mathbb{K}^{n_j}$, i.e. $\mathcal{B}_{n_j} = \{e_1^j, \ldots, e_{n_j}^j\}$ for every $j \in \{1, \ldots, d\}$. Then the set

$$\mathcal{B} = \{e_{i_1}^1 \otimes e_{i_2}^2 \otimes \ldots \otimes e_{i_d}^d \mid e_{i_j}^j \in \mathcal{B}_{n_j} \text{ for every } i_j \in \{1, \ldots, n_j\} \text{ for every } j \in \{1, \ldots, d\}\}$$

is the *canonical tensor basis* of $\mathbb{K}^{n_1} \otimes \ldots \otimes \mathbb{K}^{n_d}$.

Definition 2.2.6. Given $\mathbb{K}$ be a field and fixed $d, n_1, \ldots, n_d \in \mathbb{N}$, the *order* of the tensor $\mathcal{A}$ is d and the *size* of tensor $\mathcal{A}$ is the tuple $(n_1, \ldots, n_d)$ for every $\mathcal{A} \in \mathbb{K}^{n_1} \otimes \ldots \otimes \mathbb{K}^{n_d}$.

Now we can give a more concrete definition of tensor. Fixed the canonical tensor basis, we can associate to $\mathcal{A}$ a d-array

$$A = [a_{i_1, i_2, \ldots, i_d}]_{i_1, i_2, \ldots, i_d = 1}^{n_1, n_2, \ldots, n_d}$$

uniquely determined with respect the the elements of $\mathcal{B}$ such that:

$$\mathcal{A} = \sum_{i_1=1}^{n_1} \sum_{i_2=1}^{n_2} \ldots \sum_{i_d=1}^{n_d} a_{i_1, i_2, \ldots, i_d} (e_{i_1}^1 \otimes e_{i_2}^2 \otimes \ldots \otimes e_{i_d}^d)$$

with $a_{i_1, i_2, \ldots, i_d} \in \mathbb{K}$ for every $i_j \in \{1, \ldots, n_j\}$ for every $j \in \{1, \ldots, d\}$. However this representation, which is related to the choice of the tensor basis, is not convenient for stating and proving theoretical results. Therefore we will not be use it frequently throughout this discussion. Obviously on tensor space sum and multiplication by scalar factors are well defined. However we want to highlight an operation, frequently used in the decomposition techniques, of multiplication between matrices and tensor.

Definition 2.2.7. Given a field $\mathbb{K}$ and $d, m, n_1, \ldots, n_d \in \mathbb{N}$, fixed $j \in \{1, \ldots, d\}$ and a basis for $\mathbb{K}^{n_i}$ for every $i \in \{1, \ldots, d\}$, the *matrix tensor product* operation is

$$\cdot_j : \mathbb{M}^{m \times n_j}(\mathbb{K}) \times \mathbb{K}^{n_1} \otimes \ldots \otimes \mathbb{K}^{n_j} \otimes \ldots \otimes \mathbb{K}^{n_d} \mapsto \mathbb{K}^{n_1} \otimes \ldots \otimes \mathbb{K}^m \otimes \ldots \otimes \mathbb{K}^{n_d}$$

such that for every $B \in \mathbb{M}^{m \times n_j}(\mathbb{K})$ and for every tensor $\mathcal{A} \in \mathbb{K}^{n_1} \otimes \ldots \otimes \mathbb{K}^{n_d}$ represented with respect to the canonical basis as

$$\mathcal{A} = \sum_{i_1=1}^{n_1} \sum_{i_2=1}^{n_2} \ldots \sum_{i_d=1}^{n_d} a_{i_1,i_2,\ldots,i_d}(e_{i_1}^1 \otimes e_{i_2}^2 \otimes \ldots \otimes e_{i_d}^d),$$

then

$$B \cdot_j \mathcal{A} = \sum_{i_1=1}^{n_1} \ldots \sum_{i_j=1}^{n_j} \ldots \sum_{i_d=1}^{n_d} a_{i_1,\ldots,i_j,\ldots,i_d}(e_{i_1}^1 \otimes \ldots \otimes Be_{i_j}^j \otimes \ldots \otimes e_{i_d}^d).$$

It is trivial to extend this operation to a tuple of matrices thanks to multilinearity:

$$\cdot_{(1,\ldots,d)} : (\mathbb{M}^{m_1 \times n_1}(\mathbb{K}) \times \ldots \times \mathbb{M}^{m_d \times n_d}(\mathbb{K})) \times (\mathbb{K}^{n_1} \otimes \ldots \otimes \mathbb{K}^{n_d}) \mapsto \mathbb{K}^{m_1} \otimes \ldots \otimes \mathbb{K}^{m_d}$$

such that

$$(B_1,\ldots,B_d) \cdot_{(1,\ldots,k)} \mathcal{A} = \sum_{i_1=1}^{n_1} \ldots \sum_{i_j=1}^{n_j} \ldots \sum_{i_d=1}^{n_d} a_{i_1,\ldots,i_j,\ldots,i_d}(B_1 e_{i_1}^1 \otimes \ldots \otimes B_j e_{i_j}^j \otimes \ldots \otimes B_d e_{i_d}^d)$$

for every $B_j \in \mathbb{M}^{m_j \times n_j}(\mathbb{K})$ and for every $\mathcal{A} \in \mathbb{K}^{n_1} \otimes \ldots \otimes \mathbb{K}^{n_d}$ for every $j \in \{1,\ldots,d\}$.

Besides henceforth the matrix multiplication symbol will be omitted, when there is no ambiguity.

Example 2.2.8. Given $d \in \mathbb{N}$ and $n_1,\ldots,n_d \in \mathbb{N}$, we consider the real tensor product vector space $\mathbb{K}^{n_1} \otimes \mathbb{K}^{n_2} \otimes \ldots \otimes \mathbb{K}^{n_d}$.
If $d = 2$, then $\mathbb{K}^{n_1} \otimes \mathbb{K}^{n_2}$ is the vector space of real matrices of size (n_1, n_2). If one wants to show this, firstly perform the existence of a bilinear map $g : \mathbb{K}^{n_1} \times \mathbb{K}^{n_2} \mapsto \mathbb{M}^{n_1 \times n_2}(\mathbb{K})$. The map g is such that for every $(v,w) \in \mathbb{K}^{n_1} \times \mathbb{K}^{n_2}$, its image through g is:

$$g(v,w) = vw^H = \begin{bmatrix} v_1 \\ v_2 \\ \vdots \\ v_{n_1} \end{bmatrix} \begin{bmatrix} w_1 & w_2 & \ldots & w_{n_2} \end{bmatrix} = \begin{bmatrix} v_1 w_1 & v_1 w_2 & \ldots & v_1 w_{n_2} \\ v_2 w_1 & v_2 w_2 & \ldots & v_2 w_{n_2} \\ \vdots & \vdots & \ddots & \vdots \\ v_{n_1} w_1 & v_{n_1} w_2 & \ldots & v_{n_1} w_{n_2} \end{bmatrix}.$$

This product is also called *outer product*. One can observe that the map g is well defined. Moreover we check the bilinearity with respect to the sum in $\mathbb{K}^{n_j}$ for $j \in \{1,2\}$ and to the multiplication by scalars. For every $v,x \in \mathbb{K}^{n_1}$, $w,y \in \mathbb{K}^{n_2}$ and $\alpha, \beta \in \mathbb{K}$, then

$$g(\alpha(v+x), \beta(w+y)) = [\alpha(v+x)][\beta(w+y)]^H = \alpha\beta[(v+x)(w^H + y^H)] =$$
$$= \alpha\beta(vw^H + vy^H + xw^H + xy^H) =$$
$$= \alpha\beta[g(v,w) + g(v,y) + g(x,w) + g(x,y)].$$

To show that the pair $(M^{n_1 \times n_2}(\mathbb{K}), g)$ is isomorphic to $\mathbb{K}^{n_1} \otimes \mathbb{K}^{n_2}$, we fix a basis for $\mathbb{K}^{n_j}$ for every $j \in \{1, 2\}$. So, without loss of generality, we choose $\mathcal{B}_{n_2} = \{e_1, \ldots, e_{n_1}\}$ and $\mathcal{B}_{n_2} = \{f_1, \ldots, f_{n_2}\}$ the canonical basis of $\mathbb{K}^{n_1}$ and $\mathbb{K}^{n_2}$ respectively. Next we underline that for every $A \in \mathbb{M}^{n_1 \times n_2}(\mathbb{K})$ such that the (i, j) entry is $a_{ij} \in \mathbb{K}$ for every $i \in \{1, \ldots, n_1\}$ and for every $j \in \{1, \ldots, n_j\}$, then

$$A = \sum_{i=1}^{n_1} \sum_{j=1}^{n_2} a_{ij} E^{(ij)}$$

where $E^{(ij)} \in \mathbb{M}^{n_1 \times n_2}(\mathbb{K})$ such that the matrix $E^{(ij)}$ has entry (i, j) equal to 1 and all the others are zeros for every $i \in \{1, \ldots, n_1\}$ and $j \in \{1, \ldots, n_2\}$. So the set

$$\mathcal{B} = \{E^{(ij)} \mid \text{ for every } i \in \{1, \ldots, n_1\} \text{ and } j \in \{1, \ldots, n_2\}\}$$

is a basis for $\mathbb{M}^{n_1 \times n_2}(\mathbb{K})$. We highlight that for every $e_i \in \mathcal{B}_{n_1}$, $f_j \in \mathcal{B}_{n_2}$, and for every $i \in \{1, \ldots, n_1\}$ and $j \in \{1, \ldots, n_2\}$ the matrix $E^{(ij)} = e_i f_j^H$. Then for every $\mathscr{P}$ vector space over $\mathbb{K}$ and for every real bilinear map $l : \mathbb{K}^{n_1} \times \mathbb{K}^{n_2} \mapsto \mathscr{P}$ there exists a the map $l' : M^{n_1 \times n_2}(\mathbb{K}) \mapsto \mathscr{P}$ such that for every $E^{(ij)} \in \mathcal{B}$, its image is

$$l'(E^{(ij)}) = l(e_i, f_j)$$

for every $e_i \in \mathcal{B}_{n_1}$ and $f_j \in \mathcal{B}_{n_2}$, $i \in \{1, \ldots, n_1\}$ and $j \in \{1, \ldots, n_2\}$. We remark that l' is a linear map. As a matter of fact for every $i, h \in \{1, \ldots, n_1\}$, for every $j, d \in \{1, \ldots, n_2\}$ and for every $\alpha \in \mathbb{K}$, then

- $\alpha l'(E^{(ij)}) = \alpha l(e_i, f_j) = l(\alpha e_i, f_j) = l'(\alpha E^{(ij)})$;

- $\alpha l'(E^{(ij)}) = \alpha l(e_i, f_j) = l(e_i, \alpha f_j) = l'(\alpha E^{(ij)})$;

- $l'(E^{(ij)}) + E^{(ik)}) = l(e_i, f_j + f_k) = l(e_i, f_j) + l(e_i, f_k) = l'(E^{(ij)}) + l'(E^{(ik)})$;

- $l'(E^{(ij)}) + E^{(hj)}) = l(e_i + e_h, f_j) = l(e_i, f_j) + l(e_h, f_j) = l'(E^{(ij)}) + l'(E^{(hj)})$.

Lastly we prove that $l = l' \circ g$:

$$(l' \circ g)(e_i, f_j) = l'(e_i f_j^H) = l'(E^{(ij)}) = l(e_i, f_j)$$

for every $e_i \in \mathcal{B}_{n_1}$ and $f_j \in \mathcal{B}_{n_2}$ for every $i \in \{1, \ldots, n_1\}$ and $j \in \{1, \ldots, n_2\}$. By the linearity and the uniqueness stated in Theorem 2.2.1, it follows that $\mathbb{M}^{n_1 \times n_2}(\mathbb{K})$ with the outer product is isomorphic to the tensor space $\mathbb{K}^{n_1} \otimes \mathbb{K}^{n_2}$.

Henceforth tensors will be the abstract elements of tensor product of vector spaces. Before discussing further results, we will try to extend some aspects of matrices to tensors. A central concept in the previous section was the rank of a matrix. As matter of fact the singular values theorem and the Schmidt, Mirsky, Eckhart, Young theorem are based on this number. But when the order d of a tensor is greater then 2, defining the rank is not natural anymore. Extending matrix rank definition requested time and efforts. Meanwhile other definitions were proposed, but the one proved to better extend the matrix rank definition is the following.

Definition 2.2.9. Given a field $\mathbb{K}$ and $d, n_1, \ldots, n_d \in \mathbb{N}$, let $\mathcal{A} \in \mathbb{K}^{n_1} \otimes \ldots \otimes \mathbb{K}^{n_d}$ be a tensor. The *rank* of $\mathcal{A}$ is the least $r \in \mathbb{N}$ such that

$$\mathcal{A} = \sum_{i=1}^{r} \alpha_i (v_i^1 \otimes v_i^2 \otimes \ldots \otimes v_i^d)$$

with $\alpha_i \in \mathbb{K}$ and $v_i^j \in \mathbb{K}^{n_j}$ for every $j \in \{1, \ldots, d\}$ and for every $i \in \{1, \ldots, r\}$.

From an applicative point of view, there are other tensor rank definitions, which will be useful. To present the *multilinear rank*, we firstly investigate relations between tensors and tensorial subspaces.

Definition 2.2.10. Given a field $\mathbb{K}$ and $d, n_1, \ldots, n_d \in \mathbb{N}$ and let $\mathcal{V}$ be a subspace of $\mathbb{K}^{n_1} \otimes \ldots \otimes \mathbb{K}^{n_d}$. If there exist $\mathcal{V}_i$ subspace of $\mathbb{K}^{n_i}$ for every $i \in \{1, \ldots, d\}$ such that

$$\mathcal{V} = \mathcal{V}_1 \otimes \ldots \otimes \mathcal{V}_d,$$

then $\mathcal{V}$ is *a separable tensor subspace* of $\mathbb{K}^{n_1} \otimes \ldots \otimes \mathbb{K}^{n_d}$.

Notice that not every subspace of $\mathbb{K}^{n_1} \otimes \ldots \otimes \mathbb{K}^{n_d}$ is separable, as shown in the following example.

Example 2.2.11. Let $\mathcal{V}$ a vector space over the field $\mathbb{K}$. We define the *symmetric 2-tensors* as

$$\mathrm{Sym}^2(\mathcal{V}) = \mathrm{span}\{v \otimes v \mid v \in \mathcal{V}\}.$$

It is a subspace of $\mathcal{V} \otimes \mathcal{V}$, but it can not be written as $\mathcal{W}_1 \otimes \mathcal{W}_2$ with $\mathcal{W}_i$ subspace of $\mathcal{V}$ for every $i \in \{1, 2\}$. Indeed elements of $\mathcal{W}_1 \otimes \mathcal{W}_2$ of the form $w_1 \otimes w_2$ will not necessary have $w_1 = w_2$ with $w_i \in \mathcal{W}_i$ for every $i \in \{1, 2\}$.

The structure of a separable tensor spaces has some consequences over its elements.

Definition 2.2.12. Given a field $\mathbb{K}$ and $d, n_1, \ldots, n_d \in \mathbb{N}$, let $\mathcal{A} \in \mathbb{K}^{n_1} \otimes \ldots \otimes \mathbb{K}^{n_d}$ be a tensor. The *multilinear rank* of $\mathcal{A}$ is the tuple $(r_1, \ldots, r_d)$ such that there exists a minimal separable tensor subspace $\mathcal{V}_1 \otimes \ldots \otimes \mathcal{V}_d$, containing $\mathcal{A}$, in the following sense:

$$r_i = \min_{\mathcal{V}_i \text{ subspace of } \mathbb{K}^{n_i}} \dim(\mathcal{V}_i)$$

for every $i \in \{1, \ldots, d\}$.

The previous definition of multilinear rank is not easy to handle. Therefore we introduce a central concept in tensor decomposition.

Definition 2.2.13. Given a field $\mathbb{K}$ and $d, n_1, \ldots, n_d \in \mathbb{N}$, let $\mathcal{V}_j$ be a subspace of $\mathbb{K}^{n_j}$ for every $j \in \{1, \ldots, d\}$ and let $\sigma \in \mathfrak{S}_d$ be a permutation over d integers. The set $\{\sigma(1), \ldots, \sigma(d)\}$ is split in two subsets $\mathcal{I}_1 = \{h_1 \ldots, h_s\}$ and $\mathcal{I}_2 = \{l_1, \ldots, l_t\}$ such that:

- the cardinality of $\mathcal{I}_1$ and $\mathcal{I}_2$ is respectively s and t, whose sum is equal to d;

- $\mathcal{I}_1 \cup \mathcal{I}_2 = \{1, \ldots, d\}$;

- $\mathcal{I}_1 \cap \mathcal{I}_2 = \emptyset$.

For every tensor $\mathcal{A} \in \mathcal{V}_1 \otimes \ldots \otimes \mathcal{V}_d$ such that:

$$\mathcal{A} = \sum_{i=1}^{r} \alpha_i v_i^1 \otimes \ldots \ldots \otimes v_i^d,$$

the $(\underline{h}, \underline{l})$-*flattening* of $\mathcal{A}$ is

$$\mathcal{A}_{(\underline{h},\underline{l})} = \sum_{i=1}^{r} (\alpha_i v_i^{h_1} \otimes \ldots \otimes v_i^{h_s}) \otimes (v_i^{l_1} \otimes \ldots \otimes v_i^{l_t})$$

with $\alpha_i \in \mathbb{K}$ and $v_i^j \in \mathcal{V}_j$ for every $j \in \mathcal{I}_1 \cup \mathcal{I}_2$ for every $i \in \{1, \ldots, r\}$.
If s is equal to 1, i.e. $\mathcal{I}_1 = \{k\}$, the k-*flattening* is

$$\mathcal{A}_{(k)} = \sum_{i=1}^{r} (\alpha_i v^k) \otimes (v_i^{l_1} \otimes \ldots \otimes v_i^{l_{d-1}})$$

with $\alpha_i \in \mathbb{K}$ and $v_i^j \in \mathcal{V}_j$ for every $j \in \mathcal{I}_1 \cup \mathcal{I}_2$ for every $i \in \{1, \ldots, r\}$.

Remark 2.2.14. *The flattening definition holds on the isomorphism between $\mathcal{V}_1 \otimes \ldots \otimes \mathcal{V}_d$ and $\mathcal{V}_{h_1} \otimes \ldots \otimes \mathcal{V}_{h_s} \otimes \mathcal{V}_{l_1}^* \otimes \ldots \otimes \mathcal{V}_{l_t}^*$.*

Actually the h-flattening of a tensor $\mathcal{A} \in \mathcal{V}_1 \otimes \ldots \otimes \mathcal{V}_d$ is a matrix for every $h \in \{1, \ldots, d\}$. Fixed $\mathcal{B}_i$ the canonical basis of $\mathcal{V}_i$ of cardinality $m_i \in \mathbb{N}$, with $m_i \leq n_i$ for every $i \in \{1, \ldots, d\}$, the flattening $\mathcal{A}_{(h)}$ is represented with a matrix $A_{(h)}$ such that its j-th column is the image through $\mathcal{A}_{(h)}$ of $e_{j_1}^{l_1} \otimes \ldots \otimes e_{j_{d-1}}^{l_{d-1}}$ with respect to $\mathcal{B}_h$ for every $e_{j_q}^{l_p} \in \mathcal{B}_{l_p}$, for every $j_q \in \{1, \ldots, m_q\}$ and for every $q \in \{1, \ldots, d\} \setminus \{h\}$.
To make the computation of multilinear rank strongly easier the following proposition is important.

Proposition 2.2.15. *Given a field $\mathbb{K}$ and $d, n_1, \ldots, n_d \in \mathbb{N}$, let $\mathcal{W}_j$ be a subspace of $\mathbb{K}^{n_j}$ for every $j \in \{1, \ldots, d\}$. If $\mathcal{A} \in \mathcal{W}_1 \otimes \ldots \otimes \mathcal{W}_d$ is a tensor of multilinear rank $(r_1, \ldots, r_d)$, then the subspace $\mathcal{V}_i$ of $\mathcal{W}_i$ generated by the columns of $\mathcal{A}_{(i)}$ has dimension r_i for every $i \in \{1, \ldots, d\}$.*

Proof. For every tensor $\mathcal{A} \in \mathcal{W}_1 \otimes \ldots \otimes \mathcal{W}_d$, we assume that $\mathcal{W}_1 \otimes \ldots \otimes \mathcal{W}_d$ is the subspace with the least dimension to which $\mathcal{A}$ belongs, i.e. $r_i = \dim(\mathcal{W}_i)$ is the i-th component of the multilinear rank for every $i \in \{1, \ldots, d\}$. Then fixed $h \in \{1, \ldots, d\}$, we define the set $\mathcal{I} = \{l_1, \ldots, l_{d-1}\}$ obtained through a permutation of $\{1, \ldots, d\} \setminus \{h\}$.
So by definition, the h-flattening $\mathcal{A}_{(h)}$ belongs to $\mathcal{W}_h \otimes \mathcal{W}_{l_1}^* \otimes \ldots \otimes \mathcal{W}_{l_{d-1}}^*$. But we

notice that fixed a basis for $\mathscr{W}_i$ for every $i \in \{1, \ldots, d\}$, $\mathcal{A}_{(h)}$ can be represented by a matrix $A_{(h)} \in \mathbb{M}^{n_h \times N}$ where $N = \prod_{\substack{i=1 \\ i \neq h}}^{d} n_i$. We can conclude that the N columns of $A_{(h)}$ are coordinates of vectors belonging to $\mathscr{W}_h$. Set $m_h = \dim(\mathscr{V}_h)$ where $\mathscr{V}_h = \text{span}\{(A_{(h)})_{(\cdot,j)}\}_{j=1}^{N}$, by definition of generated subspace, we have that:

$$\mathscr{V}_h \subseteq \mathscr{W}_h$$

which means that $m_h \leq r_h$. However by the minimality of r_h, we obtain that $m_h = r_h$. $\square$

Corollary 2.2.16. *Given a field $\mathbb{K}$ and $d, n_1, \ldots, n_d \in \mathbb{N}$, let $\mathscr{W}_j$ be a subspace of $\mathbb{K}^{n_j}$ for every $j \in \{1, \ldots, d\}$. Then for every $\mathcal{A} \in \mathscr{W}_1 \otimes \ldots \otimes \mathscr{W}_d$ of multilinear rank $(r_1, \ldots, r_d)$ the rank of matrix associated to the i-th flattening is r_i for every $i \in \{1, \ldots, d\}$.*

Proof. By the main proposition $r_i = \dim(\mathscr{V}_i)$, with $\mathscr{V}_i$ the subspace generated by the columns of $A_{(i)}$ for every $i \in \{1, \ldots, d\}$. So we get that $\dim(\mathscr{V}_i) = \text{rank}(A_{(h)})$. $\square$

Example 2.2.17. Consider the matrix space $\mathbb{M}^{m \times n}(\mathbb{K})$ which in Example 2.2.8 has been proved to be isomorphic to $\mathbb{K}^m \otimes \mathbb{K}^n$. Reminding that example, we get that for every $v \in \mathbb{K}^m$ and $w \in \mathbb{K}^n$ then

$$v \otimes w = vw^T = \begin{bmatrix} v_1 \\ v_2 \\ \vdots \\ v_m \end{bmatrix} \begin{bmatrix} w_1 & w_2 & \ldots & w_n \end{bmatrix} = \begin{bmatrix} v_1 w_1 & v_1 w_2 & \ldots & v_1 w_n \\ v_2 w_1 & v_2 w_2 & \ldots & v_2 w_n \\ \vdots & \vdots & \ddots & \vdots \\ v_m w_1 & v_m w_2 & \ldots & v_m w_n \end{bmatrix}.$$

whose rank is 1. We remind that every matrix of rank r can be seen trivially as sum of r rank 1 matrices. Therefore writing every $\mathcal{A} \in \mathbb{K}^m \otimes \mathbb{K}^n$ as a linear combination of r rank 1 tensors is equivalent to writing its correspondent matrix as a linear combination of rank 1 matrices. So the rank of A as matrix is also the rank r of A seen as tensor. Computing now the multilinear rank of $\mathcal{A}$ is equivalent, thanks to Corollary 2.2.16 to compute the rank of the 1-st and 2-nd flattening. So we have that $r_1 = \text{rank}(A) = r$. Then applying the hermitian operator to the first flattening and again Corollary 2.2.16, we get that

$$r_1 = r = \text{rank}(A) = \text{rank}(A^H) = r_2.$$

We have shown a well known concept: row rank and column rank are equal for matrices.

The 2-order tensors constitute a special case: indeed for $d \geq 3$ the previous properties do not hold anymore. We can find a d order tensor such that the rank and components of the multilinear rank do not coincide anymore. We will also show a less trivial multilinear rank computation with the next example.

Example 2.2.18. Given a field $\mathbb{K}$ and $n \in \mathbb{N}$, such that $n \geq 2$, let $u, v \in \mathbb{K}^n$ be linearly independent vectors. Then the tensor

$$\mathcal{A} = u \otimes u \otimes v + u \otimes v \otimes u + v \otimes u \otimes u$$

belongs to $\mathbb{K}^n \otimes \mathbb{K}^n \otimes \mathbb{K}^n$. We know that the rank of $\mathcal{A}$ is 3, cf. [**mate7**]. This is a classical tensor result, cf. eg. [**mate8**].

Since $n \geq 2$ and since u and v are linearly independent, there exists a basis $\mathcal{B}$ of $\mathbb{K}^n$ such that $u, v \in \mathcal{B}$. Let $\mathcal{B}'$ be the tensor basis of $\mathbb{K}^n \otimes \mathbb{K}^n \otimes \mathbb{K}^n$, generated from $\mathcal{B}$ elements. So tensors $u \otimes u \otimes v$, $u \otimes v \otimes u$ and $v \otimes u \otimes u$ belongs to $\mathcal{B}'$. However since $\mathcal{A}$ has not zero coefficients only with respect $u \otimes u \otimes v$, $u \otimes v \otimes u$ and $v \otimes u \otimes u$, we can suppose $n = 2$. Let $u = \begin{bmatrix} 1 \\ 0 \end{bmatrix}$ and let $v = \begin{bmatrix} 0 \\ 1 \end{bmatrix}$. With this choice computing explicitly the tensor, its flattenings and its multilinear rank will be easy.
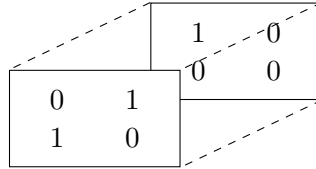

Figure 2.3: Tensor $\mathcal{A}$

The flattenings are

$$\mathcal{A}_{(1)} = \begin{bmatrix} 0 & 1 & 1 & 0 \\ 1 & 0 & 0 & 0 \end{bmatrix} \qquad \mathcal{A}_{(2)} = \begin{bmatrix} 0 & 1 & 1 & 0 \\ 1 & 0 & 0 & 0 \end{bmatrix} \qquad \mathcal{A}_{(3)} = \begin{bmatrix} 0 & 1 & 1 & 0 \\ 1 & 0 & 0 & 0 \end{bmatrix}$$

whose rank is 2. Therefore the rank is different from each component of the multilinear rank, which is $(2, 2, 2)$.

We can find an upper bound for the multilinear rank components with the following result, cf. [**mate9**].

Lemma 2.2.19. *Given $d, n_1, \ldots, n_d \in \mathbb{N}$ and a field $\mathbb{K}$ for every $\mathcal{A} \in \mathbb{K}^{n_1} \otimes \ldots \otimes \mathbb{K}^{n_d}$ of multilinear rank $(r_1, \ldots, r_d)$, we show that:*

$$r_i \leq \prod_{\substack{j=1 \\ j \neq i}}^{d} r_j$$

for every $i \in \{1, \ldots, d\}$.

Proof. Let $\mathcal{A} \in \mathbb{K}^{n_1} \otimes \ldots \otimes \mathbb{K}^{n_d}$ a tensor. By the definition of multilinear rank $\mathcal{A} \in \mathcal{V}_1 \otimes \ldots \otimes \mathcal{V}_d$ with $\mathcal{V}_i$ subspace of $\mathbb{K}^{n_i}$ of dimension r_i for every $i \in \{1, \ldots, d\}$. Chosen $h \in \{1, \ldots, d\}$ and ρ a permutation of $d - 1$ elements, we remind that the

h-flattening $\mathcal{A}_{(h)}$ belongs to $\mathcal{V}_h \otimes \mathcal{V}_{l_1}^* \otimes \ldots \otimes \mathcal{V}_{l_{d-1}}^*$ where $\mathcal{I} = \{l_1, \ldots, l_{d-1}\}$ is the image through ρ of $\{1, \ldots, d\} \setminus \{h\}$. Since we can associate a matrix $A_{(h)}$ to the h-flattening, then we define $N = \prod_{\substack{j=1 \\ j \neq h}}^{d} n_j$ and $\mathcal{V}$ the subspace of $\mathbb{K}^N$ generated by the n_h rows of $A_{(h)}$. Consequently $\mathcal{V}$ is a subspace of $\mathcal{V}_{l_1}^* \otimes \ldots \otimes \mathcal{V}_{l_{d-1}}^*$, which is isomorphic to $\mathcal{V}_{l_1} \otimes \ldots \otimes \mathcal{V}_{l_{d-1}}$. So we have:

$$r_h = \dim(\mathcal{V}_h) = \dim(\mathcal{V}) \leq \prod_{\substack{j=1 \\ j \neq h}}^{d} \dim(\mathcal{V}_j),$$

which is the thesis. $\square$

Before illustrating some of the principal tensor decomposition techniques developed recently, we want to state clearly the problem under investigation, i.e. when and how we can approximate a tensor as we did for matrices. Firstly we explicit the Frobenius norm, our measure, for tensors to describe numerically the goodness of the approximation.

Definition 2.2.20. Given a field $\mathbb{K}$ and $d, m, n_1, \ldots, n_d \in \mathbb{N}$, fixed $j \in \{1, \ldots, d\}$ and a basis for $\mathbb{K}^{n_i}$ for every $i \in \{1, \ldots, d\}$, then the *inner product* is defined as

$$< \cdot, \cdot >: (\mathbb{K}^{n_1} \otimes \ldots \otimes \mathbb{K}^{n_d}) \times (\mathbb{K}^{n_1} \otimes \ldots \otimes \mathbb{K}^{n_d}) \mapsto \mathbb{K}$$

such that for every tensor $\mathcal{A}, \mathcal{B} \in \mathbb{K}^{n_1} \otimes \ldots \otimes \mathbb{K}^{n_d}$ written with respect to the canonical basis as

$$\mathcal{A} = \sum_{i_1=1}^{n_1} \sum_{i_2=1}^{n_2} \ldots \sum_{i_d=1}^{n_d} a_{i_1, i_2, \ldots, i_d} (e_{i_1}^1 \otimes e_{i_2}^2 \otimes \ldots \otimes e_{i_d}^d)$$

and

$$\mathcal{B} = \sum_{i_1=1}^{n_1} \sum_{i_2=1}^{n_2} \ldots \sum_{i_d=1}^{n_d} b_{i_1, i_2, \ldots, i_d} (e_{i_1}^1 \otimes e_{i_2}^2 \otimes \ldots \otimes e_{i_d}^d)$$

then

$$< \mathcal{A}, \mathcal{B} > = \sum_{i_1=1}^{n_1} \sum_{i_2=1}^{n_2} \ldots \sum_{i_d=1}^{n_d} \overline{a}_{i_1, i_2, \ldots, i_d} b_{i_1, i_2, \ldots, i_d}$$

The inner product induces on tensors the Frobenius norm:

$$\|\mathcal{A}\| = \sqrt{< \mathcal{A}, \mathcal{A} >} = \sqrt{\sum_{i_1=1}^{n_1} \sum_{i_2=1}^{n_2} \ldots \sum_{i_d=1}^{n_d} |a_{i_1, i_2, \ldots, i_d}|^2}.$$

In the previous section, we have proved the Schmidt, Mirsky, Eckhart, Young theorem, which states that for every matrix $A \in \mathbb{M}^{m \times n}(\mathbb{K})$ of rank r there exists the best rank s approximation for every $s \in \{1, \ldots, r\}$ in the Frobenius norm. Looking at this optimal results, mathematicians tried to expand it to tensors, developing *High*

Order Singular Value Decomposition, HOSVD. With our discussion we will focus on some important improvements of HOSVD. Vannieuwenhoven et others showed that this truncation optimality holds only for some special tensors, called *orthogonally decomposable tensors*, cf. [**mate10**]. However they highlighted that only elements from an open set of positive Lebesgue measure satisfy this condition in the space of complex tensors. Moreover when the rank is small, this set has zero Lebesgue measure. This is the reason why we will talk about *infimum* instead of *minimum* in the following problem presentation.

Let $\mathcal{A} \in \mathbb{K}^{n_1} \otimes \ldots \otimes \mathbb{K}^{n_d}$ be a tensor and let $(r_1, \ldots, r_d) \in \mathbb{N}^d$ be a tuple. We will discuss if there exists and in case how to determine a tensor $\mathcal{M}$ of multilinear rank lower or equal component-wise than $(r_1, \ldots, r_d)$ which minimizes the Frobenius norm of the tensor difference, i.e.

$$\mathcal{M} = \arg \inf_{\mathrm{mlrank}(\mathcal{T}) \leq (r_1,\ldots,r_d)} \|\mathcal{A} - \mathcal{T}\| \, .$$

This problem is also known as *Low MultiLinear Rank Approximation*, or LMLRA. M. Ishteva and L. De Lathauwer firstly stated these problem and introduced this acronym, cf. [**mate11**].

We underline that looking for a tensor $\mathcal{M} \in \mathbb{K}^{n_1} \otimes \ldots \otimes \mathbb{K}^{n_d}$ satisfying the stated rank properties means searching $\mathcal{V}_i$ subspace of $\mathbb{K}^{n_i}$ of dimension r_i for every $i \in \{1, \ldots, d\}$ such that if the approximation tensor $\mathcal{M}$ exists, it belongs to $\mathcal{V}_1 \otimes \ldots \otimes \mathcal{V}_d$. In other words we search if there exists a family of subspaces of a fixed dimension, whose tensor product minimizes the distance from a fixed tensor, and in case which is. Therefore we can define the following problem, equivalent to the LMLRA. Fixed a tensor $\mathcal{A} \in \mathbb{K}^{n_1} \otimes \ldots \otimes \mathbb{K}^{n_d}$ and a tuple $(r_1, \ldots, r_d) \in \mathbb{N}^d$, we investigate the existence of the subspace $\mathcal{V}_i$ of $\mathbb{K}^{n_i}$ of dimension r_i for every $i \in \{1, \ldots, d\}$ such that

$$\mathcal{V}_1 \otimes \ldots \otimes \mathcal{V}_d = \arg \min_{\substack{\dim(\mathcal{S}_i)=r_i \\ \mathcal{S}_i \subseteq \mathbb{K}^{n_i} \\ \text{for every } i \in \{1,\ldots,d\}}} \inf_{\mathcal{T} \in \mathcal{S}_1 \otimes \ldots \otimes \mathcal{S}_d} \|\mathcal{A} - \mathcal{T}\| \, .$$

Remark 2.2.21. *Fixed $\mathcal{A} \in \mathbb{K}^{n_1} \otimes \ldots \otimes \mathbb{K}^{n_d}$ and a multilinear rank $(r_1, \ldots, r_d)$, by the last formulation of the LMLRA, let $\mathcal{B} \in \mathbb{K}^{n_1} \otimes \ldots \otimes \mathbb{K}^{n_d}$ be such that*

$$\mathcal{B} = \arg \inf_{\mathcal{T} \in \mathcal{S}_1 \otimes \ldots \otimes \mathcal{S}_d} \|\mathcal{A} - \mathcal{T}\|$$

for every $\mathcal{S}_i$ subspace of $\mathbb{K}^{n_i}$ of dimension r_i for every $i \in \{1, \ldots, d\}$. Consequently for every $\mathcal{X} \in \mathcal{S}_1 \otimes \ldots \otimes \mathcal{S}_d$ we define the function $f_{\mathcal{X}} : \mathbb{K} \mapsto \mathbb{K}$ such that $f_{\mathcal{X}}$ is $f_{\mathcal{X}}(t) = \|\mathcal{A} - \mathcal{B} + t\mathcal{X}\|^2$ for every $t \in \mathbb{K}$. Using the definition of Frobenius norm, we get that

$$f_{\mathcal{X}}(t) = <\mathcal{A} - \mathcal{B} + t\mathcal{X}, \mathcal{A} - \mathcal{B} + t\mathcal{X}> = <\mathcal{A} - \mathcal{B}, \mathcal{A} - \mathcal{B}> + 2t <\mathcal{A} - \mathcal{B}, \mathcal{X}> + t^2 <\mathcal{X}, \mathcal{X}>$$

which is equivalent to

$$f_{\mathcal{X}}(t) = \|\mathcal{A} - \mathcal{B}\|^2 + t^2 \|\mathcal{X}\|^2 + 2t <\mathcal{A} - \mathcal{B}, \mathcal{X}> \, .$$

Since f is a polynomial of degree 2, it follows that

$$\lim_{t \to +\infty} f_{\mathcal{X}}(t) = +\infty.$$

Moreover by the definition of $\mathcal{B}$, we get that $f_{\mathcal{X}}(0) = \|\mathcal{A} - \mathcal{B}\|^2 \leq f_{\mathcal{X}}(t)$ for every $t \in \mathbb{K}$. Therefore $t_0 = 0$ has to be a inf of f. Consequently computing the first derivative of $f_{\mathcal{X}}$ with respect to t in t_0, we obtain

$$0 = \frac{df_{\mathcal{X}}}{dt}\Big|_{t_0=0} = 2(t\|\mathcal{X}\|^2 - <\mathcal{A} - \mathcal{B}, \mathcal{X}>)\Big|_{t_0=0} = <\mathcal{A} - \mathcal{B}, \mathcal{X}>.$$

Because the last equation holds for every $\mathcal{X} \in \mathscr{S}_1 \otimes \ldots \otimes \mathscr{S}_d$, we get

$$\mathcal{A} - \mathcal{B} \in (\mathscr{S}_1 \otimes \ldots \otimes \mathscr{S}_d)^{\perp},$$

which means that $\mathcal{B}$ is the orthogonal projection of $\mathcal{A}$ in $\mathscr{S}_1 \otimes \ldots \otimes \mathscr{S}_d$.

Remark 2.2.22. *The idea of this remark is visually trivial, considering the following example. In $\mathbb{R}^2$ we fix a line r and a vector P, which correspond to the linear subspace and the tensor respectively. The point of r with the minimum distance from P is the orthogonal projection of P on r.*
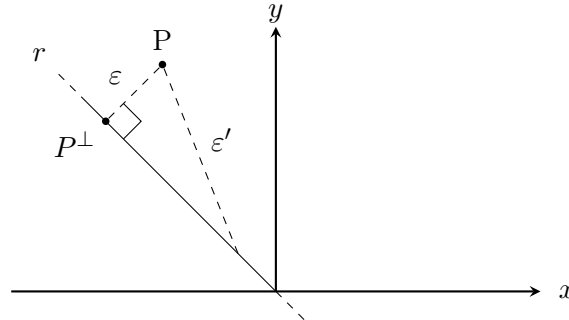
Figure 2.4: The line which minimizes the distance between P and r is ε, which is evidently shorter than ε' for example.

The last remark becomes a clue to develop the HOSVD decomposition techniques, which will be presented. However it is necessary to investigate the structure of projectors over a separable tensor subspace.

Definition 2.2.23. Given $m, n \in \mathbb{N}$ and a field $\mathbb{K}$, fixed $\mathscr{V}$ a subspace of $\mathbb{K}^m$ of dimension n, the *projector into $\mathscr{V}$* is the linear application $P_{\mathscr{V}} : \mathbb{K}^m \mapsto \mathbb{K}^m$ such that:

- $P_{\mathscr{V}}(x) = x$ for every $x \in \mathscr{V}$;

- $P_{\mathscr{V}} \circ P_{\mathscr{V}} = P_{\mathscr{V}}$.

If there is a vector product $< \cdot , \cdot >$ defined over $\mathbb{K}$, then $P_\mathcal{V}$ is an *orthogonal projector over* $\mathcal{V}$ if for every $x, y \in \mathbb{K}$ we get

$$< P_\mathcal{V}(x), y > = < x, P_\mathcal{V}(y) > .$$

Fixed $x \in \mathcal{V}$ and $y \in \mathcal{V}^\perp$, applying the orthogonal projector $P_\mathcal{V}$ to their scalar product we get

$$0 = < x, y > = < P_\mathcal{V}(x), y > = < x, P_\mathcal{V}(y) > .$$

Since the previous equation holds for every $x \in \mathcal{V}$, we have that $P_\mathcal{V}(y) = 0$. Moreover we remind that for every $x \in \mathbb{K}$ there exists $x^\| \in \mathcal{V}$ and $x^\perp \in \mathcal{V}^\perp$ such that $x = x^\perp + x^\|$. So we can observe that

$$P_{\mathcal{V}^\perp}(x^\perp) = x^\perp = x - x^\| = x - P_\mathcal{V}(x^\|) = (I - P_\mathcal{V})(x).$$

For any $P_\mathcal{V}$ orthogonal projector, its orthogonal projector is

$$P_\mathcal{V}^\perp = P_{\mathcal{V}^\perp} = 1 - P_\mathcal{V}.$$

Remark 2.2.24. *Given a vector space $\mathbb{K}^n$ and an orthogonal projector $P_\mathcal{V}$ over its subspace $\mathcal{V}$, if there is a norm over $\mathbb{K}^n$ then $\|P_\mathcal{V}(x)\| \leq \|x\|$. For every $x \in \mathbb{K}^n$ there exists $x^\| \in \mathcal{V}$ and $x^\perp \in \mathcal{V}^\perp$ such that $x = x^\perp + x^\|$. Obviously x can be written as $x = (x - x^\perp) + x^\perp$. So computing the norm we have that*

$$\left\| x \right\|^2 = < (x - x^\perp) + x^\perp, (x - x^\perp) + x^\perp > = \left\| (x - x^\perp) \right\|^2 + \left\| x^\perp \right\|^2$$

since $x^\| = x - x^\perp$ and $x^\perp$ are orthogonal. But reminding that $P_\mathcal{V}(x^\perp) = 0$, we get

$$\left\| x \right\|^2 = \left\| (x - x^\perp) \right\|^2 + \left\| x^\perp \right\|^2 \geq \left\| (x - x^\perp) \right\|^2 = \left\| x^\| \right\|^2 = \left\| P_\mathcal{V}(x^\|) \right\|^2 = \left\| P_\mathcal{V}(x) \right\|^2 .$$

This definition of projector into a subspace is not very handy. However it can be represented with a matrix. Before stating and proving this result, we introduce the Moore-Penrose inverse.

Definition 2.2.25. Given a matrix $A \in \mathbb{M}^{m \times n}(\mathbb{K})$, the *Moore-Penrose inverse*, or *pseudoinverse*, $A^\dagger \in \mathbb{M}^{n \times m}$ is such that

- $AA^\dagger A = A$;

- $A^\dagger A A^\dagger = A^\dagger$;

- $(AA^\dagger)^H = AA^\dagger$;

- $(A^\dagger A)^H = A^\dagger A$.

The following proposition explicit Moore-Penrose inverse.

Proposition 2.2.26. *Given a matrix $A \in \mathbb{M}^{m \times n}(\mathbb{K})$:*

  1. if m is equal to n and A is invertible, then $A^{-1} = A^{\dagger}$;

  2. if m is different from n and if $A^H A$ is invertible, then $A^{\dagger} = A^H (AA^H)^{\dagger}$;

  3. if m is different from n and if AA^H is invertible, then $A^{\dagger} = A^H (AA^H)^{\dagger}$.

Proof.   Cf. [**mate2**].   □

We can now explicit projectors with matrices through the next lemma.

Lemma 2.2.27. *Given $m, n \in \mathbb{N}$ and a field $\mathbb{K}$, let $\mathcal{B} = \{v_1, \ldots, v_m\}$ of $\mathbb{K}^m$ be an orthonormal basis. If $\mathcal{V}$ is a subspace of $\mathbb{K}^m$ of dimension n, then the projector $P_{\mathcal{V}}$ can be represented with respect to the given basis as*

$$P_{\mathcal{V}} = QQ^H$$

where the j-th column of Q is the vector v_{h_j} for every $h_j \in \{1, \ldots, m\}$, for every $j \in \{1, \ldots, n\}$.

Proof. Since $\mathcal{V}$ is a subspace of $\mathbb{K}^m$ of dimension n, we can assume, without loss of generality, that $\mathcal{V}$ is generated by the first n vectors of $\mathcal{B}$. Therefore the matrix $Q \in \mathbb{M}^{m \times n}(\mathbb{K})$ is such that the j-th column of Q is v_j for every $j \in \{1, \ldots, n\}$. We notice that by definition for every $x \in \mathcal{V}$ and for every $i \in \{1, \ldots, n\}$ there exists $\alpha_i \in \mathbb{K}$ such that $v = \sum_{i=1}^{n} \alpha_i v_i$, i.e.

$$\begin{bmatrix} x_1 \\ \vdots \\ v_m \end{bmatrix} = \begin{bmatrix} v_{11} & \ldots & v_{1n} \\ \vdots & \ddots & \vdots \\ v_{m1} & \ldots & v_{mn} \end{bmatrix} \begin{bmatrix} \alpha_1 \\ \vdots \\ \alpha_m \end{bmatrix}.$$

Fixed $a \in \mathbb{K}^m$ such that $a_i = \alpha_i$ for every $i \in \{1, \ldots, m\}$, the last equation is equivalent to $x = Qa$. Applying the Moore-Penrose inverse we get

$$a = (Q^H Q)^{-1} Q^H x.$$

But $Q^H Q$ is the identity matrix, thanks to the orthonormality of the elements of $\mathcal{B}$ and consequently $a = Q^H x$. By definition of linear projector we have that

$$P_{\mathcal{V}}(x) = x = Qa = QQ^H x.$$

Then for every $y \in \mathcal{V}^{\perp}$, obviously $Q^H y = 0$, since

$$Q^H_{(i, \cdot)} y = \overline{v}_i^H y = <\overline{v}_i, y> = 0$$

for every $i \in \{1, \ldots, n\}$ by the definition of orthogonality. On the other side $P_S(y) = 0$ by the orthogonality. Every $w \in \mathbb{K}^m$ can be written as $w = x + y$ with $x \in \mathcal{V}$ and $y \in \mathcal{V}^{\perp}$. Applying the linearity of the projector and the previous identities we obtain

$$P_{\mathcal{V}}(w) = P_{\mathcal{V}}(x + y) = P_{\mathcal{V}}(x) + P_{\mathcal{V}}(y) = QQ^H x + 0 = QQ^H x + QQ^H y = QQ^H w.$$

                                                                                                        □

This link between projectors and matrices will make easier the proof related to the structure of projectors over separable tensor subspaces.

Proposition 2.2.28. *Given $d, n_1, \ldots, n_d \in \mathbb{N}$ and a field $\mathbb{K}$, it is set $\mathcal{V}_i$ a subspace of $\mathbb{K}^{n_i}$ for every $i \in \{1, \ldots, d\}$. If $P_{\mathcal{V}_i}$ is the projector over $\mathcal{V}_i$ for every $i \in \{1, \ldots, d\}$, then fixed the separable tensor subspace $\mathcal{V} = \mathcal{V}_1 \otimes \ldots \otimes \mathcal{V}_d$ its projector is the multi-linear application*

$$P_{\mathcal{V}} = P_{\mathcal{V}_1} \otimes \ldots \otimes P_{\mathcal{V}_d}.$$

Proof. For every subspace $\mathcal{V}_i$ of dimension r_i we fix an orthonormal basis $\mathcal{B}_i = \{q_1^i, \ldots, q_{r_i}^i\}$ for every $i \in \{1, \ldots, d\}$. Consequently we define the matrix $Q_i \in \mathbb{M}^{n_i \times r_i}(\mathbb{K})$ such that its j-th column is the vector q_j^i for every $j \in \{1, \ldots, r_i\}$ for every $i \in \{1, \ldots, d\}$. Applying Lemma 2.2.27, we have that $P_{\mathcal{V}_i}(x) = Q_i Q_i^H x$ for every $x \in \mathbb{K}^{n_i}$ for every $i \in \{1, \ldots, d\}$. Moreover since $\mathcal{B}_i$ is a basis for $\mathcal{V}_i$ for every $i \in \{1, \ldots, d\}$, we get that

$$\mathcal{B} = \{q_{i_1}^1 \otimes \ldots \otimes q_{i_d}^d \mid \text{ for every } i_j \in \{1, \ldots, r_j\} \text{ and for every } j \in \{1, \ldots, d\}\}$$

is a basis for $\mathcal{V}$ by definition. So every tensor $\mathcal{X} \in \mathcal{V}$ can be seen as a linear combination of $\mathcal{B}$ elements, i.e. there exists $\mathcal{B} \in \mathbb{K}^{n_1} \otimes \ldots \otimes \mathbb{K}^{n_d}$ such that

$$\mathcal{X} = (Q_1, \ldots, Q_d)\mathcal{B}.$$

Applying the Moore-Penrose pseudo-inverse for the matrix Q_j for every $j \in \{1, \ldots, d\}$ in the last equation, we obtain

$$(Q_1^H, \ldots, Q_d^H)\mathcal{X} = \mathcal{B}$$

since $(Q_j^H Q_j)$ is the identity matrix of order r_j for every $j \in \{1, \ldots, d\}$.
By properties of the projector, we get that if $\mathcal{X} \in \mathcal{V}$, then

$$P_V(\mathcal{X}) = \mathcal{X} = (Q_1 Q_1^H, \ldots, Q_d Q_d^H)\mathcal{X}.$$

On the other side if $\mathcal{Y} \in \mathbb{K}^{n_1} \otimes \ldots \mathbb{K}^{n_d} \setminus \mathcal{V}$, then $\mathcal{Y} \in \mathcal{V}^\perp = \mathcal{V}_1^\perp \otimes \ldots \otimes \mathcal{V}_d^\perp$ by separability. So since the matrices $Q_j Q_j^H$ act independently one from the other for every $j \in \{1, \ldots, d\}$, we get that

$$P_{\mathcal{V}}(\mathcal{Y}) = 0 = (Q_1 Q_1^H, \ldots, Q_d Q_d^H)\mathcal{Y},$$

because when Q_j^H is applied to $\mathcal{Y}$ we have the 0 vector thanks to the orthogonality. In conclusion every $\mathcal{A} \in \mathbb{K}^{n_1} \otimes \ldots \otimes \mathbb{K}^{n_d}$ can be written as $\mathcal{A} = \mathcal{X} + \mathcal{Y}$ and projecting in $\mathcal{V}$ we obtain:

$$P_{\mathcal{V}}(\mathcal{A}) = P_{\mathcal{V}}(\mathcal{X}) + P_{\mathcal{V}}(\mathcal{Y}) = (Q_1 Q_1^H, \ldots, Q_d Q_d^H)(\mathcal{X} + \mathcal{Y}) = (Q_1 Q_1^H, \ldots, Q_d Q_d^H)\mathcal{A}$$

by linearity of $P_{\mathcal{V}}$ and previous identities. Since the multi-linearity of the projector and the definition of $P_{\mathcal{V}_i}$ for every $i \in \{1, \ldots, d\}$ the thesis follows. $\square$

Corollary 2.2.29. *Given $d, n_1, \ldots, n_d \in \mathbb{N}$ and a field $\mathbb{K}$, we set $\mathcal{V}_i$ a subspace of $\mathbb{K}^{n_i}$ for every $i \in \{1, \ldots, d\}$. Defined $P_{\mathcal{V}_i}$ the projector over $\mathcal{V}_i$ for every $i \in \{1, \ldots, d\}$, for every $i \in \{1, \ldots, d\}$ the multilinear application $\pi_i : \mathbb{K}^{n_1} \otimes \ldots \otimes \mathbb{K}^{n_d} \mapsto \mathbb{K}^{n_1} \otimes \ldots \otimes \mathbb{K}^{n_d}$ is such that*

$$\pi_i(\mathbb{K}^{n_1} \otimes \ldots \otimes \mathbb{K}^{n_d}) = \mathbb{K}^{n_1} \otimes \ldots \otimes P_{\mathcal{V}_i}(\mathbb{K}^{n_i}) \otimes \ldots \otimes \mathbb{K}^{n_d} = \mathbb{K}^{n_1} \otimes \ldots \otimes \mathcal{V}_i \otimes \ldots \otimes \mathbb{K}^{n_d}.$$

Then for every tensor $\mathcal{A} \in \mathbb{K}^{n_1} \otimes \ldots \otimes \mathbb{K}^{n_d}$ we have the commutativity of the projectors composition

$$\pi_i \circ \pi_j(\mathcal{A}) = \pi_j \circ \pi_i(\mathcal{A})$$

for every $i, j \in \{1, \ldots, d\}$.

Proof. If j is equal to i, then the thesis is trivial. We assume that $i, j \in \{1, \ldots, d\}$ are distinct and without loss of generality we suppose that $j = i - 1$. By the multi-linearity of the projector, we have that

$$\pi_i \circ \pi_{i-1}(\mathbb{K}^{n_1} \otimes \ldots \otimes \mathbb{K}^{n_{i-1}} \otimes \mathbb{K}^{n_i} \otimes \ldots \otimes \mathbb{K}^{n_d}) = \pi_i(\mathbb{K}^{n_1} \otimes \ldots \otimes \mathcal{V}_{i-1} \otimes \mathbb{K}^{n_i} \otimes \ldots \otimes \mathbb{K}^{n_d}) =$$
$$= \mathbb{K}^{n_1} \otimes \ldots \otimes \mathcal{V}_{i-1} \otimes \mathcal{V}_i \otimes \ldots \otimes \mathbb{K}^{n_d},$$

while

$$\pi_{i-1} \circ \pi_i(\mathbb{K}^{n_1} \otimes \ldots \otimes \mathbb{K}^{n_{i-1}} \otimes \mathbb{K}^{n_i} \otimes \ldots \otimes \mathbb{K}^{n_d}) = \pi_{i-1}(\mathbb{K}^{n_1} \otimes \ldots \otimes \mathbb{K}^{n_i} \otimes \mathcal{V}_i \otimes \ldots \otimes \mathbb{K}^{n_d}) =$$
$$= \mathbb{K}^{n_1} \otimes \ldots \otimes \mathcal{V}_{i-1} \otimes \mathcal{V}_i \otimes \ldots \otimes \mathbb{K}^{n_d},$$

thanks to the independency of the linear components. $\qquad\square$

Corollary 2.2.30. *Given $d, n_1, \ldots, n_d \in \mathbb{N}$ and a field $\mathbb{K}$, we set $\mathcal{V}_i$ a subspace of $\mathbb{K}^{n_i}$ for every $i \in \{1, \ldots, d\}$. Defined $P_{\mathcal{V}_i}$ the projector over $\mathcal{V}_i$ for every $i \in \{1, \ldots, d\}$, for every $i \in \{1, \ldots, d\}$ we fix the multilinear application $\pi_i : \mathbb{K}^{n_1} \otimes \ldots \otimes \mathbb{K}^{n_d} \mapsto \mathbb{K}^{n_1} \otimes \ldots \otimes \mathbb{K}^{n_d}$ such that*

$$\pi_i(\mathbb{K}^{n_1} \otimes \ldots \otimes \mathbb{K}^{n_d}) = \mathbb{K}^{n_1} \otimes \ldots \otimes P_{\mathcal{V}_i}(\mathbb{K}^{n_i}) \otimes \ldots \otimes \mathbb{K}^{n_d} = \mathbb{K}^{n_1} \otimes \ldots \otimes \mathcal{V}_i \otimes \ldots \otimes \mathbb{K}^{n_d}.$$

Then for every tensor $\mathcal{A} \in \mathbb{K}^{n_1} \otimes \ldots \otimes \mathbb{K}^{n_d}$ we have that

$$P_{\mathcal{V}}(\mathcal{A}) = (\pi_d \ldots \pi_1)(\mathcal{A}),$$

Proof. The thesis follows from the definition of π_i for every $i \in \{1, \ldots, d\}$ and Corollary 2.2.29. $\qquad\square$

With all the necessary instruments well analysed, we can now investigate the Hitchcock-Tucker decomposition. Let's recall the LMLRA problem structure. Given $\mathcal{A} \in \mathbb{K}^{n_1} \otimes \ldots \otimes \mathbb{K}^{n_d}$ and a multilinear rank $(r_1, \ldots, r_d)$, if we know that $\mathcal{A}$ belongs to a separable tensor subspace, i.e. $\mathcal{A} \in \mathcal{V}_1 \otimes \ldots \otimes \mathcal{V}_d$, then fixed a basis $\mathcal{B}_i = \{v_1^i, \ldots, v_{r_i}^i\}$ for $\mathcal{V}_i$ for every $i \in \{1, \ldots, d\}$ there exists $\mathcal{C} \in \mathbb{K}^{n_1} \otimes \ldots \otimes \mathbb{K}^{n_d}$ such that

$$\mathcal{A} = (V_1, \ldots, V_d)\mathcal{C}$$

where the j-th column of V_i is the vector v_j^i for every $j \in \{1, \ldots, r_i\}$. Applying the Moore-Penrose pseudo-inverse we obtain the inverse relation:

$$(V_1^\dagger, \ldots, V_d^\dagger)\mathcal{A} = \mathcal{C}.$$

Unifying the two relations we get the wanted decomposition

$$\mathcal{A} = (V_1, \ldots, V_d)(V_1^\dagger, \ldots, V_d^\dagger)\mathcal{A},$$

which reveals through a basis the separable tensor subspace in which $\mathcal{A}$ lives.
A significant problem is choosing a basis, which is not unique, even when the separable tensor subspace is known. Recently, in 2012, De Lathauwer, De Moor and Vandewalle made a step forward. As reported in [**mate12**] their strategy is based on the singular value decomposition. Fixed $\mathcal{A} \in \mathbb{K}^{n_1} \otimes \ldots \mathbb{K}^{n_d}$, which belongs to the separable tensor subspace $\mathcal{V}_1 \otimes \ldots \otimes \mathcal{V}_d$ with multilinear rank $(r_1, \ldots, r_d)$, we compute for every $j \in \{1, \ldots, d\}$ the singular value decomposition of the j-th flattening of $\mathcal{A}$, i.e.

$$\mathcal{A}_{(j)} = U_j \Sigma_j V_j^T.$$

Then we choose as basis of $\mathcal{V}_j$ the r_j rows of U_j, i.e. those which are naturally found following the proof of Theorem 2.1.11. This technique, called *High Order Singular Value Decomposition* is extremely advantageous. As a matter of fact the Moore-Penrose pseudo-inverse is $U_j^\dagger = (U_j^H U_j)^{-1} U_j^H = U_j^H$ since $U_j^H U_j$ is the identity matrix of order r_j for every $j \in \{1, \ldots, d\}$. Therefore the decomposition formula becomes

$$\mathcal{A} = (U_1, \ldots, U_d)(U_1^H, \ldots, U_d^H)\mathcal{A} = (U_1 U_1^H, \ldots, U_d U_d^H)\mathcal{A}.$$

Thanks to Corollary 2.2.29, the last identity is equivalent to

$$\mathcal{A} = \overline{\pi}_1 \circ \ldots \circ \overline{\pi}_d \mathcal{A}.$$

Definition 2.2.31. Given $d, n_1, \ldots, n_d \in \mathbb{N}$ and a field $\mathbb{K}$ for every $\mathcal{A} \in \mathbb{K}^{n_1} \otimes \ldots \otimes \mathbb{K}^{n_d}$ of multilinear rank $(r_1, \ldots, r_d)$, let $U_i \in \mathbb{M}^{n_i \times r_i}(\mathbb{K})$ be the left orthogonal matrix from the thin SVD of $\mathcal{A}_{(i)}$ flattening for every $i \in \{1, \ldots, d\}$. The *tensor basis of HOSVD* is the set

$$\mathcal{B} = \{u_{i_1}^1 \otimes \ldots \otimes u_{i_d}^d \mid u_{j_i}^i = (U_i)_{(\cdot, j_i)} \text{ for every } j \in \{1, \ldots, r_i\} \text{ for every } i \in \{1, \ldots, d\}\}.$$

The *core tensor* is the tensor $\mathcal{C} \in \mathbb{K}^{n_1} \otimes \ldots \otimes \mathbb{K}^{n_d}$ such that

$$\mathcal{C} = (U_1^H, \ldots, U_d^H)\mathcal{A}.$$

Obviously the HOSVD technique has the proper algorithmic structure. Its pseudo code is the following.

Algorithm 2 HOSVD

Input: a tensor $\mathcal{A} \in \mathcal{V}_1 \otimes \ldots \otimes \mathcal{V}_d$
Output: the HOSVD basis in matrix form $(U_1, \ldots, U_d)$
Output: the HOSVD core tensor $\mathcal{C} \in \mathbb{K}^{n_1} \otimes \ldots \otimes \mathbb{K}^{n_d}$
for $i = 1, 2, \ldots, k$ **do**
$\quad \big|\quad$ Compute SVD of $\mathcal{A}_{(i)}$, i.e. $\mathcal{A}_{(i)} = U_i \Sigma_i V_i^T$;
end
$\mathcal{C} \leftarrow (U_1^H, \ldots, U_d^H)\mathcal{A}$;

We can notice that HOSVD provides a sparse representation of the given tensor, whose costs in terms of storage use can be easily computed. Indeed given $\mathcal{A} \in \mathbb{K}^{n_1} \otimes \ldots \otimes \mathbb{K}^{n_d}$ whose multilinear rank is known to be $(r_1, \ldots, r_d)$, from HOSVD we get that $\mathcal{A} = (U_1, \ldots, U_d)\mathcal{C}$. The storage of each matrix U_i costs $n_i r_i$ memory units for every $i \in \{1, \ldots, d\}$, while storing the core tensor $\mathcal{C}$ needs $\prod_{i=1}^{d} r_i$ memory units. In conclusion the sparse representation of $\mathcal{A}$ costs

$$\sum_{i=1}^{d} n_i r_i + \prod_{i=1}^{d} r_i \text{ memory units.}$$

The error made using orthogonal projection is exactly estimated reminding the projectors properties.

Proposition 2.2.32. *Given $d, n_1, \ldots, n_d \in \mathbb{N}$ and a field $\mathbb{K}$, let $\mathcal{V} = \mathcal{V}_1 \otimes \ldots \mathcal{V}_d$ be a separable tensor subspace of $\mathbb{K}^{n_1} \otimes \ldots \otimes \mathbb{K}^{n_d}$, where $\mathcal{V}_i$ is a subspace of dimension r_i of $\mathbb{K}^{n_i}$ for every $i \in \{1, \ldots, d\}$. Set an orthogonal tensor basis*

$$\mathcal{B} = \{u_{i_1}^1 \otimes \ldots \otimes u_{i_d}^d \mid \text{ for every } j \in \{1, \ldots, r_i\} \text{ for every } i \in \{1, \ldots, d\}\}$$

of $\mathcal{V}$, we define the orthonormal matrix $U_i \in \mathbb{M}^{n_i \times r_i}(\mathbb{K})$ such that $(U_i)_{(\cdot, j_i)} = u_{j_i}^i$ for every $j \in \{1, \ldots, r_i\}$, for every $i \in \{1, \ldots, d\}$. The projector

$$\pi_i : \mathbb{K}^{n_1} \otimes \ldots \otimes \mathbb{K}^{n_d} \mapsto \mathbb{K}^{n_1} \otimes \ldots \otimes \mathbb{K}^{n_d}$$

is such that for every $\mathcal{A} \in \mathbb{K}^{n_1} \otimes \ldots \otimes \mathbb{K}^{n_d}$

$$\pi_i(\mathcal{A}) = (U_i^H U_i) \cdot_i \mathcal{A}$$

for every $i \in \{1, \ldots, d\}$. Then for every $\mathcal{A} \in \mathbb{K}^{n_1} \otimes \ldots \otimes \mathbb{K}^{n_d}$ and for every ρ permutation of d elements, we get that:

$$\|\mathcal{A} - P_{\mathcal{V}}(\mathcal{A})\|^2 = \sum_{i=1}^{d} \big\|\pi_{\rho_{i-1}} \ldots \pi_{\rho_1}(\mathcal{A}) - \pi_{\rho_i} \ldots \pi_{\rho_1}(\mathcal{A})\big\|^2. \tag{2.22}$$

Proof. To shorten the notation, we fix ρ equal to the identity permutation without loss of generality. Firstly chosen a tensor $\mathcal{A} \in \mathbb{K}^{n_1} \otimes \ldots \otimes \mathbb{K}^{n_d}$ we observe that using a telescopic sum the expression $\mathcal{A} - P_{\mathcal{V}}(\mathcal{A})$ can be written as

$$\mathcal{A} - P_{\mathcal{V}}(\mathcal{A}) = (1 - \pi_1 \ldots \pi_d)(\mathcal{A}) = [(1 - \pi_1) + (\pi_1 - \pi_2 \pi_1) + \ldots + (\pi_{d-1} \ldots \pi_1 - \pi_d \ldots \pi_1)](\mathcal{A}). \tag{2.23}$$

thanks to Corollary 2.2.30. Then reminding that $P^\perp = 1 - P$ for any P orthogonal projector, we get that

$$\pi_{i-1} \ldots \pi_1(\mathcal{A}) - \pi_i \ldots \pi_1(\mathcal{A}) = (1 - \pi_i)(\pi_{i-1} \ldots \pi_1(\mathcal{A})) = \pi_i^\perp \pi_{i-1} \ldots \pi_1(\mathcal{A}) \quad (2.24)$$

for every $i \in \{1, \ldots, d\}$. Besides for every $j, i \in \{1, \ldots, d\}$ such that $j > i$, we highlight that

$$\pi_{j-1}...\pi_i...\pi_1(\mathcal{A}) - \pi_j...\pi_i...\pi_1(\mathcal{A}) = (\pi_i \pi_{j-1}...\pi_{i+1}\pi_{i-1}...\pi_1 - \pi_i \pi_j...\pi_{i+1}\pi_{i-1}...\pi_1)(\mathcal{A}). \quad (2.25)$$

thanks to the commutativity of the projectors, stated in 2.2.29. But recalling again the expression of the orthogonal projections, Equation 2.24 becomes

$$\pi_{j-1}...\pi_i...\pi_1(\mathcal{A}) - \pi_j...\pi_i...\pi_1(\mathcal{A}) = \pi_i \pi_j^\perp \pi_{j-1} \ldots \pi_1(\mathcal{A}). \quad (2.26)$$

Now we notice that the quantities in Equation 2.24 and in Equation 2.26 are orthogonal. Lastly we point out that by Equation 2.23 and by definition of norm the term $\|\mathcal{A} - P_V(\mathcal{A})\|^2$ is equivalent to the scalar product between

$$[(1 - \pi_1) + (\pi_1 - \pi_2\pi_1) + .. + (\pi_{d-1}..\pi_1 - \pi_d..\pi_1)](\mathcal{A})$$

and itself. Since for every $i, j \in \{1, \ldots, d\}$ such that $j > i$ the j-th and the i-th term are orthogonal, as highlighted in the previous equations, all the cross products are null, i.e. the thesis. $\qquad\square$

Corollary 2.2.33. *Under Proposition 2.2.32 hypothesis for every $\mathcal{A} \in \mathbb{K}^{n_1} \otimes \ldots \otimes \mathbb{K}^{n_d}$ the error made is upper bounded, i.e.*

$$\|\mathcal{A} - P_V(\mathcal{A})\|^2 \leq \sum_{i=1}^d \left\|\pi_i^\perp(\mathcal{A})\right\|^2.$$

Proof. If we choose the identity permutation in Proposition 2.2.32, fixed $\mathcal{A} \in \mathbb{K}^{n_1} \otimes \ldots \otimes \mathbb{K}^{n_d}$, the i-th term of the summation is

$$\pi_{i-1} \ldots \pi_1(\mathcal{A}) - \pi_i \ldots \pi_1(\mathcal{A}) = (1 - \pi_i)(\pi_{i-1} \ldots \pi_1(\mathcal{A})) = \pi_i^\perp \pi_{i-1} \ldots \pi_1(\mathcal{A})$$

thanks to the relation between an orthogonal projector and its orthogonal. Using the commutativity of projectors stated in 2.2.29, the last equation becomes

$$\pi_{i-1} \ldots \pi_1(\mathcal{A}) - \pi_i \ldots \pi_1(\mathcal{A}) = \pi_{i-1} \ldots \pi_1(\pi_i^\perp(\mathcal{A})).$$

We apply Remark 2.2.24 using as projector $\pi_{i-1} \ldots \pi_1$ and $\pi_i^\perp$, getting

$$\left\|\pi_{i-1} \ldots \pi_1(\mathcal{A}) - \pi_i \ldots \pi_1(\mathcal{A})\right\|^2 = \left\|\pi_{i-1} \ldots \pi_1(\pi_i^\perp(\mathcal{A}))\right\|^2 \leq \left\|\pi_i^\perp(\mathcal{A})\right\|^2.$$

Repeating the same inequality for each term of Proposition 2.2.32 summation we get the thesis. $\qquad\square$

Example 2.2.34. Let $\mathcal{A} \in \mathbb{K}^{n_1} \otimes \mathbb{K}^{n_2} \otimes \mathbb{K}^{n_3}$ be a tensor, which we want to approximate at multilinear rank $(r_1, r_2, r_3) \in \mathbb{N}^3$. Under this choice we can visualize the desired output.
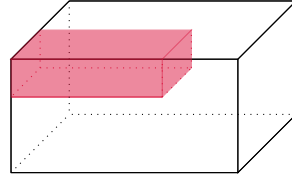

Figure 2.5: A 3-order tensor and its multilinear approximation in light red.

But easily we can decompose the error made, which in the picture is the remaining white part, in three parts as stated in Proposition 2.2.32.
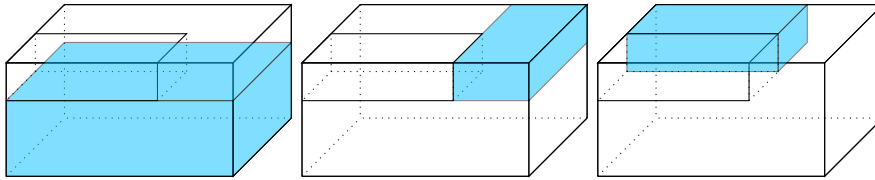
Figure 2.6: The error decomposed in three light blue parts.

The first component, most on the left picture in Figure 2.6 is the difference between the original tensor and the projection of the approximation of $\mathcal{A}$ in the first direction, i.e. $\pi_1^\perp(\mathcal{A})$. The central parallelepiped is the difference from the previous remainder and parallelepiped the projection of the approximation of $\mathcal{A}$ in the second direction, i.e. $\pi_2^\perp \pi_1(\mathcal{A})$. In conclusion the last picture is the difference from the previous reminder and the projection of the approximation of $\mathcal{A}$ in the third direction, i.e. $\pi_3^\perp \pi_2 \pi_1(\mathcal{A})$. The figures highlight what is happening when a projection is taken and subtracted. Moreover visually it is obvious that the order of projection is not significant, i.e. projections commute.

Beside it could be useful for the next steps to visualize what is stated by Corollary 2.2.33.

Figure 2.7: The error decomposed in three green approximated parts.

The first element of the summation is the same and consequently the most left picture of Figure 2.7 is equal to the left one of Figure 2.6, i.e. $\pi_1^\perp(\mathcal{A})$. However the

next picture shows the difference between the original tensor $\mathcal{A}$ and its projection in the second dimension, i.e. $\pi_2^\perp(\mathcal{A})$. Similarly the most right picture displays the remainder of the difference between $\mathcal{A}$ and its projection in the third dimension, i.e. $\pi_3^\perp(\mathcal{A})$. Through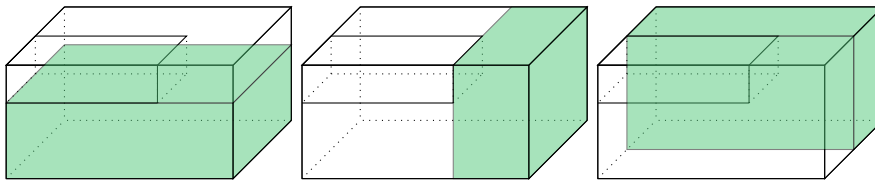 the image is clear that this is an approximation of the exact error, which is smaller. Furthermore it is evident that the size of the object projected through a composition of projections is smaller than the one obtained using just one projection.

Usually, in applications, the multilinear rank of the given tensor is not known, that explains the LMLRA formulation. Consequently working on the exact error is not convenient. Therefore N. Vannieuwenhoven K. Meerbergen and R. Vandebril developed two new strategies to approximate tensors, as reported in [**mate13**], when the multilinear rank is not known. The first is based on the approximation of the upper bound of the error, stated in 2.2.33. As a matter of fact reducing the upper bound implies, thanks to the cited inequality, reducing the the exact error.
The *Truncated Higher Order Singular Value Decomposition* has the SVD as key concept. Fixed $\mathcal{A} \in \mathbb{K}^{n_1} \otimes \ldots \otimes \mathbb{K}^{n_d}$, we consider the i-th term of the upper bound summation, i.e.

$$0 \leq \left\| \pi_i^\perp(\mathcal{A}) \right\|^2 = \left\| \mathcal{A} - (U_i U_i^H) \cdot_i \mathcal{A} \right\|^2$$

by definition. By the positivity of each term of the previous expression, minimizing the upper bound means minimizing each term of the upper bound summation, i.e. taking the minimum of the norm of the difference between $\mathcal{A}$ and its projection in just one direction each time. Because we approximate the tensor only with respect to one direction each time, the minimization problem is mathematically

$$\arg \min_{\pi_i \text{ projection into } \mathscr{V}_i} \left\| \pi_i^\perp(\mathcal{A}) \right\|^2 = \arg \min_{U_i \in \mathbb{O}(n_i \times r_i)} \left\| \mathcal{A}_{(i)} - (U_i U_i^H)\mathcal{A}_{(i)} \right\|^2,$$

i.e. we research the best approximation at rank r_i for the i-th flattening of $\mathcal{A}$ for every $i \in \{1, \ldots, d\}$. However thanks to Theorem 2.1.19 the previous problem has a close solution which is obtained through SVD of the i-th flattening of $\mathcal{A}$, truncated at the r_i singular values for every $i \in \{1, \ldots, d\}$.

Definition 2.2.35. Given $d, n_1, \ldots, n_d \in \mathbb{N}$, a field $\mathbb{K}$ and and a target multilinear rank $(r_1, \ldots, r_d)$, for every $\mathcal{A} \in \mathbb{K}^{n_1} \otimes \ldots \otimes \mathbb{K}^{n_d}$, let $U_i \in \mathbb{M}^{n_i \times r_i}(\mathbb{K})$ be the left orthogonal matrix from the thin SVD of $\mathcal{A}_i$ flattening for every $i \in \{1, \ldots, d\}$, the *tensor basis of T-HOSVD* is the set

$$\mathscr{B} = \{u_{i_1}^1 \otimes \ldots \otimes u_{i_d}^d \mid u_{j_i}^i = (\overline{U}_i)_{(\cdot, j_i)} \text{ for every } j \in \{1, \ldots, r_i\} \text{ for every } i \in \{1, \ldots, d\}\}.$$

The *T-HOSVD core tensor* is the tensor $\mathcal{C} \in \mathbb{K}^{n_1} \otimes \ldots \otimes \mathbb{K}^{n_d}$ such that

$$\mathcal{C} = (\overline{U}_1^H, \ldots, \overline{U}_d^H)\mathcal{A}.$$

Again we can rewrite this technique in an algorithm form as follows.

Algorithm 3 T-HOSVD

Input: a tensor $\mathcal{A} \in \mathbb{K}^{n_1} \otimes \ldots \otimes \mathbb{K}^{n_d}$
Input: a target multilinear rank $(r_1, \ldots, r_d)$
Output: the T-HOSVD basis in matrix form $(\overline{U}_1, \ldots, \overline{U}_d)$
Output: the T-HOSVD core tensor $\mathcal{C} \in \mathbb{K}^{n_1} \otimes \ldots \otimes \mathbb{K}^{n_d}$
for $i = 1, 2, \ldots, d$ **do**
  Compute SVD of $\mathcal{A}_{(i)}$, i.e. $\mathcal{A}_{(i)} = U_i \Sigma_i V_i^T$;
  Store in $\overline{U}_i$ the first r_i columns of U_i
end
$\mathcal{C} \leftarrow (\overline{U}_1^H, \ldots, \overline{U}_d^H)\mathcal{A}$;

The amount of storage memory used is exactly the same as in the HOSVD. However we can prove the relation between the error made with the T-HOSVD approximation and the best $(r_1, \ldots, r_d)$ approximation, if it exists.

Proposition 2.2.36. *Given $d, n_1, \ldots, n_d \in \mathbb{N}$ and a field $\mathbb{K}$, let $\mathcal{V}_i$ be a subspace of dimension $r_i \in \mathbb{N}$ of $\mathbb{K}^{n_i}$ for every $i \in \{1, \ldots, d\}$. Let $\mathcal{A} \in \mathbb{K}^{n_1} \otimes \ldots \otimes \mathbb{K}^{n_d}$ be a tensor such that there exists $\mathcal{A}^* \in \mathbb{K}^{n_1} \otimes \ldots \otimes \mathbb{K}^{n_d}$ the best multilinear rank $(r_1, \ldots, r_d)$ approximation, i.e.*

$$\mathcal{A}^* = \arg \min_{mlrank(\mathcal{T}) \leq (r_1, \ldots, r_d)} \|\mathcal{A} - \mathcal{T}\|.$$

If the T-HOSVD approximation tensor is $\mathcal{A}_T \in \mathbb{K}^{n_1} \otimes \ldots \otimes \mathbb{K}^{n_d}$ of multilinear rank $(r_1, \ldots, r_d)$, then we have that:

$$\|\mathcal{A} - \mathcal{A}_T\| \leq \sqrt{d}\|\mathcal{A} - \mathcal{A}^*\|.$$

Proof. Thanks to Corollary 2.2.33, we know that:

$$\left\|\mathcal{A} - \mathcal{A}_T\right\|^2 \leq \sum_{i=1}^{d} \left\|\pi_{T_i}^{\perp}(\mathcal{A})\right\|^2$$

where by the structure of the T-HOSVD the i-th term of the summation is

$$\left\|\pi_{T_i}^{\perp}(\mathcal{A})\right\|^2 = \left\|\mathcal{A} - (\overline{U}_i \overline{U}_i^H)_{\cdot_i} \mathcal{A}\right\|^2 = \left\|\mathcal{A}_{(i)} - (\overline{U}_i \overline{U}_i^H)\mathcal{A}_{(i)}\right\|^2.$$

But Corollary 2.1.20 states that the best rank r approximation for every $A \in \mathbb{M}^{m \times n}(\mathbb{K})$ matrix is $\sum_{i=1}^{r} U S^{(i)} V^H$, with the usual meaning for each matrix. Using the associativity of the matrix product, we get that

$$\sum_{i=1}^{r} U S^{(i)} V^H = U(\sum_{i=1}^{r} S^{(i)}) V^H.$$

Noticing that the $(r+k)$-th column of $S^{(i)}$ is the null vector for every $d \in \{1, \ldots, n\}$, we can conclude that

$$\sum_{i=1}^{r} US^{(i)}V^H = U(\sum_{i=1}^{r} S^{(i)})V^H = \overline{U}(\sum_{i=1}^{r} \overline{S}^{(i)}V^H$$

where $\overline{U}$ is the matrix U truncated at the r_i-th column, the matrix $\overline{S}^{(i)}$ is the matrix $S^{(i)}$ truncated at the r-th row. Applying the generalised SVD to the i-th flattening, i.e.

$$\mathcal{A}_{(i)} = U_i \Sigma_i V_i^H$$

and assuming that $(U_i)_{(\cdot,d)} = u_d$ for every $d \in \{1, \ldots, m\}$ we notice that

$$\overline{U}_i^H U_i = \begin{bmatrix} u_1^H \\ \vdots \\ u_r^H \end{bmatrix} \begin{bmatrix} u_1 & \ldots & u_r & \ldots & u_m \end{bmatrix} = \left[\begin{array}{c|c} I_{r_i} & 0 \\ \hline 0 & 0 \end{array} \right]$$

by the orthogonality. Moreover we can notice that

$$\overline{U}_i \overline{U}_i^H \mathcal{A}_{(i)} = \overline{U}_i \overline{U}_i^H U_i \Sigma_i V_i^H = \overline{U}_i \overline{\Sigma}_i V_i^H,$$

where $\overline{\Sigma}_i$ is the matrix Σ_i truncated at the r_i-th row for every $i \in \{1, \ldots, d\}$. But obviously $\overline{\Sigma}_i = \sum_{j=1}^{r_i} \overline{S}_i^{(j)}$, where $S_i^{(j)}$ is defined as in Corollary 2.1.20 for every $j \in \{1, \ldots, r_i\}$ and consequently

$$\overline{U}_i \overline{U}_i^H \mathcal{A}_{(i)} = \overline{U}_i \overline{\Sigma}_i V_i^H = \overline{U}(\sum_{i=1}^{r_i} \overline{S}^{(i)})V^H,$$

i.e.

$$\left\| \mathcal{A}_{(i)} - (\overline{U}_i \overline{U}_i^H)\mathcal{A}_{(i)} \right\|^2 = \min_{\substack{B \in \mathbb{M}^{n_i \times N}(\mathbb{K}) \\ \text{rank}(B)=r_i}} \left\| \mathcal{A}_{(i)} - B \right\|^2 \quad \text{with } N = \prod_{\substack{j=1 \\ j \neq i}}^{d} n_j$$

for every $i \in \{1, \ldots, d\}$. By the last equation we have that

$$\left\| \pi_{T_i}^{\perp}(\mathcal{A}) \right\|^2 = \left\| \mathcal{A}_{(i)} - (\overline{U}_i \overline{U}_i^H)\mathcal{A}_{(i)} \right\|^2 \leq \left\| \mathcal{A}_{(i)} - \mathcal{A}_{(i)}^* \right\|^2$$

for every $i \in \{1, \ldots, d\}$ and consequently

$$\left\| \mathcal{A} - \mathcal{A}_T \right\|^2 \leq \sum_{i=1}^{d} \left\| \pi_{T_i}^{\perp}(\mathcal{A}) \right\|^2 \leq \sum_{i=1}^{d} \left\| \mathcal{A}_{(i)} - \mathcal{A}_{(i)}^* \right\|^2 = d \left\| \mathcal{A}_{(i)} - \mathcal{A}_{(i)}^* \right\|^2.$$

$\square$

Remark 2.2.37. *Since we are interested in application of T-HOSVD algorithm, we take*

$$\mathcal{A}^* = \arg \inf_{mlrank(\mathcal{T}) \leq (r_1, \ldots, r_d)} \|\mathcal{A} - \mathcal{T}\|$$

for every $\mathcal{A} \in \mathbb{K}^{n_1} \otimes \ldots \otimes \mathbb{K}^{n_d}$, *neglecting to check the existence of the best approximation.*

We have seen that the order of projectors application is not significant for the T-HOSVD, but for the ST-HOSVD this is not true anymore. However in order to simplify the notation we will follow the natural increasing order in applying the projections in the ST-HOSVD, i.e. we will project firstly in the 1st direction, than in the 2nd and so on, until the d-th direction. The idea of the *Sequentially Truncated High Order Singular Value* is minimizing each term of the summation of Proposition 2.2.32. Let $\mathcal{A} \in \mathbb{K}^{n_1} \otimes \ldots \otimes \mathbb{K}^{n_d}$ be a chosen tensor and $(r_1, \ldots, r_d)$ the target multilinear rank of approximation, the first step is searching the projector which minimises the first error term of Equation 2.22 is

$$\arg \min_{\pi_1} \left\| \pi_1^{\perp}(\mathcal{A}) \right\|^2 = \arg \min_{U_1 \in \mathbb{O}(n_1 \times r_1)} \left\| (U_1 U_1^H)_{\cdot_1} \mathcal{A} \right\|^2,$$

i.e.

$$\arg \min_{U_1 \in \mathbb{O}(n_1 \times r_1)} \left\| (U_1 U_1^H) \mathcal{A}_{(1)} \right\|^2.$$

The last formulation of this first step has a close solution, thanks to Theorem 2.1.19. We are looking for the best rank r_1 approximation of the 1-st flattening. So we can conclude that the matrix U_1 obtained from the SVD of the first flattening of $\mathcal{A}$ truncated at the r_1-th column is such that

$$\hat{U}_1 = \arg \min_{U_1 \in \mathbb{O}(n_1 \times r_1)} \left\| (U_1 U_1^H) \mathcal{A}_{(1)} \right\|^2.$$

We defined the core tensor of the first step,

$$\mathcal{C}^{(1)} = (\hat{U}_1^H, I, \ldots, I)\mathcal{A}.$$

Fixed the $\hat{\pi}_1 = \hat{U}_1 \hat{U}_1^H$, the next step is looking for the projector which minimises the second error term of Equation 2.22, i.e.

$$\arg \min_{\pi_2} \left\| \pi_2^{\perp} \hat{\pi}_1(\mathcal{A}) \right\|^2 = \arg \min_{U_2 \in \mathbb{O}(n_2 \times r_2)} \left\| (U_2 U_2^H)_{\cdot_2} (\hat{U}_1 \hat{U}_1^H)_{\cdot_1} \mathcal{A} \right\|^2.$$

But thanks to multilinearity, we can rewrite the last equation as

$$\arg \min_{U_2 \in \mathbb{O}(n_2 \times r_2)} \left\| (\hat{U}_1)_{\cdot_1} (U_2 U_2^H)_{\cdot_2} (\hat{U}_1^H)_{\cdot_1} \mathcal{A} \right\|^2$$

i.e.

$$\arg \min_{U_2 \in \mathbb{O}(n_2 \times r_2)} \left\| (\hat{U}_1)_{\cdot_1} (U_2 U_2^H)_{\cdot_2} \mathcal{C}^{(1)} \right\|^2.$$

Since $\hat{U}_1$ is fixed, the second step problem becomes

$$\arg \min_{U_2 \in \mathbb{O}(n_2 \times r_2)} \left\| (U_2 U_2^H)_{\cdot 2} \mathcal{C}^{(1)} \right\|^2 = \arg \min_{U_2 \in \mathbb{O}(n_2 \times r_2)} \left\| (U_2 U_2^H)_{\cdot 2} \mathcal{C}^{(1)}_{(2)} \right\|^2$$

whose solution is the matrix $\hat{U}_2$ obtained through the SVD of the 2-nd flattening of $\mathcal{C}^{(1)}$ truncated at the r_2-th column. Defined the new core tensor

$$\mathcal{C}^{(2)} = (\hat{U}_1^H, \hat{U}_2^H, I, \ldots, I)\mathcal{A},$$

we proceed similarly with the next ones. At the $(d-1)$-th step we have defined the $(d-1)$-th core tensor

$$\mathcal{C}^{(d-1)} = (\hat{U}_1^H, \hat{U}_2^H, \ldots, \hat{U}_{d-1}^H, I)\mathcal{A}.$$

For the last d-th step, we look for the projectors which minimises the last error term of Equation 2.22, i.e.

$$\arg \min_{\pi_d} \left\| \pi_d^\perp \hat{\pi}_{d-1} \ldots \hat{\pi}_1(\mathcal{A}) \right\|^2,$$

which using the multilinearity and the projectors properties becomes

$$\arg \min_{U_d \in \mathbb{O}(n_d \times r_d)} \left\| (U_d U_d^H)_{\cdot d} (\hat{U}_1 \hat{U}_1^H, .., \hat{U}_{d-1} \hat{U}_{d-1}^H)_{\cdot 1, .., d-1} \mathcal{A} \right\|^2$$

i.e.

$$\arg \min_{U_d \in \mathbb{O}(n_d \times r_d)} \left\| (U_d U_d^H) \mathcal{C}^{(d-1)}_{(k)} \right\|^2.$$

The solution of the last problem is the matrix $\hat{U}_d$ obtained through the SVD of the d-th flattening of the core tensor $\mathcal{C}^{(d-1)}$ truncated at the r_d-th column. We can now define the central structures of ST-HOSVD.

Definition 2.2.38. Given $d, n_1, \ldots, n_d \in \mathbb{N}$ and a field $\mathbb{K}$, we fix the natural increasing processing order from 1 up to d. For every $\mathcal{A} \in \mathbb{K}^{n_1} \otimes \ldots \otimes \mathbb{K}^{n_d}$ and a target multilinear rank $(r_1, \ldots, r_d)$, if i is equal to 0, the i-th *partially truncated core tensor* is

$$\mathcal{C}^{(0)} = \mathcal{A}$$

while if $i \in \{1, \ldots, d\}$, it is

$$\mathcal{C}^{(i)} = (\hat{U}_1^H, \hat{U}_2^H, \ldots, \hat{U}_{i-1}^H, I, \ldots, I)\mathcal{A}$$

where $\hat{U}_j \in \mathbb{M}^{n_j \times r_j}$ is the left orthogonal matrix from the thin SVD of the j-th flattening of the core tensor $\mathcal{C}^{(j-1)}$ for every $j \in \{1, \ldots, i\}$. The *ST-HOSVD core tensor* is

$$\mathcal{C}_{ST} = \mathcal{C}^{(k)}.$$

With the same meaning for $\hat{U}_j$ for every $j \in \{1, \ldots, d\}$, the *tensor basis of ST-HOSVD* is

$$\mathcal{B} = \{u^1_{i_1} \otimes \ldots \otimes u^d_{i_d} \mid u^i_{j_i} = (\hat{U}_i)_{(\cdot, j_i)} \text{ for every } j \in \{1, \ldots, r_i\} \text{ for every } i \in \{1, \ldots, d\}\}.$$

For every $i \in \{1, \ldots, d\}$, the *i-th ST-HOSVD partial approximation* of $\mathcal{A}$ is

$$\mathcal{A}^{(i)} = (\hat{U}_1 \hat{U}_1^H, \ldots, \hat{U}_i \hat{U}_i^H, I, \ldots, I)\mathcal{A}.$$

The *ST-HOSVD approximation* of $\mathcal{A}$ is

$$\mathcal{A}_{ST} = \mathcal{A}^{(k)}.$$

Also in this case we write the decomposition techniques in an algorithmic form.

Algorithm 4 ST-HOSVD

Input: a tensor $\mathcal{A} \in \mathbb{K}^{n_1} \otimes \ldots \otimes \mathbb{K}^{n_d}$
Input: a target multilinear rank $(r_1, \ldots, r_d)$
Output: the ST-HOSVD basis in matrix form $\hat{U}_1, \ldots, \hat{U}_d$
Output: the ST-HOSVD core tensor $\mathcal{C} \in \mathbb{K}^{n_1} \otimes \ldots \otimes \mathbb{K}^{n_d}$
$\mathcal{C} \leftarrow \mathcal{A}$;
  for $i = 1, 2, \ldots, d$ **do**
    Compute SVD of $\mathcal{C}_{(i)}$, i.e. $\mathcal{C}_{(i)} = U_i \Sigma_i V_i^T$;
    Store in $\hat{U}_i$ the first r_i columns of U_i;
    $\mathcal{C} \leftarrow (\hat{U}_i^H)_{\cdot i} \mathcal{C}$;
end

Moreover we can describe the relation between the ST-HOSVD error and the one made with the best $(r_1, \ldots, r_d)$ approximation, if it exists.

Proposition 2.2.39. *Given $d, n_1, \ldots, n_d \in \mathbb{N}$ and a field $\mathbb{K}$, let $\mathcal{V}_i$ be a subspace of dimension $r_i \in \mathbb{N}$ of $\mathbb{K}^{n_i}$ for every $i \in \{1, \ldots, d\}$. Let $\mathcal{A} \in \mathbb{K}^{n_1} \otimes \ldots \otimes \mathbb{K}^{n_d}$ be a tensor such that there exists $\mathcal{A}^* \in \mathbb{K}^{n_1} \otimes \ldots \otimes \mathbb{K}^{n_d}$ the best multilinear rank $(r_1, \ldots, r_d)$ approximation, i.e.*

$$\mathcal{A}^* = \arg \min_{mlrank(\mathcal{T}) \leq (r_1, \ldots, r_d)} \|\mathcal{A} - \mathcal{T}\|.$$

If the ST-HOSVD approximation tensor is $\mathcal{A}_{ST} \in \mathbb{K}^{n_1} \otimes \ldots \otimes \mathbb{K}^{n_d}$ of multilinear rank $(r_1, \ldots, r_d)$, then:

$$\|\mathcal{A} - \mathcal{A}_{ST}\| \leq \sqrt{d} \|\mathcal{A} - \mathcal{A}^*\|.$$

Proof. By Proposition 2.2.32 the error of approximation by the ST-HOSVD is

$$\|\mathcal{A} - \mathcal{A}_{ST}\|^2 = \sum_{i=1}^{d} \|\hat{\pi}_{i-1} \ldots \hat{\pi}_1(\mathcal{A}) - \hat{\pi}_i \ldots \hat{\pi}_1(\mathcal{A})\|^2 = \sum_{i=1}^{d} \left\| \hat{\pi}_i^{\perp}(\hat{\pi}_{i-1} \ldots \hat{\pi}_1)(\mathcal{A}) \right\|^2$$

where $\hat{\pi}_i = \hat{U}_i \hat{U}_i^H$ as defined in the ST-HOSVD for every $i \in \{1, \ldots, d\}$. Since the construction of the ST-HOSVD, we notice that

$$\hat{\pi}_i = \arg\min_{\pi_i} \left\| \hat{\pi}_i^\perp (\hat{\pi}_{i-1} \ldots \hat{\pi}_1)(\mathcal{A}) \right\|^2.$$

Consequently if we denote with π_i the projections such that

$$\mathcal{A}^* = (\pi_d \ldots \pi_1)(\mathcal{A})$$

we have

$$\left\| \hat{\pi}_i^\perp (\hat{\pi}_{i-1} \ldots \hat{\pi}_1)(\mathcal{A}) \right\|^2 \leq \left\| \pi_i^\perp (\hat{\pi}_{i-1} \ldots \hat{\pi}_1)(\mathcal{A}) \right\|^2,$$

for every $i \in \{1, \ldots, d\}$. By Remark 2.2.24, we get

$$\left\| \hat{\pi}_i^\perp (\hat{\pi}_{i-1} \ldots \hat{\pi}_1)(\mathcal{A}) \right\|^2 \leq \left\| \pi_i^\perp (\hat{\pi}_{i-1} \ldots \hat{\pi}_1)(\mathcal{A}) \right\|^2 \leq \left\| \pi_i^\perp (\mathcal{A}) \right\|^2.$$

Moreover thanks to Proposition 2.2.32, we have that

$$\left\| \pi_i^\perp (\mathcal{A}) \right\|^2 \leq \left\| \mathcal{A} - \mathcal{A}^* \right\|^2$$

since for every $i \in \{1, \ldots, d\}$ we can choose a ρ permutation of d elements ρ such that $\rho(i) = 1$. So we can conclude that

$$\left\| \mathcal{A} - \mathcal{A}_{ST} \right\|^2 \leq \sum_{i=1}^d \left\| \hat{\pi}_i^\perp (\hat{\pi}_{i-1} \ldots \hat{\pi}_1)(\mathcal{A}) \right\|^2 \leq \sum_{i=1}^d \left\| \pi_i^\perp (\mathcal{A}) \right\|^2 \leq d \left\| \mathcal{A} - \mathcal{A}^* \right\|^2.$$

$\square$

Remark 2.2.40. *Similarly to T-HOSVD, we are interested in application of ST-HOSVD algorithm. So we take*

$$\mathcal{A}^* = \arg \inf_{mlrank(\mathcal{T}) \leq (r_1, \ldots, r_d)} \| \mathcal{A} - \mathcal{T} \|$$

for every $\mathcal{A} \in \mathbb{K}^{n_1} \otimes \ldots \otimes \mathbb{K}^{n_d}$, neglecting to check the existence of the best approximation.

We have stated that the order of proceeding in ST-HOSVD is not negligible for the approximation and the following example will confirm it.

Example 2.2.41. Let $\mathcal{A} \in \mathbb{R}^3 \otimes \mathbb{R}^3 \otimes \mathbb{R}^3$ be a tensor of order three, such that:

$$\mathcal{A}_{(\cdot, \cdot, 1)} = \begin{bmatrix} 2 & 4 & 7 \\ 5 & 6 & 3 \\ 9 & 3 & 5 \end{bmatrix} \quad \mathcal{A}_{(\cdot, \cdot, 2)} = \begin{bmatrix} 7 & 5 & 3 \\ 9 & 2 & 8 \\ 9 & 2 & 3 \end{bmatrix} \quad \mathcal{A}_{(\cdot, \cdot, 2)} = \begin{bmatrix} 8 & 4 & 6 \\ 3 & 2 & 5 \\ 9 & 3 & 4 \end{bmatrix}.$$

Fixed the target multilinear rank $(2, 2, 2)$, we approximate $\mathcal{A}$ to this multilinear rank using ST-HOSVD and every ρ permutation of 3 elements. Denoting the approximation with the hat symbol, the results are

$$\left\|\mathcal{A} - \hat{\mathcal{A}}_{[1,2,3]}\right\| = 8.1912; \qquad \left\|\mathcal{A} - \hat{\mathcal{A}}_{[1,3,2]}\right\| = 8.1932;$$

$$\left\|\mathcal{A} - \hat{\mathcal{A}}_{[2,1,3]}\right\| = 7.4799; \qquad \left\|\mathcal{A} - \hat{\mathcal{A}}_{[2,3,1]}\right\| = 7.4497;$$

$$\left\|\mathcal{A} - \hat{\mathcal{A}}_{[3,1,2]}\right\| = 7.5001; \qquad \left\|\mathcal{A} - \hat{\mathcal{A}}_{[3,2,1]}\right\| = 7.4835.$$

Besides we have said that T-HOSVD tries to minimize the upper bound while ST-HOSVD acts on the exact error. We are naturally induced to think that the approximation obtained with the sequential truncated strategy is better with respect to the truncated one in term of error reduction. However a simple example will show that this is not always true.

Example 2.2.42. Let $\mathcal{A} \in \mathbb{R}^2 \otimes \mathbb{R}^2 \otimes \mathbb{R}^2 \otimes \mathbb{R}^2$ be a tensor of order four, such that:

$$\mathcal{A}_{(\cdot,\cdot,1,1)} = \begin{bmatrix} 0.5 & -1.7 \\ -1.3 & -0.6 \end{bmatrix} \qquad \mathcal{A}_{(\cdot,\cdot,2,1)} = \begin{bmatrix} -2.4 & -0.1 \\ -0.7 & 1.4 \end{bmatrix}$$

$$\mathcal{A}_{(\cdot,\cdot,1,2)} = \begin{bmatrix} 0.1 & 0.1 \\ 2.2 & -0.8 \end{bmatrix} \qquad \mathcal{A}_{(\cdot,\cdot,2,2)} = \begin{bmatrix} -0.3 & -2.5 \\ 0 & 0.3 \end{bmatrix}.$$

The target rank is $(1, 1, 1, 1)$ and the best T-HOSVD approximation, denoted with the overline symbol entails the error

$$\left\|\mathcal{A} - \overline{\mathcal{A}}\right\| = 18.68700.$$

On the other hand the best ST-HOSVD approximation is realized by the proceeding order $p^* = \{1, 3, 2, 4\}$. Denoting the ST-HOSVD error with the hat symbol, the associated approximation error is

$$\left\|\mathcal{A} - \hat{\mathcal{A}}_{p^*}\right\| = 18.90896,$$

which is greater than the previous one.

Even if this result can discourage us from searching a relation between T-HOSVD and ST-HOSVD approximation error, for a special case there is a proposition.

Proposition 2.2.43. *Given $n_1, \ldots, n_3, r_1, r_2 \in \mathbb{N}$ and the real field $\mathbb{R}$, let $\mathcal{A} \in \mathbb{R}^{n_1} \otimes \mathbb{R}^{n_2} \otimes \mathbb{R}^{n_3}$ be a three order tensor. Fixed the target multilinear rank $(1, r_1, r_2)$ and a processing order $p^* = \{1, 3, 2, 4\}$, we denote with $\overline{\mathcal{A}}_T$ the T-HOSVD approximation and with $\hat{\mathcal{A}}$ the ST-HOSVD with proceeding order p^* approximation. Then*

$$\left\|\mathcal{A} - \hat{\mathcal{A}}\right\| \leq \left\|\mathcal{A} - \overline{\mathcal{A}}\right\|$$

Proof. We will denote with the hat symbol the projectors obtained with ST-HOSVD and with the overline one the projectors from the T-HOSVD. As we have already pointed out previously $\hat{\pi}_1 = \overline{\pi}_1$ because the first step of T-HOSVD and ST-HOSVD is solving

$$\arg\min_{\pi_1} \left\| \pi_1^\perp(\mathcal{A}) \right\|.$$

The partial truncated core tensor $\mathcal{C}^{(1)} = \hat{\pi}_1(\mathcal{A}) = \overline{\pi}_1(\mathcal{A})$ is a matrix. So in the next step of ST-HOSVD we define the second projector $\hat{\pi}_2 = \hat{U}_2\hat{U}_2^H$ taking the first r_1 vectors of the left orthogonal matrix U from SVD of $\hat{\pi}_1(\mathcal{A})$. Comparing term by term the first two elements of the summation of the error Equation 2.22 we get that $\left\| \hat{\pi}_1^\perp(\mathcal{A}) \right\|^2 = \left\| \overline{\pi}_1^\perp(\mathcal{A}) \right\|^2$ and since Theorem 2.1.19 $\left\| \hat{\pi}_2^\perp\hat{\pi}_1(\mathcal{A}) \right\|^2 \leq \left\| \overline{\pi}_2^\perp\overline{\pi}_1(\mathcal{A}) \right\|^2$. In the third and last step we define $\hat{\pi}_3 = \hat{U}_3\hat{U}_3^H$ taking the first r_2 vectors of the left orthogonal matrix U from SVD of $\hat{\pi}_2\hat{\pi}_1(\mathcal{A})$. However since $\hat{\pi}_1(\mathcal{A})$ is a matrix, the r_2 column vectors which compose the matrix $\hat{U}_3$ are the the first r_2 vectors of the right orthogonal matrix V from the SVD of $\hat{\pi}_1(\mathcal{A})$. Consequently $\hat{\pi}_3\hat{\pi}_2\hat{\pi}_1(\mathcal{A})$ is the best rank $\min\{r_1, r_2\}$ approximation of $\hat{\pi}_1(\mathcal{A})$ and so

$$\left\| \hat{\pi}_3^\perp\hat{\pi}_2\hat{\pi}_1(\mathcal{A}) \right\|^2 \leq \left\| \overline{\pi}_3^\perp\overline{\pi}_2\overline{\pi}_1(\mathcal{A}) \right\|^2.$$

Collecting the previous equality and inequality we get the thesis, since Proposition 2.2.32. □

Besides the presented results, mathematicians, physics, engineers and computer scientists are still studying solutions to the LMLRA problem. In 2018 Jan Draisma, Giorgio Ottaviani and Alicia Tocino made a step forward. They presented a generalisation of Schmidt, Mirsky, Eckhart, Young theorem for tensors, cf. [**mate14**]. Fixed a tensor in a Euclidean tensor space, they look for the critical points of the distance function from the chosen tensor to the set of tensors of rank at most k. They prove that for 2-order tensors critical rank-at-most-1 matrices correspond to the singular pairs of the chosen order 2 matrix. Moreover they show that critical rank-1 tensors form a linear subspace, which, under some conditions, contains also the critical rank-at-most-k.

Chapter 3

Loss of volume, loss of information

In this last chapter we will present the chosen operative way and the results obtained.

3.1 Datasets and codes

Spectral images represent one of the possible applications of tensor theory. We can store in a three order tensor rasters corresponding to different bands. Therefore the first step of our discussion is collecting the data. From MOD13C2v006 and MOD13A3v006, both sensed by MODIS, 1.1.1, and available at [**res0**; **res00**], we select the RED, NIR rasters and the NASA computed NDVI, subsection 1.2.1.

MOD13C2v006 is a product characterized by 13 layers, each of which stores an imagery of the Earth with different properties. Each raster is a matrix of 3600 rows and 7200 columns. The side of each pixel corresponds to 5600 m, which is the spatial resolution. The imageries are monthly: they are obtained from the daily data through NASA's algorithms. We download the data in `hdf` format from January 2010 until December 2018.

MOD13A3v006 is a similar product with a higher spatial resolution. While each image from the previous dataset covered the entire Earth's surface, each one from this second dataset covers just $1200 \times 1200\,\mathrm{m}^2$. We select the 20 components, called *granules*, to compose an Europe's map. We download in `GEOTiff` the granules from RED, NIR bands and from the NDVI computed by the NASA, from June 2011 until December 2018. Also in this case they are obtained by the NASA scientists from daily data. The final dimension of most of the rasters is 4800 rows and 6000 columns. We do not talk about every used test elements from MOD13A3v006 dataset, since those of December 2016 and of December 2017 do not include all the 20 granules. Moreover we remark that rasters of December 2012 and December 2015 have the correct dimension, but they have respectively one and two missing area. Lastly NASA stores the data in 16-bit signed integers.

The programming language used to perform all the computations is `python`, since we want to follow the free and open source NASA's policy. The next step is creating

3-order tensors. Taking advantage of the GDAL dependencies for python, we simply convert rasters into matrix, removing the metadata useless for our aims. Then we store NDVI, RED and NIR into 3-order arrays.

Definition 3.1.1. Let $\mathcal{T}_E$ be the set of all the tensor $\mathcal{A} \in \mathbb{R}^{n_1} \otimes \mathbb{R}^{n_2} \otimes \mathbb{R}^3$ with $n_1 = 4800$ and $n_2 = 6000$ such that

- $\mathcal{A}_{\cdot,\cdot,1}$ is the NDVI raster,

- $\mathcal{A}_{\cdot,\cdot,2}$ is the RED raster band,

- $\mathcal{A}_{\cdot,\cdot,3}$ is the NIR raster band

of Europe dataset for every month and for every year. The cardinality of $\mathcal{T}_E$ is $n_E = 91$.

Similarly let $\mathcal{T}_W$ be the set of all the tensor $\mathcal{A} \in \mathbb{R}^{m_1} \otimes \mathbb{R}^{m_2} \otimes \mathbb{R}^3$ with $m_1 = 3600$ and $m_2 = 7200$ such that

- $\mathcal{A}_{\cdot,\cdot,1}$ is the NDVI raster,

- $\mathcal{A}_{\cdot,\cdot,2}$ is the RED raster band,

- $\mathcal{A}_{\cdot,\cdot,3}$ is the NIR raster band

of Earth dataset for every month and for every year. The cardinality of $\mathcal{T}_W$ is $n_W = 108$.

Remark 3.1.2. *Henceforth we will denote with $\mathbb{R}^{\otimes E}$ the tensor space $\mathbb{R}^{n_1} \otimes \mathbb{R}^{n_2} \otimes \mathbb{R}^3$, where $E = (n_1, n_2, 3)$, $n_1 = 4800$ and $n_2 = 6000$ and with $\mathbb{R}^{\otimes W}$ the tensor space $\mathbb{R}^{m_1} \otimes \mathbb{R}^{m_2} \otimes \mathbb{R}^3$, where $W = (m_1, m_2, 3)$, $m_1 = 3600$ and $m_2 = 7200$.*

Lastly we code the high order singular value decomposition and the indexes computation algorithms. To implement T-HOSVD and ST-HOSVD we need a python library expressly developed to manage tensors. Moreover we also want that this library to interact properly with `NumPy` the python most used library for numerical computation, cf. [**res1**]. Therefore we choose `TensorLy`, cf. [**res2**]. This recently developed library, firstly presented in 2016, wants to make tensor study and manipulation easy and accessible. Besides their creators projected `TensorLy` to perfectly match with other famous `Python` libraries, as `NumPy`, `SciPy`. They developed most of the main tensor operations and related functions. To compute the singular value decomposition of tensors flattening, the central step in T-HOSVD and ST-HOSVD algorithm, we need a `Python` function able to manage huge arrays. This is not a trivial task, since development of `Python` functions is left to singular initiative. We decide to use `svds` function from `SciPy` sparse linear algebra function, cf. [**res3**]. This implementation of SVD takes advantage of matrix sparsity in performing the matrix-vector multiplication. The final implementation of T-HOSVD is

```python
import numpy as np
import tensorly as tl
from tensorly import decomposition as decompose
from tensorly import tenalg as Tla
import scipy
from scipy.sparse.linalg import svds

def T_hosvd(T, rango, projector = Ture):
  L = []
  dim = T.shape
  for i in range(len(dim)):
    flat = tl.unfold(T,i)
    res = svds(flat, k = rango[i])
    L.append(np.transpose(res[0][0:dim[i], :]))
  core = Tla.multi_mode_dot(T,L)
  if projector:
    P = [np.transpose(u) for u in L]
    return [core,P]
  return core
```

This function takes as input variable a tensor `T` and a list or a tuple, whose values are the target multilinear rank components. Inside the `T_hosvd` we declare an empty list, `L`, in which at the i-th step we store matrix `U_i` from the thin SVD of the i-th flattening for every $i \in \{1, \ldots, d\}$. The `dim` variable is a tuple containing the size of tensor `T`. Then for each direction we compute the flattening and its SVD. After the for loop, we get the core tensor with the multilinear product between a list of matrices, our projectors, and the original tensor `T`. Notice that there is the option `projector`, with `True` as default value, to return a list `L` containing the projectors matrices together with the standard result, i.e. the core tensor.
On the other hand the ST-HOSVD implementation is

```python
import numpy as np
import tensorly as tl
from tensorly import decomposition as decompose
from tensorly import tenalg as Tla
import scipy
from scipy.sparse.linalg import svds

def ST_hosvd(T, rango, projector = True):
  dim = T.shape
  core = T
  if projector:
    P=[]
  for i in range(len(dim)):
```

```
      flat = tl.unfold(core,i)
      res = svds(flat, k=rango[i])
      core =
  ↪    Tla.mode_dot(core,np.transpose(res[0][0:dim[i],:]),mode=i)
      if projector:
        P.append(res[0][0:dim[i],:])
  if projector:
    return [core,P]
  return core
```

The input arguments of ST-HOSVD and T-HOSVD implementations are the same.
The first difference with `T_hosvd` is the initial declaration of `core` tensor, set equal
to `T`. Next we highlight that only if the `projector` variable is `True`, we initialize an
empty list to store the projectors matrices. Another difference is the computation of
the core tensor inside the for loop with the component wise product between matrix
and tensor, fixed a certain direction. The basic output is still the final core tensor.
The last codes, we present, are used to compute the biodiversity index. Remind that
for every raster of size (m, n), fixed l the side of the moving window, the biodiversity
index is computed $l \times m \times n$ times. In our case $l = 11$ and so we have to repeat the
computations 28512×10^4 and 31680×10^4 times for the World and the Europe maps
respectively. To speed up the entire work we decide to implement a parallel version
of the biodiversity index algorithm. When a computer executes a parallel functions
its cores perform independently different tasks at the same time. An example will
help us.

Example 3.1.3. Let A be an array of 10^n elements, with $n \in \mathbb{N}$ and enough big. If
we want to compute and to save in another array the square value of each element
of A we can do:

```
  B = []
  for x in A:
    y = x**2
    B.append(y)
```

If we assume that each execution of the for loop body costs 1 time unite, the total
time to have the complete list `B` will be of 10^n time unite. However if there are two
cores available on our machine, each core can do the operations inside the for loop
at the same time. For example one core can compute the square value of elements
in even position, the other the wanted value of elements in odd position. Therefore
the total time will be reduced to $\frac{10^n}{2}$ time units.

In our case we want that each core of the used machine works on different position
of the moving window. Consequently to implement this mechanism we needed a
parallel computing library, as `Joblib`, cf. [**res4**]. We chose `Joblib` since its ease
of use. To explain how we used the `Joblib` commands `Parallel` and `delayed`, we
present an example.

Example 3.1.4. Reminding Example 3.1.3, we define a function which depends on the for loop index.

```python
def f(i, array = A):
    y = A[i]**2
    return y
```

The variable i is the index which will scroll A elements, while the array A is set as default. Then to perform parallel computation we will call

```python
from joblib import Parallel, delayed

B = (Parallel(n_jobs = 2)(delayed(f)(i) for i in range(0,len(A))))
```

Indeed it is sufficient to write a function, which takes as input a scrolling index and the `Joblib Parallel` and `delayed` will distribute the steps on the available cores. Other two used library are `itertools`, to create iterable elements and `spatial` from `SciPy`, to compute the distance element wise for two matrix. The main Rao computation code is

```python
#### computation
import numpy as np
import scipy
from scipy import spatial
import itertools
#### parallelisation
import joblib
from joblib import Parallel, delayed
import multiprocessing

def raop(rw):
  def raout (cl, rw = rw, rasterm = rasterm, missing = missing, w =
  ↪  w, distance_m = distance_m):
    tw_labels, tw_values =
    ↪  np.unique(rasterm[(rw-w):(rw+w+1),(cl-w):(cl+w+1)],
    ↪  return_counts=True)
    if len(np.where(tw_labels == missing)) != 0:
      tw_values = np.delete( tw_values, np.where(tw_labels ==
      ↪  missing))
      tw_labels = np.delete( tw_labels, np.where(tw_labels ==
      ↪  missing))
    if len(tw_values) > 1:
      d1 = spatial.distance.cdist(np.diag(tw_labels),
      ↪  np.diag(tw_labels), distance_m)
      p = tw_values/np.sum(tw_values)
```

```python
        p1 = np.zeros((len(tw_values),len(tw_values)))
        comb = np.array([x[0]*x[1] for x in
        ↪   list(itertools.combinations(p, 2))])
        p1[np.triu_indices(len(tw_values), k=1)] = comb
        p1[np.tril_indices(len(tw_values), k=-1)] = comb
        return ((np.sum(np.multiply(p1,d1))))
      elif len(tw_values) == 1:
        return (((0)))
      else:
        return ((missing))
    Raout = Parallel(n_jobs = NcCores)(delayed(raout)(cl) for cl in
    ↪   range(w,c-w))
    return (Raout)
  out[w:(r-w), w:(c-w)] = (np.asarray(Parallel(n_jobs =
  ↪   NcCores)(delayed(raop)(rw) for rw in
  ↪   range(w,(r-w)))).reshape(r-2*w,c-2*w))
```

Since the moving window scrolls over rows and over columns, we define two parallelised functions one inside the others, `raop` whose variable is just the row index and `raout`. This second function takes as variable the column index and has other set parameters: `rasterm` the raster, `rw` the row index, `missing` a value used in the raster when data are not present, `w` the window side and a function `distance_m` to compute the distance between raster values. The first step is storing the values and their frequencies of the raster area covered by the moving window in the arrays `tw_labels` and `tw_values` respectively. Then we check if there is the `missing` value in the considered area: if it is present, we remove it and its frequency from the storing arrays. Next if there are at least two different elements in the `tw_values` array, we compute their distance with the function `spatial.distance.cdist` and save the result in the `d1` matrix. In `p` we put the relative frequencies, obtained from the absolute ones. We define a 0 matrix `p1`, with the same size of `d1`. We compute and store in `comb` all the possible combination between different elements of vector `p`. We assign these values to the upper and lower triangular part of the matrix `p1`. Lastly we return the sum of the elements of the product matrix between `d1` and `p1`. Notice that the product matrix will coherently have diagonal null elements, since the distance between a pixel value and itself is 0.

If there is just one element in vector `tw_values`, we return 0 since the difference between a pixel value and itself is 0. Otherwise, if `tw_values` is an empty vector, we return `missing`. Inside function `raop` we define function `raout` and we do the first `Parallel` call. Outside it we do the second `Parallel call`. Notice that this function is thought to compute Rao's index when the moving window is completely contained into the raster. When this condition is not satisfied, to speed up computation, there is a special implementation of Rao's, available here [**res5**].

With a similar approach, we implemented Rényi's index, whose core is

```python
  #### computation
```

```python
import numpy as np
import scipy
#### parallelisation
import joblib
from joblib import Parallel, delayed
import multiprocessing

def IndexOP (rw):
  def IndexOut (cl, rw = rw, rasterm = rasterm, missing = missing, w
  ↪  = w, alpha=alpha, base = base):
    tw_labels, tw_values =
    ↪  np.unique(rasterm[(rw-w):(rw+w+1),(cl-w):(cl+w+1)],
    ↪  return_counts=True)
    if len(np.where(tw_labels == missing)[0]) != 0:
      tw_values = np.delete( tw_values, np.where(tw_labels ==
      ↪  missing))
    if len(tw_values) != 0:
      p = tw_values/np.sum(tw_values)
      if np.log(np.sum(p**alpha)) == 0:
        return(0)
      else:
        return((1/(1-alpha)) * np.log(np.sum(p**alpha)) /
        ↪  np.log(base))
    else:
      return (missing)
  Index_Out = Parallel(n_jobs = NcCores)(delayed(IndexOut)(cl) for
  ↪  cl in range(w,c-w))
  return (Index_Out)
out[w:(r-w), w:(c-w)] = np.asarray(Parallel(n_jobs =
↪  NcCores)(delayed(IndexOP)(rw) for rw in
↪  range(w,r-w))).reshape(r-2*w,c-2*w)
```

Similarly we declare function `IndexOP`, with just row index as variable and inside it function `IndexOut`. This one takes as parameters some equals to `raout`'s ones. The variable proper of `IndexOut` are `alpha`, the Rényi's parameter, and `base`, the logarithm base. The code structure is similar to `raout`. We compute and store the values, present inside the raster area covered by the moving window, and their frequencies. We check and in case delete `missing` value and its frequency. If there is still an element into `tw_values`, we compute the relative frequencies and return the Rényi index. We test if the index final value is 0 and return 0 to avoid sign problem. If `tw_value` is an empty vector, we return `missing` value. We do a first `Parallel` call of `IndexOut` inside `IndexOP` and a second one of `IndexOP` outside it. Even in this case the presented code will work only when the moving window is entirely contained in the raster. To speed up the computations, we develop a special version of this

code which works when the moving window is not fully contained. It is available
here [**res5**].

3.2   Approach and results

Since obtaining the biodiversity indexes takes long time and needs many re-
sources, we perform all the computation over the university of Trento cluster. The
indexes we estimate are Rao and Rényi, i.e. Rényi family index with $\alpha = 2$, both
with window side equal to 11. We maintain the same window side on Europe's and
on Earth's images, since the different spatial resolutions lead to similar rasters di-
mensions.
The first step is computing both Rao and Rényi index over the NDVI raster, extrac-
ted from the loaded tensor, i.e. over $(\mathcal{A}_{k_h})_{\cdot,\cdot,1}$ for every $k_h \in \{1, \ldots, n_h\}$ for every
$h \in \{E, W\}$.

Definition 3.2.1. Let $\mathscr{R}_h$ be the set of Rao index computed over $(\mathcal{A}_{k_h})_{\cdot,\cdot,1}$ for every
$k_h \in \{1, \ldots, n_h\}$ for every $h \in \{E, W\}$. Similarly let $\mathscr{I}_h$ be the set of Rényi index
computed over $(\mathcal{A}_{k_h})_{\cdot,\cdot,1}$ for every $k_h \in \{1, \ldots, n_h\}$ for every $h \in \{E, W\}$.
We call R *original estiamtes* for every $R \in \mathscr{R}_E \cup \mathscr{R}_W \cup \mathscr{I}_E \cup \mathscr{I}_W$.

We use the obtained images as comparison term. Then we compute also an NDVI
from the RED and NIR raster, using Definition 1.2.3, since the algorithm used by
NASA for NDVI creation is not public.

Definition 3.2.2. Let $g_E : \mathbb{R}^{\otimes E} \mapsto \mathbb{M}^{n_1 \times n_2}(\mathbb{R}) \times \mathbb{M}^{n_1 \times n_2}(\mathbb{R})$ and $g_W : \mathbb{R}^{\otimes W} \mapsto$
$\mathbb{M}^{m_1 \times m_2}(\mathbb{R}) \times \mathbb{M}^{m_1 \times m_2}(\mathbb{R})$ be such that

$$g_h(\mathcal{A}) = (\mathcal{A}_{\cdot,\cdot,2}, \mathcal{A}_{\cdot,\cdot,3})$$

for every $\mathcal{A} \in \mathscr{T}_h$ for every $h \in \{E, W\}$. Let $M \in \mathbb{R}$ be a default missing value and
let $l : \mathbb{R} \times \mathbb{R} \mapsto \mathbb{R}$ be a function such that

$$l(a, b) = \begin{cases} \frac{a-b}{a+b} & \text{if } a + b \neq 0 \\ M & \text{if } a + b = 0 \end{cases}$$

for every $a, b \in \mathbb{R}$.
Let $\bar{l} : \mathbb{M}^{p \times q}(\mathbb{R}) \times \mathbb{M}^{p \times q}(\mathbb{R}) \mapsto \mathbb{M}^{p \times q}(\mathbb{R})$ be such that for every $A, B \in \mathbb{M}^{p \times q}(\mathbb{R})$
then $\bar{l}(A, B) = C$ such that $C_{i,j} = l(A_{i,j}, B_{i,j})$ for every $i \in \{1, \ldots, p\}$ for every
$j \in \{1, \ldots, q\}$. Let $f_h = \bar{l} \circ g_k$ be the function that associates to each tensor $\mathcal{A} \in \mathscr{T}_h$
the NDVI raster obtained from $(\mathcal{A}_{k_h})_{\cdot,\cdot,2}$ and $(\mathcal{A}_{k_h})_{\cdot,\cdot,3}$ for every $k_h \in \{1, \ldots, n_h\}$ for
every $h \in \{E, W\}$. Then
$$\tilde{\mathscr{T}}_h = f(\mathscr{T}_h)$$

for every $h \in \{E, W\}$.
We call elements of $\mathcal{n}_h$ *self-made NDVI images* for every $h \in \{E, W\}$.

Remark 3.2.3. *Since for every NIR and RED band rasters have only non negative elements, the second case of function l in the previous definition is verified when both elements of NIR and RED rasters are zero. In that case we assign to NDVI the default missing value, $M = -3000$.*

Consequently we perform again index estimation over n_h elements for every $h \in \{E, W\}$.

Definition 3.2.4. Let $\tilde{\mathcal{R}}_h$ be the set of Rao index computed over $A_{k_h} \in n_h$ for every $k_h \in \{1, \ldots, n_h\}$ for every $h \in \{E, W\}$. Similarly let $\tilde{\mathcal{G}}_h$ be the set of Rényi index computed over $A_{k_h} \in n_h$ for every $k_h \in \{1, \ldots, n_h\}$ for every $h \in \{E, W\}$.
We call R *relative estiamtes* for every $R \in \tilde{\mathcal{R}}_E \cup \tilde{\mathcal{R}}_W \cup \tilde{\mathcal{G}}_E \cup \tilde{\mathcal{G}}_W$.

Remark 3.2.5. *Henceforth we assume that elements of the same set pairs $(\mathcal{R}_h, \tilde{\mathcal{R}}_h)$ and $(\mathcal{G}_h, \tilde{\mathcal{G}}_h)$ are ordered equally for every $i \in \{E, W\}$.*

Even now, we can present some comparison between these two types of estimates. We calculate the error $\|A_j - B_j\|_F$ for every $A_j \in \mathcal{R}_i$ and $B_j \in \tilde{\mathcal{R}}_i$ for every $j \in \{1, \ldots, n_i\}$ for every $i \in \{E, W\}$. Next we also compute the error per pixel dividing the error by the number of pixels. Since we are working with huge matrices, this type of distributed error is useful to understand how big is on average the error made pointwise. In the following table we present some statistics, where e is the error vector and ep is the error per pixel vector. Besides for every $v \in \mathbb{K}^n$, we set

$$\min v = \min v_1, \ldots, v_n \quad \text{and} \quad \max v = \max\{v_1, \ldots, v_n\}.$$

Dataset	$\mathbb{E}[e]$	$\mathbb{E}[ep]$	$\mathbb{V}\mathrm{ar}[e]$	$\mathbb{V}\mathrm{ar}[ep]$	$\min ep$	$\max ep$
Europe	197877.819	0.0069	409993378560.6464	0.0005	0.0013	0.1777
World	1731817.3949	0.0668	687978783275.7339	0.001	0.0155	0.1678

Table 3.1: Rao index statistics for original and relative data.

We make on average a 0.6% error per pixel for Europe Rao estimation using self-made NDVI instead of NASA NDVI, while the error is on average of 6% per pixel when Rao is computed over Earth NDVIs. Both the unbiased variance are very small, which means that errors are not very different from the mean. Notice that also the minimum approximation error is greater for Earth than for Europe data. We suppose that this is linked with the water surface greater in World rasters than in Europe's ones.
Similarly we calculate the error $\|A_j - B_j\|_F$ for every $A_j \in \mathcal{G}_i$ and $B_j \in \tilde{\mathcal{G}}_i$ for every $j \in \{1, \ldots, n_i\}$ for every $i \in \{E, W\}$. With the same notation, introduced previously, we presents some statistics for Rényi index.

Dataset	$\mathbb{E}[e]$	$\mathbb{E}[ep]$	$\mathbb{V}\mathrm{ar}[e]$	$\mathbb{V}\mathrm{ar}[ep]$	$\min ep$	$\max ep$
Europe	132517.8038	0.0046	420678175017.4135	0.0005	0.0003	0.1777
World	782043.4085	0.0302	148701528527.3207	0.0002	0.0112	0.0688

Table 3.2: Rényi index statistics for original and relative data.

In this case the mean error per pixel for both the dataset is smaller than the previous mean. We explain this remarking that Rényi index takes into account only frequencies while Rao index considers both values and their frequencies. However in this case variance in World error is smaller than in Europe, while in the Rao case we have the opposite situation. Moreover we observe that the minimum and the maximum approximation error in the World dataset for both the indexes is realised by the same element, i.e. April 2018 and May 2014 respectively. This consideration holds also for Europe dataset: the minimum error comes from April 2018 and the maximum from December 2012.

The next steps of our analysis include the HOSVD. Before starting our approach description, we have to highlight one limit of `svds`. It takes as additional parameter `k`, which is the rank of the wanted approximation and which has to be strictly lower than the rank of the given matrix. So if we had passed just a 3-order tensor such that $n_3 = 2$, for the third flattening we would have fixed `k` equal to 1, getting a vector: too little for our aims. Therefore we decide to increase n_3 up to 3, adding another matrix to our tensor: in the first case we take twice RED band raster and once NIR one, in the second case we take twice NIR band and once RED one.

Definition 3.2.6. Let $g_{R,h} : R^{\otimes h} \mapsto \mathbb{R}^{\otimes h}$ be the function that associate the tensor $\mathcal{B}$ such that

$$\mathcal{B}_{\cdot,\cdot,1} = \mathcal{B}_{\cdot,\cdot,2} = \mathcal{A}_{\cdot,\cdot,2} \quad \text{and} \quad \mathcal{B}_{\cdot,\cdot,3} = \mathcal{A}_{\cdot,\cdot,3},$$

to each tensor $\mathcal{A} \in \mathcal{T}_h$ for every $h \in \{E, W\}$. Then

$$\mathcal{T}_{R,h} = g_{R,h}(\mathcal{T}_h)$$

for every $h \in \{E, W\}$.

Similarly let $g_{N,h} : R^{\otimes h} \mapsto \mathbb{R}^{\otimes h}$ be the function that associates the tensor $\mathcal{B}$ such that

$$\mathcal{B}_{\cdot,\cdot,1} = \mathcal{B}_{\cdot,\cdot,3} = \mathcal{A}_{\cdot,\cdot,3} \quad \text{and} \quad \mathcal{B}_{\cdot,\cdot,2} = \mathcal{A}_{\cdot,\cdot,2},$$

to each tensor $\mathcal{A} \in \mathcal{T}_h$ for every $h \in \{E, W\}$. Then

$$\mathcal{T}_{N,h} = g_{N,h}(\mathcal{T}_h)$$

for every $h \in \{E, W\}$,

To compute T-HOSVD and ST-HOSVD, we decide five multilinear target ranks.

Definition 3.2.7. Let $\mathcal{R} = \{10, 50, 100, 500, 1000\}$ be a set with the given order fixed, then the *target multilinear rank* we choose are

$$r_j = (i_j, i_j, 2)$$

for every $i_j \in \mathcal{R}$ for every $j \in \{1, \ldots, 5\}$. Let $\mathcal{T}_{k,h,j}$ be the set of T-HOSVD approximation at multilinear rank r_j of tensors from the set $\mathcal{T}_{k,h}$ for every $h \in \{E, W\}$, for every $k \in \{N, R\}$ and for every $j \in \{1, \ldots, 5\}$. Similarly let $\mathcal{S}_{k,h,j}$ be the set of ST-HOSVD approximation.

Before presenting results related to indexes computation, we have some data about storage memory use. Since it depends on the core tensor and on the projectors dimension, which are equal for T-HOSVD and ST-HOSVD, we report only a table for each dataset. For each tensor $\mathcal{A} \in \mathcal{T}_{k,W}$ the ratio between the memory used for storing the core tensor and the projectors over the memory used for storing $\mathcal{A}$ is the same, for every $k \in \{N, R\}$. For tensors of $\mathcal{T}_{k,E}$ it holds the same, except for those elements composed by a lower number of granules, for every $k \in \{N, R\}$. Since they are 2 over 91, we neglect them and in the table we present the ratios of memory usage for each rank approximation. We call these ratios *absolute compression ratios*, because they have as denominator the memory used to store two time RED band and once NIR band, or vice-versa. In the following table they are present as *Abs. R*, when RED band raster is repeated and as *Abs. N*, in the other case. Beside we list also a *relative compression ratio*, where the denominator is the amount of memory necessary to store once RED and once NIR band. In the table it is *Rel*. Moreover here and all along the thesis for simplicity we write as rank only the significant components of the multilinear rank, i.e. i_j for every $i_j \in \mathcal{R}$ for every $j \in \{1, \ldots, 5\}$.

Rank	Europe			Earth		
	Rel	Abs. R	Abs. N	Rel	Abs. R	Abs. N
10	0.0019	0.0013	0.0013	0.0021	0.0014	0.0014
50	0.0095	0.0063	0.0063	0.0105	0.007	0.007
100	0.0191	0.0127	0.0127	0.0212	0.0141	0.0141
500	0.1024	0.0683	0.0683	0.1138	0.0759	0.0759
1000	0.2222	0.1481	0.1481	0.2469	0.1646	0.1646

Table 3.3: Rate of compression.

Notice that even with the greater component wise target multilinear rank, we need just a small percentage of memory with respect the one used for storing the entire tensor. In addiction to this, even the relative ratio at the highest multilinear rank present a significant saving in memory use.

In order to not get lost during the presentation with the numerous indexes used, we will give a general idea. After having generated new tensors and having approximated

them, we we calculate new NDVIs through function f_h defined in 3.2.2 applied on $\mathcal{T}_{k,h,j} \cup \mathcal{S}_{k,h,j}$ for every $h \in \{E, W\}$, for every $k \in \{N, R\}$ and for every $j \in \{1, \ldots, 5\}$. Over them we calculate biodiversity index. Lastly we measure the difference in estimating biodiversity from approximated NDVI and NASA or self-made NDVI.

Definition 3.2.8. Let $\mathcal{N}_{k,h,T,j} = f_h(\mathcal{T}_{k,h,R,j})$ and let $\mathcal{N}_{k,h,S,j} = f_h(\mathcal{S}_{k,h,R,j})$ every $j \in \{1, \ldots, 5\}$ and for every $h \in \{E, W\}$. We call elements of $\mathcal{N}_{k,h,T,j} \cup \mathcal{N}_{k,h,S,j}$ *approximated NDVIs*.

To describe hour work in a clearer way, we present the discussion for Rényi and Rao index in different subsections.

3.2.1   Rényi Index

The next step is computing Rényi index over approximated NDVIs

Definition 3.2.9. Let $\mathcal{G}_{R,h,k,j}$ be the set of Rényi index computed over elements of $\mathcal{N}_{R,h,t,j}$ for every $h \in \{E, W\}$, for every $t \in \{T, S\}$ and for every $j \in \{1, \ldots, 5\}$. We decide to call these i_j-*approximated estimates* for every $i_j \in \mathcal{R}$ and for every $j \in \{1, \ldots, 5\}$.

Remark 3.2.10. *Notice that we have 4 indexes for approximated estimates set:*

1st index *indicates the repeated matrix in the starting tensor R for RED and N for NIR;*

2nd index *the belonging dataset, E for Europe and W for Earth;*

3rd index *the approximation algorithm, T for T-HOSVD and S for ST-HOSVD;*

4th index *is associated with the target multilinear rank.*

The last step is computing the error with respect to the original estimates, i.e.

$$\left\| A_k - C_{k,j} \right\|$$

for every $A_k \in \mathcal{G}_h$ and for every $C_{h,j} \in \mathcal{G}_{R,h,T,j} \cup \mathcal{G}_{R,h,S,j}$, for every $k \in \{1, \ldots, n_h\}$, for every $j \in \{1, \ldots, 5\}$ and for every $h \in \{E, W\}$. Moreover we compute the error with respect to relative estimates, i.e.

$$\left\| B_k - C_{k,j} \right\|$$

for every $B_k \in \tilde{\mathcal{G}}_h$ and for every $C_{h,j} \in \mathcal{G}_{R,h,T,j} \cup \mathcal{G}_{R,h,S,j}$, for every $k \in \{1, \ldots, n_h\}$, for every $j \in \{1, \ldots, 5\}$ and for every $h \in \{E, W\}$.

Now we present some statistics for the errors per pixel with respect to original estimates, stored in vector epO and relative estimates, in vector epR for Europe dataset, for both the decomposition techniques for each target multilinear rank. In the following table we report as rank only the significant component of the multilinear rank, for simplicity.

Remark 3.2.11. *To simplify the discussion henceforth and all along the thesis the i_j-original error will be the error between original estimate and i_j-approximated estimate, while the i_j-relative error will be the error between relative estimate and i_j-approximated estimate for every $i_j \in \mathcal{R}$.*

Rank	**T-HOSVD**					**ST-HOSVD**				
	10	50	100	500	1000	10	50	100	500	1000
$\mathbb{E}[epO]$	0.1576	0.0914	0.0801	0.0807	0.0766	0.1597	0.0917	0.0803	0.0798	0.0755
$\mathbb{V}\mathrm{ar}[epO]$	0.0003	0.0002	0.0002	0.0013	0.0013	0.0003	0.0002	0.0002	0.0013	0.0013
$\mathbb{E}[epR]$	0.1572	0.0913	0.0807	0.0821	0.079	0.1593	0.0918	0.0811	0.0815	0.0779
$\mathbb{V}\mathrm{ar}[epR]$	0.0003	0.0002	0.0003	0.0014	0.0014	0.0003	0.0002	0.0003	0.0014	0.0014
$\min epO$	0.1164	0.0641	0.0537	0.0466	0.0383	0.1164	0.0646	0.0542	0.0459	0.0367
$\min epR$	0.1164	0.0641	0.0537	0.0466	0.0383	0.1164	0.0646	0.0542	0.0459	0.0367
$\max epO$	0.1864	0.1377	0.1136	0.194	0.1999	0.1931	0.1371	0.1104	0.1932	0.199
$\max epR$	0.1864	0.147	0.1618	0.194	0.1999	0.1931	0.1479	0.1679	0.1932	0.199

Table 3.4: Statistics for Rényi index over $N \in \mathcal{n}_{R,E,t,j}$.

Notice that ST-HOSVD original and relative error per pixel is lower than T-HOSVD original and relative error per pixel, when the first two components of target multilinear rank are greater or equal than 500. The variance of both errors is quite low, even if it increases in the last three multilinear ranks. Moreover we underline that the minimum relative error and the minimum original error are equal up to the forth decimal digit. They also decrease, when the components of multilinear ranks increase. On the other hand the maximum of relative errors and the maximum of original errors do not coincide. Beside, they increase significantly in the forth and fifth approximation. Finally we notice that the average relative error per pixel is frequently slightly greater than the original one. For T-HOSVD this inequality between original and relative error average happens from the third approximation, while for the ST-HOSVD from the second one. The difference seems to grow for increasing multilinear rank components. This could appear a bit strange, since we expect the contrary. However we have to remark that Rényi index takes in account only raster values frequencies, neglecting the values themselves. In addiction from the complete data, we observe that the relative error of elements with missing granules, tensors of December 2012 and December 2015, is more than 3 times the original error.

Next we list statistics about the Earth dataset. Similarly for each target multilinear rank in vector epO there are the errors per pixel with respect to original Rényi estimates, while in epR with respect to relative estimates.

	T-HOSVD					**ST-HOSVD**				
Rank	10	50	100	500	1000	10	50	100	500	1000
$\mathbb{E}[epO]$	0.1376	0.0943	0.0875	0.0626	0.0584	0.1383	0.0946	0.0876	0.0625	0.058
$\mathbb{V}\mathrm{ar}[epO]$	0.0001	0.0001	0.0002	0.0002	0.0001	0.0001	0.0001	0.0002	0.0002	0.0001
$\mathbb{E}[epR]$	0.1358	0.091	0.0846	0.0545	0.0491	0.1365	0.0912	0.0847	0.0545	0.0485
$\mathbb{V}\mathrm{ar}[epR]$	0.0001	0.0001	0.0002	0.0002	0.0	0.0001	0.0001	0.0002	0.0002	0.0
$\min epO$	0.1156	0.0758	0.0668	0.048	0.0471	0.1156	0.076	0.0672	0.0483	0.0468
$\min epR$	0.1155	0.0748	0.0655	0.0421	0.0414	0.1155	0.075	0.0658	0.0421	0.0411
$\max epO$	0.1599	0.1182	0.1255	0.0967	0.0813	0.1601	0.1185	0.1253	0.0973	0.0811
$\max epR$	0.155	0.1134	0.1263	0.0968	0.0744	0.1551	0.1138	0.126	0.0974	0.0731

Table 3.5: Statistics for Rényi index over $N \in \mathcal{n}_{R,W,t,j}$.

We remark that even in this case on average ST-HOSVD technique leads to lower original and relative error on average than T-HOSVD, for the last two and for the last one target multilinear rank respectively. Moreover we observe that relative error mean is slightly lower than original one, as we expected. Minimum and maximum of both errors decrease for increasing multilinear rank components. Certainly the most stunning value is the variance of relative error per pixel at the last approximation. For both T-HOSVD and ST-HOSVD it is lower than 10^{-4}.

We include the image associated to Rényi index in the five approximations for the Europe worst case and the Earth beast case, both from ST-HOSVD approximation technique.

Example 3.2.12. Let's observe the Rényi index computed over Europe NDVI of February 2013, in Figure 3.1. We immediately notice that biodiversity seems to be quite high, near 4.5 almost everywhere in Europe. However as remarked in [**eco33**], Rényi index tends to overestimate biodiversity. Besides we underline that Rényi index computed over self-made NDVI does not differ much from its computation over NASA NDVI.

Next we have in Figure 3.2a the same index computed over self-made NDVI and the approximated Rényi estimates at different multilinear ranks. We can notice that when the first two components of the multilinear rank grow, in the Rényi estimation some new noising elements appear, leading to high errors. We believe that this type of phenomenon deserves further analysis. However in the internal land of Europe, the biodiversity estimation is quite close to the relative and original one, for multilinear rank components strictly greater than 100.

In the next example we will consider the element of Earth dataset, which realises the minimum error.

Example 3.2.13. Firstly we show in Figure 3.3 the Rényi estimate over NASA NDVI of October 2017. As we said in the Example 3.2.12 Rényi index provides quite high biodiversity values. Indeed also in this case there are many Earth areas with a biodiversity value close to 4.5. Then we present the same index over self-made and approximated NDVI. Even if the small printing dimensions reduce the
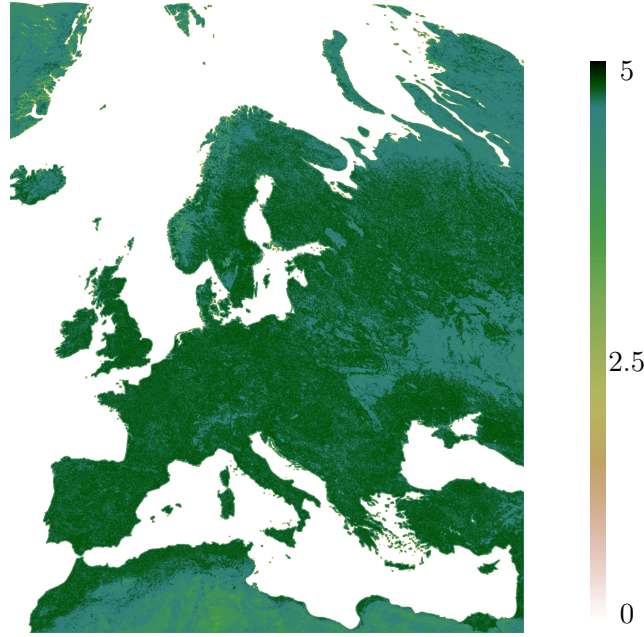
Figure 3.1: Rényi index computed over NASA NDVI of February 2013.

detail precision, our eyes do not perceive at first glance much difference between the last three index approximations and the index of Figure 3.3. However with a closer observation, we notice that islands of the Pacific ocean disappear in first approximation and they partially reappear in the last two images. We therefore conclude that the original and relative error is in this case strongly linked with these missing territories.

The next step in our discussions is computing Rényi index over approximated NDVI of $\mathcal{n}_{N,h,t,j}$ for every $h \in \{E, W\}$, for every $t \in \{T, S\}$ and $j \in \{1, \ldots, 5\}$.

Definition 3.2.14. Let $\mathcal{I}_{N,h,t,j}$ be the set of Rényi index computed over elements of $\mathcal{n}_{N,h,t,j}$ for every $h \in \{E, W\}$, for every $t \in \{T, S\}$ and for every $j \in \{1, \ldots, 5\}$. We decide to call these i_j-*approximated estimates* for every $i_j \in \mathcal{R}$ and for every $j \in \{1, \ldots, 5\}$.

As before, we compute the error with respect to the original estimates, i.e.

$$\left\| A_k - C_{k,j} \right\|$$

for every $A_k \in \mathcal{I}_h$ and for every $C_{h,j} \in \mathcal{I}_{N,h,T,j} \cup \mathcal{I}_{N,h,S,j}$, for every $k \in \{1, \ldots, n_h\}$, for every $j \in \{1, \ldots, 5\}$ and for every $h \in \{E, W\}$. Moreover we compute the error with respect to relative estimates, i.e.

$$\left\| B_k - C_{k,j} \right\|$$

(a) Relative approximation   (b) Component rank 10 compression

(c) Component rank 50 compression   (d) Component rank 100 compression

(e) Component rank 500 compression   (f) Component rank 1000 compression

Figure 3.2: Approximation of Rényi index for February 2013, from NDVI of $n_{R,E,S,j} \cup n_E$.

Figure 3.3: Rényi index computed over NASA NDVI of October 2017.

(a) Relative approximation

(b) Component rank 10 compression

(c) Component rank 50 compression

(d) Component rank 100 compression

(e) Component rank 500 compression

(f) Component rank 1000 compression

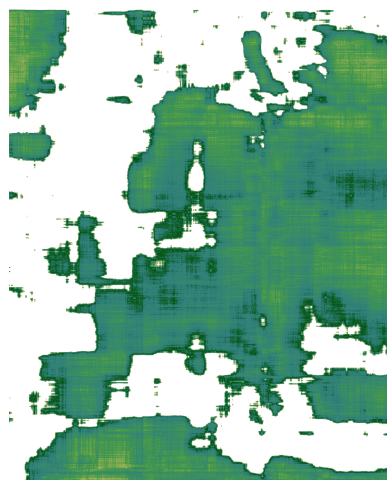Figure 3.4: Approximation of Rényi index for October 2017, from NDVI of $n_{R,W,S,j} \cup n_W$.

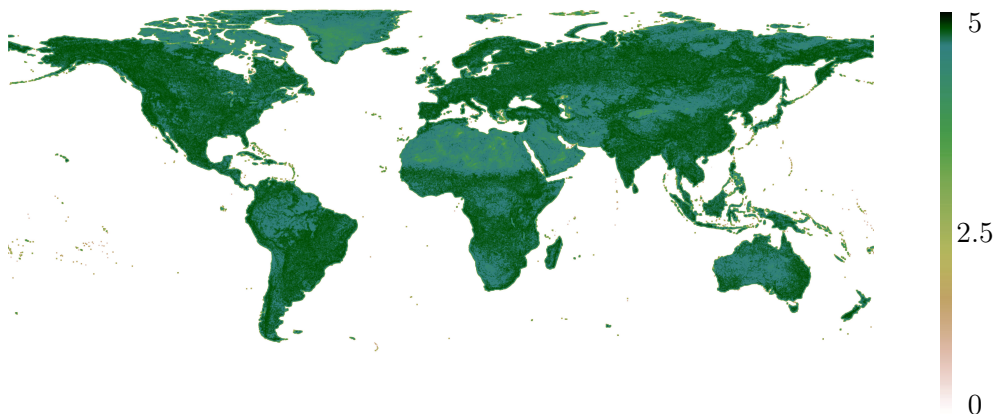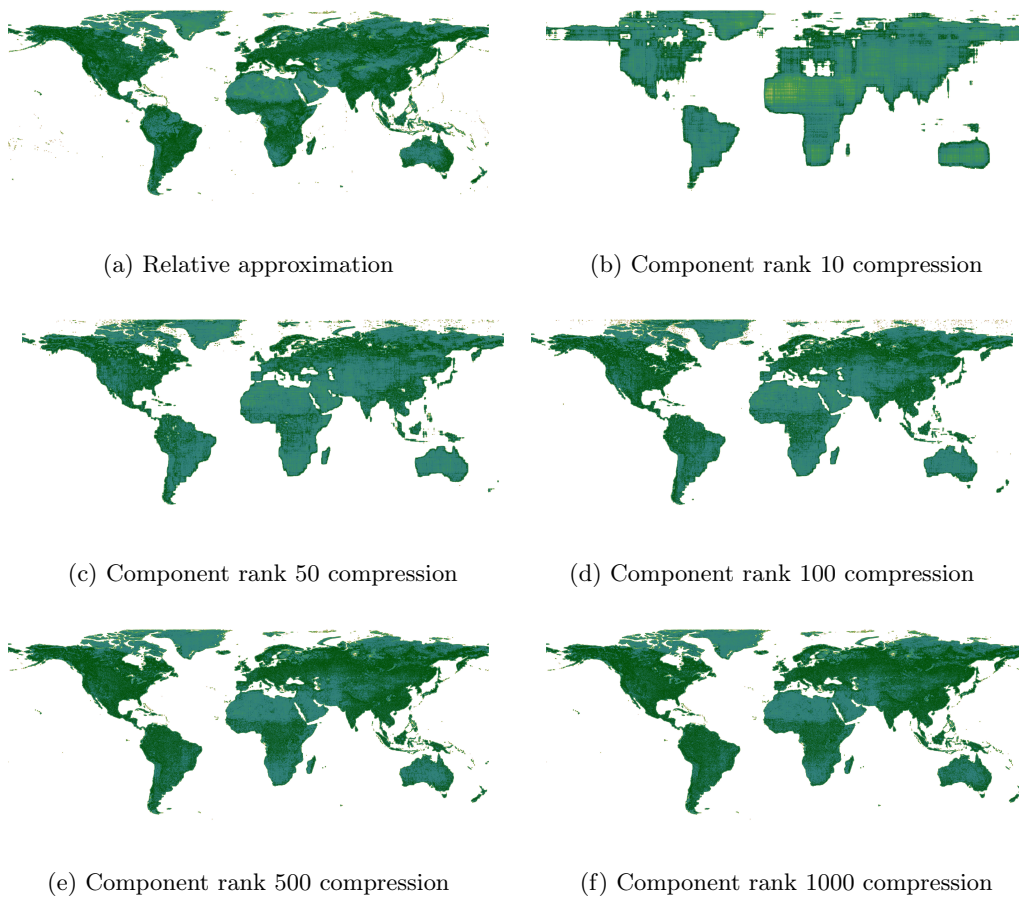for every $B_k \in \tilde{\mathcal{G}}_h$ and for every $C_{h,j} \in \mathcal{G}_{N,h,T,j} \cup \mathcal{G}_{N,h,S,j}$, for every $k \in \{1, \dots, n_h\}$, for every $j \in \{1, \dots, 5\}$ and for every $h \in \{E, W\}$.

Lastly we report some statistics about i_j-original and i_j-relative errors per pixel, stored in vector epO and epR for each $i_j \in \mathcal{R}$. Firstly a table for Europe related data.

	T-HOSVD					ST-HOSVD				
Rank	10	50	100	500	1000	10	50	100	500	1000
$\mathbb{E}[epO]$	0.1569	0.09	0.0782	0.0794	0.076	0.1586	0.0903	0.0783	0.0787	0.0751
$\mathbb{V}\mathrm{ar}[epO]$	0.0003	0.0002	0.0002	0.0013	0.0013	0.0003	0.0002	0.0002	0.0013	0.0013
$\mathbb{E}[epR]$	0.1563	0.0894	0.0782	0.0806	0.0783	0.158	0.0897	0.0785	0.08	0.0775
$\mathbb{V}\mathrm{ar}[epR]$	0.0003	0.0002	0.0002	0.0014	0.0015	0.0004	0.0002	0.0002	0.0015	0.0014
$\min epO$	0.1128	0.0623	0.0518	0.0449	0.0436	0.1143	0.0625	0.0521	0.0438	0.0428
$\min epR$	0.1128	0.0623	0.0518	0.0449	0.0436	0.1143	0.0625	0.0521	0.0438	0.0428
$\max epO$	0.1827	0.1462	0.1275	0.1936	0.1991	0.1855	0.1457	0.1222	0.1932	0.1988
$\max epR$	0.1827	0.1363	0.1484	0.1936	0.1991	0.1855	0.1373	0.1528	0.1932	0.1988

Table 3.6: Statistics for Rényi index over $N \in \mathcal{n}_{N,E,t,j}$s.

Almost every consideration for statistics of approximated estimates of $\mathcal{G}_{R,E,t,j}$ for every $t \in \{T, S\}$ and for every $j \in \{1, \dots, 5\}$ holds also in this case. However we can notice that on average the 100 relative and original error are lower that 500 one, but this is not true anymore for 1000 relative and original. Moreover when the first two multilinear rank components are strictly smaller that 500, T-HOSVD relative and original errors are lower than ST-HOSVD. For the statistics over $\mathcal{G}_{R,E,t,j}$ elements, this consideration holds only for relative error at the third multilinear rank. The remarks about not full granules elements are true also in this case. We underline that at each multilinear rank both the original and the relative errors on average are smaller in this second case, i.e. applying the procedure to tensors where the NIR band raster is repeated.

Lastly some statistical aspects about Earth approximated estimates. As previously, we list a table with mean, variance, min and max for i_j-original and i_j-relative errors per pixel, stored respectively in vector epO and epR for each $i_j \in \mathcal{R}$.

	T-HOSVD					ST-HOSVD				
Rank	10	50	100	500	1000	10	50	100	500	1000
$\mathbb{E}[epO]$	0.1351	0.0915	0.084	0.0601	0.0556	0.1355	0.0917	0.0842	0.0601	0.0553
$\mathbb{V}\mathrm{ar}[epO]$	0.0001	0.0001	0.0002	0.0002	0.0001	0.0001	0.0001	0.0002	0.0002	0.0001
$\mathbb{E}[epR]$	0.1334	0.0879	0.0807	0.052	0.046	0.1338	0.0882	0.0809	0.052	0.0455
$\mathbb{V}\mathrm{ar}[epR]$	0.0001	0.0001	0.0002	0.0002	0.0001	0.0001	0.0001	0.0002	0.0002	0.0
$\min epO$	0.1119	0.0727	0.0638	0.0443	0.0424	0.112	0.0728	0.0641	0.0445	0.0422
$\min epR$	0.1114	0.0715	0.0623	0.0385	0.0382	0.1115	0.0716	0.0628	0.0387	0.0379
$\max epO$	0.1564	0.1154	0.121	0.0963	0.0792	0.1574	0.116	0.1209	0.0964	0.0791
$\max epR$	0.1536	0.1098	0.1218	0.0965	0.0731	0.1536	0.1099	0.1217	0.0966	0.0719

Table 3.7: Statistics for Rényi index over $N \in \mathcal{n}_{N,W,t,j}$.

Also in this case the considerations presented for $\mathcal{G}_{R,W,t,j}$ error statistics hold. We simply remark that we find again variance lower than 10^{-4} for 1000-relative error of ST-HOSVD. Moreover also in this case ST-HOSVD is convenient only when the first two multilinear ranks components are greater than 500. Lastly original and relative error on average are again smaller in this second case, i.e. when we apply our method to tensors where NIR band is repeated.
We will present briefly the Rényi index image over Europe NDVI of February 2013 and over Earth NDVI of October 2017, obtained starting from the correspondent tensors of $\mathcal{T}_{N,h}$ for $h \in \{E, W\}$.

Example 3.2.15. In example 3.2.12 we presented the Rényi index image, which realises the highest original and relative error, starting from RED repeated band. Here we have the image of Rényi index for the same element, with approximation obtained from NIR band repeated. We remark that also starting from twice NIR and once RED band tensor element of February 2013 realises the highest original and relative error. Again we observe an increasing presence of noise in the north Europe area for growing multilinear rank components.

Example 3.2.16. Similarly we display approximation of Rényi index for Earth element of October 2017, obtained from tensors where is repeated twice the NIR band. As in Example 3.2.12 we underline that the number of detected territories in the Pacific ocean grows when the first two multilinear rank components grow. Moreover comparing Figure 3.4f and Figure 3.6e, we notice that the amount of Pacific island present is greater in the second one. Indeed this is the element which minimises the original error also in this second proceeding way. Therefore also in this case we can affirm that choosing a starting tensor with repeated NIR band is more convenient that starting with repeated RED band.
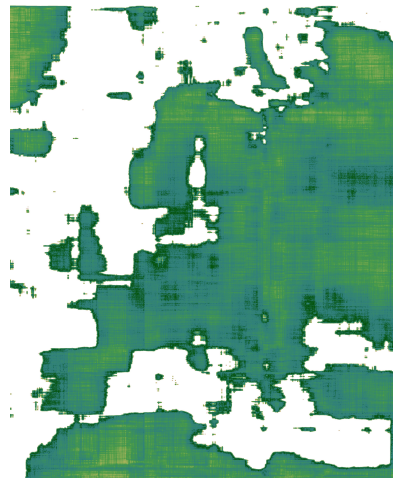
(a) Component rank 10 compression

(b) Component rank 50 compression

(c) Component rank 100 compression

(d) Component rank 500 compression

(e) Component rank 1000 compression

Figure 3.5: Approximation of Rényi index for February 2013, from NDVI of $n_{R,E,S,j}$.

(a) Component rank 10 compression
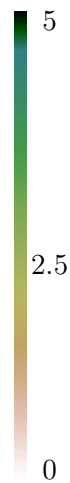
(b) Component rank 50 compression

(c) Component rank 100 compression

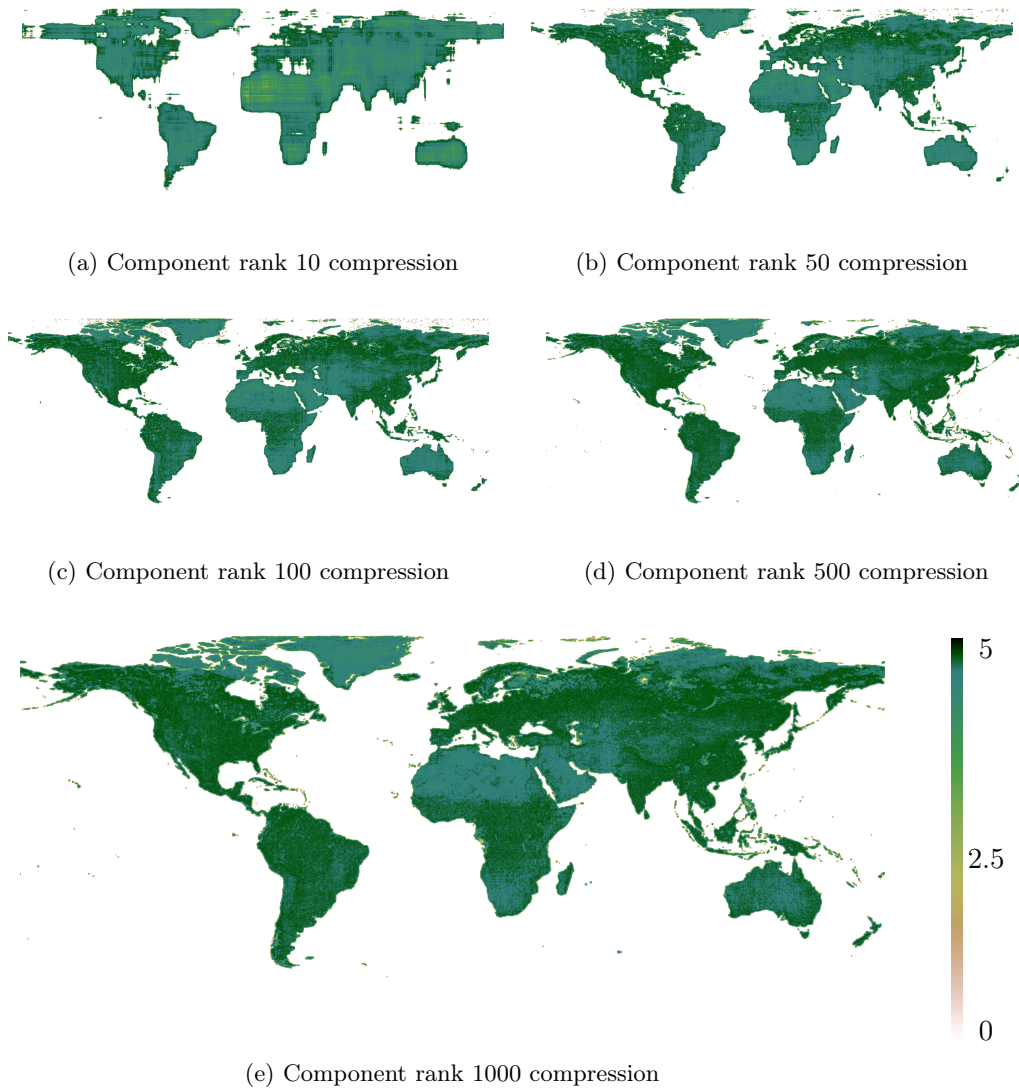(d) Component rank 500 compression

(e) Component rank 1000 compression

Figure 3.6: Approximation of Rényi index for October 2017, from NDVI of $n_{N,W,S,j}$.

3.2.2   Rao index

Now we compute Rao index over approximated NDVIs.

Definition 3.2.17. Let $\mathcal{R}_{R,h,k,j}$ be the set of Rao index computed over elements of $\mathcal{n}_{R,h,t,j}$ for every $h \in \{E, W\}$, for every $t \in \{T, S\}$ and for every $j \in \{1, \ldots, 5\}$. We call these i_j-*approximated estimates* for every $i_j \in \mathcal{R}$ and for every $j \in \{1, \ldots, 5\}$.

As we did previously, we compute the error with respect to the original estimates, i.e.

$$\left\| A_k - C_{k,j} \right\|$$

for every $A_k \in \mathcal{R}_h$ and for every $C_{h,j} \in \mathcal{R}_{R,h,T,j} \cup \mathcal{R}_{R,h,S,j}$, for every $k \in \{1, \ldots, n_h\}$, for every $j \in \{1, \ldots, 5\}$ and for every $h \in \{E, W\}$. Moreover we compute the error with respect to relative estimates, i.e.

$$\left\| B_k - C_{k,j} \right\|$$

for every $B_k \in \tilde{\mathcal{R}}_h$ and for every $C_{h,j} \in \mathcal{R}_{R,h,T,j} \cup \mathcal{R}_{R,h,S,j}$, for every $k \in \{1, \ldots, n_h\}$, for every $j \in \{1, \ldots, 5\}$ and for every $h \in \{E, W\}$.
To describe our results we report some statistics about i_j-original and i_j-relative error per pixel, stored in vector epO and epR for each $i_j \in \mathcal{R}$. Firstly we list information for Europe related data.

	T-HOSVD					ST-HOSVD				
Rank	10	50	100	500	1000	10	50	100	500	1000
$\mathbb{E}[epO]$	0.6419	0.3621	0.2944	0.2059	0.1922	0.6328	0.3604	0.293	0.2081	0.1951
$\mathbb{V}\mathrm{ar}[epO]$	0.0044	0.0012	0.001	0.0031	0.003	0.0038	0.0011	0.001	0.003	0.003
$\mathbb{E}[epR]$	0.6424	0.3633	0.2961	0.2083	0.1953	0.6333	0.3615	0.2946	0.2104	0.1982
$\mathbb{V}\mathrm{ar}[epR]$	0.0043	0.0013	0.0013	0.0038	0.0045	0.0037	0.0013	0.0013	0.0038	0.0045
$\min epO$	0.4819	0.274	0.2194	0.1306	0.1128	0.4825	0.2724	0.2185	0.1326	0.1144
$\min epR$	0.4819	0.274	0.2193	0.1306	0.1127	0.4825	0.2724	0.2185	0.1326	0.1144
$\max epO$	0.7559	0.4081	0.3591	0.3858	0.3894	0.7246	0.4065	0.3722	0.3871	0.3917
$\max epR$	0.7558	0.465	0.4659	0.4486	0.5599	0.7246	0.4746	0.4734	0.4742	0.5618

Table 3.8: Statistics for Rao index over $N \in \mathcal{n}_{R,E,t,j}$.

The most evident aspect is the high mean of both original and relative error made, even when the components of multilinear rank grow. Indeed this average is close to a 20% of error per pixel. If we compare it with the average error per pixel made for Rényi index, we could think that in this case HOSVD is not performant. However we want to remark two critical aspects: firstly Rao index takes into account also the values of NDVI, not only their frequencies. Next we remind that at multilinear rank r_5 we are keeping nearly 15% of the total information, as in Table 3.3. Therefore even if results on average are not as good as in Rényi case, we do not exclude the power of this method for Rao index computation. Besides in the best case we have

both original and relative error per pixel near to 11%, which is appreciable, as we will see. Lastly we remark an interesting and unexpected twist. If in the previous analysis T-HOSVD performed better when multilinear rank components were small with respect to ST-HOSVD, in the present case T-HOSVD provides better result than the other algorithm when the first two multilinear rank components are greater that 500. Lastly we highlight that also in this case the relative error is on average greater than the original one. Even in this case this phenomenon is at least in part linked to the two incomplete elements, December 2012 and 2015. Indeed in this case the relative error slightly less than twice the original error.

Now we present some statistics for the original and relative errors of Rao estimates for Earth dataset. As before in the following table epO is a vector with the errors per pixel computed with respect to original estimates, while in epR we store the errors per pixel with respect to relative estimates.

	T-HOSVD					**ST-HOSVD**				
Rank	10	50	100	500	1000	10	50	100	500	1000
$\mathbb{E}[epO]$	0.5442	0.3476	0.2949	0.1845	0.1735	0.5416	0.3467	0.2945	0.1851	0.1752
$\mathbb{V}\mathrm{ar}[epO]$	0.0021	0.0007	0.0007	0.0004	0.0002	0.0018	0.0007	0.0007	0.0004	0.0002
$\mathbb{E}[epR]$	0.5425	0.3445	0.2917	0.175	0.1628	0.54	0.3436	0.2913	0.1756	0.1646
$\mathbb{V}\mathrm{ar}[epR]$	0.0019	0.0006	0.0006	0.0003	0.0001	0.0017	0.0005	0.0006	0.0003	0.0001
$\min epO$	0.4614	0.2943	0.2472	0.1551	0.1523	0.4613	0.294	0.2475	0.1555	0.1539
$\min epR$	0.4625	0.2952	0.2473	0.1509	0.1454	0.4623	0.295	0.2473	0.1513	0.1473
$\max epO$	0.613	0.3992	0.3498	0.2442	0.2295	0.5991	0.3994	0.3497	0.2435	0.2306
$\max epR$	0.6124	0.3859	0.3512	0.2396	0.1895	0.5971	0.3834	0.3511	0.2387	0.1926

Table 3.9: Statistics for Rao index over $N \in n_{R,W,t,j}$.

We immediately notice that even with the greatest multilinear rank components, the average original and relative error per pixel is quite high if compared with the ones in Rényi case. We again remark that 16% error per pixel on average has been obtained using only 16.4% of the total information. However we underline that even in the best case the relative error per pixel is 14.5%, which is quite enough if compared with the minimum error per pixel in the Europe previous case. Besides we notice that the relative error per pixel is slightly lower that the original ones as in the previous Earth case. As before we have that when the first two components of the multilinear rank are lower than 500, ST-HOSVD techniques leads to better results than T-HOSVD. Moreover as in the previous case the variance decreases when the first two multilinear rank components grow.

Before moving to the analysis of error related to Rao index over $n_{N,h,p,j}$ for every $h \in \{E, W\}$, for every $p \in \{T, S\}$ and for every $j \in \{1, \ldots, 5\}$, we show the best case for Rao index for Europe and the worst case for Earth dataset in the following examples. In both the case we use as approximation technique T-HOSVD.

Example 3.2.18. Let's observe the Rao index computed over Europe NDVI of December 2014, in Figure 3.1.

Figure 3.7: Rao index computed over NASA NDVI of December 2014.

We immediately notice that biodiversity is much lower than in the Rényi case, 3.2.12. Besides we underline that even in this case Rao index computed over self-made NDVIs is enough close to its computation over NASA NDVI. Lastly we highlight that in this particular NDVI element the north Europe area is entirely set to missing values, which are shown with white colour. Therefore we can assume that the error is the minimum since this peculiarity.
Next we have in Figure 3.8a the same index computed over self-made NDVI and the approximated Rao estimated at different multilinear rank. It is significant that it is hard to distinguish the Rao index computed over NASA NDVI from the ones computed over the forth and the fifth approximated NDVI.

Next we analyse the worst case of Rao index for the Earth dataset.

Example 3.2.19. Firstly in Figure 3.9 we present Rao index over NASA NDVI of May 2014. As in the Europe case, biodiversity estimated by Rao index has lower values. In the tropical regions for example we have the highest values, which are close to 2. Moreover there are some areas as the Sahara desert which present extremely low biodiversity values.
Next we show in Figure 3.10 the Rao index computed over self-made and approximated NDVIs. It is clear observing the approximated Rao pictures and index of 3.9 that the high error is linked with the overestimation of biodiversity in the North

(a) Relative approximation

(b) Component rank 10 compression

(c) Component rank 50 compression

(d) Component rank 100 compression

(e) Component rank 500 compression

(f) Component rank 1000 compression
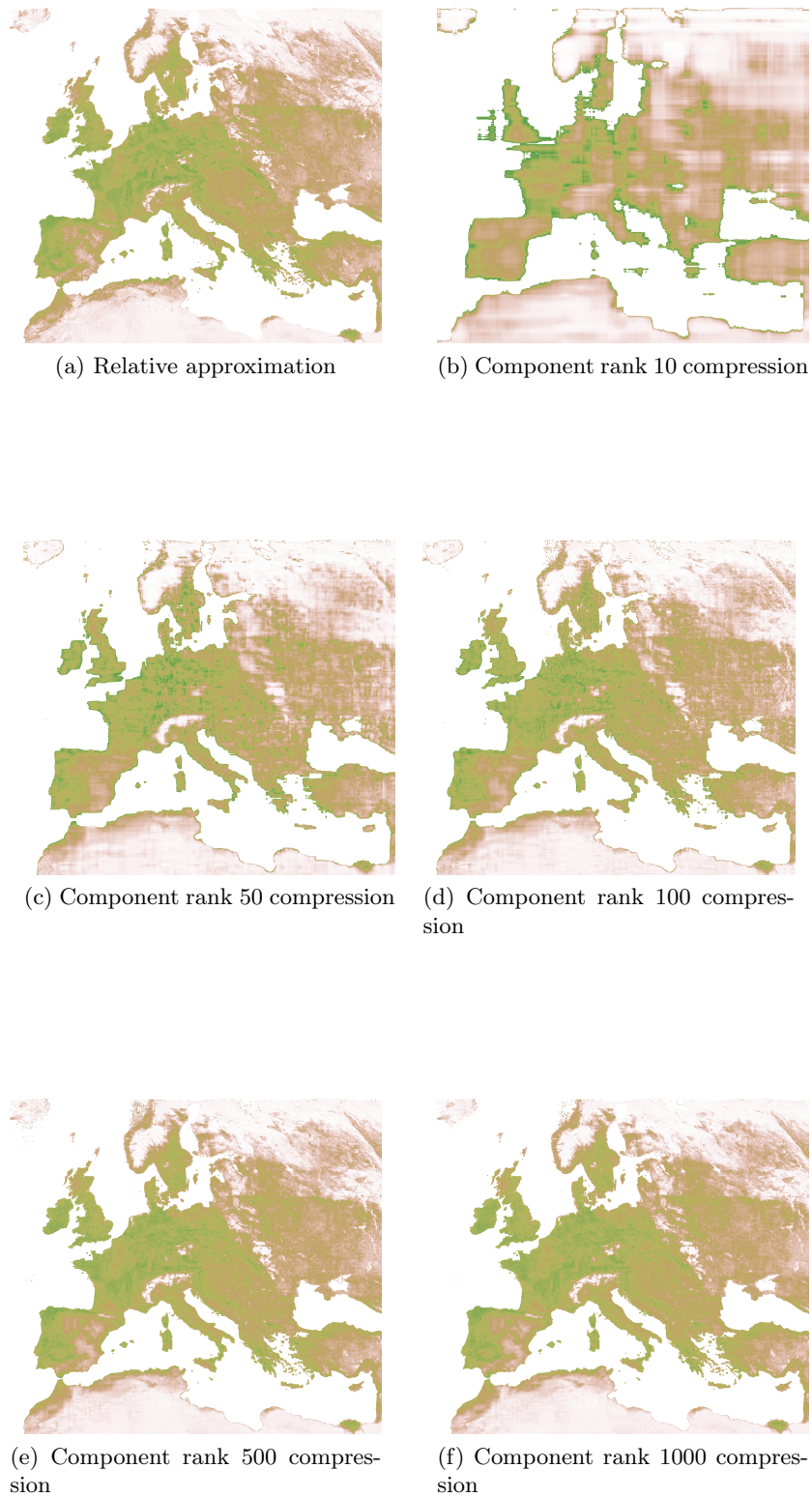
Figure 3.8: Approximation of Rao index for December 2014, from NDVI of $n_{R,E,S,j} \cup n_E$.
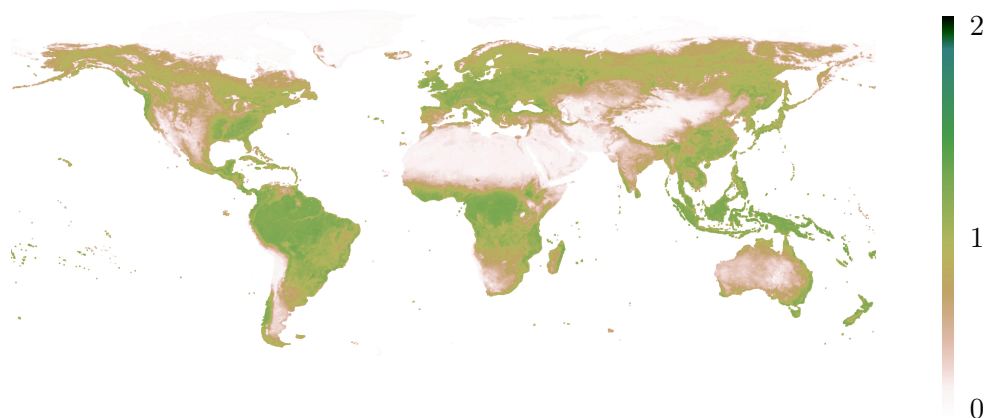
Figure 3.9: Rao index computed over NASA NDVI of December 2014.

(a) Relative approximation

(b) Component rank 10 compression

(c) Component rank 50 compression

(d) Component rank 100 compression

(e) Component rank 500 compression

(f) Component rank 1000 compression
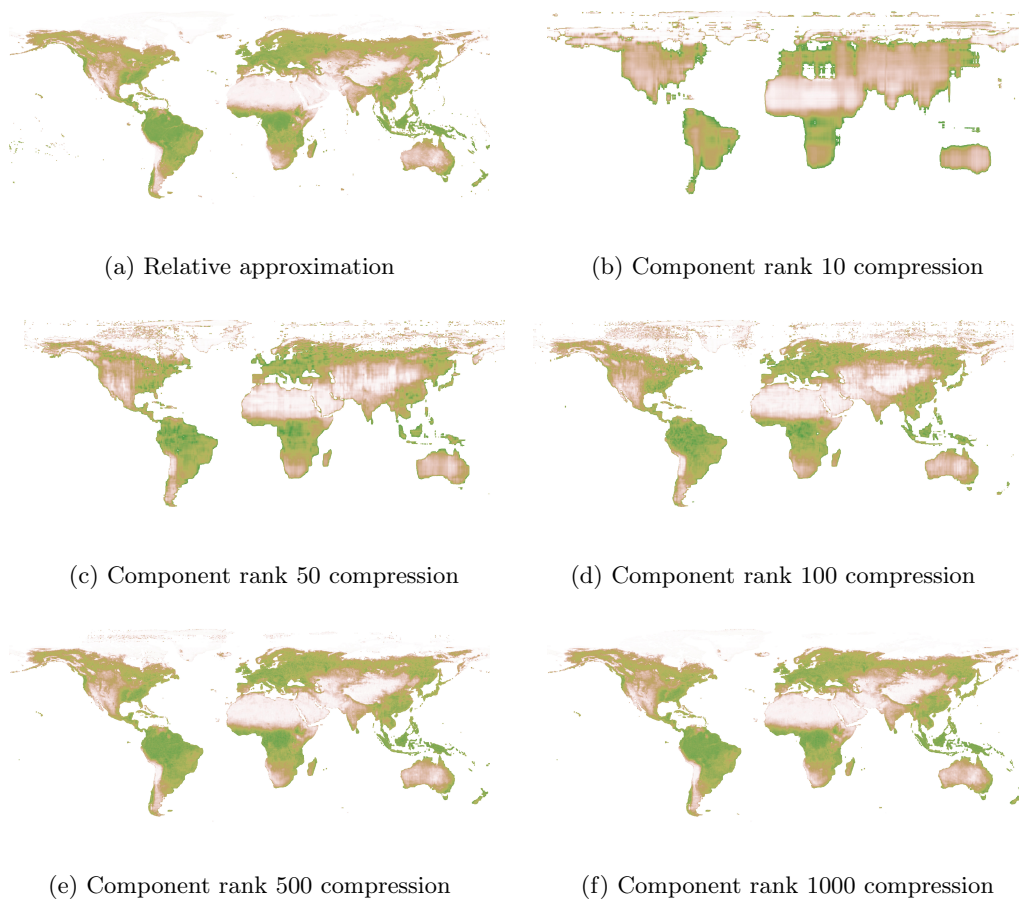
Figure 3.10: Approximation of Rao index for May 2014, from NDVI of $n_{R,W,S,j} \cup n_W$.

pole area. Indeed it is a remarkable phenomenon in the first four approximations. In Figure 3.10f it tends to decrease, but it is still appreciable. Besides also the majority of the Pacific ocean islands are missing in all the approximations. These regions were probably related to error also in Example 3.2.13.

As before, the following step is computing Rao index over approximated NDVI of $\mathcal{n}_{N,h,t,j}$ for every $h \in \{E, W\}$, for every $t \in \{T, S\}$ and $j \in \{1, \ldots, 5\}$.

Definition 3.2.20. Let $\mathcal{R}_{N,h,t,j}$ be the set of Rao index computed over elements of $\mathcal{n}_{N,h,t,j}$ for every $h \in \{E, W\}$, for every $t \in \{T, S\}$ and for every $j \in \{1, \ldots, 5\}$. We decide to call these i_j-*approximated estimates* for every $i_j \in \mathcal{R}$ and for every $j \in \{1, \ldots, 5\}$.

Then we compute the error with respect to the original estimates, i.e.

$$\left\| A_k - C_{k,j} \right\|$$

for every $A_k \in \mathcal{R}_h$ and for every $C_{h,j} \in \mathcal{R}_{N,h,T,j} \cup \mathcal{R}_{N,h,S,j}$, for every $k \in \{1, \ldots, n_h\}$, for every $j \in \{1, \ldots, 5\}$ and for every $h \in \{E, W\}$. Moreover we compute the error with respect to relative estimates, i.e.

$$\left\| B_k - C_{k,j} \right\|$$

for every $B_k \in \tilde{\mathcal{R}}_h$ and for every $C_{h,j} \in \mathcal{R}_{N,h,T,j} \cup \mathcal{R}_{N,h,S,j}$, for every $k \in \{1, \ldots, n_h\}$, for every $j \in \{1, \ldots, 5\}$ and for every $h \in \{E, W\}$.
In conclusion we report some statistics about the original and the relative errors per pixel, stored respectively in vector epO and epR for the Europe dataset.

	T-HOSVD					**ST-HOSVD**				
Rank	10	50	100	500	1000	10	50	100	500	1000
$\mathbb{E}[epO]$	0.6392	0.3641	0.2962	0.2094	0.1969	0.6332	0.3632	0.2953	0.2111	0.1991
$\mathbb{V}\mathrm{ar}[epO]$	0.0049	0.0012	0.001	0.003	0.0029	0.0044	0.0012	0.0009	0.0029	0.0029
$\mathbb{E}[epR]$	0.6396	0.3647	0.2974	0.2115	0.1999	0.6335	0.3638	0.2964	0.2131	0.2021
$\mathbb{V}\mathrm{ar}[epR]$	0.0048	0.0013	0.0011	0.0036	0.0043	0.0044	0.0013	0.0011	0.0035	0.0042
$\min epO$	0.4801	0.2741	0.2204	0.1357	0.1246	0.4777	0.2739	0.2202	0.1378	0.1269
$\min epR$	0.4801	0.2741	0.2204	0.1357	0.1246	0.4778	0.2739	0.2202	0.1377	0.1269
$\max epO$	0.7505	0.4167	0.3497	0.3933	0.3953	0.7403	0.4155	0.3491	0.3946	0.3949
$\max epR$	0.7504	0.4444	0.4259	0.4487	0.5458	0.7402	0.4506	0.4193	0.4472	0.547

Table 3.10: Statistics for Rao index over $N \in \mathcal{n}_{N,E,t,j}$.

Comparing Table 3.8 and 3.10, we notice that the average original and relative error per pixel are greater in the second case. In other words, starting from a tensor with repeated NIR band is not convenient for computing Rao index over Europe elements. Besides we underline that the minimum relative and original error are realised when the starting tensor has RED band repeated. We can not affirm nothing

about the maximum original and relative error, since for $i_j \in \{10, 100, 1000\}$ it is lower when the repeated band is the NIR, while for $i_j \in \{10, 100, 1000\}$ when it is repeated RED band. Lastly we highlight as always for the Europe dataset that the average original error is lower than the relative one. As in the repeated RED case, we remark that ST-HOSVD is convenient when the first two components of multilinear are strictly lower than 500, otherwise T-HOSVD provides better results.

Then we present the final statistics fro original and relative errors of Rao estimates for Earth dataset. As always in vector epO and in epR are respectively stored the original and relative errors per pixel.

	T-HOSVD					ST-HOSVD				
Rank	10	50	100	500	1000	10	50	100	500	1000
$\mathbb{E}[epO]$	0.5366	0.3419	0.2869	0.1754	0.1636	0.5353	0.3417	0.287	0.1761	0.1651
$\mathbb{V}\text{ar}[epO]$	0.0023	0.0008	0.0007	0.0005	0.0003	0.0022	0.0008	0.0007	0.0005	0.0003
$\mathbb{E}[epR]$	0.535	0.3389	0.2835	0.1657	0.1525	0.5337	0.3387	0.2836	0.1664	0.1541
$\mathbb{V}\text{ar}[epR]$	0.0021	0.0006	0.0006	0.0003	0.0001	0.002	0.0006	0.0006	0.0003	0.0001
$\min epO$	0.4481	0.2873	0.2385	0.1431	0.1357	0.4487	0.2881	0.2394	0.1437	0.1376
$\min epR$	0.4486	0.2885	0.2385	0.1382	0.1331	0.4483	0.2887	0.2394	0.1391	0.135
$\max epO$	0.6063	0.3887	0.3458	0.2352	0.226	0.6006	0.389	0.3466	0.2352	0.2267
$\max epR$	0.6057	0.375	0.3478	0.2307	0.1746	0.5997	0.3743	0.3487	0.2314	0.1768

Table 3.11: Statistics for Rao index over $N \in \mathcal{n}_{N,W,t,j}$.

In this case comparing Table 3.9 and 3.11, we notice that starting tensors with repeated NIR band provides on average better results. Indeed for both the decomposition techniques at the fifth approximation the average original and relative error per pixel is around 17.5% in the first case and 16.5% in the second one. This 1% difference between the RED and NIR case is present also in the minimum and maximum relative and original error per pixel for both the decomposition techniques. Besides we underline that also in this case ST-HOSVD is convenient when the first two multilinear components are lower than 100, otherwise T-HOSVD is preferable. In conclusion to these analysis we want to empathise that an average error of 16.5% per pixel is not much, if we remind that we use only 16.4% of the total information available.

Lastly as before, we display the Rao index computed for December 2014 and May 2014, starting from tensors with repeated NIR bands.

Example 3.2.21. Firstly we remark that December 2014 realises the minimum original and relative error, also starting from tensors with repeated NIR band. In Figure 3.11 there are approximated Rao estimates at different multilinear rank, for this NIR repeated case. As before, our eyes hardly perceive differences between Rao computed over NASA or self-made NDVI and over the last three approximated NDVIs.

Example 3.2.22. Firstly we remark that May 2014 do not realise the minimum original and relative error starting from tensors with repeated NIR band. Therefore
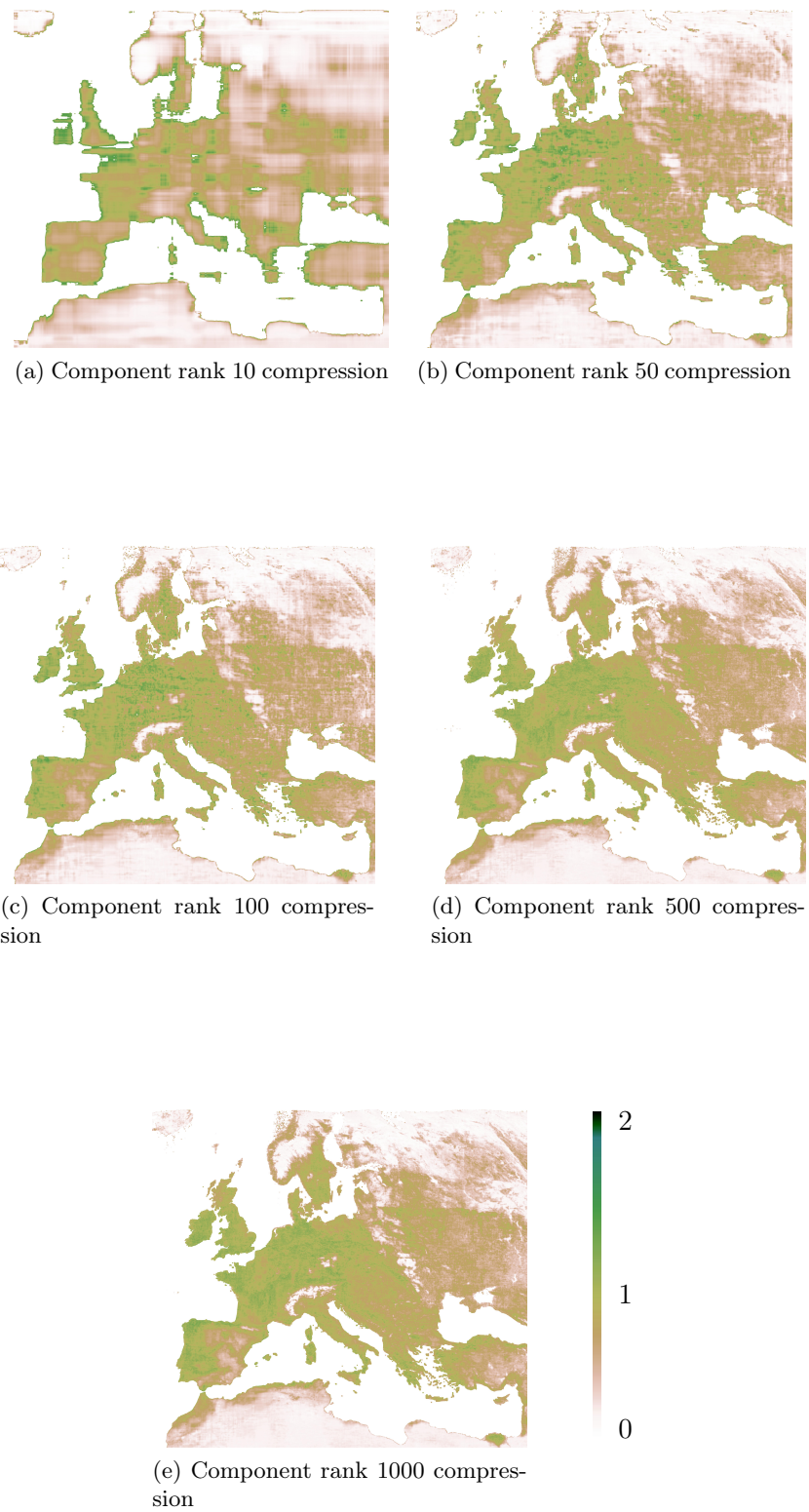
(a) Component rank 10 compression   (b) Component rank 50 compression

(c) Component rank 100 compression   (d) Component rank 500 compression

(e) Component rank 1000 compression

Figure 3.11: Approximation of Rao index for December 2014, from NDVI of $n_{N,E,S,j}$.

these are not the worst approximations in the Earth dataset, in this second case In Figure 3.12 there are approximated Rao estimates at different multilinear rank, for this NIR repeated case. As in Figure 3.10 the problematic areas are the North pole and the Pacific ocean. Indeed we notice that there is a biodiversity overestimation in the Arctic regions, which decreases for increasing multilinear rank components. Moreover even in the fifth approximation there are missing island in the Pacific ocean. However we can not easily notice much difference with respect to Figure 3.10.

3.3   Conclusion

We have shown that the presented approach is extremely convenient for Rényi index estimation to save storage memory. As reported in Tables 3.4,3.6, 3.5 and 3.7 the average error per pixel is around 5.5% for Earth dataset and is close to 7.6% for the Europe's one when using respectively 16.4% and 14.8% of the total tensor information. Moreover we have seen that starting from tensor with repeated NIR bands is more convenient than with RED repeated, in the Rényi case. For Rao case starting with twice NIR rasters is more convenient only for Earth dataset. Lastly we have remarked that in Rényi case T-HOSVD makes on biodiversity estimation lower error than ST-HOSVD with the first two multilinear rank components are relatively low.

In the Rao case we face greater on average original and relative error per pixel. Indeed in Tables 3.8 and 3.9 we have an average error per pixel close to 19.5% and 17% for Europe and for Earth dataset respectively. In this case we have noticed that the roles of T-HOSVD and ST-HOSVD are inverted. Indeed T-HOSVD makes on biodiversity estimation lower error than ST-HOSVD with the first two multilinear rank components are relatively great.

In conclusion we believe that the presented work might help ecologists in their remote sensed plant biodiversity estimation. Indeed fixed a certain accuracy, they can compress through T-HOSVD and ST-HOSVD the tensors with NIR and RED bands, to save storage memory and at the same time computing with fixed tolerance the biodiversity estimates.
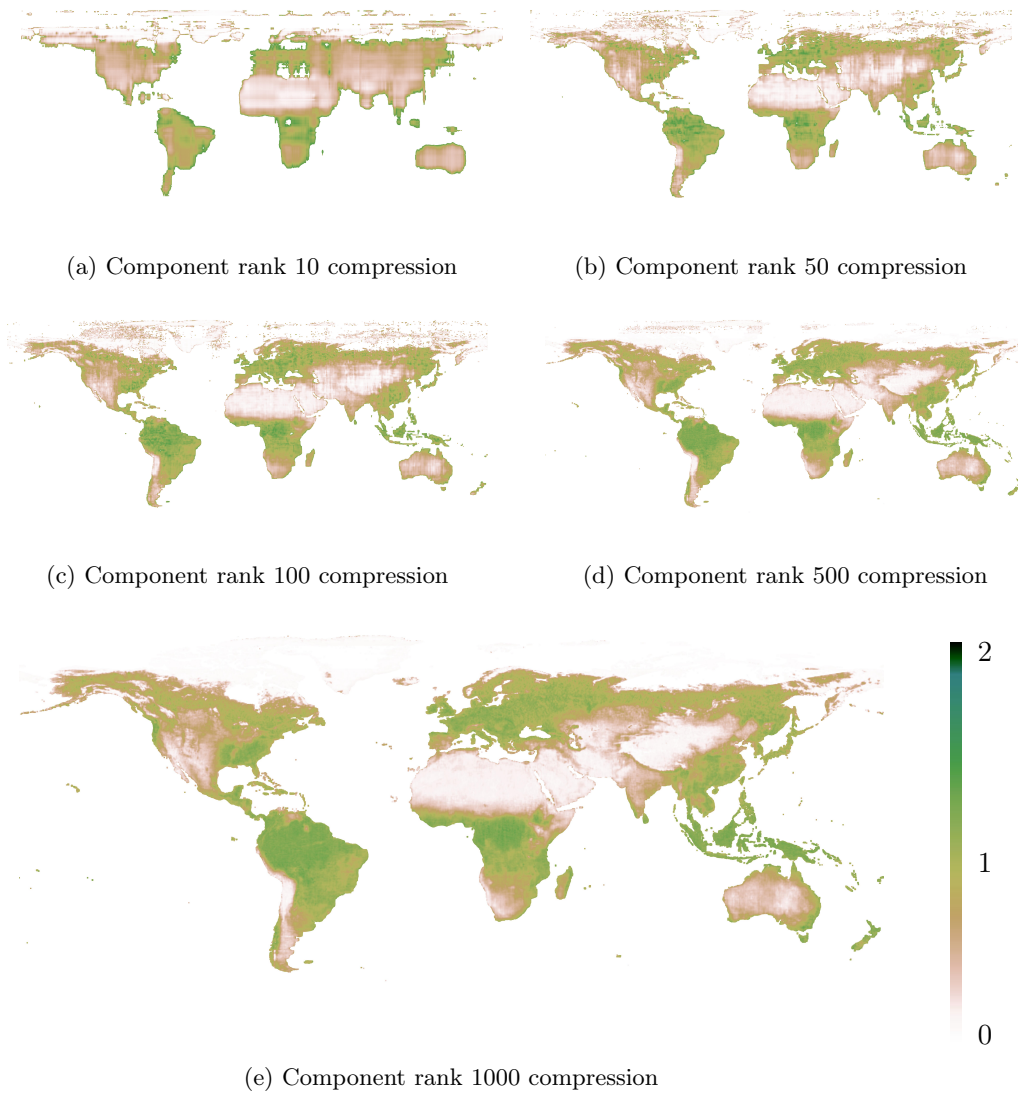
(a) Component rank 10 compression

(b) Component rank 50 compression

(c) Component rank 100 compression

(d) Component rank 500 compression

(e) Component rank 1000 compression

Figure 3.12: Approximation of Rao index for May 2014, from NDVI of $n_{N,W,S,j}$.