Westfälische
Wilhelms-Universität
Münster

› Non-Negative Matrix Factorization for Dynamic Positron Emission Tomography

Fach: Mathematik

Inaugural Dissertation

zur Erlangung des Doktorgrades der Naturwissenschaften
- Dr. rer. nat. -
im Fachbereich Mathematik und Informatik
der Mathematisch-Naturwissenschaftlichen Fakultät
der Westälischen Wilhelms-Universität Münster

vorgelegt von
Ralf Engbers
aus Papenburg

- 2018 -

Angewandte
Mathematik
Münster

Dekan:	Prof. Dr. Xiaoyi Jiang
Erster Gutachter:	Prof. Dr. Martin Burger
Zweiter Gutachter:	Prof. Dr. Carsten Wolters
Tag der mündlichen Prüfung:	13.12.2018
Tag der Promotion:	13.12.2018

Abstract

Coronary heart disease is the most common cause of death in the world and non-invasive medical imaging techniques like positron emission tomography (PET) are important to help diagnose a patient suffering from atherosclerosis as early as possible. Unfortunately, dynamic PET measurements using radioactive water to examine blood flow create challenging image reconstruction and parameter estimation problems.

Non-negative matrix factorization (NMF) has been successfully used as a data analysis tool for many different applications. In this thesis, we will motivate the use of NMF through model-based approaches to dynamic PET reconstruction and examine the results and performance of different NMF algorithms when applied to dynamic PET measurements with very poor statistics.

Zusammenfassung

Koronare Herzkrankheiten sind weltweit die häufigste Todesursache und nicht-invasive medizinische Bildgebungsverfahren wie Positronen-Emissions-Tomographie (PET) sind wichtig um bei der möglichst frühen Diagnose von Patienten zu helfen, die unter Atherosklerose leiden. Leider führt die Qualität der zur Blutflussuntersuchung genutzten dynamischen PET Daten basierend auf radioaktivem Wasser zu schwierigen Rekonstruktions- und Parameterschätzungsproblemen.

Nicht-negative Matrix Faktorisierung (NMF) wird erfolgreich in vielen verschiedenen Bereichen als Datenanalysemethode angewandt. In dieser Arbeit werden wir die Anwendung von NMF auf dynamisches PET über passende Modelle motivieren und untersuchen die Ergebnisse verschiedener NMF Algorithmen anhand von dynamischen PET Messungen mit sehr schlechter Statistik.

Acknowledgements

I would like to express my gratitude towards Martin Burger for giving me the opportunity to work on this Ph.D. thesis and supporting me throughout the entire time. I would also like to thank Frank Wübbeling and the whole Imaging Workgroup for providing the best work environment possible, and Carolin Gietz and Claudia Giesbert, who are always able to sort out any kind of organizational problem.

I acknowledge support of the German Research Foundation (DFG) through the Collaborative Research Center 656 "Molecular Cardiovascular Imaging".

Contents

List of Figures

List of Algorithms

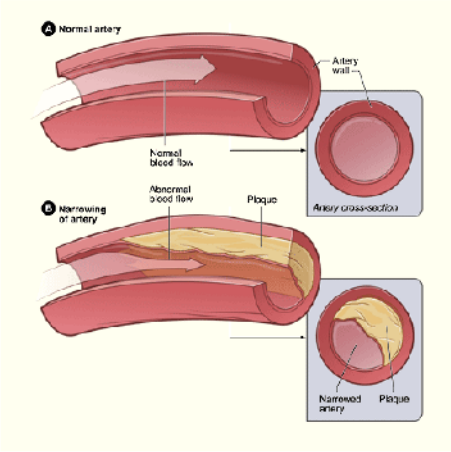
List of Tables

1

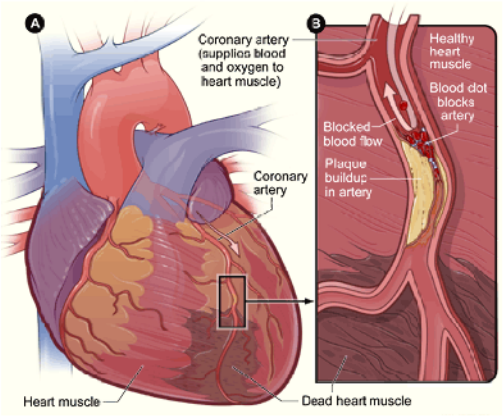
Introduction

According to the World Health Organization, the leading cause of death worldwide is coronary heart disease (CHD), making up 15.9% of all deaths. In 2015, 110 million people have been affected by CHD and although the rates of death have declined in most countries, 8.9 million people have died as a consequence of suffering from a CHD, e.g. from myocardial infarction or sudden cardiac death [Wang et al., 2018]. There are many risk factors associated with CHD, e.g. high blood pressure, diabetes, poor diet, lack of exercise and excessive alcohol, but most importantly smoking (36%) and obesity (20%).

One of the main reasons for CHD is atherosclerosis, the buildup of a waxy substance inside the coronary arteries called *plaque* (see Fig. 1.1a.). Atherosclerosis often begins in childhood and typically no symptoms are noticeable for decades. In the beginning, the arteries enlarge at the plaque locations without reducing the blood flow and even plaque ruptures do not necessarily cause any complications if the arteries are still wide enough. Symptoms start to appear when arteries are narrowed to the degree that the blood flow is constrained and the heart muscle is not provided with enough oxygen, leading to chest pain or pressure known as *angina*. The combination of narrowed arteries and ruptured plaque can cause a complete blockage of the affected artery and lead to severe cardiovascular disorders such as a stroke or a heart attack (see Fig. 1.1b). The



(a) The image shows a normal artery above and a narrowed artery due to plaque buildup below.



(b) On the left the dead heart muscle is shown below a blocked coronary artery. On the right a cross-section of the coronary artery

Figure 1.1.: Visualization of atherosclerosis and the damage to the heart muscle after a heart attack as a result of a blocked artery. Source: NATIONAL HEART, LUNG, AND BLOOD INSTITUTE [2018].

slow progress in combination with the severity of the effect of a blocked artery demand methods to diagnose and treat a patient as early as possible.

Positron emission tomography (PET) is an imaging technique that allows us to visualize physiological activities with the help of radioactive tracers. A tracer is indiscernible for the body from its non-radioactive counterpart, takes part in the normal body metabolism and can be followed by its radioactive decay. To visualize myocardial blood flow, we can use dynamic PET measurements using radioactive water $H_2{}^{15}O$, that mixes perfectly with the blood pool. The low half-life of oxygen-15 keeps the radioactive burden small for the patient, but also leads to low-quality measurements which make image reconstruction and the estimation of physiological parameters very challenging.

Most approaches try to solve this problem by including some form of apriori information to reduce the complexity. Typically, a kinetic model is used to either employ a parameter fitting to the noisy image reconstructions or as a motivation to find parameters of the model via a variational formulation. There have also been attempts in using data analysis techniques like principal component analysis (PCA) and independent component analysis (ICA) to reduce the dimensionality

of the data. While they do, with some success, reduce the noise in the data, the respective components do not have an obvious physiological interpretation.

In this thesis, we will use a data analysis technique, called non-negative matrix factorization (NMF), which has been used for a variety of problems in the last two decades. NMF reduces the dimensionality like PCA/ICA but leads to more useful non-negative components. Particularly, using NMF we can separate the temporal and the spatial dimension from each other, which gives us a nice interpretation in terms of a spectral approach to kinetic modeling.

We will start this thesis in Chapter 2 by providing the necessary medical and mathematical background for the dynamic PET problem of myocardial blood flow imaging. In Chapter 3 we introduce the concept of kinetic modeling and describe how we can use a spectral approach to define a general linear operator describing the model. We also refer to a different approach based on a non-linear model which provides a consistent interpretation for the NMF components. Chapter 4 briefly describes the problem of using PCA on non-negative physiological measurements and introduces the NMF problem. We close the chapter by describing four algorithms for solving NMF. Finally, we present numerical results in Chapter 5, first in the optimal case of exact data without any noise, and then trying to identify components for very noisy data, emulating the noise level of a real dynamic PET dataset. Chapter 6 concludes this thesis.

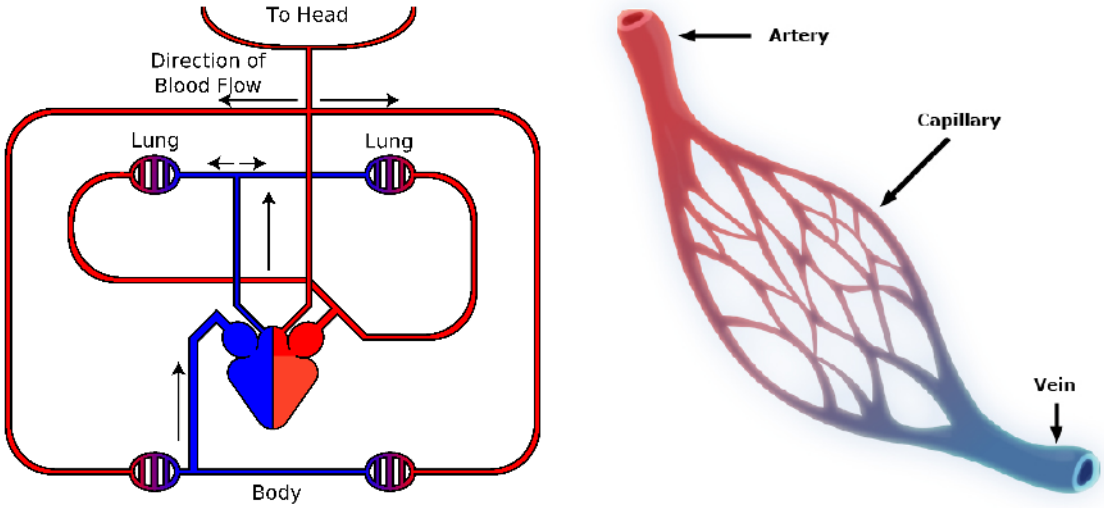
2

MEDICAL IMAGING OF MYOCARDIAL BLOOD FLOW

This chapter provides the necessary medical and mathematical background needed in the course of this thesis. We start in Section 2.1 by summarizing the medical background to better understand the medical side of the problem of myocardial blood flow quantification. We will introduce the basics of positron emission tomography in Section 2.2 and describe a mathematical model for PET in Section 2.3. In Section 2.4 we look at the specific challenges that occur with dynamic PET.

2.1. Cardiovascular System and Perfusion

The human circulatory system consists of two parts, with the heart muscle being the pump in between driving the whole system. One part is the pulmonary circulation which transports deoxygenated blood away from the right ventricle of the heart to the lungs, where the blood is oxygenated and returned to the left ventricle of the heart. The second part, the systemic circulation, transports the oxygenated blood to the rest of the body, assuring the supply of the body with



(a) Circulatory System. Source: B. [2005] **(b)** Capillary. Source: COMMUNITY EMERGENCY RESPONSE TEAM [2013]

Figure 2.1.: Schematic representations of the circulatory system and a capillary.

oxygen and returning the deoxygenated blood back to the right ventricle, where the circulation starts anew. Figure 2.1a shows a schematic representation of the whole circulatory system.

Arteries are the blood vessels responsible for the transportation of the oxygenated blood to the organs, while veins transport the deoxygenated blood back to the heart. The actual supply of the organs with the oxygen, however, happens in the smallest vessels in the body, the capillaries. This is true also for the heart muscle itself. Capillaries are tubes with a one cell thick wall with diameters ranging from 5 to 10 micrometers. A simplified illustration is shown in Figure 2.1b.

In general, the spatial resolution of PET is not good enough to show the capillaries directly, which is even truer for dynamic PET as we will discuss below. We can, however, visualize the left ventricle of the heart muscle, which is tissue that consists of capillaries and where oxygen extraction takes place, i.e. we can visualize perfusable tissue. Looking at the physical quantities, blood flow is measured as the volume of blood over time (ml/min) while perfusion is the volume of blood per unit mass (ml/min/mg). Now, if we are able to connect the perfusion to the perfusable tissue, we are also able to say something about the

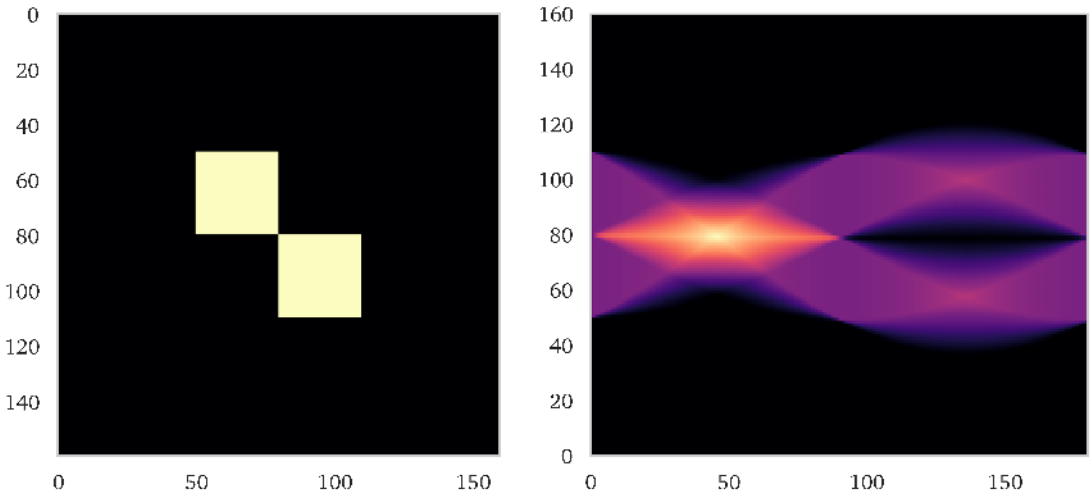
Figure 2.2.: Image of a typical PET facility with an ECAT Exact HR+ PET scanner. Source: LANGNER [2003b]

blood flow through the blood vessels in the heart itself. Kinetic models can be used to model this relation and we will describe how in the next chapter.

2.2. Positron Emission Tomography

PET is a functional imaging method that makes biochemical and physiological functions visible by displaying the distribution of a radiopharmaceutical, a so-called tracer. Depending on the tracer, it reacts with the bodies molecules in different ways, e.g. radioactive glucose is commonly used to examine tumors since it accumulates in areas with high metabolic activity.

As the name suggests, the radioisotopes used in PET undergo positron emission decay, also known as β^+ decay. In the decay process, it emits a positron, the positively charged antiparticle of the electron. The positron travels inside the body a very short distance (usually less than 1 mm, but depending on the specific isotope) until it loses enough kinetic energy to interact with an electron which leads to an electron-positron annihilation. In this annihilation process, both particles get annihilated and two gamma photons are created, traveling in approximately opposite directions. The PET scanner detects these gamma photon



(a) Original image with size 160×160 pixels.

(b) Sinogram with the projection angle on the x-axis and the respective 1D projection along the y-axis.

Figure 2.3.: Sinogram Example. Figure (a) shows the original image and (b) shows the respective sinogram.

pairs as coincidence events. An example of a PET scanner is shown in Figure 2.2.

A typical way to store the coincidences is to group them according to their angle, which means that we get a 1D projection of the 2D object for every angle. The resulting collection is called *sinogram* since under this representation, a single point source will appear as a sine wave, and multiple objects as blurred, superimposed sine waves with different phases and amplitudes. A simple geometric example can be found in Figure 2.3 while a schematic representation of the PET measuring process is shown in Figure 2.4.

2.3. Mathematical model for PET

As mentioned above the gamma photons travel at (almost) 180 degrees to each other, which means that the source of the decay event is located somewhere on a straight line, the *line of response* (LOR). Thus, instead of measuring the tracer density directly we can only measure the projection of the density onto lines.

The mathematical equivalent to this projection is the *Radon transform* (see Def. 2.1), an integral transform introduced in 1917 by the Austrian mathematician

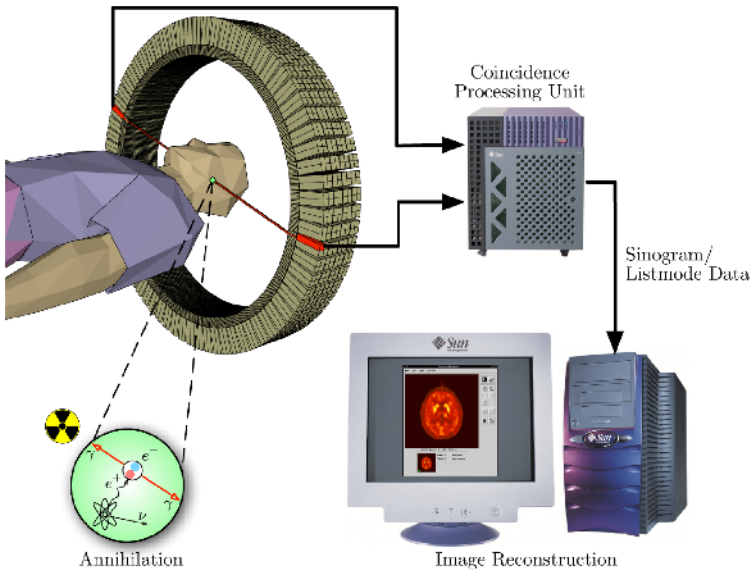


Figure 2.4.: Schematic representation of the different steps of a PET scan. Source: LANGNER [2003a]

Johann Radon [RADON, 1917]. We can understand the sinogram PET data as the Radon transform of the function describing the unknown tracer distribution. While the Radon transform is defined for two-dimensional functions, there exist generalizations to higher dimensions. The most important one for tomography is the *Ray transform* (see Def. 2.2), which integrates over lines instead of hyperplanes. In 2D, both transforms coincide.

Definition 2.1 (Radon Transform). *Given a function u on $\mathbb{R}^n$, an angle $\theta \in S^{n-1}$ and a distance $s \in \mathbb{R}$, then the Radon transform $(Ku)(\theta, s)$ is defined as*

$$(Ku)(\theta, s) = \int_{x \cdot \theta = s} u(x)dx \tag{2.1}$$

Definition 2.2 (Ray Transform). *Given a function u on $\mathbb{R}^n$, an angle $\theta \in S^{n-1}$ and a point $x \in \theta^{\perp}$, then the Ray transform $(Ku)(\theta, x)$ is defined as*

$$(Ku)(\theta, x) = \int_{\mathbb{R}} u(x + t\theta)dt \tag{2.2}$$

Given sinogram data f, the PET image reconstruction problem is to recover the function u, i.e. the radiotracer distribution inside the patient, which we can write

as

$$f = \text{Pois}(Ku). \tag{2.3}$$

Pois$(\cdot)$ simply expresses the fact that radioactive decay is a stochastic process following a Poisson distribution (see Def. 2.3) and thus, we can understand our measurements as random samples of a Poisson distribution.

Definition 2.3 (Poisson Distribution). *Let a* $\lambda \geq 0$ *and* $k \in \mathbb{N}_0$*, the probability of observing* k *events in an interval is given by*

$$Pois_\lambda(k) = \frac{\lambda^k}{k!} e^{-\lambda}, \tag{2.4}$$

with both the expected value and the variance being equal to λ*.*

Radon also provided an analytical formula for the inverse transform, which is the basis of the filtered backprojection formula (FBP) [Natterer, 2001]. FBP is based on the intuitive idea to simply take the 1D projections and project them back into the image domain. In the unfiltered variant, this leads to a blurred version of the original image since every projection is spread over the whole image domain. Using a suitable high-pass filter, an unblurred result can be achieved.

However, the Radon inversion is an ill-posed problem [Natterer and Wübbeling, 2001] and FBP might not be able to provide acceptable results, especially with incomplete or noisy data. While computational more expensive, iterative reconstruction methods have become the norm in image reconstruction problems due to much better artifact and noise handling.

To derive the most commonly used algorithm, the *EM algorithm*, we will start by describing the stochastic model for emission tomography introduced by Shepp and Vardi [1982]. Since the stochasticity in emission tomography is very pronounced due to the small number of events, the idea is to include the apriori information into the model that the number of events in a certain point is a Poisson variable.

We start by discretizing the reconstruction region into n pixels (or voxels in

the three dimensional case), with the Poisson variable φ_j denoting the number of events in pixel j, for $j \in [1, \dots, n]$. The expected value of φ_j is $u_j = E(\varphi_j)$, which is the value that we are interested in recovering. Let $\gamma = (\gamma_1, \dots, \gamma_n)$ be a random variable with γ_i representing the number of events detected by detector i, and let the measurement $f = (f_1, \dots, f_n)$ be a realization of γ. If we now define a (n, m)-matrix K with entries k_{ij} to represent the probability that an event in pixel j is detected in detector i, we get

$$E(\gamma) = Ku. \tag{2.5}$$

We can now recover u by maximizing the likelihood function

$$L(f) = \prod_{i=1}^{n} \frac{(Ku)_i^{f_i}}{f_i!} e^{-(Ku)_i}. \tag{2.6}$$

In stead of maximizing 2.6, we can also maximize the log likelihood function

$$\ell(u) = \sum_{i=1}^{n} \left(f_i \log(Ku)_i - (Ku)_i \right), \tag{2.7}$$

which is obtained by taking the logarithm of L and omitting the constant term $[\log(f_i!)]$.

Here we can see that the Kullback-Leibler divergence arises as a natural choice for the data fidelity term in the variational problem since we are dealing with data that are sampled following a Poisson process. Going from the discrete to the continuous model and adding the term $[f \log(f) - f]$, we get the Kullback-Leibler divergence as given in definition 2.4.

Definition 2.4 (Kullback Leibler divergence). *Let $Ku(x) > 0$ and $f(x) > 0$ for all $x \in \Omega$. Then the Kullback-Leibler divergence is given by*

$$D_{KL}(Ku(x), f(x)) = \int_{\Omega} Ku \log \frac{Ku(x)}{f(x)} + f(x) - Ku(x) dx. \tag{2.8}$$

We can now formulate the image reconstruction problem as an optimization

problem, i.e. we are looking for the argument that minimizes

$$\hat{u} = \arg\min_u D_{KL}(Ku, f) + R(u), \tag{2.9}$$

with $R(u)$ representing any additional regularization on the solution $\hat{u}$. For example, MÜLLER [2013] analyzed the use of (higher order) total variation regularization in the context of PET image reconstruction. However, oftentimes this minimization problem is solved without additional regularization. In that case, the simplest way solve (2.9) iteratively is the *EM algorithm*.

Definition 2.5 (EM algorithm). *Let $K, u > 0$ elementwise and let $u^0 > 0$. Then the EM algorithm is given by*

$$u^{k+1} = u^k K^T \frac{f}{Ku^k}, \quad k = 0, 1, \ldots \tag{2.10}$$

Since the entries of K are probabilities and f is our non-negative data, the assumptions for the EM algorithm are obviously satisfied and the algorithm itself can easily be derived by checking the Kuhn-Tucker conditions for ℓ from (2.7). We have

$$\nabla \ell(u) = K^T \left(\frac{f}{Ku} - \mathbb{1} \right) \tag{2.11}$$

where $\mathbb{1}$ is the vector containing only 1's. Each global maximum u of ℓ has to satisfy

$$uK^T \left(\frac{f}{Ku} - \mathbb{1} \right) = 0, \tag{2.12}$$

which leads to the simple iteration scheme

$$u_{k+1} = \frac{u_k}{K^T \mathbb{1}} K^T \frac{f}{Ku}. \tag{2.13}$$

In fact, the EM algorithm described in Algorithm 1 is an instance of the EM algorithm by DEMPSTER et al. [1977], which has been shown by SHEPP AND

VARDI [1982]. VARDI et al. [1985] then proved convergence.

Algorithm 1 EM algorithm

Input: PET data f, PET matrix K
Output: reconstructed image u

1: initialize $u_0 > 0$
2: **for** $k = 1, 2, \ldots$ **do**
3: compute $u_{k+1} = \frac{u_k}{K^T \mathbb{1}} K^T \frac{f}{Ku}$
4: **end for**

2.4. Dynamic Positron Emission Tomography

In our case, we are interested in a dynamic process, i.e. myocardial blood flow, and we need a suitable tracer. A good candidate for these kinds of measurements is radioactive water $H_2{}^{15}O$ since it behaves exactly like water and therefore is highly diffusible in the blood pool of the patient. Compared to a normal water molecule, the oxygen-16 isotope is replaced by the radioactive oxygen-15. The downside of oxygen-15 is its very short half-life of 122.24 seconds, which results in low measurement times and poor statistics. In fact, due to the fast decay, the measurement will start even before the tracer is injected to make sure that no coincidences are missed.

Furthermore, we can not simply collect all coincidences for the image reconstruction, since we would not be able to see any dynamics. We have to separate the sinogram data into smaller temporal frames. The length of the frames usually is chosen such that each frame consists of a similar amount of coincidences. Since the activity is much higher at the beginning of the scan, the first time frame might encompass 5 seconds while a time frame at the end of the scan could be as long as 30 seconds.

The obvious problem with this separation into frames is that each frame consists of even fewer coincidences than the already low count $H_2{}^{15}O$ scan. This leads to noisy, low-quality images if we reconstruct each frame independently with the EM algorithm. Nonetheless, these images are usually the basis for any kind of

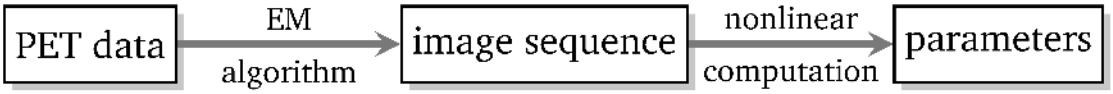


Figure 2.5.: Blood flow quantification process. From the measured data f an image sequence is reconstructed using the EM algorithm. These images are then used as a basis to compute the parameters of a suitable compartment model.

parameter estimation.

The state of the art for blood flow quantification is using this reconstructed image sequence. The images are being masked to include only the heart and are segmented to identify the myocardial tissue and both ventricles, which is often done by computing the principal components of the images to get *factor images* [BARBER AND MARTEL, 1992; BARBER, 1980]. Finally, a non-linear least squares fit to a suitable model is performed to estimate the relevant model parameters (see e.g. SCHÄFERS et al. [2002] and references therein).

This method (see Figure 2.5) obviously carries with it all the negative side effects mentioned already in the context of dynamic PET reconstruction. The quality of the parameter fitting does depend on the quality of the reconstructed images. We can use additional regularization with the EM algorithm, but we still have reconstructions that neglect the temporal information inherent in the dataset.

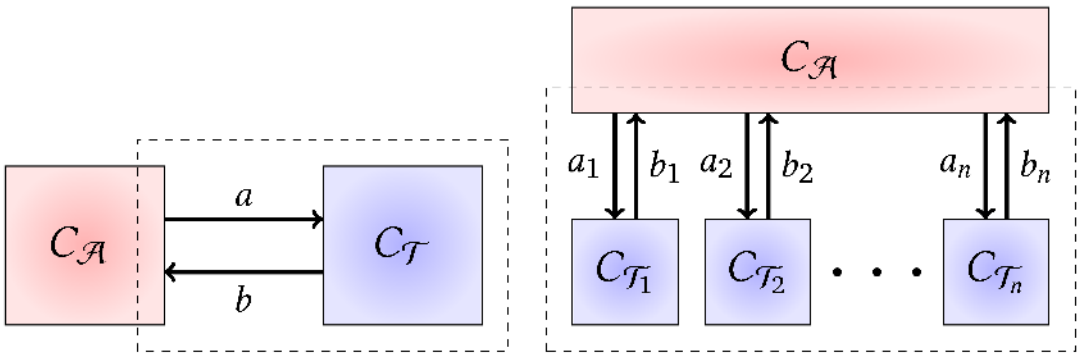
In the next chapter, we will try to remedy these problems. We will introduce the kinetic models that are also used in the parameter fitting mentioned above and construct an operator that decouples the temporal and spatial information. This way we can hopefully identify the important tracer dynamics while having much more control over the reconstruction of the spatial information.

3

Model-based image reconstruction

The reconstruction problem for dynamic PET is difficult due to the quality of the measurements, therefore, apriori information in form of a mathematical model is used to improve the results. Since the information content of PET data typically is inadequate to support very complex models, we are compelled to use a model that simplifies the true physiology to a usable degree. Kinetic models are being used extensively in pharmacokinetics to describe the interaction of chemicals in a living organism and thus appear to be appropriate to describe myocardial blood flow.

In Section 3.1 we describe the basic concept behind kinetic modeling and compartmental models. We will motivate the dynamic PET image reconstruction problem in Section 3.2 in terms of the operator we derived from the compartment model and also introduce a non-linear blood flow model in Section 3.3 and examine the connection between both models.



(a) One-tissue compartment model. **(b)** Multiple-tissue compartment model.

Figure 3.1.: Linear compartment models, with the input function $C_{\mathcal{A}}$, the tissue tracer concentrations $C_{\mathcal{T}}$ and $C_{\mathcal{T}_i}$, and exchange variables a, b and a_i, b_i, respectively.

3.1. Kinetic Modeling

The reconstructed PET images consist of superimposed signals, but we want to be able to examine a single signal by itself. A mathematical model describing the dynamics of the tracer enables us to relate the possible physiological or chemical states to the PET images. These states are called *compartments*.

A typical assumption in compartmental modeling is the uniform distribution of the tracer inside a compartment since we use compartments to describe the concentration change of a tracer over time and not in space. Each spatial region we are interested in, e.g. a single pixel or voxel, therefore consists of unique, homogeneous states, with each state being assigned a compartment. The connection between the compartments and the tracer concentration change inside the compartments can be described by linear, first order, ordinary differential equations (ODE).

Figure 3.1a shows the simplest compartment model, the two-compartment model. In nuclear medicine, it is custom for the first compartment to represent the tracer concentration in the plasma or blood pool. However, the tracer concentration in the blood is not considered to be a compartment, since it is not part of the model being calculated but rather a measured quantity usually called the *input function*. Due to this, the standard two-compartment model is also called the *one-tissue compartment model*.

Given an input function $C_{\mathcal{A}}$, the corresponding ODE to calculate the tracer concentration in the tissue is

$$\frac{dC_{\mathcal{T}}(t)}{dt} = aC_{\mathcal{A}}(t) - bC_{\mathcal{T}}(t). \tag{3.1}$$

Since we are dealing with $H_2{}^{15}O$ PET scans, the measurement is started before the tracer is injected due to its short half-life. This means the initial condition is zero, $C_{\mathcal{T}}(0) = 0$, and we can solve the ODE (3.1), obtaining

$$C_{\mathcal{T}}(t) = a \int_0^t C_{\mathcal{A}}(t) e^{-b(t-\tau)} d\tau. \tag{3.2}$$

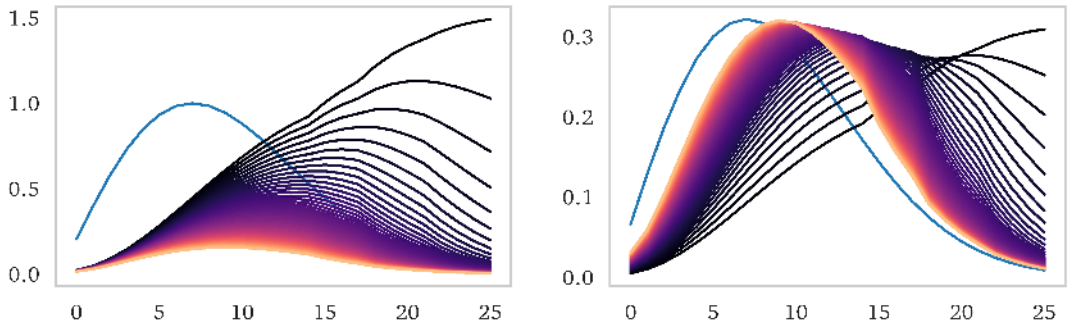
There are several versions of multiple compartment models. For example, two-tissue compartment models are used to describe cerebral glucose use with [^{18}F]FDG and three-tissue compartment models are common for brain receptor studies. However, CUNNINGHAM AND JONES [1993] have proposed an alternative to using a specific compartment model by applying spectral analysis to general linear multiple compartment models. The time-activity curve (TAC) for the tissue compartment is modeled as the convolution of the input function with several exponential terms,

$$C_{\mathcal{T}}(t) = \int_0^t C_{\mathcal{A}}(\tau) \sum_{i=1}^{N} a_i e^{-b_i(t-\tau)} d\tau, \tag{3.3}$$

with N being the (large) maximum number of terms being included in the model and the b_i's being fixed to cover a suitable spectral range. The idea is that most of the coefficients will be equal to zero and there are only a few positive peaks in the spectrum. A representation of the model (3.3) is shown in Figure 3.1b.

We can understand (3.3) as a natural extension of (3.2) with both formulations being equal for $N = 1$.

READER et al. [2007, 2006] have used this spectral analysis technique to decrease the noise levels in dynamic PET image reconstruction.



(a) Basis functions according to (3.5).

(b) Basis functions normed with the 2-norm.

Figure 3.2.: A set of 61 basis functions generated according to Equation (3.5) with parameters $b_i \in [0, 6]$ and step size 0.1. The input curve is shown in blue. Figure (a) shows the computed basis functions and Figure (b) shows the basis functions normalized using the L_2 norm.

3.2. Linear model operator

We now want to apply the model given in (3.3) to each pixel in our image domain and add a spatial dimension to the coefficients a_i. Assuming the input function and the coefficients $a_i(x)$ are known, and the b_i's are given, we can construct an image sequence,

$$u(x,t) = \int_0^t C_{\mathcal{A}}(\tau) \sum_{i=1}^{N} a_i(x) \mathrm{e}^{-b_i(t-\tau)} d\tau. \tag{3.4}$$

By rearranging the terms we can define temporal basis functions as

$$\tilde{b}_i(t) = \int_0^t C_{\mathcal{A}}(\tau) \mathrm{e}^{-b_i(t-\tau)} d\tau, \tag{3.5}$$

with Figure 3.2 showing an example of a set of basis functions. We are now able to map the spatial coefficients onto an image sequence using a linear operator B,

$$[B(a)](x,t) := \sum_{i=1}^{n} a_i(x) \tilde{b}_i(t). \tag{3.6}$$

Instead of having to solve the image reconstruction problem (2.3), the problem

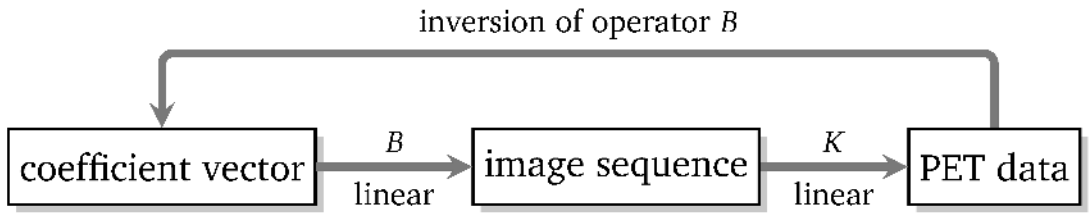


Figure 3.3.: Model based inversion problem. The model operator B maps a coefficient vector of the compartment model onto an image sequence u. Using the linear PET operator K the images sequence is transformed into the sinogram space. The inversion of this process (i.e. the inversion of B) allows us to estimate the coefficients directly from the data.

is now transformed into finding the coefficient vector a such that

$$Pois(KBa) = f. \tag{3.7}$$

This problem formulation is depicted in Figure 3.3 and has the advantage that we can use all the information of the PET dataset f to reconstruct the coefficient vector directly instead of estimating them form reconstructed low quality images.

To solve problem (3.7), HEINS [2014] used $\ell^{1,\infty}$ regularization to promote local sparsity on the coefficients, since the coefficient vector should be sparse regarding the basis functions for every pixel. In this approach works well to identify the support of different regions, however, it is difficult to identify the true sparse coefficients. The problem arises due to the fact that some of the basis functions are very similar to each other (see Figure 3.2) and the true coefficients belonging to a specific basis function might get distributed on two or more very similar basis functions. For a detailed analysis of the method and experimental results, we refer to HEINS [2014].

We will focus on another feature of the operator B. As we see in (3.6), the operator separates the spatiotemporal image sequence $u(x,t)$ into two parts, the space dependent coefficients $a_i(x)$ and the temporal basis functions $\vec{b}_i(t)$. We will use exactly this separation in the next chapter as the motivation to use data analysis techniques.

The decoupling of the space and time dimension also has another immediate advantage. As mentioned in Section 2.4, the standard approach to reconstructing

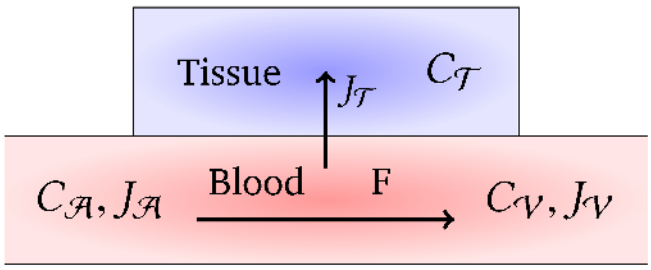


Figure 3.4.: One-tissue compartment model with flow.

a dynamic PET dataset results in low-quality images, since we need to separate the measured data into temporal frames. But if we are able to extract the temporal information from the dataset we are left with a number of factors consisting of spatially dependent coefficients in the sinogram space. We can then use the EM algorithm (see Algorithm 1) on these factors to reconstruct the coefficient vectors and get the coefficients in the image space, which gives us a way to visually interpret the coefficients with regards to the corresponding temporal basis functions (see Chapter 5).

3.3. Non-linear kinetic model

For the rest of this thesis we will use the linear model operator based on the spectral analysis technique introduced in Section 3.2, however, we close this chapter describing a non-linear model developed by BENNING et al. [2010, 2008] to motivate another interpretation of the linear model.

We want to apply the compartment model from Section 3.1 on the smallest scale, with the blood compartment representing a single capillary and the tissue compartment representing the surrounding tissue in the immediate neighborhood. While the tracer flows through the capillary some of it is removed from the blood via extraction into the tissue and the tracer concentration in the blood is being reduced.

We model this behavior by extending the one-tissue compartment model as shown in Figure 3.4. The tracer enters the capillary via the arterial side with a concentration $C_{\mathcal{A}}$ and leaves the capillary with concentration $C_{\mathcal{V}}$ on the venous

side. The blood flows with the flow rate F.

If we assume that the compartment is in a steady state, Fick's principle states, that the flux of blood entering the compartment has to equal the flux of blood leaving the compartment. The arterial tracer flux is just the arterial tracer concentration multiplied by the flow rate, $J_{\mathcal{A}}(t) = FC_{\mathcal{A}}(t)$, and the venous tracer flux analogously $J_{\mathcal{V}}(t) = FC_{\mathcal{V}}(t)$. Fick's principle now simply states, that $J_{\mathcal{A}}(t) = J_{\mathcal{T}}(t) + J_{\mathcal{V}}(t)$. Again, we can derive an ODE describing the net tracer flux into the tissue,

$$J_{\mathcal{T}}(t) = \frac{dC_{\mathcal{T}}(t)}{dt} = J_{\mathcal{A}}(t) - J_{\mathcal{V}}(t) = F(C_{\mathcal{A}}(t) - C_{\mathcal{V}}(t)). \tag{3.8}$$

Since we are dealing with a highly diffusible tracer, the tissue tracer concentration $C_{\mathcal{T}}(t)$ and the venous tracer concentration $C_{\mathcal{V}}(t)$ will quickly equilibrate. Thus, we can introduce the so-called partition coefficient $\lambda = \frac{C_{\mathcal{T}}}{C_{\mathcal{V}}}$, which usually is set between 0.91 and 0.92 [IIDA et al., 1988].

Combining the partition coefficient with the ODE (3.8), we get

$$\frac{dC_{\mathcal{T}}(t)}{dt} = F\left(C_{\mathcal{A}}(t) - \frac{C_{\mathcal{T}}(t)}{\lambda}\right). \tag{3.9}$$

Again, we can solve the ODE (3.9) explicitly for $C_{\mathcal{T}}$ using the zero initial condition and receive

$$C_{\mathcal{T}}(t) = F\int_0^t C_{\mathcal{A}}(t)\mathrm{e}^{-\frac{F}{\lambda}(t-\tau)}d\tau. \tag{3.10}$$

Due to the low resolution of the PET scans we can not determine the perfusable tissue exactly but need to choose larger regions for the compartments. This, however, creates new problems that we are briefly addressing here following the ideas of IIDA et al. [1988].

First, since we are enlarging the compartments to make sure that perfusable tissue is actually contained in the compartment, we need to introduce a new variable, the *tissue fraction* $r(x)$, that estimates the fraction of purely perfused tissue in the compartment.

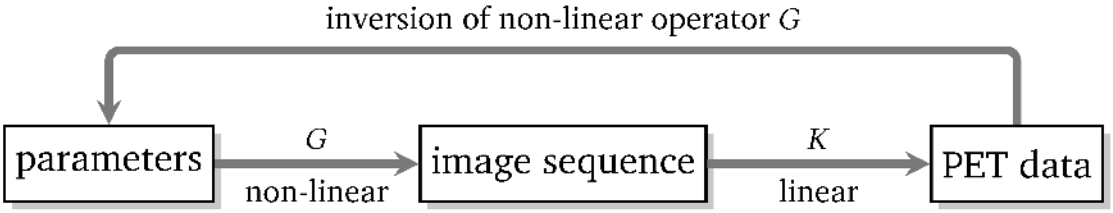


Figure 3.5.: Model based non-linear inversion problem. The model operator G couples the parameters of the compartment model with an image sequence u. Using the linear PET operator K the image sequence is transformed into the sinogram space. The inversion of this process (i.e. the inversion of G) allows us to estimate the parameters directly from the data.

Second, with the ODE model above we are separating the tracer concentrations into the arterial and venous tracer concentrations, $C_{\mathcal{A}}(t)$ and $C_{\mathcal{V}}(t)$. Since the arterial blood originates from the left ventricle and the venous blood from the right ventricle, we can identify $C_{\mathcal{A}}(t)$ with the tracer concentration in the left ventricle and $C_{\mathcal{V}}(t)$ with the tracer concentration in the right ventricle. However, due to the size of the compartments and the heart motion during the PET measurement, there will be tracer concentration spillover in the border regions of the compartments. To account for this effect we introduce the *spillover* term $s(x)$.

Combining all these effects linearly

$$C_{\mathcal{T}}^{\text{real}}(t) = rC_{\mathcal{T}}(t) + sC_{\mathcal{A}}(t), \tag{3.11}$$

we get the model operator G

$$G(F, r, s, C_{\mathcal{A}}) = r(x)F(x) \int_0^t C_{\mathcal{A}}(\tau) \mathrm{e}^{-\frac{F(x)}{\lambda}(t-\tau)} d\tau + s(x)C_{\mathcal{A}}(t). \tag{3.12}$$

As in Section 3.2, we can now reformulate the original problem as a parameter identification problem. We can model blood perfusion with a limited set of parameters $p = (F, r, s)$ and create time series of images by computing $G(p)$. Applying the PET operator K to this sequence results in the sinogram data f. The inversion of this process allows us to identify the parameters p directly from the data f, as schematically represented in Figure 3.5.

To actually solve this problem we can write it as a minimization problem,

$$D_{KL}(Ku, f) + \mathcal{R}(p) \rightarrow \min_{p} \quad \text{subject to} \quad u(x,t) = G(p), \tag{3.13}$$

where D_{KL} is the Kullback-Leibler divergence as defined in (2.4) and $\mathcal{R}$ is the functional containing additional regularization on the parameters p. Again, the advantage is in this case that we use the whole dataset to identify the parameters, but we also see that this procedure gives us the opportunity to add regularization directly to the parameters instead of regularizing the image reconstruction process. This gives us much more control over the solution.

BENNING et al. [2010, 2008] are using a forward-backward operator splitting to solve the constrained minimization problem (3.13), based on the corresponding Lagrange functional,

$$\begin{aligned}\mathcal{L}(u, p; \lambda) = &\int_0^T \int_\Omega (Au - f \log(Au)) dx dt + \mathcal{R}(p) \\ &+ \int_0^T \int_\Omega (G(p) - u)\, \lambda dx dt.\end{aligned} \tag{3.14}$$

with λ being the Lagrangian multiplier, Ω the spatial domain and T the end time of the time interval. The main difficulties with this non-linear problem are the identifiability of the parameters since neither convergence nor uniqueness can be guaranteed.

While we will not further investigate this non-linear model based approach, there is a nice connection between the spectral model approach of Section 3.2 and the non-linear approach in this section. Let us compare both operators,

$$[G(F, r, s)](x,t) = r(x)F(x) \int_0^t C_{\mathcal{A}}(\tau) \mathrm{e}^{-\frac{F(x)}{\lambda}(t-\tau)} d\tau + s(x) C_{\mathcal{A}}(t), \tag{3.15}$$

$$[Ba](x,t) = \qquad a_j(x)\tilde{b}_j(t) \qquad\qquad + a_0(x)\tilde{b}_0(t). \tag{3.16}$$

The motivation behind the spectral approach was to have a large number of basis functions $\tilde{b}_i$ and find ideally only one non-zero coefficient $a_j(x)$. In this case, that single coefficient would represent the perfusion, corrected by the tissue

fraction. By adding the input function itself to the set of basis functions as $\tilde{b}_0(t)$, corrected by the spillover, the linear operator B is equivalent to the non-linear operator G.

While under ideal assumptions, the reconstruction of a single coefficient might be theoretically possible, in practice we can not hope to achieve this. In addition to the mentioned spillover effect, we also have to deal with partial volume effects, which is the underestimation of the activity of a small object do to an insufficient resolution of the imaging method. Given all the problems mentioned in this section, a single pixel or voxel will most likely not consist of a single type of tissue and we expect the activity to be represented by several basis functions, and thus by more than one non-zero coefficient.

4

MATHEMATICAL DATA ANALYSIS

In medical imaging, the measured data can be very complex. A dataset usually consists of many different superimposed two- or three-dimensional time-varying signals, while we are only interested in a few key parameters. Data analysis techniques promise to reduce the complexity of these datasets and help recover the relevant information.

In Chapter 2 we have already mentioned the use of factor images to improve the region of interest analysis. Principal component analysis (PCA) is the most prominent way to do a factor analysis and has been used for at least 40 years [BARBER, 1980; SCHMIDLIN, 1979]. Since the popularization by LEE AND SEUNG [1999, 2001], non-negative matrix factorization (NMF) also became a popular choice as well.

While we follow similar ideas than the approaches above, we do not want to use these data analysis methods for a factor analysis, but rather apply them to the measurements directly and identify the time-activity curves and the corresponding coefficients.

We start this chapter in Section 4.1 by providing a brief introduction to PCA and the inherent problems that arise when used on dynamic PET data. In Section 4.2 we introduce the NMF problem and discuss the corresponding minimization problem and its properties. We close this chapter with Section 4.3 by presenting

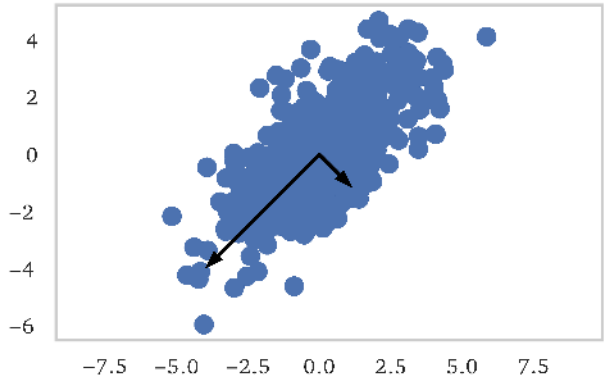


Figure 4.1.: PCA of 2D data cloud.

several algorithms to solve the NMF problem and consider ways to include additional regularization.

4.1. Principal component analysis

Principal component analysis has been invented by PEARSON [1901] and is a statistical data analysis method that transforms a dataset of observations into linearly uncorrelated variables, the *principal components*. The transformation is done in a way that the first principal component has the largest variance, i.e. explains the largest part of the observation data. The second, and all further components, are being chosen to have the largest variance regarding the residual dataset under the constraint of being orthogonal to all preceding components.

In a mathematical sense, PCA is a linear orthogonal transformation of the dataset into a coordinate system where each coordinate corresponds to a principal component, starting by the coordinate with the highest variance going to the coordinate with the least variance. An easy way of visualizing this transformation is to imagine a two-dimensional elliptical point cloud, with the major axis of the ellipsis corresponding to the first principal component, and the minor axis to the second principal component, as shown in Figure 4.1.

Let us assume a p-by-n dataset X, consisting of p variables $x_1, \ldots, x_p$ and n samples or observations. There are no additional assumptions on the dataset X, except that the first and second order statistics are known or can be estimated

from the samples. Thus, the first step is to center the dataset, i.e. shift the sample mean of each column to zero.

Consider now the linear combination

$$y_1 = \sum_{k=1}^{p} w_{1k} x_k = w_1^T X \tag{4.1}$$

where $w_{11}, \ldots, w_{1p}$ are scalar weights of a p dimensional vector w_1. The variable y_1 is the first principal component of X if its variance is maximal. Since the variance increases as the norm of the weight vector grows, we restrict the vector w_1 to be a unit vector. Thus, w_1 has to satisfy

$$w_1 = \arg\max_{\|w\|=1} \|w^T X\|^2 = \arg\max_{\|w\|=1} w^T X^T X w, \tag{4.2}$$

which can be rewritten as

$$w_1 = \arg\max \frac{w^T X^T X w}{w^T w} \tag{4.3}$$

since w_1 is defined as a unit vector.

Having found w_1, we can compute the weights for the second component by simply subtracting the first principal component from X,

$$\hat{X} = X - w_1^T X w_1, \tag{4.4}$$

and use the same formula to find the weights resulting in the maximum variance for the second component

$$w_2 = \arg\max \frac{w^T \hat{X}^T \hat{X} w}{w^T w}. \tag{4.5}$$

If we repeat this process for all principal components, we get the full principal component decomposition of X as

$$Y = WX \tag{4.6}$$

with W being a p-by-p matrix containing the weights and Y the transformed data.

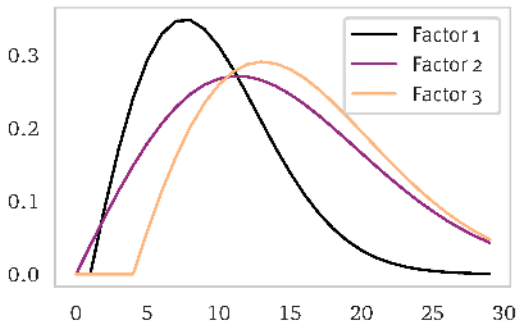
While it is possible to compute all principal components, this is usually not preferable. We want to reduce the complexity of the dataset by identifying the most relevant components first and stop after enough components have been computed. How many components are necessary clearly is a difficult problem in and of itself and depends greatly on the quality of the data and the application. In this sense, we are not looking for a principal component decomposition, but for a principal component approximation.

In fact, when used as a method for compression or dimension reduction, the motivation behind PCA is, that the elements of the dataset are mutually correlated. By computing a transformation consisting of linearly uncorrelated variables, we can reduce the redundancy inherent in the data due to the correlation. Thus, we can compress the data since we need fewer components to describe the transformed dataset without losing information.

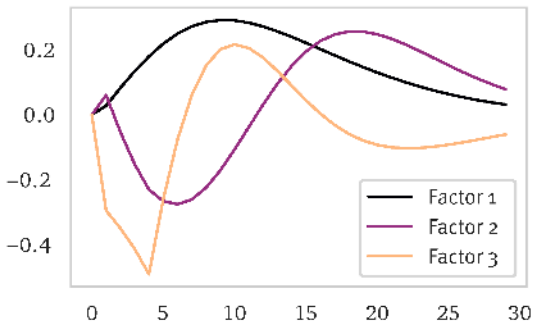
Although PCA is very good at reducing the complexity and dimensionality of a dataset without losing much information, we will not follow this approach going forward. While the components have a statistical interpretation (maximizing variance), they lack a connection to the physiological factors we are looking for when dealing with dynamic PET data. The most obvious problem is that we can not expect the principal components to be non-negative, while the factors we are interested in are based on physical, non-negative quantities.

A simple illustration of the behavior of PCA for non-negative data is shown in Figure 4.2. The dataset consists of three non-negative squares, each having a distinct TAC (see Figures 4.2a and 4.2c). We see in the first PCA factor in Figure 4.2d, that all three squares are represented partly by the first factor. In fact, all three squares and the TAC for the first factor are positive, since PCA tries to explain as much of the data as possible in the first factor. However, for the next factors, PCA is limited to factors orthogonal to the previous factors which results in negative values in both the TACs as well as the factors.

There have been several attempts to improve PCA to mitigate this problem. The idea is, that the space spanned by the principal components might at least contain

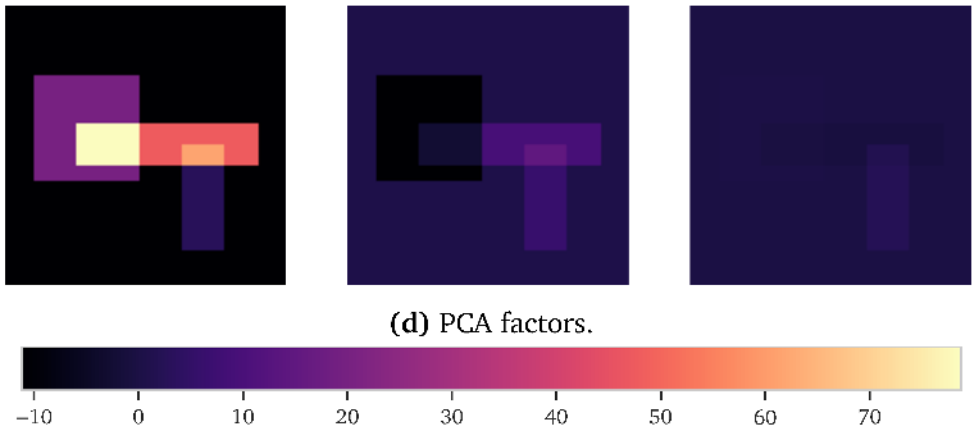


(a) True time activity curves.



(b) PCA time activity curves.

(c) True factors.



(d) PCA factors.

Figure 4.2.: PCA decomposition example. (a) shows the time-activity curves of the factors in (c); (b) shows the time-activity curves of the PCA factors in (d).

the physiological factors that one is interested in and it is simply a matter of finding them in the reduced PCA space. Since there are infinitely many solutions, several constraints have been used to find physiologically meaningful factors, one of the first of course being a positivity constraint. These efforts only showed limited success, as has been noted by BARBER AND MARTEL [1992].

Therefore, we will use another statistical analysis method, called *non-negative matrix factorization* that gets rid of the problem of non-negative factors.

4.2. Non-negative matrix factorization

Non-negative matrix factorization (NMF) is a multivariate analysis method that factorizes a data matrix X into two non-negative matrices W and H. Similar to PCA, we are usually not interested in a full decomposition of the data, but rather in a few components which reasonably approximate the dataset. Since non-negativity is inherent in dynamic PET datasets, NMF leads to a decomposition into factors that are much easier to interpret compared to PCA. In fact, since only additive combinations are allowed due to the non-negativity, NMF goes well with the idea that we want to decompose data into separate parts constituting the whole dataset.

NMF has been introduced for spectrophotometry by LAWTON AND SYLVESTRE [1971] under the name *self modeling curve resolution*, while PAATERO AND TAPPER [1994] called the concept *positive matrix factorization*. The name *non-negative matrix factorization* became prevalent after LEE AND SEUNG [1999, 2001] published simple update rules to solve the NMF problem.

To state the problem mathematically, let X be a non-negative matrix of dimension $M \times N$. We want to find two non-negative matrices $W \in \mathbb{R}^{M \times K}$ and $H \in \mathbb{R}^{K \times N}$ such that

$$X \approx WH. \tag{4.7}$$

The dimension K is typically chosen in a way to reduce the dimensionality, i.e. $K \ll M, N$. Due to the reduction of dimensionality, we can not necessarily expect an exact solution but only an approximation and thus, NMF is also sometimes called *non-negative matrix approximation*.

To calculate a factorization as in (4.7) we consider the following minimization problem

$$W, H \leftarrow \arg\min_{W,H} D(X, WH) \quad \text{such that } W > 0,\ H > 0, \tag{4.8}$$

with $D(X, WH)$ being some kind of distance measure or cost function.

The most common choice for the distance measure is the Euclidean distance

$$D_{EU}(X, WH) = \|X - WH\|_F^2 = \sum_{i,j} \left(X_{ij} - (WH)_{ij}\right)^2, \tag{4.9}$$

with $\|\cdot\|_F$ denoting the Frobenius norm. This assumes implicitly an additive Gaussian noise model, which is a reasonable assumption for many applications. In our case, however, the natural choice is to use the Kullback-Leibler divergence,

$$D_{KL}(X, WH) = \sum_{i,j} \left(X_{ij} \log \frac{X_{ij}}{WH_{ij}} - X_{ij} + (WH)_{ij}\right), \tag{4.10}$$

since we are dealing with data gathered from measurements that follow a Poisson process (see 2.3).

Before discussing various algorithms to compute the NMF, we will briefly mention challenges arising with NMF.

4.2.1. Non-uniqueness

An apparent problem with any factorization (W, H) of X, is its non-uniqueness. It is easy to see, that any matrix S that satisfies $WS \geq 0$ and $S^{-1}H \geq 0$ creates another valid NMF. In the simplest case, S is just a permutation matrix scaling and permuting single factors, which in practice is not an issue. However, there might exist matrices S satisfying the above conditions that are not simple permutation matrices, which in turn could lead to completely different factors and, hence, interpretations.

Under certain conditions, however, DONOHO AND STODDEN [2004] have been able to prove uniqueness. To summarize the results we need to start with a few definitions about *cones*.

Definition 4.1 (Conical hull). *Let S be any set. The* conical hull *is defined as*

$$\text{cone}(S) = \left\{ \sum_{i=1}^{k} \alpha_i x_i \mid x_i \in S,\ 0 \leq \alpha_i \in \mathbb{R},\ i, k \in \mathbb{N} \right\}. \tag{4.11}$$

Definition 4.2 (Polyhedral cone). *A polyhedral cone C in $\mathbb{R}^n$ is the intersection of a finite number of halfspaces, including the origin, i.e. given a matrix A,*

$$C = \left\{x \in \mathbb{R}^n \mid A^T x = 0\right\}. \tag{4.12}$$

Definition 4.3 (Extreme ray). *An* extreme ray *of a convex cone C is a ray $\mathcal{R}_x = \{ax \mid a \geq 0\}$ where $x \in C$ can not be represented as a convex combination of two points $x_0, x_1 \in C \setminus \mathcal{R}_x$.*

Definition 4.4 (Simplicial cone). *A* simplicial cone *is a polyhedral cone with all extreme rays being linearly independent.*

Note that, given a set of vectors, e.g. the columns of a matrix A, the conical hull

$$\text{cone}(A) = \{x = A\lambda \mid \lambda \geq 0\} \tag{4.13}$$

is a polyhedral cone. Thus, if cone(A) is a simplicial cone, there exists a unique λ for every $x \in C$. On the other hand, if A has full rank, the conical hull cone(A) is a simplicial cone.

Given these definitions, we can state the following Lemma.

Lemma 4.1. *Let rank(X) = K. The NMF of a matrix $X \geq 0$ is essentially unique if and only if a unique simplicial cone $\mathcal{W}$ exists, with K extreme rays satisfying*

$$\text{cone}(X) \subseteq \mathcal{W} \subseteq \mathcal{P}_V \tag{4.14}$$

with $\mathcal{P}_V = \{w \mid U^T w = 0, w \geq 0\}$ and U being a matrix whose columns span Null(X^T).

Proof. See Section 4, Theorem 4.1 in TAM [1981]. □

We can understand Lemma 4.1 as a reformulation of the NMF problem. Given a full rank of the non-negative data matrix X, we can get the factor W by finding a simplicial cone $\mathcal{W}$ and taking its extreme rays as the columns of W.

Finally, DONOHO AND STODDEN [2004] found the following conditions guarantee uniqueness.

Theorem 4.1. *Let $X \geq 0$ with $\text{rank}(X) = K$. The non-negative matrix factorization $X = WH$ is unique under the following conditions.*

- ***Generative Model.*** *Let the groups $\mathcal{P}_1, \ldots, \mathcal{P}_A$ be a partition of the set $\{1, \ldots, K\}$ with $|\mathcal{P}_a| = B$ for all $a = 1, \ldots, A$. For all $i = 1, \ldots, M$ and $a = 1, \ldots, A$, there exists a $W_{i,k}$ such that*

$$W_{i,k} \neq 0, \quad k \in \mathcal{P}_a,$$
$$W_{i,l} = 0, \quad \forall l \in \mathcal{P}_a, l \neq k.$$

- ***Separability.*** *For each $k = 1, \ldots, K$ there exists a $j \in \{1, \ldots, J\}$ such that*

$$H_{k,j} \neq 0,$$
$$H_{k,l} = 0, \quad \forall l \neq j.$$

- ***Complete Factorial Sampling.*** *For any $k_a \in \mathcal{P}_a$, there exist an $i \in \{1, \ldots, M\}$ such that $W_{i,k_a} \neq 0$.*

Proof. See Section 6 in DONOHO AND STODDEN [2004]. □

Under the assumption that a particular factorization is given, LAURBERG et al. [2008] proved uniqueness under a different set of conditions, and HUANG et al. [2014] improved on these results while also extending the uniqueness results to the special case of symmetric NMF. In general, however, we can not expect an NMF to be unique since these kind of conditions are very restrictive. Therefore, we will not discuss these results here but refer to the references for details.

We will deal with the non-uniqueness in two ways. First, since the minimization problem 4.8 is scaling invariant, we incorporate additional constraints on the factors W, H, like column or row normalization using the L_2 norm, which allows us to reasonably compare different factorization results. Second, we incorporate a priori information in form of regularization terms into to the minimization

problem. Instead of 4.8, we the have to solve

$$W, H \leftarrow \underset{W,H}{\arg\min}\, D(X, WH) + r_W(W) + r_H(H), \tag{4.15}$$

with r_W and r_H representing the regularization terms for W and H.

4.2.2. NP-hardness

As mentioned above, we are interested in a non-negative matrix factorization that *approximates* a non-negative matrix, which is a generalization of what is often called *exact NMF*. Using the same notation as in 4.7, *exact NMF* can be stated as the problem of finding the factorization $X = WH$, given a non-negative matrix with $\text{rank}(X) = K$. Note that a solution to for exact NMF does not need to exist, but that an optimal algorithm for the approximated version of NMF should solve the exact NMF problem, when the matrix X does have $\text{rank}(X) = K$.

Vavasis [2009] have shown that the exact NMF problem is NP-hard, and we can assume that any generalization of exact NMF is NP-hard as well. In practice, however, this does not seem to be a problem with the algorithms working successfully for most application. We will examine numerical convergence again specifically for the algorithms that we use below.

4.2.3. Factorization rank

Another very apparent problem is the choice of the factorization rank k. As mentioned above, we are usually interested in reducing the dimensionality of the dataset by identifying only the most relevant factors, but it might not be obvious beforehand how many factors are necessary.

Aside from simply testing several values for k, we could use apriori information in form of the help of an expert, who might be able to guess a realistic number of relevant factors. Another more objective approach would be to use the singular values of the data matrix as an indicator. Either way, this continues to be an open problem without a clear solution.

4.3. Algorithms for non-negative matrix factorization

To find a solution for the minimization problem (4.7), we have to minimize over both W and H. Most algorithms for NMF do this alternatingly, i.e. fixing either W or H and optimizing for the other. For example, we start by keeping H fixed and solve the minimization problem

$$\hat{W} = \arg\min_{W \geq 0} D(X, WH) \tag{4.16}$$

and then use $\hat{W}$ and keep it fixed when we solve

$$\hat{H} = \arg\min_{H \geq 0} D(X, \hat{W}H). \tag{4.17}$$

The result is a block coordinate descent scheme as shown in Algorithm 2, which is the general framework of most alternating algorithms. Since the problem is symmetric in that $D(X, WH) = D(X^T, H^T W^T)$, most algorithms use the same update function for both W and H.

Algorithm 2 Block coordinate descent

Input: data matrix $X \in \mathbb{R}^{m \times n} \geq 0$, factorization rank k

Output: factorization matrices $W \in \mathbb{R}_+^{m \times k}, H \in \mathbb{R}_+^{k \times n}$

1: initialize $W^{(0)}, H^{(0)} \geq 0$
2: **for** $i = 1, 2, \ldots$ **do**
3: $\quad W^{(i)} = \text{update}(X, W^{(i-1)}, WH^{(i-1)})$
4: $\quad H^{(i)^T} = \text{update}(X^T, H^{(i)^T}, W^{(i-1)^T})$
5: **end for**

4.3.1. NMF initialization

All NMF algorithms are iteratively updating the factors W and H and therefore depend on some kind of initialization. The simplest and most used way to initialize the factors is to simply create random non-negative matrices. A randomly bad

initialization, however, might lead to convergence to a local minimum, so that a different random initialization needs to be used. Further, to make different NMF algorithms comparable, one should use the same randomly generated matrices for all algorithms.

BOUTSIDIS AND GALLOPOULOS [2008] have proposed an SVD based initialization method, called *non-negative double singular value decomposition* (NNDSVD), that does not depend on randomness. Given the factorization rank k, the idea is to compute the first k singular values of the data matrix and then using the positive parts of the unit-rank matrices corresponding to the singular vectors. Since the basic form simply sets the negative parts equal to zero, this initialization results in sparse positive factors, that already approximate the dataset to a certain degree, improving the convergence. We summarize the approach in the appendix in Section A.1.

Aside from the basic NNDSVD algorithm, they also propose two dense variants that do not set the negative values to zero but initialize them with the arithmetic mean of the dataset, NNDSVDa, or a random value depending on the mean of the dataset, NNDSVDar.

NNDSVD performs very well in most algorithms and we will use it as the standard initialization method, except for the multiplicative updates rules. We will examine why in Subsection 4.3.2.

4.3.2. Multiplicative update rules (MUR)

While multiplicative update rules (MUR) have been used by DAUBE-WITHERSPOON AND MUEHLLEHNER [1986] for non-negative least squares (NNLS) problems, LEE AND SEUNG [2001] described them for NMF in two variants, using the Euclidean distance and the Kullback-Leibler divergence as cost functions, thereby popularizing the use of NMF.

In the case where the cost function is the Euclidean distance D_{EU}, the update

rules are

$$W \leftarrow W * \frac{XH^T}{WHH^T}, \qquad H \leftarrow H * \frac{W^TX}{W^TWH}. \tag{4.18}$$

Using the Kullback-Leibler divergence, we have the update rules

$$W \leftarrow W * \frac{\frac{X}{WH}H^T}{1_{M\times N}H^T}, \qquad H \leftarrow H * \frac{W^T\frac{X}{WH}}{W^T1_{M\times N}}, \tag{4.19}$$

with $\mathbb{1}_{M\times N}$ being a matrix of dimension $(M \times N)$ with every entry equal to one. Note that the division is considered element-wise, while "$*$" denotes element-wise multiplication.

They have shown that the respective cost functions are non-increasing under these updates, and are invariant under these updates if and only if W and H are at a stationary point of the divergence. The proof can be found in LEE AND SEUNG [2001] as well. It is however easy to see through simple substitution, that a perfect reconstruction $X = WH$ is a fixed point under the above update rules.

Due to the nature of multiplicative updates, it is obvious that a zero entry of W or H cannot be changed anymore, and thus, convergence cannot be guaranteed in general. In practice, a simple solution to this problem is to reinitialize all zero entries with small positive constants.

The problem with zero entries is also the reason, why we will not use NNDSVD to initialize the factors. NNDSVD in its basic form will result in a sparse initial factorization since all negative parts will be set to zero. Using these initial factors we see that under the multiplicative update rules the solution gets stuck in a local minimum close to the initialization, even with small positive reinitializations as mentioned above. Therefore, we will usually use NNDSVDa, however, a completely random initialization seems to lead to good results as well.

The multiplicative updates rules for the regularized NMF problem (4.15) with $r_W(W) = \frac{\alpha_W}{2}\|W\|_F^2 + \beta_W\|W\|_1$ and $r_H(H) = \frac{\alpha_H}{2}\|H\|_F^2 + \beta_H\|H\|_1$ have been given by LECHARLIER AND DEMOL [2013]. For the Euclidean distance $D_{EU}(X, WH)$ the

updates are

$$W \leftarrow W * \frac{XH^T}{WHH^T + \alpha_W W + \beta_W \mathbb{1}_{M\times K}}, \tag{4.20}$$

$$H \leftarrow H * \frac{W^T X}{W^T WH + \alpha_H H + \beta_H \mathbb{1}_{K\times N}}. \tag{4.21}$$

Using the Kullback-Leibler divergence $D_{KL}(X, WH)$, we have the updates

$$W \leftarrow \frac{2A}{B + \sqrt{B^2 + 4\alpha_W A}} \quad \text{with} \quad A = W * \frac{X}{WH} H^T, \quad B = \beta_W \mathbb{1}_{M\times K} + \mathbb{1}_{M\times N} H^T, \tag{4.22}$$

$$H \leftarrow \frac{2C}{D + \sqrt{D^2 + 4\alpha_H C}} \quad \text{with} \quad C = H * W^T \frac{X}{WH}, \quad D = \beta_H \mathbb{1}_{K\times N} + W^T \mathbb{1}_{M\times N}, \tag{4.23}$$

with the exponentiation being defined element-wise as $Z^2 = Z * Z$. The regularization parameters $\alpha_{W,H}, \beta_{W,H}$ balance the respective regularizations.

4.3.3. Alternating (non-negative) least squares (ALS/ANLS)

In the case where the Euclidean distance is used, another popular choice for alternatingly solving the subproblems (4.16) and (4.17) is the alternating least squares (ALS) method. The idea is to solve the unconstrained least squares problem and simply project all negative values onto zero. If we consider the subproblem for W, we have the minimization problem

$$W = \max\left(\arg\min_{W\in\mathbb{R}^{m\times p}} D_{EU}(X, WH), 0\right), \tag{4.24}$$

where $\max(\cdot, 0)$ is applied element-wise. While ALS is easy to implement and not expensive to compute, it usually does not converge, since the cost function might oscillate.

A natural extension to ALS is called alternating non-negative least squares (ANLS). In contrast to ALS, ANLS aims to solve the subproblems (4.16) and (4.17)

exactly. For example, we get the non-negative least squares (NNLS) problem

$$W = \arg\min_{W \geq 0} D_{EU}(X, WH) \tag{4.25}$$

as the update for W.

The active set method by LAWSON AND HANSON [1995] was the first widely used algorithm to solve the NNLS problem. The aim is to find the optimal active set of variables corresponding to *active* constraints, i.e. equality constraints, exchanging one variable between the active and passive set each iteration. Since solving the NNLS problem exactly is computationally fairly expensive, many approaches have been introduced to solve them more efficiently. For example, the block principal pivoting (BPP) algorithm by PORTUGAL et al. [1994] tries to improve on the active set method by exchanging multiple variables at once per iteration.

Another fast way of solving NNLS problems, called *fast combinatorial NNLS (FCNNLS)*, has been introduced by VANBENTHEM AND KEENAN [2004] and is based on the assumption that for many applications single problems are being solved with many observation vectors. In such large-scale problems, there is much potential to reduce the number of arithmetic operations by exploiting the structure of the dataset.

Other ideas include approaches to reduce the computation cost by finding reasonable first approximations for the W and H, as well as only solving the NNLS subproblem in a refinement step in cheaper algorithms like ALS.

Given the regularized NMF problem (4.15) with $r_W(W) = \alpha_W \|W\|_F^2$ and $r_H(H) = \alpha_H \|H\|_F^2$, KIM AND PARK [2008] have shown that we can rewrite the NMF problem as

$$W \leftarrow \arg\min_{W \geq 0} \left\| \begin{pmatrix} H^T \\ \sqrt{\alpha_W} I_K \end{pmatrix} W^T - \begin{pmatrix} X^T \\ 0_{K \times M} \end{pmatrix} \right\|_F^2 \tag{4.26}$$

$$H \leftarrow \arg\min_{H \geq 0} \left\| \begin{pmatrix} W \\ \sqrt{\alpha_H} I_K \end{pmatrix} H - \begin{pmatrix} X \\ 0_{K \times M} \end{pmatrix} \right\|_F^2, \tag{4.27}$$

with I_K being a $K \times K$ identity matrix and $0_{K\times M}, 0_{K\times N}$ matrices with only zero entries of dimension $K \times M$ and $K \times N$ respectively.

4.3.4. Alternating direction method of multipliers (ADMM)

As a more general algorithm, SUN AND FEVOTTE [2014]; ZHANG et al. [2015] have proposed using the alternating direction method of multipliers (ADMM) as a flexible framework for NMF. Before formulating the ADMM problem for NMF, we will briefly introduce ADMM in general terms.

Given convex functions f and g, we have a problem in the form

$$\begin{aligned} &\text{minimize} && f(x) + g(z) \\ &\text{subject to} && Ax + Bz = c \end{aligned} \tag{4.28}$$

with $x \in \mathbb{R}^n$, $z \in \mathbb{R}^m$, $A \in \mathbb{R}^{k\times n}$, $B \in \mathbb{R}^{k\times m}$ and $c \in \mathbb{R}^k$. We see that the objective function in the above problem is separated with respect to the two variables x and z. The corresponding augmented Lagrangian is given by

$$L_\rho(x, z, y) = f(x) + g(z) + y^T(Ax + Bz - c) + \frac{\rho}{2}\|Ax + Bz - c\|_2^2 \tag{4.29}$$

and the ADMM algorithm is given by the iterations

$$x^{k+1} := \arg\min_x L_\rho(x, z^k, y^k) \tag{4.30}$$

$$z^{k+1} := \arg\min_z L_\rho(x^{k+1}, z, y^k) \tag{4.31}$$

$$y^{k+1} := y^k + \rho(Ax^{k+1} + Bz^{k+1} - c) \tag{4.32}$$

with some $\rho > 0$.

We can formulate a scaled version of ADMM by scaling the dual variable and combining the terms dealing with the constraints. With $u = \frac{1}{\rho}y$ as the *scaled dual*

variable, ADMM can be written as

$$x^{k+1} := \arg\min_x \left(f(x) + \frac{\rho}{2} \|Ax + Bz^k - c + u^k\|_2^2 \right) \tag{4.33}$$

$$z^{k+1} := \arg\min_z \left(g(z) + \frac{\rho}{2} \|Ax^{k+1} + Bz - c + u^k\|_2^2 \right) \tag{4.34}$$

$$u^{k+1} := u^k + Ax^{k+1} + Bz^{k+1} - c \tag{4.35}$$

Both ADMM variants, the *unscaled* version (4.30) and the *scaled* version (4.33), are equivalent, but we will mainly use the scaled version going forward, because the updates usually are shorter than in the unscaled variant.

We can now rewrite the NMF problem (4.8) as

$$\begin{aligned} \text{minimize} \quad & D(X, V) + r_H(H) + r_W(W) \\ \text{subject to} \quad & V = \tilde{W}\tilde{H}, \\ & W = \tilde{W}, H = \tilde{H}, \end{aligned} \tag{4.36}$$

with the corresponding augmented Lagrangian

$$\begin{aligned} L_\rho(X, W, H, \tilde{W}, \tilde{H}, \alpha_X, \alpha_W, \alpha_H) = \; & D(V|X) + \langle \alpha_X, X - \tilde{W}\tilde{H} \rangle + \frac{\rho}{2} \|X - \tilde{W}\tilde{H}\|_F^2 \\ & + \langle \alpha_W, W - \tilde{W} \rangle + \frac{\rho}{2} \|W - \tilde{W}\|_F^2 + r_W(W) \\ & + \langle \alpha_H, H - \tilde{H} \rangle + \frac{\rho}{2} \|H - \tilde{H}\|_F^2 + r_H(H), \end{aligned} \tag{4.37}$$

where $r_W(W)$ and $r_H(H)$ contain additional constraints on W and H, like non-negativity.

The W and H updates in Algorithm 3 are so-called *proximity operators* of the function $\frac{1}{\rho} r(\cdot)$ around the point $\bar{W} = \tilde{W} + \alpha_W$ and $\bar{H} = \tilde{H} + \alpha_H$ respectively. While there are many different possible ways to impose additional regularization or constraints on the problem, we will concentrate on the two variants we are most interested in, i.e. non-negativity and a smoothness regularization. For a detailed overview of proximal algorithms, see PARIKH et al. [2014].

Non-negativity The non-negativity constraint can be expressed simply as a projection onto $\mathbb{R}_+$ and the function $r(\cdot)$ becomes the indicator function of $\mathbb{R}_+$.

Algorithm 3 ADMM Algorithm for NMF

Input: data matrix $X \in \mathbb{R}^{m \times n} \geq 0$

Output: factorization matrices $W \in \mathbb{R}_+^{m \times p}, H \in \mathbb{R}_+^{p \times n}$

initialize $X, W, H, \tilde{W}, \tilde{H}, \alpha_V, \alpha_W, \alpha_H$
for $i = 1, 2, \ldots$ **do**
  $\tilde{W}^T \leftarrow (\tilde{H}\tilde{H}^T + I)^{-1}(\tilde{H}V^T + W^T + \frac{1}{\rho}(\tilde{H}\alpha_V^T - \alpha_W^T))$
  $\tilde{H} \leftarrow (\tilde{W}^T\tilde{W} + I)^{-1}(\tilde{W}^T V + H + \frac{1}{\rho}(\tilde{W}^T\alpha_V - \alpha_H))$
  $V \leftarrow \arg\min_{V \geq 0} D(X|V) + \frac{\rho}{2}\|V - WH + \alpha_V\|_F^2$
  $W \leftarrow \arg\min_W r_W(W) + \frac{\rho}{2}\|W - \tilde{W} + \alpha_W\|_F^2$
  $H \leftarrow \arg\min_H r_H(H) + \frac{\rho}{2}\|H - \tilde{H} + \alpha_H\|_F^2$
  $\alpha_V \leftarrow \alpha_V + V - \tilde{W}\tilde{H}$
  $\alpha_H \leftarrow \alpha_H + H - \tilde{H}$
  $\alpha_W \leftarrow \alpha_W + W - \tilde{W}$
end for

The update reduces to an element-wise projection of the negative values to zero.

Smoothness regularization Smoothness can be enforced by adding Tikhonov regularization $\|\Gamma \cdot\|_F^2$ with the Tikhonov matrix Γ being a difference operator, i.e. a tridiagonal matrix with 2's on the diagonal and (-1)'s on the sub- and superdiagonal. Using the factor H as an example, the regularization function becomes $r_H(H) = \frac{\lambda}{2}\|\Gamma H\|_F^2$, with some regularization parameter $\lambda > 0$, and the corresponding proximity operator $H = \rho(\lambda\Gamma^T\Gamma + \rho I)^{-1}\bar{H}$.

The choice of the ADMM parameter ρ is still an open problem since it affects both the overall convergence as well as the convergence speed. While there exist different approaches to heuristically determine a good value, in practice one often has to simply test different values and choose the best performing one.

4.3.5. Alternating optimization ADMM (AO-ADMM)

HUANG et al. [2015, 2016] have shown that the combination of alternating

optimization (AO) and alternating direction method of multipliers (ADMM) creates a versatile framework for matrix and tensor factorization. In contrast to ADMM, the proposed AO-ADMM framework does not try to solve the optimization problem (4.36) for both factors simultaneously, but rather splits the problem into two subproblems and alternatingly solves these subproblems using ADMM. Similar to the other alternating algorithms, the subproblems arise by fixing one of the factors and solving the minimization problem for the other factor. In this case, however, additional auxiliary variables $\tilde{W}$, $\tilde{H}$ and $\tilde{X}$ and the dual variables α_W, α_H and α_X are being introduced in the ADMM algorithm. Fixing W, the subproblem for H can be written as

$$\begin{aligned} \min_{H,\tilde{H},\tilde{X}} \quad & D(X,\tilde{X}) + r_H(H) \\ \text{subject to} \quad & H = \tilde{H}, \quad \tilde{X} = W\tilde{H} \end{aligned} \tag{4.38}$$

with $r_H(H)$ representing additional constraints on the factor H. Accordingly, fixing H, we get the subproblem for W,

$$\begin{aligned} \min_{W,\tilde{W},\tilde{X}} \quad & D(X,\tilde{X}) + r_W(W) \\ \text{subject to} \quad & W = \tilde{W}, \quad \tilde{X} = \tilde{W}H \end{aligned} \tag{4.39}$$

with $r_W(W)$ representing additional constraints on the factor W. The AO-ADMM framework is described in Algorithm 4.

Algorithm 4 AO-ADMM framework

Input: data matrix $X \in \mathbb{R}^{m\times n} \geq 0$, factorization rank p

Output: factorization matrices $W \in \mathbb{R}_+^{m\times p}, H \in \mathbb{R}_+^{p\times n}$

1: initialize $W^{(0)}, H^{(0)} \geq 0$
2: initialize dual variables $U_W \leftarrow 0$, $U_H \leftarrow 0$
3: **for** $i = 1, 2, \ldots$ **do**
4: $\quad (W^{(i)}, U_W) \leftarrow$ solve subproblem 4.38 with algorithm 5
5: $\quad (H^{(i)^T}, U_H) \leftarrow$ solve subproblem 4.39 with algorithm 5
6: **end for**

Each subproblem can be solved in an analogous way since the NMF problems $X = WH$ and $X^T = H^T W^T$ are equivalent. Therefore, we will describe the ADMM algorithm only for the subproblem (4.38) in Algorithm 5. Note that, since W is fixed in the subproblem, the product $W^T W$ can be precomputed and we can use the Cholesky decomposition of $LL^T = W^T W + \rho I$ to efficiently compute the $\tilde{H}$ update instead of computing the inverse $(W^T W + \rho I)^{-1}$ directly.

Algorithm 5 ADMM update for NMF subproblem (4.38)

Input: data matrix $X \in \mathbb{R}^{m \times n} \geq 0$, factorization rank k

Output: $H \in \mathbb{R}_+^{p \times n}$, $\alpha_H \in \mathbb{R}_+^{p \times n}$, $\tilde{X} \in \mathbb{R}_+^{n \times m}$, $\alpha_X \in \mathbb{R}_+^{m \times n}$

1: initialize $H, \alpha_H, \tilde{X}, \alpha_X \geq 0$
2: $\rho = \frac{\|W\|_F^2}{k}$
3: **for** $i = 1, 2, \ldots$ **do**
4: $\quad \tilde{H} \leftarrow (W^T W + \rho I)^{-1}(W^T(\tilde{X} + \alpha_X) + \rho(H + \alpha_H)^T)$
5: $\quad H \leftarrow \arg\min_H r(H) \frac{\rho}{2} \|H - \tilde{H} + U\|_F^2$
6: $\quad \tilde{X} \leftarrow \arg\min_{\tilde{X}} D(X, \tilde{X}) + \frac{1}{2}\|\tilde{X} - W\tilde{H} + \alpha_X\|_F^2$
7: $\quad \alpha_H \leftarrow \alpha_H + H - \tilde{H}$
8: $\quad \alpha_X \leftarrow \alpha_X + \tilde{X} - W\tilde{H}$
9: **end for**

The updates for H and $\tilde{X}$ are again proximity operators and for H we can use the same updates as mentioned in Section 4.3.4. The update for $\tilde{X}$, of course, depends on the cost function, that we will consider being either the Euclidean distance or the Kullback-Leibler divergence.

Euclidean distance In the case of $D_{EU}(X, \tilde{X}) = \|X - \tilde{X}\|_F^2$, Algorithm 5 actually simplifies to trivial updates of $\tilde{X}$ and α_X since $\tilde{H}$ is being fitted to X directly instead of $(\tilde{X} + \alpha_X)$. This also means the $\tilde{H}$ update can be computed even more efficiently, since the term $W^T(\tilde{X} + \alpha_X)$ reduces to $W^T X$, which does not change during the iteration and, therefore, can be precomputed as well.

Kullback-Leibler divergence The proximity operator for $D_{KL}(X, \tilde{X})$ as defined in

(4.10) is

$$\tilde{X} = \frac{1}{2}\left((\bar{X} - 1) + \sqrt{(\bar{X} - 1)^2 + 4X}\right), \tag{4.40}$$

where all operations are element-wise [SUN AND FEVOTTE, 2014].

Again, choosing a good value for ρ is a difficult problem itself. Choosing $\rho = \frac{\|W\|_F^2}{k}$ seems to be work well in practice. Especially, since we are alternating the ADMM subproblems, we can use the old iterates of the dual variables as initializations for the next iteration, reducing the amount of ADMM iterations needed to usually less than 10, down to even only one iteration.

5 Numerical Results

In this chapter we will examine the results of the different NMF algorithms introduced in Section 4.3 using synthetic dynamic PET data. Before applying NMF algorithms, we need to reshape the dynamic PET data such that all spatial information for each temporal frame is flattened into a one-dimensional vector. Concatenating these vectors gives us the PET measurements in form of a two-dimensional matrix X. Given a factorization $X \approx WH$, the columns of W consist of the spatial factors, while the rows of H contain the TACs for each factor. In terms of the linear model operator (3.6), W contains the coefficients $a(x)$ and H the discretized temporal basis functions $\tilde{b}(t)$.

In Section 5.1 we will look at the performance of the algorithms when no noise is present to get an idea of what results we can expect in the best case scenario. In Section 5.2 we will add Poisson noise to the data, simulating the bad statistics of a dynamic PET measurement.

5.1. Clean data

We will test the different NMF methods using a synthetic test dataset representing a two-dimensional cross-section through a human body. The whole sequence is 26 frames with Figure 5.1a showing the 13th frame, containing different regions with respective tracer dynamics. We can see, that the dataset basically consists of five

different regions, with five different temporal behaviors. However, the temporal dynamics are very similar, suggesting that it might be difficult to separate the signals from each other.

Since we apply the NMF algorithms to the sinogram data, the factors are also sinograms and therefore difficult to compare visually. We can, however, reconstruct a static image from each sinogram to see what parts of the original image are being represented by which factor, and also how well the different parts have been separated. For the image reconstruction, we use the EM algorithm.

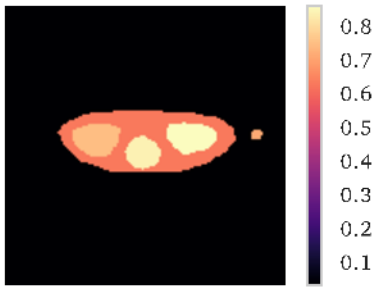
Due to the invariance to scaling of NMF (see Section 4.2.1), we will normalize the time activity curves, i.e. the factor H, using the L_2-norm and scale the factor W accordingly. Also, every factor is shown with respect to the same scale to make it possible to compare the impact of a specific factor on the factorization.

Since the actual values of the factors are meaningless without additional constraints or a problem specific scaling, we omit any numbering of the scales in favor of a more clear representation, focusing on a visual comparison of the results. The color bar in all figures is shown only as an orientation, with the natural interpretation of black representing zero and white representing the maximum value of the dataset.

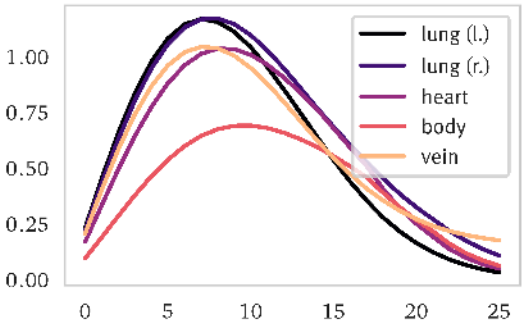
5.1.1. MUR

The multiplicative update rules by LEE AND SEUNG [2001] are the easiest to implement and are very quick to compute. However, they do have problems separating similar signals from each other as we will see below. To get an understanding for the NMF results in this chapter, we will look at the MUR results with increasing the factorization rank one by one.

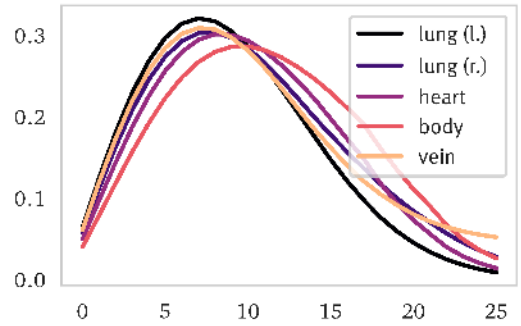
In Figure 5.2 we see the factorization result for $k = 1$, i.e. we only look for one factor, that approximates the dataset the best. We can immediately expect the result to be close to the result for the first principal component of a PCA since for a non-negative dataset, the first component of a PCA will be non-negative as well while explaining most of the variance in the dataset. In both cases, PCA and NMF



(a) 13th frame of synthetic dataset.



(b) Time dynamics of different regions.



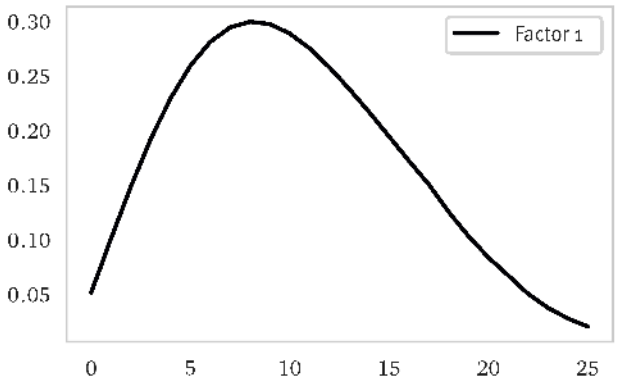
(c) Normalized time dynamics of different regions.

Figure 5.1.: Synthetic dynamic PET dataset. (a) shows the 13th frame of the dataset. (b) and (c) show the true and normalized TACs of the different regions.

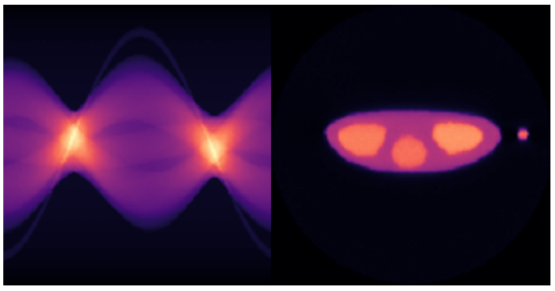
will return a kind of average of the dataset, both in the temporal factors as well as in the spatial factors. The spatial factor 5.2b can be understood to show the most active parts of the dataset over time, on average.

Increasing the factorization rank to $k = 2$, we see in Figure 5.3 that we get a separation. One factor mainly represents the body, while the other factor describes the dynamics of the lungs and the heart. Especially the left lung seems to be well represented by the second factor. Note, however, that the separation is not perfect, which is to be expected since the signals are very close together. It also makes sense in a physiological way, since if there is a lot of blood flow in a certain region, that means that in that region, there is also implicitly some amount of background flow, as in every region.

Increasing the factorization rank to $k = 5$, we can see the five factors in Figure 5.4. While we get different factors with distinct dynamics, each factor contains



(a) Time activity curves.



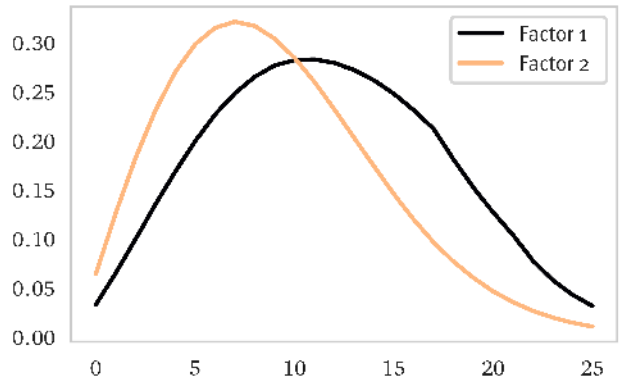
(b) Factor 1.

Figure 5.2.: Factorization results using MUR with factorization rank $k = 1$ and regularization parameters $\alpha_{W,H} = 0$ and $\beta_{W,H} = 0$.

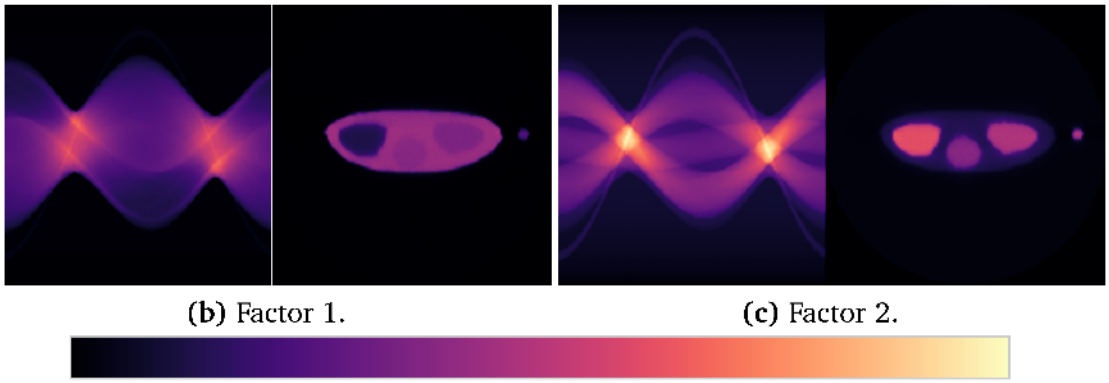
part of the signal for the different regions. For example, the dynamics of the left lung consists of the superposition of the first, third and fifth factor.

Increasing the factorization rank even further to $k = 6$, i.e., even higher than there are signals in the original dataset, we can see in Figure 5.5 that we get redundant factors. For example, apart from a slight scaling difference, the first and fifth factor, i.e. Figures 5.5b and 5.5f, and the third and sixth factor, i.e. Figures 5.5d and 5.5g, look very similar, and the time activity curves of both factor pairs are almost identical as well. This is the usual behavior for all methods when we try to find more factors than the method is able to separate.

Note that for all factorization ranks, even when dealing with clean data, we can improve the results by adding at least a small amount of regularization. This is true for all algorithms below, and thus some regularization has been added to all results, as noted in the captions of the respective figures.



(a) Time activity curves.



(b) Factor 1. (c) Factor 2.

Figure 5.3.: Factorization results using MUR with factorization rank $k = 2$ and regularization parameters $\alpha_W = 1$, $\alpha_H = 5$ and $\beta_{W,H} = 0$.

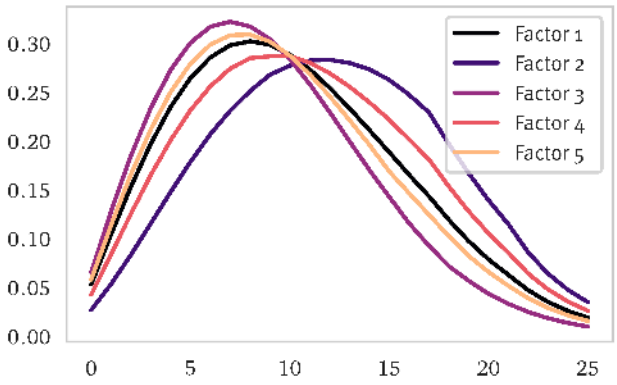
5.1.2. ANLS

Next, we will examine the results of the ANLS algorithm as described in Section 4.3.3. In Figure 5.6 we see the results for a factorization rank of $k = 5$.

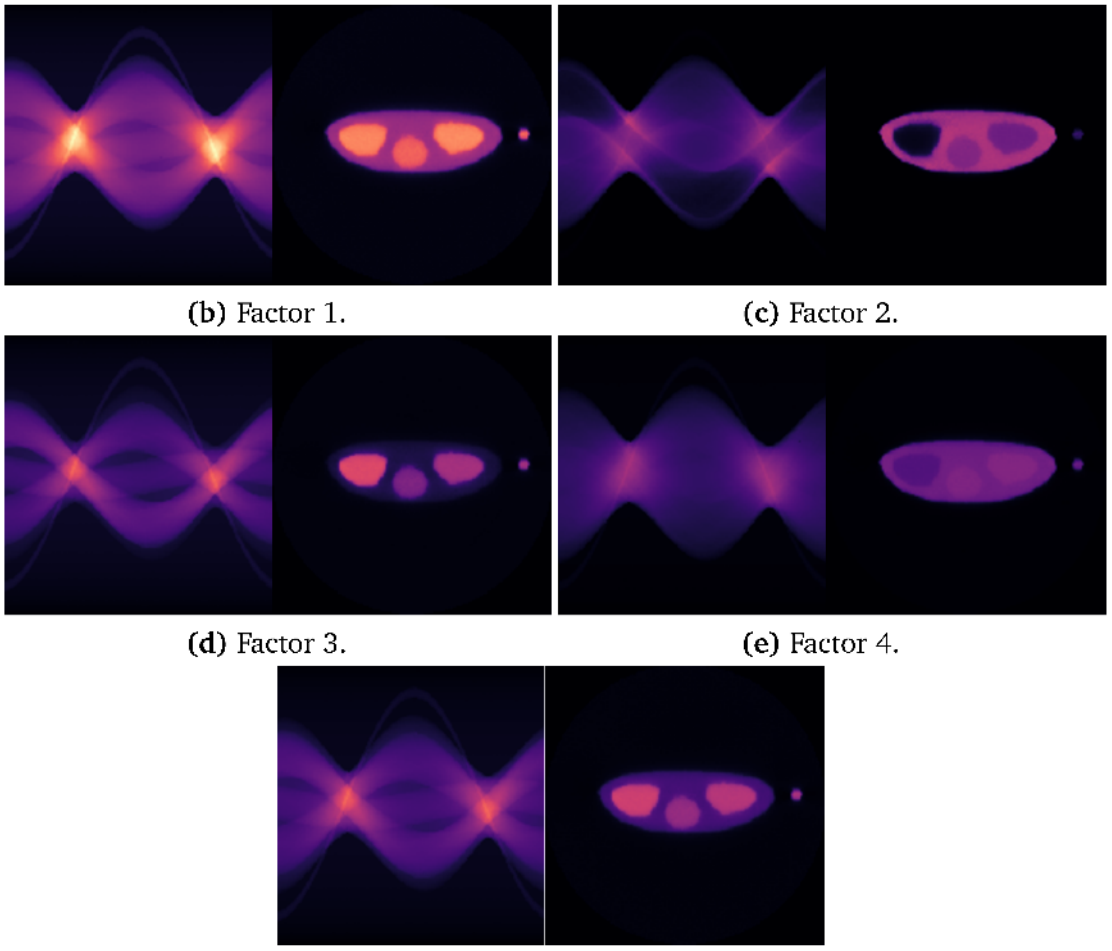
While we still get some overlap in the regions, we also get a fairly good separation. In Figure 5.6f for example, we can see that the fifth factor clearly represents the body without the lungs or the heart. The heart itself can only be found in the second factor, see Figure 5.6c. The lungs are represented by all of the first four factors.

5.1.3. ADMM

The factorization results using the ADMM algorithm from Section 4.3.4 can be found in Figure 5.7. Again, using at least some additional regularization improves



(a) Time activity curves.

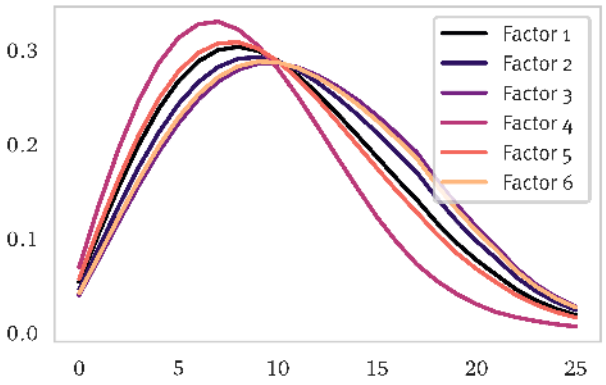


(b) Factor 1. **(c)** Factor 2.

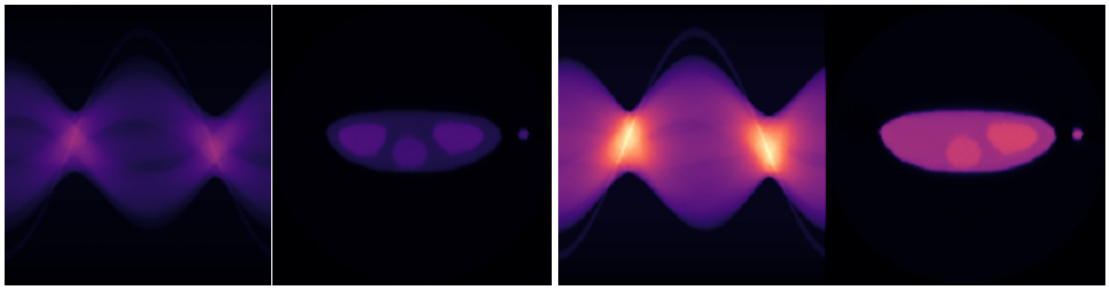
(d) Factor 3. **(e)** Factor 4.

(f) Factor 5.

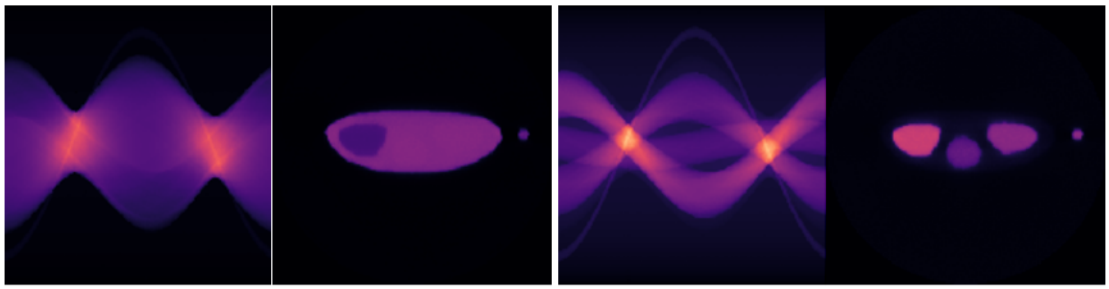
Figure 5.4.: Factorization results using MUR with factorization rank $k = 5$ and regularization parameters $\alpha_W = 5$, $\alpha_H = 4$ and $\beta_{W,H} = 0$.



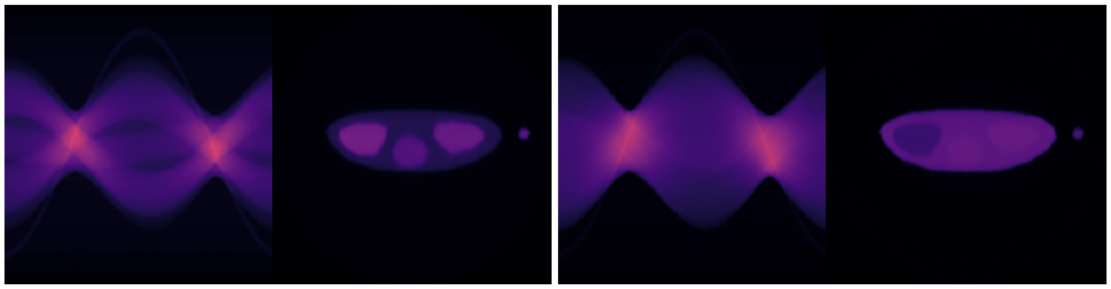
(a) Time activity curves.



(b) Factor 1. **(c)** Factor 2.

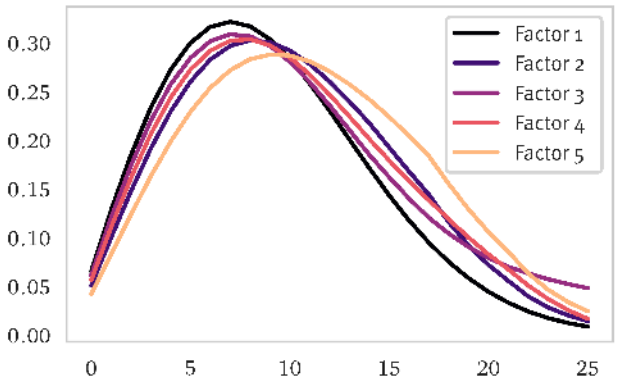


(d) Factor 3. **(e)** Factor 4.

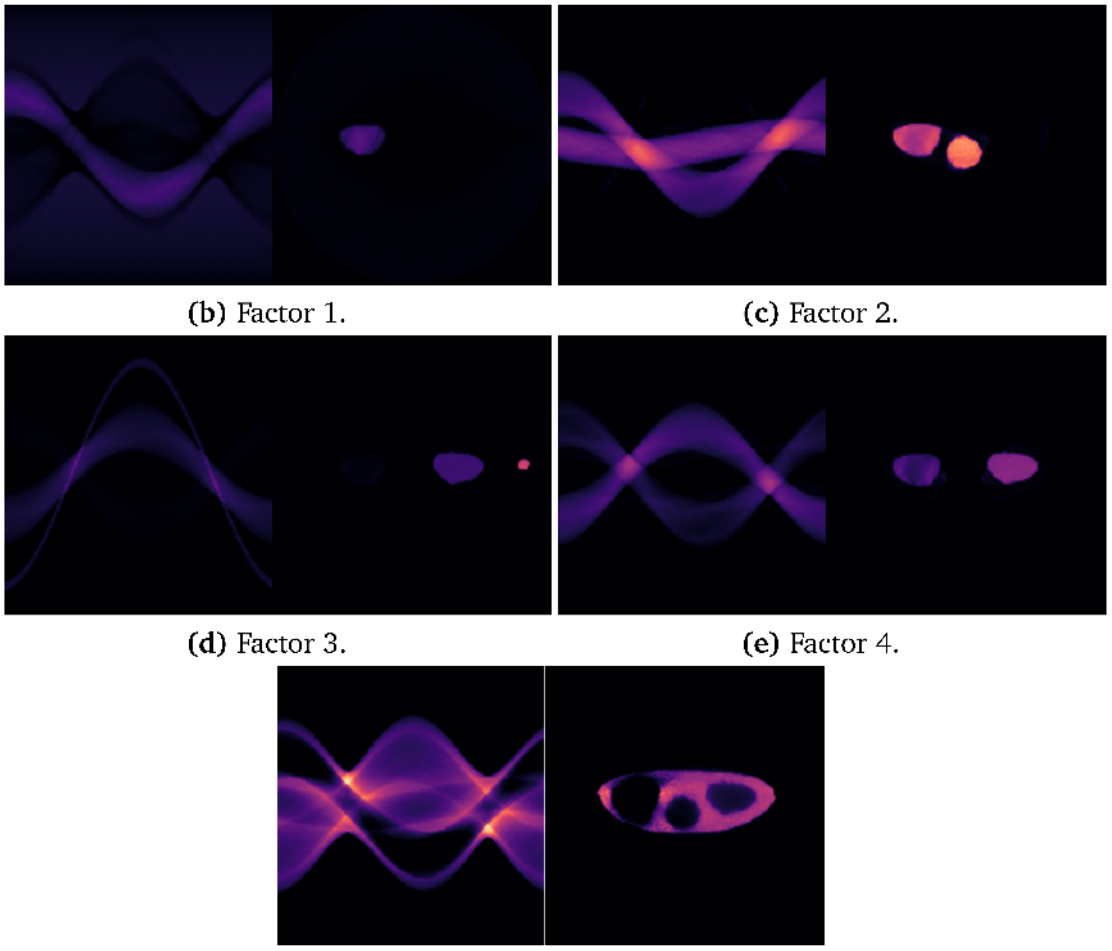


(f) Factor 5. **(g)** Factor 6.

Figure 5.5.: Factorization results using MUR with factorization rank $k = 6$ and regularization parameters $\alpha_W = 5$, $\alpha_H = 5$ and $\beta_{W,H} = 0$.



(a) Time activity curves.



(b) Factor 1. **(c)** Factor 2.

(d) Factor 3. **(e)** Factor 4.

(f) Factor 5.

Figure 5.6.: Factorization results using ANLS with factorization rank $k = 5$ and regularization parameters $\alpha_W = 0$, $\alpha_H = 1$.

the results, therefore we use smoothness regularization for both W and H with $\lambda_W = 0.01$ and $\lambda_H = 1$.

Similar to the results obtained by the ANLS algorithm we do get some separation, e.g. the first factor in Figure 5.7b represents both lungs (plus the injection site). However, part of the dynamic of the right lung is represented by the fifth factor, see Figure5.7f. Also, the body region is represented by two factors, the third and the fourth, see Figure 5.7d and 5.7e respectively.

5.1.4. AO-ADMM

Lastly, we will look at the factorization results for AO-ADMM shown in Figure 5.8. We will use the same additional regularization as for ADMM above, i.e. smoothness regularization on both W and H with $\lambda_W = 0.01$ and $\lambda_H = 1$, and as we can see, we obtain comparable results to the ADMM algorithm.

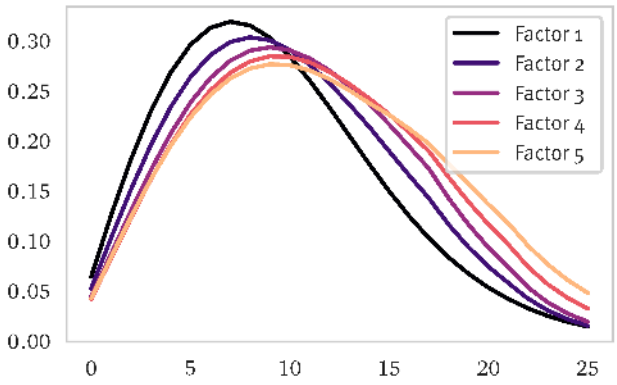
5.1.5. Comparison

We will compare the factorization results by looking at the respective distance values and also at the relative error (RE), which is defined as $RE = \frac{\|X-WH\|_F}{\|X\|_F}$.

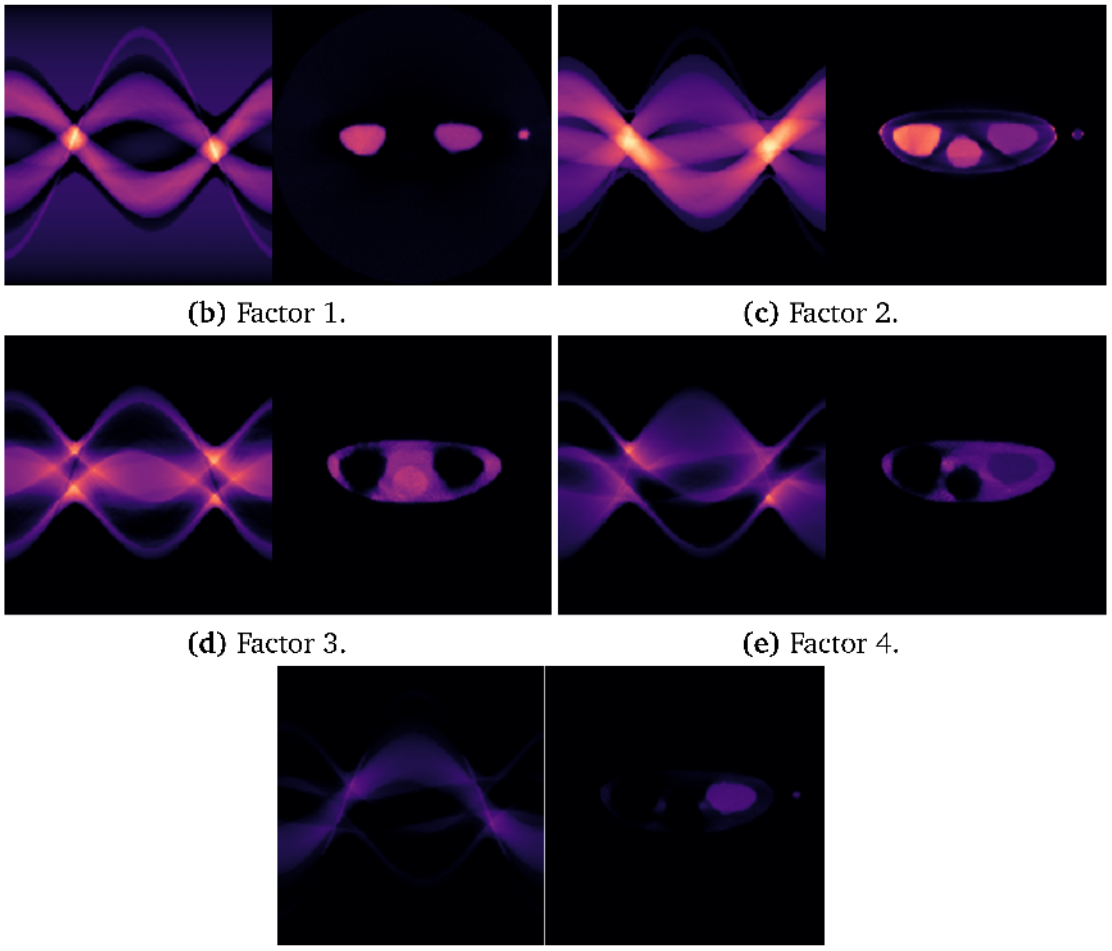
Methods

We start by looking at the approximation errors for the case $k = 5$. Figure 5.9a shows the distance value $D_{EU}(X, WH)$ per iteration for the different algorithms presented in the above sections, while Table 5.1 shows the distance values and the relative error for the end results.

We can see that while the MUR results converge quickly, the distance of the factorization to the dataset is significantly larger compared to the other algorithms. ADMM and AO-ADMM seem to converge to the same solution, which can also be confirmed visually by comparing Figures 5.7 and 5.8, while AO-ADMM needs fewer iterations. Both observations are to be expected. Both algorithms are based on the ADMM algorithm and thus are expected to give comparable results, while AO-ADMM is solving two ADMM problems per iteration, leading to fewer overall



(a) Time activity curves.

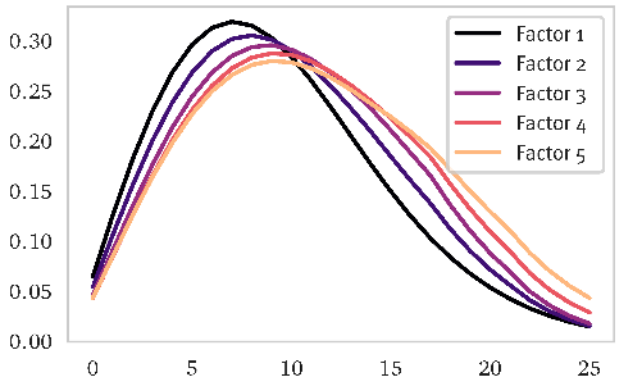


(b) Factor 1. **(c)** Factor 2.

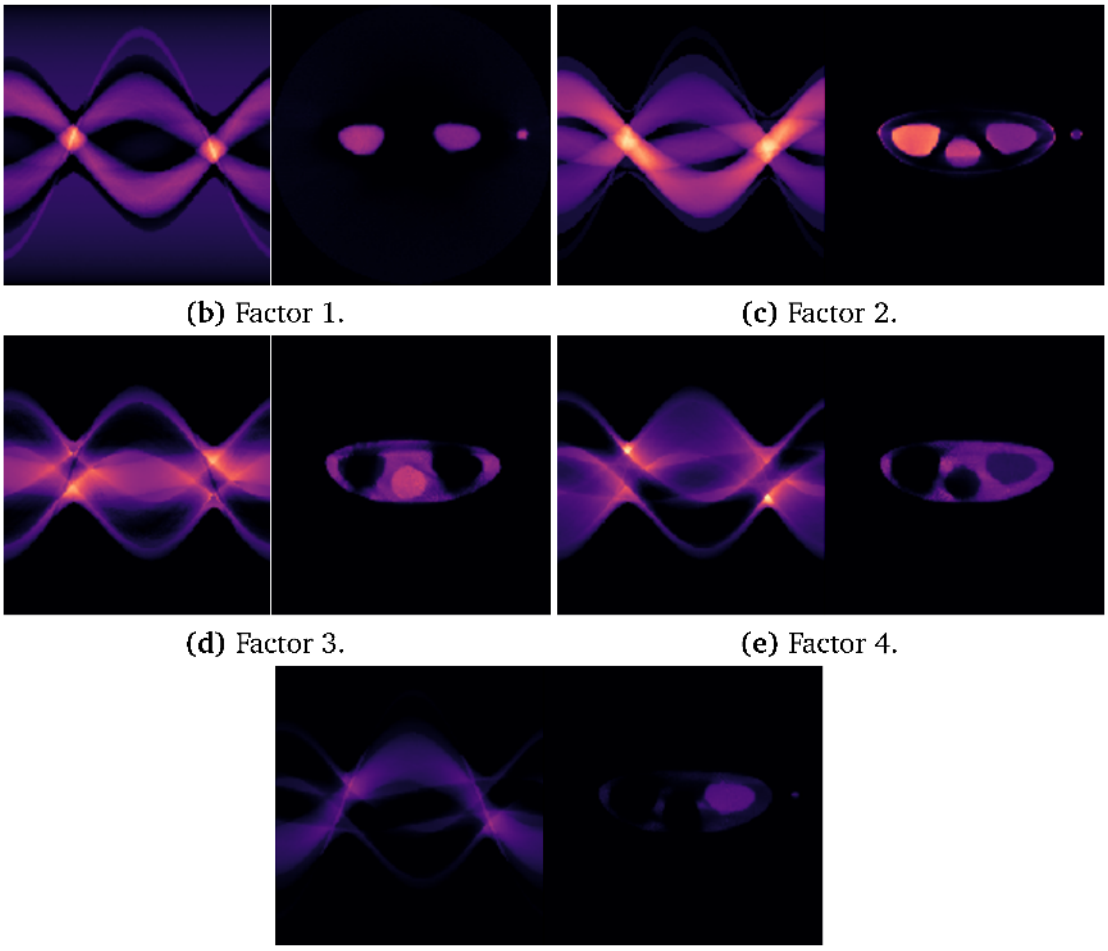
(d) Factor 3. **(e)** Factor 4.

(f) Factor 5.

Figure 5.7.: Factorization results using ADMM with factorization rank $k = 5$ and regularization parameters $\alpha_W = 0.01$, $\alpha_H = 1$.



(a) Time activity curves.



(b) Factor 1. (c) Factor 2.

(d) Factor 3. (e) Factor 4.

(f) Factor 5.

Figure 5.8.: Factorization results using AO-ADMM with factorization rank $k = 5$ and regularization parameters $\alpha_W = 0.01$, $\alpha_H = 1$.

iterations. However, while fewer main iterations are needed, overall it results in a higher computational effort. We tested this by performing 100 iterations of the AO-ADMM algorithms and used the corresponding distance value as the threshold for the ADMM algorithm. On a PC with Intel Core i5-6500 CPU clocked at 3.2 GHz and 16 GB memory, AO-ADMM needed 90.45 seconds, while ADMM finished in only 12.38 seconds. AO-ADMM does seem to converge in a more stable manner, however.

The best factorization, according to the distance value, is obtained by ANLS.

Number of factors

Let us now look at the impact of the factorization rank on the factorizations. Since ANLS gives the best factorization for the exact dataset, we will examine the results for different numbers of factors, with $k = 1, \ldots, 6$ and keeping the regularization parameters equal. In Figure 5.9b we can see that the distance decreases by increasing the number of factors up to $k = 5$. The results become unstable when increasing the factors to $k = 6$, indicating that $k = 5$ is indeed the optimal number of factors.

Using AO-ADMM, however, the situation changes. Figure 5.9c shows that we can increase the factorization rank to $k = 9$ without any stability problems while decreasing the distance. Even though, the original dataset only consists of 5 factors, looking at the error it is advantageous to find more factors since we have more degrees of freedom to represent the dataset. This usually means that a single signal is split between two (or more) factors.

Initialization

We have already mentioned the importance of the initialization, especially when using multiplicative updates. In Figure 5.9d we can see that it is equally important when using ANLS. While the algorithm converges in stable manner when initialized using NNDSVDa, i.e. the mean NNDSVD variant, it shows highly unstable convergence for the other two variants.

When using ADMM, it is the other way around and NNDSVDa actually produces

Algorithm	D_{EU}	RE
MUR	23.1005	0.0441
ANLS	0.0808	0.0026
ADMM	0.3563	0.0055
AO-ADMM	0.3860	0.0057

Table 5.1.: Distance values and relative error of the factorization results for the different NMF algorithms.

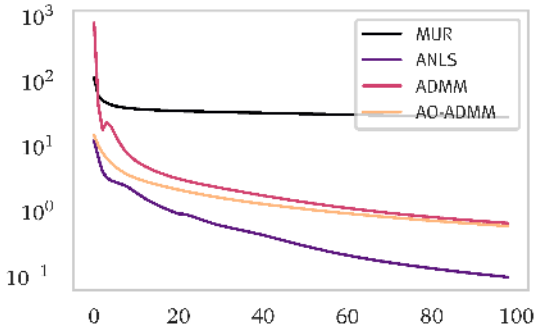
small instabilities during the first few iterations while the basic NNDSVD initialization leads to good results. AO-ADMM shows no dependence on the NNDSVD variant at all and results in the same factorization for every variant. A comparison of AO-ADMM and ADMM for different initializations can be found in Figure 5.9e.

In general, it is always possible to encounter instabilities in these alternating update schemes, regardless of the initialization. For this reason, we always perform a minimal number of iterations before checking for any convergence criteria.

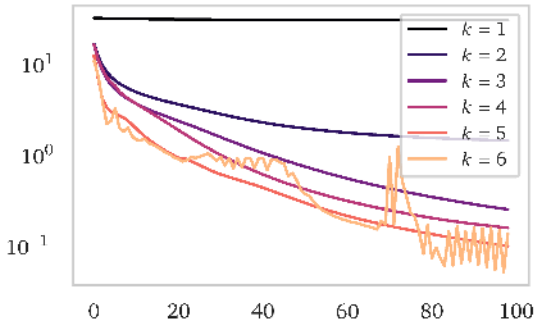
Reconstruction

Finally, in Figure 5.10 we compare the reconstructions from the factorizations with the original dataset. To do this, we generate the approximated sinogram dataset using the factorization, $\tilde{X} = WH$. We then take the 13th frame of $\tilde{X}$ and statically reconstruct it using the EM algorithm.

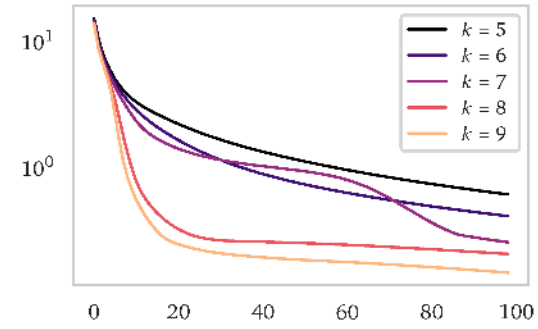
We can see that we get comparable results for both methods, and both reconstructions are very close to the original dataset.



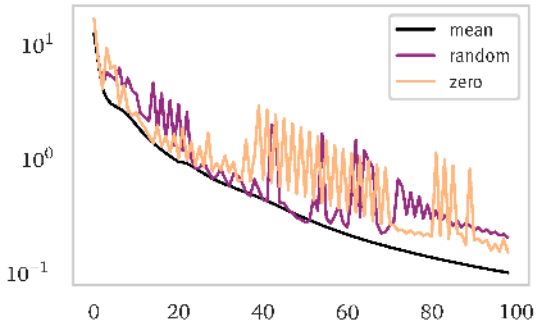
(a) Convergence plot for the different NMF algorithms presented in this section for $k = 5$.



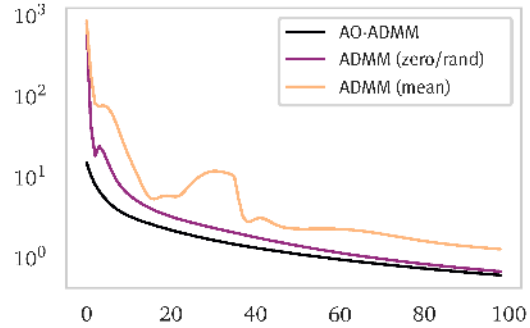
(b) Convergence plot for different number of factors k using ANLS.



(c) Convergence plot for different number of factors k using AO-ADMM.

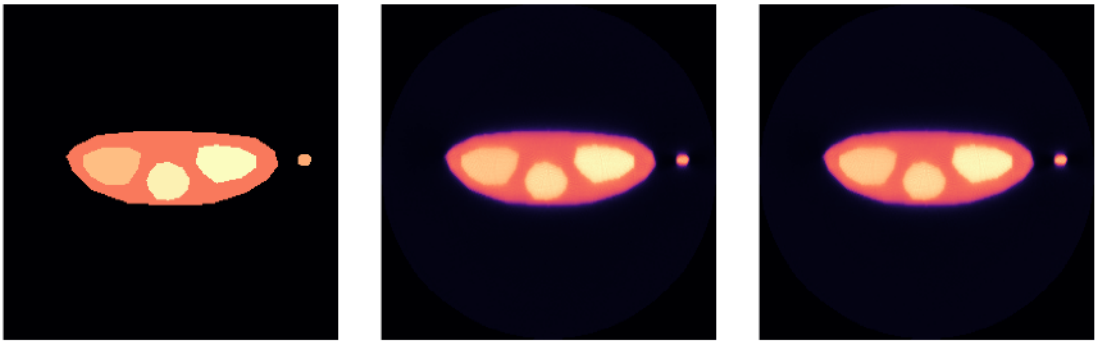


(d) Convergence plot for different initializations for ANLS.



(e) Convergence plot for different initializations for ADMM compared to AO-ADMM.

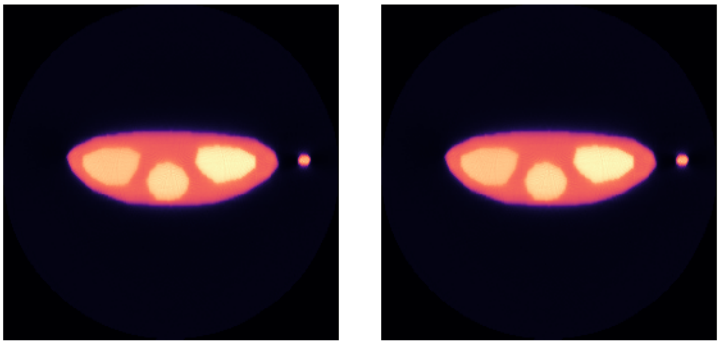
Figure 5.9.: Convergence plots for different factorization results of the exact dataset. The distance $D_{EU}(X, WH)$ is plotted logarithmically on the y-axis against the number of iterations on the x-axis.



(a) Frame 13 of the original dataset.

(b) Frame 13, reconstructed from the factorization using MUR.

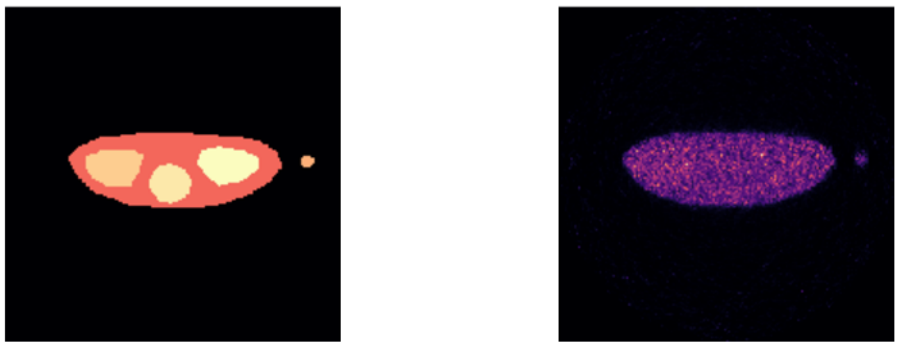
(c) Frame 13, reconstructed from the factorization using ANLS.



(d) Frame 13, reconstructed from the factorization using ADMM.

(e) Frame 13, reconstructed from the factorization using AO-ADMM.

Figure 5.10.: The 13th frame of the original data set and of the reconstructions from the factorizations obtained by the different methods and a factorization rank $k = 5$.



(a) Clean data. **(b)** Noisy data.

Figure 5.11.: 13th frame of the clean and noisy synthetic dataset.

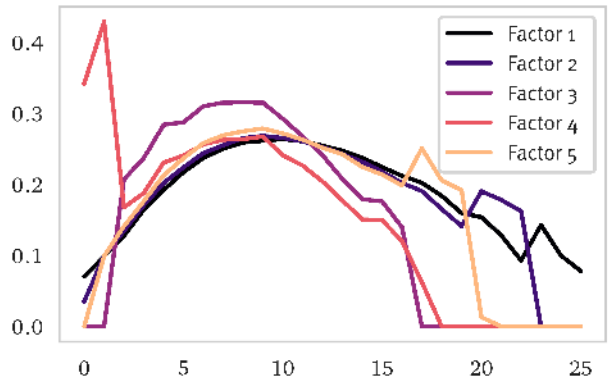
5.2. Noisy data

For the noisy data, we take the same sequence as in Section 5.1 and use a Monte-Carlo simulation to simulate PET data averaging 10000 photon counts per frame. In Figure 5.11 we see the 13th frame of the clean and the simulated noisy data side by side. As can be seen in Figure 5.11b, the image is very noisy due to the comparatively low photon count, simulating the low photon count of a dynamic radioactive water PET scan.

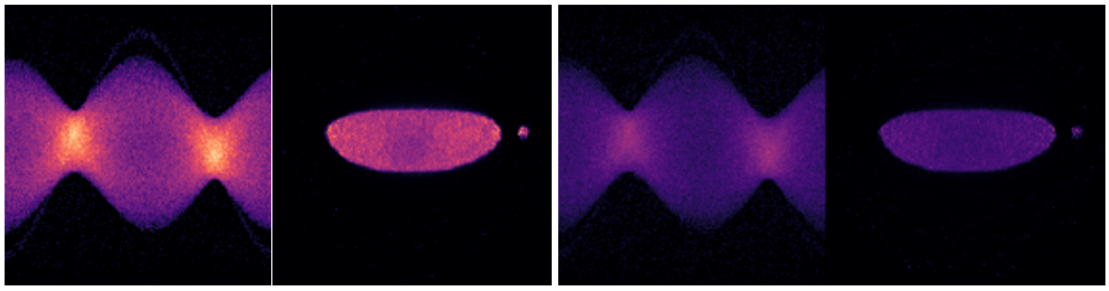
5.2.1. MUR

We again start by looking at the factorizations computed by the MUR algorithm. In the case of clean data, we were able to extract 5 reasonable, albeit not perfectly separated factors. Due to the high noise level in the noisy dataset, we can not expect to be able to find as many factors, and in fact, Figure 5.12 shows the best factorization results we were able to obtain using strong smoothness regularization especially on the time-activity curves. We see an unrealistic TAC for the fourth factor in Figure 5.12a, while the fourth factor itself (Figure 5.12e) is scaled to be basically zero and thereby does only have a negligible influence on the factorization.

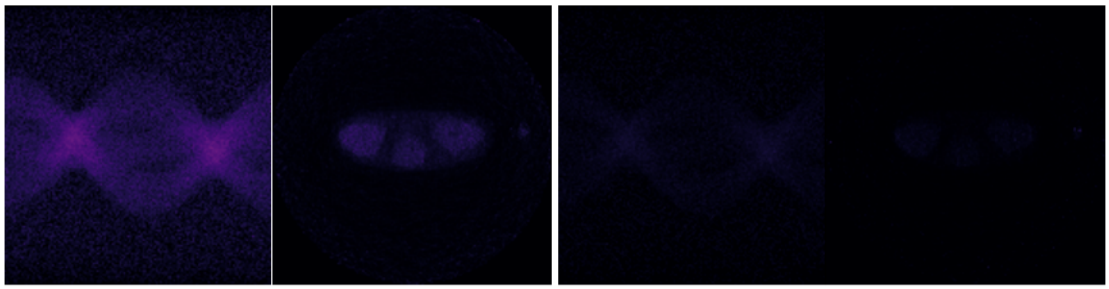
This is another typical factorization behavior, if we try to find more factors than the algorithm is able to separate. One (or more) factor(s) will be scaled down



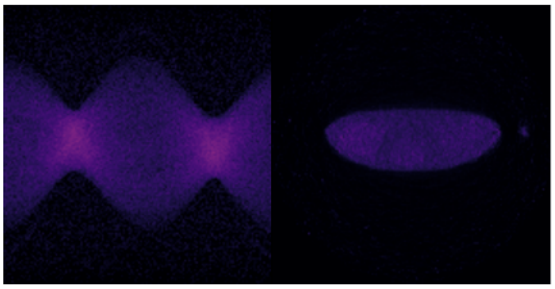
(a) Time activity curves.



(b) Factor 1. **(c)** Factor 2.



(d) Factor 3. **(e)** Factor 4.



(f) Factor 5.

Figure 5.12.: Factorization results using MUR with factorization rank $k = 5$, the cost function $D_{KL}(X, WH)$ and regularization parameters $\alpha_W = 50$, $\alpha_H = 1000$ and $\beta_{W,H} = 0$.

compared to the other factors and the corresponding TAC(s) will show a peak in one point in time. Additional smoothness regularization does not help because it is cheaper in the optimization to simply scale the corresponding coefficient than try to find a smooth TAC and satisfy the regularization.

In fact, we have to reduce the factorization rank to $k = 3$ to get reasonable factorizations for the noisy dataset. Figures 5.13 and 5.14 show the factorization results using the Euclidean distance D_{EU} and the Kullback-Leibler divergence D_{KL}, respectively, with the same regularization results.

While for the clean data, the different loss functions lead to similar factorizations, we can see now that in the noisy case, the factorization using the Kullback-Leibler divergence is clearly better. In fact, using the Euclidean distance with even stronger regularization, we were not able to get a factorization with different factors and without spikes in the TACs.

5.2.2. ANLS

The factorization results using ANLS and the factorization rank $k = 3$ can be seen in Figure 5.15 and we notice the same problems as mentioned above when using MUR with D_{EU}, i.e. we are not able to get three different factors without spikes in the TACs. This is to be expected since ANLS is based on the Euclidean distance.

5.2.3. ADMM

Next, we look at the factorization using ADMM in Figure 5.16. While it does perform better than ANLS, which makes sense since we are able to use the Kullback-Leibler divergence as a cost function due to the flexibility of the algorithm, it was difficult to find good parameters even using a factorization rank $k = 3$. We do get fairly nice factors, with the first factor mainly representing the lungs and the heart, while the third factor mainly represents the body, and the second factor containing an average over the whole region of interest. Unfortunately, we were not able to find smoother time activity curves, since increasing the smoothness regularization results in one of the factors vanishing.

Using a factorization rank $k = 4$ we were not able to find reasonable factors at all.

5.2.4. AO-ADMM

Lastly, we will examine the factorization results using AO-ADMM. Interestingly, AO-ADMM performs much better than ADMM. Comparing the factorization results for the factorization ranks $k = 3$ and $k = 4$, shown in Figures 5.17 and 5.18, respectively, we can see, that we actually benefit from using one more factor. The fourth factor, shown in Figure 5.18e, almost only shows the lungs and the hearts, with only small values for the body. However, the factor itself is scaled down compared to the other factors and examining all factors, it is apparent that we can not hope to find clean regional separation for a dataset of this noise level. Only looking at Figure 5.18a, though, we do get four distinct time activity curves.

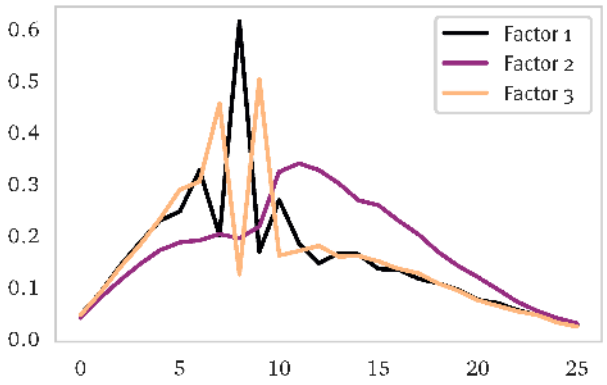
We can in fact increase the factorization rank to $k = 5$ and still get distinct TACs (Fig. 5.19a) and a factor only containing the lungs and the heart (Fig. 5.19f). But if we compare the first and the second factor, and the third and the fourth factor, they do look similar apart from scaling. This might be an indication that we are trying to find too many factors.

5.2.5. Comparisons

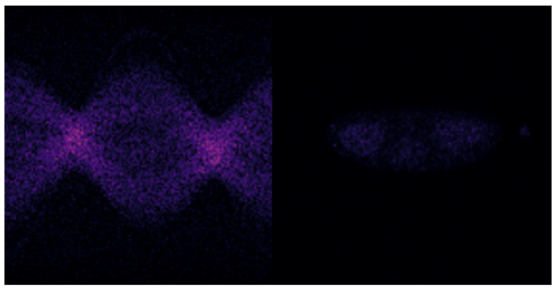
We will again compare the results by looking at the distance values of the factorizations.

Methods

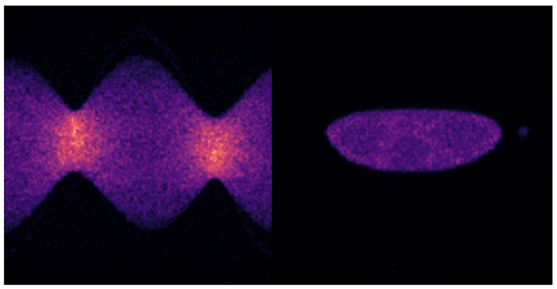
Figure 5.21a shows the Kullback-Leibler divergence plotted against the iteration number for the first 100 iterations. The plot confirms what we have already noticed visually, i.e. AO-ADMM gives the best results, followed by ADMM and ANLS, with MUR resulting in the largest KL divergence value. This is reflected also in the relative error in Table 5.2.



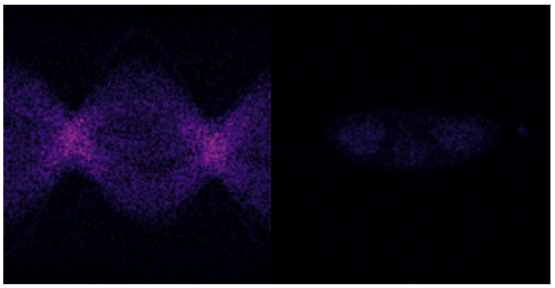
(a) Time activity curves.



(b) Factor 1.



(c) Factor 2.



(d) Factor 3.

Figure 5.13.: Factorization results using MUR with D_{EU} as loss function, factorization rank $k = 3$ and regularization parameters $\alpha_W = 50$, $\alpha_H = 1000$ and $\beta_{W,H} = 0$.

Algorithm	D_{KL}	RE
MUR	$8.8332 \cdot 10^5$	0.4297
ANLS	$8.0639 \cdot 10^5$	0.3555
ADMM	$7.9930 \cdot 10^5$	0.3499
AO-ADMM	$7.5364 \cdot 10^5$	0.3490

Table 5.2.: Distance values and relative error of the factorization results for the different NMF algorithms.

Unfortunately, neither the divergence nor the relative error is a good indicator of the quality of the factorizations. ANLS, ADMM, and AO-ADMM all result in comparable relative errors but using ANLS we were not able to find a good factorization, as mentioned in Subsection 5.2.2. Even with strong regularization, we were not able to find smooth time activity curves, while most of the signal was represented by only one factor. ADMM and AO-ADMM are more flexible allowing us to use different distances as cost functions, with AO-ADMM in general producing the best results.

Cost function

We already have seen in above that we get much better factorization results using ADMM and AO-ADMM because we are able to use the Kullback-Leibler divergence as a cost function which is much more suitable to deal with such a high Poisson noise. To emphasize the influence of the cost function, we present in Figure 5.20 the results using the Euclidean distance with AO-ADMM and a factorization rank $k = 3$. While the factors themselves do look reasonable, the same is not true for the TACs. Even with strong smoothness regularization, we are not able to find TACs that even somewhat resemble the true TACs of the dataset.

Number of factors

Using AO-ADMM we examine the results for a different number of factors k. Figure 5.21b shows that we can improve on the factorization results by including more than 3 factors. We do also see that using 6 factors seems to result in the smallest error, when for the exact dataset in Section 5.1.5 using more than 5 factors resulted in instability in the ANLS algorithm. Examining the reconstructed factors in Figure 5.22, we see that the sixth factor is scaled almost to zero. The sixth factor is basically an additional degree of freedom to account for some part of the noise without representing any part of the signal, thus reducing the error.

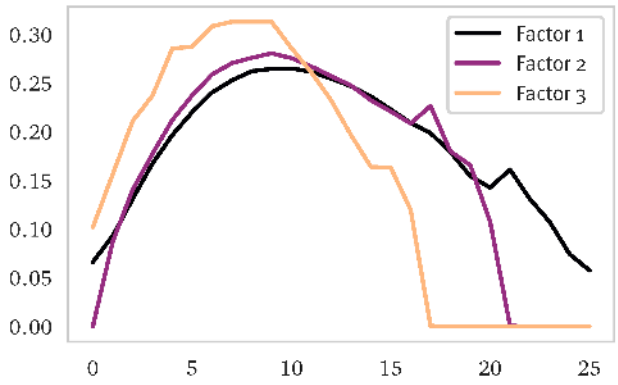
Initialization

In Figure 5.21c we see the effect of the different initializations used for the AO-ADMM algorithm and $k = 3$, with Figure 5.21d shows the same plot, zoomed in to better see the instabilities. Not only do wee see instabilities for every NNDSVD variant, for this case the NNDSVDa variant actually produces the best results, but also needed more iterations to overcome the instabilities.

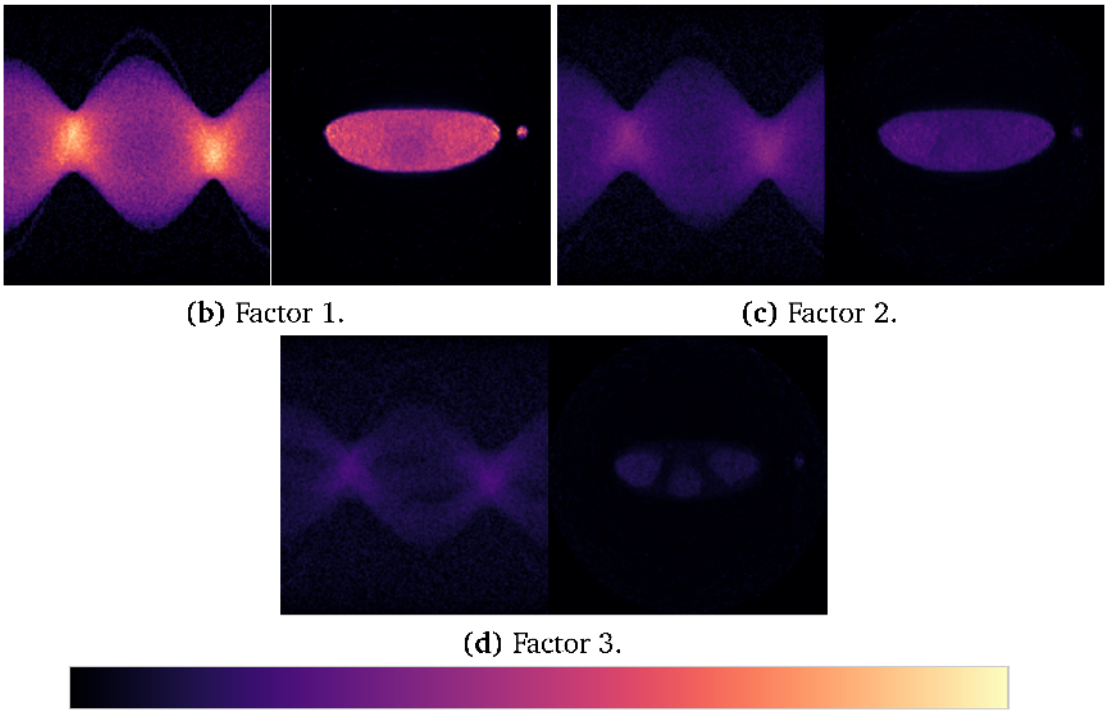
Reconstructions

Finally, we compare the effect of NMF on the standard frame-by-frame EM reconstructions. Figure 5.23 shows the 13th frame of the respective image sequences and we can see the noise reduction effect of factorization methods, even for the MUR results. Again, ANLS, ADMM, and AO-ADMM all show similar results in line with the distance and error of the factorization.

We can also compare the reconstructed time-activity curves of the regions in reconstructed image sequence. In Figure 5.24 we compare the TAC of four regions for all four NMF algorithms. To reduce the chance of unfairly choosing a single noisy pixel, we took the average activity of a 5-by-5 pixel patch from each region.



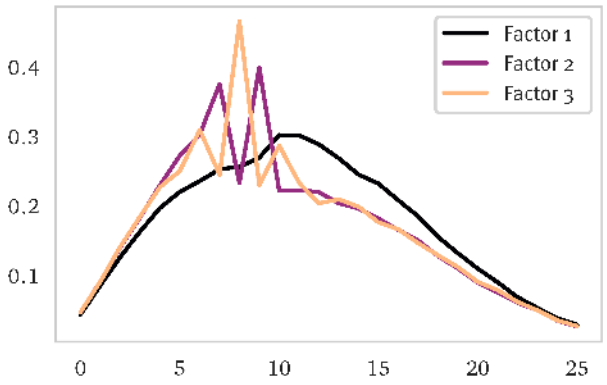
(a) Time activity curves.



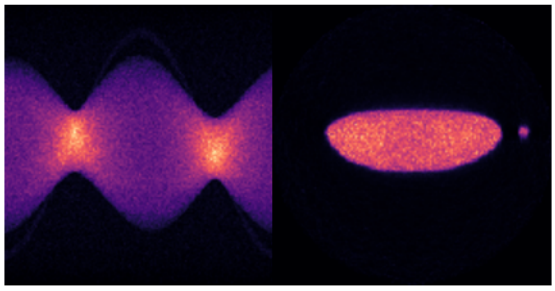
(b) Factor 1. (c) Factor 2.

(d) Factor 3.

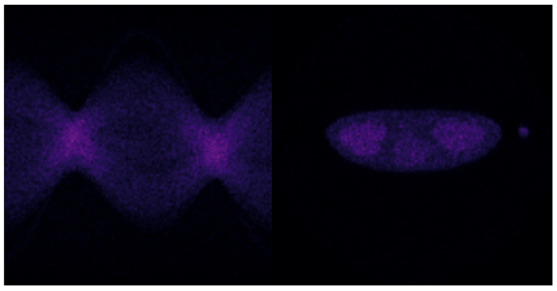
Figure 5.14.: Factorization results using MUR with D_{KL} as loss function, factorization rank $k = 3$ and regularization parameters $\alpha_W = 50$, $\alpha_H = 1000$ and $\beta_{W,H} = 0$.



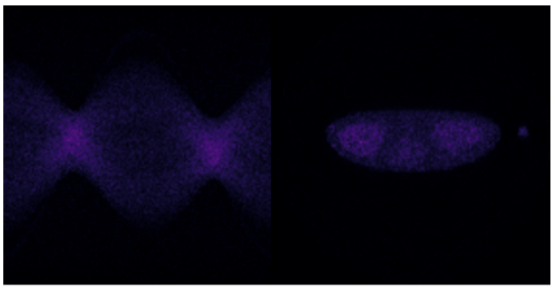
(a) Time activity curves.



(b) Factor 1.

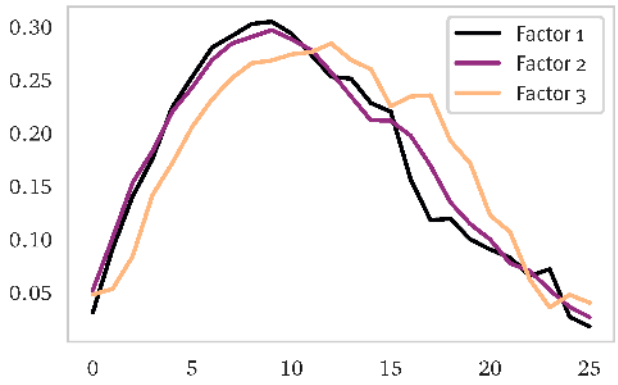


(c) Factor 2.

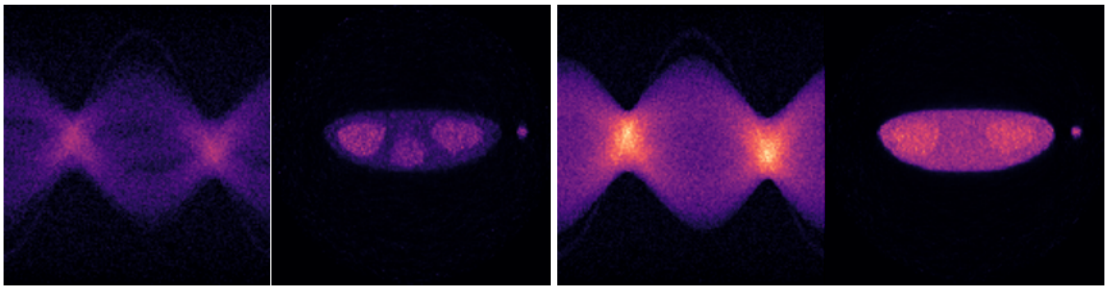


(d) Factor 3.

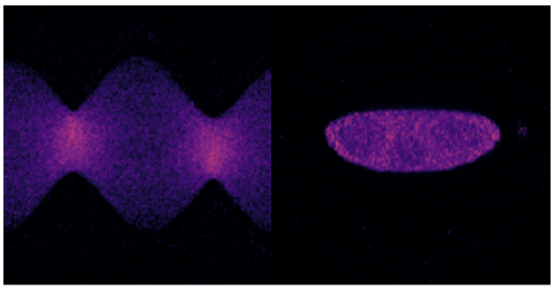
Figure 5.15.: Factorization results using ANLS with factorization rank $k = 3$ and regularization parameters $\alpha_W = 60$ and $\alpha_H = 1000$.



(a) Time activity curves.

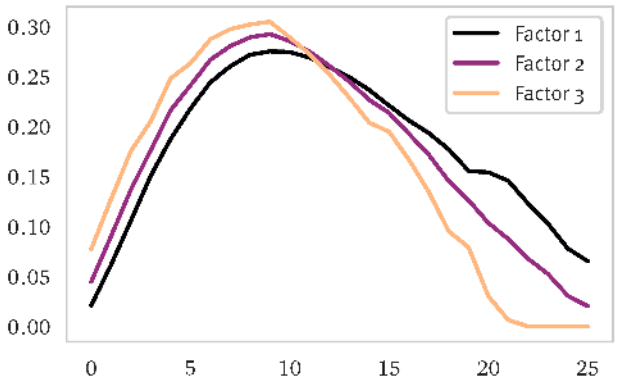


(b) Factor 1. **(c)** Factor 2.

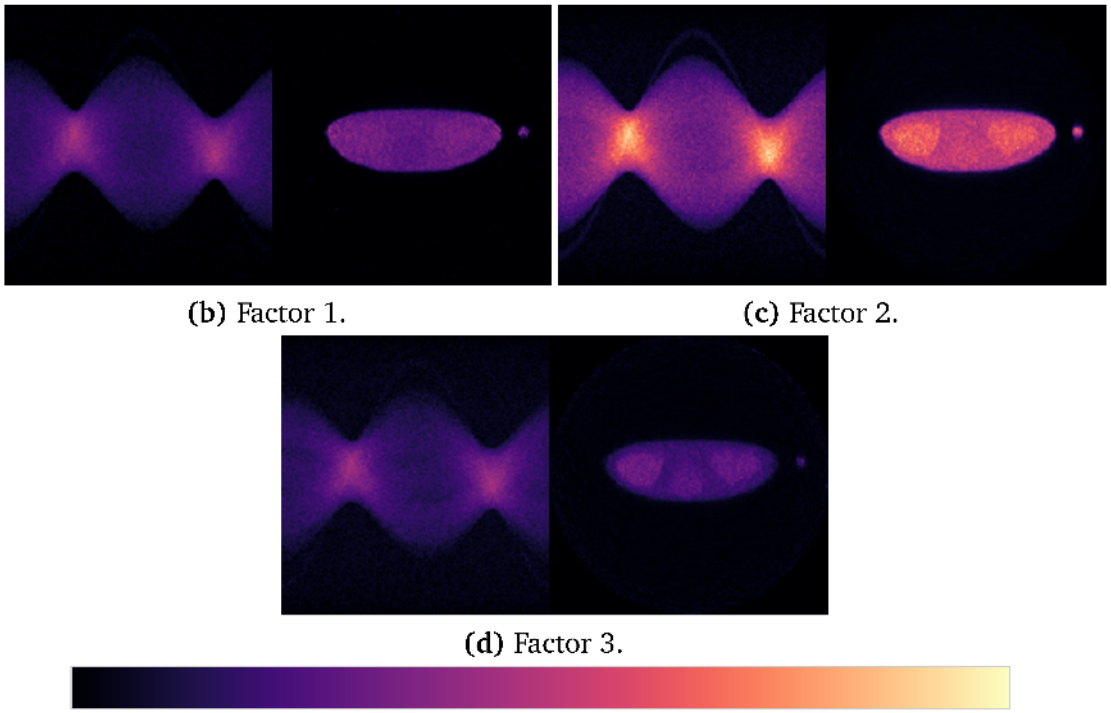


(d) Factor 3.

Figure 5.16.: Factorization results using ADMM with factorization rank $k = 3$ and regularization parameters $\alpha_W = 5$ and $\alpha_H = 30$.



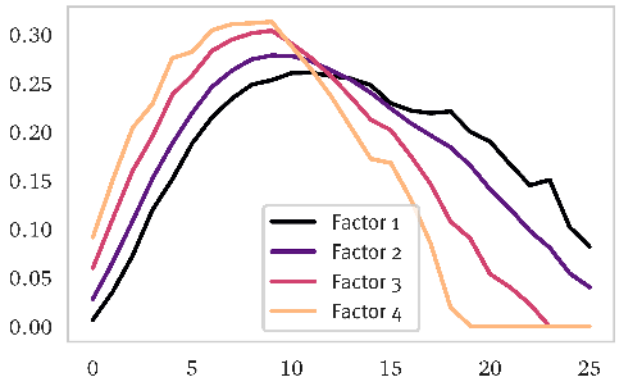
(a) Time activity curves.



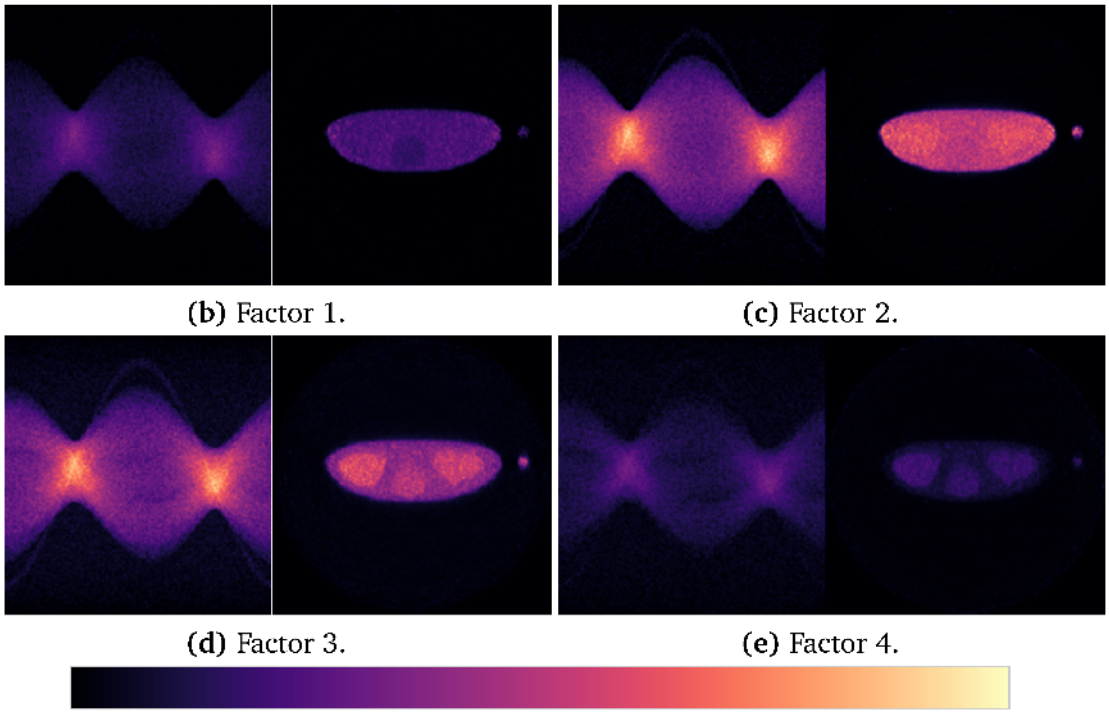
(b) Factor 1. **(c)** Factor 2.

(d) Factor 3.

Figure 5.17.: Factorization results using AO-ADMM with factorization rank $k = 3$ and regularization parameters $\alpha_W = 10$ and $\alpha_H = 70$.



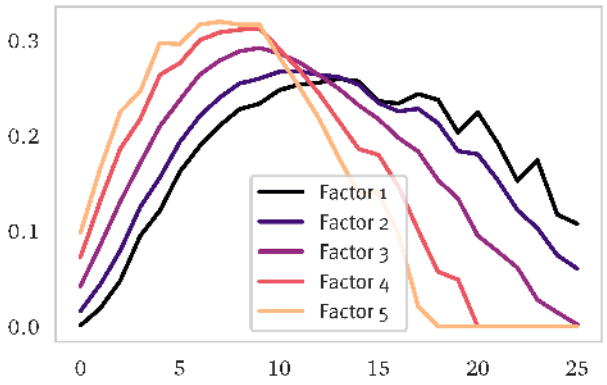
(a) Time activity curves.



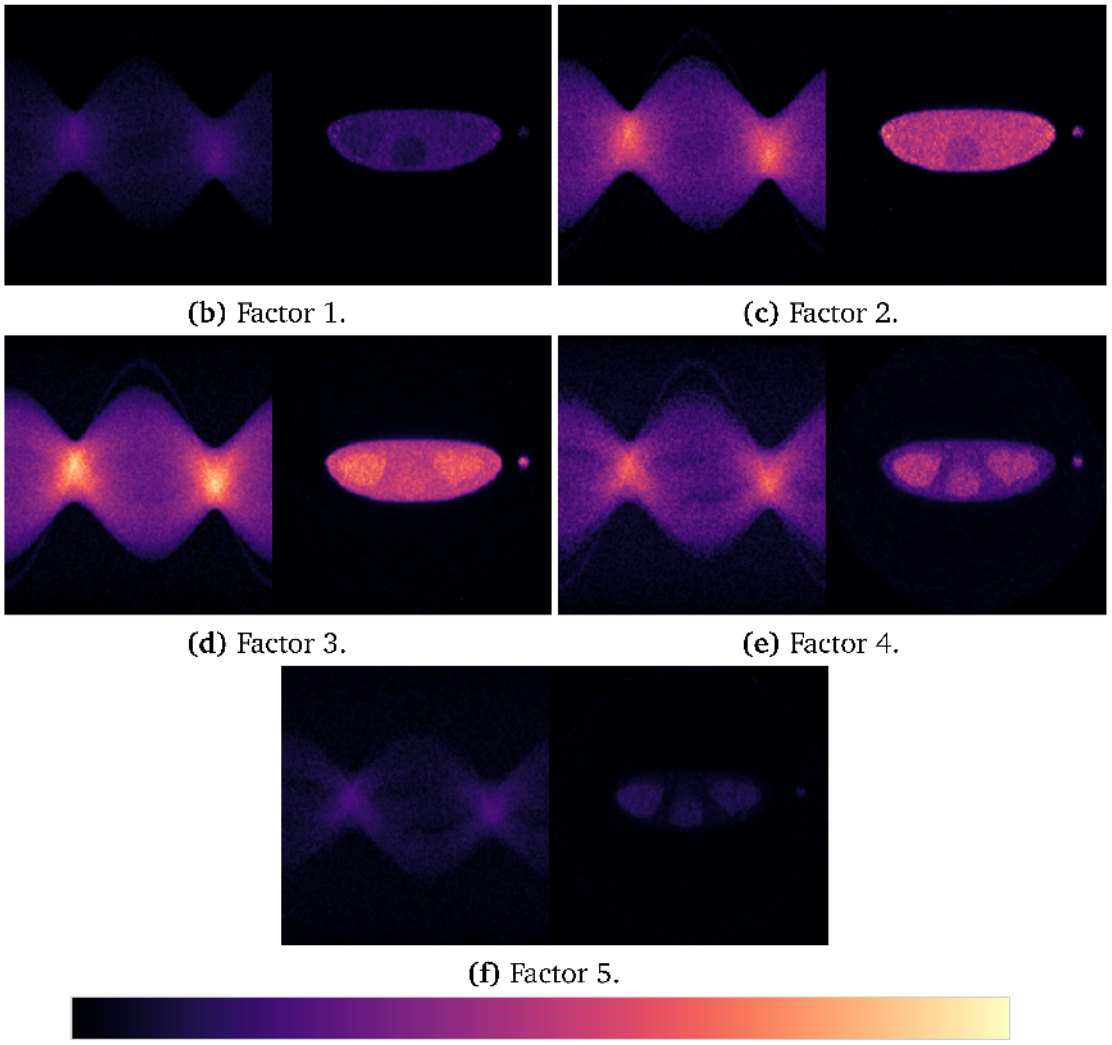
(b) Factor 1. **(c)** Factor 2.

(d) Factor 3. **(e)** Factor 4.

Figure 5.18.: Factorization results using AO-ADMM with factorization rank $k = 4$ and regularization parameters $\alpha_W = 20$ and $\alpha_H = 80$.



(a) Time activity curves.

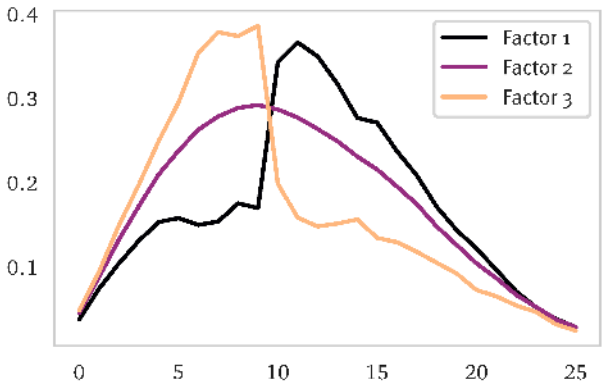


(b) Factor 1. **(c)** Factor 2.

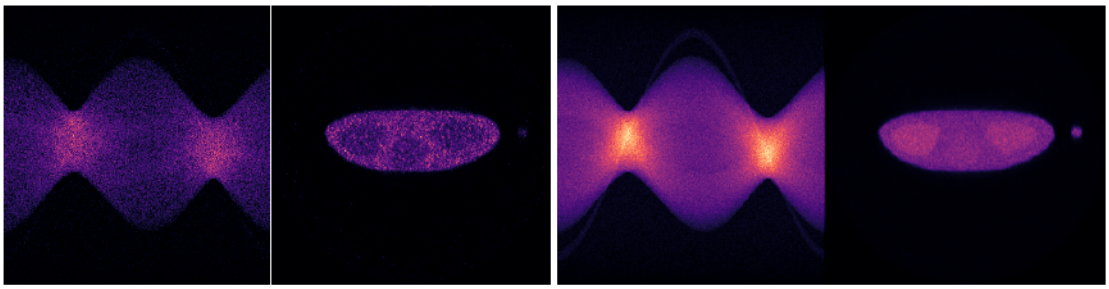
(d) Factor 3. **(e)** Factor 4.

(f) Factor 5.

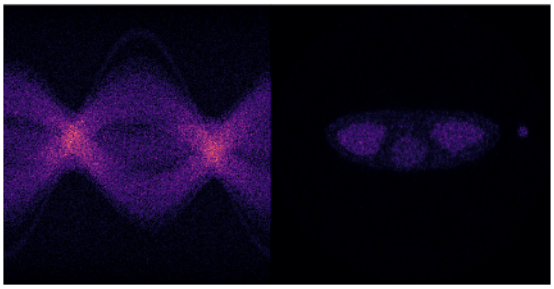
Figure 5.19.: Factorization results using AO-ADMM with factorization rank $k = 5$ and regularization parameters $\alpha_W = 20$ and $\alpha_H = 100$.



(a) Time activity curves.

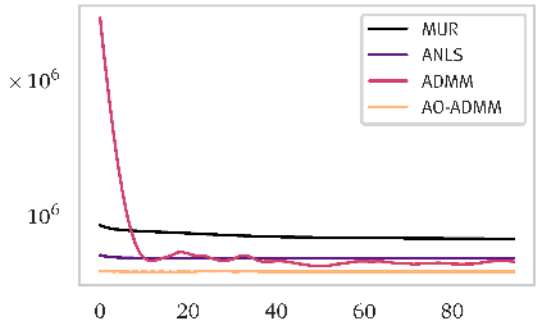


(b) Factor 1. **(c)** Factor 2.

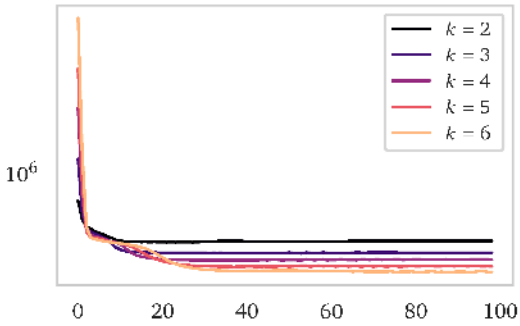


(d) Factor 3.

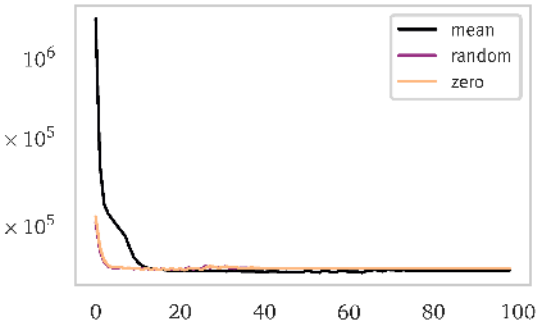
Figure 5.20.: Factorization results using AO-ADMM with the Euclidean distance D_{EU} as cost function, factorization rank $k = 3$ and regularization parameters $\alpha_W = 10$ and $\alpha_H = 300$.



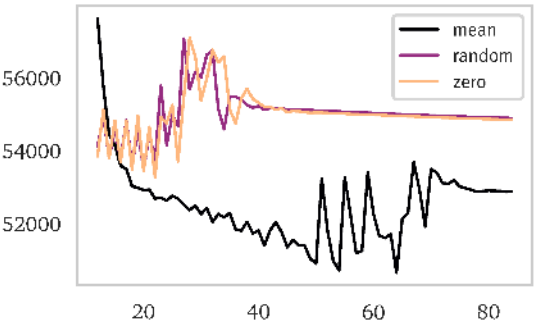
(a) Convergence plot for the different NMF algorithms presented in this section for $k = 3$.



(b) Convergence plot for different number of factors k using AO-ADMM.



(c) Convergence plot for different initializations for AO-ADMM.



(d) Convergence plot for different initializations for AO-ADMM (zoomed in).

Figure 5.21.: Convergence plots for different factorization results of the noisy dataset. The distance $D_{KL}(X, WH)$ is plotted logarithmically on the y-axis against the number of iterations on the x-axis.

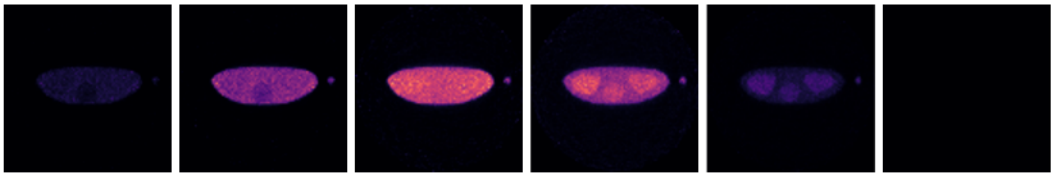
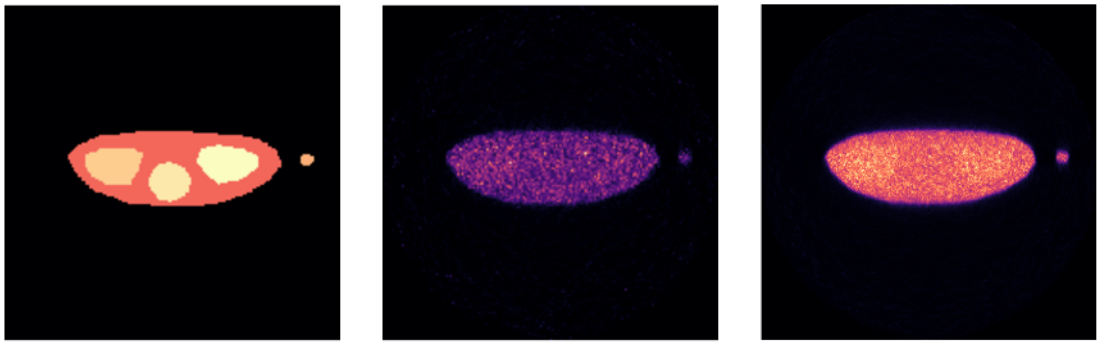


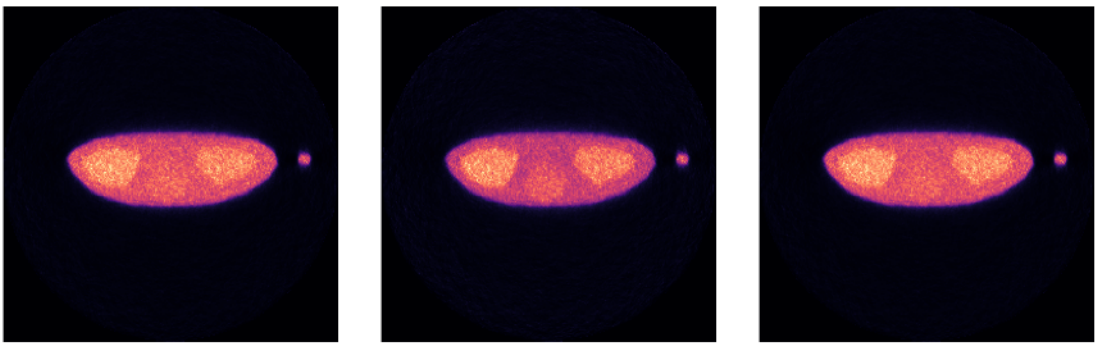
Figure 5.22.: Reconstructed factors using AO-ADMM, with $k = 6$, $\alpha_W = 20$, $\alpha_H = 80$.



(a) Frame 13 of the original dataset.

(b) Frame 13, directly reconstructed from the noisy data set using the EM-Algorithm.

(c) Frame 13, reconstructed from the factorization using MUR.

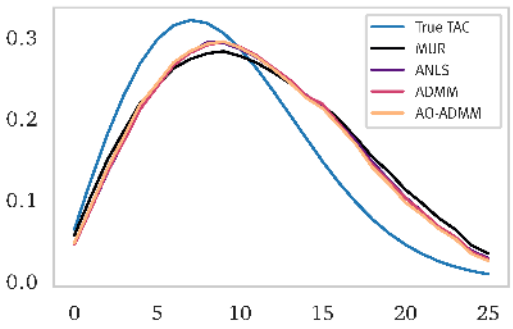


(d) Frame 13, reconstructed from the factorization using ANLS.

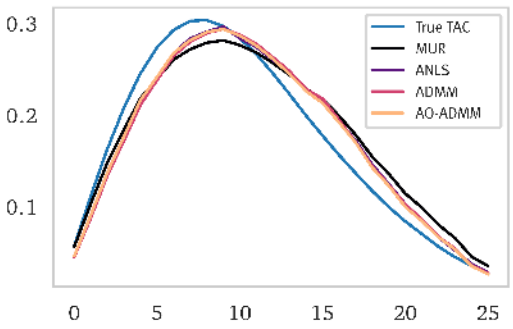
(e) Frame 13, reconstructed from the factorization using ADMM.

(f) Frame 13, reconstructed from the factorization using AO-ADMM.

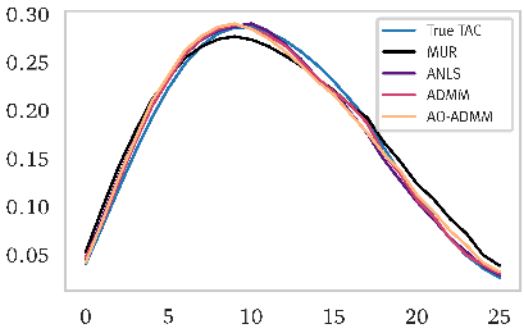
Figure 5.23.: The 13th frame of the original data set and of the reconstructions from the factorizations obtained by the different methods, using a factorization rank $k = 3$.



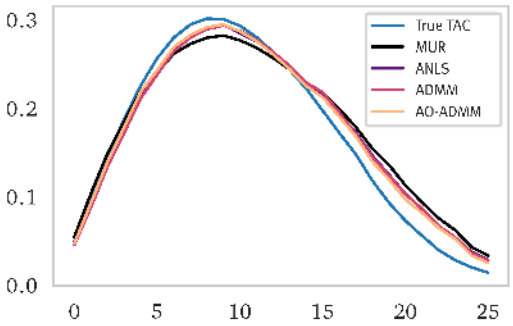
(a) Sample TAC from left lung.



(b) Sample TAC from right lung.



(c) Sample TAC from body tissue.



(d) Sample TAC from heart.

Figure 5.24.: Reconstructed time-activity curves from different regions. The activity averaged over a 5-by-5 pixel patch.

6 Conclusion

In this thesis, we have examined the use of non-negative matrix factorization for dynamic positron emission tomography. With the help of kinetic modeling, we built a motivation and interpretation for data analysis techniques in general and discussed the advantages of NMF over other factorization methods. After summarizing the properties of NMF, we introduced four different algorithms to solve the NMF problem, ranging from simple multiplicative update rules to more flexible methods based on the alternating direction method of multipliers. Finally, we investigated the performance of the algorithms using a synthetic dynamic PET dataset.

In general, NMF is well suited for non-negative data since the factorization does not depend on artificial conditions like statistical independence or orthogonality. When used on noise-free data, we were able to decompose the dataset into meaningful factors and time-activity curves especially using ANLS, ADMM, and AO-ADMM. In the more realistic case where a lot of Poisson noise is present, the more flexible and computationally expensive algorithms based on ADMM had a distinct advantage, with AO-ADMM resulting in the best results and being the most stable.

There are some drawbacks using NMF. The most obvious problem is the non-

uniqueness of NMF. We cannot guarantee to find the best solution, however, in practice, we do seem to find reasonable factors, especially when using apriori information in form of regularization. More importantly, the factorization results are even non-unique in terms of scaling, so that some form of normalizing has to be employed to make the factors and TACs comparable.

Another unsolved problem for this class of methods is the choice of the factorization rank. There is no clear criterion to choose the number of factors even in hindsight. For example, the relative error decreases with more factors since if all of the true signal is covered there are still degrees of freedom to account for the noise in the additional factors. A reasonable factorization can only be determined by manually examining the factors and TACs. One possible idea to mitigate this problem is by computing sufficiently many factors so that spare factors are present and then discard all factors below a certain norm value, assuming that the small factors only consists of noise. However, this is computationally more expensive and the norm threshold does not have a problem specific meaning, shifting the problem only from the number of factors to the choice of the threshold.

In general, the use of NMF seems to be well suited for dynamic PET dataset, even with large amounts of noise. Due to the flexibility of the ADMM algorithm, there are a lot of possible extensions that can make use of the specific problem properties, e.g. different cost functions and various types of regularization.

A Algorithms

A.1. NNDSVD

The NNDSVD algorithm by Boutsidis and Gallopoulos [2008] is based on the singular value decomposition of a dataset X. Given a factorization rank k, the k largest singular triplets, consisting of the left singular vectors U, the right singular vectors V and the singular values S. The factors W, H are then initialized depending on the norm of the positive and negative projections of the left and right singular vectors.

The whole algorithm is described in Algorithm 6. We use the notation a_j to reference the j-th column of a matrix A, and $\text{pos}(\cdot)$, $\text{neg}(\cdot)$ being defined as

$$\text{pos}(x) = \begin{cases} x, & \text{if } x > 0 \\ 0, & \text{otherwise} \end{cases}, \quad \text{neg}(x) = \begin{cases} -x, & \text{if } x < 0 \\ 0, & \text{otherwise} \end{cases} \tag{A.1}$$

and are applied component-wise.

In addition to Algorithm 6, Boutsidis and Gallopoulos [2008] also have introduced two dense NNDSVD variants. Taking the initializations from the NNDSVD algorithm, the NNDSVDa variant sets the zero entries to the arithmetic mean of the dataset X, while the NNDSVDar variant sets the zero entries to a

random value taken from a uniform distribution in $[0, \frac{\text{mean}(X)}{100}]$.

Algorithm 6 Nonnegative Double Singular Value Decomposition (NNDSVD)

Input: data matrix $X \in \mathbb{R}^{m \times n} \geq 0$, factorization rank k

Output: factorization matrices $W \in \mathbb{R}_+^{m \times k}, H \in \mathbb{R}_+^{k \times n}$

Compute k largest singular triplets (U, S, V) of X
Initialize $w_0 = \sqrt{s_0} * u_0$, $h_0 = \sqrt{s_0} * v_0^T$
for $i = 1, \ldots, k$ **do**
    $u_i^{\text{pos}} \leftarrow \text{pos}(u_i)$, $u_i^{\text{neg}} \leftarrow \text{neg}(u_i)$
    $v_i^{\text{pos}} \leftarrow \text{pos}(v_i)$, $v_i^{\text{neg}} \leftarrow \text{neg}(v_i)$
    $n^{\text{pos}} \leftarrow \|u_i^{\text{pos}}\|_2 * \|v_i^{\text{pos}}\|_2$
    $n^{\text{neg}} \leftarrow \|u_i^{\text{neg}}\|_2 * \|v_i^{\text{neg}}\|_2$
    if $n^{\text{pos}} > n^{\text{neg}}$ **then**
        $u \leftarrow \frac{u_i^{\text{pos}}}{\|u_i^{\text{pos}}\|_2}$, $v \leftarrow \frac{v_i^{\text{pos}}}{\|v_i^{\text{pos}}\|_2}$
        $\sigma \leftarrow n^{\text{pos}}$
    else
        $u \leftarrow \frac{u_i^{\text{neg}}}{\|u_i^{\text{neg}}\|_2}$, $v \leftarrow \frac{v_i^{\text{neg}}}{\|v_i^{\text{neg}}\|_2}$
        $\sigma \leftarrow n^{\text{neg}}$
    end if
    $w_i \leftarrow \sqrt{s_i \cdot \sigma} \cdot u$, $h_i \leftarrow \sqrt{s_i \cdot \sigma} \cdot v^T$
end for

A.2. FC-NNLS

The fast combinatorial non-negative least squares algorithm by VANBENTHEM AND KEENAN [2004] attempts to efficiently solve the NNLS problem with multiple right-hand sides, which can be stated as

$$\min_K \|CK - A\|_2^2 \quad \text{subject to } K \geq 0. \tag{A.2}$$

It uses the observation that in large-scale problems, the number of unique pseudoinverses required to solve the problem is much lower than the number of

right-hand side vectors. Grouping all right-hand side vectors that share the same pseudoinverse reduces the number of times the pseudoinverse has to be computed.

Algorithm 7 describes the main FC-NNLS algorithm, which uses a subroutine called combinatorial subspace least squares (CSSLS) as described in Algorithm 8. Note that a pre- and post-subscription notation is used to represent submatrices, e.g. for two index sets $\mathcal{P}, \mathcal{E}$, the submatrix ${}_{\mathcal{P}}K_{\mathcal{E}}$ contains the rows and columns according to the indices in $\mathcal{P}$ and $\mathcal{E}$.

Algorithm 7 Fast combinatorial NNLS (FC-NNLS)

Input: Coefficient matrix $C \in \mathbb{R}^{n \times l}$, observation matrix $A \in \mathbb{R}^{n \times p}$

Output: NNLS solution $K \in \mathbb{R}^{l \times p}$

Index columns of K: $\mathcal{M} = \{1, \ldots, p\}$
Index rows of K: $\mathcal{N} = \{1, \ldots, l\}$
Precompute constant parts of pseudoinverse, $C^T C, C^T A$
Solve $\min_K \|CK - A\|_F$ using CSSLS
Initialize passive sets: $\mathcal{P} = \{\mathcal{P}_1, \ldots, \mathcal{P}_p\}$, where $\mathcal{P}_j = \{x \in \mathcal{N} : {}_xK_j > 0\}$
Find non-optimal columns: $\mathcal{F} = \{j \in \mathcal{M} : \mathcal{P}_j \neq \mathcal{N}\}$
Overwrite solution: ${}_xK_j = 0 \forall j \in \mathcal{F}, \forall x \notin \mathcal{P}_j$
repeat
    Solve $\min_{K_\mathcal{F}} \|CK_\mathcal{F} - A_\mathcal{F}\|$ using CSSLS and $\mathcal{P}_\mathcal{F}$
    repeat
        Gather columns with neg. variables: $\mathcal{H} = \{j \in \mathcal{F} : \min_{x \in \mathcal{P}_j} xK_j < 0\}$
        $\forall i \in \mathcal{H}$ select variables to move out of the passive sets $\mathcal{P}_\mathcal{H}$
        Solve $\min_{K_\mathcal{H}} \|CK_\mathcal{H} - A_\mathcal{H}\|$ using CSSLS and $\mathcal{P}_\mathcal{H}$
    until $\mathcal{H} = \emptyset$
    Test K_i for optimality $\forall i \in \mathcal{F}$
    Remove indices of optimal solutions from $\mathcal{F}$
    $\forall i \in \mathcal{F}$ select variables to move into the passive sets $\mathcal{P}_\mathcal{F}$
until $\mathcal{F} = \emptyset$

Algorithm 8 Combinatorial subspace least squares (CSSLS)

Input: Coefficient matrix $C \in \mathbb{R}^{n \times l}$, observation matrix $A \in \mathbb{R}^{n \times p}$, passive set $\mathcal{P}$

Output: Solution $K \in \mathbb{R}^{l \times p}$

1: Initialize $K = 0$, index columns of K: $\mathcal{M} = \{1, \ldots, p\}$

2: Find set of k unique passive sets: $\mathcal{U} = \{\mathcal{U}_1, \ldots, \mathcal{U}_k\}$

3: Index columns of K with identical passive sets: $\mathcal{E}_j = \{i \in \mathcal{M} : \mathcal{P}_i = \mathcal{U}_i\}$

4: Solve: $\min_{\mathcal{U}_j K_{\mathcal{E}_j}} \|C_{\mathcal{U}_j \mathcal{U}_j} K_{\mathcal{E}_j} - A_{\mathcal{E}_j}\| \quad \forall j \in \{1, \ldots, k\}$

Bibliography

B., L., *Vereinfachte schematischer Darstellung eines doppelten Blutkreislaufs von gleichwarmen Wirbeltieren.*, `https://upload.wikimedia.org/wikipedia/commons/0/0c/Blutkreislauf_Gleichwarme.svg` (2005), [Online; accessed 19-September-2018]. 22

Barber, D. and Martel, A., *Factor analysis revisited.*, European journal of nuclear medicine (1992), 19: pp. 467–468. 30, 45

Barber, D.C., *The use of principal components in the quantitative analysis of gamma camera dynamic studies.*, Physics in medicine and biology (1980), 25: pp. 283–292. 30, 41

Benning, M., Heins, P. and Burger, M., *A solver for dynamic pet reconstructions based on forward-backward-splitting*, in *AIP Conference Proceedings*, volume 1281, AIP (2010) pp. 1967–1970. 36, 39

Benning, M., Kösters, T., Wübbeling, F., Schäfers, K. and Burger, M., *A nonlinear variational method for improved quantification of myocardial blood flow using dynamic $H_2{}^{15}O$ pet*, in *Nuclear Science Symposium Conference Record, 2008. NSS'08. IEEE*, IEEE (2008) pp. 4472–4477. 36, 39

Boutsidis, C. and Gallopoulos, E., *SVD based initialization: A head start for nonnegative matrix factorization*, Pattern Recognition (2008), 41(4): pp. 1350–1362. 52, 97

COMMUNITY EMERGENCY RESPONSE TEAM, *Schematic representation of a capillary.*, https://upload.wikimedia.org/wikipedia/commons/f/f6/Capillary_system_CERT.jpg (2013), [Online; accessed 29-August-2018]. 22

CUNNINGHAM, V.J. AND JONES, T., *Spectral analysis of dynamic pet studies*, Journal of Cerebral Blood Flow & Metabolism (1993), 13(1): pp. 15–23, pMID: 8417003. 33

DAUBE-WITHERSPOON, M.E. AND MUEHLLEHNER, G., *An iterative image space reconstruction algorithm suitable for volume ECT*, IEEE transactions on medical imaging (1986), 5: pp. 61–66. 52

DEMPSTER, A.P., LAIRD, N.M. AND RUBIN, D.B., *Maximum likelihood from incomplete data via the EM algorithm*, Journal of the Royal Statistical Society, Series B (1977), 39(1): pp. 1–38. 28

DONOHO, D. AND STODDEN, V., *When does non-negative matrix factorization give a correct decomposition into parts?*, in *Advances in neural information processing systems* (2004) pp. 1141–1148. 47, 48, 49

HEINS, P., *Reconstruction Using Local Sparsity: A Novel Regularization Technique and an Asymptotic Analysis of Spatial Sparsity Priors*, Ph.D. thesis, Westfälische Wilhelms-Universität Münster (2014). 35

HUANG, K., SIDIROPOULOS, N.D. AND LIAVAS, A.P., *Efficient algorithms for 'universally' constrained matrix and tensor factorization*, in *Signal Processing Conference (EUSIPCO), 2015 23rd European*, IEEE (2015) pp. 2521–2525. 58

HUANG, K., SIDIROPOULOS, N.D. AND LIAVAS, A.P., *A flexible and efficient algorithmic framework for constrained matrix and tensor factorization*, IEEE Transactions on Signal Processing (2016), 64(19): pp. 5052–5065. 58

HUANG, K., SIDIROPOULOS, N.D. AND SWAMI, A., *Non-negative matrix factorization revisited: Uniqueness and algorithm for symmetric decomposition*, IEEE Transactions on Signal Processing (2014), 62(1): pp. 211–224. 49

IIDA, H., KANNO, I., TAKAHASHI, A., MIURA, S., MURAKAMI, M.T., TAKAHASHI, K., ONO, Y., SHISHIDO, F., INUGAMI, A. AND TOMURA, N., *Measurement of absolute myocardial blood flow with $H_2{}^{15}O$ and dynamic positron-emission tomography*, Circulation (1988), 78(1): pp. 104–115. 37

KIM, H. AND PARK, H., *Nonnegative matrix factorization based on alternating nonnegativity constrained least squares and active set method*, SIAM journal on matrix analysis and applications (2008), 30(2): pp. 713–730. 55

LANGNER, J., *Development of a Parallel Computing Optimized Head Movement Correction Method in Positron-Emission-Tomography*, Master's thesis, University of Applied Sciences, Dresden (2003a). 25

LANGNER, J., *ECAT Exact HR+ PET scanner*, `https://upload.wikimedia.org/wikipedia/commons/b/b8/ECAT-Exact-HR--PET-Scanner.jpg` (2003b), [Online; accessed 29-August-2018]. 23

LAURBERG, H., CHRISTENSEN, M.G., PLUMBLEY, M.D., HANSEN, L.K. AND JENSEN, S.H., *Theorems on positive data: on the uniqueness of NMF*, Computational intelligence and neuroscience (2008), p. 764206. 49

LAWSON, C.L. AND HANSON, R.J., *Solving least squares problems*, volume 15 of *Classics in Applied Mathematics*, Society for Industrial and Applied Mathematics (SIAM), Philadelphia, PA (1995), revised reprint of the 1974 original. 55

LAWTON, W.H. AND SYLVESTRE, E.A., *Self modeling curve resolution*, Technometrics (1971), 13(3): pp. 617–633. 46

LECHARLIER, L. AND DE MOL, C., *Regularized blind deconvolution with poisson data* (2013), 464: p. 012003. 53

LEE, D.D. AND SEUNG, H.S., *Learning the parts of objects by non-negative matrix factorization*, Nature (1999), 401(6755): p. 788. 41, 46

LEE, D.D. AND SEUNG, H.S., *Algorithms for non-negative matrix factorization*, in

Advances in neural information processing systems (2001) pp. 556–562. 41, 46, 52, 53, 64

MÜLLER, J., *Advanced Image Reconstruction and Denoising. Bregmanized (Higher Order) Total Variation and Application in PET*, Ph.D. thesis, Westfälische Wilhelms-Universität Münster (2013). 28

NATIONAL HEART, LUNG, AND BLOOD INSTITUTE, `https://www.nhlbi.nih.gov` (2018), [Online; accessed 29-August-2018]. 18

NATTERER, F., *The Mathematics of Computerized Tomography*, Classics in Applied Mathematics, Society for Industrial and Applied Mathematics (2001). 26

NATTERER, F. AND WÜBBELING, F., *Mathematical methods in image reconstruction*, SIAM (2001). 26

PAATERO, P. AND TAPPER, U., *Positive matrix factorization: A non-negative factor model with optimal utilization of error estimates of data values*, Environmetrics (1994), 5(2): pp. 111–126. 46

PARIKH, N., BOYD, S. ET AL., *Proximal algorithms*, Foundations and Trends in Optimization (2014), 1(3): pp. 127–239. 57

PEARSON, K., *On lines and planes of closest fit to systems of points in space*, The London, Edinburgh, and Dublin Philosophical Magazine and Journal of Science (1901), 2(11): pp. 559–572. 42

PORTUGAL, L.F., JUDICE, J.J. AND VICENTE, L.N., *A comparison of block pivoting and interior-point algorithms for linear least squares problems with nonnegative variables* (1994), 63. 55

RADON, J., *Über die Bestimmung von Funktionen durch ihre Integralwerte längs gewisser Mannigfaltigkeiten*, Akad. Wiss. (1917), 69: pp. 262–277. 25

READER, A.J., MATTHEWS, J.C., SUREAU, F.C., COMTAT, C., TREBOSSEN, R. AND BUVAT, I., *Fully 4d image reconstruction by estimation of an input function*

and spectral coefficients, in *Nuclear Science Symposium Conference Record, 2007. NSS'07. IEEE*, volume 5, IEEE (2007) pp. 3260–3267. 33

READER, A.J., SUREAU, F.C., COMTAT, C., TRÉBOSSEN, R. AND BUVAT, I., *Joint estimation of dynamic PET images and temporal basis functions using fully 4D ML-EM*, Physics in Medicine and Biology (2006), 51(21): p. 5455. 33

SCHMIDLIN, P., *Quantitative evaluation and imaging of functions using pattern recognition methods*, Physics in Medicine & Biology (1979), 24(2): p. 385. 41

SCHÄFERS, K.P., SPINKS, T.J., CAMICI, P.G., BLOOMFIELD, P.M., RHODES, C.G., LAW, M.P., BAKER, C.S.R. AND RIMOLDI, O., *Absolute quantification of myocardial blood flow with $H_2{}^{15}O$ and 3-dimensional PET: an experimental validation.*, Journal of nuclear medicine (2002), 43: pp. 1031–40. 30

SHEPP, L.A. AND VARDI, Y., *Maximum likelihood reconstruction for emission tomography*, IEEE Transactions on Medical Imaging (1982), 1(2): pp. 113–122. 26, 28

SUN, D. AND FEVOTTE, C., *Alternating direction method of multipliers for non-negative matrix factorization with the beta-divergence* (2014), pp. 6201–6205. 56, 61

TAM, B.S., *A geometric treatment of generalized inverses and semigroups of nonnegative matrices* (1981), 41: pp. 225–272. 48

VAN BENTHEM, M.H. AND KEENAN, M.R., *Fast algorithm for the solution of large-scale non-negativity-constrained least squares problems*, Journal of chemometrics (2004), 18(10): pp. 441–450. 55, 98

VARDI, Y., SHEPP, L. AND KAUFMAN, L., *A statistical model for positron emission tomography*, Journal of the American statistical Association (1985), 80(389): pp. 8–20. 29

VAVASIS, S.A., *On the complexity of nonnegative matrix factorization*, SIAM J. on Optimization (2009), 20(3): pp. 1364–1377. 50

WANG, H. ET AL., *Global, regional, and national life expectancy, all-cause mortality, and cause-specific mortality for 249 causes of death, 1980-2015: a systematic analysis for the global burden of disease study 2015*, The Lancet (2018), 388(10053): pp. 1459–1544. 17

ZHANG, S., HUANG, D.Y., XIE, L., CHNG, E., LI, H. AND DONG, M., *Regularized non-negative matrix factorization using alternating direction method of multipliers and its application to source separation* (2015). 56