IMPERIAL

Department of Mathematics

Probabilistic Sequential Matrix Factorisation for 12-Lead ECG Data

Joana Levtcheva

CID: 01252821

Supervised by Dr Deniz Akyildiz

August 31, 2024

Submitted in partial fulfilment of the requirements for the
MSc in Machine Learning and Data Science of
Imperial College London

The work contained in this thesis is my own work unless otherwise stated.

Signed: Joana Levtcheva Date: August 31, 2024

Abstract

Matrix factorisation (MF) techniques are highly effective and widely used in unsupervised machine learning. By decomposing the original matrix into multiple simpler lower-dimensional matrices, MF aims to uncover latent structures that are not immediately obvious in the original matrix. MF finds applications in areas such as image processing, natural language processing, missing data imputation, and recommendation systems. Despite considerable progression in the probabilistic versions, there is demand for such methods in applications such as uncertainty quantification, managing time-series data, and executing efficient probabilistic computations.

In this thesis, we show three novel applications of the algorithm Probabilistic Sequential Matrix Factorisation (PSMF) (Akyildiz et al. (2021)) to 12-lead ECG data. We explore the following tasks related to this complex high-dimensional time-series data with nonlinear subspace: missing data imputation with PSMF and the robust version of PSMF (rPSMF) (Akyildiz et al. (2021)) and compare their performance with other probabilistic sequential MF algorithms, we propose a novel R-peaks detection method, and we attempt to forecast an ECG component based on previous normal heartbeats using a Fourier basis with multiple terms and rank higher than one. We perform our experiments using the high-quality comprehensive dataset "A Large Scale 12-lead Electrocardiogram Database for Arrhythmia Study" (Zheng (2022), Zheng et al. (2020), Goldberger et al. (2000)). We describe and outline the experimenting process and the challenges we encountered modelling the ECG data, and summarise our results. We find that PSMF and rPSMF perform well when used for imputing missing data. We also find that applying PSMF to Lead II signal and then removing the reconstruction from the original signal indeed results in a useful R-peak detection method, but when it comes to forecasting there are certain challenges with modelling high amplitudes which open the door for further research on extending the PSMF algorithm to better handle the complex structure of ECG data.

Acknowledgements

I extend my deepest gratitude to my supervisor, Dr Deniz Akyildiz, for his support and very useful feedback throughout my research journey for this MSc thesis. I would also like to thank my friends and my family who have supported me through these two very intense years.

Contents

1. Introduction

Matrix factorisation (or also matrix decomposition) in the context of linear algebra is simply a factorisation of a matrix into a product of multiple matrices. Many different decompositions exist, and they find various applications in mathematical problems such as solving systems of linear equations, matrix inversion, determinant computation, eigenvalues problems, solving systems of first order ODEs, etc.

In this thesis, we are interested in matrix factorisation (MF) in the context of machine learning. Nowadays, MF techniques are highly effective and widely used in unsupervised machine learning. These methods aim to decompose the original matrix into multiple lower-dimensional matrices. By breaking the matrix into these simpler components MF aims to uncover latent structures that are not immediately apparent in the original matrix. Some applications are in image processing: for reducing dimensionality and noise in images, NLP for topic modelling, missing data imputation, recommendation systems, etc.

Formally, we are interested in the general problem of factorising a data matrix $Y \in \mathbb{R}^{m \times n}$ as

$$Y \approx CX, \tag{1.1}$$

where $C \in \mathbb{R}^{m \times r}$ is the *dictionary matrix*, $X \in \mathbb{R}^{r \times n}$ is the *coefficient matrix* (with columns the coefficients), and r is the *approximation rank* (Akyildiz and Míguez (2019)). Visually we can present the problem as

$$\underbrace{\begin{bmatrix} \times & \times & \times & \times & \times \\ \times & \times & \times & \times & \times \\ \times & \times & \times & \times & \times \end{bmatrix}}_{\mathrm{Y} \in \mathbb{R}^{m \times n}} \approx \underbrace{\begin{bmatrix} \times & \times \\ \times & \times \\ \times & \times \end{bmatrix}}_{\mathrm{C} \in \mathbb{R}^{m \times r}} \underbrace{\begin{bmatrix} \times & \times & \times & \times & \times \\ \times & \times & \times & \times & \times \end{bmatrix}}_{\mathrm{X} \in \mathbb{R}^{r \times n}}. \tag{1.2}$$

There are also probabilistic versions of MF which incorporate probabilistic models to better handle uncertainty and variability in the data, leading to more accurate predictions. Such methodologies postulate a prior distribution over the latent factors and necessitate the computation of the posterior distribution to derive updated estimates. With that the matrix is not only decomposed but probabilistic interpretations of the factors are also possible.

We should also note that some algorithms are suitable for sequential data: updating C

and X incrementally as new data points are observed and thus incorporating temporal dynamics and sequential dependencies into the factorisation process, and others are non-sequential: treating the dataset as a batch, independent of the time varying component.

Throughout this thesis we are going to focus on probabilistic sequential MF algorithms, along with their application to 12-lead ECG data, targeting the problem of managing high-dimensional time-series data with nonlinear subspace. Some examples of probabilistic MF algorithms are Probabilistic Matrix Factorisation (PMF) (Mnih and Salakhutdinov (2007)), Dictionary filtering (Akyildiz and Míguez (2019)), Probabilistic Sequential Matrix Factorization (PSMF) (Akyildiz et al. (2021)).

The paper "Probabilistic matrix factorisation" (PMF) (Mnih and Salakhutdinov (2007)) introduces an efficient and scalable probabilistic model for collaborative filtering. The algorithm performs well on large, sparse and imbalanced datasets. This is demonstrated by using a Netflix dataset, where PMF models the user preference matrix R as a product of two lower-dimensional matrices: user feature matrix U and movie feature matrix V. The conditional distribution over observed ratings is modeled using Gaussian noise, and zero-mean spherical Gaussian priors are placed on the user and movie feature vectors. The paper also presents two extensions to the initial PMF model: incorporating *adaptive priors* to automatically control the model complexity through these priors over the model parameters, and a *constrained PMF* version to handle and improve predictions for users with few ratings by incorporating constraints based on the assumption that users with similar movie ratings have similar preferences. The authors show that PMF significantly outperforms traditional Singular Value Decomposition (SVD) (Stewart (1993)) models and scales linearly with the number of observations. It's worth noting that PMF treats each rating as an independent event meaning the time varying component is not taken into consideration, making PMF a batch learning model designed to process large datasets in a non-sequential manner.

Later, in the paper "Dictionary Filtering: A Probabilistic Approach to Online Matrix Factorization" (DF) (Akyildiz and Míguez (2019)), the authors introduce a novel online MF algorithm known as dictionary filtering. It leverages probabilistic models, specifically using recursive linear filters, and efficiently factorises the original data matrix into a dictionary matrix and a coefficients matrix. This is an online and sequential algorithm, meaning it is suitable for high-dimensional and time-varying data, and it also has easy to tune parameters. DF is efficient for high-dimensional data with its computational complexity of $O(mr^2)$ independent of the number of data points. Although the model can learn non-stationary and dynamic data, it is developed for linear and Gaussian state space models (SSM).

Two years later, Akyildiz et.al. develop Probabilistic Sequential Matrix Factorization (PSMF) (Akyildiz et al. (2021)). This method is tailored to time-varying and non-stationary datasets consisting of high-dimensional time-series. Nonlinear Gaussian SSMs are considered, decomposing the original matrix into a dictionary matrix and time-

varying coefficient matrix. This time, the matrices are with potentially nonlinear dependencies, with PSMF efficiently capturing temporal dependencies through Markovian structures on the coefficients, making it possible to encode the dependencies into a lower dimensional latent space. The model is demonstrated to work on tasks such as forecasting, changepoint detection, missing data imputation, and is shown to work on real-world data with a periodic subspace. There is also a robust version, rPSMF, using Student-t filters to handle model misspecification, and a version for imputing missing data. Although the model is suitable for reducing high-dimensional data with periodic subspaces to lower-dimensional latent space, PSMF might struggle with very large datasets, having many data points.

1.1. Contributions

Using probabilistic methods, and specifically probabilistic sequential MF ones, on ECG data is not widely explored. Hence, we aim to introduce a few novel real-world applications of the PSMF method to 12-lead ECG data. The contributions made in this thesis are:

- **Application 1:** We use both PSMF and rPSMF for imputing missing data in the ECG signals, and compare the results with other sequential probabilistic MF models.
- **Application 2:** We show that PSMF can be used for R-peaks detection by introducing a simple approach. We remove the reconstructed signal, which has modelled the R-peaks smoother than the real ones, from the original data, and determine a suitable threshold for selecting the peaks.
- **Application 3:** We forecast an ECG component based on previous normal heartbeats by incorporating a Fourier basis with multiple Fourier terms and rank higher than 1.

1.2. Notation

We are going to denote the original data matrix as $Y \in \mathbb{R}^{m \times n}$, and let y_k denote the k-th column of the matrix. Also, let $y = \text{vec}(Y) \in \mathbb{R}^{nm}$ be the vectorization of the matrix Y, where

$$y := \text{vec}(Y) = \begin{bmatrix} y_1 \\ y_2 \\ \vdots \\ y_n, \end{bmatrix}, \tag{1.3}$$

and $y_1, ..., y_n$ are the columns of Y. The inverse is denoted as $Y := \text{vec}^{-1}(Y, m, n)$, where m, and n specify the size of the resulting matrix. With $y_{1:k} = \{y_1, ..., y_k\}$ we are going

to denote sequences. Further, let $C \in \mathbb{R}^{m \times r}$ be the dictionary matrix, $X \in \mathbb{R}^{r \times n}$ be the coefficient matrix, and r be the approximation rank.

With $I_m \in \mathbb{R}^{m \times m}$ we are going to denote the identity matrix, with $\mathcal{N}(x; \mu, \Sigma)$ the multivariate normal distribution with mean μ and Σ the covariance matrix, with $\mathcal{MN}(X; M, U, V)$ the matrix normal distribution with M the mean-matrix, U the row-covariance, and V the column covariance, with $\mathcal{IG}(s; \alpha, \beta)$ the inverse gamma distribution with shape α and scale β, and with $\mathcal{T}(x; \mu, \Sigma, \lambda)$ the multivariate t distribution, where μ is the mean, Σ the scale matrix, and λ is the degrees of freedom.

Finally, with $\odot$ we are going to denote element-wise multiplication, with tr(.) the trace of a matrix, and with $|.|$ the determinant of a matrix.

2. Background

2.1. Preliminaries

Here, we are going to introduce the matrix normal distribution, Kronecker product, and outline certain linear algebra properties in relation to Kronecker products. These properties are later used in some of the proofs.

2.1.1. Matrix Normal Distribution

We start with introducing the matrix normal distribution, which we are going to mainly use in its vectorised version.

Definition 2.1.1. Let $X \in \mathbb{R}^{p\times n}$ be a random matrix. Then X has a *matrix normal distribution* $\mathcal{MN}(X; M, U, V)$ with mean-matrix $M \in \mathbb{R}^{p\times n}$, row-covariance $U \in \mathbb{R}^{p\times p}$, and column-covariance $V \in \mathbb{R}^{n\times n}$, if $\text{vec}(X) \sim \mathcal{N}(\text{vec}(M), U \otimes V)$.

We can formally write that if $X \sim \mathcal{MN}(X; M, U, V)$, then

$$x \sim \mathcal{N}(x; \text{vec}(M), U \otimes V), \tag{2.1}$$

where $x = \text{vec}(X) \in \mathbb{R}^{pn}$, $\text{vec}(M) \in \mathbb{R}^{pn}$ is the mean, $U \times V \in \mathbb{R}^{pn\times pn}$ is the covariance matrix, and $\otimes$ is the Kronecker product (Gupta and Nagar (2000)).

Further, for the random matrix $X \in \mathbb{R}^{p\times n}$ the probability density function (p.d.f.) of the matrix normal distribution is defined as (Gupta and Nagar (2000))

$$(2\pi)^{-\frac{1}{2}np}|U|^{-\frac{1}{2}n}|V|^{-\frac{1}{2}p}\exp\{-\frac{1}{2}\text{tr}(U^{-1}(X-M)V^{-1}(X-M)^T)\}, \tag{2.2}$$

where $|U|$ and $|V|$ are the determinants of U and V respectively, tr(.) denotes the *trace* of a matrix, $X \in \mathbb{R}^{p\times n}$, $M \in \mathbb{R}^{p\times n}$.

2.1.2. Kronecker Product

Now, let's introduce the Kronecker product we already mentioned in relation to Definition 2.1.1.

Definition 2.1.2. Let $A \in \mathbb{R}^{m\times n}$ and $B \in \mathbb{R}^{p\times q}$ be matrices. Then their *Kronecker product* denoted as $A \otimes B \in \mathbb{R}^{mp\times nq}$ is given by (Harville (1997)):

$$A \otimes B = \begin{bmatrix} a_{11}B & a_{12}B & \dots & a_{1n}B \\ a_{21}B & a_{22}B & \dots & a_{2n}B \\ \vdots & \vdots & \ddots & \vdots \\ a_{m1}B & a_{m2}B & \dots & a_{mn}B \end{bmatrix}, \tag{2.3}$$

where $\{a_{ij}\}$, $i \in \{1, ..., m\}$, $j \in \{1, ..., n\}$ are the elements of A.

We are also going to need certain Kronecker product related properties. If we let $A \in \mathbb{R}^{m \times n}$, $X \in \mathbb{R}^{n \times p}$, and $B \in \mathbb{R}^{p \times q}$, then (Harville (1997))

$$\text{vec}(AXB) = (B^T \otimes A)\text{vec}(X). \tag{2.4}$$

Now, if we let $x \in \mathbb{R}^n$, from Equation 2.4 we can obtain

$$Ax = \text{vec}(Ax) = (x^T \otimes I_n)\text{vec}(A), \tag{2.5}$$

where $I_n \in \mathbb{R}^{n \times n}$ is the identity matrix. Harville (1997) also introduces a mixed product property. For matrices $A \in \mathbb{R}^{m \times n}$, $B \in \mathbb{R}^{p \times q}$, $C \in \mathbb{R}^{n \times u}$, and $D \in \mathbb{R}^{q \times v}$ it holds

$$(A \otimes B)(C \otimes D) = (AC) \otimes (BD). \tag{2.6}$$

Finally, for nonsingular matrices $A \in \mathbb{R}^{m \times m}$ and $B \in \mathbb{R}^{n \times n}$ we have that their product is invertible, and (Harville (1997))

$$(A \otimes B)^{-1} = A^{-1} \otimes B^{-1}. \tag{2.7}$$

2.2. PSMF

In the next sections we are outlining the PSMF models we are going to use in our experiments with 12-lead ECG data in Chapter 3. We are going to follow Akyildiz et al. (2021). First, we are going to introduce PSMF: model definition, inference, parameter estimation, marginal-likelihood approximation, and algorithm. Then, similarly, we are going to introduce the robust version of PSMF (rPSMF), and finally the version of PSMF for handling missing data.

We begin by introducing the PSMF model which can be thought of as being a probabilistic dimensionality reduction scheme. The dynamics of the subspace are modeled by a transition density (introduced in Equation 2.10), meaning the underlying dynamical structure present in the data will be reflected in the dynamics of the coefficients x_k, $k \geq 1$. Simply said, the sequential patterns in the data will be captured and represented by the behaviour of the coefficients over time.

2.2.1. Model

For observations $y_k \in \mathbb{R}^m$, $k \geq 1$, latent coefficients $x_k \in \mathbb{R}^r$, $k \geq 1$, and a dictionary matrix $C \in \mathbb{R}^{m \times r}$ the PSMF model can be described with the following state-space equations (Akyildiz et al. (2021))

$$p(C) = \mathcal{MN}(C; C_0, I_m, V_0) \tag{2.8}$$

$$p(x_0) = \mathcal{N}(x_0; \mu_0, P_0) \tag{2.9}$$

$$p_\theta(x_k \mid x_{k-1}) = \mathcal{N}(x_k; f_\theta(x_{k-1}), Q_k) \tag{2.10}$$

$$p(y_k \mid x_k, C) = \mathcal{N}(y_k; Cx_k, R_k) \tag{2.11}$$

where the nonlinear mapping $f_\theta : \mathbb{R}^r \times \Theta \to \mathbb{R}^r$ governs the dynamics of the coefficients, with $\Theta \subset \mathbb{R}^{m_\theta}$ representing the parameter subspace. The noise covariances of the coefficient dynamics in (2.10) and the observation model in (2.11) are denoted by Q_k and R_k for $k \geq 1$. P_0 and V_0 are respectively the initial covariances of the coefficients and the dictionary. Informally, we can refer to Equation (2.8) as the dictionary prior, Equation (2.9) as the initial state of the coefficients, Equation (2.10) as the transition density, and Equation (2.11) as the observation model.

2.2.2. Inference

In this section, we outline the algorithm for conducting sequential inference (Akyildiz et al. (2021)) within the model defined by equations (2.8)–(2.11). We begin by stating the optimal inference recursions which are intractable and thus need to be made tractable. This is achieved by introducing approximate sequential inference.

Optimal Inference

For the optimal inference recursions we have θ fixed, and we assume we know the following filters at time $k-1$: $p(x_{k-1}|y_{1:k-1})$ and $p(c|y_{1:k-1})$.

The **predictive distribution**, which is in the base of the update steps, is given by

$$p(x_k \mid y_{1:k-1}) = \int p(x_{k-1} \mid y_{1:k-1}) p(x_k \mid x_{k-1}) dx_{k-1}. \tag{2.12}$$

Given that past marginal is known, Equation (2.12) is independent of the dictionary.

In order to be able to compute updates, we state the **incremental marginal likelihood**

$$p(y_k \mid y_{1:k-1}) = \int \int p(y_k \mid c, x_k) p(x_k \mid y_{1:k-1}) p(c \mid y_{1:k-1}) dx_k dc. \tag{2.13}$$

Now, knowing $p(y_k \mid y_{1:k-1})$, for the **dictionary update** of C we have

$$p(c \mid y_{1:k}) = p(c \mid y_{1:k-1}) \frac{p(y_k \mid c, y_{1:k-1})}{p(y_k \mid y_{1:k-1})}, \tag{2.14}$$

where

$$p(y_k|c, y_{1:k-1}) = \int p(y_k|c, x_k) p(x_k|y_{1:k-1}) \mathrm{d}x_k. \tag{2.15}$$

For the **coefficients update** of x_k we have

$$p(x_k \mid y_{1:k}) = p(x_k \mid y_{1:k-1}) \frac{p(y_k \mid x_k, y_{1:k-1})}{p(y_k \mid y_{1:k-1})}, \tag{2.16}$$

where

$$p(y_k|x_k, y_{1:k-1}) = \int p(y_k|x_k, c) p(c|y_{1:k-1}) \mathrm{d}c. \tag{2.17}$$

As mentioned earlier, the integrals can be computed but the resulting distributions are not suitable for exact implementations of the update rules.

Approximate Sequential Inference

In this section we make the recursions tractable through using approximations, and outline the approximate sequential inference. Equation (2.8) can be rewritten as

$$p(c) = \mathcal{N}(c; c_0, V_0 \otimes I_m). \tag{2.18}$$

Let the given filters at time $k-1$ be

$$p(x_{k-1}|y_{1:k-1}) = \mathcal{N}(x_{k-1}; \mu_{k-1}, P_{k-1}) \tag{2.19}$$

and

$$p(c|y_{1:k-1}) = \mathcal{N}(c; c_{k-1}, V_{k-1} \otimes I_m) \tag{2.20}$$

Once again, using these distributions cannot give us exact updates for $p(c|y_{1:k})$ and $p(x_k|y_{1:k})$. Hence, let's introduce the notation $\tilde{p}(.)$ for the approximate densities when the distribution is not exact.

For the **prediction** we have to compute (2.12) which is not analytically tractable when $f_\theta(x)$ is a nonlinear mapping. By using the extended Kalman update (EKF) (McLean et al. (1962), Anderson and Moore (1979)) it is obtained (Akyildiz et al. (2021))

$$\tilde{p}(x_k|y_{1:k-1}) = \mathcal{N}(x_k; \bar{\mu}_k, \bar{P}_k), \tag{2.21}$$

where $\bar{\mu}_k = f_\theta(\mu_{k-1})$, $\bar{P}_k = F_k P_{k-1} F_k^T + Q_k$, and $F_K = \frac{\partial f_\theta(x)}{\partial x}|_{x=\bar{\mu}_{k-1}}$ is the Jacobian of f_θ calculated at $\bar{\mu}_{k-1}$.

For the **update** step we are once again interested in the dictionary update and the

coefficient update. All of the derivation details of the updates can be found in Akyildiz et al. (2021). Here, we are just going to outline the main steps of the update rules. First, we can see that (2.15) can be computed as

$$p(y_k|c, y_{1:k-1}) = \mathcal{N}(y_k; C\bar{\mu}_k, R_k + C\bar{P}_k C^T), \tag{2.22}$$

but to make this update tractable we make use of the approximation

$$C\bar{P}_k C^T \approx C_{k-1}\bar{P}_k C_{k-1}^T, \tag{2.23}$$

and apply it to (2.22), making this update tractable with likelihood $\mathcal{N}(y_k; C\bar{\mu}_k, R_k + C_{k-1}\bar{P}_k C_{k-1}^T)$. We also choose a Gaussian with a constant diagonal covariance, and we obtain the approximation

$$\tilde{p}(y_k|c, y_{1:k-1}) = \mathcal{N}(y_k; C\bar{\mu}_k, \eta_k \otimes I_m), \tag{2.24}$$

where

$$\eta_k = \frac{\mathrm{tr}(R_k + C_{k-1}\bar{P}_k C_{k-1}^T)}{m}, \tag{2.25}$$

which allows the analytical computation of the dictionary update $\tilde{p}(c|y_{1:k})$. In order to obtain the approximate **dictionary update** $\tilde{p}(c|y_{1:k})$ we have to make use of Proposition 1 in Akyildiz et al. (2021):

Proposition 2.2.1. Given $\tilde{p}(c|y_{1:k-1}) = \mathcal{N}(c; c_{k-1}, V_{k-1} \otimes I_m)$ and the likelihood $\tilde{p}(y_k|c, y_{1:k-1}) = \mathcal{N}(y_k; C\bar{\mu}_k, \eta_k \otimes I_m)$ the approximate posterior distribution is

$$\tilde{p}(c|y_{1:k}) = \mathcal{N}(c; c_k, V_k \otimes I_m), \tag{2.26}$$

where $c_k = \mathrm{vec}(C_k)$ and the posterior column-covariance matrix V_k is given by

$$V_k = V_{k-1} - \frac{V_{k-1}\bar{\mu}_k\bar{\mu}_k^T V_{k-1}}{\bar{\mu}_k^T V_{k-1}\bar{\mu}_k + \eta_k}, \quad k \geq 1, \tag{2.27}$$

and the posterior mean C_k of the dictionary C can be obtained in matrix-from as

$$C_k = C_{k-1} + \frac{(y_k - C_{k-1}\bar{\mu}_k)\bar{\mu}_k^T V_{k-1}^T}{\bar{\mu}_k^T V_{k-1}\bar{\mu}_k + \eta_k}, \quad k \geq 1. \tag{2.28}$$

Proof. See Akyildiz et al. (2021). □

To be able to derive the coefficient update, we need to use Proposition 2 from Akyildiz et al. (2021):

Proposition 2.2.2. Given $p(y_k|c, x_k) = \mathcal{N}(y_k; Cx_k, R_k)$ (as in 2.11) and $p(c|y_{1:k-1}, x_k) = \mathcal{N}(c; c_{k-1}, V_{k-1} \otimes I_m)$, we obtain

$$p(y_k|y_{1:k-1}, x_k) = \mathcal{N}(y_k; C_{k-1}x_k, R_k + x^T V_{k-1}x_k \otimes I_m). \tag{2.29}$$

Proof. See Akyildiz et al. (2021). □

We can approximate Equation 2.29 as

$$\tilde{p}(y_k|y_{1:k-1}, x_k) = \mathcal{N}(C_{k-1}x_k, \bar{R}_k), \tag{2.30}$$

where

$$\bar{R}_k = R_k + \bar{\mu}_k^T V_{k-1}\bar{\mu}_k \otimes I_m. \tag{2.31}$$

This result, along with an application of the Kalman update (Anderson and Moore (1979)), allows us to define the approximate posterior for the **coefficient update** (2.16)

$$\tilde{p}(x_k|y_{1:k}) = \mathcal{N}(x_k; \mu_k, P_k), \tag{2.32}$$

where

$$\mu_k = \bar{\mu}_k + \bar{P}_k C_{k-1}^T \left(C_{k-1}\bar{P}_k C_{k-1}^T + \bar{R}_k\right)^{-1} (y_k - C_{k-1}\bar{\mu}_k), \tag{2.33}$$

$$P_k = \bar{P}_k - \bar{P}_k C_{k-1}^T \left(C_{k-1}\bar{P}_k C_{k-1}^T + \bar{R}_k\right)^{-1} C_{k-1}\bar{P}_k. \tag{2.34}$$

2.2.3. Parameter Estimation

We need to estimate the parameters of f_θ in Equation 2.10. In order to do that we have to solve the following optimisation problem

$$\theta^\star \in \operatorname{argmax} \log p_\theta(y_{1:n}). \tag{2.35}$$

In this section we are going to outline the offline gradient ascent scheme for a limited number of data points, and call it **iterative estimation**, or *iterative* PSMF. There is also an online, recursive version (refer to Akyildiz et al. (2021)) that can be used on streaming data.

Iterative Estimation:

We can perform multiple passes over data with

$$\theta_i = \theta_{i-1} + \gamma \nabla \log \tilde{p}_\theta(y_{1:n})|_{\theta=\theta_{i-1}} \tag{2.36}$$

at the i-th iteration. We have that $\nabla \log p_\theta(y_{1:n})$ is intractable. Thus, during forward filtering we are going to use the approximation

$$\nabla \log \tilde{p}_\theta(y_{1:n}) = \sum_k^n \tilde{p}_\theta(y_k|y_{1:k-1}). \tag{2.37}$$

2.2.4. Approximating the marginal-likelihood

We are left with the task of approximating the log-marginal likelihood $\log \tilde{p}_\theta(y_k|y_{1:k-1})$. Given $\tilde{p}_\theta(y_k|y_{1:k-1}, c) = \mathcal{N}(y_k; Cf_\theta(\mu_{k-1}), \eta_k \otimes I_m)$ (from (2.24)), and $\tilde{p}(c|y_{1:k-1}) =$

$\mathcal{N}(c; c_{k-1}, V_{k-1} \otimes I_m)$, for the negative log-likelihood we have (Akyildiz et al. (2021))

$$-\log \tilde{p}_\theta(y_k|y_{1:k-1}) \stackrel{c}{=} \frac{m}{2} \log\left(\|f_\theta(\mu_{k-1})\|^2_{V_{k-1}} + \eta_k\right) + \frac{1}{2} \frac{\|y_k - C_{k-1} f_\theta(\mu_{k-1})\|^2}{\eta_k + \|f_\theta(\mu_{k-1})\|^2_{V_{k-1}}}, \quad (2.38)$$

where $\stackrel{c}{=}$ signifies equality up to additive constants that do not depend on θ. The gradients of the negative log-likelihood can be obtained through automatic differentiation for any general coefficient dynamics f_θ.

2.2.5. PSMF Algorithm

The iterative version of the PSMF algorithm is given in Algorithm 1. The online, recursive version of PSMF is not given here but can be found in Akyildiz et al. (2021).

Algorithm 1 Iterative PSMF

Initialize γ, θ_0, C_0, V_0, μ_0, P_0, $(Q)_k \geq 1$, $(R)_k \geq 1$.
for $i \geq 1$ **do**
  $C_0 = C_n$, $\mu_0 = \mu_n$, $P_0 = P_n$, $V_0 = V_n$
  for $1 \leq k \leq n$ **do**
    Compute predictive mean of x_k:
      $\bar{\mu}_k = f_{\theta_{i-1}}(\mu_{k-1})$
    Compute predictive covariance of x_k:
      $\bar{P}_k = F_k P_{k-1} F_k^\top + Q_k$, with $F_k = \frac{\partial f(x)}{\partial x}\Big|_{x=\bar{\mu}_{k-1}}$
    Update dictionary mean C_k:
      $C_k = C_{k-1} + \frac{(y_k - C_{k-1}\bar{\mu}_k)\bar{\mu}_k^T V_{k-1}^T}{\bar{\mu}_k^T V_{k-1} \bar{\mu}_k + \eta_k}$
    where $\eta_k = \mathrm{tr}(C_{k-1}\bar{P}_k C_{k-1}^\top + R_k)/m$.
    Update dictionary covariance V_k:
      $V_k = V_{k-1} - \frac{V_{k-1}\bar{\mu}_k \bar{\mu}_k^T V_{k-1}}{\bar{\mu}_k^T V_{k-1}\bar{\mu}_k + \eta_k}$
    Update coefficient mean μ_k:
      $\mu_k = \bar{\mu}_k + \bar{P}_k C_{k-1}^T (C_{k-1}\bar{P}_k C_{k-1}^T + \bar{R}_k)^{-1}(y_k - C_{k-1}\bar{\mu}_k)$
    Update coefficient covariance P_k:
      $P_k = \bar{P}_k - \bar{P}_k C_{k-1}^T (C_{k-1}\bar{P}_k C_{k-1}^T + \bar{R}_k)^{-1} C_{k-1}\bar{P}_k$
  Update parameters:
    $\theta_i = \theta_{i-1} + \gamma \sum_{k=1}^n \nabla \log \tilde{p}_\theta(y_k|y_{1:k-1})|_{\theta=\theta_{i-1}}$

2.3. rPSMF

There are cases when the model described in equations (2.8)-(2.11) may not perform optimally when the practitioner is unsure about how to set the hyperparameters or when they are misspecified. That's when the robust version of the PSMF model (also introduced in Akyildiz et al. (2021)), rPSMF for short, is of good use: incorporating

robustness to outliers and model misspecifications by integrating a multivariate t distribution into the framework. In rPSMF an inverse-gamma-distributed scale variable s is introduced, as shown in equations (2.39)-(2.43), and only the initial noise covariances Q_0 and R_0 need to be specified. The inference and parameter estimation follows the one for PSMF, but with some modifications to accommodate the usage of the multivariate t distribution.

2.3.1. Model

The rPSMF model can be described with the following state-space equations (Akyildiz et al. (2021))

$$p(s) = \mathcal{IG}(s; \lambda_0/2, \lambda_0/2) \tag{2.39}$$

$$p(C \mid s) = \mathcal{MN}(C; C_0, I_m, sV_0) \tag{2.40}$$

$$p(x_0 \mid s) = \mathcal{N}(x_0; \mu_0, sP_0) \tag{2.41}$$

$$p_\theta(x_k \mid x_{k-1}, s) = \mathcal{N}(x_k; f_\theta(x_{k-1}), sQ_0) \tag{2.42}$$

$$p(y_k \mid x_k, C, s) = \mathcal{N}(y_k; Cx_k, sR_0), \tag{2.43}$$

where s is an inverse-gamma-distributed scale variable, R_0 and Q_0 are the initial noise covariances.

For clarity, we are going to define the inverse-gamma-distribution and the multivariate t distribution we are going to use.

Definition 2.3.1. The *inverse-gamma distribution* is defined as

$$\mathcal{IG}(s; \alpha, \beta) = \frac{\beta^\alpha}{\Gamma(\alpha)} \left(\frac{1}{s}\right)^{\alpha+1} \exp\left(-\frac{\beta}{s}\right), \tag{2.44}$$

for $\alpha > 0$, $\beta > 0$, and $\Gamma(.)$ denoting the *Gamma function*.

Definition 2.3.2. The *multivariate t distribution* over $y \in \mathbb{R}^m$ with λ degrees of freedom is defined as

$$\mathcal{T}(y; \mu, \Sigma, \lambda) = \frac{1}{(\pi\lambda)^{m/2}|\Sigma|^{1/2}} \frac{\Gamma((\lambda+m)/2)}{\Gamma(\lambda/2)} \left(1 + \frac{\Delta^2}{\lambda}\right)^{-(\lambda+m)/2}, \tag{2.45}$$

where $\Delta^2 = (y-\mu)^T\Sigma^{-1}(y-\mu)$.

2.3.2. Inference

The inference in rPSMF follows the one for PSMF, but with some modifications to accommodate the usage of the multivariate t distribution. The introduction of multivariate t distribution leads to an increase of the degrees of freedom in the update equations by m at every iteration:

$$\lambda_k = \lambda_{k-1} + m. \tag{2.46}$$

If we let

$$\Delta_{1,k}^2 = (y_k - C_{k-1}\bar{\mu}_k)^T (C_{k-1}\bar{P}_k C_{k-1}^T + \bar{R}_k)^{-1} (y_k - C_{k-1}\bar{\mu}_k) \tag{2.47}$$

and

$$\omega_k = (\lambda_{k-1} + \Delta_{1,k}^2)/(\lambda_{k-1} + m), \tag{2.48}$$

then (Akyildiz et al. (2021))

$$s_0 = s, \quad s_k = \omega_k^{-1} s_{k-1} \tag{2.49}$$

and for the noise covariances update we have

$$Q_k = \omega_k Q_{k-1}, \tag{2.50}$$

$$R_k = \omega_k R_{k-1}. \tag{2.51}$$

The updates for the mean for the coefficients and the dictionary are the same, but the updates for the coefficient covariance P_k and the dictionary column-covariance V_k undergo changes. For P_k the Student's t update contributes to multiplying the right-hand side of Equation 2.39 by ω_k, leading to

$$P_k = \omega_k \left(\bar{P}_k - \bar{P}_k C_{k-1}^T \left(C_{k-1} \bar{P}_k C_{k-1}^T + \bar{R}_k \right)^{-1} C_{k-1} \bar{P}_k \right). \tag{2.52}$$

If we let

$$\bar{\rho}_k = \bar{\mu}_k^T V_{k-1} \bar{\mu}_k + \eta_k \tag{2.53}$$

and

$$\Delta_{2,k}^2 = \|y_k - C_{k-1}\bar{\mu}_k\|^2 / \bar{\rho}_k, \tag{2.54}$$

then the update of V_k (Equation 2.27) becomes

$$V_k = \varphi_k \left(V_{k-1} - \frac{V_{k-1}\bar{\mu}_k \bar{\mu}_k^T V_{k-1}}{\bar{\mu}_k^T V_{k-1} \bar{\mu}_k + \eta_k} \right), \tag{2.55}$$

where

$$\varphi_k = \frac{\lambda_{k-1} + \Delta_{2,k}^2}{\lambda_{k-1} + m} \tag{2.56}$$

Detailed derivation can be found in Supp. F in Akyildiz et al. (2021).

2.3.3. Parameter Estimation

The parameter estimation in rPSMF is analogous to the one in PSMF outlined in Section 2.2.3.

2.3.4. Approximating the marginal-likelihood

Following the approach in Section 2.2.4, for the negative log-likelihood it can be obtained (Akyildiz et al. (2021))

$$-\log \tilde{p}_\theta(y_k|y_{1:k-1}) \stackrel{c}{=} \frac{m}{2}\log\left(\|f_\theta(\mu_{k-1})\|^2_{V_{k-1}} + \eta_k\right) + \\ + \left(\frac{\lambda_{k-1}+m}{2}\right)\log\left(1 + \frac{\|y_k - C_{k-1}f_\theta(\mu_{k-1})\|^2}{\lambda_{k-1}\left(\|f_\theta(\mu_{k-1})\|^2_{V_{k-1}} + \eta_k\right)}\right), \quad (2.57)$$

where $\stackrel{c}{=}$ signifies equality up to additive constants that do not depend on θ.

2.3.5. rPSMF Algorithm

The iterative version of the rPSMF algorithm is given in Algorithm 2. The online, recursive version of rPSMF is not given here but can be found in Akyildiz et al. (2021).

Algorithm 2 Iterative rPSMF

Initialize $\gamma, \theta_0, C_0, V_0, \mu_0, P_0, Q_0, R_0$.
for $i \geq 1$ **do**
  $C_0 = C_T$, $\mu_0 = \mu_T$, $P_0 = P_T$, $V_0 = V_T$
  for $1 \leq k \leq T$ **do**
    Predictive mean of x_k:
      $\bar{\mu}_k = f_{\theta_{i-1}}(\mu_{k-1})$
    Predictive covariance of x_k:
      $\bar{P}_k = F_k P_{k-1} F_k^\top + Q_k$, where $F_k = \frac{\partial f(x)}{\partial x}\Big|_{x=\bar{\mu}_{k-1}}$
    Compute scaling factor for the dictionary update
      $\varphi_k = \frac{\lambda_{k-1}}{\lambda_{k-1}+m} + \frac{(y_k - C_{k-1}\bar{\mu}_k)^\top (y_k - C_{k-1}\bar{\mu}_k)}{\bar{\mu}_k^\top V_{k-1}\bar{\mu}_k + \eta_k}$
    where $\eta_k = \mathrm{tr}(C_{k-1}\bar{P}_k C_{k-1}^\top + R_{k-1})/m$.
    Mean and covariance updates of the dictionary
      $C_k = C_{k-1} + \frac{(y_k - C_{k-1}\bar{\mu}_k)\bar{\mu}_k^\top V_{k-1}}{\bar{\mu}_k^\top V_{k-1}\bar{\mu}_k + \eta_k}$ and $V_k = \varphi_k\left(V_{k-1} - \frac{V_{k-1}\bar{\mu}_k\bar{\mu}_k^\top V_{k-1}}{\bar{\mu}_k^\top V_{k-1}\bar{\mu}_k + \eta_k}\right)$
    Compute scaling factor for the coefficient update
      $\omega_k = \lambda_{k-1} + (y_k - C_{k-1}\bar{\mu}_k)^\top S_k^{-1}(y_k - C_{k-1}\bar{\mu}_k)$
    where $S_k = C_{k-1}\bar{P}_k C_{k-1}^\top + \bar{R}_k$ and $\bar{R}_k = R_{k-1} + \bar{\mu}_k^\top V_{k-1}\bar{\mu}_k \otimes I_m$.
    Mean and covariance updates of coefficients
      $\mu_k = \bar{\mu}_k + \bar{P}_k C_{k-1}^\top S_k^{-1}(y_k - C_{k-1}\bar{\mu}_k)$ and $P_k = \omega_k(\bar{P}_k - \bar{P}_k C_{k-1}^\top S_k^{-1} C_{k-1}\bar{P}_k)$
    Update noise covariances:
      $Q_k = \omega_k Q_{k-1}$ and $R_k = \omega_k R_{k-1}$
    Update degrees of freedom:
      $\lambda_k = \lambda_{k-1} + m$
  Parameter update:
    $\theta_i = \theta_{i-1} + \gamma \sum_{k=1}^T \nabla_\theta \log p_\theta(y_k|y_{1:k-1})|_{\theta=\theta_{i-1}}$

2.4. PSMF for Handling Missing Data

The PSMF algorithm can be modified to handle missing data and solve tasks for missing data imputation in sequential context. We can also note that handling missing values with rPSMF follows the same reasoning we are going to outline here for PSMF.

2.4.1. Model

Let $m_k \in \{0,1\}^m$ be a mask vector, where zero corresponds to missing entries, and ones corresponds to non-missing values. If an observation vector has missing entries we can model it as $z_k = m_k \odot y_k$. Further, $z_k = M_k y_k$, where $M_k = \text{diag}(m_k)$, and hence

$$p(z_k|c, x_k) = \mathcal{N}(z_k; M_k C x_k, M_k R_k M_k^T). \tag{2.58}$$

This modification of the likelihood is the only change we need to do in the PSMF model described by equations (2.8)-(2.11) in order to be able to handle missing data. Meaning, the PSMF model for handling missing data can be described with the following state-space equations (Akyildiz et al. (2021))

$$p(C) = \mathcal{MN}(C; C_0, I_m, V_0) \tag{2.59}$$

$$p(x_0) = \mathcal{N}(x_0; \mu_0, P_0) \tag{2.60}$$

$$p_\theta(x_k|x_{k-1}) = \mathcal{N}(x_k; f_\theta(x_{k-1}), Q_k) \tag{2.61}$$

$$p(z_k|x_k, C) = \mathcal{N}(z_k; M_k C x_k, M_k R_k M_k^T). \tag{2.62}$$

2.4.2. Inference

Let

$$\tilde{p}(c|z_{1:k-1}) = \mathcal{N}(c; c_{k-1}, V_{k-1} \otimes I_m) \tag{2.63}$$

and

$$\tilde{p}(z_k|c, z_{1:k-1}) = \mathcal{N}(z_k; M_k C \bar{\mu}_k, \eta_k \otimes I_m), \tag{2.64}$$

where

$$\eta_k = \frac{\text{tr}(M_k R_k M_k^T + M_k C_{k-1} \bar{P}_k C_{k-1}^T M_k^T)}{m}. \tag{2.65}$$

By applying the property given in (2.4) we get

$$\tilde{p}(z_k|c, z_{1:k-1}) = \mathcal{N}(z_k; H_k c, \eta_k \otimes I_m), \tag{2.66}$$

where $c = \text{vec}(C)$, and $H_k = \bar{\mu}_K^T \otimes M_k$. Following the proof of Proposition 2.2.1 (refer to Akyildiz et al. (2021)) along with the approximation

$$\bar{\mu}_k^T V_{k-1} \bar{\mu}_k \otimes M_k \approx \bar{\mu}_k^T V_{k-1} \bar{\mu}_k \otimes I_m \tag{2.67}$$

for the covariance update it follows

$$P_k = V_{k-1} \otimes I_m - \frac{V_{k-1}\bar{\mu}_k\bar{\mu}_k^T V_{k-1}}{\bar{\mu}_k^T V_{k-1}\bar{\mu}_k + \eta_k} \otimes M_k, \tag{2.68}$$

but this can't be simplified so we have to approximate it as

$$P_k \approx V_k \otimes I_m, \tag{2.69}$$

where $V_k = V_{k-1} - \frac{V_{k-1}\bar{\mu}_k\bar{\mu}_k^T V_{k-1}}{\bar{\mu}_k^T V_{k-1}\bar{\mu}_k + \eta_k}$.
For the mean update, again by following the proof of Proposition 2.2.1, it follows (Akyildiz et al. (2021))

$$C_k = C_{k-1} + \frac{(z_k - M_k C_{k-1}\bar{\mu}_k)\,\bar{\mu}_k^T V_{k-1}}{\bar{\mu}_k^T V_{k-1}\bar{\mu}_k + \eta_k}, \quad k \geq 1. \tag{2.70}$$

We also have to update the coefficients related to x_k. If we fix C_{k-1} we can derive the updates by replacing C_{k-1} with $M_k C_{k-1}$ in the update rules for $(x_k)_{k\geq 1}$. We get

$$\mu_k = \bar{\mu}_k + \bar{P}_k C_{k-1}^\top M_k^\top S_k^{-1}(z_k - M_k C_{k-1}\bar{\mu}_k), \tag{2.71}$$

$$P_k = \bar{P}_k - \bar{P}_k C_{k-1}^\top M_k^\top S_k^{-1} M_k C_{k-1}\bar{P}_k, \tag{2.72}$$

where

$$S_k = M_k C_{k-1}\bar{P}_k C_{k-1}^\top M_k^\top + M_k R_k M_k^\top. \tag{2.73}$$

2.4.3. Approximating the marginal-likelihood

Finally, for the negative log-likelihood $-\log \tilde{p}_\theta(z_k|z_{1:k-1})$ we have (Akyildiz et al. (2021))

$$-\log \tilde{p}_\theta(z_k|z_{1:k-1}) \stackrel{c}{=} \frac{1}{2}\sum_{j=1}^{m} \log u_{jk} + \frac{1}{2}(z_k - M_k C_{k-1} f_\theta(\mu_{k-1}))^\top U_k^{-1}(z_k - M_k C_{k-1} f_\theta(\mu_{k-1})), \tag{2.74}$$

where $\stackrel{c}{=}$ signifies equality up to additive constants that do not depend on θ and $U_k = \|f_\theta(\mu_{k-1})\|_{V_{k-1}^2} \otimes M_k + \eta_k \otimes I_m$ is a m-dimensional diagonal matrix with elements u_{jk} for $j = 1, \ldots, m$.

2.4.4. Algorithm

The iterative PSMF algorithm for handling missing data is given in Algorithm 3.

Algorithm 3 Iterative PSMF for Handling Missing Data

Initialize γ, θ_0, C_0, V_0, μ_0, P_0, $(Q)_{k\geq 1}$, $(R)_{k\geq 1}$, and missing data mask M.
for $i \geq 1$ **do**
  $C_0 = C_n$, $\mu_0 = \mu_n$, $P_0 = P_n$, $V_0 = V_n$
  for $1 \leq k \leq n$ **do**
    Compute predictive mean of x_k:
      $\bar{\mu}_k = f_{\theta_{i-1}}(\mu_{k-1})$
    Compute predictive covariance of x_k:
      $\bar{P}_k = F_k P_{k-1} F_k^\top + Q_k$, with $F_k = \frac{\partial f(x)}{\partial x}\Big|_{x=\bar{\mu}_{k-1}}$
    Update dictionary mean C_k:
      $C_k = C_{k-1} + \frac{(z_k - M_k C_{k-1}\bar{\mu}_k)\bar{\mu}_k^\top V_{k-1}^\top}{\bar{\mu}_k^\top V_{k-1}\bar{\mu}_k + \eta_k}$
    where $\eta_k = \mathrm{tr}(C_{k-1}\bar{P}_k C_{k-1}^\top + R_k)/m$.
    Update dictionary covariance V_k:
      $V_k = V_{k-1} - \frac{V_{k-1}\bar{\mu}_k\bar{\mu}_k^\top V_{k-1}}{\bar{\mu}_k^\top V_{k-1}\bar{\mu}_k + \eta_k}$
    Update coefficient mean μ_k:
      $\mu_k = \bar{\mu}_k + \bar{P}_k C_{k-1}^\top M_k^\top S_k^{-1}(z_k - M_k C_{k-1}\bar{\mu}_k)$
      where $S_k = M_k C_{k-1}\bar{P}_k C_{k-1}^\top M_k^\top + M_k R_k M_k^\top$
    Update coefficient covariance P_k:
      $P_k = \bar{P}_k - \bar{P}_k C_{k-1}^\top M_k^\top S_k^{-1} M_k C_{k-1}\bar{P}_k$
  Update parameters:
    Iterative: $\theta_i = \theta_{i-1} + \gamma \sum_{k=1}^{n} \nabla \log \tilde{p}_\theta(z_k|z_{1:k-1})|_{\theta=\theta_{i-1}}$

2.5. Introduction to ECG

2.5.1. ECG

An electrocardiogram (ECG) is a medical graphical representation of the changes in the strength and direction of the heart's electrical activity over a period of time using electrodes placed on the skin. These electrodes detect the tiny electrical changes on the skin that arise from the heart muscle's electrophysiologic pattern of depolarizing and repolarizing during each heartbeat. It is a very common, non-invasive procedure used to diagnose rhythm abnormalities, electrical conduction changes, myocardial ischemia and infarction, and generally to monitor cardiac health, and be of help to treatment decisions.

The electrical signals generated during the cardiac cycle, consisting of depolarization and repolarization of the heart muscle cells, propagate through the conductive tissues surrounding the heart. By placing electrodes at specific locations on the body, these electrical signals can be detected and recorded as an ECG. The repeating patterns observed in an ECG reflect the sequence of electrical activity in the atria and ventricles. Instead of measuring the absolute voltage, an ECG records the voltage changes relative to a baseline. (Klabunde (2012 - 2012))

2.5.2. PQRST Complex

In Figure 2.1 we can see the components of the ECG trace. A closer look at one of the repeating waveforms in an ECG rhythm strip (PQRST complex) reveals several distinct components, each representing a specific phase of the cardiac cycle. The P wave corresponds to atrial depolarization, while the QRS complex represents ventricular depolarization. The T wave, which follows the QRS complex, indicates ventricular repolarization. The PR interval is the time taken for the depolarization wave to travel through the atria and the atrioventricular (AV) node. The QT interval encompasses the entire period of ventricular depolarization and repolarization. Between the QRS complex and the T wave lies the ST segment, an isoelectric period during which the entire ventricle is in a depolarized state.

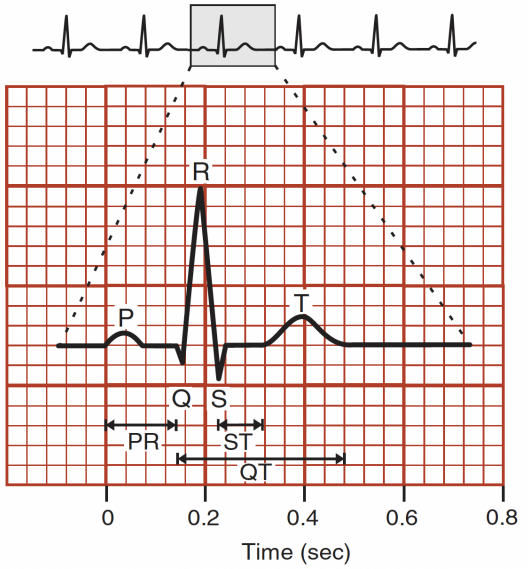

Figure 2.1.: Components of the ECG trace. (Klabunde (2012 - 2012))

2.5.3. R-peak

The R-peak is the tallest and most prominent wave in the QRS complex of an ECG signal, the peak of the R-wave. It represents the depolarization of the left and right ventricles, which are the main pumping chambers of the heart. The ability to accurately detect R-peaks is crucial for several reasons:

- Heart Rate Calculation: The time interval between consecutive R-peaks, known as the RR interval, is used to calculate the heart rate. By detecting R-peaks, we can determine the heart rate and monitor any changes or irregularities.
- Cardiac Rhythm Analysis: R-peak detection helps in identifying the regularity and pattern of the heart's rhythm. By analyzing the sequence of R-peaks, we can

detect various arrhythmias, such as premature ventricular contractions (PVCs), atrial fibrillation, and heart blocks.

- Signal Quality Assessment: The consistency and clarity of R-peaks can provide insights into the overall quality of the ECG signal. If R-peaks are difficult to detect or have variable amplitudes, it may indicate the presence of noise, artifacts, or poor electrode contact.
- Temporal Alignment: R-peaks serve as reference points for aligning and comparing different ECG leads or signals from multiple subjects. By aligning the signals based on R-peaks, we can analyze the relative timing and morphology of other ECG components.
- Feature Extraction: Various features can be extracted from the ECG signal using R-peaks as anchor points. These features, such as RR intervals, QRS duration, and ST segment changes, provide valuable information for diagnosing and monitoring cardiac conditions.

Many methods have been developed for R-peak detection, such as:

- Pan-Tompkins algorithm
- Wavelet-based methods
- Hilbert transform-based methods
- Machine learning approaches

These algorithms aim to accurately and efficiently detect R-peaks in the presence of noise, baseline wander, and other artifacts commonly found in ECG signals. Accurate R-peak detection is essential for the reliable interpretation of ECG data and the diagnosis of various cardiac conditions.

2.5.4. 12-Lead ECG

An ECG can contain 12 ECG traces, called 12-lead ECG. Simply said, a 12-lead ECG provides a comprehensive view of the heart's electrical activity from different angles. It consists of 12 different leads, each representing a specific view of the heart. The leads are divided into two main groups: limb leads and precordial leads.

The so called **Einthoven's Triangle** (can be seen in Figure 3.1.3) is a theoretical formation of the limb leads (I, II, III), forming an equilateral triangle with the heart at the center. This relationship helps in understanding the orientation and magnitude of the heart's electrical activity. The voltages in these leads relate to each other in the following way:

$$\text{Lead I} + \text{Lead III} = \text{Lead II}. \tag{2.75}$$

The voltages in augmented limb leads are calculated by amplifying the electrical difference between one limb electrode and a central point formed by the other limb electrodes. These relationships are based on the principle that the sum of the voltages from all three limb leads should be zero when the heart's electrical axis is normal. Let's outline the leads in the limb leads group:

Limb Leads: The electrodes for these signals are one on each arm and one on the left leg.

- **Bipolar Limb Leads** (Figure 3.1.3):
  - **Lead I:** Measures the voltage difference between the left arm (LA) and right arm (RA).
$$\text{Lead I} = \text{LA} - \text{RA} \tag{2.76}$$
  - **Lead II:** Measures the voltage difference between the left leg (LL) and the right arm.
$$\text{Lead II} = \text{LL} - \text{RA} \tag{2.77}$$
  - **Lead III:** Measures the voltage difference between the left leg and the left arm.
$$\text{Lead III} = \text{LL} - \text{LA} \tag{2.78}$$
- **Augmented Unipolar Limb Leads** (Figure 3.1.3):
  - **aVR:** Measures the electrical difference from the right arm to a central point between the left arm and left leg.
$$\text{aVR} = \text{RA} - \frac{1}{2}(\text{LA} + \text{LL}) \tag{2.79}$$
  - **aVL:** Measures the electrical difference from the left arm to a central point between the right arm and left leg.
$$\text{aVL} = \text{LA} - \frac{1}{2}(\text{RA} + \text{LL}) \tag{2.80}$$
  - **aVF:** Measures the electrical difference from the left leg to a central point between the right arm and left arm.
$$\text{aVF} = \text{LL} - \frac{1}{2}(\text{RA} + \text{LA}) \tag{2.81}$$

The precordial leads are positioned in the horizontal plane, at right angles to the other six leads. These leads consist of six electrodes (V1 through V6) that serve as positive poles for their respective precordial leads. The negative pole for these leads is provided by Wilson's central terminal.

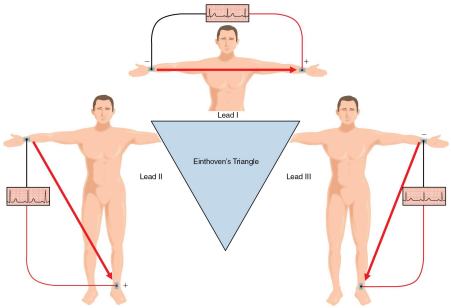

Figure 2.2.: Bipolar limb leads. (Wesley and Huszar (2021))

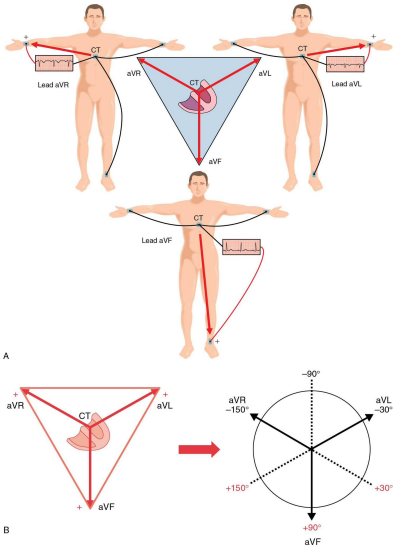

Figure 2.3.: Augmented unipolar limb leads. (Wesley and Huszar (2021))

Precordial Unipolar Leads (Figure 2.4):

- **V1 to V6:** These leads provide views of the horizontal plane of the heart, from right to left.

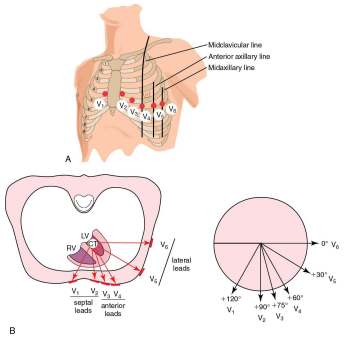

Figure 2.4.: Precordial unipolar leads. (Wesley and Huszar (2021))

3. Experiments and Results

For the experiments we are going to use the already available dataset "A Large Scale 12-lead Electrocardiogram Database for Arrhythmia Study" (Zheng (2022), Zheng et al. (2020), Goldberger et al. (2000)). This is a comprehensive database of high-quality 12-lead ECG signals collected from 45152 patients. Each signal is with length of 10 seconds corresponding to 5000 data points. The dataset is designed to support arrhythmia research, containing labeled data for various cardiac conditions such as atrial fibrillation, premature ventricular contractions, and bundle branch blocks. This large dataset has high-quality labels from professional experts, diverse arrhythmia types, and additional cardiovascular conditions, making it suitable for performing our tests on it. Code to reproduce our experiments is available in an online GitHub repository.[1]

3.1. Missing Data Imputation

In this section, we evaluate PSMF and rPSMF on the task of imputing missing values in 12-lead ECG time-series data.

3.1.1. Data

The 12-lead ECG signal is chosen to be one representing a normal heart rhythm (sinus rhythm). As already mentioned each of the $m = 12$ leads contains $n = 5000$ data points. One heartbeat (PQRST complex) is of approximate length of 300 data points. That's why, to assess the imputation accuracy, we choose to randomly remove segments of length 300 from the signal, and construct datasets with 20%, 30%, and 40% missing data. We compare the results for each dataset against two other baseline sequential methods used in Akyildiz et al. (2021). The first, which we call MLE-SMF, is a maximum likelihood estimation (MLE) method for online probabilistic matrix factorization where the state transition matrix C remains constant over time. This builds on prior work by Yildirim et al. (2012), Sun et al. (2012), and Févotte et al. (2013). The second approach adapts the temporal matrix factorization (TMF) optimization technique proposed by Yu et al. (2016).

[1] See: https://github.com/JoeJoe1313/rPSMF

3.1.2. Models

For PSMF and rPSMF we choose a random walk subspace model $f_\theta(x) = x$, and for the imputation we use the final estimates of C and X. We test the four sequential models by setting the rank first to $r = 3$, and then to $r = 10$. As in Akyildiz et al. (2021), for TMF we set the weight matrix to identity, for PSMF and MLE-SMF we set $R_k := R = \rho \otimes I_m$, where $\rho = 10$, $P_0 = I_r$, $Q_k := Q = q \otimes I_r$, and $q = 0.1$. For rPSMF we use $R_0 = R$ and $Q_0 = Q$ and choose $\lambda_0 = 1.8$, and for both PSMF and rPSMF we set $V_0 = v_0 \otimes I_r$ with $v_0 = 2$. We evaluate the performance of the four methods PSMF, rPSMF, MLE-SMF, and TMF by running each of them for two epochs. To ensure the robustness and reliability of our findings, we conduct the experiments 100 times, each time using different initialization and randomly generated patterns of missing data.

3.1.3. Results

Table 3.1 presents the results for rank $r = 3$ across scenarios with 20%, 30%, and 40% missing data. The findings show that PSMF and rPSMF achieve lower imputation Root Mean Square Errors (RMSEs) than MLE-SMF and TMF, with rPSMF slightly outperforming PSMF. For rank $r = 10$, as shown in Table 3.2, PSMF emerges as the top performer across all test cases. While MLE-SMF and TMF results remain relatively consistent with their rank $r = 3$ performance, rPSMF shows a significant decline in performance, severely worsening as the percentage of missing data increases. PSMF's performance is also worse than for $r = 3$ but it is not that much affected as rPSMF. This observation contradicts our initial hypothesis that higher rank might lead to lower imputation RMSE, and is caused by overfitting.

$r = 3$

	Imputation RMSE			Runtime (s)		
	20%	30%	40%	20%	30%	40%
PSMF	104.46 (32.19)	115.99 (31.61)	138.18 (31.77)	0.58	0.59	0.58
rPSMF	**98.51** (26.56)	**109.04** (29.08)	**124.91** (29.58)	0.65	0.64	0.64
MLE-SMF	271.04 (34.07)	263.65 (33.46)	244.21 (28.69)	0.50	0.51	0.50
TMF	202.76 (16.18)	202.90 (15.97)	208.98 (19.24)	0.28	0.28	0.27

Table 3.1.: Imputation error and runtime using 20%, 30% and 40% missing values, rank 3, averaged over 100 random repetitions.

$r = 10$

	Imputation RMSE			Runtime (s)		
	20%	30%	40%	20%	30%	40%
PSMF	**122.79** (38.89)	**154.79** (45.69)	**186.31** (52.85)	0.92	1.07	1.07
rPSMF	192.19 (71.96)	253.77 (92.27)	377.75 (132.56)	1.10	1.08	1.07
MLE-SMF	268.57 (21.63)	267.81 (30.37)	270.77 (33.82)	0.82	0.90	0.90
TMF	205.67 (20.04)	199.81 (17.31)	193.28 (20.25)	0.42	0.47	0.44

Table 3.2.: Imputation error and runtime using 20%, 30% and 40% missing values, rank 10, averaged over 100 random repetitions.

We can evaluate the effectiveness of our uncertainty quantification by calculating the percentage of missing values that fall within a two-standard-deviation (2σ) interval of the approximate posterior distribution. The results, presented in Table 3.3, show that rPSMF has the most coverage compared to PSMF and MLE-SMF (TMF does not provide a posterior distribution). While rPSMF has the highest coverage we can notice that it lowers when the rank is higher. For PSMF the coverage improves as the rank is higher. MLE-SMF has the same coverage for both ranks.

$r = 3$

Missing %:	20%	30%	40%
PSMF	0.14	0.12	0.10
rPSMF	**0.80**	**0.75**	**0.66**
MLE-SMF	0.07	0.06	0.06

$r = 10$

Missing %:	20%	30%	40%
PSMF	0.24	0.18	0.14
rPSMF	**0.60**	**0.50**	**0.37**
MLE-SMF	0.07	0.06	0.05

Table 3.3.: Average coverage proportion of the missing data by the 2σ uncertainty bars of the posterior predictive estimates, averaged over 100 repetitions. Left: results for rank 3, right: results for rank 10.

In Figure 3.1 we can see reconstructions of the first 1500 data points of the 12-lead ECG signal with rank $r = 3$ and 20% missing data achieved by applying PSMF (left) and rPSMF (right). The parts in orange correspond to the segments of length 300 we have randomly removed, the red parts are the non-missing ones from the original signal, and the reconstruction is shown with a blue dashed line. There are missing parts containing a whole heartbeat, parts of the heartbeat, or parts between two heartbeats. We can observe that both PSMF and rPSMF manage to capture the dynamics of the signal, with PSMF visually doing a better job with the high amplitude of the R wave in missing heartbeats, especially in the beginning of the signal. In Figure 3.2 we can see reconstructions of the same 12-lead ECG signal but with rank $r = 10$ and 20% missing data achieved by applying PSMF (left) and rPSMF (right). It is noticeable that PSMF has generally worse reconstruction than for rank $r = 3$, with some of the R wave amplitudes being too low or too high. For rPSMF in some of the leads we can see very big discrep-

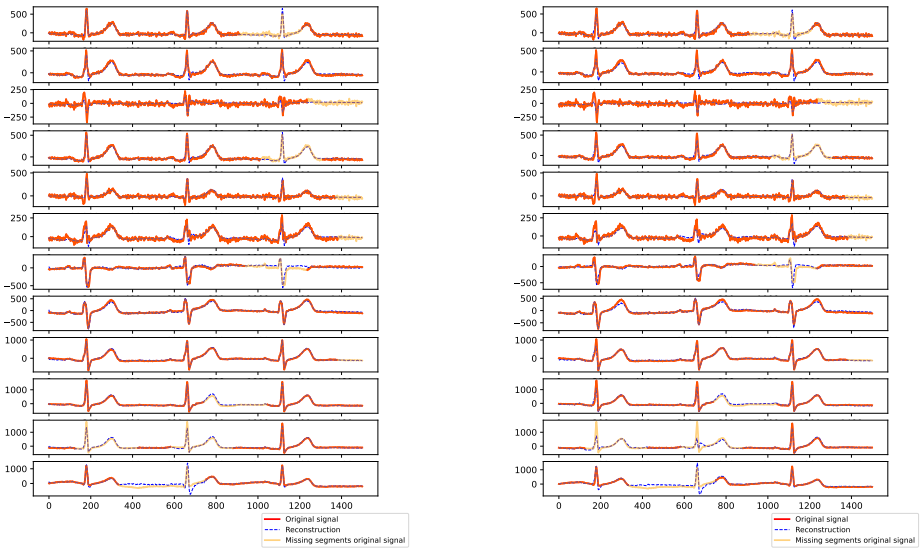

Figure 3.1.: Reconstruction of the first 1500 data points with rank 3 of 20% missing data. Left: PSMF; Right: rPSMF.

ancies between the reconstruction and the original signal. All of the visualisations for all 5000 data points for PSMF, rPSMF, MLE-SMF, and TMF for both ranks 3 and 10 and the three scenarios with 20%, 30%, and 40% missing data can be found in Appendix A.

3.1.4. Conclusion

In this section, we evaluated the performance of PSMF and rPSMF on the task of imputing missing values in 12-lead ECG time-series data, comparing them against MLE-SMF and TMF as baselines. Our experiments, conducted across various missing data scenarios and matrix ranks, reveal that PSMF and rPSMF generally outperform the baseline methods in terms of lower imputation RMSE, particularly at a lower rank of $r = 3$. Notably, rPSMF consistently achieved the best performance for rank 3 across all levels of missing data, though its efficacy diminished significantly at the higher rank of $r = 10$, especially as the proportion of missing data increased. Additionally, our uncertainty quantification analysis indicated that rPSMF provided the highest coverage of the missing data within a 2σ interval, although this coverage decreased with increasing rank. Visual inspection of the reconstructions further corroborated the quantitative findings, with PSMF showing superior performance, particularly in retaining the high amplitude of the R wave. Overall, these results highlight the strengths and limitations of both PSMF and rPSMF in handling missing data in ECG signals and suggest that rank selection plays a crucial role in their effectiveness.

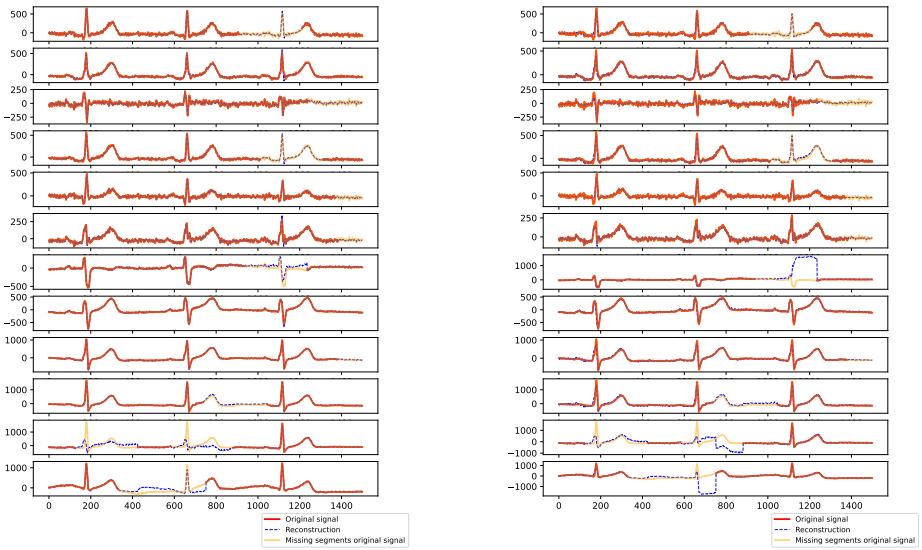

Figure 3.2.: Reconstruction of the first 1500 data points with rank 10, 20% coverage. Left: PSMF; Right: rPSMF.

3.2. R-peaks Detection

As discussed in Section 2.5.3, precise R-peak detection is crucial for accurate ECG data interpretation and cardiac condition diagnosis. Our numerous unsuccessful attempts to forecast the high-amplitude, sharp R waves with PSMF resulted in predictions with lower amplitudes and smoother peaks. This consistent forecasting challenge inspired us to explore a novel approach: leveraging this prediction discrepancy as a method for R-peak detection.

3.2.1. R-peaks Detection Method

Here, we outline the main approach steps for R-peak detection:

- Apply PSMF to achieve smoother lower amplitude R wave forecast
- Remove the reconstruction from the original data
- Define a threshold as a percentile of the differences between the reconstruction and the original data
- Determine the points above this threshold as R-peaks

3.2.2. Data

For this experiment we are going to work with the same sinus rhythm signal as in Section 3.1. For demonstrating the approach steps we are going to show only the results for the Lead II signal.

To ease the computational resources needed we are going to work with a resampled version of the signal. The resampling would make the original signal (Figure 3.3) from containing 5000 points to containing only 1500 points (Figure 3.4). This is achieved by first applying Fast Fourier Transform (FFT) to transform the original signal into the frequency domain, then removing coefficients from the higher frequency components, effectively performing a low-pass filter, and finally applying an Inverse Fast Fourier Transform (IFFT) to convert the modified frequency domain signal back to the time domain. For clarity, this is done by using the `resample` function from the Python library `scipy.signal`. Observing Figure 3.3 and Figure 3.4 confirms that the original structure and important characteristics of the data are preserved.

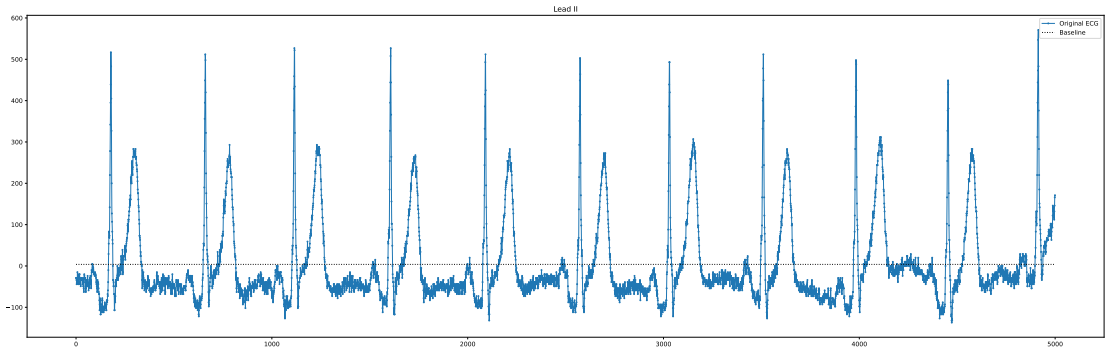

Figure 3.3.: Original Lead II signal, 5000 points.

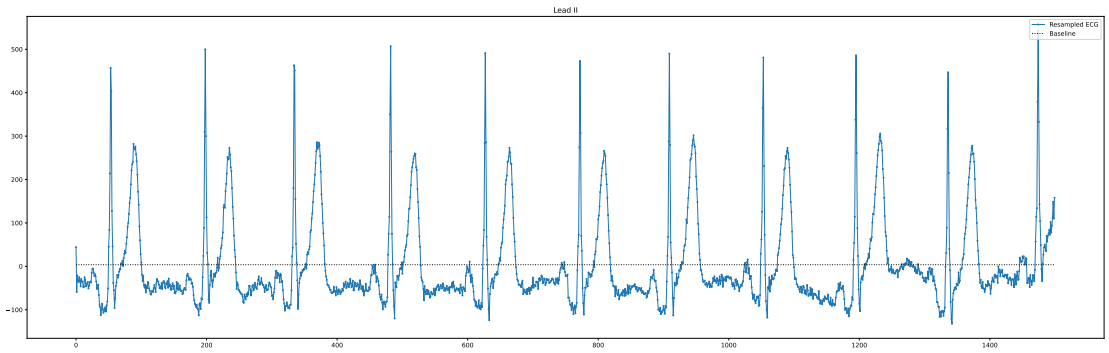

Figure 3.4.: Resampled Lead II signal, 1500 points.

After that, the data is denoised via applying wavelet transform. First, the signal is decomposed into wavelet coefficients at different scales using Daubechies wavelet, known for its efficacy in processing signals with sharp peaks and smooth variations, typical of ECG data. The decomposition level is set to the maximum useful level of decomposition, indicating that the signal is decomposed multiple times to extract various frequency components. This step is done using the function `wavedec` from the Python library `PyWavelets`. Next, a threshold to differentiate significant signal components from noise is calculated based on the noise level estimated from the smallest scale coefficients, using median absolute deviation to approximate the standard deviation. The coefficients are subjected to soft thresholding, where values below the threshold are set to zero, and values above the threshold are shrunk towards zero. Finally, the denoised signal is reconstructed from the thresholded wavelet coefficients using the inverse wavelet transform. This process effectively removes high-frequency noise components from the ECG signal while preserving the important signal features, resulting in a cleaner and more interpretable signal. After the data is resampled and denoised we also standardise the data. In Figure 3.5 we can see the resampled, denoised, standardised Lead II signal compared to the resampled and standardised Lead II signal.

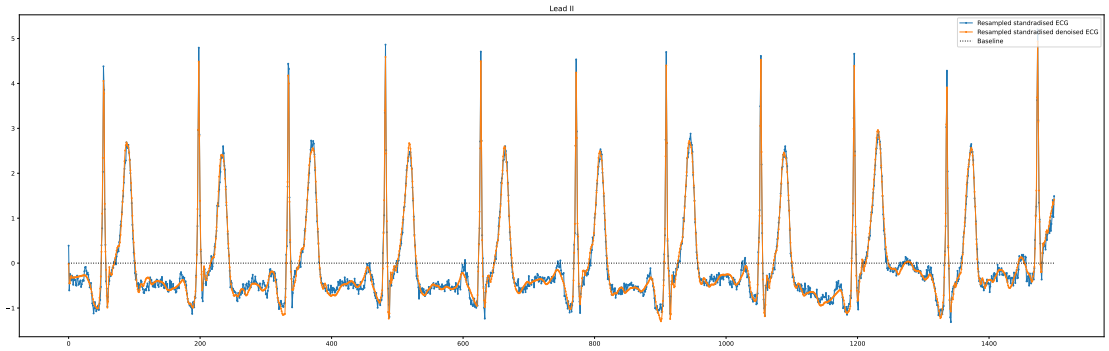

Figure 3.5.: Resampled and standardised Lead II signal (blue), and the denoised version (orange) of the same signal after applying wavelet transform and standardisation.

3.2.3. PSMF Model

To summarise, the ECG signal we want to apply PSMF to has $n = 1500$ observations, containing 11 heartbeats, and $m = 12$ variables (corresponding to the 12 leads introduced in Section 2.5.4). To model the subspace with PSMF we choose a periodic subspace model (Fourier basis)

$$x_k = f_\theta(x_{k-1}) = \sum_{i=1}^{3} \theta_{1,i} \sin(2\pi\theta_{2,i}k + \theta_{3,i}x_{k-1}) + \theta_{4,i} \cos(2\pi\theta_{5,i}k + \theta_{6,i}x_{k-1}), \qquad (3.1)$$

where $\theta_{1,i}, \theta_{4,i} \in \mathbb{R}^{r\times r}$ and $\theta_{2,i}, \theta_{3,i}, \theta_{5,i}, \theta_{6,i} \in \mathbb{R}^{r\times 1}$ for $i = 1, 2, 3$. We also set $r = 3$, $R_k = I_m$, $P_0 = I_r$, $Q_k = I_r$, $V_0 = v_0 \otimes I_r$ with $v_0 = 5$, and run iterative PSMF with 100 iterations, and withhold 10% of the data for testing purposes. In Figure 3.6 we can see the reconstruction on the training data. It is noticeable that the fit can model the smaller amplitudes, but when it comes to the R wave the amplitude is very low, close to the baseline. Another observation is that the fit is very periodic around the smoother parts which were initially very noisy. Next, we remove the reconstruction from the original data, and choose a threshold as the 99.2-th percentile of the differences between the reconstruction and the original data. Figure 3.7 visualises the differences (blue) and the chosen threshold line (red).

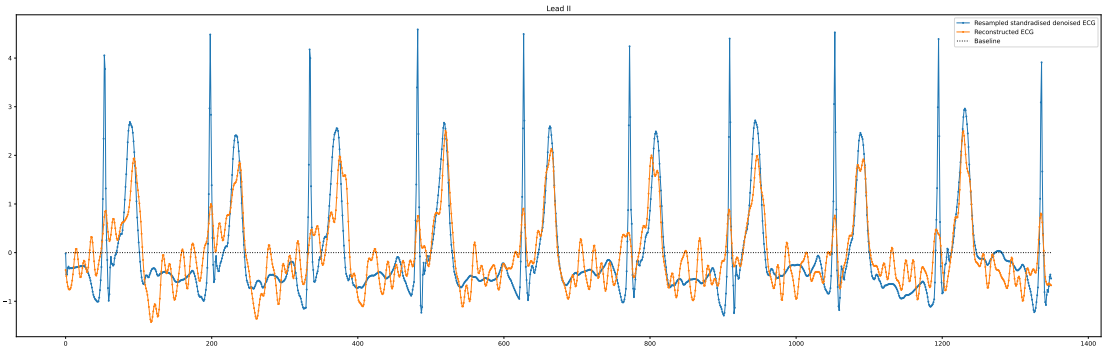

Figure 3.6.: Lead II signal (blue), and the reconstruction (orange) of the same signal after applying PSMF to the 12 leads with rank $r = 3$, Fourier basis with $n = 3$ terms, and 100 iterations.

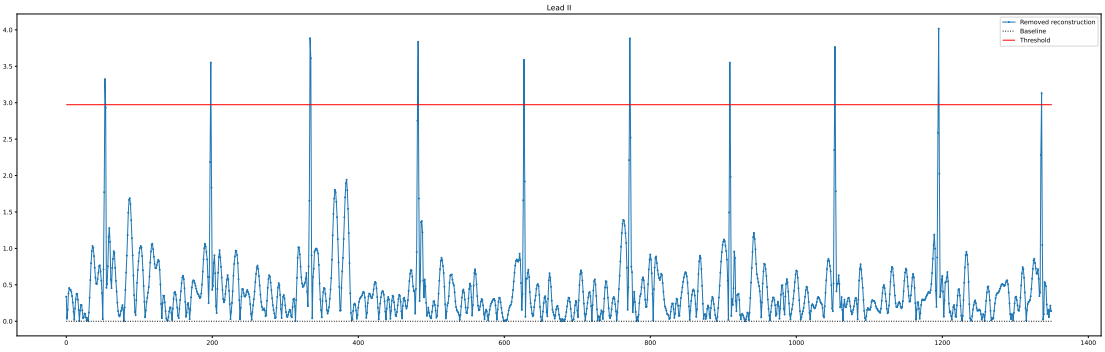

Figure 3.7.: Removed reconstruction from the modelled ECG signal (blue), and threshold (red) for isolating the R-peaks.

Finally, in Figure 3.8 we can see the detected R-peaks (red x) which were chosen as the points above the defined threshold, and the true R-peaks (circle). The method manages to determine the R-peaks, but in the third heartbeat we can notice that there is a second

point detected as a peak. This is due to the last peak having a noticeably lower amplitude than all of the other peaks, which enforced choosing a lower threshold. This is an example where variable amplitudes might contribute to difficult detection of R-peaks and possibly indicate an issue with the signal such as artifacts, poor electrode contact, etc. This highlights the potential impact of variable amplitudes on R-peak detection and the need for robust methods to handle such variations.

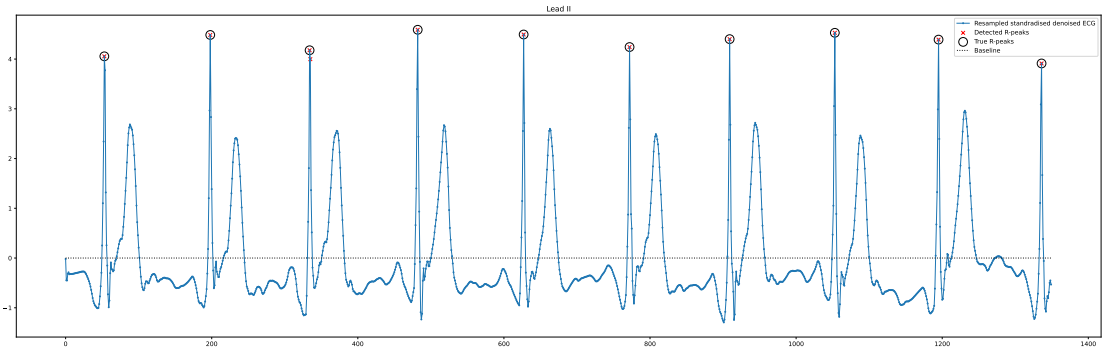

Figure 3.8.: Detected R-peaks (red x) and the true R-peaks (circle) in the resampled, standardised, denoised Lead II signal (blue).

3.2.4. Conclusion

In conclusion, the novel R-peak detection method presented in this section effectively harnesses the predictive discrepancies inherent to the PSMF model as a feature rather than a drawback. By deliberately using a model that underestimates the amplitude of R-peaks, we create a predictable deviation from the original ECG signal, which can then be quantitatively identified using a threshold-based approach. Our empirical results demonstrate that this method can reliably detect R-peaks within a resampled, standardized, and denoised ECG signal: Lead II in particular. Further refinements may be necessary to address outliers and ensure universal applicability across different types of ECG signals and conditions.

3.3. Forecasting

In this section we are going to show our attempt at forecasting 12-lead ECG data with PSMF. We apply the algorithm to 12-lead ECG data containing 2 whole heartbeats, and attempt to make a forecast for the future ECG behaviour.

3.3.1. Data

The data used and the preprocessing is the same as the one described in Section 3.2, with the only difference that this time we are going to use only 400 data points, corresponding

to 3 heartbeats from the ECG.

3.3.2. PSMF Model

The ECG signal we want to apply PSMF to has $n = 400$ observations, containing 3 heartbeats, and $m = 12$ variables (corresponding to the 12 leads introduced in Section 2.5.4). To model the subspace with PSMF we choose a Fourier basis with $n = 3$ terms

$$x_k = f_\theta(x_{k-1}) = \sum_{i=1}^{3} \theta_{1,i} \sin(2\pi\theta_{2,i}k + \theta_{3,i}x_{k-1}) + \theta_{4,i} \cos(2\pi\theta_{5,i}k + \theta_{6,i}x_{k-1}), \quad (3.2)$$

where $\theta_{1,i}, \theta_{4,i} \in \mathbb{R}^{r\times r}$ and $\theta_{2,i}, \theta_{3,i}, \theta_{5,i}, \theta_{6,i} \in \mathbb{R}^{r\times 1}$ for $i = 1, 2, 3$. We also set $r = 6$, $R_k = I_m$, $P_0 = I_r$, $Q_k = I_r$, $V_0 = v_0 \otimes I_r$ with $v_0 = 5$, and run iterative PSMF with 400 iterations, and withhold 20% of the data for prediction. In other words, we use an observed time series of length 320 (corresponding to 2 heartbeats) and a series of unobserved future data of length 80 (corresponding to approximately 1 heartbeat).

3.3.3. Results

In Figure 3.9 we can see the PSMF fitting on 12-lead ECG data with rank $r = 6$, $n = 3$ Fourier terms, over 400 iterations. The observed time series is shown in blue, the unobserved future data is shown in yellow, and the reconstruction from the model is shown in red. We can see, as mentioned in Section 3.2, that the model has a hard time with modelling the high amplitudes. This also affects the forecast, although we can see that the prediction stays in reasonable bounds and manages to follow the dynamics of the signal. If we refer to Figure 3.10 we can notice that the high amplitudes with sharp peaks corresponding mainly to the R waves are described by significantly less points compared to the other waves from the PQRST complex. This is due to the R wave representing the rapid depolarization of the ventricles, particularly the left ventricle. This quick process creates a sharp, narrow spike on the ECG with high amplitude but short duration, requiring fewer points to capture it compared to slower waves like the P or T waves. These fewer points in the part towards the peak might be considered by the PSMF as outliers, or it might just be the case that the ECG data is too complex for the PSMF model.

Figure 3.11 shows how the Frobenius norm $\|Y - CX\|_F^2$ between the reconstructed data and the true data behaves with the number of iterations. The training loss starts very high but rapidly decreases in the first few iterations, stabilizing around a lower value by approximately 50 iterations. This indicates that the model is quickly learning to fit the training data. After the initial drop, the training loss continues to decrease gradually but remains relatively stable, indicating that the model has reached a point where additional iterations yield only marginal improvements. The prediction loss is initially lower than the training loss but shows more fluctuation. After about 50 iterations, it starts

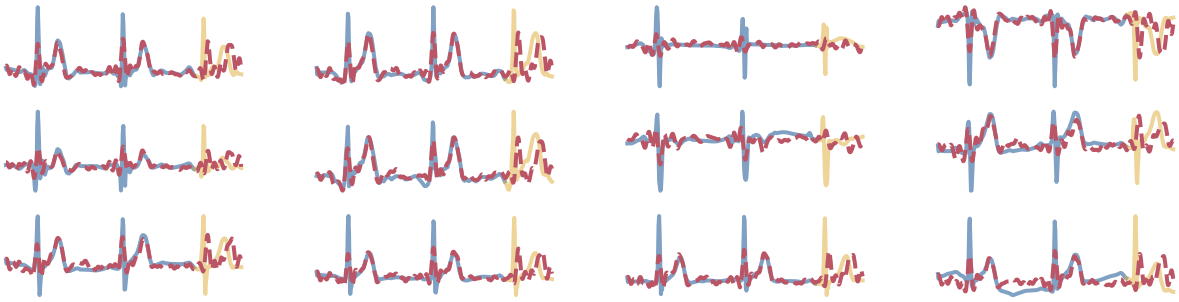

Figure 3.9.: Fitting PSMF on 12-lead ECG data with rank $r = 6$, $n = 3$ Fourier terms, 400 iterations. Observed time series (blue) with unobserved future data (yellow) and the reconstruction from the model (red).

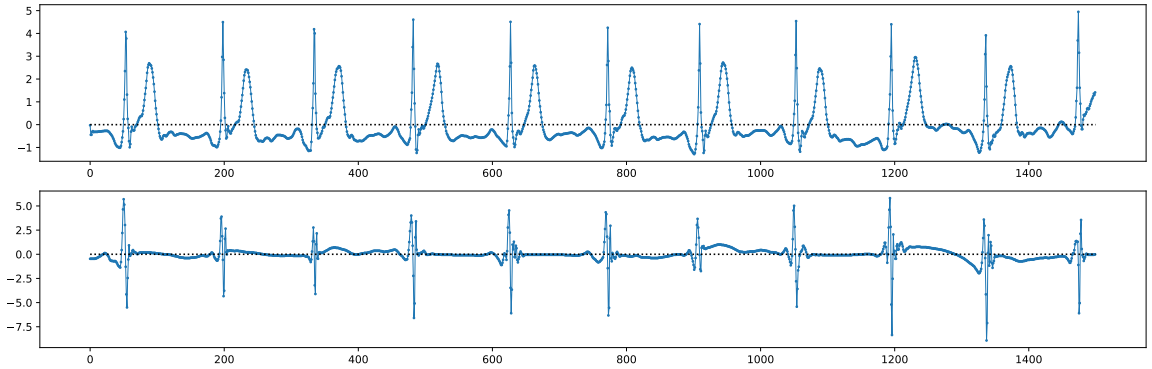

Figure 3.10.: A visualisation of the high sharp-peak amplitudes being described by fewer data points compared to the other waves.

to stabilize as well, though it remains consistently higher than the training loss. This gap between the training and prediction loss suggests that the model may be overfitting the training data, as it performs better on the training set than on the prediction task. Experimenting with stopping the training earlier resulted in predictions which had worse wave amplitudes and were shifted compared to the original data. Finally, Figure 3.12 shows how the underlying subspace is recovered.

3.3.4. Conclusion

In this section, we applied PSMF trained on two heartbeats to forecast a 12-lead ECG signal. The results showed that while the model could reasonably follow the dynamics of the ECG signal, it struggled with accurately capturing high-amplitude sharp peaks, particularly the R waves. The training and prediction loss analysis indicated that the model

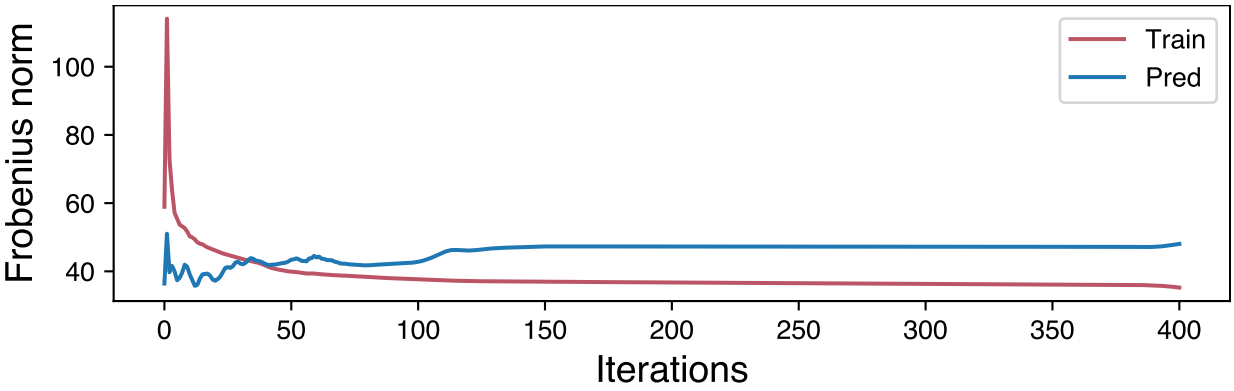

Figure 3.11.: Reconstruction error $\|Y - CX\|_F^2$ for the observed data (red) and the unobserved future data (blue).

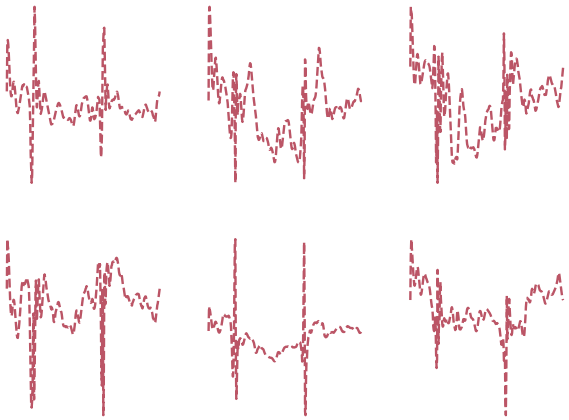

Figure 3.12.: Recovered subspace.

might be overfitting, as it performed better on the training data than on the unobserved future data. Despite these challenges, the PSMF model maintained predictions within reasonable bounds, though with some limitations in amplitude accuracy and timing.

4. Conclusion

In this thesis, we presented novel applications of the probabilistic sequential matrix factorization (PSMF) algorithm to 12-lead ECG data. Our goal was to explore the potential of PSMF in addressing tasks related to this complex, high-dimensional time-series data with nonlinear subspace.

We began by introducing the PSMF model and its robust variant, rPSMF, along with the relevant mathematical framework. We also outlined the inference process, parameter estimation techniques, and the algorithms for both PSMF and rPSMF. Additionally, we discussed the adaptation of PSMF for handling missing data.

In Chapter 3, we demonstrated three applications of PSMF to 12-lead ECG data:

- **Missing data imputation:** We applied both PSMF and rPSMF to impute missing data in ECG signals and compared their performance with other probabilistic sequential matrix factorization algorithms. Our results showed that PSMF and rPSMF generally outperformed the baseline methods, particularly at lower matrix ranks. We also concluded that PSMF and rPSMF perform better when the rank is lower.
- **R-peak detection:** We introduced a novel approach for R-peak detection by leveraging the discrepancy between the original ECG signal and the PSMF reconstruction. By removing the smoother reconstructed signal from the original data and applying a suitable threshold, we successfully identified R-peaks in the ECG signal.
- **Forecasting:** We attempted to forecast an ECG component based on previous normal heartbeats by incorporating a Fourier basis with multiple terms and rank higher than one. While the PSMF model captured the general dynamics of the ECG signal, it faced challenges in accurately modeling the high-amplitude, sharp peaks associated with R waves.

Our experiments, conducted on a comprehensive, high-quality 12-lead ECG dataset, demonstrated the potential of PSMF in handling complex ECG data. However, we also identified limitations and challenges, particularly in forecasting tasks involving rapid depolarization events like R waves.

In conclusion, this thesis highlights the applicability of PSMF to real-world ECG data and its effectiveness in tasks such as missing data imputation and R-peak detection.

However, further research is needed to refine the model and improve its performance in forecasting complex ECG features.

4.1. Future work

Future work could focus on model enhancements, probabilistic extensions, alternative inference techniques, and real-time applications to fully harness the potential of PSMF in ECG analysis and monitoring. Given the findings of this thesis, one promising direction is to refine the PSMF model to better handle the sharp, high-amplitude peaks in ECG signals, particularly the R waves. This could involve modifying the periodic subspace model to account for rapid depolarization events more effectively. If this cannot be achieved, the PSMF model can be further extended by adopting different likelihoods which can better capture such complex signal dynamics. Another possibility is to work on enhancing the computational efficiency, and provide new computational tools for inference. By addressing these challenges and expanding the scope of PSMF's application, we can unlock its full potential in ECG analysis and beyond.

Bibliography

Ömer Deniz Akyildiz and Joaquín Míguez. Dictionary filtering: a probabilistic approach to online matrix factorisation. *Signal, Image and Video Processing*, 13(4):737–744, 2019.

Ömer Deniz Akyildiz, Gerrit J. J. van den Burg, Theodoros Damoulas, and Mark F. J. Steel. Probabilistic sequential matrix factorization, 2021.

B.D.O. Anderson and J.B. Moore. *Optimal Filtering*. Information and system sciences series. Prentice-Hall, 1979. ISBN 9780136381228. URL https://books.google.bg/books?id=1o0oAQAAMAAJ.

Cédric Févotte, Jonathan Le Roux, and John R. Hershey. Non-negative dynamical system with application to speech and audio. In *2013 IEEE International Conference on Acoustics, Speech and Signal Processing*, pages 3158–3162, 2013. doi: 10.1109/ICASSP.2013.6638240.

Ary L. Goldberger, Luis A. N. Amaral, Leon Glass, Jeffrey M. Hausdorff, Plamen Ch. Ivanov, Roger G. Mark, Joseph E. Mietus, George B. Moody, Chung-Kang Peng, and H. Eugene Stanley. Physiobank, physiotoolkit, and physionet: Components of a new research resource for complex physiologic signals. *Circulation [Online]*, 101(23): e215–e220, 2000.

A. K. (Arjun K.) Gupta and D. K. Nagar. *Matrix variate distributions*. Chapman Hall/CRC monographs and surveys in pure and applied mathematics ; 104. Chapman Hall, Boca Raton, FL, 2000. ISBN 1584880465.

David A. Harville. *Matrix algebra from a statistician's perspective*. Springer, New York ;, 1997. ISBN 038794978X.

Richard E. Klabunde. *Cardiovascular physiology concepts*. Wolters Kluwer Health, Philadelphia, second edition. edition, 2012 - 2012. ISBN 9781451113846.

J.D. McLean, S.F. Schmidt, L.A. McGee, United States. National Aeronautics, and Space Administration. *Optimal Filtering and Linear Prediction Applied to a Mid-course Navigation System for the Circumlunar Mission*. NASA technical note. National Aeronautics and Space Administration, 1962. URL https://books.google.bg/books?id=78URnS1ePAgC.

Andriy Mnih and Russ R Salakhutdinov. Probabilistic matrix factorization. In J. Platt, D. Koller, Y. Singer, and S. Roweis, editors, *Advances in Neural Information Processing Systems*, volume 20. Curran Associates, Inc., 2007.

G. W. Stewart. On the early history of the singular value decomposition. *SIAM Review*, 35(4):551–566, 1993. ISSN 00361445, 10957200. URL http://www.jstor.org/stable/2132388.

John Z. Sun, Kush R. Varshney, and Karthik Subbian. Dynamic matrix factorization: A state space approach. In *2012 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 1897–1900, 2012. doi: 10.1109/ICASSP.2012.6288274.

Keith Wesley and Robert J. Huszar. *Huszar's ECG and 12-lead interpretation.* Elsevier, Amsterdam, sixth edition / keith wesley, md, facep. edition, 2021.

Sinan Yildirim, A. Taylan Cemgil, and Sumeetpal S. Singh. An online expectation-maximisation algorithm for nonnegative matrix factorisation models. *IFAC Proceedings Volumes*, 45(16):494–499, 2012. ISSN 1474-6670. doi: https://doi.org/10.3182/20120711-3-BE-2027.00312. URL https://www.sciencedirect.com/science/article/pii/S1474667015379994. 16th IFAC Symposium on System Identification.

Hsiang-Fu Yu, Nikhil Rao, and Inderjit S Dhillon. Temporal regularized matrix factorization for high-dimensional time series prediction. In D. Lee, M. Sugiyama, U. Luxburg, I. Guyon, and R. Garnett, editors, *Advances in Neural Information Processing Systems*, volume 29. Curran Associates, Inc., 2016. URL https://proceedings.neurips.cc/paper_files/paper/2016/file/85422afb467e9456013a2a51d4dff702-Paper.pdf.

Guo H. Chu H. Zheng, J. A large scale 12-lead electrocardiogram database for arrhythmia study (version 1.0.0). *PhysioNet*, 2022. URL https://doi.org/10.13026/wgex-er52.

Jianwei Zheng, Huimin Chu, Daniele Struppa, Jianming Zhang, Sir Magdi Yacoub, Hesham El-Askary, Anthony Chang, Louis Ehwerhemuepha, Islam Abudayyeh, Alexander Barrett, Guohua Fu, Hai Yao, Dongbo Li, Hangyuan Guo, and Cyril Rakovski. Optimal multi-stage arrhythmia classification approach. *Scientific Reports*, 10(1):2898, 2020.

A. Missing Data Imputation Results

Full results of testing on the 20%, 30%, and 40% missing data with both $r = 3$ and $r = 10$.

A.1. PSMF

A.1.1. $r = 3$

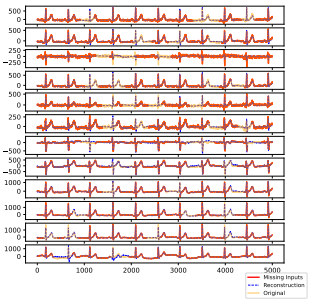

Figure A.1.: 20% missing data.

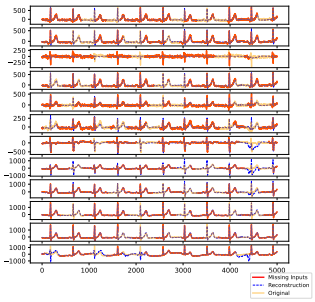

Figure A.2.: 30% missing data.

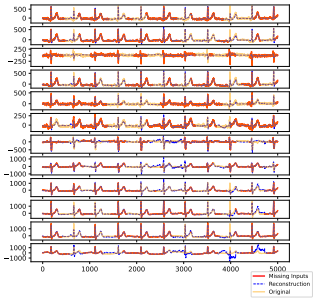

Figure A.3.: 40% missing data.

A.1.2. $r = 10$

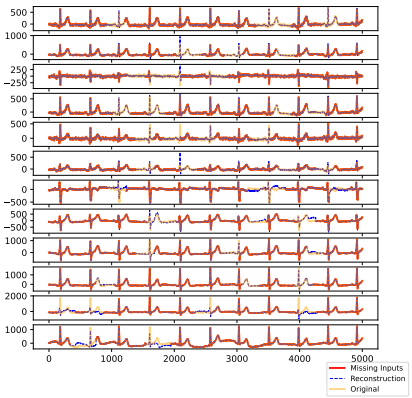

Figure A.4.: 20% missing data.

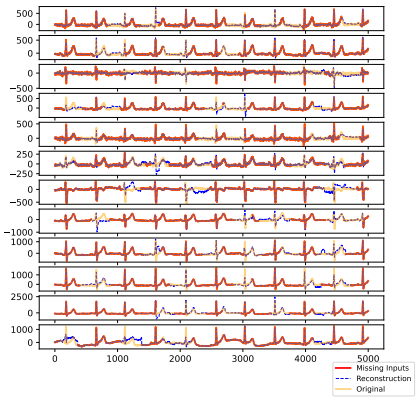

Figure A.5.: 30% missing data.

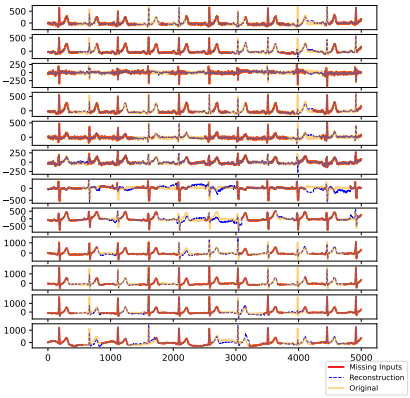

Figure A.6.: 40% missing data.

A.2. rPSMF

A.2.1. $r = 3$

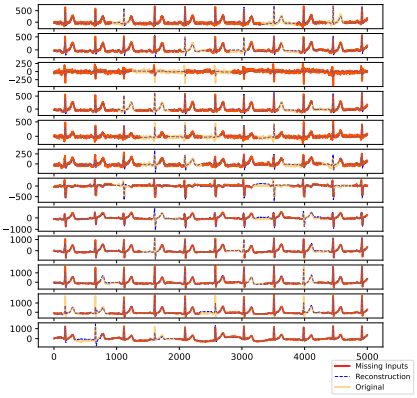

Figure A.7.: 20% missing data.

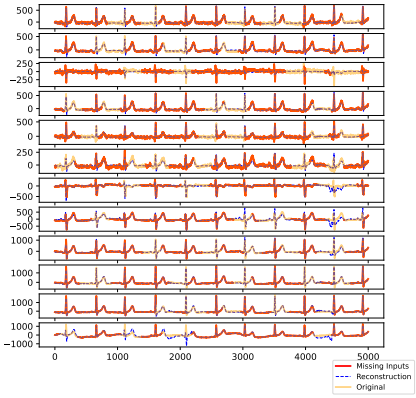

Figure A.8.: 30% missing data.

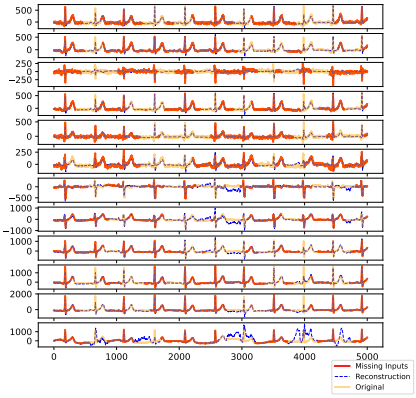

Figure A.9.: 40% missing data.

A.2.2. $r = 10$

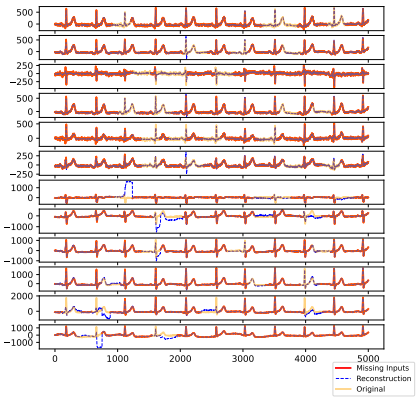

Figure A.10.: 20% missing data.

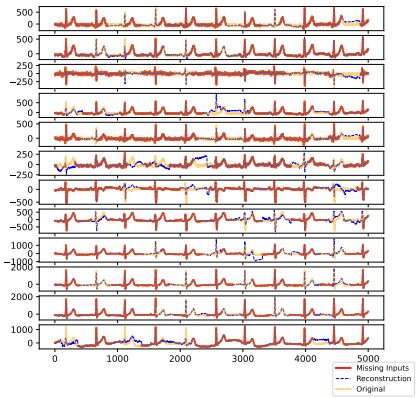

Figure A.11.: 30% missing data.

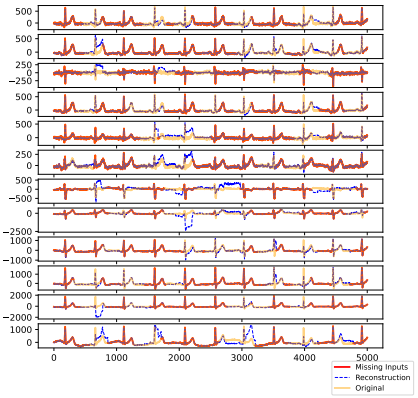

Figure A.12.: 40% missing data.

A.3. MLE-SMF

A.3.1. $r = 3$

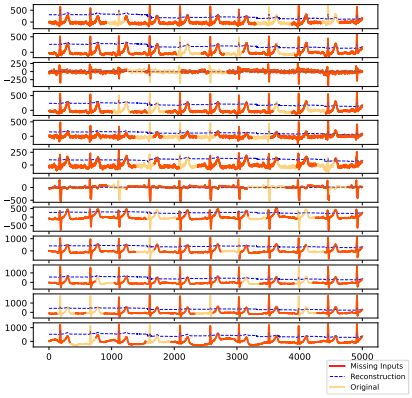

Figure A.13.: 20% missing data.

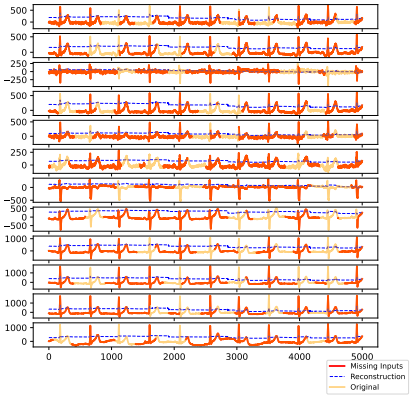

Figure A.14.: 30% missing data.

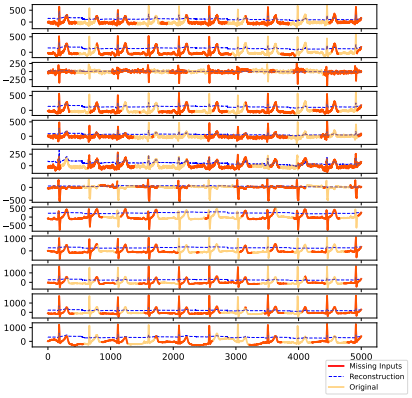

Figure A.15.: 40% missing data.

A.3.2. $r = 10$

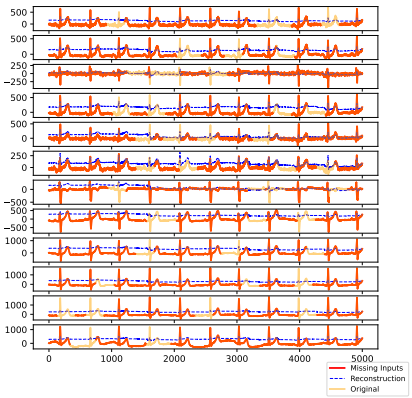

Figure A.16.: 20% missing data.

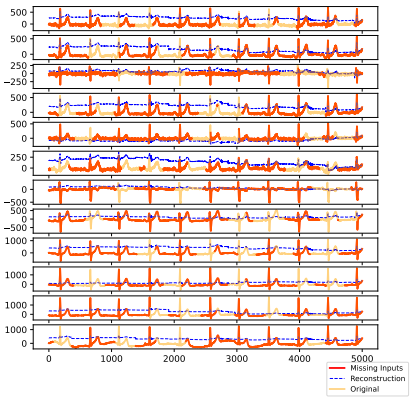

Figure A.17.: 30% missing data.

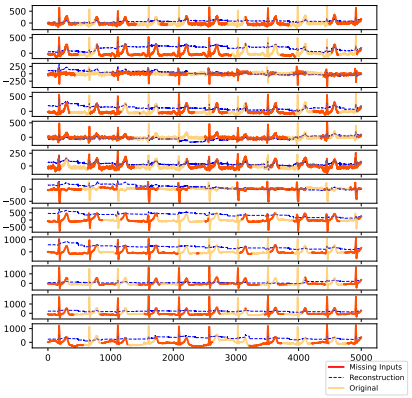

Figure A.18.: 40% missing data.

A.4. TMF

A.4.1. $r = 3$

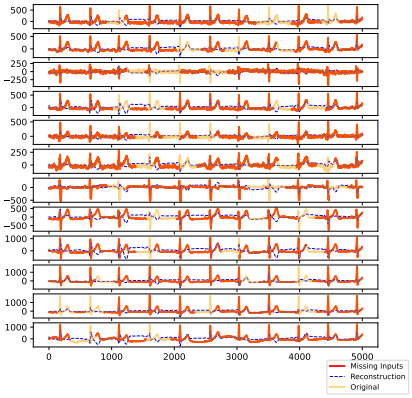

Figure A.19.: 20% missing data.

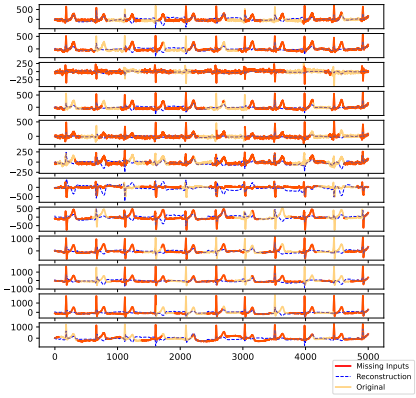

Figure A.20.: 30% missing data.

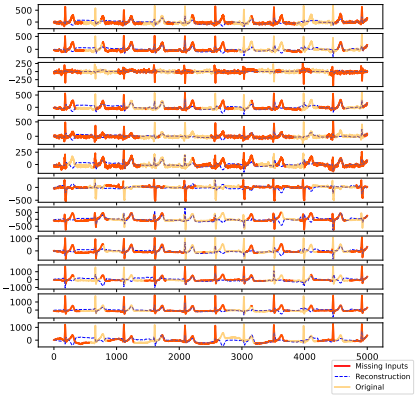

Figure A.21.: 40% missing data.

A.4.2. $r = 10$

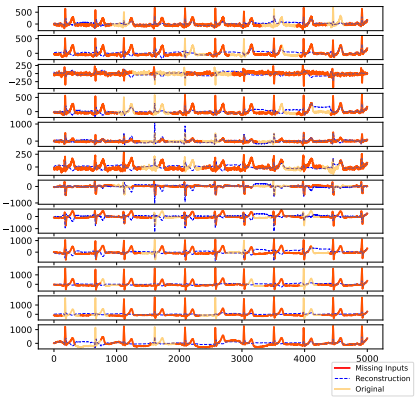

Figure A.22.: 20% missing data.

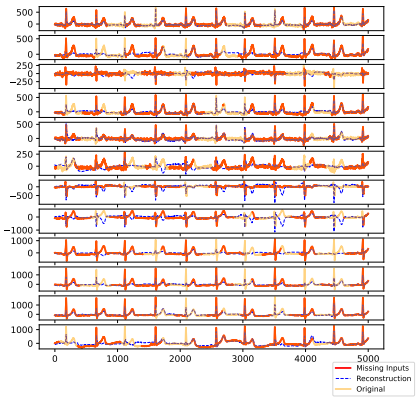

Figure A.23.: 30% missing data.

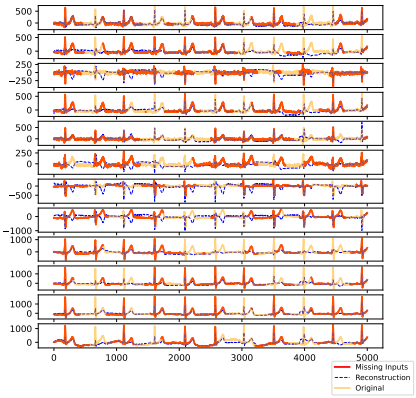

Figure A.24.: 40% missing data.