DTU Compute
Department of Applied Mathematics and Computer Science

Parametric Shift-Invariant Modeling of Nuclear Magnetic Resonance Data

Bachelor Thesis

Lucas Sylvester Høyberg-Nielsen & Lucas Mørch Emcken

Parametric Shift-Invariant Modeling of Nuclear Magnetic Resonance Data

Bachelor Thesis
June, 2024

By
Lucas Sylvester Høyberg-Nielsen & Lucas Mørch Emcken

Cover photo: Vibeke Hempler, 2012

Approval

This bachelor thesis was prepared at the department of Applied Mathematics and Computer Science at the Technical University of Denmark in partial fulfillment of the requirements for acquiring the degree Bachelor of Science (B.Sc.) in Engineering in Artificial Intelligence and Data Science. The project was conducted under the supervision of Prof. Morten Mørup and PostDoc. Jesper Løve Hinrich from the Department of Applied Mathematics and Computer Science at DTU between February 2024 and June 2024.

It is assumed that the reader has a basic knowledge in the areas of statistics.

Lucas Sylvester Høyberg-Nielsen- s214636

.....

Signature

.....

Date

Lucas Mørch Emcken - s214625

.....

Signature

.....

Date

Abstract

In this thesis we explore the development and application of parametric shift-invariant modeling to analyze Nuclear Magnetic Resonance (NMR) data. Combining hard modeling techniques with shifted Non-negative Matrix Factorization (shiftNMF), a novel framework was created to extract detailed insights from complex NMR datasets. The model was validated on synthetic and real-world data, including wine samples and data from the Human Metabolome Database (HMDB). The results demonstrate the model's capability to accurately identify and quantify chemical components, even with overlapping peaks and variable shifts. The model could accurately identify underlying multiplet parameters and gain insightful knowledge of the structure of the underlying compounds.

Acknowledgements

Lucas Sylvester Høyberg-Nielsen- s214636
Lucas Mørch Emcken - s214625

Special thanks to our supervisor Jesper Løve Hinrich and Morten Mørup.

AI use

ChatGPT, Grammarly, and Copilot were used to provide proposals on grammar and clearer phrasing and to aid in translating code from MatLab to Python.

Contents

1 Introduction

1.1 Motivation

Nuclear Magnetic Resonance (NMR) spectroscopy is a powerful and widely used analytical technique for characterizing substances at the molecular level. It has extensive applications in diverse fields, from chemistry to biology, medicine, and materials science. The technique's strength lies in its ability to identify specific compounds and quantify their concentrations within complex chemical mixtures[1].

The majority of NMR studies employ 1D Hydrogen-NMR (H-NMR), which measures the resonances of hydrogen atoms. The data obtained from these measurements typically take the form of a graph featuring peaks or clusters of peaks. These peaks serve as fingerprints for the chemical bonds that the hydrogen atoms participate in, thereby facilitating the identification of the structure of the organic compound[1].

NMR often faces challenges with its signal-to-noise ratio. The presence of noise can obscure the signal, making it challenging to accurately identify and quantify the chemical entities. This issue is particularly pronounced in low-concentration samples, where the signal is inherently weak. Noise can originate from various sources, including thermal noise in the electronics, environmental interference, and imperfect shimming. These factors can significantly affect the quality of the NMR data, necessitating advanced noise reduction techniques and careful experimental setup to ensure reliable results.

The interpretation of NMR spectra is often further complicated by non-linear effects such as peak shifts, peak broadening, and baseline distortions. These artifacts, commonly present in NMR data, can interfere with post-processing techniques and make the spectrum less interpretable. It is not always feasible or desirable to correct these artifacts a priori[2][3].

While direct analysis of the spectrum can provide answers to many chemical questions, the extraction of all useful information often requires additional processing methods. These include direct peak integration and peak picking[4]. **Peak picking** is the process of identifying and quantifying the signal peaks in the spectrum. Each of these peaks corresponds to a specific resonance, representing a particular chemical entity in the sample. **Direct peak integration** is the process of identifying appropriate integration intervals so the area under each peak can be calculated. This is typically done using numerical methods.

Peak Picking followed by Direct Peak Integration is a primary approach to quantitatively analyzing NMR spectra. It enables the determination of the relative numbers of spins responsible for different multiplets and the measurement of chemical shifts and scalar coupling[4]. However, peak shifts, peak broadening, baseline distortions as well as overlapping peaks can complicate the task of identification of peaks during peak picking and capturing all NMR signals within the appropriate integration range during direct peak integration[5][6].

Therefore, reliable and robust methods are vital components of NMR spectroscopy. They help interpret NMR data and address problems related to spectral artifacts.

1.2 State-of-the-art

Several state-of-the-art methods can address different aspects of the aforementioned spectral artifacts and provide additional insight into data structure.

Data alignment

One of the most notable algorithms for the alignment of spectroscopic data is the Icoshift algorithm. This algorithm is particularly effective because it divides the data into discrete segments, which are then systematically shifted to maximize cross-correlation with a reference spectrum. This process eliminates spectral shifts, ensuring more accurate and reliable data alignment. The efficacy and robustness of the Icoshift algorithm have been well-documented in the literature, highlighting its utility in various chemometric applications[3][7][8].

Baseline fitting

Baseline removal methods are crucial in spectroscopic data processing as they aim to model the spectrum's baseline accurately. This is typically achieved by fitting mathematical functions, such as polynomials or spline functions, to the baseline component of the spectrum. Several reliable semi-automatic approaches exist, requiring some initial manual baseline identification. Fully automated baseline correction exists but only for simple spectres[9]. Once the baseline is successfully modeled, it is subtracted from the original spectrum to produce a baseline-corrected spectrum. This correction is essential for enhancing the accuracy and interpretability of the spectral data[8].

Decomposition methods

Latent variable decomposition is a valuable tool for NMR analysis, and soft models like Non-Negative-Matrix-Factorization(NMF) and Multivariate Curve Resolution (MCR) have previously proven useful in finding underlying components[10][6]. Furthermore, extensions of these models like shift-NMF have been capable of taking into account spectral shifts[11]. Decomposition tools eliminate overlapping peaks and provide useful information about relative concentrations of components associated with compounds.

NMF approximates the original data matrix $\mathbf{X}$ as a component of two non-negative matrices, $\mathbf{W}$ and $\mathbf{H}$, such that:

$$\mathbf{X} \approx \mathbf{WH}$$

In the context of NMR, $\mathbf{W}$ corresponds to the weights, and $\mathbf{H}$ corresponds to the basis spectra. The non-negativity constraint ensures that the components are additive, which is a natural requirement for spectral data. This parts-based representation makes the interpretation of the results straightforward, as each observation is constructed as a mixture of the latent signals in $\mathbf{H}$ given by the weights in $\mathbf{W}$.

ShiftNMF extends this concept by incorporating shifts in the data domain, which is crucial for handling the peak shifts in NMR specters. The model can be described as:

$$\mathbf{X}_{n,m} \approx \mathbf{W}_{n,d}\mathbf{H}_{d,m-\boldsymbol{\tau}_{n,d}}$$

Where τ represents the shift parameters, which allows the model to capture the time delays or shifts in the peaks, improving the alignment and interpretation of the spectra

Matrix factorization is a powerful mathematical technique that decomposes complex data matrices into simpler, interpretable components. In NMR analysis, techniques such as NMF and shiftNMF decompose the NMR spectra into a set of basis spectra and corresponding coefficients. This decomposition reveals the underlying chemical compounds

present in the sample, making it easier to interpret NMR data[6]. ShiftNMF, in particular, accounts for shifts in the data, which is crucial for handling peak shifts in NMR spectra, thus enhancing alignment and interpretation.

We can align and separate components in the NMR specter by applying shifted matrix factorization techniques to NMR data. This reduces complexity and helps gain insights into the relative concentrations of the compounds present, enhancing the overall interpretability of NMR analysis.

Hardmodelling

Hard modeling uses known physical profiles to describe the peaks, making the spectrum more easily interpretable by eliminating noise [12]. Current hard modeling methods include Indirect Hard Modeling (IHM)[13]. IHM utilizes a non-linear spectral hard model by peak fitting of the pure spectra, followed by a linear calibration model to predict the concentrations[14][15]. The main advantage of using the automatic IHM method is that it can model non-linear effects on a per sample basis[14]. Making it capable of modeling the non-linear variation of peak parameters between samples, which in turn makes the weights more suitable for calibration and calculation of compound concentrations[12]. One significant drawback of IHM is that it requires known pure component spectra[14].

1.3 Project Description

By integrating hard modeling techniques with soft decomposition models, we can make progress toward developing a joint model. This joint model incorporates the advantages of both techniques. First, it can decompose the initially convoluted data. Second, it can extract the spectra of pure components. Third, it can model non-linear effects and generate simple, interpretable components incorporating domain knowledge.

In this project, we want to investigate combinations of ”soft” decomposition models with hard modeling techniques to automatically identify and assign multiplet structures to pure components in the PyTorch framework. In the project, we will implement the PyTorch gradient version of shiftNMF to account for shifts and identify pure underlying components, detangling overlapping peaks. From there, we will use different estimation methods to initialize and fit single Voigt profiles to identify peaks. Using the single peak fittings, we will generate sets of multiplet hypotheses. Using the NLARS regularization algorithm, we will identify the most significant multiplet components that best fit the pure component spectra.

1.4 Objective and Research questions

This bachelor project aims to develop and facilitate a new parametric shift-invariant approach to modeling NMR data. To facilitate this goal, we have formulated three research questions:

- **How can PyTorch be used as an efficient framework to optimize shiftNMF?**

  By implementing the shiftNMF method into the PyTorch framework, we wish to streamline the optimization process by utilizing its automatic differentiation capabilities to optimize the implicit loss function. This will show the shortcomings of the naive implementation and explain why an alternative method of estimating the shifts is necessary for optimizing shiftNMF in PyTorch.

- **How can the latent components of shiftNMF be translated into structures for the hard model approach?**

By separating an NMR specter into its compounds via shiftNMF, we will use the identified latent components to construct parameterized hard models of the pure compounds, which lend themselves to a high degree of interpretability. Furthermore, we will create a basis for a joint shiftNMF model using hard components capable of fitting non-linear effects.

- **How can the developed methodology improve our understanding of NMR data?** By applying the model to synthetic studies and real NMR data to extract the true components and identify the compounds, we can potentially improve NMR as a tool for chemical analysis. Additionally, we can test our hard modeling approaches ability to extract underlying multiplets parameters and automate analytical methods used to interpret NMR data. This will allow us to directly inspect and evaluate the quality of extracted multiplicity, J-coupling, and the width and position of multiplets.

By addressing these research questions, we attempt to provide insights into the effectiveness of combining different models to create a joint model pipeline. The findings of this study could be valuable for NMR analysts seeking to develop more effective analytical strategies. Furthermore, they present significant steps towards a joint model capable of using both hard modeling techniques capable of handling non-linear effects, such as peak broadening, and soft modeling techniques capable of handling complex and overlapping peaks, extracting pure components, mixing, and shifts.

2 Methods

2.1 Hard modelling

Hard modelling reconstructs a dataset using parameterized line shape models of individual peaks instead of flexible underlying components. Data can easily be modelled using individual peaks, but by using peak constructs known as multiplets we can gain additional insight into the data, as they contain information about properties of the compounds present in the sample. In addition to contaning information about the cluster mean, multiplicity, and J-coupling. Additionally the integral value of the multiplets present in pure component spectra relative to each are proportional to the number of identical nuclei (Protons in the case of 1H NMR)[16].

Line shape modelling

Individual peaks can be modelled using Gaussian and Lorentzian profiles, as well as their convolution known as the Voigt profile[15]. Where $G\left(x';\sigma\right)$ is the centered Gaussian profile and $L\left(x-x';\gamma\right)$ is the centered Lorentzian profile

$$V(x;\sigma,\gamma) \equiv \int_{-\infty}^{\infty} G\left(x';\sigma\right) L\left(x-x';\gamma\right) dx'$$

In the limiting cases of $\sigma = 0$ and $\gamma = 0$ the function simplifies to the Lorentzian and Gaussian function respectively.

Pseudo Voigt function

For our purpose we will consider the Pseudo Voigt function as it is much less computationally expensive. It simplifies the convolution to a simple linear combination of Gaussian and Lorentzian profiles with a shared width parameter Γ (FWHM), weighted by an additional parameter n(values between 0 and 1)[17].

$$PV(x;\mu,\Gamma,n) \equiv n \cdot G(x-\mu,\Gamma/(2\cdot\sqrt{2\cdot \ln 2}) + (1-n)\cdot L(x-\mu,\Gamma/2)$$

Multiplets

Multiplets are collections of peaks with known relative scales and equal distance. The relative scale of the peaks follow the structure of pascal triangle, as seen in fig. 2.1. The spacing (J-coupling) and width is always equal between peaks.

A single multiplet with single peaks modelled using the pseudo Voigt function can therefore be described as:

$$M(x;\mu,\Gamma,n,j,m) = \sum_{i=0}^{m-1} \binom{m}{i} \cdot \begin{cases} PV(x;\mu - m/2\cdot j + j\cdot i,\Gamma,n) & m \in 2Z \\ PV(x;\mu - m/2\cdot j + j/2 + j\cdot i,\Gamma,n) & m \notin 2Z \end{cases}. \quad (2.1)$$

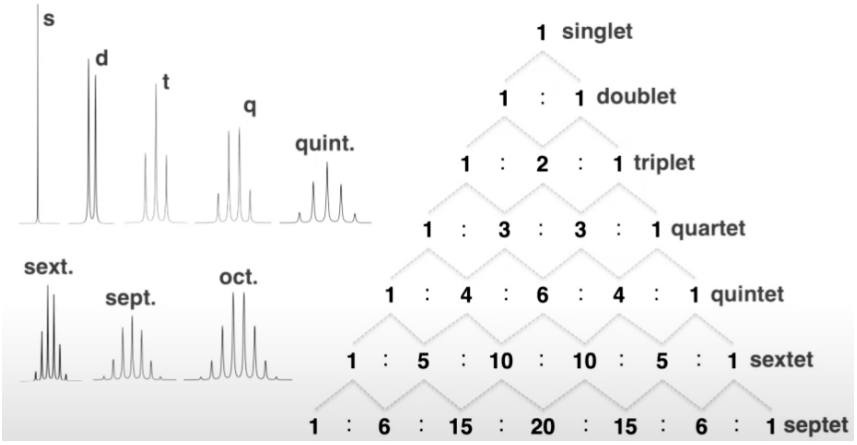


Figure 2.1: Simple multiplet structures following pascals triangle

2.2 Non-negative Matrix Factorization (NMF)

NMF is as previously stated a matrix factorization technique that finds a basis and weights in order to estimate the data matrix as $\mathbf{X} \approx \mathbf{WH}$. As the name suggests we do this approximation under the constraints that we want our basis vectors and weights to be non-negative, $\mathbf{W}, \mathbf{H} \geq 0$ and we also assume non-negativity in the data-matrix, $\mathbf{X} \geq 0$. This constraint gives a component-based representation, which is easy to interpret[18]. Each observation is constructed as a mixture of the latent signals in **H** given by the weights in **W**.

Algorithm for NMF

There exists a variety of algorithms for NMF but they all revolve around defining some loss-function, L and then choosing the **W** and **H** that minimize this loss-function. A simple choice of loss-function is the sum of squared differences [11].

$$L = \frac{1}{2}||\mathbf{X} - \mathbf{WH}||_F^2 \tag{2.2}$$

Where $||\cdot||_F$ is the Frõbenius norm defined as $||\mathbf{A}||_F = \sqrt{\sum_i \sum_j \mathbf{A}_{i,j}^2}$. We can then calculate the gradient of the loss function, and proceed with gradient descent until convergence. This method is easily implemented with automatic gradient tools such as Pytorch [19]. The non-negativity constraint is handled by passing the estimated matrices through the softplus-function ensures non-negativity.

2.3 Shifted Non-negative Matrix Factorization (shiftNMF)

ShiftNMF proposed in ”Shifted non-negative matrix factorization”[11] uses the same decomposition approach as NMF but includes an extra shift parameter τ. The model can be described as

$$\mathbf{X}_{n,m} \approx \mathbf{W}_{n,d}\mathbf{H}_{d,m-\tau_{n,d}} \tag{2.3}$$

. The entries in the τ matrix can be seen as time-delays from the d'th latent component to the n'th observation. $\boldsymbol{\tau}$ is constrained to appropriate shift intervals and the idea is that the model learns appropriate shifts for for each underlying component to describe the natural variant in shift each samples has..

Algorithm for shiftNMF

We wish to optimize the parameters $\mathbf{W}$, $\mathbf{H}$ and τ. We define a loss-function to optimize over all parameters. We use the sum of squared errors as before

$$L = \sum_{n,m} \frac{1}{2}(\mathbf{X}_{n,m} - \sum_{d} \mathbf{W}_{n,d}\mathbf{H}_{d,m-\tau_{n,d}}) \tag{2.4}$$

Rewriting the cost-function in frequency space

To reduce computation, we rewrite the cost-function into the frequency domain.

We begin by recalling the Fourier transform of a function x given by:

$$X(f) = \int_{-\infty}^{\infty} x(t)e^{-i2\pi ft}\,\mathrm{d}t \tag{2.5}$$

Consider a time shifted function $x(t - t_0)$ and let $\mathcal{F}(\cdot)$ define the Fourier transform, We have:

$$\mathcal{F}(x(t - t_0))(f) = \int_{-\infty}^{\infty} x(t - t_0)e^{-i2\pi ft}\,\mathrm{d}t$$

Using the substitution $u = t - t_0$ we can express this as:

$$\begin{aligned}\int_{-\infty}^{\infty} x(u)e^{-i2\pi f(u+t_0)}\,\mathrm{d}u &= \int_{-\infty}^{\infty} x(u)e^{-i2\pi f(u+t_0)}\,\mathrm{d}u \\ &= e^{-i2\pi ft_0}\int_{-\infty}^{\infty} x(u)e^{-i2\pi fu}\,\mathrm{d}u \\ &= e^{-i2\pi f(t_0)}X(f)\end{aligned}$$

Which shows that a time-domain shift corresponds to multiplication by a complex exponential in the frequency domain. We will use this property to rewrite the shiftNMF model in the frequency domain. Since X represents a discrete signal we will approximate it using the discrete Fourier transform (DFT).

$$X(f) = \sum_{n=0}^{N-1} x(n) \cdot e^{i2\pi kf/N} \tag{2.6}$$

Recognizing that a shift of τ in the time domain is equivalent to a multiplication by $e^{-i2\pi f\tau}$ in the frequency domain we can express the model as

$$\hat{\mathbf{X}}_{n,f} \approx \sum_{d} \mathbf{W}_{n,d}\hat{\mathbf{H}}_{d,f}e^{-i2\pi\frac{f-1}{M}\tau_{\mathbf{n},\mathbf{d}}} \tag{2.7}$$

Here, $\hat{\mathbf{A}}$ denotes a matrix transformed to the frequency domain via the Discrete Fourier Transform. From Equation 2.7, we can observe that each entry in the reconstructed observation matrix, $\mathbf{W}\hat{\mathbf{H}}$, is scaled by a complex exponential. If we define $\widetilde{\mathbf{W}}^{(f)}$ as $\mathbf{W}$ but where each entry n, d is multiplied by the complex exponential $e^{-i2\pi\frac{f-1}{M}\tau_{n,d}}$, we can write the model in matrix notation as:

$$\hat{\mathbf{X}}_f \approx \tilde{\mathbf{W}}^{(f)}\hat{\mathbf{H}}_f \tag{2.8}$$

Using $\hat{\cdot}$ to represent a matrix in the frequency domain, we want to write a cost-function using the Fourier transformed model. To do so, we utilize Parseval's identity which describes the summability of a Fourier series of a function. For a square-integrable function $x(t)$ it is given by:

$$\int_{-\infty}^{\infty} |x(t)|^2 \, \mathrm{d}t = \frac{1}{2\pi} \int_{-\infty}^{\infty} |X(\omega)|^2 \tag{2.9}$$

For a function, f, to be square-integrable on an interval $[a, b]$ it must hold that

$$\int_{a}^{b} |f(x)|^2 \, \mathrm{d}x < \infty \tag{2.10}$$

Parsevals identity arises when we take the square of the absolute value as the product of the function with its complex conjugate

$$\int_{-\infty}^{\infty} |x(t)|^2 \, \mathrm{d}t = \int_{-\infty}^{\infty} x(t)x^*(t) \, \mathrm{d}t \tag{2.11}$$

By inserting the inverse Fourier transform into the right hand side we can derive:

$$\int_{-\infty}^{\infty} x(t)x^*(t) \, \mathrm{d}t = \int_{-\infty}^{\infty} x(t)[\frac{1}{2\pi} \int_{-\infty}^{\infty} X(\omega)e^{-i\omega t} \, \mathrm{d}\omega] \, \mathrm{d}t \tag{2.12}$$

$$= \frac{1}{2\pi} \int_{-\infty}^{\infty} X^*(\omega)[\int_{-\infty}^{\infty} x(t)e^{-i\omega t} \, \mathrm{d}t] \, \mathrm{d}\omega \tag{2.13}$$

$$= \frac{1}{2\pi} \int_{-\infty}^{\infty} X(\omega)X^*(\omega)\mathrm{d}\omega \tag{2.14}$$

$$= \frac{1}{2\pi} \int_{-\infty}^{\infty} |X(\omega)|^2\mathrm{d}\omega \tag{2.15}$$

Where we used the angular frequency, $\omega = 2\pi f$, instead of the linear frequency. This identity is useful, since the time-domain cost-function 2.3 is the sum of squares of the function. Parseval's theorem also holds in the discrete cases and is expressed as:

$$\sum_{n=0}^{N-1} |x(n)|^2 = \frac{1}{N} \sum_{k=0}^{N-1} |X(k)|^2 \tag{2.16}$$

A proof of Parseval's theorem for the DFT is included in appendix ii The cost-function can now be re-written in the frequency domain. Recall the cost function as stated in 2.4. Here we sum over the squared elements of a matrix in the time-domain, so using 2.16 and the model in the frequency domain 2.8, the cost-function can be expressed as:

$$L = \sum_{n,m} \frac{1}{2}(\mathbf{X}_{n,m} - \sum_{d} \mathbf{W}_{n,d}\mathbf{H}_{d,m-\tau_{n,d}})^2 = \frac{1}{2M}||\hat{\mathbf{X}}_f - \tilde{\mathbf{W}}^{(f)}\hat{\mathbf{H}}_f||_F^2 \tag{2.17}$$

Parseval's theorem is applicable because the left-hand side sums the squares of the signal, while the right-hand side sums the squares in the frequency domain. Due to the equality given by 2.17 we can optimize H, W and τ in the frequency domain in order to minimize the right-hand side of 2.17 and the corresponding parameter in the time-domain will be optimal for the left-hand side of 2.17. We update the parameters using automatic gradients from PyTorch, stepping in the direction of the gradient after constructing the data approximation with a forward pass and calculating the loss.

Methods for shiftNMF optimization

For implementing and optimizing shiftNMF in Python, we are interested in comparing the naive PyTorch implementation that updates all the model parameters using gradients, and the cross correlation method proposed in "Shifted Non-Negative Matrix Factorization"[11].

Naive gradient based PyTorch implementation

One way to optimize the shiftNMF problem is by implementing the full shift model naively in PyTorch, and optimizing it by the cost function in frequency space using the PyTorch automatic gradient framework. This will lead to the shift matrix $\boldsymbol{\tau}$ being estimated iteratively and unconstrained along with the non-negative **W** and **H** factor matrices.

Cross-correlation for $\boldsymbol{\tau}$ optimization

Alternatively to finding the shifts iteratively, we can also estimate the shifts completely by the cross-correlation method from [11], which estimates $\boldsymbol{\tau}$ optimally at every 25th. In this method, no update to **W** or $\boldsymbol{\tau}$ are made using gradients, so PyTorch only estimates **H**. The method works by calculating the residual signal $\widetilde{\mathbf{R}}_{n,f}$ by removing all but one latent component.

$$\widetilde{\mathbf{R}}_{n,f} = \widetilde{\mathbf{V}}_{n,f} - \sum_{d \neq d'} \widetilde{\mathbf{W}}^{(f)}_{n,d} \widetilde{\mathbf{H}}_{d,f}$$

Where $\widetilde{\mathbf{V}}$ is the observed data in the frequency domain and $\widetilde{R}$ is the residual signal when projecting out on all but the d'-th source

Then compute the cross correlation between the residual signal of the latent component

$$\widetilde{\mathbf{c}}_f = \widetilde{\mathbf{R}}^*_{n,f} \widetilde{\mathbf{H}}_{d',f}$$

The shift can then be estimated as t which maximizes the cross correlation

$$t = \arg\max_m \mathbf{c}_m, \quad \boldsymbol{\tau}_{n,d'} = t - (M+1)$$

When using this method, we constrain the cross-correlation such that no value in $\boldsymbol{\tau}$ can exceed ± 1000. This constraint ensures that the shifts will stay closer to 0, such that the peaks can not be shifted far away from their source, while the **H** matrix just accounts for this by constructing the peak at a different location.

When the shiftNMF model has converged, we subtract the mean shift from each column of $\boldsymbol{\tau}$ and adjust the **H** matrix accordingly, such that **H** is placed in the center of the shifts, leading to a more robust mean estimation of the clusters during the hard-modeling phase. The models were trained for 1000 epochs, decided by the loss curve available in the appendix iv.

2.4 Hard modelling of shiftNMF components

Single peak fitting

As the problem is non convex, proper initialisation of the multiplet components is needed. Fitting a random set of multiplets as found in equation 2.1, is not possible. While it might be technically possible to initialize multiplets directly, it is most simply and accurately done from a set of single peaks modelled by the Pseudo Voigt function previously described. Fitting the single peaks is also a non convex problem, for proper fitting a combination of proper initialization and grid search of certain parameters is needed.

First step is identification of peak mean, which can be found using a peak finder. Our model uses SciPy find_peaks function which uses a local maxima peak finder. As later

discussed the number of peaks found is not strictly crucial as long as the desired significant peaks are identified.

Next the width of the peak can be estimated using Full Width Half Maximum at location of the peak mean. From these initial value the W matrix is fitted using gradients.
Lastly, the N weight can be estimated using grid search along with a more accurate FWHM in surrounding area of initial estimation. The implemented grid-search checked in 20x20 variable grid, with values of HMFW from half to double of initial estimation and from all possible N values of 0 to 1.

Following these initial values and fitting of W, the remain parameters and W can be fitted. The parameters where not all simultaneously fitted as this did not seem to perform well. The model alternates between training mean, width and N, and the weight matrix W.

The convergence criteria of this fitting was set to be minimum relative improvement of 1e-7 and minimum improvement of 1e-3. The model trained for a maximum of 1000 iterations.

For the estimation of the pure component spectra x with K identified peak the following reconstruction is fitted using the squared loss. The K peaks being described by 3 vectors containing their Pseudo Voigt profile parameters $\boldsymbol{\mu}, \boldsymbol{\Gamma}, \boldsymbol{N}$, and a vector w their weight.

$$\mathbf{x} \approx \sum_{i=1}^{K} \mathbf{w}_i \cdot PV(\boldsymbol{\mu}_i, \boldsymbol{\Gamma}_i, \mathbf{n}_i)$$

Multiplet hypotheses

From a set single peaks, hypotheses can be created as any combination of peaks. The center of the multiplet can be estimated as the mean of the single peak locations, the FWHM as the mean of the FWHMs, the J-coupling as the mean of the distance between each set of neighboring single peaks, N as the average N of the peaks and lastly the multiplicity is known from the number of peaks. The height of the multiplet is measured from the mean point of the multiplet in the extracted component (plus half the J-coupling in cases of multiplets with order 2), This ensures that the weighting of the multiplets will be consistent for a perfect single multiplet fit. This also allows the regularization to be more consistent across the hypothesis, leading to a more robust fit.

The multiplet hypotheses can be described as follows. Where each vector, $\boldsymbol{\mu}, \boldsymbol{\Gamma}, \mathbf{n}, \mathbf{j}, \mathbf{m}, \mathbf{h}$ describes, mean, FWHM, Gaussian-Lorentz weight, J-coupling, multiplicity and height respectively, approximated from combinations of single peaks as previously explained.

$$\mathbf{C}_i = \mathbf{h}_i \cdot M(\boldsymbol{\mu}_i, \boldsymbol{\Gamma}_i, \mathbf{n}_i, \mathbf{j}_i, \mathbf{m}_i)$$

Some heuristics of a good hypothesis can be used to reduce the overall number of hypotheses. For example, hypotheses consisting of peak with vastly different width can be discarded. Likewise, peak where the J-coupling is not consistent, and so on.

For our experiments we ended up not using any heuristics for hypothesis creation, but did use some heuristic to limit the number of identified single peaks. As this both eliminated time used for single peak fitting and the overall number of multiplet hypotheses. Pruning of single peaks, can be done by setting minimum requirements on relative height and width of the peak. An example of this on the artificial dataset can be found in Appendix iii.

While helpful for computation, pruning of peaks and hypotheses are not strictly needed for the method to work. From a static set of multiplet hypotheses the problem becomes convex and regularization is used to identify the best fitting peaks. Any malformed or badly fitting peaks are therefore automatically discarded during this process.

Regularization and component selection with NLARS

After modelling the multiplets on the shiftNMF components, we can then estimate the true component by applying L1 regularization to the weights, which will increase sparsity in the matrix, leading to elimination of multiplet hypothesis that are comparatively less impactful than others during the hard modelling.

To do this, we perform NLARS (non-negative Least Angel Regression Selection) as described in "Approximate L0 constrained Non-negative Matrix and Tensor Factorization" [20] , which controls the degree of sparsity present in the **W** matrix, corresponding to which multiplets are present in the estimated components.

Algorithm description

The NLARS algorithm updates the coefficient vector β iteratively, such that the objective function is minimized while maintaining non-negativity. The algorithm works as followed.

1. **Initialize and compute correlations**

   $$\mathbf{c} = \mathbf{X}^T(\mathbf{y} - \mathbf{X}\beta)$$

   Where **c** is the correlation vector, **X** is the data matrix, and **y** is the target vector

2. **Identify the variable to enter the active set**

   $$j = argmax(\mathbf{c}_I), \mathbf{c}_j > 0$$

   Add the identified variable j to the active set A and remove it from the inactive set I

3. **Update the coefficients**

   $$\beta_A = \beta_A + \mu(\mathbf{X}^T\mathbf{X})^{-1}_{A,A}\mathbf{1}$$

   where μ is determined by minimizing the objective function along the direction of the update.

4. **Check for 0 coefficients, update set and iterate until convergence**
   If a coefficient β_{A_k} reaches 0, it is removed from the active set A and added back to the inactive set I. The process is repeated until no more variables meet the criterion for the active set, i.e. All correlations c_j for $j \in I$ are non-positive.

The sparse NMF problem is then decomposed into N LASSO problems, solved using the objective function

$$C_{SparseNMF} = \frac{1}{2}||\mathbf{v}_n - \mathbf{W}\mathbf{h}_n||^2_F + \lambda||\mathbf{h}_n||_1$$

Where λ is the regularization parameter and W and v_n are the factor matrices.

Updating the W factor matrix

The factor matrix **W** is updated using normalization invariant multiplicative updates

$$\mathbf{W} \leftarrow \mathbf{W} \cdot \frac{\mathbf{V}\mathbf{H}^T + \mathbf{W} \cdot diag(1(\mathbf{W}\mathbf{H}\mathbf{H}^T \cdot \mathbf{W}))}{\mathbf{W}\mathbf{H}\mathbf{H}^T + \mathbf{W} \cdot diag(1(\mathbf{V}\mathbf{H}^T \cdot \mathbf{W}))}$$

Where $\mathbf{W}_d$ is constrained to have unit L2 norm, ensuring $\mathbf{W}_{n,d} = \frac{\mathbf{W}_{n,d}}{||\mathbf{W}_d||_2}$. The algorithm iteratively updates **W** until convergence, defined by a relative change in λ of less than 10^{-6} or after 100 iterations

Applying NLARS to Hard modelling
When applying NLARS to our hard-modeled components, we focus solely on re-estimating the weights, as the **H** matrix has already been determined from the hard modelling step. The process works as follows:

1. **Extract multiplets**: we extract the multiplets from the shiftNMF estimated **H** and model them into multiplet components **C**. This allows for the NLARS-regularized weights $\mathbf{W}_{NLARS}$ to be estimated.
2. **Freeze C and estimate $\mathbf{W}_{NLARS}$**: We fix the component matrix **C** such that only the component weights $\mathbf{W}_{NLARS}$ are estimated using NLARS. This provides a path of all the weights as they are iteratively zeroed.
3. **Calculate reconstruction error at each interval**: As each component weight is zeroed, calculate the reconstruction error between the fully active component weights $\mathbf{W}_{full}$ and having N active components. Based on a predefined threshold, extract the re-estimated weights $\mathbf{W}_{reg}$ with the fewest active components that still maintain a reconstruction error below the threshold. The threshold should be chosen based on the underlying components, a higher threshold will disable more hard modelled components, and a lower threshold will keep more alive.

Peak interpretation and integration
One of the main advantages of using hard modelling techniques is to ease Direct peak integration to extract information about relative amount of equivalent protons in the associated compound[16]. Since our model is ideally capable of extracting and separating the multiplet structures in a pure component, it should be able to automatically extra this information. Since The multiplets are separated no integration integral is needed as the multiplet component can be integrated directly. Furthermore, extracting the multiplet structure also gives insight into the spin-spin coupling of present non-equivalent protons[21].

2.5 Final decomposition and model pipeline
For decomposing a set of NMR spectres into its multiplets, we have developed the following pipeline, which combines shiftNMF with hard modelling techniques to ensure accurate component extraction and peak identification.

Pipeline Overview
1. **Fitting shiftNMF**
    - Apply shiftNMF to the original data with a determined rank and extract the estimated components **H**
2. **Hard modelling with Pseudo Voigt and Multiplets**
    - Initialize single peaks with peak finder and FWHM.
    - Fit the extracted components **H** from shiftNMF using the Pseudo Voigt function.
    - Construct the singlet and multiplet components **C** based on the J-coupling structure and known relative scales.
3. **Regularization with NLARS**
    - Reconstruct the shiftNMF components **H** using the parameterized hard model components **C** and a trained Weight parameter **W**
    - Apply NLARS to weight matrix **W** with frozen **C** to estimate $\mathbf{W}_{reg}$ containing the most significant singlet and multiplet structures to ensure the reconstructed

hard model components accurately reflects the true signal while deactivating the insignificant components.

3 Data

3.1 Artificial dataset

To construct the artificial dataset, we looked at what a subset of a full-length NMR might look like. Pure component spectra usually consisted of a few different multiplets.

The final artificial dataset was constructed with three different underlying components consisting of various multiplets. Each sample was then constructed as a mixing of the components with a shift mimicking the spectral shifts occurring in NMR spectroscopy. In Fig. 3.1, the underlying components, along with their shifts and mixing, can be seen. In the table. 3.1 contains the parameters of each component's underlying multiplets. As the data was generated artificially, there is no underlying connection to ppm as typically seen on NMR specters; as such, the x-axis of the artificial data and their plots is simply the index of the data points.

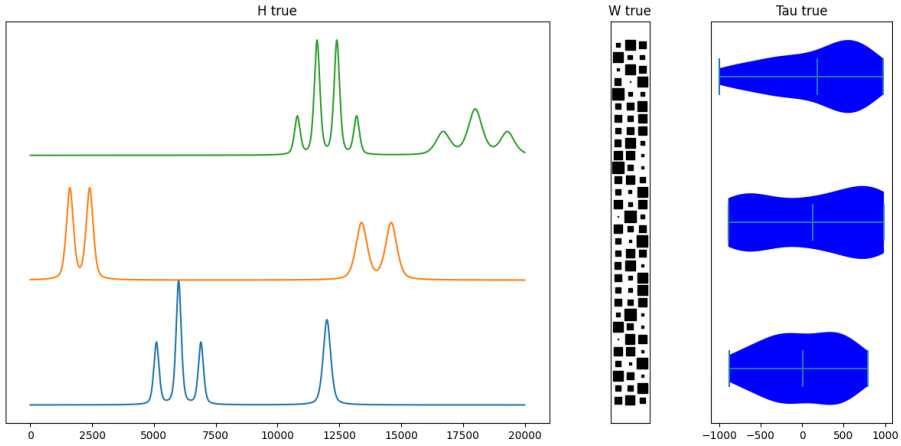
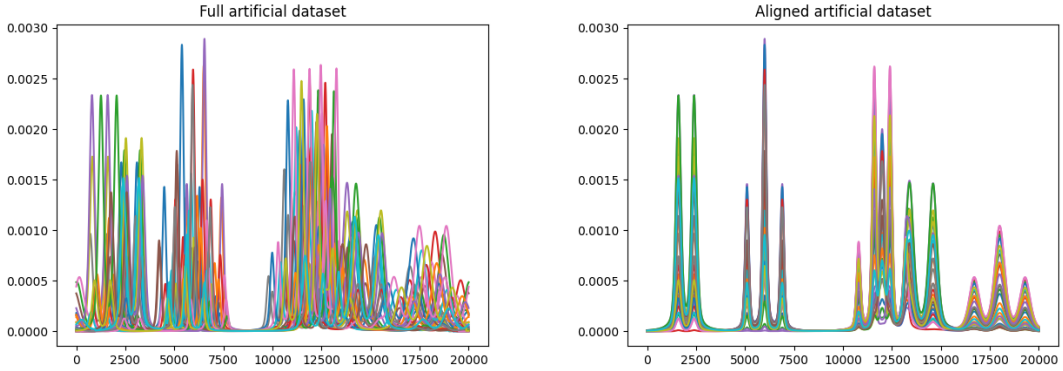


Figure 3.1: Components and their multiplets of the artificial dataset. Blue (Component 1) Orange (Component 2) Green (Component 3)

Table 3.1: Table of component multiplet construction values

H	Multiplicity	Mean	sigma	J-coupling	n
Component 1	3	6000	220	900	0.5
	1	12000	320	0	0.5
Component 2	2	2000	300	800	0.5
	2	14000	480	1200	0.5
Component 3	3	18000	600	1300	0.5
	4	12000	240	800	0.5

Combining the three components into the final dataset gives the following shifted and aligned datasets.



(a) Full unaligned artificial dataset

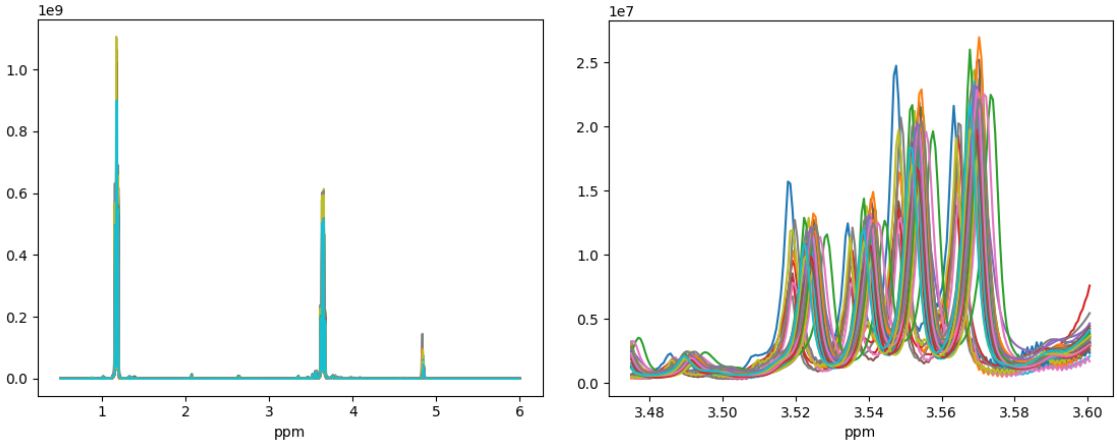
(b) Full aligned artificial dataset

Figure 3.2: The full artificial dataset unaligned (left) and aligned (right)

3.2 Wine dataset

The wine dataset stems from a 1H-NMR analysis of 40 table wines of different origins and colors[22]. The data was intentionally prepared without a buffer solution or pH adjustment and, therefore, contains significant misalignment of the NMR resonance signals.

The measurements were done using 1H-NMR with a frequency of 400.13 MHz. Notably, several peaks labeled in the chemometrics study do not align with expected values recorded in The Human Metabolome Database[23]. Reference values from The Human Metabolome Database, recorded with the same frequency, a D_2O solution, and 1D 1H NMR spectroscopy do not align, but the labeled peaks fall in a similar area. This does, however, mean that they can not be used as a reference of found multiplet positional parameters (except J-coupling), but merely to identify the corresponding compound and general area of its peaks.



(a) Full spectrum 0.5-6 ppm

(b) Zoom on 3.5 - 3.6 ppm. Showing misaligned doublet of doublets indicative of Glycerol

Figure 3.3: Raw (uncorrected) 1 H NMR spectra of 40 table wines

3.3 The Human Metabolome Database (HMDB)

The Human Metabolome Database (HMDB) is a comprehensive electronic database containing details about small molecule metabolites found in the human body[23]. This makes it extremely useful for applications in fields like metabolomics, clinical chemistry, and biomarker discovery. For this project, we will be using their extensive library of NMR specters of chemical compounds to serve as a reference for the parameters in the extracted hard models found in the wine dataset. This information includes the number of peaks present in a cluster, the cluster midpoints and peak centers, their coupling type, and the number of hydrogen atoms present. Furthermore, we will also test the hard modeling techniques on a pure component specter of Lactic Acid, also from the HMDB[24].

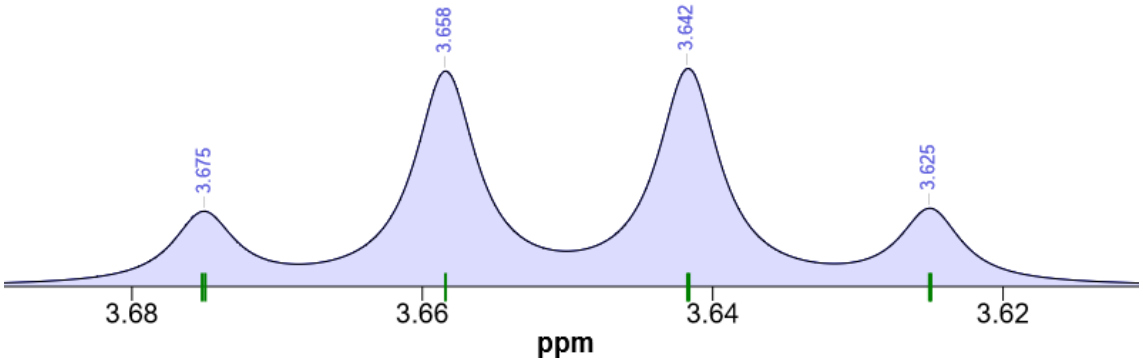


Figure 3.4: Example of ethanol cluster with the midpoint at 3.65 ppm from HMDB

4 Results and Discussion

4.1 Implementing Alternating updates in Python and PyTorch framework

The two implementations of shiftNMF in PyTorch are demonstrated here on the artificial dataset with a known number of components (3). The methods were all run till convergence, and the loss curves of individual runs can be seen in the appendix. iv. The stop criterion for this test was a maximum number of training epochs of 5000.

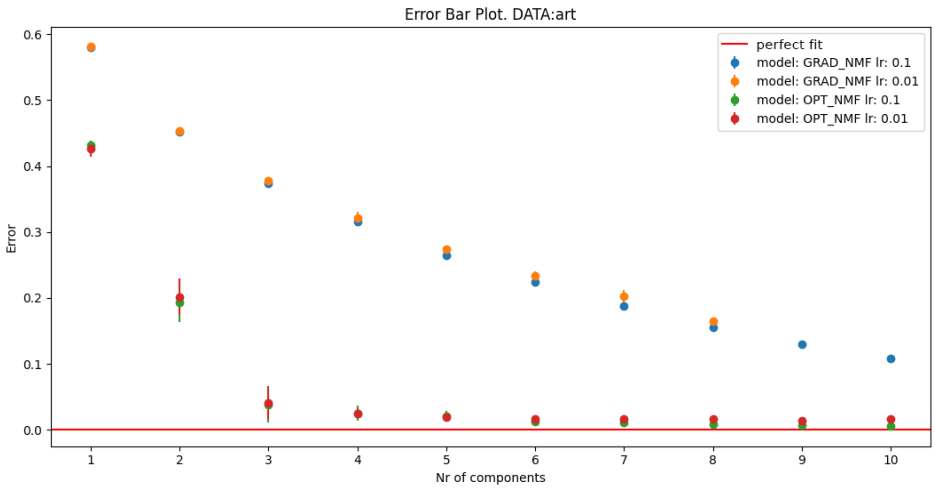
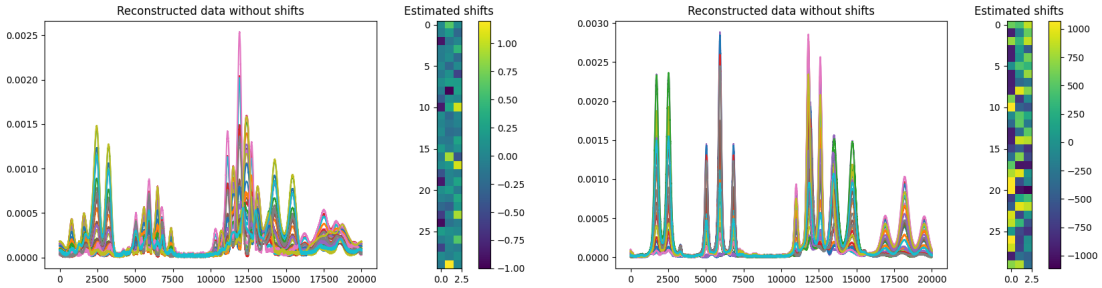


Figure 4.1: Difference between gradient and cross-correlation tau estimation

Comparing the Naive, gradient-based tau optimization with the cross-correlation estimation method, we can observe that the naive implementation underperforms significantly. We can further demonstrate this by plotting the reconstruction of the naive and cross-correlation estimation implementation.



(a) ShiftNMF reconstruction with tau estimation by naive gradients

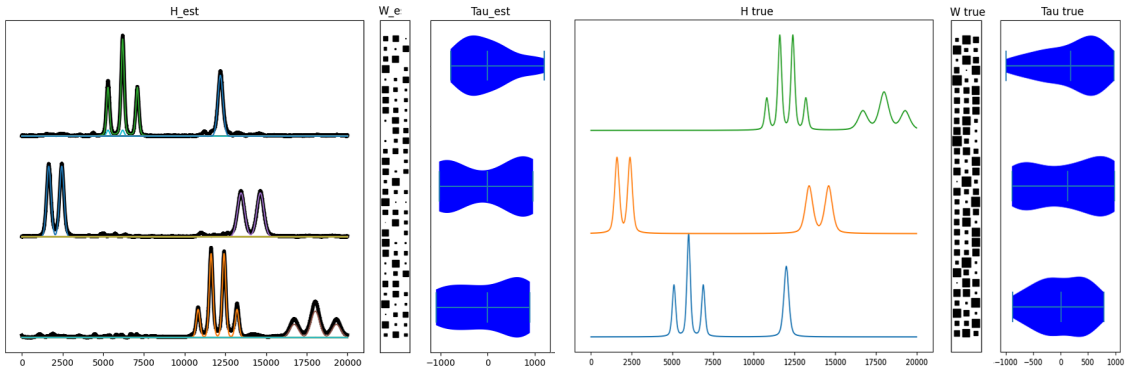
(b) ShiftNMF reconstruction with tau estimations by cross correlation

Figure 4.2: Aligned reconstruction of artificial data for three components using the naive gradient estimation (a) and cross-correlation estimation (b)

As shown in figure 4.2, the naive implementation of shiftNMF aligns the data significantly worse than by utilizing cross-correlation. The cause for this lies in the table of estimated shifts. Where the cross-correlation estimates the shifts in the true range of -1000 to 1000, the shifts estimated by the gradient-based optimization are instead ranging from -1 to 1. This happens due to the sinc-interpolation of the Fourier transform used to shift the components. While the cross-correlation estimates the index of the shifts immediately, the gradients adjust them by a decimal value every time, causing the sinc-interpolation in the Fourier transform to inaccurately reconstruct the data, leading to a suboptimal loss landscape. The discussion section 5.1 gives a more in-depth explanation of this. The result is that the Naive PyTorch implementation of ShiftNMF is sub-optimal for estimating shifts and, therefore, aligning the NMR data.

4.2 Pipeline results on artificial data

The full model pipeline was applied to the artificial data described in section 3.1. The components decomposed by shiftNMF (with cross-correlation updates) can be seen in black, along with the respective weights of samples and shift distributions in Fig. 4.3a. Overlaying colored multiplets in Fig. 4.3a are the final selected multiplets of the pipeline after single peak fitting, multiplet hypotheses generation, and final selection based on regularization. Table. 4.1 shows the model's estimated parameters of the final selected multiplets of each component.



(a) shiftNMF components, mixing, and shifts with hard modeled multiples. Black: underlying component extracted by shiftNMF. Colored signal: weighted multiplets as found by hard modeling. Regularization threshold: 0.35

(b) True artificial components, mixings and shifts

Figure 4.3: Estimation of components and hard modeling of artificial data (a) compared to the ground truth (b)

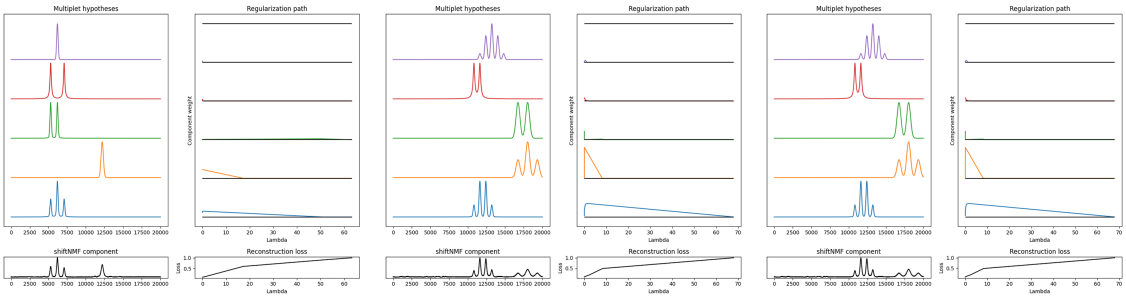
From the estimated components in figure 4.3a, we can see that the estimated components closely match the ground truths of the artificial data in figure 4.3b. Because the components $\mathbf{H}_{est}$ are well estimated with a low reconstruction error, they lend themselves well to the hard modeling phase. As can be seen, the hard-modeled multiplets overlayed on the black components are very close to the ground truths. We can further confirm this by comparing the parameters of the hard modeled multiples with the ground truth parameters.

Table 4.1: Estimated and true parameters of the artificial data, estimated parameters ranked by weight

H	Multiplicity	Mean	FWHM	J-coupling	n
Est. Component 1	3	6197	215	902	0.00
	1	12196	369	0	0.00
	2	1883	902		0.00
True Component 1	3	6000	220	900	0.5
	1	12000	320	0	0.5
Est. Component 2	2	2057	323	797	0.00
	2	14059	515	1185	0.00
True Component 2	2	2000	300	800	0.5
	2	14000	480	1200	0.5
Est. Component 3	3	18025	613	1295	0.00
	4	12025	263	791	0.12
	2	17376	610	1295	0.00
True component 3	3	1800	600	1300	0.5
	4	12000	240	800	0.5

Comparing the estimated parameters with the ground truth from the artificial dataset, we see that the model is very good at estimating the multiplicity of correctly identified peaks and the mean, sigma, and J-coupling. However, the model finds additional peaks in components 1 and 3. The model also fails to estimate n in the Pseudo-Voigt function, estimating all peaks except one as almost pure Lorentzian ($n = 0$). This is despite using grid search to help estimate that parameter. It's not clear where the problem stems from exactly, but we suspect that n has a negligible impact on the fit of the pseudo-Voigt function as long as the FWHM parameter is accurately tuned.

Fig. 4.4 contains the regularization path of the top 5 multiplet hypotheses generated from the single peak fitting of the components. An overview of the regularization path for all hypotheses can be seen in the appendix. v. Looking at the regularization paths, we can see how the model quickly prioritizes reconstruction using the correct components, almost immediately discarding all wrong components. 2 components linger, the green doublet in the first and third components. From their path, we can see that the weights of these components are so low, that the contribution to the reconstruction is minimal, however still enough to not be pruned completely.



(a) Component 1 regularization path

(b) Component 2 regularization path

(c) Component 3 regularization path

Figure 4.4: NLARS Regularization path of the top 5 multiplet components, Regularization path of all hypotheses can be found in appendix v.

4.3 Complete model pipeline on real wine dataset

This section will apply the complete model pipeline to a real wine dataset. This dataset comprises 1H-NMR spectra of 40 table wines of various origins. Unlike the artificial data used earlier, the wine dataset presents real-world challenges that will help assess how effective the model is at estimating the underlying components. All intervals were trained using a 0.1 learning rate and convergence criteria of minimum improvement of 1e-6 and relative improvement of 1e-7. The full spectrum fitting in 4.5 was done with 5000 iterations.

Full spectrum fitting

Initial tests on the wine dataset were done with the full spectrum. Several learning rates were used with different numbers of components, as seen in Fig. 4.5. As seen in the figure, adding additional components beyond the first did not significantly reduce the reconstruction error, uncovering a fundamental problem using the squared loss function in scenarios where the component's magnitudes vary greatly. As seen in Fig. 3.3a, the dataset contains a few peaks that are so large they overshadow smaller features present, as shown in Fig. 3.3b, making it challenging to capture the full complexity of the data accurately.

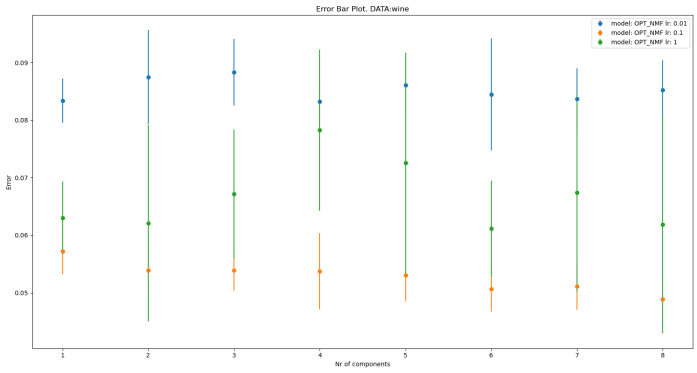


Figure 4.5: Learning rate and component test on the full wine dataset

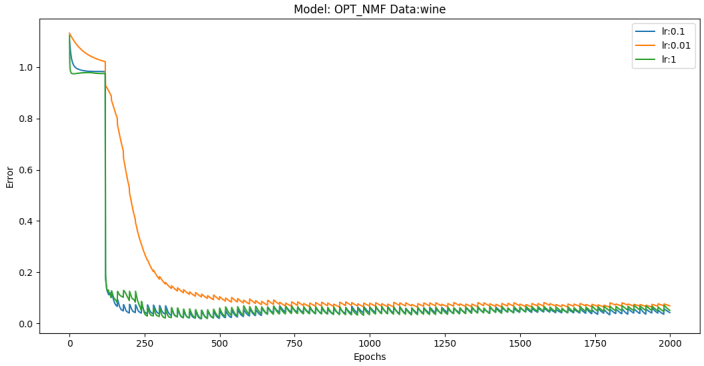


Figure 4.6: Learning curves of shiftNMF with cross-correlation update of Tau at different learning rates

Additionally, in figure 4.5, we can observe that the lowest learning rate does not produce the lowest loss. This happens towards model convergence, where the tau updates lead

to a higher loss, and the lower learning rate makes the model fail to adjust before the next update. This effect is visible in fig.4.6, where the loss curve becomes jagged as it reaches convergence. We suspect this happens because the weights are updated after τ is fully fitted, meaning as changes are made during the τ updates, the new weights **W** are not estimated immediately, leading to residuals containing signals that otherwise should be removed. A more stable implementation would update the corresponding weight after each individual shift is estimated. The original ShiftNMF paper[11] includes ideal direct optimization of W while updating Tau. Correctly implementing this update to W during the Tau update would likely solve the unstable behavior of the current shiftNMF implementation. It remains unclear whether the correct implementation of weight updates will solve the problem of correctly separating components of significant size differences. This is further explored in the more interpretable 3.5-3.7 ppm interval or when the components are more similar in size, like in the 1.3-1.6 ppm interval.

Interval Fitting

By instead focusing on the intervals presented in the paper, we can more effectively utilize shiftNMF to improve the accuracy of our component separation and identification. This targeted approach allows us to overcome some limitations encountered when analyzing the entire spectrum, ultimately leading to more robust results.

3.5-3.7 ppm interval

We will look at training in the 3.5-3.7 ppm interval to investigate this further. According to the chemometric study, the dataset originates from the interval contains a quartet pertaining to Glycerol (3.6-3.7 ppm) and a set of 2 doublets pertaining to Ethanol (3.5-3.6 ppm)[22]. The raw data can be seen in Fig. 4.7. As seen in the figure, the relative height of the doublets is almost insignificant compared to that of the quartet. In this interval, we have two present compounds (Ethanol and Glycerol), and in theory, shiftNMF could separate these, but as seen in Fig. 4.8, this is not the case. While the components capture both the quartet and set of doublets, it doesn't manage to separate them. Seemingly, the first component (orange) captures some of the noise surrounding the second component (blue), fig. 4.8.

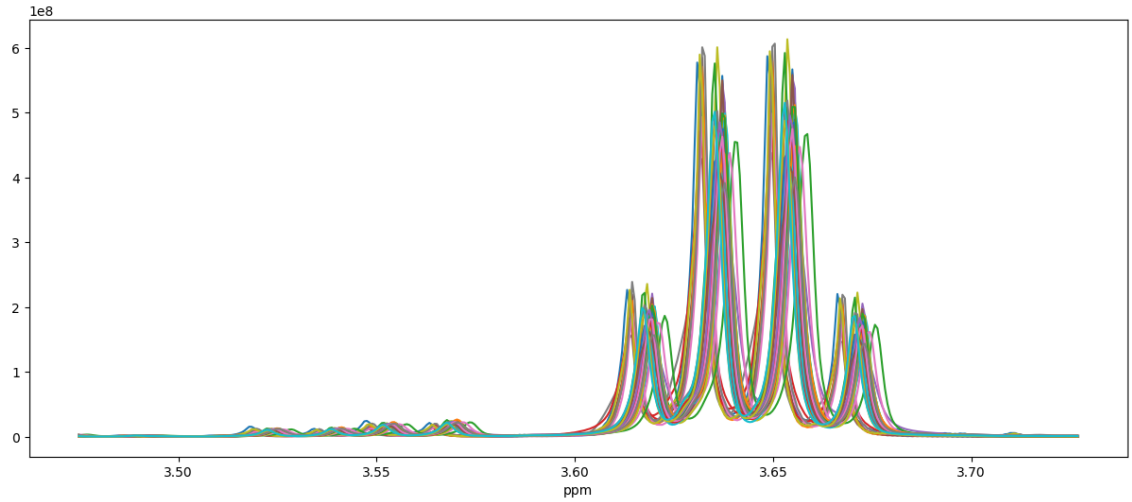


Figure 4.7: 3.5-3.7 ppm of the wine dataset

Fig. 4.8 seemingly captures the quartet in component 2, along with a noisy part of the two doublets found in 3.5-3.6 ppm.

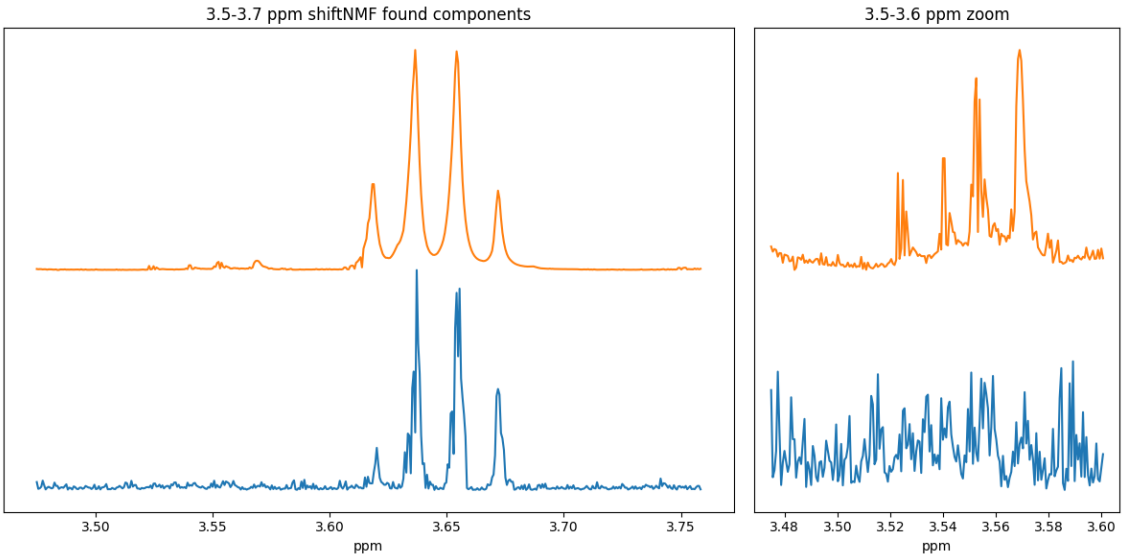
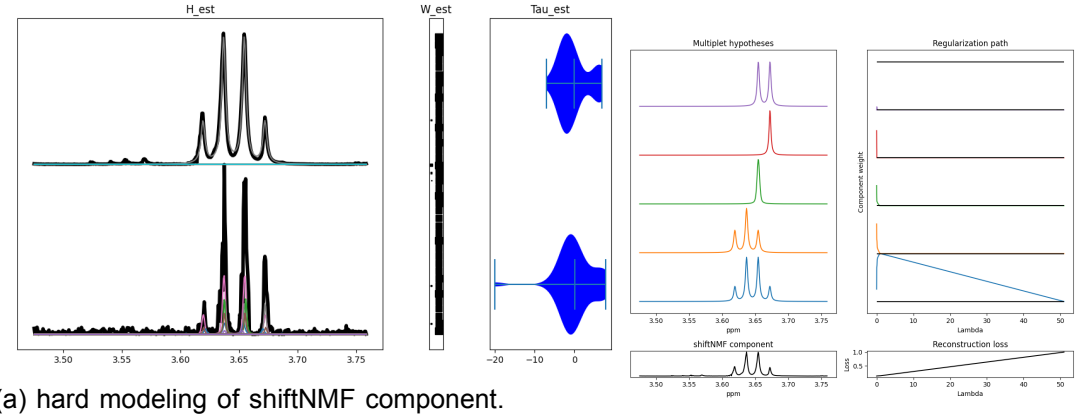


Figure 4.8: 2 component shiftNMF fit on data in 3.5-3.7 ppm interval



(a) hard modeling of shiftNMF component. Black: shiftNMF component. Colored multiplets: hard model components, threshold: 0.25

(b) Regularization path of the top 5 hard model hypotheses for the second shiftNMF component

Figure 4.9: Results from hard modeling of 3.5-3.7 ppm interval

Looking at the hardmodelling process, fig. 4.11a, the first component is not modeled cleanly because it is not pure. It attempts to account for the significant peaks instead of allowing the second component to adjust for these variations. On the other hand, the second component is more refined. This makes the hard modeling process less ambiguous and more straightforward, capturing the quartet more cleanly, as seen on the regularization of the found multiplets in 4.11b.

Table 4.2: Estimated and true parameters on interval 3.5-3.7 ppm

H	Multiplicity	Mean (ppm)	J-coupling (ppm)	n
Est. Component 2 (Ethanol)	4	3.645	0.018	0.99
Ethanol	4	3.64	0.017	N/A

Looking at the results from the estimated and ground truth parameters from HMDB, the found multiplicity is perfectly estimated, while the J-coupling varies by 0.001. However, it neither captures the second pair of doublets nor separates the pure components.

3.5-3.6 ppm Interval

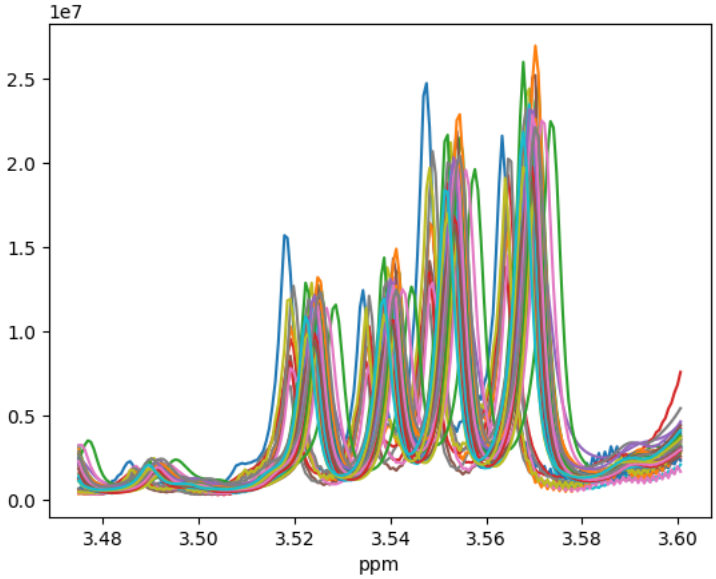
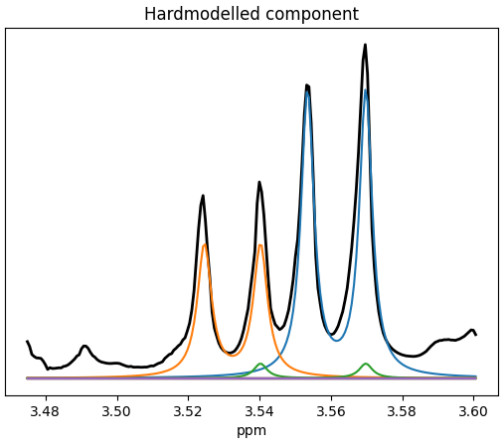
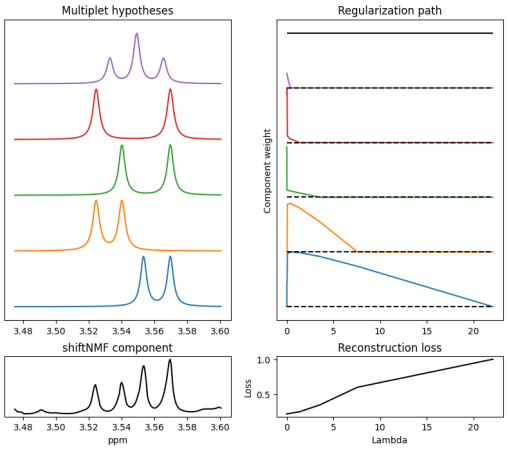


Figure 4.10: 3.5 - 3.6 ppm of the wine dataset

In the 3.5-3.6 ppm interval seen in figure 4.10, we have a set of 2 doublets corresponding to that of Glycerol. In addition to the shifted data, the roofing phenomenon is also present in this interval.



(a) hard modeling of shiftNMF component. Black: shiftNMF component. Colored multiplets: hard model components, threshold: 0.15



(b) Regularization path of the top 5 hard model hypotheses for the shiftNMF component. The regularization path of all hypotheses can be found in appendix v.

Figure 4.11: Results from hard modeling of 3.5-3.6 ppm interval

Applying the model to the interval, we see that in addition to fitting the two doublets, it also fits a third more spread out doublet that helps account for the roofing for the doublet

of doublets. As such, the model still considers it important for reconstruction. In the regularization path, we can see that the two doublets making up the doublet of doublets are active the longest, with the green doublet also staying active.

Table 4.3: Estimated and true parameters on interval 3.5-3.6 ppm, estimated parameters ranked by weight

H	Multiplicity	Mean (ppm)	J-coupling (ppm)	n
Est. Glycerol	2	3.561	0.016	0.98
	2	3.532	0.016	1.00
	2	3.555	0.029	1.00
Glycerol	2	3.626	0.013	N/A
	2	3.600	0.013	N/A

The hard modeling fits the multiplicity, but the J-coupling deviates from the expected 0.013 to the estimated 0.016. Additionally, the hard modeling finds a third doublet, which helps adjust the reconstruction for the roofing phenomenon. As such, this doublet is not present in the ground truth data, and the J-coupled spacing is larger than the others. In total, 15 hypotheses were generated for this, from which NLARS removed 12. The complete hypothesis set and regularization paths are available in the appendix.

1.3-1.6 ppm Interval

Looking at the 1.3-1.6 ppm interval where multiple compounds are present but at more similar concentrations, the results significantly improve; however, other problems are presented. The interval, as seen in Fig. 4.12, contains a doublet pertaining to Lactic Acid and a triplet related to Ethanol.

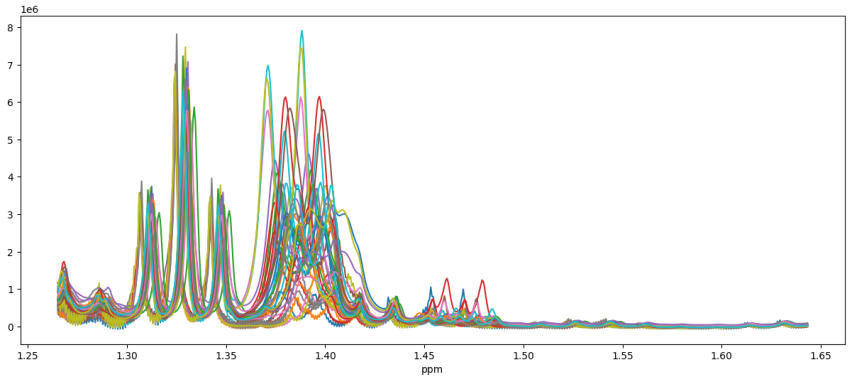


Figure 4.12: Raw wine data at 1.3-1.6 ppm

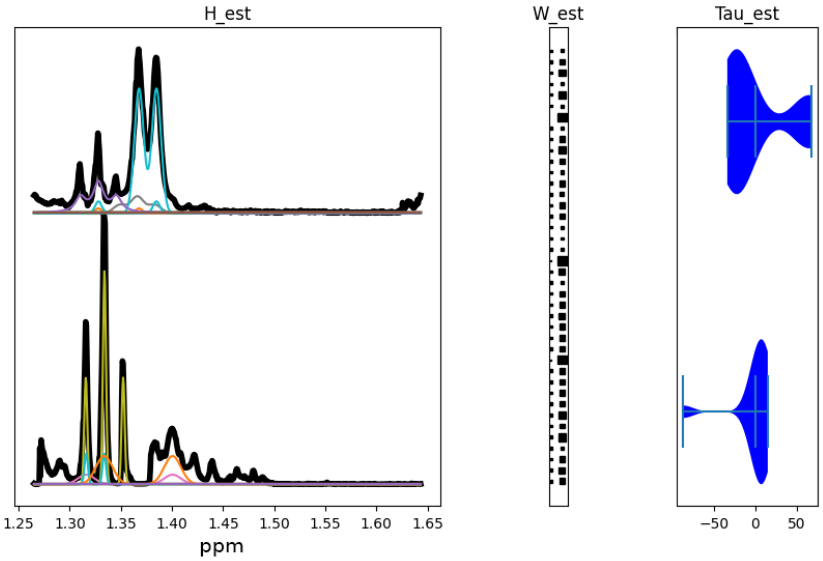
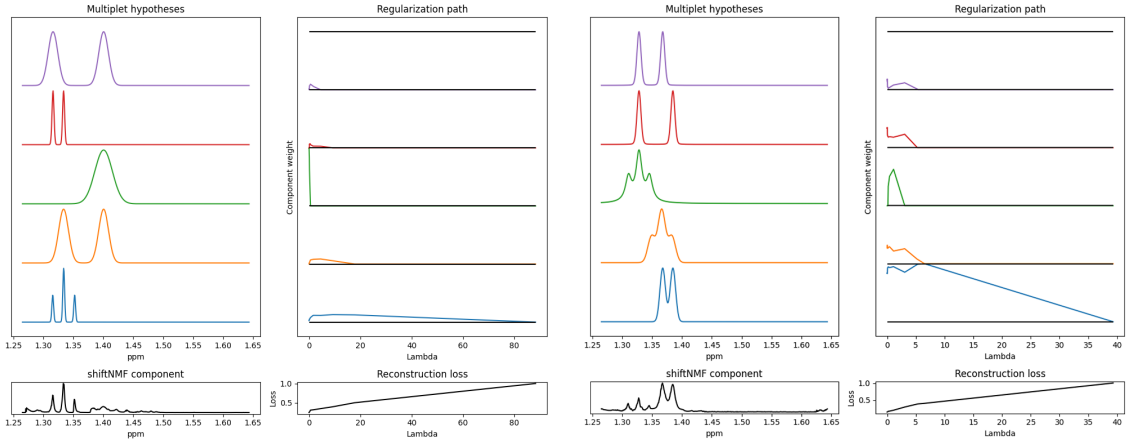


Figure 4.13: Estimated components (black), with weights and shifts. Underlined are estimated hard model multiplets of wine data at 1.3-1.6 ppm, threshold: 0.25

While the extracted doublet component pertaining to Lactic acid and triplet pertaining to Ethanol are extracted, both components contain a muddy region at the points where the peaks overlap, causing the areas not to be separated. As can be seen by the hard-modeled multiplets in the overlapping areas, this results in a segment containing incorrectly modeled multiplets, leading to a more ambiguous result. We can further explore this by looking at the regularization path.



(a) Component 1 path.
Triplet correctly identified.

(b) Component 2 path.
Doublet correctly identified

Figure 4.14: Top 5 hard model components regularization path for each shiftNMF components. The regularization path of all hypotheses can be found in appendix v.

While the model quite nicely finds both the triplet and the doublet, the extracted components are impure, with some of the doublet component bleeding into the triplet and vice versa. We see multiplet structures with peaks in the messy regions that are still activated longer by NLARS. Despite this, the extracted triplet and doublet are still deemed the most

significant part of the reconstruction, as the multiplet is active the longest. We can still see how this result affects the interpretability when looking at the parameterized values from the hard modeling.

Table 4.4: Estimated and true parameters on interval 1.3-1.6 ppm, estimated parameters ranked by weight

H	Multiplicity	Mean (ppm)	J-coupling (ppm)	n
Est. Lactic Acid	2	1.376	0.017	0.00
	3	1.328	0.018	1.00
	3	1.366	0.018	0.02
	2	1.356	0.057	0.14
	2	1.348	0.039	0.12
Lactic Acid	2	1.320	0.016	N/A
Est. Ethanol	3	1.334	0.018	0.01
	2	1.358	0.085	0.00
	2	1.324	0.018	0.00
	2	1.367	0.067	0.00
Ethanol	3	1.170	0.017	N/A

While the Lactic Acid is fitted very well and achieves a similar J-coupling and multiplicity as the ground truth from HMDB, the Ethanol is fitted quite poorly due to the impurity of the found components, which results in the creation of additional wrongly placed and spaced multiplets. At the same time, the underlying triplet is a near-perfect fit compared to the expected J-coupling, doublets and singlets are fitted in the impure area with a higher J-coupling, resulting in an ambiguous result. The complete hypothesis list can be found in appendix v.

4.4 Applying hard modeling to extract additional insight into chemical compound structure

To test the viability of automating the interpretation methods by hard modeling, as described in section 2.4, we also applied the hard model pipeline to a pure component spectra of Lactic Acid acquired from the Human Metabolome Database[24]. Looking at Fig. 4.15. it fits and extracts the appropriate multiplets perfectly. Likewise, in tab. 4.5 the model finds the reference values from HMDB perfectly as well.

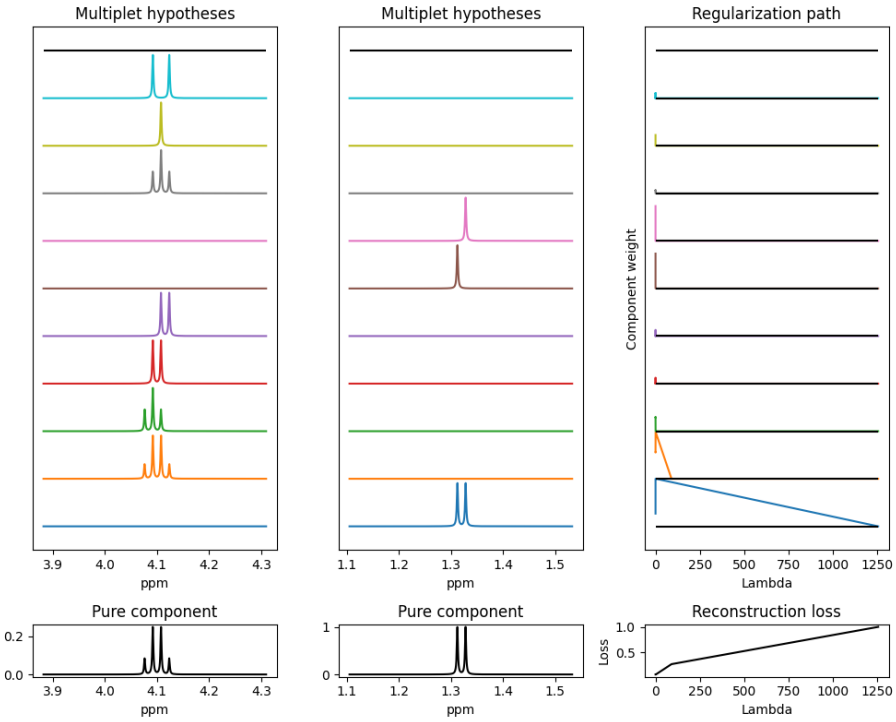


Figure 4.15: Hardmodelling regularization path for pure Lactic Acid spectrum

Estimated values	**Multiplicity**	**Mean**	**sigma**	**J-coupling**	**n**
Component 1	2	1.31997 ppm	12.19 (index)	0.0158 ppm	1.00
Component 2	4	4.1001 ppm	11.73 (index)	0.0158 ppm	1.00
Reference values	**Multiplicity**	**Mean**	**sigma**	**J-coupling**	**n**
Component 1	2	1.32 ppm	N/A	0.016	N/A
Component 2	4	4.10 ppm	N/A	0.016	N/A

Table 4.5: Estimated values using hardmodel approach. Ground truth reference value of Lactic Acid from Human Metabolome Database(HMDB)[24]

Found values	**Multiplicity**	**Integral**	**Ratio**
Component 1	2	0.01512	3
Component 2	4	0.00504	1

Table 4.6: Integral values of components and their ratio

In table 4.6 is the integrated values of the found multiplets. From these values, we can determine that the molecule has two sets of equivalent hydrogen nuclei that have a ratio of 3 to 1. Since there are only two sets of equivalent nuclei, they must be coupled to each other, inferencing from the multiplicity that there are three hydrogen of one unique hydrogen nuclei and 1 of another unique hydrogen. All in accordance with the structure of Lactic Acid, as seen in Fig. 4.16. Corresponding to multiplet with three equivalent hydrogen nuclei on CH_3 group and one unique hydrogen on the middle carbon, CH. This shows how, given a pure constituent component, our multiplet hard model approach can automatically extract additional insightful information about present compounds.

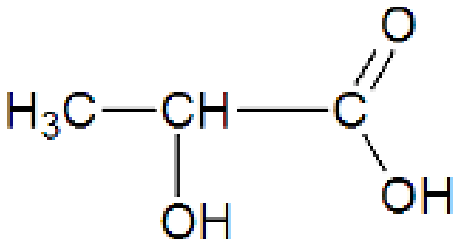


Figure 4.16: Lactic acid structure

5 Further discussion

5.1 Failure of the naive shiftNMF PyTorch implementation

Implementing shiftNMF in the PyTorch framework is technically possible using pure gradient optimization. This method, however, is insufficient since the estimated τ parameter can take on any real value, leading to an inaccurate reconstruction due to the sinc-interpolation.

In the context of shiftNMF, the sinc-interpolation is used in the Fourier transform to shift values in the time domain by τ. The sinc function is defined as:

$$sinc(x) = \frac{sin(\pi x)}{\pi x}$$

And the relationship of two continuous-time signals $x(t)$ and $y(t)$ where y is a time-shifted from x is:

$$y(t) = x(t - \Delta)$$
$$Y(f) = e^{-i2\pi f\Delta}X(f)$$

As the change in Δ assumes non-integer values, x is sampled using sinc interpolation.

$$x(t) = \sum_{n=-\infty}^{\infty} x[n] \cdot \text{sinc}(\pi(\frac{t}{T} - n))$$

T is the sampling interval, n is the sample index and $x[n]$ is the value at the n-th sample.

As t takes on non-integer values, the sinc-interpolation can cause artifacts in the reconstruction, like negative values, which violate the non-negativity. In addition, the sinc-interpolation causes the gradients calculated to be unrepresentative of the desired shift, causing the pure gradient implementation to be stuck in a local minima.

5.2 Relative component size

When applying the shiftNMF model to the full spectrum, we encountered issues with the fitment due to the Frobenius norm loss. The model disproportionately emphasizes a few peaks with high magnetic responses because the Frobenius norm squares the error. Consequently, instead of distributing multiplets distinctly across the shiftNMF components, the model tends to construct the same multiplet multiple times but with different shifts rather than adjusting the τ value. This approach leads to many smaller peaks being poorly modeled, rendering them inadequate for hard modeling and not properly separating compounds to different components.

5.3 Regularization robustness and NLARS

The NLARS regularization is fairly efficient in pruning unimportant multiplets from the hard modeling, as a majority of the hypotheses are inactivated. However, we observe some difficulties when the underlying components have roofing present as on 4.11a. Here, the model will still keep some activation of hard-modeled components that help account for the effect. We also observe some difficulties in the 1.3-1.6 ppm of the raw wine dataset, where the doublet and triplet of the extracted NMF components bleed into each other

instead of remaining pure. This causes the hard modeling step to construct multiplets in the muddy region, as seen in figure 4.14, negatively affecting the interpretation.

However, the regularization paths generated by NLARS help this interpretability, as you can see which components are disabled in which order, thereby ranking the importance of each multiplet as determined by regularization as the strength increases.

5.4 Future improvements

Modeling roofing

The model does not account for any roofing; because of this, the model will construct additional multiplets that help account for the roofing as seen in figure 4.11a. By allowing the model to account for the roofing phenomenon, situations like glycerol could, in theory, be modeled less ambiguously, resulting in a better reconstruction with a smaller number of active components, which additionally should make the results easier to interpret, more accurately extract underlying parameters and allow for more accurate peak integration.

Creating more complex multiplet structures

The current hard model only generates multiplets using simple multiplet structures, as seen in red in Fig. 5.1. However, more complex structures like triplets of doublets exist, as seen below.

Further identification of complex multiplet structures could include initial fitting and selection of simple multiplet structures before hypothesis generation using the same multiplicity structures. This is viable because the more complex multiplet structures consist of the simpler multiplet structures.

The model could iteratively fit more complex structures from existing found structures until simpler structures are preferred or no more viable complex structures can be generated.

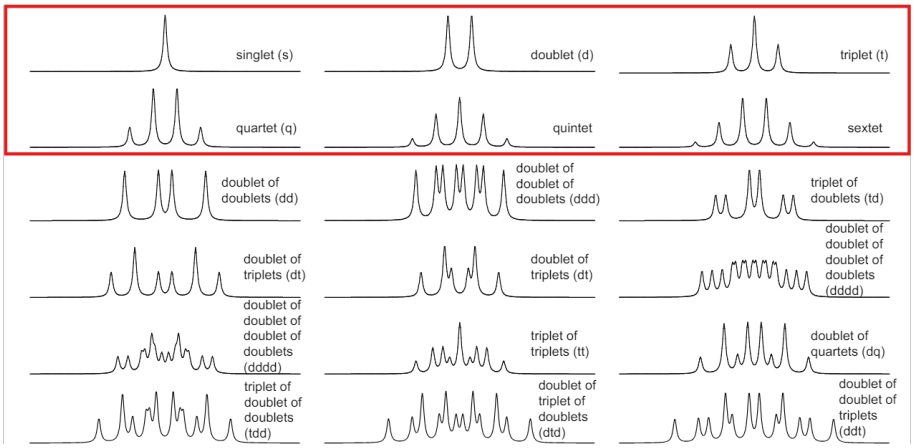


Figure 5.1: Red: Simple multiplet structures. Below: complex multiplet structure. [25]

Fixing unstable behavior of shiftNMF

As previously mentioned, implementing the correct update of τ in shiftNMF as seen in ”Shifted Non-Negative Matrix Factorization”[11], should alleviate the unstable behavior of the current implementation. The current weight update was implemented as an afterthought, as the proposed method after translation to Python did not work as intended. Instead, the current weights are re-estimated after τ has been fully fit.

Incorporating extracted multiplets in a joint model

While the current model is capable of many of the perks associated with both hard modeling and soft modeling, it still lacks the modeling of non-linear effects, like broadening, that the original Indirect Hard Modelling has. The model extracts the multiplets components from the shiftNMF component, which could be seen as an average of the peak widths after broadening. These multiplets should serve as a good starting baseline for a direct joint model that would model the data directly using multiplet components, weights, and shifts. The model would already know initial shifts and weights from the initial shiftNMF run and be able to learn the peak variation on top of the ”baseline” selected multiplets. This would obtain the non-linear advantages of hard modeling on top of multiplet parameter detection.

6 Conclusion

In this thesis, we demonstrate the potential and effectiveness of parametric shift-invariant modeling for analyzing Nuclear Magnetic Resonance data. By combining hard modeling techniques with shifted Non-negative Matrix Factorization, we developed a framework capable of extracting detailed insights into the chemical structures present in complex datasets.

We have demonstrated that a naive PyTorch implementation of shiftNMF is insufficient since the non-integer τ updates cause issues due to sinc-interpolation, leading to poor gradients resulting in the model being stuck, with the τ values being unable to move outside of a $\pm$ 1 range. Instead of a naive implementation, cross-correlation in the latent variables can be used to estimate the shifts more efficiently, leading to better model convergence.

The results obtained from both synthetic and real datasets, specifically the wine dataset, demonstrated our model pipeline's effectiveness while highlighting some shortcomings. The shiftNMF approach successfully identified underlying components in the presence of spectral shifts, while hard modeling provided a more interpretable representation of the NMR spectra. In cases of compounds being close together or artifacts like roofing, the results of our model became more ambiguous and, therefore, more challenging to interpret directly.

One significant finding of this study is the improved accuracy and interpretability achieved by integrating domain knowledge into the shiftNMF formulation. By using peak-finding algorithms to establish initial conditions of singlets and L1 regularization to identify correct significant multiplets, we were able to identify the parameters multiplicity, cluster mean, and J-coupling in present multiplets. Hard modeling of the peaks also allowed the extraction of additional pure component information relating to compound structure.

Future work could focus on refining the model to handle more complex multiplet structures, improving shiftNMF stability, and implementing a final joint model to account for non-linear effects, like roofing and peak broadening, that prove challenging for the current model.

In conclusion, the parametric shift-invariant modeling approach developed in this project represents a step towards a fully joint model capable of finding pure spectra, modeling non-linear effects, finding relative concentrations, and extracting and describing underlying multiplet structures and parameters in NMR specters. While the final non-linear effects were not modeled, the model still offers deeper insights into the molecular composition of complex chemical mixtures.

Bibliography

[1] Abdul Hamid Emwas et al. “NMR Spectroscopy for Metabolomics Research”. In: *Metabolites* 9.7 (July 2019). ISSN: 22181989. DOI: 10.3390/METABO9070123. URL: /pmc/articles/PMC6680826/%20/pmc/articles/PMC6680826/?report=abstract%20https://www.ncbi.nlm.nih.gov/pmc/articles/PMC6680826/.

[2] Max T. Rogers. *NMR Artifacts - Michigan State University*. URL: https://nmr.natsci.msu.edu/experiments/nmr-artifacts.aspx.

[3] F. Savorani, G. Tomasi, and S. B. Engelsen. “icoshift: A versatile tool for the rapid alignment of 1D NMR spectra”. In: *Journal of Magnetic Resonance* 202.2 (Feb. 2010), pp. 190–202. ISSN: 1090-7807. DOI: 10.1016/J.JMR.2009.11.012.

[4] Gareth A Morris. *NMR Data Processing*. URL: https://www.nmr.chemistry.manchester.ac.uk/sites/default/files/NMR%20data%20processing.pdf.

[5] peaxact. *How does NMR Spectroscopy Benefit from Spectral Hard Modeling? | PEAXACT Blog*. URL: https://blog.peaxact.com/2015/06/how-does-NMR-spectroscopy-benefit-from-IHM.html.

[6] Suhas Tikole et al. “Peak picking NMR spectral data using non-negative matrix factorization”. In: *BMC Bioinformatics* 15.1 (Feb. 2014), pp. 1–7. ISSN: 14712105. DOI: 10.1186/1471-2105-15-46/FIGURES/4. URL: https://bmcbioinformatics.biomedcentral.com/articles/10.1186/1471-2105-15-46.

[7] Guro F. Giskeødegård et al. “Alignment of high resolution magic angle spinning magnetic resonance spectra using warping methods”. In: *Analytica Chimica Acta* 683.1 (Dec. 2010), pp. 1–11. ISSN: 0003-2670. DOI: 10.1016/J.ACA.2010.09.026.

[8] Parvaneh Ebrahimi et al. “Chemometric Analysis of NMR Spectra”. In: *Modern Magnetic Resonance* (2017), pp. 1–20. DOI: 10.1007/978-3-319-28275-6{_}20-1. URL: https://www.researchgate.net/publication/318156361_Chemometric_Analysis_of_NMR_Spectra.

[9] Abdul-Hamid Emwas et al. “Recommended strategies for spectral processing and post-processing of 1D 1 H-NMR data of biofluids with a particular focus on urine”. In: *Metabolomics* 14 (123), p. 31. DOI: 10.1007/s11306-018-1321-4. URL: https://doi.org/10.1007/s11306-018-1321-4.

[10] Anna De Juan, Joaquim Jaumot, and Romà Tauler. “Multivariate Curve Resolution (MCR). Solving the mixture analysis problem”. In: *Analytical Methods* 6.14 (July 2014), pp. 4964–4976. ISSN: 17599679. DOI: 10.1039/C4AY00571F.

[11] Morten Mørup, Kristoffer Hougaard Madsen, and Lars Kai Hansen. “Shifted Non-negative Matrix Factorization”. In: *IEEE International Workshop on MACHINE LEARNING FOR SIGNAL PROCESSING* (2007), pp. 139–144. DOI: 10.1109/MLSP.2007.4414296. URL: https://doi.org/10.1109/MLSP.2007.4414296.

[12] *How does Spectral Hard Modeling work? | PEAXACT Blog*. URL: https://blog.peaxact.com/2011/12/how-does-indirect-hard-modeling-work.html.

[13] Frank Alsmeyer, Hans Jürgen Koß, and Wolfgang Marquardt. “Indirect spectral hard modeling for the analysis of reactive and interacting mixtures”. In: *Applied spectroscopy* 58.8 (Aug. 2004), pp. 975–985. ISSN: 0003-7028. DOI: 10.1366/0003702041655368. URL: https://pubmed.ncbi.nlm.nih.gov/18070391/.

[14] E. Kriesten et al. “Identification of unknown pure component spectra by indirect hard modeling”. In: *Chemometrics and Intelligent Laboratory Systems* 93.2 (Oct. 2008), pp. 108–119. ISSN: 0169-7439. DOI: 10.1016/J.CHEMOLAB.2008.05.002.

[15] E. Kriesten et al. “Fully automated indirect hard modeling of mixture spectra”. In: *Chemometrics and Intelligent Laboratory Systems* 91.2 (Apr. 2008), pp. 181–193. ISSN: 0169-7439. DOI: 10.1016/J.CHEMOLAB.2007.11.004.

[16] Steven Farmer, Dietmar Kennepohl, and Tim Soderberg. *13.4: Integration of ¹H NMR Absorptions - Proton Counting - Chemistry LibreTexts*. URL: https://chem.libretexts.org/Bookshelves/Organic_Chemistry/Organic_Chemistry_%28Morsch_et_al.%29/13%3A_Structure_Determination_-_Nuclear_Magnetic_Resonance_Spectroscopy/13.04%3A_Integration_of_H_NMR_Absorptions-_Proton_Counting.

[17] O. Arnold et al. “Mantid - Data analysis and visualization package for neutron scattering and μ SR experiments”. In: *Nuclear Instruments and Methods in Physics Research, Section A: Accelerators, Spectrometers, Detectors and Associated Equipment* 764 (Nov. 2014), pp. 156–166. ISSN: 01689002. DOI: 10.1016/J.NIMA.2014.07.029.

[18] Daniel D. Lee and H. Sebastian Seung. “Learning the parts of objects by non-negative matrix factorization”. In: *Nature 1999 401:6755* 401.6755 (Oct. 1999), pp. 788–791. ISSN: 1476-4687. DOI: 10.1038/44565. URL: https://www.nature.com/articles/44565.

[19] Adam Paszke et al. “PyTorch: An Imperative Style, High-Performance Deep Learning Library”. In: (2019).

[20] M ; Mørup, K H Madsen, and L K Hansen. “Approximate L0 constrained Non-negative Matrix and Tensor Factorization”. In: *Citation* (2008). DOI: 10.1109/ISCAS.2008.4541671. URL: https://doi.org/10.1109/ISCAS.2008.4541671.

[21] Chris Schaller et al. *13.5: Spin-Spin Splitting in ¹H NMR Spectra - Chemistry LibreTexts*. URL: https://chem.libretexts.org/Bookshelves/Organic_Chemistry/Organic_Chemistry_(Morsch_et_al.)/13%3A_Structure_Determination_-_Nuclear_Magnetic_Resonance_Spectroscopy/13.05%3A_Spin-Spin_Splitting_in_H_NMR__Spectra.

[22] Flemming H Larsen, Frans Van Den Berg, and Søren B Engelsen. “An exploratory chemometric study of 1 H NMR spectra of table wines”. In: *JOURNAL OF CHEMOMETRICS* 20.5 (2006), pp. 198–208. DOI: 10.1002/cem.991. URL: www.interscience.wiley.com.

[23] *Human Metabolome Database*. URL: https://hmdb.ca/.

[24] *Human Metabolome Database: 1H NMR Spectrum (1D, 400 MHz, D2O, predicted) (HMDB0000190)*. URL: https://hmdb.ca/spectra/nmr_one_d/6078.

[25] J Nowick. “Multiplet Guide and Workbook”. In: ().

A Appendix

i Source Code

Link to GitHub repository: https://github.com/LucasEmcken/Bachelorprojekt

ii Parseval's Theorem DFT

From the DFT we have

$$\mathbf{X}(k) = \sum_{n=0}^{N-1} \mathbf{x}(n) e^{-i2\pi \frac{nk}{N}} \tag{A.1}$$

Squaring the frequency domain is then equivalent to multiplying the time domain with its complex conjugate

$$|\mathbf{X}(k)|^2 = \sum_{n=0}^{N-1} \mathbf{x}(n) e^{i2\pi \frac{nk}{N}} \sum_{n'=0}^{N-1} \mathbf{x}(n) e^{-i2\pi \frac{nk}{N}} \tag{A.2}$$

$$= \sum_{n=0}^{N-1} \mathbf{x}(n) \sum_{n'=0}^{N-1} \mathbf{x}^*(n') e^{i2\pi \frac{(n'-n)k}{N}} \tag{A.3}$$

Summing over the frequencies then gives:

$$\sum_{k=0}^{N-1} |\mathbf{X}(k)|^2 = \sum_{k=0}^{N-1} \sum_{n=0}^{N-1} \mathbf{x}(n) \sum_{n'=0}^{N-1} \mathbf{x}^*(n) e^{i2\pi \frac{(n'-n)k}{N}} \tag{A.4}$$

$$= \sum_{k=0}^{N-1} \mathbf{x}(n) \sum_{n=0}^{N-1} \mathbf{x}^*(n) \sum_{n'=0}^{N-1} e^{i2\pi \frac{(n'-n)k}{N}} \tag{A.5}$$

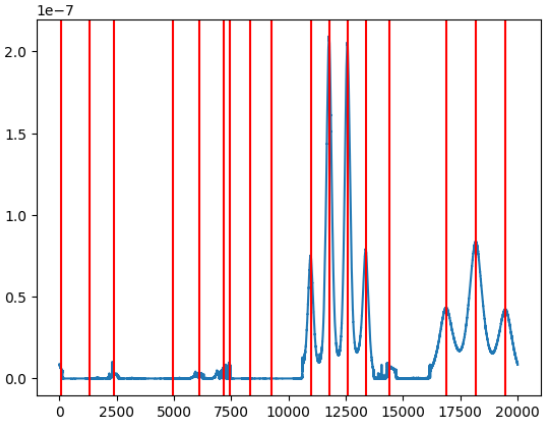
The innermost sum is a geometric sum which can be written as

$$\sum_{n'=0}^{N-1} e^{i2\pi \frac{(n'-n)k}{N}} = \frac{e^{i2\pi(n'-n)k} - 1}{e^{i2\pi \frac{(n'-n)k}{N}} - 1} \tag{A.6}$$

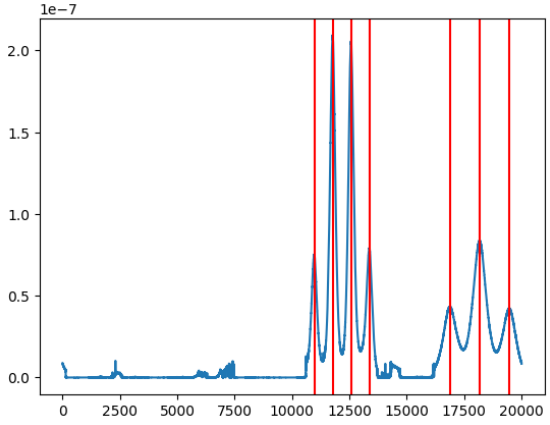
Which is zero unless $n = n'$. Since this makes all the exponential terms cancel and Parseval's theorem follows

$$\sum_{k=0}^{N-1} |\mathbf{X}(k)|^2 = N \sum_{n=0}^{N-1} |\mathbf{x}(n)|^2 \tag{A.7}$$

iii Filtering initialization



(a) Initial peaks found using local maxima peak finder



(b) peak after filtering with minimum height and width requirement. Peak location estimated with scipy find peaks and width using FWHM.

Figure A.1: Comparing peaks before filtering on the first shiftNMF component of the artificial dataset

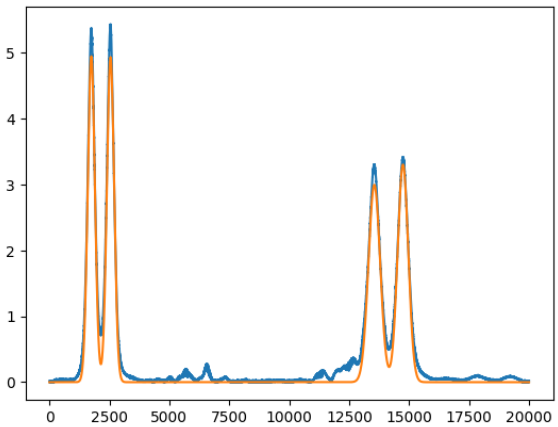


Figure A.2: Single peak fit of component 1

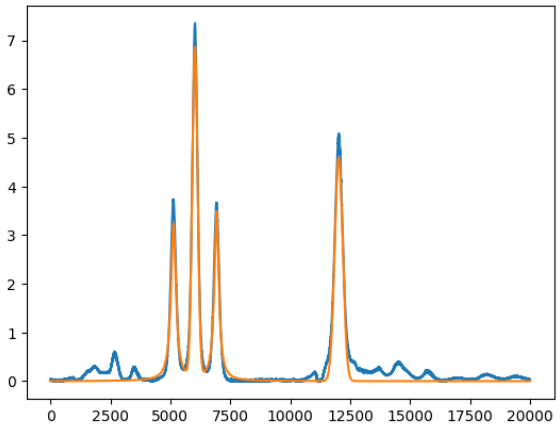


Figure A.3: Single peak fit of component 2

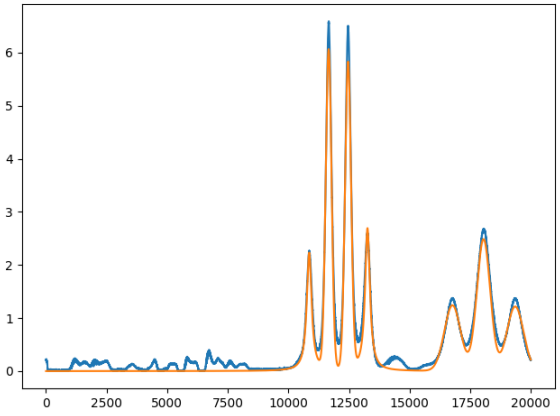


Figure A.4: Single peak fit of component 3

iv Learning rate test on artificial data

Error bar plots on different number of components

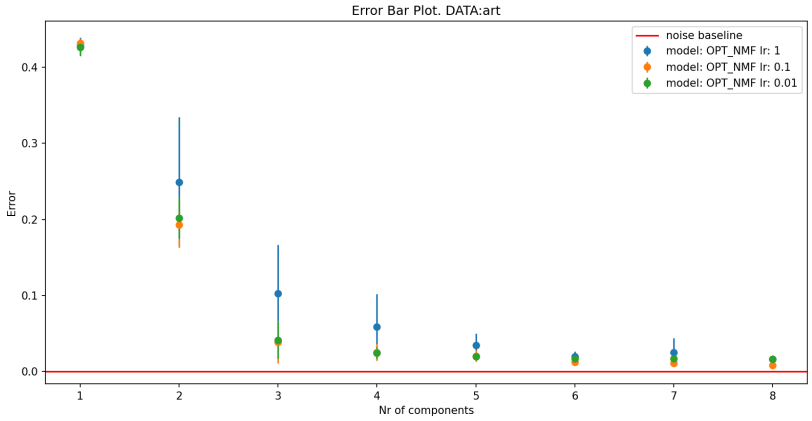


Figure A.5: learning rate test OPT

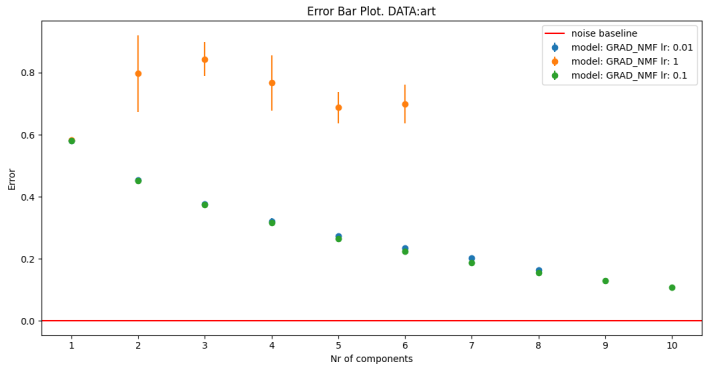


Figure A.6: Gradient optimized Tau learning rate test

Loss Curves

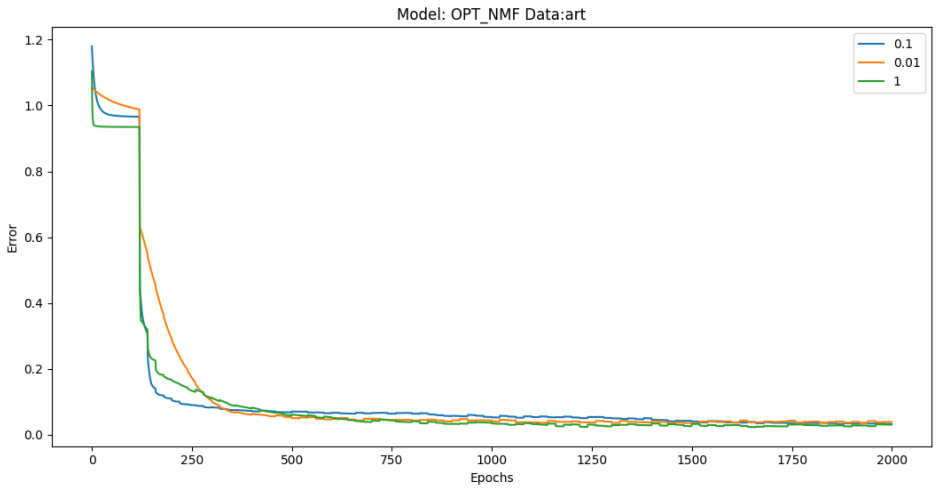


Figure A.7: Loss curve of cross correlation shiftNMF model on artificial data, using correct number of components (3)

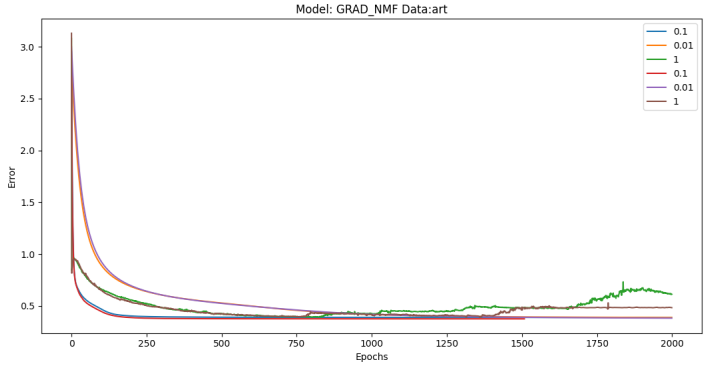


Figure A.8: Loss curve of Tau gradient optimized shiftNMF model on artificial data, using correct number of components (3)

v Full hardmodel regularization path

Artificial data

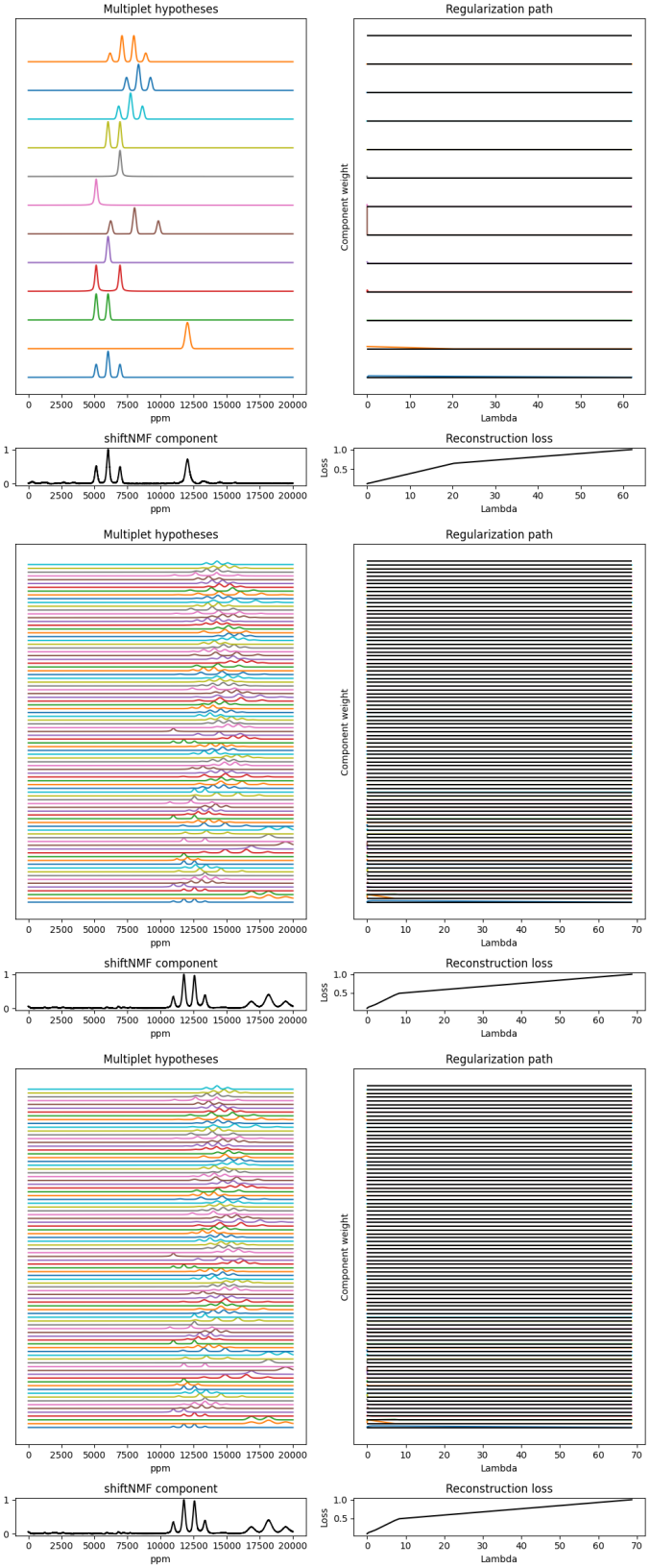


Figure A.9: Artificial components all hypotheses regularization path

1.3-1.6 ppm interval

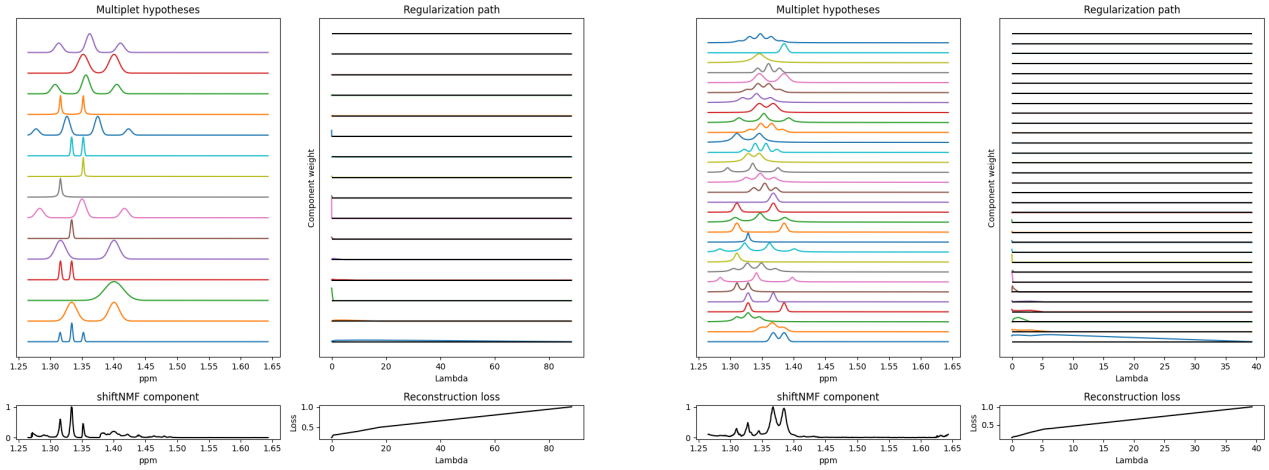


Figure A.10: Component 1 and component 2 all hypotheses regularization path

3.5-3.7 ppm interval

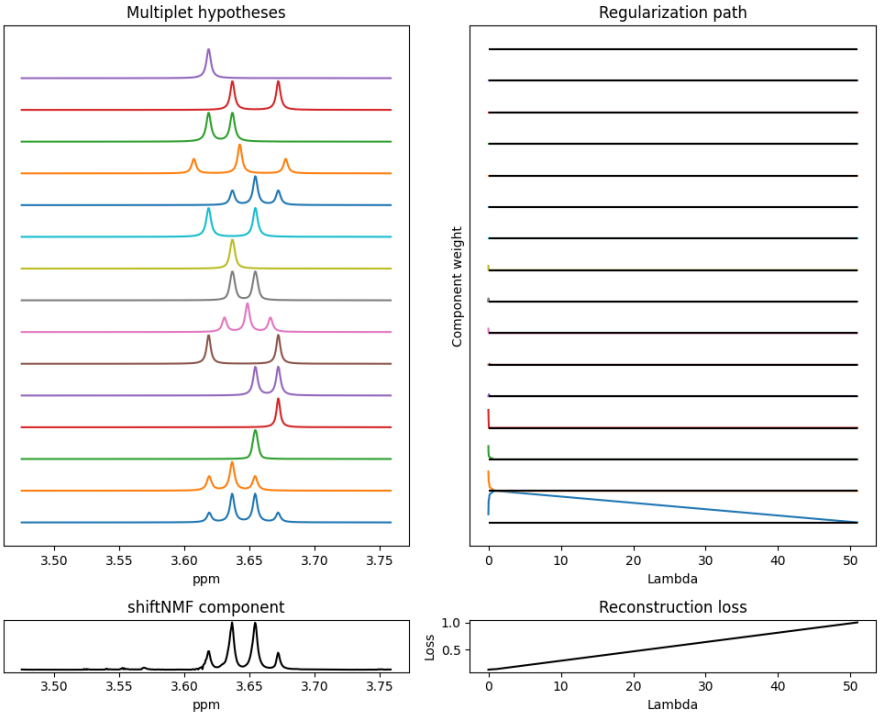


Figure A.11: Component 2 full hypotheses regularization path of wine dataset ppm 3.5-3.7

3.5-3.6 ppm interval

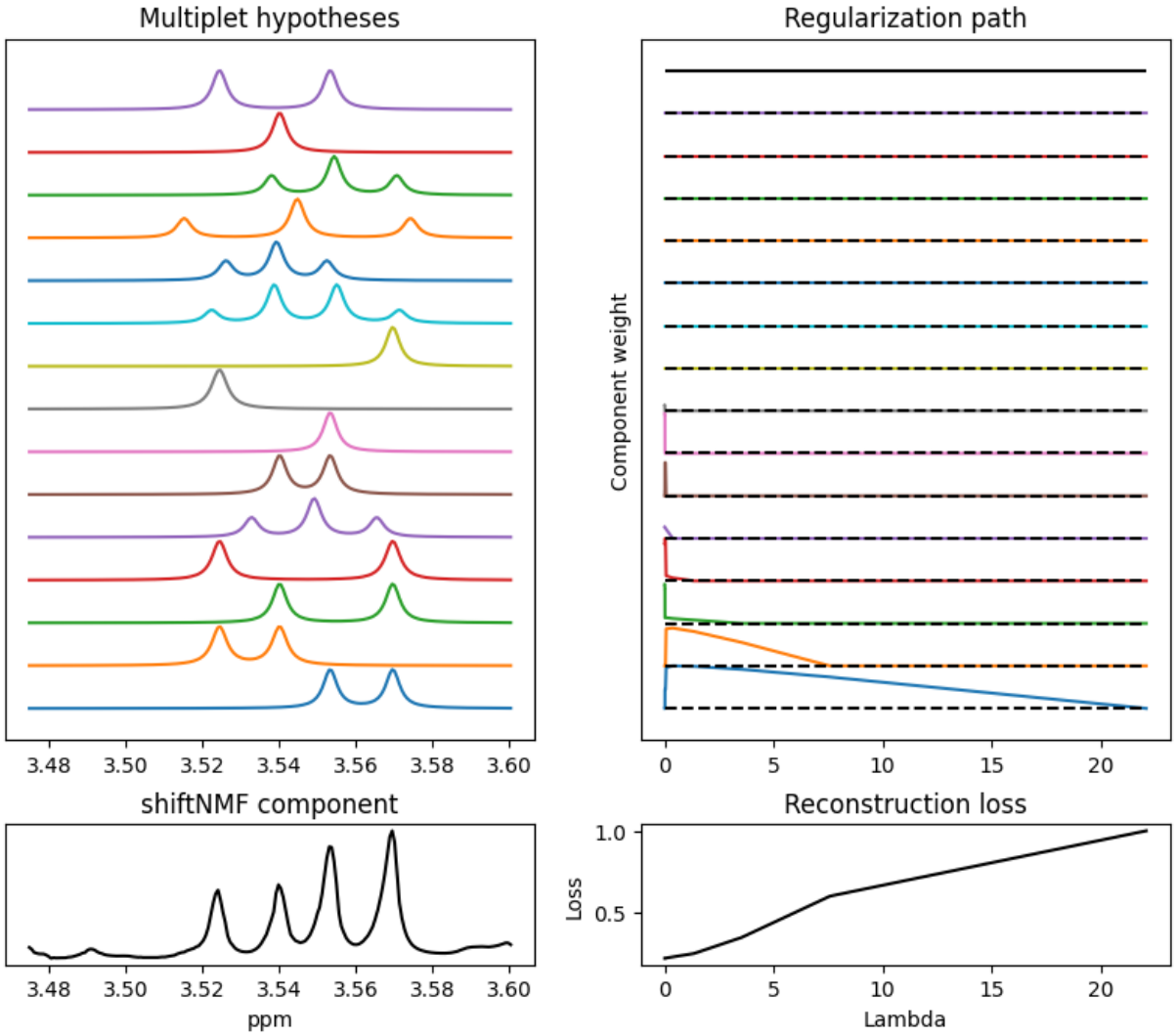


Figure A.12: All regularization paths for hard modelling of wine dataset ppm 3.5-3.6

Technical
University of
Denmark